bullx cluster suite

Administrator's Guide

extreme computing

REFERENCE
86 A2 20FA 02

extreme computing

bullx cluster suite

Administrator's Guide

Software

July 2009

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

REFERENCE
86 A2 20FA 02

Printed in France

Trademarks and Acknowledgements

Preface

Scope and Objectives

The purpose of this guide is to explain how to configure and manage **Bull extreme computing** clusters, using the administration tools recommended by Bull.

It is not in the scope of this guide to describe **Linux** administration functions in depth. For this information, please refer to the standard Linux distribution documentation.

Intended Readers

This guide is for administrators of bullx cluster suite system.

Prerequisites

The installation of all hardware and software components must have been completed.

Structure

This guide is organized as follows:

Chapter 1. Lists Bull's Cluster Management functions and products.

Chapter 2. *Initial Configuration Tasks*
Describes some basic configuration tasks including password management, security settings and running parallel commands.

Chapter 3. *Cluster Database Management*
Describes the commands and the tools which enable the administrator to display and to change the Cluster Database.

Chapter 4. *Software Deployment*
Describes how to use **KSIS** to deploy, manage, modify and check software images.

Chapter 5. *Kerberos – Network Authentication Protocol*
Describes how to set up and use **Kerberos** for cluster security.

Chapter 6. *Storage Devices Management*
Explains how to setup the management environment for storage devices, and how to use storage management services.

Chapter 7. *Parallel File Systems*
Explains how these file systems operate on a Bull HPC system. It describes in detail how to install, configure and manage the **Lustre** file system.

Chapter 8. *Resource Management*
Explains how the **SLURM** Resource Manager works to help ensure the optimal management of cluster resources.

Chapter 9. *Batch Management with PBS Professional*
Describes post installation checks and some useful commands for the **PBS Professiona**l Batch Manager.

Chapter 10. *Monitoring*
Describes the **Bull System Manager - HPC Edition** monitoring tool for Bull HPC systems.

Glossary Some of the acronyms and glossary terms for **bullx cluster suite** are detailed in the Glossary

Bibliography

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

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

- *bullx cluster suite Installation and Configuration Guide* (86 A2 19FA)
- *bullx cluster suite Administrator's Guide* (86 A2 20FA)
- *bullx cluster suite User's Guide* (86 A2 22FA)
- *bullx cluster suite Maintenance Guide* (86 A2 24FA)
- *bullx cluster suite Application Tuning Guide* (86 A2 23FA)
- *bullx cluster suite High Availability Guide* (86 A2 25FA)
- *InfiniBand Guide* (86 A2 42FD)
- *LDAP Authentication Guide* (86 A2 41FD)

The following document is delivered separately:

- The *Software Release Bulletin* (SRB) (86 A2 73EJ)

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

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

For clusters which use the **PBS Professional** Batch Manager:

- *PBS Professional 10.0 Administrator's Guide (*on the *PBS Professional CD-ROM)*
- *PBS Professional 10.0 User's Guide* (on the *PBS Professional CD-ROM*)

For clusters which use **LSF**:

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

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

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

Highlighting

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

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

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

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

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

  COM1

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

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

- The use of *Italics* identifies publications, chapters, sections, figures, and tables that are referenced.

- < > identifies parameters to be supplied by the user, for example:

  <node_name>

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

Table of Contents

Chapter 1. Cluster Management Functions and Corresponding Products 1-1

Chapter 2. Initial Configuration Tasks 2-1

2.1 Configuring Services 2-1

2.2 Modifying Passwords and Creating Users 2-2

2.3 Configuring Security 2-3
2.3.1 Setting up SSH 2-3

2.4 Running Parallel Commands with pdsh 2-4
2.4.1 Using pdsh 2-4
2.4.2 Using pdcp 2-7
2.4.3 Using dshbak 2-7

2.5 Day to Day Maintenance Operations 2-9

Chapter 3. Cluster Database Management 3-1

3.1 Architecture of ClusterDB 3-1

3.2 ClusterDB Administrator 3-2

3.3 Using Commands 3-2
3.3.1 ChangeOwnerProperties 3-3
3.3.2 dbmConfig 3-5
3.3.3 dbmCluster 3-7
3.3.4 dbmNode 3-8
3.3.5 dbmHwManager 3-11
3.3.6 dbmGroup 3-12
3.3.7 dbmEthernet 3-15
3.3.8 dbmIconnect 3-16
3.3.9 dbmTalim 3-17
3.3.10 dbmSerial 3-18
3.3.11 dbmFiberChannel 3-20
3.3.12 dbmServices 3-21
3.3.13 dbmDiskArray 3-22

3.4 Managing the ClusterDB 3-24
3.4.1 Saving and Restoring the Database 3-24
3.4.2 Starting and Stopping PostgreSQL 3-26
3.4.3 Viewing the PostgreSQL Alert Log 3-26

3.5 ClusterDB Modeling 3-27
3.5.1 Physical View of the Cluster Networks 3-27
3.5.2 Physical View of the Storage 3-35
3.5.3 Machine View 3-42
3.5.4 HWMANAGER View 3-47
3.5.5 Complementary Tables 3-49
3.5.6 Nagios View 3-51
3.5.7 Lustre View 3-53

Chapter 4. Software Deployment (KSIS) 4-1
4.1 Overview 4-1
4.2 Configuring and Verifying a Reference Node 4-2
4.3 Main Steps for Deployment 4-3
4.4 Checking Deployed Images 4-4
4.4.1 Checking Principles 4-4
4.4.2 Ksis Tests and Test Groups 4-5
4.4.3 Modifying the Checks Database 4-6
4.4.4 Examining the Check Results 4-6
4.5 Ksis Commands 4-7
4.5.1 Syntax 4-7
4.5.2 Advanced ksis create options 4-8
4.5.3 Creating the Image of the Reference Node 4-8
4.5.4 Deleting an Image or a Patch 4-9
4.5.5 Deploying an Image or a Patch 4-9
4.5.6 Removing a Patch 4-9
4.5.7 Getting Information about an Image or a Node 4-9
4.5.8 Listing Images on the Image Server 4-9
4.5.9 Listing Images by Nodes 4-10
4.6 Building a Patch 4-11
4.7 Checking Images 4-12
4.8 Importing and Exporting an Image 4-12
4.9 Rebuilding ClusterDB Data before Deploying an Image 4-13
Chapter 5. Kerberos - Network Authentication Protocol 5-1
5.1 Environment 5-1
5.1.1 Kerberos Infrastructure 5-1
5.1.2 Authentication of the SSHv2 Connections 5-1
5.2 KERBEROS Infrastructure Configuration 5-2
5.2.1 secu0 Server including KDC Server and Administration Server 5-2
5.2.2 Configuration Files 5-2
5.2.3 Creating the Kerberos Database 5-3
5.2.4 Creating the Kerberos Administrator 5-3
5.2.5 Starting the KDC Server 5-4
5.2.6 Adding Access Control List (ACL) Rights for the Kerberos Administrator Created 5-4
5.2.7 Starting the Administration Daemon 5-4
5.2.8 Creating Principals Associated with Users 5-4
5.2.9 Creating Principals Associated with Remote Kerberized Services 5-5
5.3 Configuring the secu1 Machine that hosts the Host Principal remote service 5-6
5.3.1 Generating the key associated with the Host Principal remote service 5-6
5.4 Kerberos Authentication and SSH 5-7
5.4.1 Configuring the SSH Server on the secu1machine 5-7
5.4.2 SSH Client 5-8
5.5 Troubleshooting Errors 5-9

5.6 Generating Associated Keys for Nodes of a Cluster 5-10
5.7 Modifying the Lifespan and Renewal Period for TGT Tickets 5-11
5.8 Including Addresses with Tickets 5-12

Chapter 6. Storage Device Management 6-1
6.1 Overview of Storage Device Management for Bull extreme computing clusters 6-2
6.2 Monitoring Node I/O Status 6-4
6.2.1 Managing I/O Reference Counters 6-4
6.2.2 I/O Counters Definitions 6-6
6.2.3 Managing I/O Resources 6-7
6.3 Monitoring Storage Devices 6-8
6.3.1 Bull System Manager - HPC Edition: Host and Service Monitoring for Storage Devices 6-8
6.3.2 Bull System Manager - HPC Edition: Storage & I/O Information 6-13
6.3.3 Querying the Cluster Management Data Base 6-17
6.4 Monitoring Brocade Switch Status 6-19
6.5 Managing Storage Devices with Bull CLI 6-22
6.5.1 Bull FDA Storage Systems 6-22
6.5.2 DataDirect Networks Systems - DDN Commands 6-23
6.5.3 Bull Optima1250 Storage Systems 6-25
6.5.4 EMC/Clariion (DGC) Storage Systems 6-26
6.6 Using Management Tools 6-26
6.7 Configuring Storage Devices 6-27
6.7.1 Planning Tasks 6-27
6.7.2 Deployment Service for Storage Systems 6-28
6.7.3 Understanding the Configuration Deployment Service 6-28
6.8 User Rights and Security Levels for the Storage Commands 6-32
6.8.1 Management Node 6-32
6.8.2 Other Node Types 6-33
6.8.3 Configuration Files 6-33

Chapter 7. Parallel File Systems 7-1
7.1 Parallel File Systems Overview 7-1
7.2 Lustre Overview 7-2
7.3 Lustre Administrator's Role 7-3
7.4 Planning a Lustre System 7-4
7.4.1 Data Pipelines 7-4
7.4.2 OSS / OST Distribution 7-4
7.4.3 MDS / MDT Distribution 7-4
7.4.4 File Striping 7-5
7.4.5 Lustre File System Limitations 7-5
7.5 Lustre System Management 7-6
7.5.1 The Lustre Database 7-6
7.5.2 /etc/lustre/storage.conf for Lustre Tools without ClusterDB 7-8
7.5.3 Lustre Networks 7-13

7.5.4 Lustre Management Configuration File: /etc/lustre/lustre.cfg 7-13
7.5.5 Lustre Services Definition 7-16
7.5.6 Creating Lustre File Systems 7-17

7.6 Installing and Managing Lustre File Systems 7-22
7.6.1 Installing Lustre File Systems using lustre_util 7-22
7.6.2 Removing Lustre File Systems using lustre_util 7-22
7.6.3 lustre_util Actions and Options 7-22
7.6.4 lustre_util Configuration File /etc/lustre/lustre_util.conf 7-33
7.6.5 Lustre Tuning File /etc/lustre/tuning.conf 7-35
7.6.6 Lustre File system Reconfiguration 7-36
7.6.7 Using Quotas with Lustre File Systems 7-37

7.7 Monitoring Lustre System 7-40
7.7.1 Lustre System Health Supervision 7-40
7.7.2 Lustre File system Indicator 7-42
7.7.3 Lustre System Performance Supervision 7-44

Chapter 8. SLURM Resource Manager 8-1

8.1 Resource Management with SLURM 8-2
8.1.1 SLURM Key Functions 8-2
8.1.2 SLURM Components 8-3
8.1.3 SLURM Daemons 8-3
8.1.4 Scheduler Types 8-5

8.2 SLURM Configuration 8-7
8.2.1 Configuration Parameters 8-7
8.2.2 SCONTROL – Managing the SLURM Configuration 8-8
8.2.3 Pam_Slurm Module Configuration 8-8
8.2.4 Installing and Configuring Munge for SLURM Authentication (MNGT) 8-9

8.3 Administrating Cluster Activity with SLURM 8-12
8.3.1 Starting the Daemons 8-12
8.3.2 SLURMCTLD (Controller Daemon) 8-13
8.3.3 SLURMD (Compute Node Daemon) 8-14
8.3.4 Node Selection 8-15
8.3.5 Logging 8-15
8.3.6 Corefile Format 8-16
8.3.7 Security 8-16
8.3.8 SLURM Cluster Administration Examples 8-16

Chapter 9. PBS Professional Batch Manager 9-1

9.1 Pre-requisites 9-1

9.2 Post Installation checks 9-2
9.2.1 Checking the status of the PBS daemons 9-2
9.2.2 Adding a Node to the Initial Cluster Configuration 9-2

9.3 Useful Commands for PBS Professional 9-3

9.4 PBS GridWorks Analytics 9-3
9.4.1 Viewing PBS GridWorks Analytics Data 9-3

9.5 Essential Configuration Settings for bullx cluster suite 9-4

9.5.1 MPIBull2 and PBS Professional for all clusters (InfiniBand and Ethernet) 9-4

Chapter 10. Monitoring with Bull System Manager - HPC Edition 10-1

10.1 Launching Bull System Manager - HPC Edition 10-2

10.2 Access Rights 10-3
10.2.1 Administrator Access Rights 10-3
10.2.2 Standard User Access Rights 10-3
10.2.3 Adding Users and Changing Passwords 10-3

10.3 Hosts, Services and Contacts for Nagios 10-4

10.4 Using Bull System Manager - HPC Edition 10-5
10.4.1 Bull System Manager - HPC Edition – View Levels 10-5

10.5 Map Button 10-6
10.5.1 All Status Map View 10-6
10.5.2 Rack View 10-7
10.5.3 bullx blade map view 10-7
10.5.4 Host Services detailed View 10-8
10.5.5 Control view 10-9
10.5.6 Ping Map View 10-9

10.6 Status Button 10-10

10.7 Log Window 10-11

10.8 Alerts Button 10-12
10.8.1 Active Checks 10-13
10.8.2 Passive Checks 10-14
10.8.3 Alert Definition 10-14
10.8.4 Notifications 10-14
10.8.5 Acknowledgments 10-15
10.8.6 Running a Script 10-15
10.8.7 Generating SNMP Alerts 10-16
10.8.8 Resetting an Alert Back to OK 10-16
10.8.9 nsmhpc.conf Configuration file 10-16
10.8.10 Comments 10-16

10.9 Storage Overview 10-17

10.10 Shell 10-18

10.11 Monitoring the Performance - Ganglia Statistics 10-18

10.12 Group Performance View 10-18

10.13 Global Performance View 10-19
10.13.1 Modifying the Performance Graph Views 10-20
10.13.2 Refresh Period for the Performance View Web Pages 10-21

10.14 Configuring and Modifying Nagios Services 10-21
10.14.1 Configuring Using the Database 10-21
10.14.2 Modifying Nagios Services 10-21
10.14.3 Changing the Verification Frequency 10-22
10.14.4 Nagios Services Service 10-22
10.14.5 Nagios Information 10-22

10.15 General Nagios Services 10-23
10.15.1 Ethernet Interfaces 10-23
10.15.2 Resource Manager Status 10-23
10.15.3 Hardware Status 10-23
10.15.4 Alert Log 10-23
10.15.5 I/O Status 10-23
10.15.6 Postbootchecker 10-23
10.16 Management Node Nagios Services 10-24
10.16.1 MiniSQL Daemon 10-24
10.16.2 Resource Manager Daemon 10-24
10.16.3 ClusterDB 10-24
10.16.4 Cron Daemon 10-24
10.16.5 Compute Power Available 10-24
10.16.6 Global File System bandwidth available 10-25
10.16.7 Storage Arrays available 10-25
10.16.8 Global File System Usage 10-25
10.16.9 I/O pairs Migration Alert 10-25
10.16.10 Backbone Ports Available 10-25
10.16.11 HA System Status 10-25
10.16.12 Kerberos KDC Daemon 10-25
10.16.13 Kerberos Admin Daemon 10-26
10.16.14 LDAP Daemon (Lustre clusters only) 10-26
10.16.15 Lustre file system access 10-26
10.16.16 NFS file system access 10-26
10.16.17 InfiniBand Links available 10-26
10.16.18 CMC Health 10-27
10.17 Ethernet Switch Services 10-27
10.17.1 Ethernet Interfaces 10-28
10.17.2 Fans 10-28
10.17.3 Ports 10-28
10.17.4 Power supply 10-28
10.17.5 Temperature 10-28
10.18 Cool Cabinet Door Services 10-29
10.18.1 Cool Cabinet Door Functional Status 10-29
10.18.2 Cool Cabinet Door Power Consumption 10-29
10.18.3 Cool Cabinet Door Delta Pressure 10-29
10.18.4 Cool Cabinet Door Ethernet Interfaces 10-30
10.18.5 Cool Cabinet Door Power Supply Status 10-30
10.18.6 Cool Cabinet Door Temperature Average 10-30
10.18.7 Cool Cabinet Door Valve Aperture 10-30

Glossary and Acronyms G-1

List of Figures

Figure 1-1. Bull HPC Management Functions 1-1
Figure 3-1. **BAS5 for Xeon** ClusterDB architecture 3-1
Figure 3-2. Cluster Network – diagram 1 3-27
Figure 3-3. Cluster Network – diagram 2 3-28
Figure 3-4. Storage physical view 3-35
Figure 3-5. Cluster Database – Machine view 1 3-42
Figure 3-6. Cluster Database – Machine view 2 3-43
Figure 3-7. HWManager view 3-47
Figure 3-8. Cluster Database – Complementary tables 3-49
Figure 3-9. Nagios View 3-51
Figure 3-10. Cluster Database – Lustre view 3-53
Figure 4-1. Main steps for deployment 4-3
Figure 6-1. I/O Status – initial screen 6-4
Figure 6-2. Bull System Manager HPC Edition - I/O Status Details 6-5
Figure 6-3. Bull System Manager –HPC Edition – I/O Resources of a node 6-7
Figure 6-4. Detailed service status for a storage host 6-9
Figure 6-5. Bull System Manager opening console window with the Storage overview icon circled 6-13
Figure 6-6. Storage overview 6-14
Figure 6-7. Inventory view of faulty storage systems and components 6-15
Figure 6-8. Storage detailed view 6-16
Figure 6-9. Nodes I/O Overview 6-17
Figure 6-10. Detailed Service status of a brocade switch 6-21
Figure 7-1. **Bull System Manager** - Map view 7-40
Figure 7-2. NovaScale Lustre FileSystems Status 7-42
Figure 7-3. Lustre Management Node web interface 7-43
Figure 7-4. Detailed view of Lustre file systems 7-44
Figure 7-5. Group performance global view pop up window 7-45
Figure 7-6. Dispatched performance view pop up window 7-45
Figure 7-7. Global performance view pop up window 7-46
Figure 8-1. SLURM Simplified Architecture 8-3
Figure 8-2. SLURM Architecture - Subsystems 8-4
Figure 10-1. Bull System Manager - HPC Edition opening view 10-5
Figure 10-2. **Map** button **all status** opening view 10-6
Figure 10-3. **Rack view** with the **Problems** window at the bottom 10-7
Figure 10-4. bullx blade map view 10-8
Figure 10-5. Host Service details 10-8
Figure 10-6. **Monitoring Control** Window 10-9
Figure 10-7. **Status Overview** screen 10-10
Figure 10-8. Monitoring - Log Window 10-10
Figure 10-9. Monitoring Service Status window for a host with the Log Alerts link highlighted. 10-11
Figure 10-10. **Alert Window** showing the different alert states 10-12
Figure 10-11. **Monitoring Control** Window used to set **Active Checks** for a Service 10-14
Figure 10-12. Hostgroups Reporting Notifications Window showing the Notification Levels 10-15
Figure 10-13. Storage overview window 10-18
Figure 10-14. Group Performance view 10-19
Figure 10-15. Global overview for a host (top screen) 10-20
Figure 10-16. Detailed monitoring view for a host (bottom half of screen displayed in Figure 10-15) 10-21
Figure 10-17. **Ethernet Switch** services 10-29
Figure 10-18. Cool Cabinet Door Services 10-31

List of Tables

Table 2-1. Maintenance Tools 2-9
Table 3-1. CLUSTER Table 3-29
Table 3-2. IP_NW table 3-29
Table 3-3. ETH_SWITCH Table 3-30
Table 3-4. IC_NW Table 3-30
Table 3-5. IC_SWITCH Table 3-31
Table 3-6. SERIAL_NW Table 3-31
Table 3-7. PORTSERVER Table 3-32
Table 3-8. ETH_VLAN table 3-32
Table 3-9. FC_NW table 3-32
Table 3-10. CLUSTER_IPV Table 3-32
Table 3-11. FC_SWITCH table 3-33
Table 3-12. TALIM table 3-34
Table 3-13. Storage – disk_array table 3-37
Table 3-14. Storage – da_enclosure table 3-37
Table 3-15. Storage – da_disk_slot table 3-37
Table 3-16. Storage – da_controller table 3-38
Table 3-17. Storage – da_fc_port.table 3-38
Table 3-18. Storage – da_serial_port table 3-38
Table 3-19. Storage – da_ethernet_port Table 3-39
Table 3-20. Storage – da_power_supply table 3-39
Table 3-21. Storage – da_fan table 3-39
Table 3-22. Storage – da_power_fan table 3-40
Table 3-23. Storage – da_temperature_sensor table 3-40
Table 3-24. da_io_path table 3-40
Table 3-25. Storage – da_iocell_component table 3-40
Table 3-26. Storage – da_cfg_model table 3-41
Table 3-27. Storage – da_power_port table 3-41
Table 3-28. Machine view – NODE table 3-44
Table 3-29. Machine view – NODE_IMAGE table 3-45
Table 3-30. Machine view – NODE_PROFILE table 3-45
Table 3-31. Machine view – IC_BOARD table 3-46
Table 3-32. Machine view – IPOIB Table 3-46
Table 3-33. Machine view – SDPOIB table 3-46
Table 3-34. Machine view – FC_BOARD table 3-47
Table 3-35. HWMANAGER Table 3-48
Table 3-36. ADMIN table 3-49
Table 3-37. RACK table 3-50
Table 3-38. RACK_PORT table 3-50
Table 3-39. CONFIG_CANDIDATE table 3-50
Table 3-40. CONFIG_STATUS table 3-51
Table 3-41. GROUP_NODE table 3-51
Table 3-42. SERVICES Table 3-51
Table 3-43. AVAILABILITY Table 3-52
Table 3-44. Lustre_FS table 3-54
Table 3-45. Lustre OST table 3-55
Table 3-46. Lustre_MDT Table 3-55
Table 3-47. Lustre_IO_node table 3-55
Table 3-48. Lustre_mount table 3-56

Table 4-1. Standard checks delivered with Ksis 4-5
Table 7-1. Inode Stripe Data 7-4
Table 8-1. Role Descriptions for SLURMCTLD Software Subsystems 8-4
Table 8-2. SLURMD Subsystems and Key Tasks..... 8-5
Table 8-3. SLURM Scheduler Types 8-6

Chapter 1. Cluster Management Functions and Corresponding Products

bullx cluster suite is a software distribution that includes a suite of tools and programs to run and monitor **Bull extreme computing** clusters.

The Bull cluster administration tools are centralized on one node, the Management Node. All nodes are controlled and monitored from this central management point, with the objective of ensuring that all CPU activity and network traffic, on the Compute and I/O nodes, runs as efficiently as is possible.

The administration tools are mainly Open Source products that are configurable and adaptable to the requirements of the cluster, and which can be deactivated on demand.

These products have been further developed and customised for Bull platforms and their environments. All the management functions are available through a browser interface or via a remote command mode. Users access the management functions according to their profile.

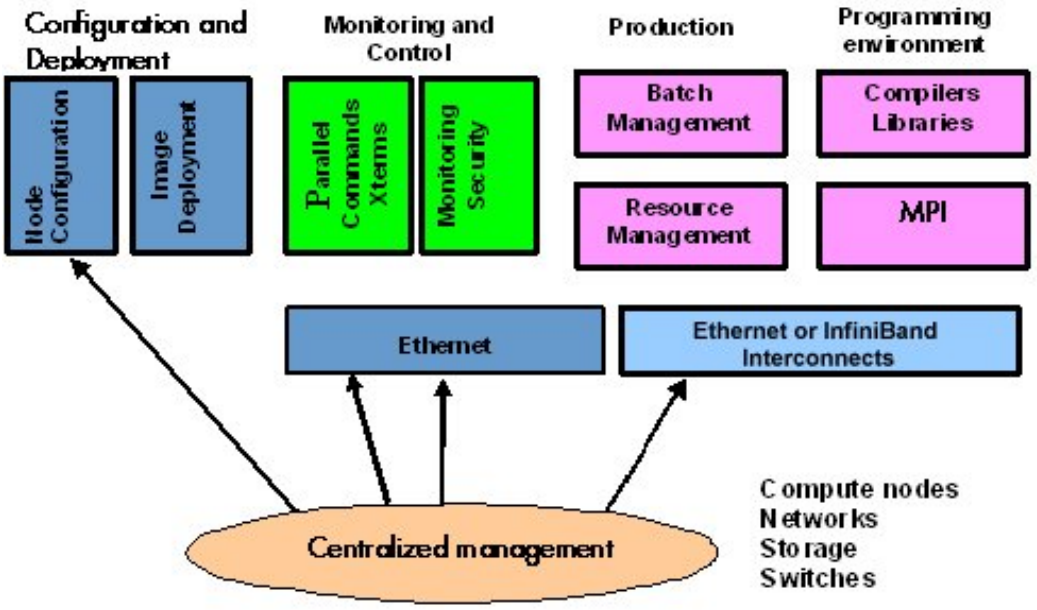


Figure 1-1. bullx cluster suite Management Functions

The management functions are performed by different products which are briefly presented below.

Configuration and Software Management

- **pdsh** is used to run parallel commands.
  See Chapter 2 – *Initial Configuration Tasks for Bull HPC Clusters* for more information.
- The Cluster DataBase - **dbmConfig**, **dbmCluster**, **dbmNode** and other commands are available to manage the Cluster Database.
  See Chapter 3 – *Cluster DataBase Management* for more information.
- **KSIS** which is used to produce and deploy node images.
  See Chapter 4 – *Software Deployment (KSIS)* for more information.
- **Kerberos** (optional) – A Security Suite used to validate users, services and machines for a whole network.
  See Chapter 5 – *Kerberos - Network Authentication Protocol* for more information.

Storage and File system Management

- Various tools are available for managing and monitoring the different storage devices which exist for Bull extreme computing Clusters.
  See Chapter 6 – *Storage Device Management* for more information.
- Parallel file systems ensure high I/O rates for clusters where data is accessed by a large number of processors at once.
  See Chapter 7 – *Parallel File Systems* for more information. This describes in detail how to install, configure and manage the **Lustre** (optional) file system.

Resource and Batch Management

- **SLURM** (Simple Linux Utilities Resource Manager) an open-source scalable resource manager.
  See Chapter 8 – *Resource Management* for more information.
- **PBS Professional** (optional) is a batch manager that is used to queue, schedule and monitor jobs.
  See Chapter 9 – *Batch Management with PBS Professional* for more information.

Important

PBS Professional does not work with **SLURM** and should only be installed on clusters which do not use **SLURM**.

Monitoring

- **Bull System Manager - HPC Edition**monitors the cluster and activity and is included in the delivery for all Bull extreme computing Clusters.
  See Chapter 10 – *Monitoring with Bull System Manager – HPC Edition* for more information.
- **HPC Toolkit** provides a set of profiling tools that help you to improve the performance of the system.

See The *Application Tuning Guide* for more details on HPC Toolkit

---

Chapter 2. Initial Configuration Tasks

Most configuration tasks are carried out at the time of installation. This chapter describes how the Administrator carries out some additional configuration tasks. It also covers the security policy for extreme computing systems.

The following topics are described:

- 2.1 *Configuring Services*
- 2.2 *Modifying Passwords and Creating Users*
- 2.3 *Configuring Security*
- 2.4 *Running Parallel Commands with pdsh*
- 2.5 *Day to Day Maintenance Operations*

See The *Installation and Configuration Guide,* which describes the initial installation and configuration steps for Bull extreme computing clusters, for more information.

2.1 Configuring Services

- To run a particular service functionality when **Linux** starts enter the command:

```
/sbin/chkconfig --level 235 <name_of_service> on
```

- To display the Help details enter the command:

```
/sbin/chkconfig --help
```

- To display the list of services available enter the command:

```
/sbin/chkconfig --list
```

Note Some utilities, such as **sendmail**, are not enabled by default. The administrator is responsible for their configuration.

2.2 Modifying Passwords and Creating Users

Two users are created when Linux is installed:

root administrator (password root)

linux ordinary user (password linux)

These passwords must be changed as soon as possible:

- To change the passwords use one of the following commands
  - **passwd user_name** command for root users
  - **passwd** command for ordinary users.
- To create new users enter the **/usr/sbin/useradd** command

```
useradd –g “group” –d “home login”
```

2.3 Configuring Security

This section provides the administrator with basic rules concerning cluster security. Different security policies can be set up according to the cluster's configuration.

The Management Node is the most sensitive part of the cluster from a security point of view. This node submits jobs in batch mode and is the central point for management. This is the reason why security has to be enforced regarding access to this node. Very few people should be able to access this node, and this access should be made using **OpenSSH** to eliminate eavesdropping, connection hijacking, and other network-level attacks.

Compute Nodes and I/O Nodes should not have interactive logins. This means that no user except root should have access to these nodes. Management tools like **Nagios** will have access to both node types, while a batch manager like **PBS-Pro** will have access to Compute Nodes only.

If **CPU** and memory resources are shared among users, each individual user should not have access to other partitions.

2.3.1 Setting up SSH

Carry out the following steps to set up **SSH** for an admin user:

1. Create a public key:

```
ssh-keygen -tdsa -N ''
```

   This creates an **ssh** protocol 2 DSA certificate without passphrase in **~/.ssh/id_dsa.pub**.

2. Append this key to the list of authorized keys in **~/.ssh/authorized_keys2**.

3. Run **ssh** manually for each node responding yes at the prompt to add the node to the list of known hosts:

```
atlas0: ssh atlas1 hostname
```

```
The authenticity of host 'atlas1 (192.168.84.2)' can't be established.
RSA key fingerprint is
9c:d8:62:b9:14:0a:a0:18:ca:20:f6:0c:f6:10:68:2c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'atlas1,192.168.84.2' (RSA) to the list of
known hosts.
```

Note For the root user there is an authorized keys file for each node as **~root/.ssh/authorized_keys2** is local. The new key must be appended to each of these files.

Please refer to the chapter in this manual on **Kerberos** for more information on **SSH** and the use of keys.

2.4 Running Parallel Commands with pdsh

A distributed shell is a tool that allows the same command to be launched on several nodes. Such a function is essential in a cluster environment so that instructions can be carried out on several nodes instead of running the command manually on each node in turn. Different tools can be used to enable this possibility.

pdsh is a utility that runs commands in parallel on all the nodes or on a group of nodes of the cluster. It is a very flexible tool especially for large cluster usage.

pdsh is a multi-threaded client for remote shell commands. It can use different remote shell services, such as **rsh**, **ssh** and **kerberos**.

Three utilities are included in **pdsh**:

- **pdsh** is used to run commands in parallel.
- **pdcp** is used to copy files on a group of nodes in parallel.
- **dshbak** is used to format, sort and display the results of a command launched with **pdsh**.

The **pdsh** utility relies on the security and authentication mechanisms provided by **ssh** and / or **Kerberos** V4 layers on which it is configured. See the chapter in this manual on Kerberos.

2.4.1 Using pdsh

Syntax:

The following commands are the ones which are used most often:

pdsh –R <rcmd_module> -w <node_list> -l user -Options Command

pdsh –R <rcmd_module> -a -x <node_list> -Options Command

pdsh –R <rcmd_module> -g <group_attributes> -Options Command

The most important options are described below. For a complete description of the options, refer to the **pdsh** man page.

Standard Target Node List Options:

-w <node_list> Targets the specified list of nodes. Do not use the -w option with any other node selection option (-a, -g). The node list can be a comma-separated list (node 1, node2, etc.); no space is allowed in the list. If you specify only the '-' character, the target hosts will be read from stdin, one per line. The node list can also be an expression such as host[1-5,7]. For more information about node list expressions, see the HOSTLIST EXPRESSIONS in the pdsh man page.

-x <node_list> Excludes the specified nodes. The -x option can be used with other target node list options (-a, -g, -A). The node list can be a comma-separated list (node1, node2, etc.); no space is allowed in the list. The node list can also be an expression such as host[1-5,7]. For more information about the node list expressions, see the HOSTLIST EXPRESSIONS in the pdsh man page.

Standard pdsh Options:

-S Displays the largest value returned by the remote commands.

-h Displays commands usage and the list of the available rcmd modules and then quits.

-q Lists the option values and the target node list and exits without action.

-b Disables the Ctrl-C status feature so that a single Ctrl-C kills parallel jobs (Batch Mode).

-l <user> This option is used to run remote commands as another user, subject to authorization.

-t <cnx_timeout>Sets the connection timeout (in seconds). Default is 10 seconds.

-u <exec_time> Sets a limit on the amount of time (in seconds) a remote command is allowed to execute. Default is no limit.

-f <remote_cds_numb>
Sets the maximum number of simultaneous remote commands. Default is 32.

-R <rcmd_module>
Sets the rcmd module to use. The list of the available rcmd modules can be displayed using the -h, -V, or -L options. The default module is listed with -h or –V options.
Note: Instead of using this option, you can set the PDSH_RCMD_TYPE environment variable.

-L Lists information about all loaded **pdsh** modules and then quits.

-d Includes more complete thread status when SIGINT is received, and displays connection and command time statistics on stderr when done.

-V Displays pdsh version information, along with the list of currently loaded pdsh modules.

Group Attributes Options:

The following options use the cluster's group attributes as defined in the **/etc/genders** file.

-A Targets all nodes defined in the /etc/genders file.

-a Targets all nodes in the /etc/genders file except those with the pdsh_all_skip group attribute.

Note The **pdsh –a** command is equivalent to the **pdsh -A -X pdsh_all_skip** command. For example, you can set the **pdsh_all_skip** group attribute to the Service Nodes to exclude these specific nodes from cluster.

-g <gp-attr1[,gp_attr2,...]> Targets the nodes that have any of the specified group attributes. This option cannot be used with the -a and -w options.

-X <gp_attr1[,gp_attr2...]> Excludes the nodes that have any of the specified group attributes. This option may be combined with any other node selection options (-w, -g, -a, -A).

Examples:

- To execute the **pwd** command on all the nodes of the cluster using the **ssh** protocol, enter:

```
pdsh -R ssh -a pwd
```

- To list the system name of all nodes using **ssh** protocol, enter:

```
pdsh -R ssh -A uname -a
```

- To define **ssh** as default protocol, enter:

```
export PDSH_RCMD_TYPE=ssh;
```

- To display the date on all nodes, enter:

```
pdsh -A date
```

```
ns1: Mon Dec 13 13:44:48 CET 2004
ns0: Mon Dec 13 13:44:47 CET 2004
ns2: Mon Dec 13 13:44:47 CET 2004
ns3: Mon Dec 13 13:44:46 CET 2004
```

- To display the date on all nodes except on node ns0, enter:

```
pdsh -A -x ns0 date
```

```
ns1: Mon Dec 13 13:44:48 CET 2004
ns2: Mon Dec 13 13:44:47 CET 2004
ns3: Mon Dec 13 13:44:46 CET 2004
```

- To display the date of the I/O group nodes and to merge the output of the nodes whose result is identical, enter:

```
pdsh -g IO -x ns0 date | dshbak –c
```

```
---------------
ns[2-3]
---------------
 Mon Dec 13 14:10:41 CET 2004
---------------
ns[1]
---------------
 Mon Dec 13 14:10:42 CET 2004
```

2.4.2 Using pdcp

pdcp is a variant of the **rcp** command. Its syntax is not in the form remote_user@node:path. All source files are on the local node. The options which enable the nodes to be reached to be defined are similar to those of **pdsh**.

Syntax:

pdcp –Options … <source [src2…]> <destination>

Examples:

```
pdcp -R ssh -w ns[1-5] /etc/hosts /etc/hosts
```

```
pdcp -R ssh -g Analyse /tmp/foo
```

In the first example one copies **/etc/hosts** from the node where **pdcp** executes to all the nodes specified using the **– w** option by copying across the same path with the command.

For a complete description of the options please refer to the **pdcp** man page.

2.4.3 Using dshbak

One of the problems linked to the execution of commands in parallel on a big cluster, is the exploitation of the results, especially if the command generates a long output. The results of a command executed with **pdsh** are displayed asynchronously and each line is stamped with the node name, as in the following example:

```
pdsh -w ns[0-2] pwd
```

```
ns0 : /root
ns2 : /root
ns1 : /root
```

The **dshbak** utility formats the results of a **pdsh** command into a more user friendly form. Note that the results must be directed into a buffer file before being processed by **dshbak**.

Syntax:

dshbak [-c] <buffer_file>

dshbak can be used to create the following output:

- The node name, which was displayed on each line, is removed and replaced by a header containing this name.
- The generated list is sorted according to the node name if this name is suffixed by a number (ns0, ns1, ns2… ns500).
- If the **–c** option is present; **dshbak** will displays the identical results for several nodes once only. In this instance the header contains the node list.

Examples:

In the following example, the result of the **pdsh** command is not formatted:

```
pdsh -R ssh w ns[0-2] rpm -qa | grep qsnetmpipwd
```

```
ns1 : qsnetmpi-1.24-31
ns2 : qsnetmpi-1.24-31
ns0 : qsnetmpi-1.24-31
```

In the following example, the **pdsh** output is re-directed to **res_rpm_qsnetmpi** file, and then the **dshbak** command formats and displays the results:

```
pdsh -R ssh w ns[0-2] rpm -qa | grep qsnetmpipwd >
/var/res_pdsh/res_rpm_qsnetmpi

dshbak -c res_rpm_qsnetmpi
```

```
---------------
ns[0-2]
---------------
qsnetmpi-1.24-31
```

2.5 Day to Day Maintenance Operations

A set of maintenance tools is provided with a Bull extreme computing cluster. These tools are mainly Open Source software applications that have been optimized, in terms of CPU consumption and data exchange overhead, to increase their effectiveness on Bull extreme computing clusters which may include hundred of nodes.

Function	Tool	Purpose
Administration	ConMan ipmitool	Managing Consoles through Serial Connection
	nsclusterstop / nsclusterstart	Stopping/Starting the cluster
	nsctrl	Managing hardware (power on, power off, reset, status, ping checking temperature, changing bios, etc)
	Remote Hardware Management CLI	
	nsfirm	Obtaining the BMC or BIOS version, upgrading firmware, etc.
	syslog-ng	System log Management
	lptools (lputils, lpflash)	Upgrading Emulex HBA Firmware (Host Bus Adapter)
Backup / Restore	Bull System Backup Restore (BSBR)	Backing-up and restoring data
Monitoring	ibstatus, ibstat	Monitoring **InfiniBand** networks
	IBS tool	Providing information about and configuring **InfiniBand** switches
	lsiocfg	Getting information about storage devices
	pingcheck	Checking device power state
Debugging	ibtracert	Identifying **InfiniBand** network problem
	crash/proc/kdump	Runtime debugging and dump tool
	hpcsnap	Collecting cluster information
Testing	postbootchecker	Making verifications on nodes as they start

Table 2-1. Maintenance Tools

See - The *Maintenance Guide* for more information.
- The *Application Tuning Guide* for details on Bull HPC Toolkit, a set of cluster profiling tools.

Chapter 3. Cluster Database Management

This chapter describes the architecture of the Cluster Database, and the commands and tools which enable the administrator to display, and to change the Cluster Database.

The following topics are described:

- 3.1 *Architecture of ClusterDB*
- 3.2 *ClusterDB Administrator*
- 3.3 *Using Commands*
- 3.4 *Managing the ClusterDB*
- 3.5 *ClusterDB Modeling*

3.1 Architecture of ClusterDB

The Cluster database (**ClusterDB**) of the bullx cluster suite delivery includes the data that is required for the cluster management tools (**BSM – HPC Edition, KSIS, pdsh, syslog-ng, ConMan** etc.). Compared with sequential configuration files, the advantages of using a database are flexibility, and data availability for all the tools, ensuring better integration without the duplication of common data. Cluster database management uses the highly-scalable, **SQL** compliant, Open Source object-relational **PostgreSQL**. The figure below shows the architecture for the **ClusterDB** and its relationship to the cluster management tools.

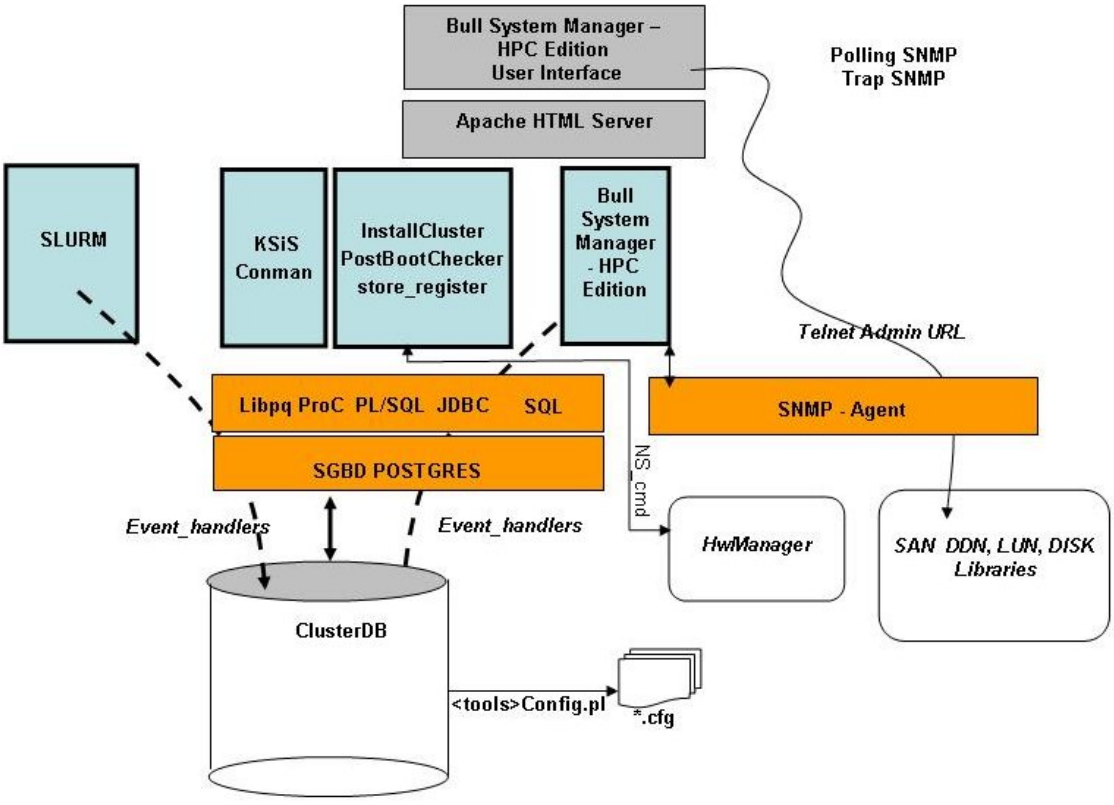


Figure 3-1. bullx cluster suite ClusterDB architecture

3.2 ClusterDB Administrator

The **ClusterDB** is installed on the Management Node. All operations on the **ClusterDB** must be performed from the Management Node.

The Database administrator is the **postgres** Linux user. This administrator is allowed to display and modify the **ClusterDB**, using the specific commands described in the next section. To manage the database (start, stop, save and restore), the administrator uses **PostgreSQL** tools (see 3.4 *Managing the ClusterDB*).

3.3 Using Commands

The administrators can consult or change the **ClusterDB** by using the following commands:

changeOwnerProperties	Changes the confidentiality parameters
dbmConfig	Controls the consistency of the **ClusterDB** with the system. All database updates are marked to be a "candidate" for synchronization.
dbmCluster	Operates on the whole cluster to get information, to check IP addresses and to check rack configuration.
dbmNode	Displays information, or change attributes at the node level.
dbmHwManager	Displays information, or change attributes at the **Hwmanager** level.
dbmGroup	Manages the groups of nodes.
dbmEthernet	Displays information, or change attributes for the Ethernet switches.
dbmIconnect	Displays information, or change attributes for the interconnect switches.
dbmTalim	Displays information, or change attributes for the remotely controlled power supply devices.
dbmSerial	Displays information, or change attributes for the Portservers.
dbmFiberChannel	Displays information about the Fiber Switches or changes the values of some attributes for a Fiber Switch or a subset of Fiber Switches.
dbmServices	Displays information about the Services or changes the values of some attributes for a Service.
dbmDiskArray	Displays information (for example **iproute**, **status**) and manages the disk array (status).

3.3.1 ChangeOwnerProperties

The cluster is handed over to the client with a name, a basename and IP address defined by Bull.

The IP address syntax used to identify equipment is of the form **A. V. U. H**.

V (the second byte) could be used for VLAN identification, **U** for Unit (Storage, Compute or Service) and **H** for Host (Host but also switch, disk subsystem or portserver).

The client may then want to change some of the attributes in keeping with their own security criteria.

These changes will in turn impact the **ClusterDB** Database, the network configuration for all the hosts, the configuration of storage bays and also the **Lustre** configuration (if installed).

Sometimes, the parameters will have been modified by the client as a result of:

- Running **ECT** (Embedded Configuration Tool) for Interconnect switches
- Running **bmcConfig** for BMC cards
- Running **swtConfig** for Ethernet switches
- The network configuration of the nodes done by **KSIS** at the time of the redeployment.
- Reconfiguring the **DDN** and **FDA** (Fibre Disk Array) subsystems to update them with the admin IP address and the gateway address.
- Manual operation of the **FDA**
- Running the **ddn_init** command on each **DDN** and for the reboot.
- Restarting the configuration of the HA Cluster Suite on I/O nodes, so that each node is aware of its peer node, using the correct names and IP addresses.
- The **Lustre** system is impacted if the node **basenames** are changed resulting in the obliteration of the file system followed by the creation of a new file system with new data.

If there is a change in the node **basenames** and of the admin IP address, the KSIS images are deleted from the database.

Consequently, when using this command, it is necessary to follow the process described below in order to reinitialize the system.

Syntax:

(This command is installed under **/usr/lib/clustmngt/clusterdb/install**).

changeOwnerProperties [--name <clustername>] [--basename <basename>]
[--adprivacy <bytes>]
[--icprivacy <interconnect privacy bytes (ic over ip)>]
[--bkprivacy <bytes>]
[--bkgw <ip gateway>] [--bkdom <backbone domain>]
[--bkoffset <backbone Unit offset>]
[--dbname <database name] [--verbose]

Options:

--dbname Specifies the name of the database to which the command applies. Default value: **clusterdb**.
Note: This option must be used only by qualified people for debugging purposes.

--name Specifies the name of the cluster. By default it is the basename.

--basename Specifies the basename of the node. (The node name is constituted of basename + netid). It is also the virtual node name.

--adprivacy Privacy bytes. According to the admin netmask, one, two or three bytes can be changed. For example, if the admin netmask is 255.255.0.0, then **adprivacy** option can specify two bytes in the form **A.V**.

--jcprivacy Privacy bytes. According to the interconnect netmask, one, two or three bytes can be changed. For example, if the interconnect netmask is 255.255.255.0, then **icprivay** option can specify three bytes in the form **A.V.U**.

--bkprivacy Privacy bytes. According to the backbone netmask, one, two or three bytes can be changed. For example, if the backbone netmask is 255.255.255.0, then **bkprivay** option can specify three bytes in the form **A.V.U**.

--bkgw Specifies the backbone gateway

--bkdom Specifies the backbone domain

--bkoffset Specifies the backbone translation offset. It permits to translate the D.E.U.H backbone ip to D.E.(U + bkoffset).H

Example:

To change the basename and byte A of the admin IP address enter:

```
changeOwnerProperties --basename node --adprivacy 15
```

Process:

1. Retrieve the current privacy bytes by running.

```
dbmEthernet show --nw admin
```

2. Change parameters using the command **changeOwnerProperties.** If you changed network parameters then you have to reconfigure the IP addresses of all equipment as follows.

3. Reconfigure admin interface of management node (**eth0** and **eth0:0** interfaces).

4. Update the **dhcpd** configuration and restart the service by running.

```
dbmConfig configure --service sysdhcpd
```

5. Restart **dbmConfig**.

6. Reconfigure Ethernet switches by running.

```
swtConfig change_owner_properties --oldadprivacy <bytes>
```

7. Reconfigure the IP addresses of the BMC cards.

```
/usr/lib/clustmngt/BMC/bmcConfig –-oldaprivacy <bytes>
```

8. Manually configure on the FDA (if present).

9. Run **ddn_init** on each DDN and reboot (if DDN storage is used).

10. HA Cluster Suite: run **storedepha** (if HA).

11. Syslog: The DDN logs are archived with the base name on the IP address, rename and the log files updated (if DDN is present)

12. For a **Lustre** configuration if the basename is changed:
    a. Run **lustre_util stop**
    b. Run **lustre_util remove**
    c. Truncate the LUSTRE_OST, LUSTRE_MDT tables and use **storemodelctl generateost** and **storemodelctl generatemdt** to repopulate the tables with the new information.
    d. Validate the recreated OSTs / MDTs: **lustre_investigate check**
    e. Verify the Lustre models and regenerate the configuration file: **lustre_config**
    f. Install new file systems: **lustre_util install**

3.3.2 dbmConfig

The **dbmConfig** command is used to maintain the consistency between the data in the **ClusterDB** and the different services and system files. The **dbmConfig** command shows the synchronization state or synchronizes different cluster services (**syshosts, sysdhcpd, conman, portserver, pdsh, nagios, snmptt, group, BSM**).

Syntax:

dbmConfig show [--service <name>] [--dbname <database name>] [--impact]

dbmConfig configure [--service <name> [--id <id> --timeout <timeout>] --restart --force --nodeps --impact] [--dbname <database name>]

dbmConfig help

Actions:

show Displays the synchronization state of all the services or of a list of specified services.

configure Runs the synchronization between the ClusterDB and all the services or a list of specified services. The configuration errors, if any, are listed on the console and in the **/var/log/synchro.log** file. It is necessary to check these messages.
Note: The command reports an OK status to indicate that it has completed. This does not mean that no configuration error occurred.

help Displays the syntax of the **dbmConfig** command.

Options:

--dbname Specifies the name of the database to which the command applies. Default value: clusterdb.
Note: This option must only be used by qualified people for debugging purposes.

--force Reconfigures the service and restarts it.

--id Reloads the configuration of the portserver identified by id. This option applies only to the portserver service (--service=portserver option).

--impact Displays the configuration files and associated services impacted by the next dbmConfig configure command.

--nodeps Forces the reconfiguration, despite the inter service dependencies.

--restart Restarts the service instead of reloading it.

--service Specifies the service from the following: syshosts, sysdhcpd, conman, portserver, pdsh, nagios, snmptt, group, BSM. For more information see Updated Configuration Files below.

--timeout Specifies the timeout (in seconds) for restarting the portserver. This option applies only to the portserver service (--service=portserver option).
Default value: 240.

Updated Configuration Files:

According to the specified service, the **dbmConfig configure --service** command updates a configuration file, as described below:

Service	Action
syshosts	Updates the **/etc/hosts** file with the data available in the administration base
sysdhcpd	Updates the **/etc/dhcpd.conf** file with the data available in the administration base.
conman	Updates the **/etc/conman.conf** file with the data available in the administration base.

portserver	Updates the portserver configuration file (/tftpboot/ps16*ConfigTS16 or /tftpboot/ps14*ConfigTS4), reloads the file on the appropriate portserver and reboots it.
pdsh	Updates the **/etc/genders** file with the data available in the administration base.
nagios	Updates several configuration files (**/etc/nagios/*.cfg**) with the data available in the administration base.
snmptt	Updates the **/etc/snmp/storage_hosts** file with the data available in the administration base.
group	Creates the predefined groups in the database. (No configuration file is updated.)
BSM	Updates the authentication file for the HW managers with the data available in the administration base.

If the administrator needs to modify these configuration files, for example, to add a machine that does not belong to the cluster, or to modify parameters, it is mandatory to use the template files created for this usage and to run the **dbmConfig** command again.

The templates files are identified by the **tpl** suffix. For example **/etc/hosts-tpl**, **/etc/dhcpd-tpl.conf**, **/etc/conman-tpl.conf**.

Examples:

- To configure the ConMan files, enter:

```
dbmConfig configure --service conman
```

- To list the synchronization state for Nagios, enter:

```
dbmConfig show --service nagios
```

3.3.3 dbmCluster

The **dbmCluster** command displays information about the whole cluster, or checks integrity and consistency of some elements of the ClusterDB.

Syntax:

dbmCluster show [-- dbname <database name>]

dbmCluster check ((--ipaddr | --rack) [--verbose]) | --unitCell [--dbname <database name>]

dbmCluster set --profile <key1>=<value1> ... --profile <keyN>=<valueN> [--dbname <database name>]

dbmCluster --h | --help

Actions:

show Displays the features of the cluster in terms of number of nodes and number of disks subsystems, as defined at the time of installation or update of the ClusterDB.

check Checks integrity and consistency of some data of the ClusterDB: single IP addresses (**--ipaddr** option) or consistency of rack equipments (**--rack** option) or consistency of Unit Cell equipment (**--unitCell** option).

set Changes the value of some profile fields in the cluster table.

help Displays the syntax of the **dbmCluster** command.

Options:

--dbname Specifies the name of the database to which the command applies. Default: **clusterdb**.
Note: this option must be used only by qualified people for debugging purposes.

--ipaddr Checks that the IP addresses are distinct within the cluster.

--rack Checks that the amount of equipment set for a rack in the database is not greater than the maximum. Also checks that there are not two sets of equipment on the same shelf.

--unitCell Checks that the object Unit and Cell number are the same as the Ethernet switch connected to.

--profile Used to set one key/value pair to be changed in table cluster.
key must be in [actif_ha, actif_crm, actif_vlan, resource_manager, batch_manager, security parallel_fs]

Examples:

- To check that each IP address is distinct, enter:

```
dbmCluster check --ipaddr
```

3.3.4 dbmNode

The **dbmNode** command displays information about the nodes (type, status, installed image etc.) or changes the values of some attributes for a node or a set of nodes (unit).

Syntax:

dbmNode show [--sysimage [--install_status={installed | not_installed | in_installed}]]

dbmNode show [--name <node name> --hwmanager | --cpu | --iproute | --serialroute]

dbmNode show [--unit <unit_num> --hwmanager | --cpu] [--dbname <database name>]

dbmNode set --name=<node name> --status={managed | not_managed} | --admin_macaddr <macaddr> | --backbone_macaddr <macaddr>

dbmNode set --unit <unit num> --status={managed | not_managed}

dbmNode set --nodelist=<node list> --status={managed | not_managed}

dbmNode set (--name=<node name> | --unit <unit num>) --cpu <total cpu chipset>

dbmNode set (--name=<node name> | --unit <unit num>) --hyperthreading={yes | no}

dbmNode set (--name=<node name> | --unit <unit num>) --cpu <total cpu chipset> --hyperthreading={yes | no} [--dbname <database name>]

dbmNode -h | --help

Actions:

show Displays type and status information for all the nodes or a set of nodes (--name option or --unit option). You can display the system images of nodes (using the --sysimage and --installed_status options), and the CPU or BMC / CMC features (using the --cpu and --hwmanager options).
The **Type** parameter specifies the node functions in the form `ACIMBNT`.
`A` means ADMIN
`C` means COMPUTE
`I` means I/O
`M` means META
`B` means INFINIBAND
`N` means NFS
`T` means TAPE
For example, the type for a compute node is displayed as "`-C-----`".

set Changes the value of some features for the specified node (--name option) or for all the nodes of the specified unit (--unit option) or for a set of nodes (--nodelist option).

Options:

--**help** Displays summary of options.

--**admin_macaddr** Specifies the MAC address of the eth0 interface connected to the administration network.

--**backbone_macaddr** Specifies the MAC address of the eth1 interface connected to the backbone network.

--**cpu** Displays the CPU feature (model and number), or changes the number of CPUs.

--**install_status** Displays only the nodes that have the specified install status (installed, in_installed, not_installed).

--**name** Specifies the node name to which the action applies.

--**iproute**	Displays the ethernet path (the localization and status of all Ethernet switches) between the node and the admin node
--**serialroute**	Displays the serial path over portserver (the localization and status of all portservers) between the node and the admin node
--**hwmanager**	Displays the name of the hwmanager that drives the node.
--**status**	Changes the status (managed / not_managed). The "not_managed" status means that the node has not to be managed by the administration tools.
--**sysimage**	Displays the nodes and the status of their system image.
--**unit**	Specifies the unit to which the action applies.
--**hyperthreading**	Changes the hyperthreading mode.
--**dbname**	Specifies the name of the database on which the command is applied. Default: clusterdb. **Note:** this option must be used only by qualified people for debugging purposes.

Examples:

- To set the status of the node16 node to “up”, enter:

```
dbmNode set --name node16 --status managed
```

- To change the MAC address of the node60 node, enter:

```
dbmNode set --name node60 --admin_macaddr 00:91:E9:15:4D
```

- Below are various examples using the **dbmNode show** command:

```
dbmNode show
```

```
     Nodes names                                      Type      Status
-------------------------------------------------------------------------
     node[0]                                           AC-M---   up
     node[1-5,9-10]                                    -C-----   up
     node[8]                                           -C-—B--   not_managed
     node[6,11]                                        -CI----   down
     node[7]                                           -CI----   up
     node[12-13]                                       --I-B--   down
```

```
dbmNode show --sysimage
```

```
     Nodes names                        Type     Sys Image          Status
-------------------------------------------------------------------------------
     node[4]                            -C-----   BAS5-16K           up
     node[3]                            -C-----   BAS5-FAME          up
     node[2,9]                          -C-----   ONEDISK            up
     node[8]                            -C—-B--   ONEDISK            up
     node[1,5,10]                       -C-----   NULL               up
     node[6,11]                         -CI----   NULL               down
     node[7]                            -CI----   NULL               up
```

```
      node[12-13]                        --I-B--    NULL                down
```

```
dbmNode show --sysimage --install_status installed
```

```
      Nodes names                      Type      Sys Image          Status
-------------------------------------------------------------------------
      node[4]                          -C-----     BAS5-16K           up
      node[3]                          -C-----     BAS5-FAME          up
      node[2,9]                        -C-----     ONEDISK            up
      node[8]                          -C-—B--     ONEDISK            up
```

```
dbmNode show --name ns0 --cpu
```

```
Name          Cpu model       Cpu total    Cpu available   Hyper threading
-------------------------------------------------------------------------
ns0        UNDEF           8            0               0
```

3.3.5 dbmHwManager

The **dbmHwManager** command displays information or change status at the level of the HW Manager.

Syntax:

dbmHwManager show [--name <hwmanager name> --node | --status | --iproute]

dbmHwManager show [--unit <unit num> --status] [--dbname <database name>]

dbmHwManager set --name <hwmanager name> --status ={managed | not_managed} | --password

dbmHwManager set --unit <unit_num> --status ={managed | not_managed} [--dbname <database name>]

dbmHwManager -h | --help

Actions:

show Displays model, type and status information for all the hwmanagers or a subset of hwmanager (--unit option).

set Changes the value of some features for the specified hwmanager (--name option) or for all the hwmanagers of the specified unit (--unit option).

Options:

--**help** Displays summary of options.

--**name** Specifies the hwmanager name to which the action applies.

--**iproute** Displays the Ethernet path (the localization and status of all Ethernet switches) between the hwmanager and the admin node

--**node** Displays the name of the nodes managed by the hwmanager.

--**status** Changes the status (managed/not_managed). The "not_managed" status means that the hwmanager has not to be managed by the administration tools.

--**password** Change the password for a given hwmanager.

--**unit** Specifies the unit to which the action applies.

--**dbname** Specifies the name of the database on which the command is applied. Default: clusterdb.
Note: This option must be used only by qualified people for debugging purposes.

Examples:

- To change the status of the PAP named `pap1` to "UP", enter:

```
dbmHwManager set –-name pap1 –-status managed
```

3.3.6 dbmGroup

The **dbmGroup** command lets the administrator of the ClusterDB show or modify (add, delete, create) the organization of the groups of nodes.

Note The groups are using commands like pdsh, KSiS, to perform actions on a set of nodes.

Syntax:

dbmGroup show [--dbname <database name>]

dbmGroup add --name <group name> --nodelist <node list> [--comment <description>] [--dbname <database name>]

dbmGroup del --name <group name> | --all [--dbname <database name>]

dbmGroup modify --name <group name> (--addnodelist <node list> | --delnodelist <node list>) [--dbname <database name>]

dbmGroup create [--dbname <database name>]

dbmGroup -h | --help

Actions:

show Displays the group of nodes.

add Adds a group to the existing ones.

del Deletes one group or all groups.

modify Adds or deletes a list of node in an existing group.

create Recreates the predefined groups (criterion groups), in the case they have been deleted.

Options:

--**help**	Displays summary of options.
--**name**	Specifies the group name.
--**nodelist**	List of the netid for the nodes of the group, in the form [x,y-w].
--**comment**	Description of the group.
--**all**	Deletes all nodes.
--**addnodelist**	Adds a node list in an existing group.
--**delnodelist**	Deletes a node list in an existing group.
--**dbname**	Specifies the name of the database on which the command is applied. Default: clusterdb. **Note:** this option must only be used by qualified people for debugging purposes.

Predefined Groups:

Once the cluster is configured, some predefined groups are automatically created, depending on the node types defined in the ClusterDB.

The **dbmGroup show** command displays the groups and a short explanation for each one.

Note A group can be mono-type, or multi-type for the nodes which combine several functions. Seven mono-type groups can be defined: **ADMIN**, **COMPUTE** (or COMP), **IO**, **META, IBA**, **NFS**, **TAPE**. See below examples of mono-type and multi_type groups.

Example of Predefined Groups:

In the following example four sorts of groups are defined:

- One Group of all the nodes **except** the nodes whose type is **ADMIN**. This group is named **ALL**.
- The group nodes per type. For instance:

ADMIN	Group of all the nodes whose type is ADMIN (mono-type).
ADMINCOMPMETA	Group of all the nodes whose type is ADMIN, compute or IO (multi-type).
COMPIBA	Group of all the nodes whose type is compute and Infiniband (multi-type).
COMPIO	Group of all the nodes whose type is compute or IO (multi-type).
COMPUTE	Group of all the nodes whose type is compute (mono-type).
IO	Group of all the nodes whose type is IO (mono-type).

IOIBA	Group of all the nodes whose type is IO and InfiniBand (multi-type).
META	Group of all the nodes whose type is METADATA (mono-type).

- The groups of COMPUTE nodes for each memory size. For instance:

COMP48GB	Group of all the nodes whose type is **compute** and with 48GBs of memory (mono-type).
COMP128GB	Group of all the nodes whose type is **compute** and with 128GB of memory (mono-type).

- The groups of nodes for each memory size. For instance:

NODES16GB	Group of all the nodes with 16GBs of memory.
NODES48GB	Group of all the nodes with 48GBs of memory.
NODES64GB	Group of all the nodes with 64GBs of memory.
NODES128GB	Group of all the nodes with 128GBs of memory.

Examples:

- To display all the groups defined in the ClusterDB, enter:

```
dbmGroup show
```

```
Group Name  Description                  Nodes Name
-------------------------------------------------------
ADMIN       Nodes by type:ADMIN          node0
ALL         All nodes except node admin  node[4-5,8-10]
COMP        Nodes by type:COMP           node[4,8]
COMP128GB   COMPUTE node with 128GB      node8
COMP48GB    COMPUTE node with 48GB       node4
IO          Nodes by type:IO             node10
META        Nodes by type:META           node[5,9]
NODES128GB  Nodes by memory size:128GB   node8
NODES48GB   Nodes by memory size:48GB    node[4,10]
NODES64GB   Nodes by memory size:64GB    node[0,5,9]
```

- To add a new group, named **GRAPH**, which includes the nodes 1 and 4, 5, 6 (netid) into the database, enter:

```
dbmGroup add --name GRAPH --nodelist [1,4-6] --comment 'Graphic Group'
```

- To delete the **GRAPH** group from the database, enter:

```
dbmGroup del --name GRAPH
```

- To re-create the predefined groups if they have been deleted, enter:

```
dbmGroup create
```

```
=>
Create ALL                              [ OK ]
Create NODES4GB                         [ OK ]
```

```
Create NODES16GB                    [ OK ]
Create ADMIN                        [ OK ]
Create INFNFS                       [ OK ]
Create INFTAPE                      [ OK ]
Create IOINF                        [ OK ]
Create METAINF                      [ OK ]
```

3.3.7 dbmEthernet

The **dbmEthernet** command displays or change attributes for the Ethernet switches.

Syntax:

dbmEthernet show [--nw ={admin | backbone}]

dbmEthernet show [--name <switch name> [--status | --macaddr | --iproute | --linkhost]]

dbmEthernet show [--unit <unit num> [--status]] [--dbname <database name>]

dbmEthernet set --name <switch name> (--status ={managed | not_managed}
| --macaddr <macaddr> | ([--password] [--enabled_password]))

dbmEthernet set --unit <unit_num> --status ={managed | not_managed}
[--dbname <database name>]

dbmEthernet -h | --help

Actions:

show Displays name, networtk, ip address, Mac address and status information for all the switches or a subset of switches (--unit option).

set Changes the value of some features for a specified switch (--name option) or for all the switchs of the specified unit (--unit option).

Options:

--**help** Displays summary of options.

--**name** Specifies the switch name to which the action applies.

--**nw** Displays information about the given network type.

--**iproute** Displays the Ethernet path (the localization and status of all Ethernet switches) between the switch and the admin node.

--**macaddr** Changes the **MACADDR** of the Ethernet Switch.

--**status** Changes the status (managed / not_managed). The "not_managed" status means that the switch has not to be managed by the administration tools.

--**password** Change the password for a given switch.

--**enabled_ password** Change the enable password for a given switch.

--**unit** Specifies the unit to which the action applies.

--**dbname** Specifies the name of the database on which the command is applied. Default: clusterdb.

Note This option must be used only by qualified people for debugging purposes.

Examples:

- To display the features of the administration network, enter:

```
dbmEthernet show --nw admin
```

- To change the mac address of the Ethernet switch named `eswu1c2` to the value `00:91:E9:15:4D`, enter:

```
dbmEthernet set --name eswu1c2 --admin_macaddr 00:91:E9:15:4D
```

3.3.8 dbmIconnect

The **dbmIconnect** command displays or change attributes for the interconnect switches.

Syntax:

dbmIconnect show [--nw ={QsNet | InfiniBand | GbEthernet}]

dbmIconnect show [--name <switch name> [--status | --iproute] |--linkhost]

dbmIconnect show [--unit <unit num> [--status]] [--dbname <database name>]

dbmIconnect set --name <switch name> (--status ={managed | not_managed} | ([--password] [--enabled_password]))

dbmIconnect set --unit <unit_num> --status ={managed | not_managed} [--dbname <database name>]

dbmIconnect -h | --help

Actions:

show Displays name, networtk, admin and standby ip addresses, status information and hwmanager (if any) for all the switches or a subset of switches (--unit option).

set Changes the value of some features for a specified switch (--name option) or for all the switches of the specified unit (--unit option).

Options:

--**help** Displays summary of options.

--**name** Specifies the switch name to which the action applies.

--**nw** Displays information about the given network type.

--**iproute** Displays the ethernet path (the localization and status of all ethernet switches) between the InterConnect switch and the admin node.

--**linkhost** Displays hosts plugged on a given interconnect switch.

--**status** Changes the status (managed / not_managed). The "not_managed" status means that the switch has not to be managed by the administration tools.

--**password** Change the password for a given switch.

--**enabled_password** Change the enable password for a given switch.

--**unit** Specifies the unit to which the action applies.

--**dbname** Specifies the name of the database on which the command is applied. Default: clusterdb.
Note: This option must be used only by qualified people for debugging purposes.

Examples:

- To display the features of the QsNet interconnect, enter:

```
dbmIconnect show --nw QsNet
```

- To change the status of the interconnect switch named `QR0N01` to the value `not_managed`, enter:

```
dbmIconnect set --name QR0N01 --status not_managed
```

3.3.9 dbmTalim

The **dbmTalim** command displays or change attributes for remotely controlled power supply devices.

Note **Talim** refers to remotely controlled power supply devices which are used to start and stop equipment.

Syntax:

dbmTalim show [--name <talim name> [--status | --macaddr | --iproute]]

dbmTalim show [--unit <unit num> [--status]] [--dbname <database name>]

dbmTalim set --name <talim name> --status ={managed | not_managed} | --macaddr <macaddr>

dbmTalim set --unit <unit_num> --status ={managed | not_managed} [--dbname <database name>]

dbmTalim -h | --help

Actions:

show Displays name, networtk, ip address, Mac address and status information for all the talim or a subset of talim (--unit option).

set Changes the value of some features for a specified talim (--name option) or for all the talim of the specified unit (--unit option).

Options:

--**help** Displays summary of options.

--**name** Specifies the talim name to which the action applies.

--**iproute** Displays the ethernet path (the localization and status of all ethernet switches) between the talim and the admin node

--**macaddr** Displays the Macaddr or changes the Macaddr of the Talim.

--**status** Displays the status or changes the status (managed / not_managed). The "not_managed" status means that the talim has not to be managed by the administration tools.

--**unit** Specifies the unit to which the action applies.

--**dbname** Specifies the name of the database on which the command is applied. Default: clusterdb.
Note: This option must be used only by qualified people for debugging purposes.

Examples:

- To display the features of the talim named talim2, enter:

```
dbmTalim show --name talim2
```

- To change the mac address of the talim named `talim2` to the value `00:91:E9:15:4D`, enter:

```
dbmTalim set --name talim2 --macaddr 00:91:E9:15:4D
```

3.3.10 dbmSerial

Note The **dbmSerial** depends on the cluster's configuration and only applies to clusters which include a portserver.

The **dbmSerial** command displays or change attributes for the Portservers.

Syntax:

dbmSerial show	[--nw ={node|pap|storage|mixed}]
dbmSerial show	[--name <portserver name> [--status | --macaddr | --iproute | --linkhost]]
dbmSerial show	[--unit <unit num> [--status]] [--dbname <database name>]
dbmSerial set	--name <portserver name> --status ={managed | not_managed} | --macaddr <macaddr> | --password
dbmSerial set	--unit <unit_num> --status ={managed | not_managed} [--dbname <database name>]
dbmSerial	-h | --help

Actions:

show	Displays name, networtk, ip address, Mac address and status information for all the Portserver or a subset of portserver (--unit option).
set	Changes the value of some features for a specified Portserver (--name option) or for all the Portserver of the specified unit (--unit option).

Options:

--**help**	Displays summary of options.
--**nw**	Displays information about the given network type.
--**name**	Specifies the Portserver name to which the action applies.
--**iproute**	Displays the Ethernet path (the localization and status of all ethernet switches) between the Portserver and the admin node.
--**status**	Displays the status or changes the status (managed / not_managed). The "not_managed" status means that the Portserver has not to be managed by the administration tools.
--**macaddr**	Display/changes the Mac address of the Portserver.
--**linkhost**	Displays hosts plugged on a given portserver.
--**password**	Change the password for a given switch.
--**unit**	Specifies the unit to which the action applies.
--**dbname**	Specifies the name of the database on which the command is applied. Default: clusterdb. **Note:** This option must be used only by qualified people for debugging purposes.

Examples:

- To display the features of all portservers, enter:

```
dbmSerial show
```

- To display the list of the hosts plugged on the portserver named ps16u1c0, enter:

```
dbmSerial show --name ps16u1c0 --linkhost
```

- To change the status of the portserver named ps16u1C0 , enter:

```
dbmSerial set --name ps16u1C0 --status managed
```

- To change the status of all portservers affiliated with unit 0, enter:

```
dbmSerial set --unit 0 --status not_managed
```

3.3.11 dbmFiberChannel

Displays the Database information about the Fiber Switches or changes the values of some attributes for a Fiber Switch or a subset of Fiber.

Syntax:

dbmFiberChannel show	[--nw]
dbmFiberChannel show	[--name <switch name> [--status | --iproute]]
dbmFiberChannel show	[--unit <unit num> [--status]] [--dbname <database name>]
dbmFiberChannel set	--name <switch name> --status ={managed | not_managed}
dbmFiberChannel set	--unit <unit_num> --status ={managed | not_managed} [--dbname <database name>]
dbmFiberChannel	-h | --help

Actions:

show	Displays name, networtk, admin ip address, and status information for all the switches or a subset of switches (--unit option).
set	Changes the value of some features for a specified switch (--name option) or for all the switches of the specified unit (--unit option).

Options:

--help	Displays summary of options.
--name	Specifies the switch name to which the action applies.
--nw	Displays information about all network type.

--iproute Displays the ethernet path (the localization and status of all ethernet switches) between the Fiber switch and the admin node.

--status Changes the status (managed / not_managed). The "not_managed" status means that the switch has not to be managed by the administration tools.

--unit Specifies the unit to which the action applies.

--dbname Specifies the name of the database on which the command is applied. Default: clusterdb.
Note: This option must be used only by qualified people for debugging purposes.

Examples:

- To change the FC switch named `fcswu0c1` to up, enter:

```
dbmFiberChannel set --name fcswu0c1 --status managed
```

- To show the hierarchy iproute of the FC switch through Ethernet switches, enter:

```
dbmFiberChannel show --name fcswu0c1 --iproute
```

- To show information about FC switch, enter:

```
dbmFiberChannel show
```

3.3.12 dbmServices

Displays the Database information about the Services or changes the values of some attributes for a Service.

Syntax:

dbmServices show --objectlist

dbmServices show --object <object name> [--name <service name>]
[--dbname <database name>]

dbmServices set --object <object name> --name <service name> (--enable | --disable)
[--dbname <database name>]

dbmServices -h | --help

Actions:

show Displays the list of all the objects contained in Services table (--**objectlist** option).

Or displays name, object type and if service is enabled or disabled (--**object** --**name** options).

set Changes the value of the **actif** field (enable or disable) for a specified service (**--object --name** options).

Options:

--**help** Displays summary of options.

--**objectlist** Displays the list of all the objects contained in Services table.

--**object** Specifies the object type of service to which the action applies.

--**name** Specifies the service name to which the action applies.

--**enable** Specifies that the service must be activated.

--**disable** Specifies that the service must be de-activated.

--**dbname** Specifies the name of the database on which the command is applied. Default: clusterdb.
Note: This option must be used only by qualified people for debugging purposes.

Examples:

- To print details on the service named "Ethernet interfaces" on object node, enter:

```
dbmServices show –-object node –-name "Ethernet interfaces"
```

- To change the service named "Ethernet interfaces" on object node to up, enter:

```
dbmServices set –-object node –-name "Ethernet interfaces" --enable
```

3.3.13 dbmDiskArray

dbmDiskArray displays information (for example **iproute**, **status**) and manages the disk array (status)

Syntax:

dbmDiskArray show [--name <diskarray name> --iproute | --serialroute]
[--dbname <database name>]

dbmDiskArray set --name < diskarray name> --status={managed | not_managed}
[--dbname <database name>]

dbmDiskArray -h | --help

Actions:

show Displays the type and status information for all the disk arrays or for a specified one (--name option).

set Changes the value of some of the features for a specified disk array (--name option).

Options:

--help Displays a summary of options.

--name Specifies the disk array name to which the action applies.

--iproute Displays the Ethernet path (including the location and status of all Ethernet switches) between the disk array and the Management Node.

--serialroute Displays the serial path which includes a portserver (the location and status of all portservers) between the disk array and the Management Node.
Note: This option depends on the cluster's configuration and only applies to clusters which include a portserver.

--status Changes the status (**managed**/ **not_managed**). The **not_managed** status means that the disk array will not be managed by the administration tools.

--dbname Specifies the name of the database to which the command is applied. Default = clusterdb.
Note This option must be used only by qualified people for debugging purposes.

Examples:

- To print details of the disk array named **da0** using Ethernet switches, enter:

```
dbmDiskArray show --name da0 –iproute
```

- To change the status of the disk array named **da0** to up, enter:

```
dbmDiskArray set --name da0 –status managed
```

3.4 Managing the ClusterDB

The administrator of the **ClusterDB** must guarantee and maintain the consistency of the data. To view and administrate the database, the ClusterDB administrator can use the following PostgreSQL tools:

- The **PostgreSQL commands**.
  The **psql** command enables the PostgreSQL editor to run. You can run it as follows:

```
psql -U clusterdb clusterdb
```

- The **phpPgAdmin Web interface**.
  You can start it with an URL similar to the following one (`admin0` is the name of the Management Node):

```
http://admin0/phpPgAdmin/
```

These tools, which let the administrator update the ClusterDB, must be used carefully since incorrect usage could break the consistency of the ClusterDB.

For more information, refer to the **PostgreSQL** documentation delivered with the product.

3.4.1 Saving and Restoring the Database

The database administrator is responsible for saving and restoring the ClusterDB.

The administrator will use the **pg_dump** and **pg_restore** PostgreSQL commands to save and restore the database.

3.4.1.1 Saving the Database (pg_dump)

The **pg_dump** command has a lot of options. To display all the options, enter:

```
pg_dump --help
```

Note The **pg_dump** command can run while the system is running.

Saving the Metadata and the Data:

It is recommended that the following command is used:

```
pg_dump –Fc –C -f /var/lib/pgsql/backups/clusterdball.dmp clusterdb
```

Saving the Data only:

It is recommended that the following command is used:

```
pg_dump –Fc –a -f /var/lib/pgsql/backups/clusterdbdata.dmp clusterdb
```

Saving Data each Day

When the **clusterdb** rpm is installed, a **cron** is initialized to save the ClusterDB daily, at midnight. The data is saved in the **clusterdball[0-6].dmp** and **clusterdata[0-6].dmp** (**0-6** is the number of the day) in the **/var/lib/pgsql/backups** directory. This **cron** runs the **make_backup.sh** script, located in the directory **/usr/lib/clustmngt/clusterdb/install/**.

3.4.1.2 Restoring the Database (pg_restore)

The **pg_restore** command has a lot of options. To display all the options, enter:

```
pg_restore --help
```

Restoring the whole ClusterDB:

Requirement: ClusterDB does not exist anymore.

To list the existing databases, use the **oid2name** command:

```
oid2name
```

If you need to remove an inconsistent **ClusterDB**, enter:

```
dropdb clusterdb
```

When you are sure that the **ClusterDB** does not exist anymore, enter the following command to restore the whole database:

```
pg_restore –Fc --disable-triggers –C -d template1
/var/lib/pgsql/backups/clusterdball.dmp
```

Restoring the ClusterDB Data:

Requirement: ClusterDB must exist and be empty.

To create an empty ClusterDB, run these commands:

```
/usr/lib/clustmngt/clusterdb/install/create_clusterdb.sh -nouser
psql -U clusterdb clusterdb
clusterdb=> truncate config_candidate;
clusterdb=> truncate config_status;
clusterdb=> \q
```

To restore the data, enter:

```
pg_restore –Fc --disable-triggers -d clusterdb
/var/lib/pgsql/backups/clusterdbdata.dmp
```

3.4.2 Starting and Stopping PostgreSQL

Starting and stopping **postgreSQL** is performed using the **service** Linux command. **postgreSQL** is configured to be launched at levels 3, 4 and 5 for each reboot.

Note Both **root** user and **postgres** user can start and stop PostgreSQL. However it is recommended to use always the **postgres** login.

To start **postgreSQL**, run the following script:

```
/sbin/service postgresql start
```

To stop **postgreSQL**, run the following script:

```
/sbin/service postgresql stop
```

3.4.3 Viewing the PostgreSQL Alert Log

The **postgreSQL** log file is **/var/log/postgres/pgsql**. This is read to view any errors, which may exist.

Note This file can increase in size very quickly. It is up to the database administrator to rotate this file when **postgreSQL** is stopped.

3.5 ClusterDB Modeling

Important

The ClusterDB diagrams and tables which follow are common to both BAS4 and bullx cluster suite systems. Certain tables will only be exploited by the functionality of BAS4, for bullx cluster suite these tables will be empty.

3.5.1 Physical View of the Cluster Networks

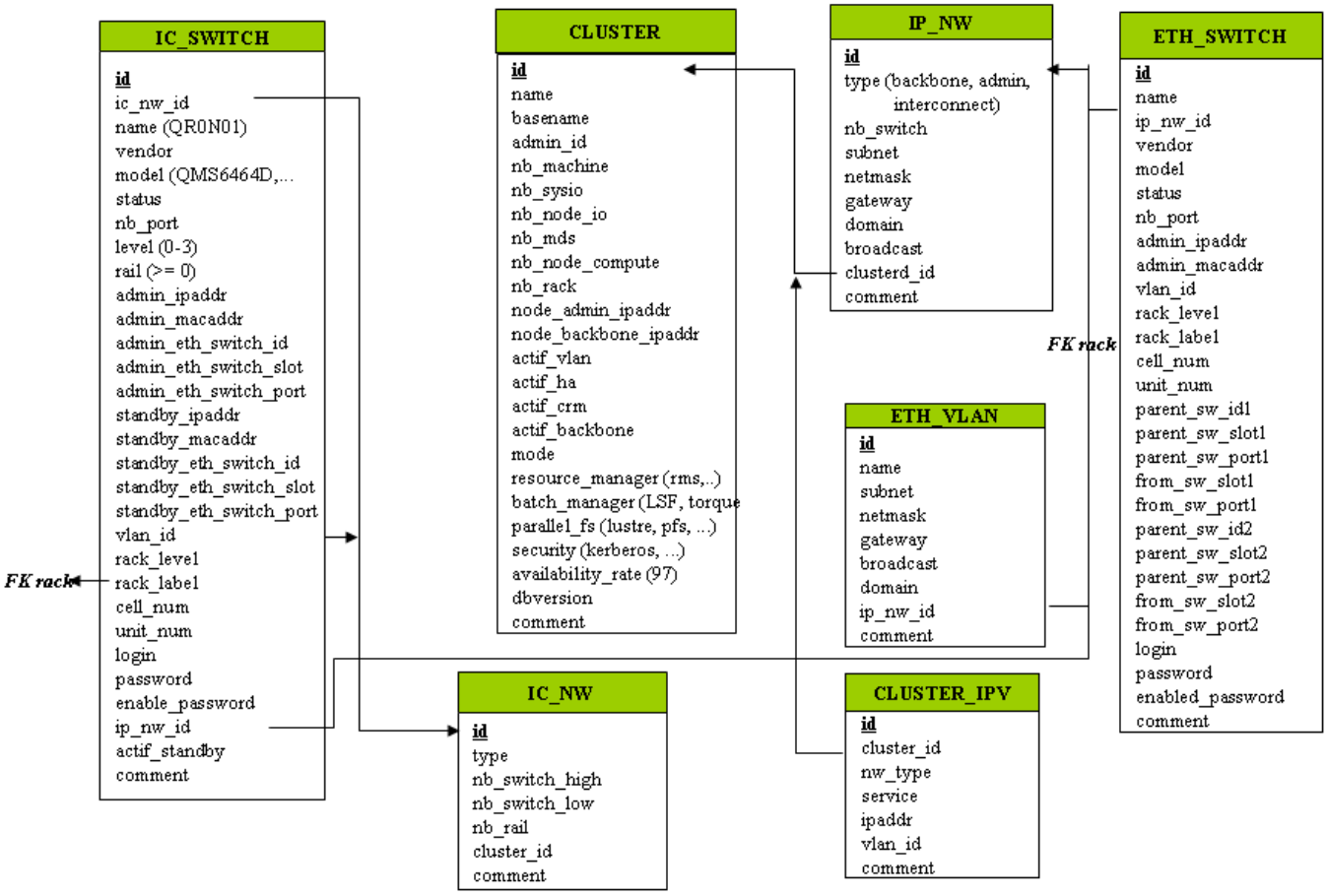


Figure 3-2. Cluster Network – diagram 1

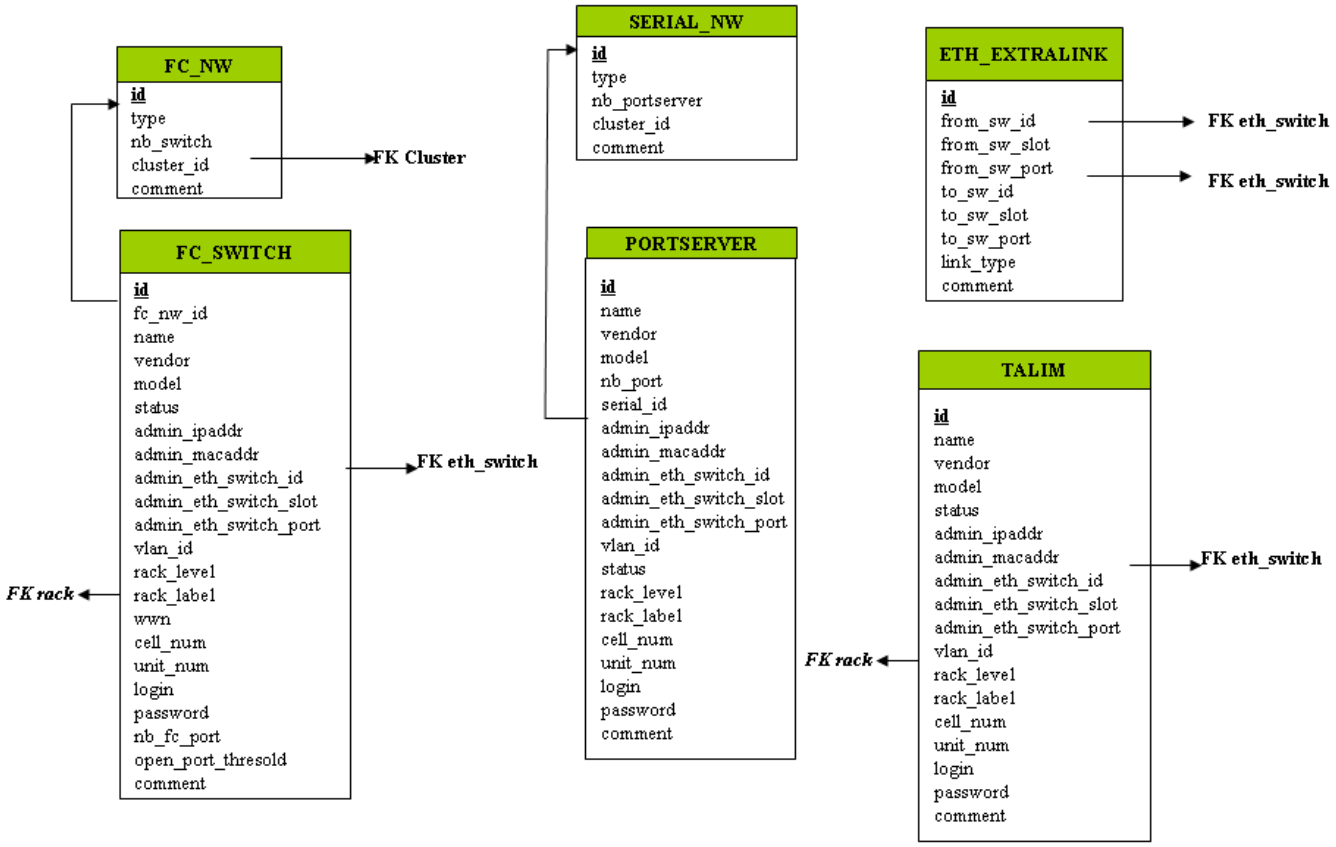


Figure 3-3. Cluster Network – diagram 2

3.5.1.1 CLUSTER Table

Column name	Description	Example	Fill in method
id	PK	540	preload - sequence
name	Name of the cluster	molecular	Preload & loadClusterdb
basename	Node basename	node	Preload & loadClusterdb
admin_id	FK table User		preload
nb_machine	Number of Nodes	601	preload – reconfigClusterdb
nb_sysio	Number of disk sub systems	56	preload – reconfigClusterdb
nb_node_io	Number of IO Nodes	54	preload – reconfigClusterdb
nb_mds	Number of MDS	2	preload – reconfigClusterdb
nb_node_compute	Number of Compute Nodes	544	preload – reconfigClusterdb
nb_rack	Number of rack	270	preload – reconfigClusterdb
node_admin_ipaddr	Virtual IP address of the Management node for the backbone network	10.1.0.65	preload
node_backbone_ipaddr	Virtual IP address of the Management node		preload
actif_vlan	Boolean on the VLAN configuration	true	preload
actif_ha	Boolean High Availability	true	Cluster Suite
actif_crm	CRM Boolean surveillance	true	preload
actif_backbone	Boolean, Use of a backbone	true	DV-true
mode	Mode 100%, 92% or 8%	100	preload – reconfigClusterdb
resource_manager	RMS or SLURM	rms	preload

Column name	Description	Example	Fill in method
batch_manager	LSF or TORQUE	torque	preload
parallel_fs	Lustre	lustre	prelaod
security	Kerberos	NULL	preload
availability_rate	Availability rate	NULL	Not used
dbversion	Development model version for the database	20.5.0	DV at creation
comment	Free field		NULL

Table 3-1. CLUSTER Table

3.5.1.2 IP_NW Table

Column name	Description	Example	Fill in method
id	PK	4	preload – Sequence
type	backbone, admin	backbone	preload
nb_switch	Number of switches	10	preload
subnet	Sub-network	10.0.0.0	preload& loadClusterdb
netmask	Sub-network mask	255.255.0.0	preload& loadClusterdb
gateway	IP address of the gateway	10.0.255.254	preload
domain	Name of the domain	frec.bull.fr	preload
broadcast	IP address of the broadcast	NULL	NULL
cluster_id	FK on the CLUSTER		preload
comment	Free field		NULL

Table 3-2. IP_NW table

3.5.1.3 ETH_SWITCH Table

Column name	Description	Example	Fill in method
id	PK		preload-Sequence
name	Name of the switch		preload
ip_nw_id	FK on IP_NW		preload
vendor	Vendor	CISCO	preload
model	Modele of the SW	CISCO6509	preload
status	Nagios host_status	up	DV = up - Nagios
nb_port	Total number of port		preload
admin_ipaddr	Admin IP address of the Ethernet switch		preload
admin_macaddr	Mac Address of the Switch		swtAdmin
vlan_id	FK on ETH_VLAN		preload
rack_level	Superposition level in the rack		preload
rack_label	Name of the rack		preload
cell_num	Name of the cell		preload
unit_num	Number of the Unit		preload
parent_sw_id1	Ethernet switch 1st parent		preload
parent_sw_slot1	Arrival slot number of the 1st parent switch	0	preload
parent_sw_port1	Connection port for the 1st switch	1	preload
from_sw_slot1	Starting slot number of the 1st switch	0	preload
from_sw_port1	Starting port number of the 1st switch	1	preload

Column name	Description	Example	Fill in method
parent_sw_id2	Ethernet switch 2nd parent		preload
from_sw_slot2	Starting slot number of the 2nd switch		preload
parent_sw_port2	Starting port number for the 2nd switch	2	preload
from_sw_slot2	Starting slot number of the 2nd switch		preload
from_sw_port2	Starting port number of the 2nd switch		preload
login	Administration login		cmdExpl
password	Administration password		cmdExpl
enabled_password	Manufacturer's enabled password		ECT
comment	Free field		

Table 3-3. ETH_SWITCH Table

3.5.1.4 IC_NW Table

Column name	Description	Example	Fill in method
id	PK		preload - Sequence
type	QSNet, Infiniband, GbEther	QSNet	preload
nb_switch_high	Number of high switches	12	preload - reconfigClusterdb
nb_switch_low	Number of low switches	33	preload - reconfigClusterdb
nb_rail	Number of rails	3	preload
cluster_id	FK on the CLUSTER		preload
comment	Free field		

Table 3-4. IC_NW Table

3.5.1.5 IC_SWITCH Table

Column name	Description	Example	Fill in method
id	PK		preload - Sequence
ic_nw_id	FK on the IC_NW		preload
name	Name of the Switch Interconnect	QR0N01	preload
vendor	Name of the Vendor	QUADRICS	preload
model	Model of the Switch	QMS6464D	preload
status	Nagios host_status	up	DV = up – Nagios
nb_port	Port number	64	preload
level	Level of the switch	1 – 2	preload
rail	Number of the rail	2	preload
admin_ipaddr	Administration IP address		preload
admin_macaddr	Mac Address of the switch	unused	NULL
admin_eth_switch_id	FK on ETH_SWITCH		preload
admin_eth_switch_slot	Arrival slot number on ETH_SW		preload
admin_eth_switch_port	Connection port on the ETH_SW	5	preload
standby_ipaddr	IP address of the standby controller		preload
standby_macaddr	Mac Address of the controller	unused	NULL
standby_eth_switch_id	FK on the ETH_SWITCH		preload
stanby_eth_switch_slot	Arrival slot number on ETH_SW		preload
standby_eth_switch_port	Connection port on the ETH_SW	6	preload
vlan_id	FK on the ETH_SWITCH		preload

Column name	Description	Example	Fill in method
rack_level	Level of superposition in the rack	G	preload
rack_label	Name of the rack	C0-A16	preload
cell_num	Number of the cell	1	preload
unit_num	Number of the Unit	0	preload
Login	Administration login	unused	preload or DV
password	Administration Password	unused	preload or DV
enable_password	Password enable		preload or DV
ip_nw_id	Foreign key on the IP_NW		preload
actif_standby	Configuration of a standby IP address		DV =false
comment	Free field		

Table 3-5. IC_SWITCH Table

3.5.1.6 SERIAL_NW Table

Column name	Description	Example	Fill in method
id	PK	1	preload – sequence
type	PAP, node, storage, mixed networks	node	preload
nb_portserver	Number of PortServer	39	preload
cluster_id	FK on the CLUSTER		preload
comment	Free field		

Table 3-6. SERIAL_NW Table

3.5.1.7 PORTSERVER Table

Note This table will not be filled for bullx cluster suite systems.

Column name	Description	Example	Fill in method
id	Primary key		preload - sequence
name	Name of the portserver	ps16u1c0	preload
vendor	Name of vendor	DIGI	preload
model	Model of the PS	TS16	preload
nb_port	Total number of TTY/PS ports	16	preload
serial_id	FK on SERIAL_NW		preload
admin_ipaddr	Administration IP address		preload
admin_macaddr	Mac address of the PS	NULL	
admin_eth_switch_id	FK on ETH_SWITCH		preload
admin_eth_switch_slot	Arrival slot number on ETH_SW		preload
admin_eth_switch_port	Connection port on the ETH_SW	10	preload
vlan_id	FK on the ETH_VLAN	40	preload
status	Nagios host_status	down	DV = up – Nagios
rack_level	Height of U in the rack		preload
rack_label	Name of the rack		preload
cell_num	Number of the cell		preload
unit_num	Number of the Unit		preload

login	Administration login		preload
password	Administration password		preload
comment	Free field		

Table 3-7. PORTSERVER Table

3.5.1.8 ETH_VLAN Table

Column name	Description	Example	Fill in method
id	PK	1	preload - sequence
name	Name of the VLAN	pad	preload
subnet	Sub-network IP address	10.4.0.0	preload
netmask	Netmask of the sub-network	255.255.0.0	preload
gateway	IP address of the gateway	10.4.255.254	preload
broadcast	IP address of the broadcast	10.4.255.255	preload
domain	Name of the domain	unused	preload – NULL
ip_nw_id	FW on the IP_NW		preload
comment	Free field		

Table 3-8. ETH_VLAN table

3.5.1.9 FC_NW Table

Note This table only applies to systems which include a Storage Area Network (**SAN**).

Column name	Description	Example	Fill in method
id	PK	1	preload - sequence
type	Role of the network	SAN-META	preload
nb_switch	Number of switches	39	preload
cluster_id	FK on the CLUSTER		preload
comment	Free field		

Table 3-9. FC_NW table

3.5.1.10 CLUSTER_IPV Table

Column name	Description	Example	Fill in method
id	PK	1	preload - sequence
cluster_id	FK on the CLUSTER		preload
nw_type	Network type	nfs	preload
service	nfs	nfs	preload
ipaddr	Virtual IP address of the service	10.11.0.99	preload
vlan_id	FK on ETH_VLAN		preload
comment	Free field		

Table 3-10. CLUSTER_IPV Table

3.5.1.11 FC_SWITCH Table

Note This table only applies to systems which include a Storage Area Network (**SAN**).

Column name	Description	Example	Fill in method
id	PK		preload-Sequence
name	Name of the switch		preload
fc_nw_id	FK on the FC_NW		preload
vendor	Name of the vendor	BROCADE	preload
model	SW model	Silkworm 200E	preload
status	Nagios host_status	up	DV = up - Nagios
admin_ipaddr	IP admin address on the fibre switch channel		preload
admin_macaddr	Mac Address of the Switch	NULL	
admin_eth_switch_id	FK on the ETH_SWITCH		preload
admin_eth_switch_slot	Arrival slot number on ETH SW		preload
admin_eth_switch_port	Connection on the ETH SW	3	preload
vlan_id	FK on the ETH_VLAN		preload
rack_level	Superposition level in the rack		preload
rack_label	Name of the rack		preload
cell_num	Number of the cell		preload
unit_num	Number of the unit		preload
login	Administration login		preload
password	Administration Password		prelaod
nb_fc_port	Number of fibre channel ports		preload
open_port_thresold			preload
comment	Free field		

Table 3-11. FC_SWITCH table

3.5.1.12 TALIM Table

Column name	Description	Example	Fill-in method
id	PK		preload-Sequence
name	Name of the power switch		preload
vendor	Vendor name		preload
model	Model of the power switch		preload
status	Nagios host_status	up	DV = up - Nagios
admin_ipaddr	Admin IP address of the power switch		preload
admin_macaddr	Mac Address of the power switch	NULL	
admin_eth_switch_id	FK on the ETH_SWITCH		preload
admin_eth_switch_slot	Arrival slot number on ETH SW		preload
admin_eth_switch_port	Connection port on the ETH SW	3	preload
vlan_id	FK on the ETH_VLAN		preload
rack_level	Superposition level in the rack		preload
rack_label	Name of the rack		preload
cell_num	Cell number		preload

Column name	Description	Example	Fill-in method
unit_num	Unit number		preload
login	Administration login		preload
password	Administration password		prelaod
comment	Free field		

Table 3-12. TALIM table

3.5.1.13 ETH_EXTRALINK Table

This table is not active in this version.

3.5.2 Physical View of the Storage

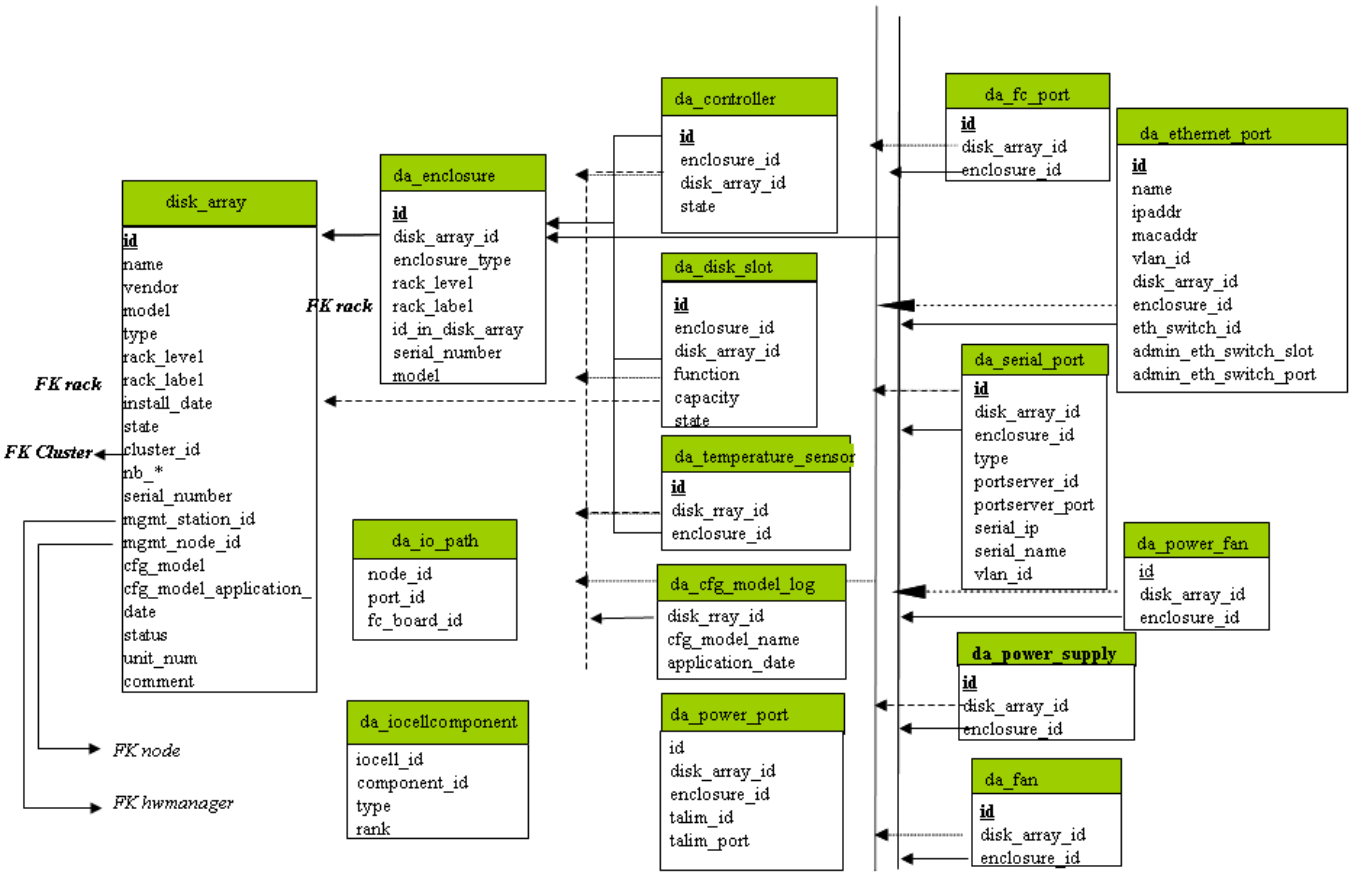


Figure 3-4. Storage physical view

3.5.2.1 disk_array Table

Field name	Field information	Fill in method
id	Unique identifier for the array in the database	preload - sequence
name	Name of the array (used for host in nagios)	preload
vendor	Vendor name: DDN, NEC, etc.	preload
model	Model name : S2A8500, FDA2300 …	preload
rack_level	Location in the rack	preload
rack_label	Label of the rack containing the disk array controller	preload
install_date	Date of bay installation	preload – current time
state	UNKNOWN, OK, WARNING, FAULTY, OFF_LINE, OUT_OF_CLUSTER	Preload: OUT_OF_CLUSTER Dynamic - BSM
cluster_id	Id of the cluster parent	preload
nb_enclosure	Number of disk enclosure	Dynamic (DV=0) - BSM
nb_controller	Number of controller	Dynamic (DV=0) - BSM
nb_controller_ok	Number of controller in OK state	Dynamic (DV=0) - BSM

Field name	Field information	Fill in method
nb_controller_faulty	Number of controller in FAULTY state	Dynamic (DV=0) - BSM
nb_fc_port	Number of FC ports	Dynamic (DV=0) - BSM
nb_fc_port_connected	Number of FC ports in CONNECTED state	Dynamic (DV=0) - BSM
nb_fc_port_not_connected	Number of FC ports in NOT_CONNECTED state	Dynamic (DV=0) - BSM
nb_fc_port_disconnected	Number of FC ports in DISCONNECTED state	Dynamic (DV=0) - BSM
nb_fc_port_faulty	Number of FC ports in FAULTY state	Dynamic (DV=0) - BSM
nb_serial_port	Number of serial ports	Dynamic (DV=0) - BSM
nb_serial_port_connected	Number of serial ports in CONNECTED state	Dynamic (DV=0) - BSM
nb_serial_port_not_connected	Number of serial ports in NOT_CONNECTED state	Dynamic (DV=0) - BSM
nb_serial_port_disconnected	Number of serial ports in DISCONNECTED state	Dynamic (DV=0) - BSM
nb_serial_port_faulty	Number of serial ports in FAULTY state	Dynamic (DV=0) - BSM
nb_eth_port	Number of Ethernet ports	Dynamic (DV=0) - BSM
nb_ethernet_port_connected	Number of ethernet ports in CONNECTED state	Dynamic (DV=0) - BSM
nb_ethernet_port_not_connected	Number of ethernet ports in NOT_CONNECTED state	Dynamic (DV=0) - BSM
nb_ethernet_port_disconnected	Number of ethernet ports in DISCONNECTED state	Dynamic (DV=0) - BSM
nb_ethernet_port_faulty	Number of ethernet ports in FAULTY state	Dynamic (DV=0) - BSM
nb_disk	Number of disks	Dynamic (DV=0) - BSM
nb_disk_slot_ok	Number of disks in OK state	Dynamic (DV=0) - BSM
nb_disk_slot_faulty	Number of disks in FAULTY state	Dynamic (DV=0) - BSM
nb_disk_slot_empty	Number of disks in EMPTY state	Dynamic (DV=0) - BSM
nb_disk_slot_used_spare	Number of disks slots in USED_SPARE state	Dynamic (DV=0) - BSM
nb_disk_slot_missing	Number of disks in MISSING state	Dynamic (DV=0) - BSM
nb_power_supply	Number of power supplies	Dynamic (DV=0) - BSM
nb_power_supply_ok	Number of power supplies in OK state	Dynamic (DV=0) - BSM
nb_power_supply_faulty	Number of power supplies in FAULTY state	Dynamic (DV=0) - BSM
nb_nb_fan	Number of fans	Dynamic (DV=0) - BSM
nb_fan_ok	Number of fans in OK state	Dynamic (DV=0) - BSM
nb_fan_faulty	Number of fans in FAULTY state	Dynamic (DV=0) - BSM
nb_nb_power_fan	Number of power_fan	Dynamic (DV=0) - BSM
nb_power_fan_ok	Number of power_fan in OK state	Dynamic (DV=0) - BSM
nb_ power_fan_faulty	Number of power_fan in FAULTY state	Dynamic (DV=0) - BSM
nb_nb_temperature_sensor	Number of temperature sensors	Dynamic (DV=0) - BSM
nb_ temperature_sensor _ok	Number of temperature sensors in OK state	Dynamic (DV=0) - BSM
nb_temperature_sensor _warning	Number of temperature sensors in WARNING state	Dynamic (DV=0) - BSM
nb_temperature_sensor _faulty	Number of temperature sensors in FAULTY state	Dynamic (DV=0) - BSM
nb_lun	Number of lun	Dynamic (DV=0) - BSM
nb_spare	Number of spare disk	Dynamic (DV=0) - BSM
serial_number	Serial number of the array	Dynamic - storegister
type	Type of the array: OSS, MDS, ADMIN.	preload

Field name	Field information	Fill in method
	Coded like UNIX rights (OMA, or – instead of the letter when the role does not apply)	
cfg_model	Name of the last applied model	Automatic - storemodelctl
cfg_model_application_date	Date of the last model application	Automatic - storemodelctl
mgmt_station_id	FK on HWMANAGER	preload
mgmt_node_id	FK on NODE	preload
status	Nagios status	Dynamic – BSM (DV="up")
unit_num	Unit Number	preload
comment	Free field	

Table 3-13. Storage – disk_array table

3.5.2.2 da_enclosure Table

Field name	Field information	Fill in method
id	Unique identifier for the disk enclosure in the database	preload –sequence
disk_array_id	Id of the parent array for this enclosure	preload
enclosure_type	Type of the disk enclosure	preload
rack_level	Level in the rack	preload
rack_label	Label of the rack containing the enclosure	preload
id_in_disk_array	Id of the enclosure in the array	preload
serial_number	Serial number of the enclosure	automatic – storeregister
model	Model of the disk enclosure	preload

Table 3-14. Storage – da_enclosure table

3.5.2.3 da_disk_slot Table

Field name	Field information	Fill in method
id	Unique identifier for the disk_slot in the database	Automatic - sequence
vendor	Vendor name of disk	Automatic - storeregister
model	Model of disk	Automatic - storeregister
serial_number	Serial number of disk	Automatic – storeregister
function	Function of disk: EMPTY, DATA, SPARE (DATA_VAULT, DATA_FLARE, SPARE_VAULT, SPARE_FLARE, etc.)	Automatic – storeregister
capacity	Disk capacity in MBs	Automatic – storeregister
enclosure_id	Id of the parent enclosure	Automatic - storeregister
disk_array_id	Id of the parent array for this disk_slot	Automatic - storeregister
state	State of the disk slot : EMPTY, OK, WARNING, FAULTY, MISSING, USED_SPARE	Dynamic – BSM
disk_enclosure_id	Disk number in the enclosure	Automatic - storeregister
vendor_location	Location of the component expressed in the vendor terms.	Automatic - storeregister

Table 3-15. Storage – da_disk_slot table

3.5.2.4 da_controller Table

Field name	Field information	Fill in method
id	Unique identifier for the controller in the database	Automatic – sequence
disk_array_id	Id of the parent array for this controller	Automatic – storeregister
enclosure_id	Id of the parent enclosure	Automatic - storeregister
State	State of the controller : OK , FAULTY, WARNING, OFF_LINE	Automatic – BSM
object_num	Controller identifier in the enclosure	Automatic – storeregister
vendor_location	Location of the component expressed in the vendor terms.	Automatic – storeregister

Table 3-16. Storage – da_controller table

3.5.2.5 da_fc_port Table

Field name	Field information	Fill in method
id	Unique identifier for the fc_port in the database	preload – sequence
wwn	World Wide Name of the host port.	Automatic – storeregister
alpa_port_id	Loop address of the port	Automatic – storeregister
disk_array_id	Id of the parent array for this fc_port	preload
enclosure_id	Id of the parent enclosure	preload
State	State of the host port : CONNECTED, NOT_CONNECTED, DISCONNECTED, FAULTY	Dynamic – BSM
object_num	fc_port identifier in the enclosure	preload
vendor_location	Location of the component expressed in the vendor terms.	Automatic – storeregister

Table 3-17. Storage – da_fc_port.table

3.5.2.6 da_serial_port Table

Field name	Field information	Fill in method
id	Unique identifier for the serial port in the database	preload – sequence
disk_array_id	Id of the parent array for this serial port	preload
enclosure_id	Id of the parent enclosure	preload
type	type of serial port	preload
port_server_id	Port_server linked to this serial connection	preload
port_server_port	Index of the port used on the portserver (start at 0)	preload
serial_ip	IP address used to access to this serial port	preload
serial_name	Name of the console for conman	preload
state	State of the serial port : CONNECTED, NOT CONNECTED, DISCONNECTED, FAULTY	Dynamic – BSM
object_num	Serial port identifier in the enclosure	Preload
vendor_location	Location of the component expressed in the vendor terms.	Automatic – storeregister

Table 3-18. Storage – da_serial_port table

3.5.2.7 da_ethernet_port Table

Field name	Field information	Fill in method
id	Unique identifier for the Ethernet port in the database	preload - sequence
name	Name attached to this IP address	preload
disk_array_id	Id of the parent array for this Ethernet port	preload
enclosure_id	Id of the parent enclosure for this Ethernet port	preload
eth_switch_id	Id of the parent Ethernet_switch or parent pap_node	preload
ipaddr	IP address of the Ethernet port	preload
macaddr	MAC address of the Ethernet port	Automatic – storeregister
vlan_id	Id of the VLAN containing this Ethernet port	preload
type	Type of the Ethernet port : PUBLIC, ADMIN	preload
state	State of the Ethernet port : CONNECTED, NOT CONNECTED, DISCONNECTED, FAULTY	Dynamic – BSM
object_num	Ethernet port identifier in the enclosure	preload – storeregister
vendor_location	Location of the component expressed in the vendor terms.	Automatic – storeregister
admin_eth_switch_slot	Arrival slot number on ETH SW	preload
admin_eth_switch_port	Connection port on the ETH SW	preload

Table 3-19. Storage – da_ethernet_port Table

3.5.2.8 da_power_supply Table

Field name	Field information	Fill in method
id	Unique identifier for the power supply in the database	Automatic – sequence
disk_array_id	Id of the parent array for this power supply	Automatic – storeregister
enclosure_id	Id of the parent enclosure for this power supply	Automatic – storeregister
state	State of the power supply : OK, FAULTY,MISSING, [WARNING]	Dynamic – BSM
object_num	Power supply identifier in the enclosure	Automatic – storeregister
vendor_location	Location of the component expressed in the vendor terms.	Automatic – storeregister

Table 3-20. Storage – da_power_supply table

3.5.2.9 da_fan Table

Field name	Field information	Fill in method
id	Unique identifier for the fan in the database	Automatic – sequence
disk_array_id	Id of the parent array for this fan	Automatic – storeregister
enclosure_id	Id of the parent controller for this power supply	Automatic – storeregister
state	State of the power supply: OK, FAULTY, MISSING, [WARNING]	Dynamic – BSM
object_num	Fan identifier in the enclosure	Automatic – storegister
vendor_location	Location of the component expressed in the vendor terms.	Automatic – storegister

Table 3-21. Storage – da_fan table

3.5.2.10 da_power_fan Table

Field name	Field information	Fill in method
id	Unique identifier for the power_fan in the database	Automatic - - sequence
disk_array_id	Id of the parent array for this power_fan	Automatic- storeregister
enclosure_id	Id of the parent enclosure for this power_fan	Automatic- storeregister
State	State of the power_fan: OK, FAULTY, MISSING, [WARNING]	dynamic – BSM
object_num	Power_fan identifier in the enclosure	Automatic- storeregister
vendor_location	Location of the component expressed in the vendor terms.	Automatic- storeregister

Table 3-22. Storage – da_power_fan table

3.5.2.11 da_temperature_sensor Table

Field name	Field information	Fill in method
id	Unique identifier for the temperature sensor in the database	Automatic – sequence
disk_array_id	Id of the parent array for this power supply (if controller_id and enclosure_id are NULL)	Automatic – storeregister
enclosure_id	Id of the parent enclosure for this power supply (if controller_id and array_id are NULL)	Automatic – storeregister
sensor_name	Name of the temperature sensor	Automatic – storeregister
state	State of the temperature sensor : OK, WARNING, FAULTY	Dynamic – BSM
object_num	Temperature sensor identifier in the enclosure	Automatic – storeregister
vendor_location	Location of the component expressed in the vendor terms.	Automatic – storeregister

Table 3-23. Storage – da_temperature_sensor table

3.5.2.12 da_io_path Table

Field name	Field information	Fill in method
node_id	Id of the node which access to this FC port	preload
port_id	Id of da_fc_port used by the node	preload
fc_board_id	Id of the HBA board	preload

Table 3-24. da_io_path table

3.5.2.13 da_iocell_component Table

Field name	Field information	Fill in method
iocell_id	Id of the IO cell	Preload - sequence
component_id	Id of a node or of a disk array	Preload
Type	Type of the component ("disk_array" or "node")	Preload
Rank	Rank of the node in the IO cell, or rank of the disk array in the IO cell. Start at 0.	preload

Table 3-25. Storage – da_iocell_component table

3.5.2.14 da_cfg_model Table

Field name	Field information	Fill in method
disk_array_id	Id of a disk array	Dynamic - storemodelctl
cfg_model_name	Model of a model which has been applied to the disk array	Dynamic - storemodelctl
application date	Date where the model has been applied	Dynamic - storemodelctl

Table 3-26. Storage – da_cfg_model table

3.5.2.15 da_power_port Table

Field name	Field information	Fill in method
id	Unique identifier for the power_port in the database	Preload sequence
disk_array_id	FK to disk array	preload
enclosure_id	FK to enclosure id	preload
talim_id	FK to T_ALIM	preload
talim_port	Plug to be powered on/off onT_ALIM	preload

Table 3-27. Storage – da_power_port table

3.5.3 Machine View

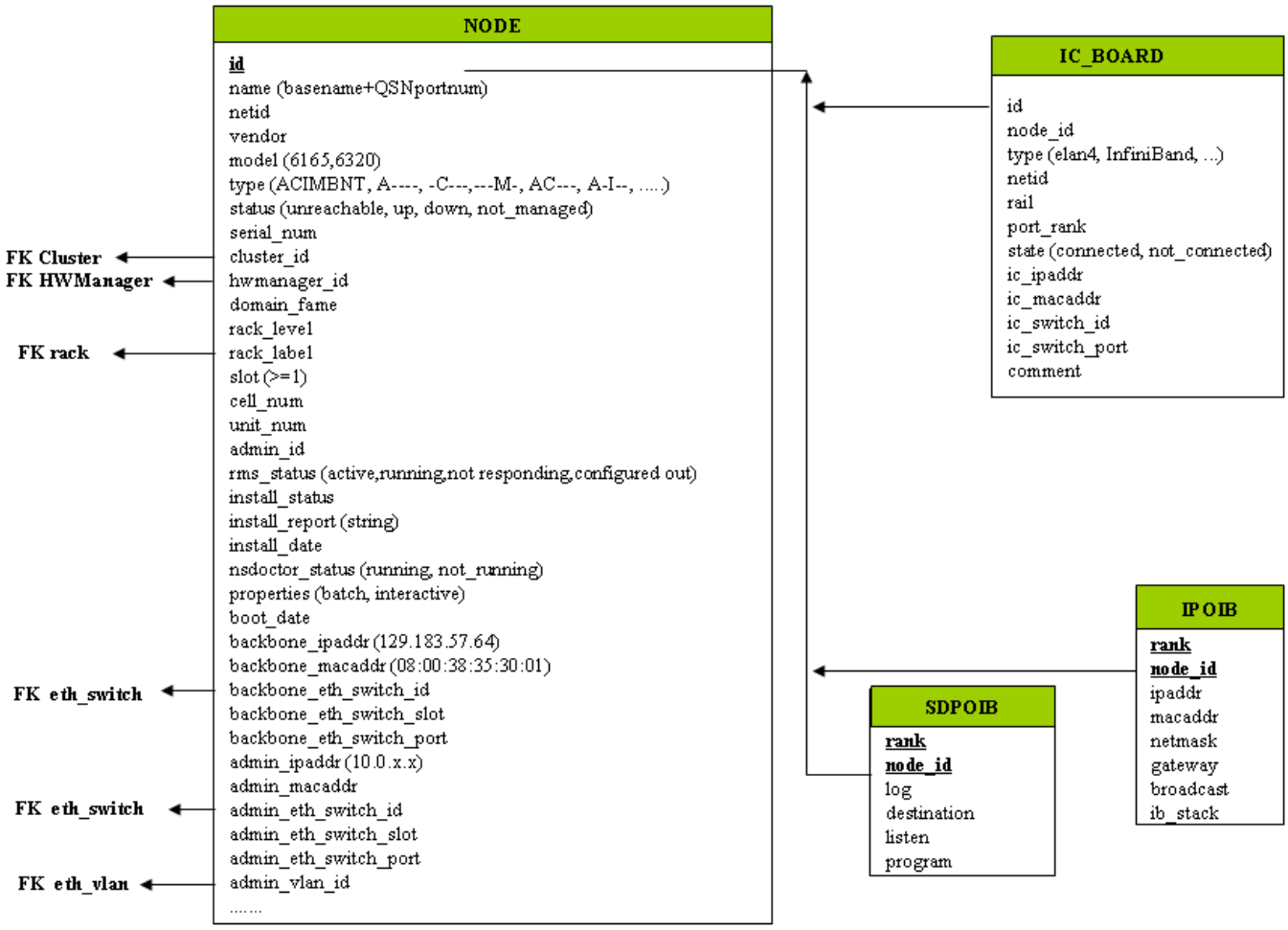


Figure 3-5. Cluster Database – Machine view 1

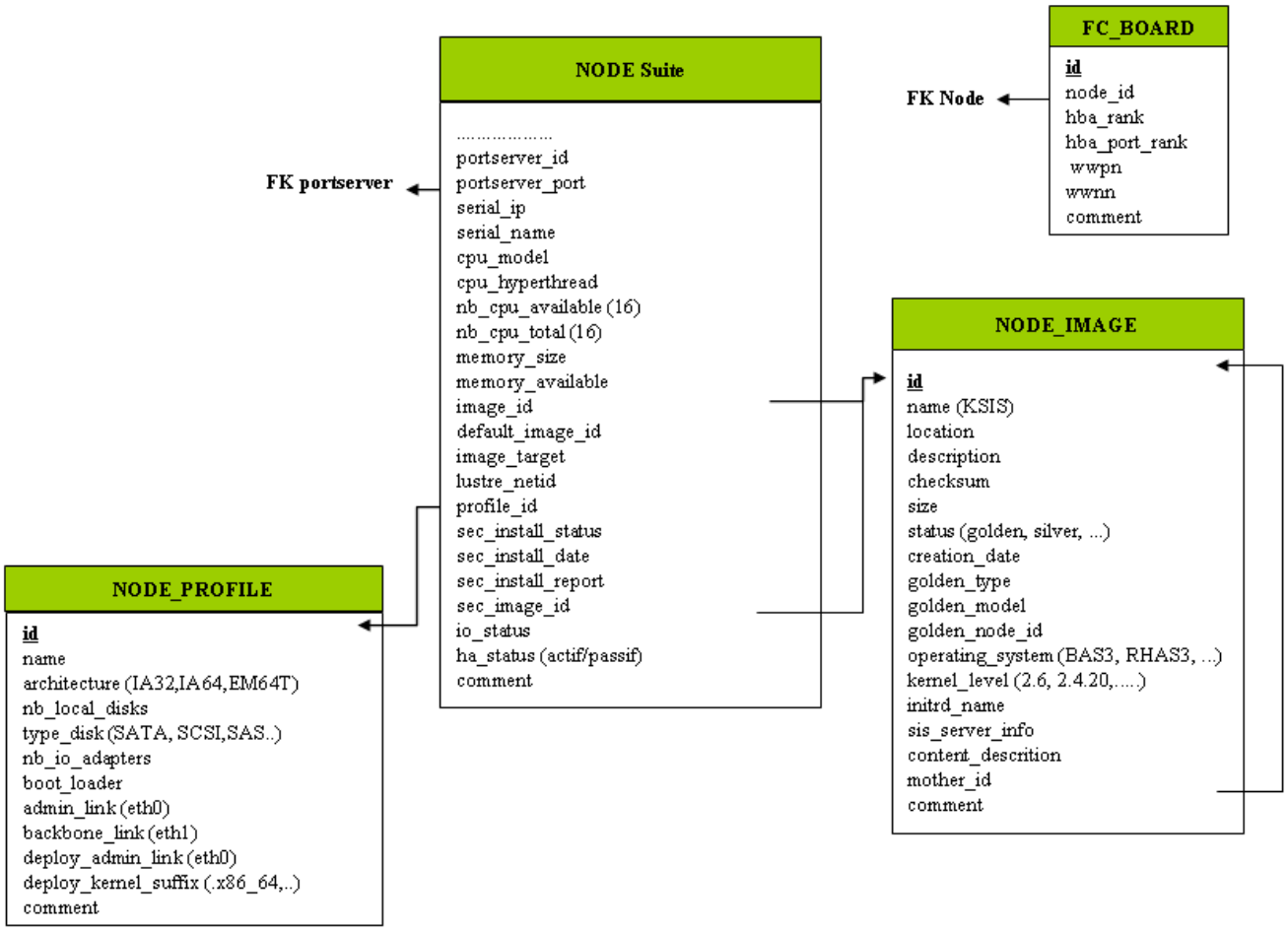


Figure 3-6. Cluster Database – Machine view 2

3.5.3.1 NODE Table

Column name	Description	Example	Fill-in method
id	primary key		preload– sequence
name	Node name	ns15	preload
netid	Node identifier number	1	preload
vendor	Name of vendor	Bull	preload
model	Node model	NS6165	preload
type	ACIMBNT node type, A------, -C-----, --I----, ---M---	A-IM---	preload
status	Nagios host_status up, down, unreachable	down	DV = up – Nagios
serial_num	Serial number		deprecated
cluster_id	FK on the CLUSTER		preload
hwmanager_id	FK on the HWMANAGER		preload
domain_fame	Machine name PAP side		deprecated
rack_level	Height in the rack	A	preload
rack_label	Rack name	CU0-A5	preload
slot	Number of slot for the node [1-14]	1	preload
cell_num	Cell number	1	preload

unit_num	Unit ID	3	preload
admin_id	FK towards ADMIN		admin
rms_status	RMS status	configure out	event handler
install_status	KsiS Status	not_installed	KsiS
install_report	message	Host not installed	KsiS
install_date	System installation date	13/12/04 10 :30 :10	KsiS
NsDoctor_status	running or not-running	not-running	NsDoctor + DV
properties	Torque properties	Batch	Torque + DV
boot_date	Date of the last boot		PostBootChecker
backbone_ipaddr	Backbone IP Address	129.183.57.64	Preload
backbone_eth_switch_id	FK on the ETH_SWITCH		Preload
backbone_eth_switch_slot	Arrival slot number on ETH SW		Preload
backbone_macaddr	mac address	NULL	
backbone_eth_switch_port	Connection port for BK_ETH_SW	2	Preload
admin_ipaddr	Admin IP address	10.1.0.1	Preload
admin_eth_switch_id	FK on the ETH_SWITCH	1	Preload
admin_eth_switch_slot	Arrival slot number on ETH SW		Preload
admin_eth_switch_port	Connection port for AD_ETH_SW	5	Prelaod
admin_vlan_id	FK for ETH_VLAN		Preload
admin_macaddr			nodeRecord or equipmentRecord
portserver_id	FK on the PORTSERVER		Preload
portserver_port	Port number for the PS		Preload
serial_ip	Serial line access IP address	129.183.75.10	Preload
serial_name	Name of the serial number	ns15s	Preload
cpu_model	CPU model	Montecito	Preload
cpu_hyperthread	Boolean	True	PostBootChecker
nb_cpu_available	Number of CPUs available	15	PostBootChecker
nb_cpu_total	Number of CPUs	16	Preload
memory_size	Memory size	64	Preload
memory_available	Size of memory available	64	PostBootChecker
image_id	FK on the NODE_IMAGE		KsiS
default_image_id	FK on the default image		KsiS
image_target	For future use	NULL	NULL
lustre_netid	For future use	NULL	NULL
profile_id	FKon the NODE_PROFILE		Preload
sec_install_status	Secondary image KSiS status		KSiS
sec_install_date	Secondary Image installation date		KSiS
sec_install_report	Secondary Image message		KSiS
sec_image_id	FK of the NODE_IMAGE		KSiS
io_status	I/O status of the node		storage
ha_status (active/passive)	HA status of the node		Cluster Suite
post_config_status	PostConfig Status		KSiS
comment	Free field	NULL	

Table 3-28. Machine view – NODE table

3.5.3.2 NODE_IMAGE Table

Column name	Description	Example	Fill-in method
id	PK		Sequence
name	Name of the image	try	KsiS
location	localisation	/path/name	KsiS
description	description		KsiS
checksum	checksum	12352425	KsiS
size	Image size		KsiS
status	image status	= golden, silver	KsiS
creation_date	date	=JJ/DD/YY HH :MI :SS	Trigger
golden_type	IO, HPC, MDS, ADMIN		KsiS
golden_model	6165,6320		KsiS
golden_node_id	id of node serving as the golden node		KsiS
operating_system	Distribution type	BAS5V2	KsiS
kernel_level	Kernel level	6.2	KsiS
initrd_name	Initrd name		KsiS
sis_server_info	name/version		KsiS
content_description	description of the image content		KsiS
mother_id	Link to original image		KsiS
comment	Free field		

Table 3-29. Machine view – NODE_IMAGE table

3.5.3.3 NODE_PROFILE Table

Column name	Description	Example	Fill in method
id	Primary Key	1	preload sequence
name	Name used to recognise the profile	SERV-A	preload
architecture	Type of architecture IA64, EM64T,etc.	IA64	preload
nb_local_disks	Number of internal disks	3	preload
type_disk	Type of disks (SATA, SCSI,SAS, etc)	SATA	preload
nb_io_adapters	Number of I/O cards	2	preload
boot_loader	elilo, grub	grub	KSiS
admin_link	admin interface (eth0)	eth0	DV
backbone_link	Interface backbone (eth1)	eth1	DV
deploy_admin_link	Deployment interface	eth0	DV
deploy_kernel_suffix	Kernel suffix (.x86_64, .x86_64G, etc.)	NULL	DV
comment	Free field		

Table 3-30. Machine view – NODE_PROFILE table

3.5.3.4 IC_BOARD Table

This table describes Interconnect parameters (**Quadrics**, **Infiniband** or **GBEthernet**).

Column name	Description	Example	Fill in method
id	Primary Key	1	preload sequence
node_id	FK on NODE	1	preload

type	type of card	elan4, infiniband	preload
netid	Node identifier number	3	preload
rail	Number of rail	0	preload
port_rank	Port number on the card	1	preload
state	Status of the port (connected, not_connected)	connected	preload
ic_ipaddr	IP address of the IC Board	10.0.10.3	preload
ic_macaddr	Mac address	unused	
rail	Number of the rail	2	preload
ic_switch_id	FK on IC_SWITCH		preload
ic_switch_port	Number of the IC_SWITCH port	64	preload
comment	Free field		

Table 3-31. Machine view – IC_BOARD table

3.5.3.5 IPOIB Table

This table describes Infiniband parameters for storage access.

Column name	Description	Example	Fill in method
rank	PK, Rank of the Infiniband adapter	0	updateIPOIB
node_id	PK, reference NODE	10	updateIPOIB
ipaddr	ip address on Infiniband	172.193.1.1	updateIPOIB
macaddr	Mac address		updateIPOIB
gateway	ip address of the gateway		updateIPOIB
broadcast	ip address of the broadcast		updateIPOIB
ib_stack	type of stack IP, SDP, BOTH	SDP	updateIPOIB

Table 3-32. Machine view – IPOIB Table

3.5.3.6 SDPOIB Table

Column name	Description	Example	Fill in method
rank	PK, Rank of the Infiniband adapter	0	updateSDPoIB
node_id	PK, reference NODE	10	updateSDPoIB
log	Log in sdplib.conf		updateSDPoIB
destination	Destination in sdplib.conf		updateSDPoIB
listen	Listen in sdplib.conf		updateSDPoIB
program	Program in sdplib.conf		updateSDPoIB

Table 3-33. Machine view – SDPOIB table

3.5.3.7 FC_BOARD table

Note This table only applies to systems which include a Storage Area Network (SAN).

Column name	Description	Example	Fill in method
id	Primary key		storage
node_id	FK on the node	1	storage
hba_rank	Rank of the adapter		storage
hba_port_rank	Rank of the port		storage

wwpn	World Wide Port Name		storage
wwnn	World Wide Node Name		storage
comment	Free field		

Table 3-34. Machine view – FC_BOARD table

3.5.4 HWMANAGER View

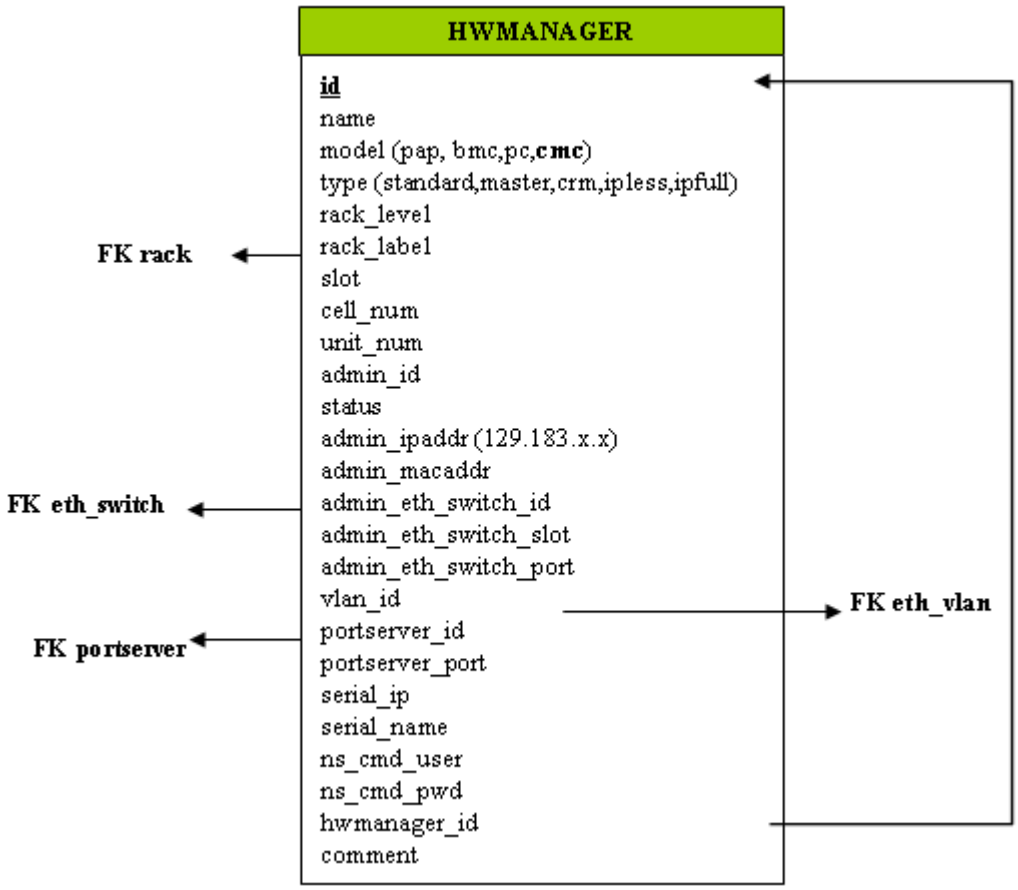


Figure 3-7. HWManager view

3.5.4.1 HWMANAGER Table

Column name	Description	Example	Fill in method
id	Primary key		preload - Sequence
name	HWMANAGER IP name	papu1c2	preload
model	Pap or bmc or pc or cmc	pap	preload
type	standard, master, crm, ipless, ipfull	standard	preload
rack_level	Height in the rack	E	preload
rack_label	Name of the rack	ISO0-H45	preload
cell_num	Number of the cell	3	preload
unit_num	Number of the unit	1	preload
admin_id	ADMIN id		admin
status	Nagios status	unreachable	DV=up – Nagios
admin_ipaddr	Admin IP address		preload

Column name	Description	Example	Fill in method
admin_macaddr	Mac address		updateMacAddr
admin_eth_switch_id	ETH_SWITCH id		preload
admin_eth_switch_slot	Arrival slot number on ETH SW		preload
admin_eth_switch_port	ETH_SWITCH connection port	2	preload
vlan_id	ETH_VLAN id		preload
portserver_id	PORTSERVER id		preload
portserver_port	Portserver port number		preload
serial_ip	Serial line access IP address	129.183.75.10	preload
serial_name	HWMANAGER serial name	papu1c2s	preload
ns_cmd_user	User NC Commande	nsc	preload
ns_cmd_pwd	password	$nsc	preload
hwmanager_id	FK on HWMANAGER		preload
comment	Free field		

Table 3-35. HWMANAGER Table

3.5.5 Complementary Tables

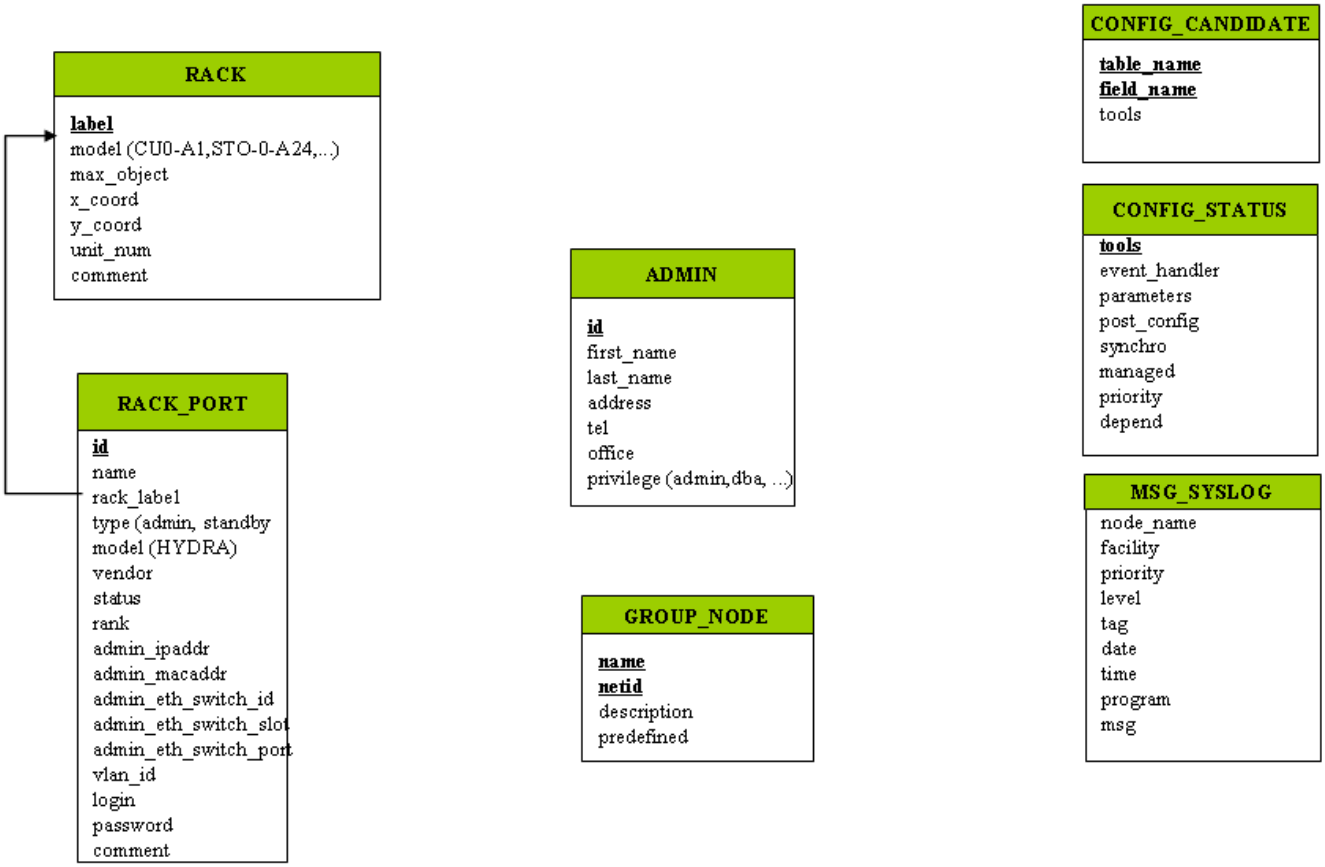


Figure 3-8. Cluster Database – Complementary tables

3.5.5.1 ADMIN Table

Column name	Description	Example	Fill in method
id	PK		Sequence
first_name	First name	Stephane	admin
last_name	surname	Dupont	admin
address	address	…	admin
tel	Phone number		admin
office	office		admin
privilege	admin, dba, etc.		admin

Table 3-36. ADMIN table

3.5.5.2 RACK Table

Column name	Description	Example	Fill in method
label	PK	RACK1	preload
model	Type of rack	ARM3	preload
max_object	Maximum number of objects in the rack	3	preload

x_coord	Abscissa in the rows of racks		preload
y_coord	Ordinate in the length of racks		preload
unit_num	Number of theUnit	5	unused
comment	Free field		

Table 3-37. RACK table

3.5.5.3 RACK_PORT Table

Column name	Description	Example	Fill in method
id	PK		Sequence
label	FK on RACK	RACK1	preload
type	Port yype	Admin	preload
model	Card model	HYDRA	preload
vendor	Vendor name	Bull	preload
status	Nagios host status	Up	DV = up - Nagios
rank	Port instance	1	preload
admin_ipaddr	Admin IP address	172.16.118.190	preload
admin_macaddr	Port Mac address		equipmentRecord
admin_eth_switch_id	FK on ETH_SWITCH		preload
admin_eth_switch_slot	Arrival slot number on ETH_SW		preload
admin_eth_switch_port	Connexion port on ETH SW	3	preload
vlan_id	FK on ETH_VLAN		preload
login	Administration login		DV = super - preload
password	Administration password		DV = pass - preload
comment	Free field	NULL	

Table 3-38. RACK_PORT table

3.5.5.4 CONFIG_CANDIDATE Table

Column name	Description	Example	Fill in method
table_name	PK	node	creation
filed_name	PK	admin_ipaddr	creation
tools	list of the candidates tools	nagios, conman	creation

Table 3-39. CONFIG_CANDIDATE table

3.5.5.5 CONFIG_STATUS Table

Column name	Description	Example	Fill in method
tools	PK	nagios	creation
event_handler	generator of conf file	initNagiosCfg	creation
parameters	parameters of the event handlers	1,5,10	trigger
post_config	service to restart	nagios	creation
synchro	boolean, to be synchronized	True	trigger - dbmConfig
managed	Deactivation of the tool	True	creation
priority	Synchronisation order	1	creation

depend	List of the inter-dependency of the tool	group	creation

Table 3-40. CONFIG_STATUS table

3.5.5.6 GROUP_NODE Table

Column name	Description	Example	Fill in method
name	PK	graphique	dbmGroup
netid	PK	10-20,25,30	dbmGroup
description	Comment about the group		dbmGroup
predefined	Predefined group	True	dbmGroup

Table 3-41. GROUP_NODE table

3.5.5.7 MSG_SYSLOG Table

This table is not active in this version.

3.5.6 Nagios View

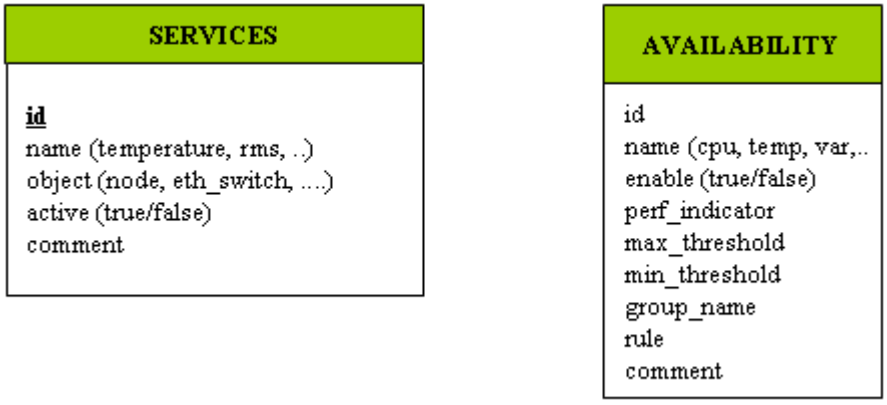


Figure 3-9. Nagios View

3.5.6.1 SERVICES Table

Column name	Description	Example	Fill in method
id	Service id		dbmConfig
name	Service name	temperature	dbmConfig
object	Node , Eth_switch, portserver, etc.	node	dbmConfig
actif	Status of the service	true	Config & dbmServices
comment	comment	Temperature of the node	dbmConfig

Table 3-42. SERVICES Table

3.5.6.2 AVAILABILITY Table

Column name	Description	Example	Fill in method
id	Service id		
name	CPU, temp, var	cpu	
enable	To check (true / false)	true	
perf_indicator	Performance indicator	true	
max_threshold	Maximum threshold		
min_threshold	Minimum threshold		
group_name	Application group		
rule	Criterion rule		
comment	comment		

Table 3-43. AVAILABILITY Table

3.5.7 Lustre View

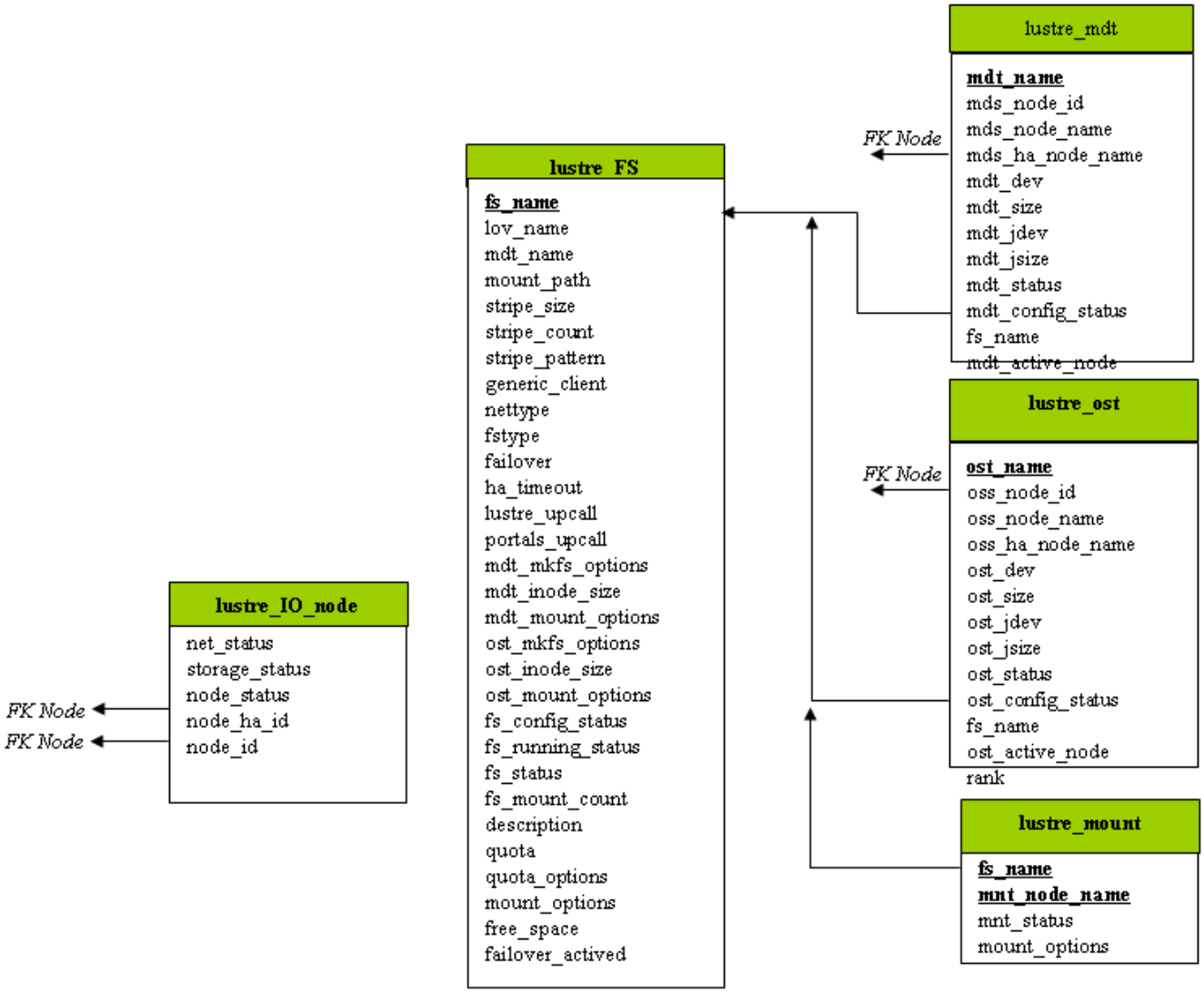


Figure 3-10. Cluster Database – Lustre view

3.5.7.1 lustre_FS Table

Each entry of the table describes a Lustre file system currently installed on the cluster.

Column name	Description	Example	Fill in method
fs_name	File system name	lustre_basic	lustre_config
mount_path	File system mount path	/mnt/lustre_basic	lustre_config
lov_name	LOV identification	lov_lustre_basic	lustre_config
mdt_name	MDT reference	mdt _ns44_1	lustre_config
stripe_size	Stripe size	4MB	lustre_config
stripe_count	Number of stripe per file	0 (all included OSTs)	lustre_config
stripe_pattern	Striping mode	0 (RAID0)	lustre_config
generic_client	Generic client profile	« client »	lustre_config

nettype	Network type	elan	lustre_config
fstype	Back-end file system type	ldiskfs	lustre_config
failover	High-Availability indicator	« YES »	lustre_config
ha_timeout (Deprecated - do not use)	High-Availability timeout for Compute Nodes	30	lustre_config
lustre_upcall	Lustre Exception processing script	/usr/bin/lustre_upcall	lustre_config
Portals_upcall	Portals layer exception processing script	/usr/bin/lustre_upcall	lustre_config
mdt_mkfs_options	MDT formatting options	mkfs command semantic	lustre_config
mdt_inode_size	Inode size for MDT back-end file system	1024	lustre_config
mdt_mount_options	MDT mount options	Mount command semantic	lustre_config
ost_mkfs_options	OSTs common formatting options	mkfs command semantic	lustre_config
ost_inode_size	Inode size for OSTs back-end file systems	1024	lustre_config
ost_mount_options	OSTs mount options	Mount command semantic	lustre_config
fs_config_status	File system configuration status		lustre_config
fs_running status	File system current running status		Lustre monitoring tools
fs_status	File system status		Lustre monitoring tools
fs_mount_count	File system mount counter	54	lustre_util
description	File system characteristics decription		lustre_config
quota	User quotas management indicator	"YES"	lustre_config
quota_options	Quotas management tuning options		lustre_config
mount_options	Default mount options for the file system		lustre_config
free_space	Size of the file system in GB	773	lustre_util
failover_activated	For future use	yes	lustre_util

Table 3-44. Lustre_FS table

3.5.7.2 lustre_ost Table

Each entry of the table describes an OST available on the cluster.

Column name	Description	Example	Fill in method
ost_name	OST logical name	OST_ns32_1	lustre_investigate
oss_node_id	OSS ident in the node table	5	lustre_investigate
oss_node_name	Supporting OSS node name	ns32	lustre_investigate
oss_ha_node_name	Secondary OSS node name	ns33	lustre_investigate
ost_active_node	In case of HA management, current node name support	ns32	lustre_migrate
ost_dev	OST back-end device name	/dev/ldn.45.1	lustre_investigate
ost_size	OST back-end device size	140000000000	lustre_investigate
ost_jdev	External journal device name	/dev/ldn.45.2	lustre_investigate
ost_jsize	External journal device size	100000	lustre_investigate

Column name	Description	Example	Fill in method
ost_config_status	OST service configuration status		lustre_config
ost_status	OST service running status		Lustre management tools
fs_name	Proprietary file system name	lustre_basic	lustre_config

Table 3-45. Lustre OST table

3.5.7.3 lustre_mdt Table

Each entry of the table describes an MDT available on the cluster.

Column name	Description	Example	Fill in method
mdt_name	MDT logical name	MDT_ns32_1	lustre_investigate
mds_node_id	MDS ident in the node table	5	lustre_investigate
mds_node_name	Supporting MDS node name	ns32	lustre_investigate
mds_ha_node_name	Secondary MDS node name	ns33	lustre_investigate
mdt_active_node	In case of HA management, current node name support	ns32	lustre_migrate
mdt_dev	MDT back-end device name	/dev/ldn.45.1	lustre_investigate
mdt_size	MDT back-end device size	140000000000	lustre_investigate
mdt_jdev	External journal device name	/dev/ldn.45.2	lustre_investigate
mdt_jsize	External journal device size	100000	lustre_investigate
mdt_config_status	MDT service configuration status		lustre_config
mdt_status	MDT service running status		Lustre management tools
fs_name	Proprietary file system name		lustre_config

Table 3-46. Lustre_MDT Table

3.5.7.4 lustre_IO_node Table

Each cluster node of I/O (I) or metadata (M) type has an entry in this table.

Column name	Description	Example	Fill in method
node_id	Ident of the node in the node table	ns32	preload
node_ha_id	Ident of the HA paired node in the node table	ns33	preload
net_status	Node network status	% available (0 – 33 – 66 – 100)	Lustre monitoring tools
storage_status	Node storage status	% available (0 – 12 – 25 - … - 100)	Lustre monitoring tools
node_Status	Node lustre status		Failover tools

Table 3-47. Lustre_IO_node table

3.5.7.5 lustre_mount Table

Each entry of this table refers to a couple compute node / mounted Lustre file system.

Column name	Description	Example	Fill in method
mnt_node_name	Compute node name	ns87	lustre_util
nnt_status	Mount point status		lustre_util
fs_name	File system name		lustre_util
mount_options	Lustre file system current mount options for the compute node		lustre_util

Table 3-48. Lustre_mount table

Chapter 4. Software Deployment (KSIS)

This chapter describes how to use KSIS to deploy, manage, modify and check software images. The following topics are described:

- 4.1 *Overview*
- 4.2 *Configuring and Verifying a Reference Node*
- 4.3 *Main Steps for Deployment*
- 4.4 *Checking Deployed Images*
- 4.5 *Ksis Commands*
- *4.6 Building a Patch*
- *4.7 Checking Images*
- 4.8 *Importing and Exporting an Image*
- 4.9 *Rebuilding ClusterDB Data before Deploying an Image*

4.1 Overview

A deployment tool is a piece of software used to install a distribution and packages on several machines at once. For large clusters, such a tool is essential, since it avoids doing the same installation a large number of times. **KSIS** is the deployment tool used on Bull extreme computing systems.
KSIS makes it easy, for a network of Linux machines, to propagate software distributions, content or data distribution changes, operating system and software updates.
KSIS is used to ensure safe production deployments. By saving the current production image before updating it with the new production image, a highly reliable contingency mechanism is provided. If the new production environment is found to be flawed, simply roll-back to the last production image.

This chapter describes how to:

- Create an image for each type of node and save it on the image server. These images are called reference/golden images. The image server is on the Management Node and is operated by the KSIS server software.
- Deploy the node images.
- Manage the evolution of the images (**workon** images and patches).
- Check discrepancies between an image on a node and its reference on the image server.

The deployment is done using the administration network.

Note The terms **reference node** and **golden node** are interchangeable. The same applies to the terms **reference image** and **golden image**.

4.2 Configuring and Verifying a Reference Node

A reference node is a node which has had all the software installed on to it, and whose image is taken and then stored on the image server. The reference image will be deployed onto the other nodes of the cluster.

Installation and Configuration

Reference nodes have the **bullx cluster suite** software installed on to them in the same way as ordinary COMPUTE/COMPUTEX or I/O nodes. A **KSIS client** is then installed on to these nodes from the bullx cluster suite media. The operating system and applications must be installed and configured to make the node operational.

4.3 Main Steps for Deployment

Once the image server, reference nodes and client nodes are ready, the steps for the deployment are:

1. Create the image of the reference node to be saved on the Image Server:

```
ksis create <imageName> <ReferenceNodeName>
```

   This command requests that a check level is chosen. Choose “basic”.

2. Deploy the image:

```
ksis deploy <imageName> node[1-5]
```

Note See *Deploying an Image or a Patch*, on page 4-9 for more details about the deployment process.

The following figure shows the creation and deployment of an image.

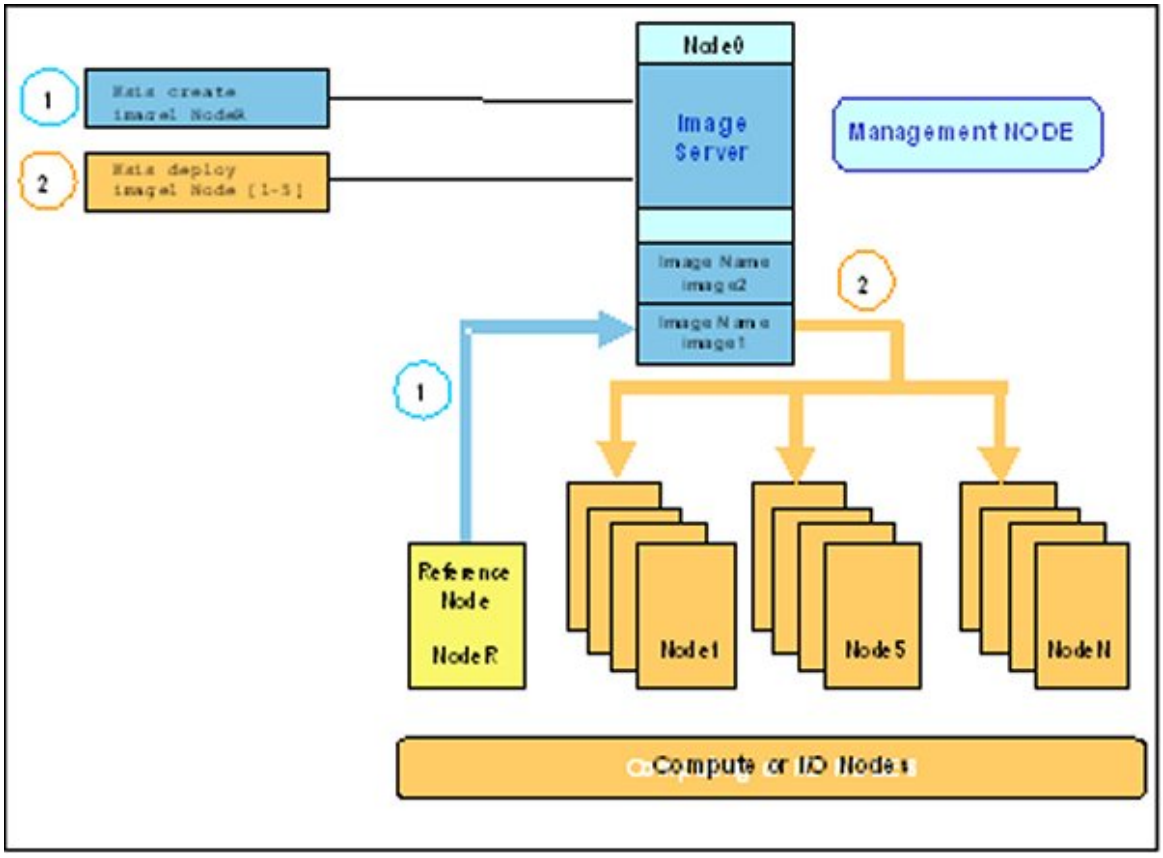


Figure 4-1. Main steps for deployment

4.4 Checking Deployed Images

The **ksis check** command is used to compare the files on a node(s) with the reference image deployed. This is done by listing the discrepancies for tests performed on the node(s), when compared with the results of the same tests on the reference image.

The general form for the **check** command is:

ksis check nodeRange [-t testGroup] [-d]

Options

[-t testGroup] Specify the **Test Group** for the checks. 3 Test Groups exist: **basic**, **basic+** and **sharp.**

[-d] View differences between the node(s) and the reference image, if they can be displayed on a few lines.

Examples

```
ksis check nc[2-45]
```

```
ksis check nc[2-45] -t basic -d
```

Note Nodes inside a node range are not always created from the same image.

4.4.1 Checking Principles

The descriptions of the image tests are stored in a database (**/etc/systemimager/ksis_check_Repository/** on the Management Node). When creating an image or a patch, the Administrator specifies the **Test Group** for an image or patch. **KSIS** then executes the commands associated with each check in the **Test Group**, and stores the results as a *reference*. This *reference* is then included in the image.

Each time the **ksis check** command is used, **KSIS** executes the checks defined for the **Test Group** on each node and generates the results. If there is a discrepancy between the results and the *reference* results, the check is set to KO, otherwise it is set to OK. The image server centralizes the node check results.

4.4.2 Ksis Tests and Test Groups

Test Name	Test Group	OK	KO
CheckRpmList	**Basic & Basic+**	List of RPMs installed on the node is the same on the reference image.	List of RPMs installed on the node is not the same on the reference image.
CheckRpmFiles	**Sharp**	None of the files delivered using the RPM seems to have been updated regarding contents and/or access rights.	One or more of the files delivered using RPM seems to have been updated regarding contents and/or access rights.
CheckFastRpmFiles	**Basic & Basic+**	None of the files delivered using RPM seems to have been updated regarding length, date, and/or access rights.	One or more of the files delivered using RPM seem to have been updated regarding length, date, and/or access rights.
CheckSRTDir	**Basic+**	None of the files of the deployed image seems to have been updated regarding length, date, and/or access rights.	One or more of the files of the deployed image seem to have been updated regarding length, date, and/or access rights.
CheckMd5sumDir	**Sharp**	None of the files of the deployed image seems to have been updated regarding content (md5 on the content).	One or more of the files of the deployed image seem to have been updated regarding content (md5 on the content).
CheckMandatoryFiles	**Basic & Basic+**	Ksis binaries are present on the node and have the same length as those on the Management Node.	Ksis binaries are not present on the node or have not the same length as those on the Management Node.
CheckUsedKernel	**Basic & Basic+**	Kernel used by the node is the same as the one used on reference/golden node when the image has been created.	Kernel used by the node does not look the same as the one used on reference/golden node when the image has been created.

Table 4-1. Standard checks delivered with Ksis

Each test belongs to one, or more, of the 3 **Test Groups** possible; **Basic**, **Basic+** and **Sharp**. Each **Test Group** includes a combination of the tests listed in Table 4-1, for example **Basic** level tests include the following tests: **CheckRpmList**, **CheckFastRpmFiles**, **CheckMandatoryFiles**, and **CheckUsedKernel**. **Sharp** includes all the **Basic**, and **Basic+** checks, in addition to its own checks.

If the -t option is not specified all the checks are executed.

4.4.3 Modifying the Checks Database

It is possible to modify the checks in the database, in order to adapt them to the way you use the image.

- To create a new **Test Name**, create a new directory **(/etc/systemimager/ksis_check_Repository/<testName>.vid)** which includes at least the following:
  - **command** file, which contains the command(s) to be run for the test.
  - **Test group**, which defines the Test groups for the test.

The new **Test Name** will be included in the checks database, and will be part of the checks performed on subsequent images.

4.4.4 Examining the Check Results

The result of the checks carried out includes a comparison between the results of the command(s) executed on the reference image and on the node(s) specified. This comparison shows the evolution of the node(s) against the reference image, and can be used to determine the necessity of deploying the reference image again.

If the discrepancies between a node and the reference image are not significant, it may still be useful to analyze their development using the **checkdiff** command.

ksis checkdiff command

The **ksis checkdiff** command displays the discrepancies between the reference image and the results for a given test on a node. The general form for the **checkdiff** command is:

ksis checkdiff testName nodeRange

Example

```
ksis checkdiff CheckSRTDir node2
```

4.5 Ksis Commands

4.5.1 Syntax

```
ksis <action> <parameters> [<options>]
```

Options:

-S Step by step

-v Verbose

-g Debug mode

-G Detailed debug mode

Format for nodeRange or groupName parameter:

The nodes, to which the Ksis command applies, are specified either as a range of nodes (**nodeRange**) or as a group name (**groupName**).

- Several formats are possible for the **nodeRange** parameter, as shown in the following examples:
  - <nodeRange> = host[1]
  - <nodeRange> = host[1,2,3,9]
  - <nodeRange> = host[1-3]
  - <nodeRange> = host[1-3,9]
- The **groupName** is the name of a group of nodes defined in the ClusterDB. See the *Cluster Database Management* chapter for more information about these groups.

Getting Help:

For a complete description of the KSIS commands, enter:

```
ksis help
```

Or:

```
ksis help <action>
```

4.5.2 Advanced ksis create options

-d

The **–d** option is used to define the individual disks of a node, which are to be included in the image.

```
ksis create <myImage> <myReferenceNode> -d <myDisks>
```

The disks to be included appear after the **–d** option in a comma-separated list, as shown in the example below. The node disks not listed will not be included in the image.

Example

```
ksis create MyImage MyGolden -d /dev/sda,/dev/sdb
```

In the command above only disks **sda** and **sdb** will be included in the image.

-dx

The **-dx** option is used in the similar fashion to the **–d** option. The only difference is that this option is exclusive. In other words, unlike the **–d** option, all the references to the mounted disks which are not included in the image will be deleted and the **/etc/fstab** file which lists the mounts points will be updated.

When to use the –d and –dx options

The **–dx** option is used, for example, if for some reason it is decided that a particular disk bay (e.g. **/dev/sdj**) connected to the reference node, should not be included in an image when it is deployed.

When the **–d** option is used, if declared in the **/etc/fstab** file, the disk(s) specified (e.g. **/dev/sdj**) will be remounted on all the newly deployed nodes. By using the **–dx** option with the **ksis create** command all references to the **/dev/sdj** bay are deleted, and it will not be remounted after deployment.

4.5.3 Creating the Image of the Reference Node

To create an image of the reference node use the **ksis create** command. This operation is done while you are logged onto the image server (Management Node).

```
ksis create <imagename> <reference_node_name> [options]
```

This command creates a copy of the image of the reference node on the image server (Management Node). The resulting status for this image is "golden".

When using this command the check level associated with this image is requested. Choose **basic** for a standard level (see *4.4 Checking Deployed Images* for other options).

4.5.4 Deleting an Image or a Patch

This command deletes the defined image or patch from the image server (Management Node).

```
ksis delete <imageNameOrPatchName>
```

4.5.5 Deploying an Image or a Patch

This command deploys an image or a patch on the nodes specified:

```
ksis deploy <imageNameOrPatchName> <nodeRangeOrGroupName> [options]
```

When you deploy an image the command performs these steps on the nodes concerned:

- Checks the state of the node
- Reboots the node in network mode
- Loads the image from the image server using special algorithms to parallelize the loading and to minimize the loading time
- Checks log files
- Boots the node with the image loaded

See

- *Installation and Configuration Guide* for details on the deployment procedure, including post deployment operations.
- *Maintenance Guide* for more details on the **Ksis** log files.

4.5.6 Removing a Patch

This action concerns only the images with the 'patch' status. It removes the last patch deployed from the nodes.

```
ksis undeploy <patchName> <nodeRangeOrGroupName> [options]
```

4.5.7 Getting Information about an Image or a Node

This command displays information for the specified image or node.

```
ksis show <imageNameOrNodeName>
```

4.5.8 Listing Images on the Image Server

This command gives the list and status of the images available on the image server. Their status is one of the following:

```
ksis list [<options>]
```

golden reference image (from a reference node - also called golden node).

patch patch (result of a store command).

patched golden modified reference image (result of a detach command).

working patch modification in progress; cannot be deployed, waiting for store command.

Example:

```
ksis list
```

```
Image Name                Status           Creation Date
BAS3-v13u1u2              golden           2005-01-14 14:33:02
Compute_hpceth_u1u2       golden           2005-01-14 15:41:25
Compute_hpceth_u1u2.s1.0 patch             2005-01-20 13:49:27
Compute_hpceth_u1u2.s1.1 working patch     2005-01-22 14:41:03
```

4.5.9 Listing Images by Nodes

This command lists the current images available and their status on the nodes.

```
ksis nodelist [<options>]
```

Example:

```
ksis nodelist
```

```
nc1  unreach  -
nc2  up Compute_hpceth_u1u2        2005-01-20 11:28:30
nc3  up Compute_hpceth_u1u2        2005-01-20 11:29:33
nc4  up Compute_hpceth_u1u2.s1.0   2005-01-21 12:03:01
nc5  down  Compute_hpceth_u1u2.s1.0   2005-01-21 12:10:43
```

4.6 Building a Patch

ksis buildpatch is used to create a patch from the differences between two images. This can then be used to transform the software structure and content of the first node which has had the first image deployed on it so that it matches a node which has had the second image deployed on it.

Note **ksis buildpatch** can only be used for two images which are derived from each other and not for images which are unrelated.

The command below would create a patch from the differences between the **<imageName1>** image and the **<imageName2>** image.

```
ksis buildpatch <imageName1> <imageName2>
```

Using ksis buildpatch

1. Make any changes required to the deployed version of the **<imageName1>** image. This is done by logging on to a node **n** which has **<imageName1>** on it and changing whatever needs to be changed. If necessary reboot on the node and check that everything is working OK.

2. Create an image of the node which has the **<imageName1>** image on it.

```
ksis create <imageName1> n
```

3. Create a patch of the differences between the **<imageName1>** and **<imageName2>** images. The patch will be automatically name e.g.**ImageName1.s1.0** for the first patch generated for **<ImageName1>** image.

```
ksis buildpatch <imageName1> <imageName2>
```

4. Deploy this patch on to the nodes which have **<imageName1>** on them.

```
ksis deploy <patch_name> <nodelist>
```

5. These nodes will now have a software content and structure which matches **<imageName2>.**

ksis buildpatch and the use of patches should only be applied for limited image changes. For fundamental image changes the best method remains the creation and the deployment of a new image.

4.7 Checking Images

The **check** command checks the image deployed on a node set.

```
ksis check <nodeRangeOrGroupName>
```

The **checkdiff** command displays the discrepancies between a reference node and the results for a given check on a given node.

```
ksis checkdiff <testName> <node>
```

4.8 Importing and Exporting an Image

KSIS provides a function to export an image to another **KSIS** installation (on another Management Node) or to import an image from another **KSIS** installation.

The **ksis export** command allows you to export a Reference image (not a Patch image). The image will be available as a tar file in the Ksis images directory: **/var/lib/systemimager/images/<imageName>.tar**

```
ksis export <imageName> [<options>]
```

Note The export operation does not automatically destroy the exported image.

The **KSIS import** command allows you to import a Reference image from a tar file in the KSIS images directory: **/var/lib/systemimager/images/<imageName>.tar.**

The **import** command imports an image previously exported from another cluster.

```
ksis import <imageName> [<options>]
```

Once the import operation is completed, the image is available and may be listed by using the **ksis list** command.

The import/export feature can be used to archive images that are no longer used on nodes, but that the administrator wants to keep.

4.9 Rebuilding ClusterDB Data before Deploying an Image

There are two cases where it may be necessary to update the reference information before deploying an image:

- Some values have changed in the ClusterDB
- An image has been imported so that its ClusterDB information must be updated.

To do so, use the **builddatanode** command, which updates the images with the latest values in the ClusterDB:

```
ksis builddatanode
```

```
Nodes context will be updated to take in account new data from DB

Continue (yes/no)
```

Answer **yes** to the question.

Chapter 5. Kerberos - Network Authentication Protocol

Kerberos is an optional security suite product that can be used to authenticate users, services and machines for a whole network. Kerberos is included within the Linux delivery.

The purpose of this chapter is to describe how to implement Kerberos on a Bull extreme computing cluster.

5.1 Environment

5.1.1 Kerberos Infrastructure

There are 3 types of machine within the **Kerberos** infrastructure:

- The **Kerberos** server that includes the Key Distribution Centre (**KDC**) server and administration server, housed on a server called **secu0**. By default, this will be part of the Management Node.
- A set of application servers (SSH, NFS, etc.) which are protected by Kerberos; these servers are named **secui**. The Kerberos configuration file for these servers is shared with the Kerberos server.
- The **Kerberos** client machines. These are not used until Kerberos authenticates the users' rights to access the applications on **secui**.

5.1.2 Authentication of the SSHv2 Connections

The remote **SSH** service (OpenSSH) will be activated on **secu1** with Kerberos support. A remote connection to **secu0** will then be made using Kerberos tickets instead of internal authentication mechanisms.

5.2 KERBEROS Infrastructure Configuration

5.2.1 secu0 Server including KDC Server and Administration Server

Verify the installation of the latest version of the Kerberos **RPM** on **secu0.**

For security reasons, the Kerberos package is compiled with the -without-krb4 option to prevent compatibility with Kerberos 4.

5.2.2 Configuration Files

/etc/krb5.conf

This file contains the details of the **KDC** addresses and the administration server, and will be copied on to all the servers containing kerberized applications, as well as on to all the client machines.

```
...
[libdefaults]
 default_realm = DOMAIN.COM
 default_tgs_enctypes = des3-hmac-sha1 des-cbc-crc des-cbc-md5
 default_tkt_enctypes = des3-hmac-sha1 des-cbc-crc des-cbc-md5
 permitted_enctypes = des3-hmac-sha1 des-cbc-crc des-cbc-md5
 forwardable = true
...

[realms]
 DOMAIN.COM = { kdc=secu0:88
                admin_server = secu0:749
                default.domain = domain.com
              }

[domain.realm]
 .domain.com = DOMAIN.COM
 domain.com = DOMAIN.COM
 localhost = DOMAIN.COM
...

[login]
 krb4_convert = false
 krb4_get_tickets = false
...
```

/var/kerberos/krb5kdc/kdc.conf

This file, containing among other things the information necessary to produce the tickets, is specific to the Kerberos server.

```
...
[realms]
DOMAIN.COM={
 preauth=yes
 admin_keytab = FILE:/etc/krb5.keytab
 max_life = 2d 0h 0m 0s
 max_renewable_life = 10d 0h 0m 0s
 ...
}
```

5.2.3 Creating the Kerberos Database

Use the following command to initialize the Kerberos database.

```
/usr/kerberos/sbin/kdb5_util create –s

enter KDC database master key : XXXX
```

5.2.4 Creating the Kerberos Administrator

The **KDC** server may be administered from any network machine using the command **kadmin** as long as the user's identity is authenticated.
As the Kerberos administrator node does not initially exist, it is possible to connect to the KDC server the first time as root using the **kadmin.local** command on the KDC server. It is not possible to authenticate oneself with this command as one is logged onto the KDC server.

```
/usr/kerberos/sbin/kadmin.local

kadmin.local : addprinc krb5adm/admin

Enter password for principal “krb5adm/admin@DOMAIN.COM”: YYYY
```

Now it should be possible to authenticate oneself as **krb5adm** from any Kerberos client machine. The **Unix** system account **krb5adm** must have been created, as shown above, in order to connect to the administrator server and to manage Kerberos, assuming the admin daemon has been launched. See below for more details.

Important

- **For security reasons remote administration using kadmin is deactivated. To enable it add the kadmin/admin and kadmin/changepw special entries in the keytabs. However, this setting is not recommended for a cluster environment.**
- **The Kerberos administrators which have been created – krb5adm in the example above - must belong to the root group in order to have access to, and to be able to modify, Kerberos files.**

5.2.5 Starting the KDC Server

Use the following command to start the KDC server:

```
/sbin/service krb5kdc start
```

Verify the local connection to Kerberos on the KDC server using the krb5adm administrator access rights:

```
/usr/kerberos/bin/kinit  krb5adm/admin
```

```
kinit(V5) : Cannot resolve network address for KDC in requested realm
while getting initial credentials
```

The problem in the above message is that krb5adm's credentials cannot be confirmed and will only be resolved when secu0 is replaced by its IP address in the krb5.conf file.

```
/usr/kerberos/bin/kinit  krb5adm/admin

Password for krb5adm@DOMAIN.COM: YYYY
```

If there is no error message then everything is OK and the krb5adm administrator will obtain a Ticket-Granting Ticket (TGT).

5.2.6 Adding Access Control List (ACL) Rights for the Kerberos Administrator Created

In the /var/kerberos/krb5kdc/kadm5.acl file, add the line:

```
krb5adm/admin  @DOMAIN.COM *
```

5.2.7 Starting the Administration Daemon

Use the following command to start the administration daemon.

```
/sbin/service  kadmin  start
```

It should now be possible to connect to the system and to administer the KDC server, with a view to specifying principals. A principal is an entity in the Kerberos realm – every user, instance and service in the Kerberos realm has a designated principal. The creation of principals has to be done from the Kerberos server using administrator access rights for krb5adm/admin.

5.2.8 Creating Principals Associated with Users

The Kerberos Administrator will create the principals associated with users on the KDC server. These users must have associated UNIX accounts on the client machines.

The Kerberos Administrator can create the principals locally on the **KDC** (using the command **kadmin.local**) without needing to authenticate himself. For example, for user durand:

```
kadmin.local
PW : YYYY
kadmin : addprinc durand
PW : ZZZZ  (add the user password on the client machines)
Principal " durand@DOMAIN.COM " created
```

The secret key shared between the **KDC** and the client machine for a user principal is derived from the user's password.

The process has to be repeated for all other users.

5.2.9 Creating Principals Associated with Remote Kerberized Services

The principals associated with services have to be created. The **Linux** distribution includes some services that have already been kerberized. The principal associated with **FTP**, **TELNET**, and **RSH** services, included as part of the default installation using the **krb5-workstation** package, is called **host principal**.

The **host principal** name is derived from the name of the machine, and this is used for **Kerberos** Authentication of the basic kerberized services - **RLOGIN**, **TELNET**, etc. residing on the host.

Creation of the host principal for the secu1 server

Connect to **Kerberos secu0** server and then create the host principal with the **kadmin** command.

```
kadmin.local
addprinc  -randkey   host/secu1.domain.com
```

Important

The hostname has to be the same as in its first appearance in the line associated with the machine in the /etc/hosts file.

5.3 Configuring the secu1 Machine that hosts the Host Principal remote service

Verify the installation of the latest version of the Kerberos RPMs on **secu1**.

Copy the configuration file **/etc/krb5.conf** from **secu0** to **secu1**, and to any other machines which may be part of the system.

5.3.1 Generating the key associated with the Host Principal remote service

This secret key is shared between the **KDC secu0** server and the server housing the **secu1** remote service. This is essential in order that **secu1** can decipher the **Kerberos** tickets which are transmitted to it. The key can be created on any one of these 2 servers but must then be copied from one to the other.

The default file for the keys is as follows:

```
/var/kerberos/krb5kdc/kadm5.keytab
```

Therefore, the file for the keys used by the command **kadmin** is defined in the realms section in the **kdc.conf** file:

```
/etc/krb5.keytab
```

Connect as the Kerberos administrator (**krb5adm**) to **secu0**:

```
kadmin
ktadd  -k /path/to/krb5.keytab.secu1 host/secu1.domain.com
```

Then recopy the **/path/to/krb5.keytab.secu1** key to **secu1** in the **/etc/krb5.keytab** file.

It is recommended to have a keytab file for each service, and to store only the keys associated with the remote services that each server hosts, and not the keys for services that are hosted by other servers. However, the KDC server must have its own specific keytab file for all the remote service keys.

5.4 Kerberos Authentication and SSH

The **SSH** remote service is installed on **secu1** with a SSH client connection from **secu0**.

Before using any Kerberized client, such as SSH, you have to request the **TGT** ticket. In the following example, this request is done for the user connected as `Durand` on **secu0**:

```
kinit
PW : xxxx  (password user durand)
klist
....
```

5.4.1 Configuring the SSH Server on the secu1machine

A typical **sshd_config** configuration file will contain the following:

```
Port 22
Protocol 2
ListenAddress xxx.xxx.xxx.xxx

RSAAuthentication no
PubkeyAuthentication no

RhostsRSAAuthentication no
HostbasedAuthentication no

PasswordAuthentication no
PermitEmptyPasswords no

# Kerberos options
KerberosAuthentication yes
# If the Kerberos authentication is denied, an Authentication password is
        not
# provided for the user :
KerberosOrLocalPasswd no
KerberosTicketCleanup yes

# GSSAPI options
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes

UsePAM yes

Subsystemsftp   /usr/local/libexec/sftp-server
```

Pre-requisites for the configuration of SSH server

- The **/etc/hosts** file of the remote machine that **SSH** is connecting to has to have its hostname in the form:

  `x.x.x.x  secu1.domain.com  secu1`

- The hostname of the remote machine may be of the form:

  `secu1.domain.com` or `secu1`.

- The principal service associated with this machine has to be the same as its Fully Qualified Domain Name **FQDN**:

  `secu1.domain.com`.

5.4.2 SSH Client

On the **secu0** machine, or other machines, a typical **ssh_config** file will appear as follows:

```
RhostsRSAAuthentication no
RSAAuthentication no
PasswordAuthentication no
HostbasedAuthentication no
Port 22
Protocol 2

GSSAPIAuthentication yes

# For tickets forwarding:
GSSAPIDelegateCredentials yes
```

Note **TGT** ticket forwarding by **SSH** is activated by the **GSSAPIDelegateCredentials yes** parameter in the **SSH** client file.

5.5 Troubleshooting Errors

```
Error : “ Permission denied (gssapi-with-mic,keyboard-interactive) ”
```

There are various possible causes for this error. Check the following:

1. The target machine has its **full name** in its **/etc/hosts** file as shown below:

```
@IP   secu1.domain.com secu1
```

2. If several names are associated with the same IP address, the name used for the connection has to be at the top of **/etc/hosts** file, as shown below:

```
@IP parallel.domain.com  parallel
@IP secu1.domain.com   secu1
```

3. Check that the **/etc/krb5.conf** file on the **KDC** server and on the **SSH** servers\clients is identical.

4. Check that the keys in the **/etc/krb5.keytab** file are identical on the **KDC** server and on the **SSH** server.

5. Verify that the user has a valid TGT ticket.

5.6 Generating Associated Keys for Nodes of a Cluster

The Perl program, below, generates the **Kerberos** key (keytab) for each node on the **Kerberos** server (hosted on the Management Node), and then transfers the key to the node using Secure Copy (**SCP**), which ensures confidentiality and authentication using a private key/public key.

The pre-requisite here is that the private key / private key infrastructure is in place between the Management Node and each Compute Node.

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

print "Lower limit of cluster nodes: ";
$inf = <STDIN>;
chomp ($inf);

print "Upper limit of cluster nodes: ";
$sup = <STDIN>;
chomp ($sup);

# Define constants
#
my $serv = "secu";
my $domain = "domain.com";
my $serv0 = "secu";
my $keytab = "_keytab";
my $krb5_keytab = "/etc/krb5.keytab";

# Key creation for each node of the cluster
# Each key is generated on the management node and is stored in a
# temporary # file (and also in the KDC base); this file will then be
# recopied on the associated node;
# The remote recopy by SCP will be secured by public/private keys.
for ($i=$inf; $i <=$sup; $i++) {
        $serv="$serv0$i";
        print("Generate keytab for host : $serv\n");
        system ("rm -f /tmp/$serv$keytab");
        system ("kadmin.local -q 'ktadd -k /tmp/$serv$keytab
         host/$serv.$domain'");
        system ("scp -rp /tmp/$serv$keytab $serv$krb5_keytab");
        system ("rm -f /tmp/$serv$keytab");
}


print("\n-----> The new keys for the nodes secu$inf to secu$sup have been
         generated \n\n");
```

5.7 Modifying the Lifespan and Renewal Period for TGT Tickets

The default duration for a Ticket-Granting Ticket (**TGT**) ticket is 10 hours, and this can be renewed while it is still active. In other words its duration must be greater than 0 to be renewed.

The ticket duration and renewal period can be modified by a user. For example, the command below is used to change the duration of a ticket to 2 days, and its renewal period to 5 days.

```
kinit  -l 2d  –r 5d
```

The ticket obtained using this command will be valid for 2 days and it may be renewed at any time during these 2 days to obtain a new ticket which is also valid for 2 days up until the 5 day limit is reached.

The values specified by the user have to be inside the maximum values defined by the Kerberos configuration. To modify the values in the Kerberos **/var/kerberos/krb5kdc/kdc.conf** configuration file do the following:

In the `[realms]` block, add:

```
max_life = 2d
max_renewable_life = 10d
```

Then relaunch the **krb5kdc** and **kadmin** daemons.

5.8 Including Addresses with Tickets

By default tickets do not include addresses.

Use the command below so that the tickets generated include the addresses of the local machine.

```
add noaddresses=no in the paragraph [libdefaults] for the file
/etc/krb5.conf
```

Chapter 6. Storage Device Management

Bull cluster management tools provide services to manage storage systems and a large amount of storage resources. This chapter explains how to setup the management environment, and how to use storage management services.

The following topics are described:

- 6.1 *Overview of Storage Device Management for Bull extreme computing clusters*
- 6.2 *Monitoring Node I/O Status*
- 6.3 *Monitoring Storage Devices*
- 6.4 *Monitoring Brocade Switch Status*
- 6.5 *Managing Storage Devices with Bull CLI*
- 6.6 *Using Management Tools*
- 6.7 *Configuring Storage Devices*
- 6.8 *User Rights and Security Levels for the Storage Commands*

6.1 Overview of Storage Device Management for Bull extreme computing clusters

Bull extreme computing clusters can contain various kinds of storage devices. Thus, storage device management may quickly become a complex task, due to the variety and the number of management interfaces.

Using Bull storage management services the cluster administrator will be able to:

- Monitor the status of storage devices
- Monitor storage within cluster nodes
- Get information about faulty components
- Get synthetic reports for the storage resources
- Automate the deployment of storage device configurations
- Ensure consistency between storage systems and I/O nodes
- Configure individual storage devices using a command line interface from the cluster management station
- Obtain access to the management tools for each storage device, regardless of its user interface.

Bull extreme computing clusters are deployed with both a specific hardware infrastructure, and with software packages, to simplify and unify these management tasks.

The hardware infrastructure enables the management of all the storage devices from the cluster Management Nodes, and includes:

- Built-in LAN management ports for the storage devices that are connected to the cluster management network.
- Built-in serial ports for the storage devices that are connected to the cluster management network, using terminal servers.
- Management stations or proxy servers (for example Windows stations) hosting device management tools that are connected to the cluster management network, or are reachable from the Management Nodes.

The software packages installed on the cluster Management Node and on other cluster nodes provide various device management services:

- **Device monitoring**
  A device inventory is performed and detailed descriptions of all the storage devices are stored in the cluster data base. The storage devices are monitored by the cluster Management Node, using standardized protocols such as **SNMP**, **syslog**, or proprietary interfaces. The Management Node waits for event notification from the devices. To prevent silent failures, forced updates are scheduled by the Management Node. All the events are automatically analyzed and the cluster DB is updated to reflect status changes. The storage device status can be monitored using **Bull System Manager – HPC Edition** and by querying the cluster DB with the **storstat** command. These services enable the browsing via a global view covering all the storage devices,and a more detailed view focusing on a single storage device.

- **Advanced device management**.
  Administrators trained to manage the storage devices, and familiar with the terminology and operations applicable to each kind of storage device, can use the command line interfaces available on the cluster Management Node. These commands are specific to a storage system family (for example **nec_admin**, etc.). They enable configuration and status information to be read, and also configuration tasks to be performed. The syntax and output are as close as is possible to the information provided by the device management tools included with the storage system. The most useful information and operations are available via these commands. Nevertheless, they do not include all the management services for each device. Their advantage is that they provide a command line interface on the cluster Management Node They can also be used to build custom tasks, by parsing command outputs or creating batches of commands.

WARNING
Changing the configuration of a storage device may affect all the cluster nodes using this device.

- **Access to management tools**.
  The storage administrator who is trained to manage storage devices can also access the management tools for each storage device. The serial ports can be used with **conman** (or telnet). The Ethernet ports can be connected to via telnet or a web browser. Management software on proxy UNIX servers can be used with **ssh** (command mode) or X11 (graphical applications). Similarly, an ssh service and a **VNC** server are provided for Windows, in order to enable access to the management software on proxy Windows servers, either in command mode or in graphical mode.
- **Storage device configuration deployment.**
  For small clusters, the administrator can use either the device specific commands installed on the cluster Management Node, or the tools for each storage device. For medium to large clusters, there are often lots of storage systems with the same hardware and logical configurations. For these kinds of complex environments, configuration deployment services are provided.

  These services are only available in command mode.

WARNING
System Administrators must be trained to manage the storage devices, and be familiar with the terminology and operations applicable to each kind of storage device. They must be aware of the impact of updating a storage device configuration.

The following sections explain how to setup and use this environment.

6.2 Monitoring Node I/O Status

Each node is monitored and any I/O errors reported in **syslog** are tracked. A global I/O status is computed locally on each node and is reported to the Management Node using dedicated **syslog** messages.

The I/O status of each node can be verified by displaying the **I/O status** service of the node via **Bull System Manager – HPC Edition**.

Figure 6-1. I/O Status – initial screen

The semantic of the service is as follows:

OK	No problem detected
WARNING	An I/O component in WARNING state is in an unstable state but the resource is still available. It may also indicate that the current number of I/O components is higher than its expected reference number.
CRITICAL	Degraded service. Maintenance operation mandatory Criteria: Hereafter is a list of possible critical errors: A fatal disk error has been reported by the Linux I/O stack in **syslog** A fatal **HBA** error has been reported by a device driver in **syslog** A link down transition has been notified by a device driver A **LUN** resource cannot be acceded by a multipath pseudo-device. A device referenced by the persistent binding mechanism (alias) is missing.
UNKNOWN	Cannot access the status
PENDING	Not yet initialized

6.2.1 Managing I/O Reference Counters

The I/O status transmitted by each node to the Management Node is a synthesis of the status detected for each I/O resource attached to the node and of the comparison between the I/O counters and their reference values.

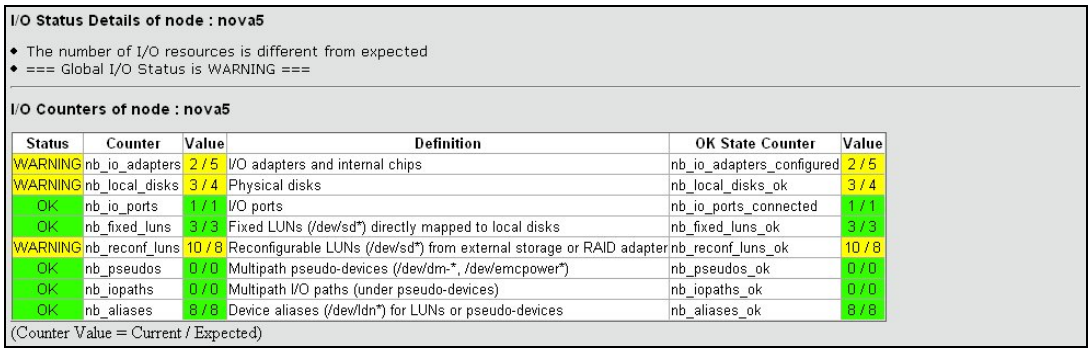
The I/O status monitoring service builds a reference during its initial startup, usually at the first boot of the node.

The reference contains the expected number of various device classes (named 'I/O counters').

Two reference counters (**nb_io_adapters** and **nb_local_disks**) are stored on the Management Node in cluster DB in the **node_profile** table. The other reference counters are stored on the local node.

At boot time the **nb_io_adapters** and **nb_local_disks** counters are automatically adjusted from the cluster DB node I/O profile.

You can view details of I/O status reference counter values for each node by the **I/O status details** link of the **I/O status** service on the node via **Bull System Manager – HPC Edition**.



I/O Status Details of node : nova5

- The number of I/O resources is different from expected
- === Global I/O Status is WARNING ===

I/O Counters of node : nova5

Status	Counter	Value	Definition	OK State Counter	Value
WARNING	nb_io_adapters	2 / 5	I/O adapters and internal chips	nb_io_adapters_configured	2 / 5
WARNING	nb_local_disks	3 / 4	Physical disks	nb_local_disks_ok	3 / 4
OK	nb_io_ports	1 / 1	I/O ports	nb_io_ports_connected	1 / 1
OK	nb_fixed_luns	3 / 3	Fixed LUNs (/dev/sd*) directly mapped to local disks	nb_fixed_luns_ok	3 / 3
WARNING	nb_reconf_luns	10 / 8	Reconfigurable LUNs (/dev/sd*) from external storage or RAID adapter	nb_reconf_luns_ok	10 / 8
OK	nb_pseudos	0 / 0	Multipath pseudo-devices (/dev/dm-*, /dev/emcpower*)	nb_pseudos_ok	0 / 0
OK	nb_iopaths	0 / 0	Multipath I/O paths (under pseudo-devices)	nb_iopaths_ok	0 / 0
OK	nb_aliases	8 / 8	Device aliases (/dev/ldn*) for LUNs or pseudo-devices	nb_aliases_ok	8 / 8

(Counter Value = Current / Expected)

Figure 6-2. Bull System Manager HPC Edition - I/O Status Details

The **iorefmgmt** command is used to manage I/O device monitoring reference counters.

To obtain the list of the reference counter enter:

```
iorefmgmt -g
```

Use the help or the man page to obtain a description of the counters used, alternatively see the definitions in the section below.

If the reference is wrong, it can be updated as follows:

```
iorefmgmt -s -n <counter_name> -v <value>
```

You can adjust reference counters to the current discovery value using the command:

```
iorefmgmt –c adjust
```

The **nb_io_adapters** and **nb_local_disks** counters cannot be adjusted on a node.

You can manage these counters in the cluster DB node profile table on the Management Node by using the command:

```
iorefmgmt –c dbset|dbget|dbdel
```

For more information use the **iorefmgmt** man page or help.

All these operations can be done from the Management Node, using **ssh** or **pdsh**.

6.2.2 I/O Counters Definitions

nb_io_adapters Expected number of I/O adapters on the node (a multi-port adapter is counted as 1, an internal I/O chip is also counted as one adapter).

nb_io_adapters_configured
Number of I/O adapters expected to be configured (driver loaded).

nb_local_disks Expected number of physical disks on a node.
A physical disk may be:
- an internal disk which is directly attached,
- a physical disk in a SCSI JBOD,
- a physical disk behind a RAID controller.

nb_local_disks_ok Number of physical disks expected to be healthy.

nb_io_ports Expected number of Fibre Channel ports.

nb_io_ports_connected
Number of Fibre Channel ports expected to be connected.

nb_fixed_luns Expected number of LUNs which are not reconfigurable.
A LUN which is not reconfigurable is directly mapped to a physical disk.

nb_fixed_luns_ok Number of LUNs which are not reconfigurable that are expected to be accessible.

nb_reconf_luns Expected number of reconfigurable LUNs.

nb_reconf_luns_ok Number of reconfigurable LUNs expected to be accessible.
A "reconfigurable LUN" is typically a LUN in an external storage system (usually a RAID system) or a LUN presented by a RAID HBA, on top of RAIDed local disks.

nb_iopaths Expected number of paths involved in multi-path to reach LUNs which are reconfigurable.

nb_iopaths_ok Number of paths involved in multipath expected to be alive.

nb_aliases Expected number of aliases on Fibre Channel block devices.
Aliases are used to obtain a persistent device naming scheme, regardless of the order that the FC devices are detected.

nb_aliases_ok Number of aliases on Fibre Channel block devices expected to be correctly mapped.

nb_pseudos Expected number of multipath pseudo-devices on a node.

nb_pseudos_ok Number of multipath pseudo-devices expected to be usable.

6.2.3 Managing I/O Resources

The I/O resources identified for each node are monitored and their status stored on the node. The I/O resources may be displayed in **Bull System Manager – HPC Edition** by looking at the **I/O status** service associated with a node; this is done by clicking on the **I/O status details** link.

I/O Resources of node : nova5

Adapter 03:01 LSI LSI LSI53C1030 Driver: mptspi CONFIGURED	**host0**	**sdb** (0:0:10:0) **OK** (Fixed LUN)	Physical Disk **sdb** **OK** SEAGATE SPI 286102MB
		sdc (0:0:11:0) **OK** (Fixed LUN)	Physical Disk **sdc** **OK** SEAGATE SPI 286102MB
		sda (0:0:9:0) **OK** (Fixed LUN)	Physical Disk **sda** **OK** SEAGATE SPI 286102MB
	host1		
Adapter 2d:01 Emulex LP11000 Driver: lpfc CONFIGURED	**host2** (Port) WWN: 10:00:00:00:c9:4b:c0:9a CONNECTED	**sdd** (2:0:0:0) **OK** DDN 1000MB (Reconfigurable LUN, FC)	← Alias **ldn.ddn0.24**
		sde (2:0:0:1) **OK** DDN 1048576MB (Reconfigurable LUN, FC)	← Alias **ldn.ddn0.25**
		sdl (2:0:0:12) **OK** DDN 49896MB (Reconfigurable LUN, FC)	
		sdm (2:0:0:13) **OK** DDN 49896MB (Reconfigurable LUN, FC)	
		sdf (2:0:0:2) **OK** DDN 1000MB (Reconfigurable LUN, FC)	← Alias **ldn.ddn0.26**
		sdg (2:0:0:3) **OK** DDN 1048576MB (Reconfigurable LUN, FC)	← Alias **ldn.ddn0.27**
		sdh (2:0:0:4) **OK** DDN 1000MB (Reconfigurable LUN, FC)	← Alias **ldn.ddn0.28**
		sdi (2:0:0:5) **OK** DDN 1048576MB (Reconfigurable LUN, FC)	← Alias **ldn.ddn0.29**
		sdj (2:0:0:6) **OK** DDN 1000MB (Reconfigurable LUN, FC)	← Alias **ldn.ddn0.30**
		sdk (2:0:0:7) **OK** DDN 1048576MB (Reconfigurable LUN, FC)	← Alias **ldn.ddn0.31**
ldn.ddn0.24 Alias **OK** linked to **sdd**			
ldn.ddn0.25 Alias **OK** linked to **sde**			
ldn.ddn0.26 Alias **OK** linked to **sdf**			
ldn.ddn0.27 Alias **OK** linked to **sdg**			
ldn.ddn0.28 Alias **OK** linked to **sdh**			
ldn.ddn0.29 Alias **OK** linked to **sdi**			
ldn.ddn0.30 Alias **OK** linked to **sdj**			
ldn.ddn0.31 Alias **OK** linked to **sdk**			

Figure 6-3. Bull System Manager –HPC Edition – I/O Resources of a node

The list of I/O resources, with their associated status, for each node can also be consulted by using the following command:

```
lsiodev -l
```

On the Management Mode, the equivalent information can be obtained remotely from the nodes by using the following command:

```
iorefmgmt -r <node> -L
```

The current status for each I/O resource is updated automatically by the I/O monitoring mechanism. However, it may be necessary to update the status of a resource manually, for example, to distinguish transient from permanent I/O errors. The status of an I/O resource can be changed remotely from the Management Node by using the following command:

```
iorefmgmt -r <node> -U -t <resource-type> -n <resource-name> -s <new-
status> -m "<associated-comment>"
```

Using this command will result in the global I/O status of the node being recomputed and its **I/O status** service in **Bull System Manager – HPC Edition** being updated.

6.3 Monitoring Storage Devices

This section explains how the Administrator can monitor and obtain information about all the managed storage systems of the cluster, using a unified interface. The two following interfaces are available for the administrator:

- Graphical User Interface (Bull System Manager – HPC Edition):
  - Hosts and service monitoring for storage devices.
  - Storage views, providing detailed information regarding the storage systems.
- Command line interface:
  - **storstat** command, to query the ClusterDB for storage information.
  - Archiving of **syslog** messages.

Note The monitoring relies on information stored in the **ClusterDB**. This information is updated periodically, and also when failures or repairs are notified by storage devices. The monitoring information is therefore not updated in real-time when silent state changes occur, such as repairs.

The Administrators can force a refresh of the Database information using the **storcheck** command:

```
storcheck -c <cluster_name>
```

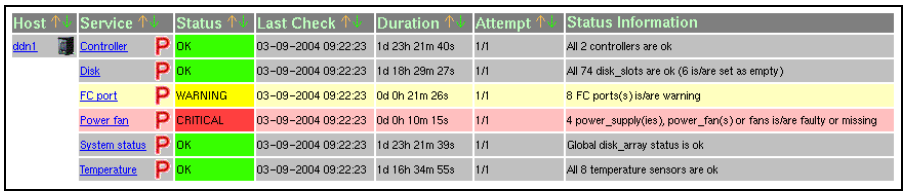
This command will check all the storage systems for a cluster. It is possible to reduce the scope to a single storage system:

```
storcheck -c <cluster_name> -n <disk_array_name>
```

6.3.1 Bull System Manager - HPC Edition: Host and Service Monitoring for Storage Devices

Storage device monitoring is integrated in the global monitoring for a cluster. Each storage system is identified by a host and its associated service, regardless of the number of controllers and Ethernet ports.

Bull System Manager - HPC Edition continuously updates the host status and service status values, without any intervention from the Administrator. All **Bull System Manager - HPC Edition** features and services apply to storage devices. Nevertheless, the Administrator using **Bull System Manager - HPC Edition** must be aware of the specific points that are explained next.



Host	Service	Status	Last Check	Duration	Attempt	Status Information
ddn1	Controller	OK	03-09-2004 09:22:23	1d 23h 21m 40s	1/1	All 2 controllers are ok
	Disk	OK	03-09-2004 09:22:23	1d 18h 29m 27s	1/1	All 74 disk_slots are ok (6 is/are set as empty)
	FC port	WARNING	03-09-2004 09:22:23	0d 0h 21m 26s	1/1	8 FC ports(s) is/are warning
	Power fan	CRITICAL	03-09-2004 09:22:23	0d 0h 10m 15s	1/1	4 power_supply(ies), power_fan(s) or fans is/are faulty or missing
	System status	OK	03-09-2004 09:22:23	1d 23h 21m 39s	1/1	Global disk_array status is ok
	Temperature	OK	03-09-2004 09:22:23	1d 16h 34m 55s	1/1	All 8 temperature sensors are ok

Figure 6-4. Detailed service status for a storage host

The host and service monitoring offers uniform monitoring for all the cluster components, with history and statistical capabilities. It provides for each storage system a general view of the major functional domains.

However, this monitoring does not allow the easy identification of storage devices among other cluster components nor individual faulty hardware components to be identified. These limitations are compensated by the use of Storage Views (see 6.3.2 *Bull System Manager - HPC Edition: Storage & I/O Information*).

6.3.1.1 Host Semantic

The host name is a logical name, which uniquely identifies a storage system. But caution, it is not bound to an IP address; it is not possible to ping using this parameter.

The host status indicates whether the storage system is manageable or not:

UP	The storage system responds through the management interfaces
UNREACHABLE	Some network problems prevent the management interface from being reached.
DOWN	The management interfaces of the storage system do not answer to requests. But note that from a storage point of view, the storage system may process I/O requests from attached hosts.

6.3.1.2 Service Semantics

Several generic services are defined for storage systems. They reflect the global status of a class of components in the selected storage system:

- Disk
- Power-Fan
- Temperature
- Controller
- FC ports
- System status.

Disk Service

This service describes the global status for the **HDD**s. It monitors both disk failures and if any disks have been removed (for example for maintenance purpose).

OK	**No problem** Criteria: • No disk errors • All referenced disks are present
WARNING	**Maintenance operation must be scheduled** Criteria: • Some disk failures, and / or removed referenced disks • Does not meet the criteria for critical status.
CRITICAL	**Degraded service. Maintenance operation mandatory** Criteria: • The number of faulty / missing disks is higher than the number of spare disks.
UNKNOWN	**Cannot access the status**
PENDING	**Not yet initialized**

Note The cluster database has been initialized with a detailed status including all disk slots, populated and empty. The Administrator, who decides to permanently remove some HDDs, must manually update the database reference configuration (using the **storstat –u** command). Otherwise, these empty slots will lead to a permanent WARNING status.

Power-Fan Service

Describes the global status for the power supply and fan modules. These two kinds of hardware parts are grouped and monitored using a single service.

OK	**No problem** Criteria: • All power supplies and fans are OK • All reference power supplies and fans are present
WARNING	**Maintenance operation must be scheduled** Criteria: • Some power supplies and/or fans are in the warning or critical state • Does not meet the criteria for critical status.
CRITICAL	**Degraded service. Maintenance operation mandatory** Criteria: • The percentage of faulty/missing power supplies or fans objects has reached the threshold defined in **/etc/storageadmin/storframework.conf** (**service_power_fan_critical_threshold** parameter).
UNKNOWN	**Cannot access the status**
PENDING	**Not yet initialized**

Temperature Service

Describes the global status for temperature sensors.

OK	**No problem** Criteria • All temperature sensors are OK
WARNING	**Maintenance operation must be scheduled** Criteria: • Some temperature sensors are not in the OK state • Critical criteria not met
CRITICAL	**Degraded service. Maintenance operation mandatory** Criteria: • Some temperature sensors are in the critical state.
UNKNOWN	**Cannot access the status**
PENDING	**Not yet initialized**

Controller Service

This service shows the controller status. The controller refers to the storage system elements in charge of host connection and I/O processing.

OK	**No problem** Criteria: • All controllers are OK
WARNING	**Maintenance operation must be scheduled** Criteria: • Some controllers have a warning state and none are faulty (or missing).
CRITICAL	**Degraded service. Maintenance operation mandatory** Criteria: • One controller or more is faulty (or missing).
UNKNOWN	**Cannot access the status**
PENDING	**Not yet initialized**

Fibre Channel Port Service

This service shows the host connectivity status:

OK	**No problem** Criteria: • All FC ports are OK.
WARNING	**Maintenance operation must be scheduled** Criteria: • Not in critical status • Some ports have a warning status
CRITICAL	**Degraded service. Maintenance operation mandatory** Criteria: • One or more ports are in a critical status.
UNKNOWN	**Cannot access the status**
PENDING	**Not yet initialized**

Note If the FC link is connected to a switch, and the link is broken 'after' the switch and not between the controller and the switch, the failure is not detected by the disk array and therefore will not be displayed by the FC port service.

System Status Service

This service is a collector and gathers together all the problems for the storage system. If one of the services described above is warning or critical, the system status service will be critical. This service also reflects the other problems which may arise, but are not classified, in one of the previously defined services. For example, all the other services may be OK, while the system status is warning or critical.

OK	**No problem** Criteria: • Disk array semantic.
WARNING	**Maintenance operation must be scheduled** Criteria: • Some of the other services are warning (but none critical). • The storage system has detected a warning which is not reported by one of the other services.
CRITICAL	**Degraded service. Maintenance operation mandatory** Criteria: • One of the other services is critical. • The storage system has detected a critical error which is not reported by one of the other services.
UNKNOWN	**Cannot access the status**
PENDING	**No yet initialized**

6.3.2 Bull System Manager - HPC Edition: Storage & I/O Information

Bull System Manager – HPC Edition contains specific views, which focus on the monitoring of storage devices and I/O systems for the nodes connected to these devices. It enables administrators to pinpoint faulty hardware components, and provides detailed reporting and configuration information for storage systems.

The Storage and I/O information view is selected by clicking on the **Storage overview** icon on left hand side of the **Bull System Manager – HPC Edition** console – see Figure 6-5. A pop-up window appears containing a summary view of the storage systems and hardware component status – see Figure 6-6.

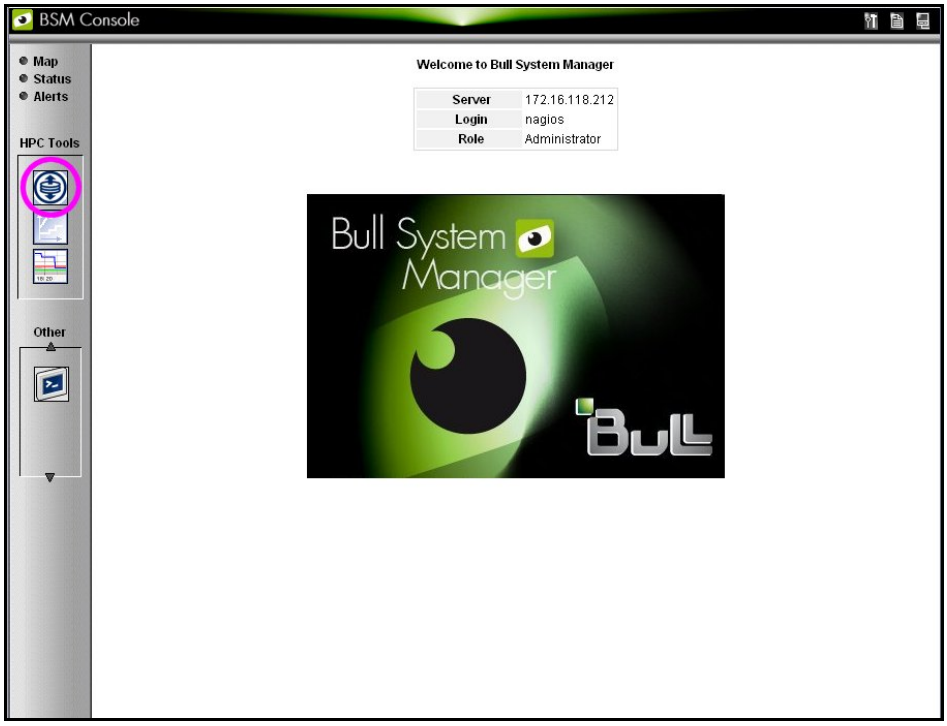


Figure 6-5. Bull System Manager opening console window with the Storage overview icon circled

6.3.2.1 Storage Views

Storage views provide information about:

- **Disk arrays**.
  Their status refers to the last known operational status of the storage system under review. It is similar to the 'system status' service in **Bull System Manager** host and service views. For example a storage system that does not answer to management requests is considered as faulty.
- **Individual hardware components**
  (Disk, FC port, RAID controller, and so on).
  There is no equivalent in the host and service monitoring services that provides a single service for all the disks of a storage system.

Note The disk array status is a superset of the individual hardware components status. It is usually managed by the disk array and is not limited to the hardware components managed by storage views. Therefore the disk array status may be more severe than the worst status of the individual hardware components.

The status used in the storage views are the following ones:

OK	No problem
ATTENTION	Maintenance operation must be scheduled, but the system still delivers the expected service.
FAILED	Degraded service. Maintenance operation mandatory.

6.3.2.2 Storage Overview

This view offers a synthesis of the Storage devices monitored in the cluster.

Figure 6-6. Storage overview

Functional Summary

This diagram refers to storage systems. It sorts the storage systems according to their operational status and to their respective roles.

Hardware Summary

This diagram provides statistics on low level hardware components such as HDDs, Fibre Channel ports, RAID controllers, power supplies, FANs, etc. The diagram is displayed by family of components, sorted by state.
The Administrator clicks the ATTENTION and FAILED percentages links in the Storage overview pop-up window to get an inventory of storage systems or hardware components in the selected state – see Figure 6-7.

6.3.2.3 Inventory View of Storage Systems and Components requiring attention

This view - Figure 6-7 - displays the list of faulty components that should either be examined or replaced. The components are grouped by storage system. For each component, the view displays:

- The description of the component
- Its status
- Location information for the component, within the device and within the cluster, its rack level and label.

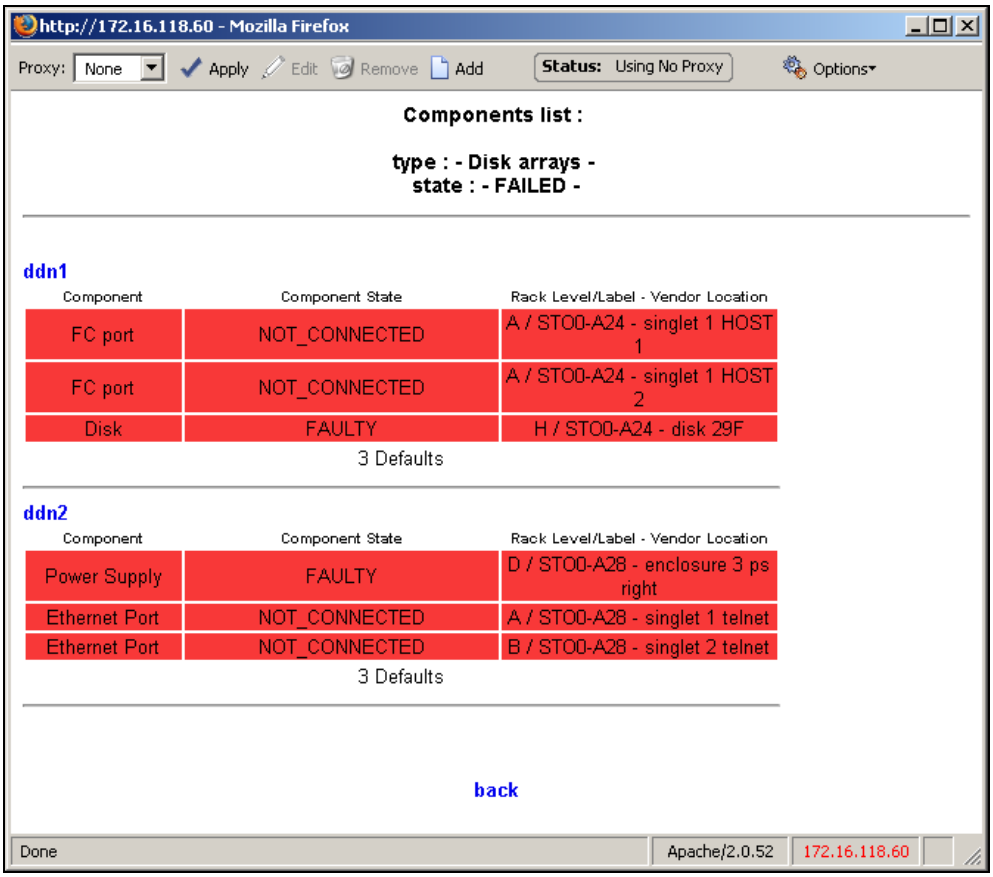


Figure 6-7. Inventory view of faulty storage systems and components

Note The hardware components whose status is OK are not listed.

This view is useful for planning maintenance operations for the components that are to be examined or replaced.

6.3.2.4 Detailed View of a Storage System

The Storage detailed view - Figure 6-8 - can be displayed by selecting a storage system in the Storage Summary Overview (see Figure 6-6).

This view provides detailed information for the selected storage system:

- Technical information (disk array status, firmware version, addressing information for management purposes, etc.).
- Front and rear diagram view, where the status of all the hardware components is represented by a color code.
- I/O cell and I/O path information:
  - An I/O cell is a set of nodes and storage systems functionally tied together.
  - An I/O path is a logical path between a node and the host port of a storage system. When a point-to-point connection is used, the I/O path is physically represented by a cable. In SAN environment, the I/O path represents both the I/O initiator (the node) and I/O target (the host port of the storage system).
- **Error List** hyperlink (list of faulty components).
- **Lun / Tier / Zoning List** hyperlink (information about the logical configuration of the storage system).

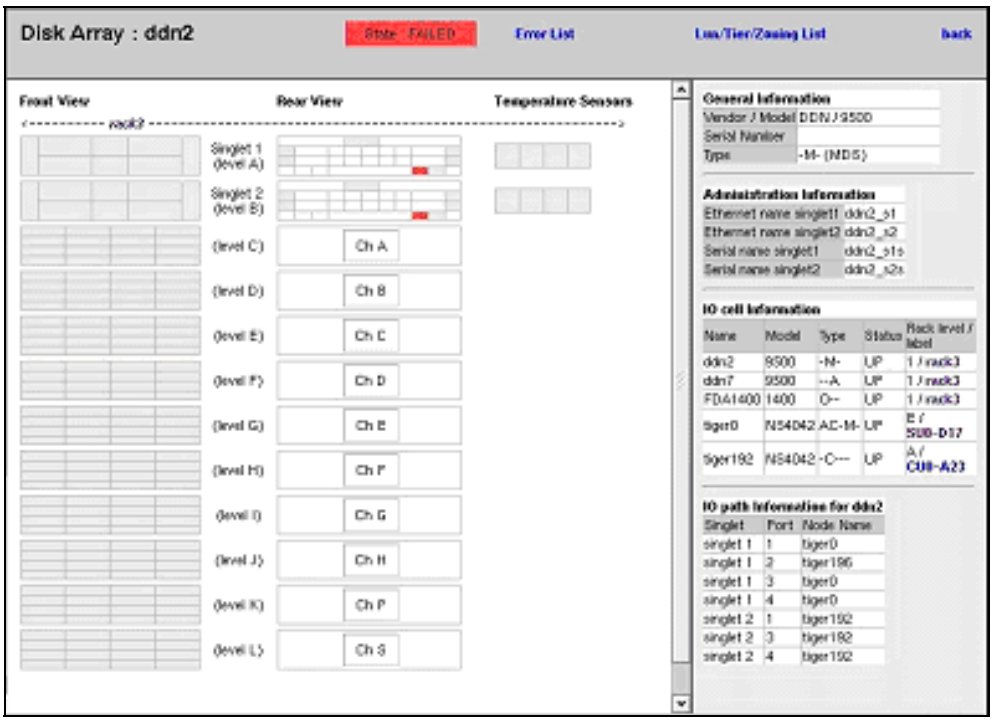


Figure 6-8. Storage detailed view

In the Storage Detailed view the item's description is shown by the use of mouse Tool tips.

6.3.2.5 Nodes I/O Overview

This view – Figure 6-9 – provides a synthesis of the I/O information for the cluster nodes.

It shows I/O status statistics and allows the list of nodes to be filtered for a selected I/O status value.

Clicking on the I/O status value of a node allows detailed information about the I/O resources of the node, and its associated I/O counters, to be displayed.

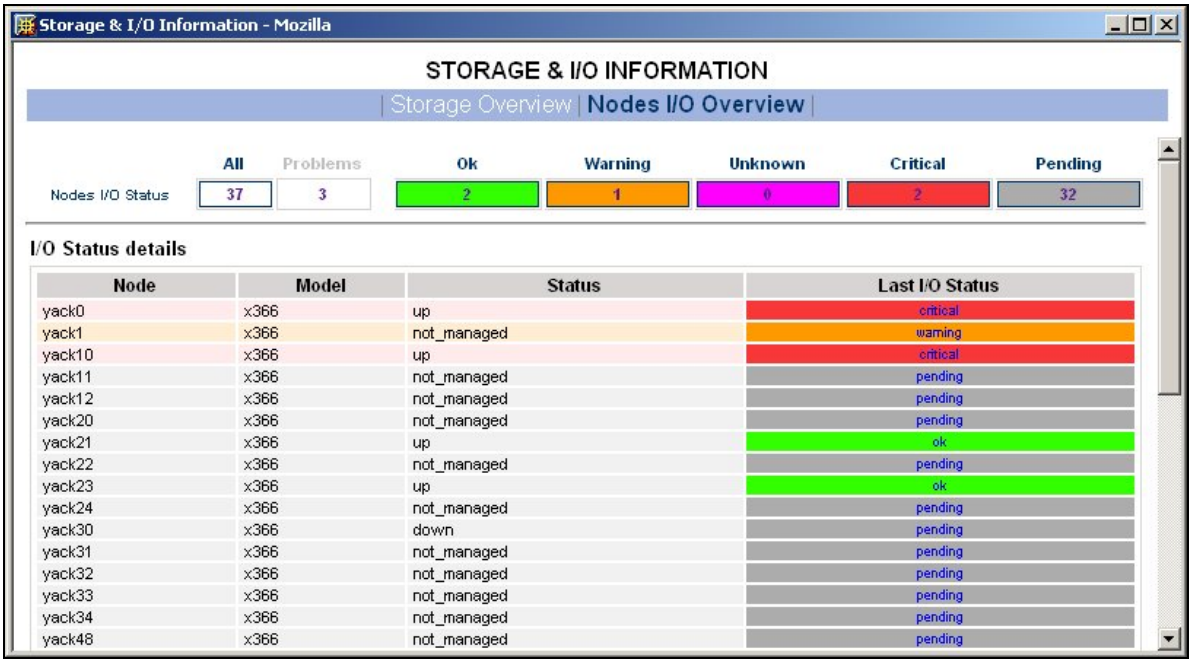


Figure 6-9. Nodes I/O Overview

6.3.3 Querying the Cluster Management Data Base

The **storstat** command obtains status information from the **ClusterDB** and formats the results for storage administrators.

See the help page for this command for more information:

```
storstat -h
```

The following paragraphs describe the most useful options.

6.3.3.1 Checking Storage System Status

Use the command below to display all the registered storage systems with their status and location in the cluster. The location is based on rack label and position in the rack:

```
storstat -a
```

To display a list of faulty storage systems:

```
storstat –a -f
```

To check the status of a storage system using the name identifying the storage system:

```
storstat -a -n <disk_array_name> -H
```

6.3.3.2 Checking Status of Hardware Elements

To display a list of faulty components for all the registered storage systems:

```
storstat -d –f -H
```

For each element, the following information is displayed:

- Disk array name
- Enclosure of the disk array housing the component
- Type of the component
- Status of the component
- Location of the component within the enclosure or disk array. This location uses vendor specific terminology
- Location of the enclosure (or disk array) in the cluster.

The **–n <disk_array_name>** flag can be used to restrict the list to a single storage system.

To display a list for all the storage system components:

```
storstat -d -n <disk_array_name>
```

Note If the **–n** flag is omitted the list will include all the registered storage systems.

To check the number of available or faulty elements in the cluster (or in a selected storage system):

```
storstat -c
```

or

```
storstat -c -n <disk_array_name>
```

6.4 Monitoring Brocade Switch Status

Each Brocade Fibre Channel switch is monitored by **Bull System Manager - HPC Edition**.

The same check period as for Ethernet switches will be used (10 minutes, this is configurable). No specific configuration is required for the FC switches in order to be able to use the **telnet** interface.

Several generic services are defined for Brocade switches. They reflect the global status of a class of components for the selected switch. A mapping between the SNMP MIB (Management Information Base) values available and returned from the switch, and the **Bull System Manager - HPC Edition** status give the following set of states for each managed services:

Ethernet interface Service

OK	**No problem** Criteria: • The Fping of the Ethernet interface is OK
CRITICAL	Criteria: • The Fping of the Ethernet interface is KO

FC port

OK	**No problem** Criteria: • All FC ports are OK.
WARNING	**Maintenance operation must be scheduled** Criteria: • Not in critical status • Some ports have a warning status • Number of operating port higher than expected in the DB (fc_switch.oper_port_threshold)
CRITICAL	**Degraded service. Maintenance operation mandatory** Criteria: • One or more ports are in a critical status. • Number of operating ports lower than expected (fc_switch.oper_port_threshold)
UNKNOWN	**Cannot access the status**
PENDING	**Not yet initialized**

Fans

OK	**No problem** Criteria: • All fans are present and OK
WARNING	**Maintenance operation must be scheduled** Criteria: • Some fans are in the warning state • Does not meet the criteria for critical status.
CRITICAL	**Degraded service. Maintenance operation mandatory** Criteria: • At least one of the fan is in a critical state
UNKNOWN	**Cannot access the status**
PENDING	**Not yet initialized**

Power Supply

OK	**No problem** Criteria: • All power supplies are present and ok • No Power Supply is detected on the switch.
WARNING	**Maintenance operation must be scheduled** Criteria: • Some power supplies are in the warning state • Does not meet the criteria for critical status.
CRITICAL	**Degraded service. Maintenance operation mandatory** Criteria: • At least one of the power supplies is in a critical state
UNKNOWN	**Cannot access the status**
PENDING	**Not yet initialized**

Temperature Sensor

OK	**No problem** Criteria: • All Temperature sensor are present and OK
WARNING	**Maintenance operation must be scheduled** Criteria: • Some Temperature sensor are in the warning state • Does not meet the criteria for critical status.
CRITICAL	**Degraded service. Maintenance operation mandatory** Criteria: • At least one of the Temperature Sensor is in a critical state
UNKNOWN	**Cannot access the status**
PENDING	**Not yet initialized**

Global Status

OK	**No problem** Criteria: • Global Brocade switch status is OK.
WARNING	**Maintenance operation must be scheduled** Criteria: • Some of the other services are warning (but none critical). • Switch name (switchX) different as expected (fcswX)
CRITICAL	**Degraded service. Maintenance operation mandatory** Criteria: • One of the other services is critical. • The storage system has detected a critical error which has not been reported by one of the other services.
UNKNOWN	**Cannot access the status**
PENDING	**No yet initialized**

The different services managed by **Bull System Manager - HPC Edition** for the Brocade switch are shown below:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
fcswu0c0	Ethernet Interfaces	OK	28-02-2006 11:01:15	0d 3h 4m 35s	1/1	down : [ ] - up : [10.0.0.90]
	FC ports	CRITICAL	28-02-2006 11:01:15	0d 3h 4m 35s	1/1	8 FC ports - OK [0 1 3 4 5 6 7] - WARNING [2]. Number of operating ports (2) lower than expected (4)
	Fans	OK	28-02-2006 11:01:15	0d 3h 4m 35s	1/1	All 3 Fans are OK
	Power supply	OK	28-02-2006 10:56:50	0d 18h 5m 23s	1/1	All 1 Power Supplies are OK
	Status	CRITICAL	28-02-2006 11:01:15	0d 0h 13m 20s	1/1	Global switch status is CRITICAL
	Temperature	OK	28-02-2006 11:01:15	0d 3h 4m 35s	1/1	All 4 Temperature Sensors are OK

Figure 6-10. Detailed Service status of a brocade switch

6.5 Managing Storage Devices with Bull CLI

This section describes the commands available for each device family.

These commands offer the most useful subset of management features, implemented for each storage system.

For storage systems not listed in the next paragraph the administration will be done via the tools delivered with the Storage System.

6.5.1 Bull FDA Storage Systems

The Administrator must be familiar with the **FDA** terminology and management tasks.

See The Bull FDA documentation for the **StoreWay FDA** model for more information on the options, parameters and possible values.

The **nec_admin** command usually requires at least two input parameters:

- The IP address (or host name) of the Windows system which hosts the FDA Storage Manager for the target FDA system.
- The name of the target FDA system.

The following services are provided by the command:

- **rankbind**
- **ldbind**
- **addldset**
- **addldsetld**
- **sparebind**
- **sparerebuild**
- **delldset**
- **ldunbind**
- **rankunbind**
- **spareunbind**
- **unconfig**
- **getstatus**
- **direct**

All the FDA arrays are supposed to be manageable using a single login/password. The **nec_admin** command enforces the parameters defined in the **/etc/storageadmin/nec_admin.conf** file as follows:

```
# NEC CLI Command path

# On Linux iSMpath="/opt/iSMSMC/bin/iSMcmd"
# On Windows iSMpath="/cygdrive/c/Program
\Files/FDA/iSMSM_CMD/bin/iSMcmd"
iSMpath = /opt/iSMSMC/bin/iSMcmd
#iSMpath="/cygdrive/c/Program\ Files/FDA/iSMSM_CMD/bin/iSMcmd"

# NEC iStorage Manager host Administrator
```

```
hostadm = administrator
# NEC iStorage Manager administrator login
necadmin = admin
# NEC iStorage Manager administrator password
necpasswd = adminpassword
```

For more information, read the man page or check the command's help.

6.5.2 DataDirect Networks Systems - DDN Commands

The administrator must be familiar with the DDN S2A terminology and management tasks. If necessary the administrator must refer to the documentation provided with S2A storage systems in order to understand the options, parameters and possible values.

The DDN specific commands usually require at least one input parameter:

- The IP address (or host name) of the target singlet for the command.

6.5.2.1 ddn_admin

This command allows you to get information from a singlet, or to configure the singlet. The following services are provided by the **ddn_admin** command:

- **deletelun**
- **formatlun**
- **getinfo**
- **getfmtstatus**
- **getstatus**
- **setlun**
- **setzoning**
- **shutdown**
- **showall**
- **setcache**

6.5.2.2 ddn_stat

This command is used to collect statistical information. The following services are provided by the **ddn_stat** command:

- **getbasic**
- **getlength**
- **repeatIO**
- **repeatMB**

For more information, read the man page or check the command's help.

6.5.2.3 ddn_init

This command is used for the initial setup of a singlet or a couplet. It must be used very carefully as it usually restarts the singlet(s).

The command uses the information preloaded in the ClusterDB. Some parameters may be overwritten using the command line.

ddn_init connects to each singlet through the serial port, using **conman**. Thus, it may be necessary to provide the name of the conman console.

A login/password is required to modify the singlet configuration. **ddn_init** attempts to connect with factory defaults login/password, using a command line supplied login/password, and with the login/password defined in **/etc/storageadmin/ddn_admin.conf**. The **ddn_admin** command then enforces the login/password defined in **ddn_admin.conf.**

6.5.2.4 ddn_conchk

This command checks the connections to a DDN system, and compares them with the connections predefined in the **ClusterDB**.

Conman, the serial network and the LAN must be ready for use in order to check the Serial/Ethernet consistency.

Attached nodes must be up, running, and reachable from the management station to check the fibre channel consistency.

6.5.2.5 ddn_set_up_date_time

This command is used to update the date and time of DDN subsystems with the UTC date and time of the management station. The administrator can specify a list of DDN systems to be synchronized.

A recommended practice, which is the installation default, is to periodically synchronize all DDN systems using a daily **cron**.

6.5.2.6 ddn_check_format

This command allows you to check the formatting status for a list of DDN systems.

6.5.2.7 ddn_firmup

This command automatically upgrades the firmware of the singlets of a DDN system. The Management Node can be used as TFTP server.

6.5.3 Bull Optima1250 Storage Systems

The administrator must be familiar with the **OPTIMA1250** Storage System terminology and management tasks.

Note The High Availability solution does not apply for nodes which are connected to **Optima1250** Storage Bays.

See The **StoreWay** OPTIMA1250 Storage System documentation for more information on the options, parameters and possible values.

The **xyr_admin** command usually requires at least one input parameter:

- The IP address of the controller of the target OPTIMA1250.

The following services are provided by the command:

- **getstatus**
- **list**
- **checkformat**
- **luninfo**
- **zoninfinfo**
- **poolbind**
- **ldbind**
- **sparebind**
- **setldmap**
- **setldwwn**
- **poolunbind**
- **ldunbind**
- **spareunbind**
- **unconfig**

The OPTIMA1250 are managed using a single login/password. The **xyr_admin** command uses the parameters that are defined in the **/etc/storageadmin/xyr_admin.conf** file as follows:

```
# XYRATEX host Administrator (where the CLI is installed)
xyr_cli_ip = 127.0.0.1
xyr_cli_user = root

# OPTIMA1250 Storeway Master Administrator login
xyradmin = admin

# OPTIMA1250 Storeway Master Administrator password
xyrpasswd = password
```

For more information, read the man page or check the command's help.

6.5.4 EMC/Clariion (DGC) Storage Systems

The administrator must be familiar with EMC/Clariion terminology and management tasks. See the **Navisphere®** CLI documentation for more information on options, parameters and possible values.

The **dgc_admin** command is used to get information or configure an EMC/Clariion disk array.

The storage system to be managed is recognized using one of the identifiers below:

- The IP address (or IP name) of one of the Service Processors
- The name of the storage system

The following services are provided by the **dgc_admin** command:

- **unconfig all** - to delete the current configuration
- **unconfig zoning** - to delete the LUN access control configuration only
- **checkformat** - to check if a formatting operation is in progress
- **direct <Navisphere CLI command>** - pass-through mode for the original **Navisphere®** CLI commands

6.6 Using Management Tools

Please refer to the storage system documentation to understand which management tools are available. Then determine how they can be accessed from Bull cluster Management Node using Linux utilities (**conman**, **telnet**, **web browser**, **X11**, **rdesktop client**, **ssh client**, etc.).

6.7 Configuring Storage Devices

6.7.1 Planning Tasks

Storage system configuration requires careful planning. At least two steps are required.

STEP 1 – DEFINE THE DEVICE CONFIGURATION

The storage administrator must define the storage configuration which is required for the cluster. It is especially important for **RAID** storage systems, which enable the creation of logical disks (LUNs) with full flexibility in terms of number and size.

Usually, the storage configuration is a compromise of several parameters:

- The available storage resources and configuration options for the storage systems.
- The performance requirements (which may drive the choice of RAID types, LUN numbers, LUN size, stripping parameters, memory cache tuning, etc.).
- The file systems and applications requirements. It is thus necessary to identify which cluster nodes will use the storage resources, the applications and/or services running on these nodes, and the system requirements for each one.

At the end of this planning phase, the administrator must be able to define for each storage system:

- The grouping of hardware disks (HDD) and the **RAID** modes to use.
- The **LUN**s to be created on each RAID volume, with their size and, if necessary, tuning parameters.
- The **LUN** access control rules. This means how the storage system should be configured to ensure that a LUN can be accessed only by the cluster node which is supposed to use this storage resource. Depending on the way the nodes are connected to a storage system, two methods of LUN access control can be used:
    1. Port-mode **LUN** access control: describes the visibility of the LUNs on each port of the storage system.
    2. **WWN**-mode LUN access control: describes the visibility of the LUNs according to the initiator's worldwide name (WWN of the host fibre channel adapter). This method requires the collection of WWN information on nodes before applying the configuration on the storage systems.

Note With some versions of Fibre Channel adapter node drivers, the correct detection of the LUNs for a storage device port is dependent on the accessibility of a LUN numbered LUN 0. It is recommended the Access Control groups for a storage device are configured so that the list of LUNs declared in each group always include an external LUN that is numbered LUN 0.

- Miscellaneous tuning parameters.

STEP 2 – DEPLOY THE STORAGE CONFIGURATION

Changing the configuration of a storage system may not be a transparent operation for the cluster nodes using storage resources which have been configured previously.

Thus the storage administrator is advised to respect the following process when deploying a storage configuration:

- Stop all the applications accessing data on the selected storage systems.
- Unmount the file systems accessing data on the selected storage systems and, if possible, shutdown the nodes.
- Modify the storage system configuration.
- Restart the attached nodes, or force them to re-discover the modified storage resources.
- Update the node's configuration.
- Mount file systems, restart applications.

6.7.2 Deployment Service for Storage Systems

Note This service is currently supported for FDA storage systems.

Medium and large clusters are usually built with multiple storage systems with the same hardware configuration. The purpose of the deployment service is to simplify the configuration tasks by:

- Automatically deploying the same logical configuration on multiple storage systems.
- Forcing I/O nodes to discover the storage resources and to setup a deterministic disk naming to simplify resource discovery on I/O nodes. This mechanism also ensures a persistent device naming.

This deployment service is well suited for storage systems and nodes dedicated to a single function, such as the I/O system of the cluster. It is hazardous to use it on storage systems or nodes which have multiple functions, such as nodes which are simultaneously Management Nodes and I/O nodes. Read the explanation and warnings of the next paragraphs carefully, to determine if this powerful and automated process is suitable for your cluster.

6.7.3 Understanding the Configuration Deployment Service

The configuration deployment service relies on modeling the storage system configuration. The model defines all the configuration parameters (see 6.7.1 Planning Tasks, Step 1). The model contains the list of the target storage systems to be configured.

The recommended process to modify the storage configuration in a large cluster, using the storage configuration deployment service, follows.

WARNING

The administrators must follow the 3 step process described in the following paragraphs. Otherwise, there is a high risk of inconsistency between storage systems and nodes, leading to a non operational file system.

STEP 1 – DEFINE THE STORAGE CONFIGURATION

The administrator must either create a model to specify the storage configuration to deploy, or use an existing model.

The administrators can define multiple models. They are responsible for managing versions and for remembering the purpose of each model.

STEP 2 – DISABLE THE GLOBAL FILE SYSTEM

If necessary, backup all the data that must be preserved.

Release the storage resources used on the I/O nodes. Typically, unmount and stop the global file system.

STEP 3 – CONFIGURE THE STORAGE SYSTEMS USED BY THE GLOBAL FILE SYSTEM

The model contains all the directives to configure the storage systems. When multiple storage systems must be configured with the same configuration, the configuration operations are performed in parallel.

WARNING

The application of a model on a storage system is destructive. The configuration deployment is not an incremental process that modifies only the differences between the current configuration and the model. The first step erases the existing configuration, and then the new configuration is built using a known reference. All data will be lost.

The application of the model stops when all the commands have been acknowledged by the target storage systems. A synthetic report is provided to quickly identify which storage devices have been successfully configured and which ones have failed.

Usually, the configuration does not complete, and tasks such as disk formatting continue to run. Another command is used to check that these tasks complete.

6.7.3.1 STEP 1 - Preparing and Managing Configuration Models

The configuration model is a text file. It uses an XML syntax style. To obtain details about the syntax, the administrator can refer to the **template.model** file, delivered with the rpm in **/usr/share/doc/storageadmin-framework-<version>**.

Another way to obtain a model template is to use the following command:

```
stormodelctl -c showtemplate
```

This template describes one LUN model for each supported storage system vendor (some parameters are vendor-specific).

A model is identified by its file name. The **.model** suffix is mandatory and a recommended practice is to store all the models in the same directory. The ClusterDB contains a history of the models applied to the storage systems. Thus the administrators should not change the contents of a model without changing its name.

A global model is made up of a list of LUN models.

A LUN model is a description of the configuration to be applied to a storage system; it includes:

- A description of LUNs using an associated label.
- LUN Access control rules describing the LUNs visibility for host ports.
- Storage system tuning parameters.
- A list of the storage systems to configure using the LUN model.

6.7.3.2 STEP 2 – Disabling the Global File System

Before changing the configuration of storage systems, it is mandatory to stop I/O activity, stop the global file system and unmount the local file systems on the nodes attached to the storage systems.

6.7.3.3 STEP 3 - Applying a Model to Storage Systems

Note It is possible to skip the storage system configuration phase and to use only the I/O Node configuration phases. In this case the administrator must manually configure the storage system, in accordance with the configuration defined in the model. This way of operating is also useful when the administrator does not want to erase the existing configuration (for example to safeguard existing data), or for the storage systems that do not support the automatic configuration.

The application of a configuration model to storage systems is performed in two phases:

1. The configuration of storage resources and tuning of parameters
2. The application of LUN access control directives

If the LUN access control method used is the **WWN**-mode (use of <NodePort> directives in the model file, see the model template for detailed description), it is necessary to update the cluster database with information about the Fibre Channel adapters of the cluster nodes before applying the configuration model. This is done using the following command:

```
ioregister -a
```

If the LUN access control method used is the Port-mode (use of <StoragePort> directives only in the model file), there is no need to use this command.

A model contains a list of storage systems to configure. The **stormodelctl** command checks the state of the storage systems in the **ClusterDB** before attempting to configure them.

```
stormodelctl -c applymodel -m <model>
```

WARNING
This operation destroys all the data stored in the storage systems listed in the model.

Important

It may be necessary to wait several minutes for the completion of the command. No message will be displayed for the intermediate steps.

The administrator can exclude storage systems from the list (**-e** flag), or add storage systems (**-i** flag).

The **stormodelctl** command returns a configuration message in the following format:

```
<disk array name> : <message>
```

The output may be processed by the **dshbak** command so the results are reformatted.

The administrator must check the output of the command. If errors are reported for some disk arrays, detailed analysis is required to diagnose and resolve the problem. The **stormodelctl** command can then be used to apply selectively the model on the disk arrays that have not been configured, using the **–i** flag.

The **–v** flag provides a verbose output, giving better control of the operations performed on the storage system.

The command only transmits the configuration information to the target storage systems. LUN formatting is a background task. To control the formatting process, use the **checkformat** sub-command:

```
stormodelctl -c checkformat -m <model>
```

Important

Wait for the command to complete before running the next step.

Please refer to the help of the **stormodelctl** command for additional options.

6.8 User Rights and Security Levels for the Storage Commands

6.8.1 Management Node

Situation 1: superuser (= root) user

All the storage commands are available but it is not recommended to launch any of them as the root user for obvious security reasons.

Situation 2: non root user

Nagios user: The storage views, like all the **Bull System Manager - HPC Edition** web pages, are only accessible for the Nagios user who is automatically created during the installation/configuration of the cluster – see Chapter 3 *Cluster Database Management* for more details.

Any specific security rules/access rights will have been applied to the storage commands. Therefore, the non root users, for example, admin, must be part of the **dba** group, and the Nagios supplementary group, in order to be able to launch storage commands.

For example:

```
useradd –g dba –G nagios <username>
```

Some of these **dba** restricted access commands must be used with the **sudo** command in order to have root privileges. The reason why this privilege is restricted is that these commands may access other nodes, in addition to the Management Node, by using **ssh**, to get or set information.

The following commands must be launched with **sudo**:

- **iorefmgmt**
- **ioregister**
- **lsiodev**
- **lsiocfg**
- **stordepha**
- **storioha**
- **stordepmap**
- **stormap**
- **stormodelctl**

Notes

- **sudo** is a standard linux command. It allows a permitted user/group to execute a command as the superuser or as another user, as specified in the **/etc/sudoers** file which is managed by the superuser only. This file contains a list of groups/commands which have these root privileges. Refer to the **sudo** man pages for more information. To use a command with **sudo**, the command has to be prefixed by the word 'sudo' as in the following example:

  ```
  <prompt>: sudo /usr/sbin/iorefmgmt
  ```

- The PATH of the **dba** 'username' must be defined in order to access these root commands without the absolute PATH in the sudo command:
  `export PATH=$PATH:/usr/sbin` in the **$HOME/.bashrc** of login "username"
  The **sudo** command is:

  ```
  <prompt>: sudo iorefmgmt
  ```

6.8.2 Other Node Types

All the available storage commands can only be launched as the root user, without exception.

6.8.3 Configuration Files

The configuration files, which an administrative user of the **dba** group can modify manually, are located in the **/etc/storageadmin/** directory of the management node. These files are named *.conf, for example **storframework.conf.**

Chapter 7. Parallel File Systems

This chapter explains how these file systems operate on a Bull extreme computing system. It describes in detail how to install, configure and manage the Lustre file system.

The following topics are described:

- *7.1 Parallel File Systems Overview*
- *7.2 Lustre Overview*
- *7.3 Lustre Administrator's Role*
- *7.4 Planning a Lustre System*
- *7.5 Lustre System Management*
- *7.6 Installing and Managing Lustre File Systems*
- *7.7 Monitoring Lustre System*

7.1 Parallel File Systems Overview

Parallel file systems are specifically designed to provide very high I/O rates when accessed by many processors at once.

A parallel file system provides network access to a "virtual" file system distributed across different disks on multiple independent servers or I/O nodes. Real files are split into several chunks of data or stripes, each stripe being written onto a different component in a cyclical distribution manner (striping).

For a parallel file system based on a client/server model, the servers are responsible for file system functionality and the clients provide access to the file system through a "mount" procedure. This mechanism provides a consistent namespace across the cluster and is accessible via the standard Linux I/O API.

I/O operations occur in parallel across multiple nodes in the cluster simultaneously. As all files are spread across multiple nodes (and possibly I/O buses and disks), I/O bottlenecks are reduced and the overall I/O performance is increased.

7.2 Lustre Overview

Lustre - a parallel file system - manages the data shared by several nodes, which is dispatched in a coherent way (cyclical distribution) on several disk systems. Lustre works in client / server mode. The server part supplies the functions of the file system, while the client part enables access to the file system through a mounting configuration.

Lustre relies on a set of Data and Meta Data servers which manage the following information related to the files:

- File attributes (name, access rights, hierarchy, etc.).
- File geometry, which means how a file is distributed across different servers.

When a node of the cluster needs access to the global file system, it will mount it locally via the client part. All the nodes can have access to the global file system.

MDS (MetaData Server)

MDS provides access to services called MDTs (MetaData Target).

A MDT provides a global NameSpace for a Lustre file system: it unifies the directory trees available from multiple file servers to provide a single global directory tree that can be mounted by the Lustre file system clients.

A MDT manages a backend ext3-like file system which contains all the metadata but none of the actual file data for an entire Lustre file system.

OSS (Object Storage Server)

OSS provides access to services called OST (Object Storage Targets).

An OST contains part of the file data (striping) for a given Lustre file system and very little metadata.
Each OST has its own block device and backend file system where it stores stripes of files in local ext3-like files.

One MDT and several OSTs make up a single Lustre file system and can be accessed through a Logical Object Volume (LOV). This set is managed as a group and can be compared to a NFS export or a LVM logical volume.
The LOV service is replicated on all the client nodes mounting the Lustre file system and distributes the I/O locking load among OSTs.

Lustre Client

A Lustre client results from the combination of an Object Storage Client (OSC) accessing the LOV.

A client node mounts the Lustre file system over the network and accesses the files with POSIX semantics.

Each client communicates directly with MDS and OSS.

7.3 Lustre Administrator's Role

Once the hardware has been setup and the software has been deployed, cluster administrators must perform the following tasks:

Determine how the hardware infrastructure will be used (number of file systems, size, storage resources used, allocation of I/O nodes, accessibility of the various file systems by the Lustre clients, etc.).

If necessary, modify the configuration of the storage devices and the configuration of the **Quadrics** or **InfiniBand** interconnects (network zoning, etc).

Configure the Lustre service and activate the configured file systems.

During the file system lifetime, administrators may have to perform operations such as stop, start, or repair. They may decide to update a configuration or to change the one loaded. They also need to monitor the file system to check the current performance in case of degradation of service.

7.4 Planning a Lustre System

7.4.1 Data Pipelines

There are many data pipelines within the Lustre architecture, but there are two in particular which have a very direct performance impact: the network pipe between clients and OSSs, and the disk pipe between the OSS software and its backend storage. Balancing these two pipes maximizes performances.

7.4.2 OSS / OST Distribution

The number of clients has no real impact on the number of OSSs to be deployed. To determine the number of OSSs and how to distribute OSTs, two things have to be considered:

- The maximum bandwidth required gives the number of OSSs.
- The total storage capacity needed gives the number of OSTs.

To increase efficiency, it is preferable to distribute OSTs evenly on OSSs and to have fewer larger OSTs in order to use space more efficiently.

When calculating the size of the OSS server nodes, it is recommended that the CPUs are divided into thirds: one third for the storage backend, one third for the network stack and one third for Lustre.

7.4.3 MDS / MDT Distribution

The Lustre file system stores the file striping information in extended attributes (EAs) on the MDT. If the file system has large-inode support enabled (> 128bytes), then EA information will be stored inline (fast EAs) in the extra space available.

The table below shows how much stripe data can be stored inline for various inode sizes:

Inode Size	#of stripes stored inline
128	0(all EA is stored externally)
256	3
512	13
1024	35
2048	77
4096	163

Table 7-1. Inode Stripe Data

It is recommended that MDT file systems be formatted with the inode large enough to hold the default number of stripes per file to improve performance and storage efficiency.
One needs to keep enough free space in the MDS file system for directories and external blocks. This represents ~512 Bytes per inode.

7.4.4 File Striping

Lustre stripes the file data across the OSTs in a round robin fashion.

It is recommended to stripe over as few objects as is possible to limit network overhead and to reduce the risk of data loss, in case of OSS failure.

The stripe size must be a multiple of the page size. The smallest recommended stripe size is 512 KB because Lustre tries to batch I/O into 512 KB blocks on the network.

7.4.5 Lustre File System Limitations

On the device it manages, an OST reserves up to 400MB for an internal journal and 5% for the root user. This reduces the storage capacity available for the user's data. Like an OST, on the device it manages a MDT reserve.

The encapsulated modified ext3 file system used by MDTs and OSTs relies on the standard ext3 file system provided by the Linux system and optimizes performance and block allocation. It has the same limits in terms of maximum file size and maximum file system size.

7.5 Lustre System Management

Bull Lustre management tools provide services to manage large parallel file systems during their whole life cycle. Using these tools the cluster administrator will be able to:

- Configure and install **Lustre** file systems using the Lustre OST/MDT services provided by the storage management model deployment (refer to the Storage Devices Management chapter).
- Perform management operations such as start/stop, mount/umount file systems.

The administrator can monitor and get information about the Lustre system via a graphical user interface for performance and health monitoring.

Status targets of management tools for Lustre file systems and components current activity.

7.5.1 The Lustre Database

The Lustre management tools rely on the cluster database (ClusterDB) to store and get information about:

- I/O and Metadata nodes (lustre_io_node table),
- Lustre OST/MDT services (lustre_ost and lustre_mdt tables),
- File systems currently installed on the cluster (lustre_fs and lustre_mount tables).

Some of these tables information is loaded during the cluster deployment phase: those related to the I/O and Metadata nodes implementation and to the OST/MDT services repartition. The rest is maintained by the Lustre management tools.

Specific commands allow the administrator to edit and adjust information when necessary, for example in the case of node replacement due to hardware.

Note Updating the information stored in the Lustre database has direct consequences on the Lustre system behaviour. This must be done only by skilled administrators.

7.5.1.1 lustre_tables_dba

SYNOPSIS

lustre_ost_dba ACTION [options]
lustre_mdt_dba ACTION [options]
lustre_fs_dba ACTION [options]
lustre_io_node_dba ACTION [options]

DESCRIPTION

The **lustre_*tables*_dba** set of commands allows the administrator to display, parse and update the information of the Lustre tables in the ClusterDB.

lustre_ost_dba Acts on the lustre_ost table, which describes the OST services

lustre_mdt_dba Acts on the lustre_mdt table, which describes the MDT services

lustre_fs_dba Acts on the lustre_fs table, which describes the Lustre file systems currently available on the cluster

lustre_io_node_dba Acts on the lustre_io_node table, which gives the current status of the cluster I/O and metadata nodes.

These utilities are useful for checking the correctness of **ClusterDB** contents according to the last configuration updates. They allow the further adjustment of values in the instance of mistakes or failures of the Lustre management utilities thus avoiding a full repeat of the operation. They can also be used to force the Lustre services behaviour for some precise and controlled cases.

As these act on the global cluster configuration information, they must be used very carefully. The changes they allow may introduce fatal inconsistencies in the Lustre ClusterDB information.

ACTIONS

add Adds an object description in the Lustre table.

update Updates configuration items of an object description in the Lustre table.

del Removes an object description from the Lustre table.

attach Creates a link between Lustre tables objects (i.e. attaches an OST to a file system).

detach Removes a link between Lustre tables objects (i.e. frees an OST).

list Displays object information according to the selected criteria provided by the options.

set Sets one or more status items of an object description.

-h(elp) Displays this help and exits.

-v(ersion) Displays the utility version and exits.

OPTIONS

The options list available for the actions depends on the kind of object they act on and on the action itself. Please, refer to the help of each command for option details.

7.5.2 /etc/lustre/storage.conf for Lustre Tools without ClusterDB

The **/etc/lustre/storage.conf** file stores information about the storage devices available on the cluster when ClusterDB is **NOT** present and it records which ones are OSTs and which ones are MDTs. It must be located on the management node. This file is composed of lines with the following syntax:

<ost|mdt>: name=<> node_name=<> dev=<> [ha_node_name=<>] [size=<kB>] [jdev=<> [jsize=<kB>]]

ost/mdt	This device is designated to be either an OST or a MDT.
name	The name given to the OST or MDT.
node_name	The hostname of the node containing the device.
dev	The device path (for example **/dev/sdd**).
ha_node_name	The hostname of the failover node. This has to be consistent with the content of **/var/lustre/status/lustre_io_nodes**.
size	Size of the device in kBs.
jdev	The name of the device where the **ext3** journal will be stored, if this is to be outside the main device. This parameter is optional. Loop devices cannot be used for this purpose.
jsize	The size of the journal device in kB.

Comments are lines beginning with # (sharp).

7.5.2.1 Filling /etc/lustre/storage.conf

This file is updated with the information obtained from the **/proc/partitions** or **/sys/block/** of the I/O nodes. For example, on a cluster where **ns13** is an I/O node:

```
>ssh ns13 -l root "cat /proc/partitions"
```

```
major minor  #blocks  name

   8     0   71687372 sda
   8     1     524288 sda1
   8     2   69115050 sda2
   8     3    2048000 sda3
   8    16   71687372 sdb
   8    32   17430528 sdc
   8    48   75497472 sdd
   8    64   17430528 sde
   8    80   75497472 sdf
   8    96   17430528 sdg
   8   112   75497472 sdh
```

sda and **sdb** are system disks of **ns13** so they must NOT be used as Lustre storage devices. Devices **sdd** to **sdh** are the devices which are available. 17430528 kB disks will be used as journal devices and 75497472 kB disks as the main devices.

This choice results in the following lines being included in **/etc/lustre/storage.conf** file for the management node:

```
mdt: name=ns13_sdd node_name=ns13 dev=/dev/sdd size=75497472
jdev=/dev/sdc jsize=17430528
ost: name=ns13_sdf node_name=ns13 dev=/dev/sdf size=75497472
jdev=/dev/sde jsize=17430528
ost: name=ns13_sdh node_name=ns13 dev=/dev/sdh size=75497472
jdev=/dev/sdg jsize=17430528
```

The decision as to which devices will be used as **MDTs** and which will be **OSTs** will be left to the administrator. This procedure fhas to be done for each I/O node and new lines appended to the **/etc/lustre/storage.conf** file of the management node. Bull provides a wizard to help the creation of the **storage.conf** file, this is **/usr/lib/lustre/lustre_storage_config.sh**.

7.5.2.2 Storage Inspection Wizard: /usr/lib/lustre/lustre_storage_config.sh

/usr/lib/lustre/lustre_storage_config.sh is a script that helps the administrator to complete the storage.conf file.

SYNOPSIS

lustre_storage_config.sh <node> <regexp_on_devices> <upper_size_limit_of_journal_in_kb>

node This details the node to be inspected

regexp_on_devices A regular expression on the device name as given in the **/dev/ directory** of the I/O node. Do not forget this is a regular expression, not a globbing expression.

upper_size_limit_of_journal_in_kb
The difference between data devices and journal devices is made according to their size in kBs. If a device has a size greater than this parameter, it is assumed to be a data device. If a device has a size smaller than this parameter, it is assumed to be journal device.

The output produced by the **lustre_storage_config.sh** script is a template of lines to be used by **storage.conf**. These lines may require minor modifications. **lustre_storage_config.sh** looks for **/var/lustre/status/lustre_io_nodes** that can be used to fill in the **ha_node** field, therefore **lustre_io_nodes** should have been filled in before running the **lustre_storage_config.sh** script.

For example, for a cluster with two High Availability I/O nodes: **ns6** and **ns7**. The content for the **lustre_io_nodes** is as follows:

```
NODE_NAME=ns6
NODE_HA_NAME=ns7
LUSTRE_STATUS=OK

NODE_NAME=ns7
NODE_HA_NAME=ns6
LUSTRE_STATUS=OK
```

1. Link to Lustre devices is run using **stordiskname** and the output is as follows:

```
[root@ns6 ~]# ll /dev/ldn.nec.*
lrwxrwxrwx  1 root root 8 May 19 13:23 /dev/ldn.nec.0 -> /dev/sdd
lrwxrwxrwx  1 root root 8 May 19 13:23 /dev/ldn.nec.1 -> /dev/sde
lrwxrwxrwx  1 root root 8 May 19 13:23 /dev/ldn.nec.2 -> /dev/sdn
lrwxrwxrwx  1 root root 8 May 19 13:23 /dev/ldn.nec.3 -> /dev/sdo

[root@ns7 ~]# ll /dev/ldn.nec.*
lrwxrwxrwx  1 root root 8 May 19 13:23 /dev/ldn.nec.0 -> /dev/sdd
lrwxrwxrwx  1 root root 8 May 19 13:23 /dev/ldn.nec.1 -> /dev/sde
lrwxrwxrwx  1 root root 8 May 19 13:23 /dev/ldn.nec.2 -> /dev/sdn
lrwxrwxrwx  1 root root 8 May 19 13:23 /dev/ldn.nec.3 -> /dev/sdo
```

   The same devices can be seen on both **ns6** and **ns7**.

2. All devices that start with **ldn** are to be used however it is not clear for the moment which are data devices and which are journal devices. From the management node the **lustre_storage_config.sh** script is run.

```
[root@ns2 /]# /usr/lib/lustre/lustre_storage_config.sh ns6 'ldn.*' 0

#call: ns6 ldn.* 0
```

   The resulting output is as follows:

```
ost: name=ost_ns6.nec.0 node_name=ns6 ha_node_name=ns7
dev=/dev/ldn.nec.0 size=262144
ost: name=ost_ns6.nec.1 node_name=ns6 ha_node_name=ns7
dev=/dev/ldn.nec.1 size=262144
ost: name=ost_ns6.nec.2 node_name=ns6 ha_node_name=ns7
dev=/dev/ldn.nec.2 size=46137344
ost: name=ost_ns6.nec.3 node_name=ns6 ha_node_name=ns7
dev=/dev/ldn.nec.3 size=46137344
```

   From the output above it can be seen that there are two sizes for the devices, the data devices (46137344 kBs) and the journal devices (262144 kBs).

3. The size of the journal device has been identified as 262144 kB, this means that the following command can be run:

```
[root@ns2 /]# /usr/lib/lustre/lustre_storage_config.sh ns6 'ldn.*' 262144
```

   The output is as follows:

```
/]# /usr/lib/lustre/lustre_storage_config.sh ns6   'ldn.*'   262144
#call: ns6 ldn.* 262144
ost: name=ost_ns6.nec.2 node_name=ns6 ha_node_name=ns7
dev=/dev/ldn.nec.2 size=46137344 jdev=/dev/ldn.nec.0 jsize=262144
ost: name=ost_ns6.nec.3 node_name=ns6 ha_node_name=ns7
dev=/dev/ldn.nec.3 size=46137344 jdev=/dev/ldn.nec.1 jsize=262144
```

4. The output is saved in the **storage.conf** file using the following command:

```
[root@ns2 /]# /usr/lib/lustre/lustre_storage_config.sh ns6 'ldn.*' 262144
>>/etc/lustre/storage.conf
```

5. The same operation now has to be run on **ns7**, as below.

```
[root@ns2 /]# /usr/lib/lustre/lustre_storage_config.sh ns7 'ldn.*' 262144
#call: ns6 ldn.* 262144
```

The output is as follows:

```
ost: name=ost_ns7.nec.2 node_name=ns7 ha_node_name=ns6
dev=/dev/ldn.nec.2 size=46137344 jdev=/dev/ldn.nec.0 jsize=262144
ost: name=ost_ns7.nec.3 node_name=ns7 ha_node_name=ns6
dev=/dev/ldn.nec.3 size=46137344 jdev=/dev/ldn.nec.1 jsize=262144
```

6. The output above is now saved in the **storage.conf** file using the following command:

```
[root@ns2 /]# /usr/lib/lustre/lustre_storage_config.sh ns7 'ldn.*'
262144 >>/etc/lustre/storage.conf
```

At this point, the same devices will be stored twice in the **storage.conf** file as shown in the output below.

```
ost: name=ost_ns6.nec.2 node_name=ns6 ha_node_name=ns7
dev=/dev/ldn.nec.2 size=46137344 jdev=/dev/ldn.nec.0 jsize=262144
ost: name=ost_ns6.nec.3 node_name=ns6 ha_node_name=ns7
dev=/dev/ldn.nec.3 size=46137344 jdev=/dev/ldn.nec.1 jsize=262144
ost: name=ost_ns7.nec.2 node_name=ns7 ha_node_name=ns6
dev=/dev/ldn.nec.2 size=46137344 jdev=/dev/ldn.nec.0 jsize=262144
ost: name=ost_ns7.nec.3 node_name=ns7 ha_node_name=ns6
dev=/dev/ldn.nec.3 size=46137344 jdev=/dev/ldn.nec.1 jsize=262144
```

7. A decision has to be made at this point as to which devices will have **ns6** as the master node, and which devices will have **ns7** as the master node. An example is shown below:

```
ost: name=ost_ns6.nec.2 node_name=ns6 ha_node_name=ns7
dev=/dev/ldn.nec.2 size=46137344 jdev=/dev/ldn.nec.0 jsize=262144
ost: name=ost_ns7.nec.3 node_name=ns7 ha_node_name=ns6
dev=/dev/ldn.nec.3 size=46137344 jdev=/dev/ldn.nec.1 jsize=262144
```

8. The first device should be designated as an **mdt**. This is done by replacing **ost** by **mdt**, as shown below:

```
mdt: name=mdt_ns6.nec.2 node_name=ns6 ha_node_name=ns7
dev=/dev/ldn.nec.2 size=46137344 jdev=/dev/ldn.nec.0 jsize=262144
ost: name=ost_ns7.nec.3 node_name=ns7 ha_node_name=ns6
dev=/dev/ldn.nec.3 size=46137344 jdev=/dev/ldn.nec.1 jsize=262144
```

9. **storage.conf** is now ready. If you have more than one pair of High Availability nodes then the same operation will have to be repeated for each pair of nodes.

10. The consistency of the **/etc/lustre/storage.conf** files can be checked using the command below optimal:

```
lustre_util check_storage.
```

lustre_storage_config.sh associates the data devices and journal devices in alphabetical order. On some devices, for example DDN, this association is not necessarily optimal and special tuning may be required to improve the performance.

Note If it is planned to upgrade the cluster from one which does not have a database installed to one which includes a database then **lustre_util check_storage** should not report any errors.

7.5.2.3 Loading storage.conf into the Cluster Database using load_storage.sh

load_storage.sh is a script that is used to load **storage.conf** information into the **lustre_ost** and **lustre_mdt** tables of the cluster database. This may be useful:

- If a cluster database is added to your system.
- If there is a database, but no management tools are provided for the storage devices, for example for **NEC** devices.

SYNOPSIS

/usr/lib/lustre/load_storage.sh < update|crush > <storage.conf file>

update	Adds the new devices but does not erase the existing **lustre_ost** and **lustre_mdt** tables.
crush	Remove the **lustre_ost** and **lustre_mdt** tables, and then add new devices.
storage.conf file	The path to your **storage.conf** file (this usually **/etc/lustre/storage.conf**)

7.5.2.4 Practical Recommendation

If you use a High Availability MDS as the management node it will be possible to move the **/etc/lustre/storage.conf** file to **/var/lustre/status/**, using the command below and to then make a symbolic link to this file on the 2 MDS nodes:

```
ln -s /var/lustre/status/storage.conf /etc/lustre/storage.conf
```

The same thing can be done for the **/etc/lustre/models** directory. In this way, information does need to be replicated and is available on the node where **/var/lustre/status** is mounted.

7.5.3 Lustre Networks

By default **Lustre** runs on all network layers that may be active in the kernel, for example **InfiniBand** or **Ethernet**. If you do not want **Lustre** to run on certain network layers, these network layers must be deactivated for the nodes in question.

If **Ethernet** is used as the **Lustre** network layer, it is possible to select the link on which Lustre will run. This is done by editing the **/etc/modprobe.d/lustre** file. For details see the *Lustre Operations Manual* from CFS (Section *Multihomed Servers*, sub-section *modprobe.conf*) at http://manual.lustre.org/

7.5.4 Lustre Management Configuration File: /etc/lustre/lustre.cfg

Lustre management tools use this file to get configuration information. This file must reside on all OSS and MDS nodes. Refer to **lustre_util** man page to know how to distribute this file easily.

File Syntax:

```
VARIABLE=VALUE
```

Lines beginning with # are comments.

/etc/lustre/lustre.cfg contents:

LUSTRE_MODE=XML

XML: Information about file systems is given to CFS Lustre tools using the XML format. These files are stored in the directory defined by LUSTRE_CONFIG_DIR on OSS, MDS and Management Node.

Default value is XML. This value is mandatory for failover configuration. **HTTP mode is no longer supported.**

CLUSTERDB=yes

When this variable is set to yes, storage, file systems and mount information is retrieved and stored from the clusterDB tables (**lustre_ost**, **lustre_mdt**, **lustre_mount** and **lustre_fs**.

LUSTRE_CONFIG_DIR=/etc/lustre/conf/

This variable contains the path of the directory where the XML/XMF files are created on the Management Node and where they have to be found on the OSSs and MDSs. The **lustre_util** command uses this path to store and read XML/XMF when required. This directory can be shared using NFS. If LUSTRE_MODE is set to XML, **lustre_util** creates this directory on all OSS and MDS nodes in order to copy the XML file associated with file systems during the install process, as required by CFS Lustre tools (**lconf**).

Default value is **/etc/lustre/conf/**.

LUSTRE_NET=tcp or elan or o2ib

This variable specifies the kind of network used by Lustre for the whole cluster. It is deprecated and should be left set to the **tcp** default value. It is now used by the **lustre_check** monitoring tool only.

LUSTRE_ADMIN=hostname

This variable contains the hostname of the I/O server used as central point of management for Lustre in case of cluster not monitored by a management station (CLUSTERDB="no"). The primary MDS node is to be chosen for that purpose.

No default value is defined.

LUSTRE_ADMIN2=hostname

LUSTRE_ADMIN2 is used only if the HA feature is enabled on the I/O nodes. It provides the hostname of the backup MDS used as alternative Lustre management point.

No default value is defined.

LUSTRE_LDAP_URL=ldap://hostname/

This variable contains the address of the ldap server used to store HA information. For example if the ldap server is on a node called ns2, then LUSTRE_LDAP_URL=ldap://ns2/.

No default value is defined.

LUSTRE_LDAP_URL2=ldap://hostname/

LUSTRE_LDAP_URL2 is used only when there is no management station supporting the full HA feature. In this case, it provides the LDAP URL of an alternative management station.

No default value is defined.

LUSTRE_DB_DAEMON_HOST=hostname
LUSTRE_DB_DAEMON_HOST2=hostname
LUSTRE_DB_DAEMON_PORT=tcp port

These variables should be set if High Availability is configured with the ClusterDB and are used to specify the http server daemon that updates the Cluster DB. LUSTRE_DB_DAEMON_HOST2 is to be used when the Management Node does not support the High Availability feature. An alternative LUSTRE_DB_DAEMON hostname will be provided as a backup.

No default values are defined for the hostnames. The default value for the tcp port is 56283, e.g. 0xDBDB so this appears as **LUSTRE_DB_DAEMON_PORT=56283**

LUSTRE_DEBUG=yes or no

If this variable is set to “yes”, Lustre management daemons are allowed to log trace information:

- in **/var/log/lustre** directory for failover
- in **/tmp/log/lustre** directory for database daemons

Default value is no.

LUSTRE_MGS_HOST=name of the Management Node where the MGS service is installed.

This value is used by the **lustre_util** tool to link the **MGS** with others Lustre entities, for example, **MDS, OSS**.

LUSTRE_MGS_NET= the name of the network used to read the MGS, for example, TCP or o2ib.

When the **o2ib** net type is used the **LUSTRE_MGS_HOST** name value has to be suffixed with'-ic0' which is hostname suffix for IB networks. For example, if you need to use an **InfiniBand** network to reach the MGS entity that runs on the node **zeus6** you have to:

- set **LUSTRE_MGS_NET** to **o2ib**
- set **LUSTRE_MGS_HOST** to **zeus6-ic0**

LUSTRE_MGS_ABSOLUTE_LOOPBACK_FILENAME = file for mgs loop device.

The default is **/home/lustre/run/mgs_loop**. When High Availability exists for the Management Node, select a directory which is shared for the Management Node pairs. This value is used by the **MGS** service when **lustre_util** is not used.

I/O scheduler for block devices

LUSTRE_OST_DEV_IOSCHED = noop or anticipatory or deadline or cfq
(I/O scheduler for OST devices)

LUSTRE_OST_JNR_IOSCHED = noop or anticipatory or deadline or cfq
(I/O scheduler for OST ext3 journal devices)

LUSTRE_MDT_DEV_IOSCHED = noop or anticipatory or deadline or cfq
(I/O scheduler for MDT devices)

LUSTRE_MDT_JNR_IOSCHED = noop or anticipatory or deadline or cfq
(I/O scheduler for MDT ext3 journal devices)

These variables define the I/O scheduler for block devices. For details about I/O schedulers refer to the **/Documentation/block** directory of kernel sources.

Default and recommended values are:

- **deadline** for LUSTRE_MDT_DEV_IOSCHED,
- **noop** for LUSTRE_OST_DEV_IOSCHED, LUSTRE_OST_JNR_IOSCHED and LUSTRE_MDT_JNR_IOSCHED.

If OSTs/MDTs are disc partitions (not the whole device) the choice of the scheduler is left to the Administrator.

LUSTRE_SNMP=yes or no

If this variable is set to yes, the **snmpd** server will be enabled on the I/O nodes when **lustre_util set_cfg** is called (`chkconfig --level 345 snmpd on && service snmpd restart`). This allows the OSS and MDS to send snmp traps to the Management Node when errors occur. These traps force the nagios lustre service to run in order to check the health of the file systems.

Default value is no.

DISABLE_LUSTRE_FS_NAGIOS=yes or no

Setting this to yes will disable the call of lustre_fs_nagios every 15 mn on management node.

Default value is no

LUSTRE_TUNING_FILE=/etc/lustre/tuning.conf

This is the path to the tuning file, the default value is **/etc/lustre/tuning.conf**.

7.5.5 Lustre Services Definition

The Lustre services MDT(s) and OST(s) rely on the devices created by the storage units configuration deployment. For this reason their distribution schema is tightly dependant of the storage configuration planning and vice versa.

A common model and deployment process is used for both storage units and Lustre services. The model describes the relationship between storage units, nodes, devices and Lustre services.

Refer to the "Storage Administration" chapter for more information.

Each Lustre service defined on the cluster I/O nodes is described by an entry in the **ClusterDB**. During the first cluster deployment phase, the model file is parsed for the storage elements which are created on the nodes and the information related to the Lustre services is stored in the Lustre tables of the **CLusterDB**, **lustre_mdt** for MDT services and **lustre_ost** for OST services.

This is theoretical information, which needs to be checked against the node reality using the **lustre_investigate check** utility. Inconsistencies may come from a model file error or elements configuration failure on nodes.

This check operation must be done after every cluster configuration or reconfiguration operation or every time the Lustre services configuration is modified.

7.5.5.1 lustre_investigate

SYNOPSIS

lustre_investigate check [-C <io_cell_list>|-n <nodes_list>|-f <file_system_name>]

lustre_investigate display [-C <io_cell_list>|-n <nodes_list>|-f <file_system_name>]

DESCRIPTION

lustre_investigate can be used only if the cluster configuration information is managed using the cluster management database, ClusterDB.
It allows the administrator to check the consistency between the information concerning the Lustre services and the real storage configuration available on I/O nodes.

Each Lustre service defined on the cluster I/O nodes is described by an entry in the ClusterDB. This entry provides information about the back-end device used by the service and the primary and the secondary node the service should run on.
Due to failures or cluster reconfiguration operations, this information may become obsolete. An availability status is maintained, which indicates if it is still correct or needs to be updated. This status is updated by running **lustre_investigate**.

lustre_investigate must be used from the management station. It issues check commands to each node of the selected range. The returned information is then evaluated against the one stored in the CLusterDB. It relies on the **pdsh** parallel shell to dispatch remote commands.

ACTIONS:

check	Parses the Lustre services entries in the ClusterDB according to the select criteria and checks their information consistency.
display	Displays the ClusterDB information about the Lustre services corresponding to the select criteria.

OPTIONS:

-h(elp)	Displays this help and exits.
-v(ersion)	Displays the current utility version and exits.
-C	Range of I/O cells (format [x,m-n]).
-n	Range of nodes (format <prefix>[x,m-n]).
-f	<file_system_name> is the name of the file system to work on.

If neither **-C**, **-n** nor **-f** are provided, all Lustre services declared in the cluster database management are processed.

7.5.6 Creating Lustre File Systems

7.5.6.1 Prerequisites

- **/etc/lustre/lustre.cfg** is assumed to be updated correctly as described in the section - Lustre Management Configuration File: /etc/lustre/lustre.cfg.
- If you are using a cluster database (CLUSTERDB=yes) **lustre_ost** and **lustre_mdt** tables are assumed to be updated correctly (use **lustre_investigate check** to verify).
- If you are not using a cluster database (CLUSTERDB=no), storage.conf must be correctly filled.
- Lustre tools use **ssh** to execute remote commands, so users must be allowed to log into nodes without being prompted for a password. This can be done by appending the right keys in **/root/.ssh/authorized_keys2**.

7.5.6.2 Lustre Model File (.lmf)

A Lustre model file describes one or several Lustre file systems that can be used at the same time. This means they do not share OSTs or MDT. Such files are stored in the **/etc/lustre/models** directory.

File Syntax:

```
keyword: <value>
```

Lines beginning with # are comments.

Possible Keywords:

stripe_size	Specify the stripe size in bytes. Default is 1048576 (1M).
stripe_count	Specify the number of OSTs each file should be striped onto. The default value is 2.
stripe_pattern	Only pattern 0 (RAID 0) is supported currently.
nettype	Possible values are **tcp** or **elan** or **o2ib**. The default is elan.
generic_client	The name of the directory from which the file system is mounted on the mds. If the network is elan, the default is clientelan. If network is tcp, default is **clienttcp**.
fstype	File system type. Possible values are **ldiskfs** or **ext3**. Default (and recommended) is **ldiskfs**.
failover	Enable failover support on OST. Possible values are yes or no. The default is no.
lustre_upcall	Location of the Lustre **upcall** script used by the client for recovery. No default script is defined.
portals_upcall	Location of the Portals **upcall** script used by the client for recovery. No default script is defined.
mdt_mkfs_options	Optional argument to **mkfs** for MDT. By default, no option is specified.
mdt_inode_size	Specify new **inode** size for underlying MDT ext3 file system. The default is self-evaluated.
mdt_mount_options	Optional argument to mount fs locally on the MDS. By default, no option is specified.
ost_mkfs_options	Optional argument to mkfs for OST. By default, no option is specified.

ost_inode_size Specify new inode size for underlying OST ext3 file system. The default is self-evaluated.

ost_mount_options Optional argument to mount fs locally on OSS. By default, no option is specified.

cluster_id Specify the cluster ID (one file system uses a single cluster id)

mount_options Defines the default options to mount the file system on clients. Options are separated with ",". Available options are: ro, rw, user_xattr, nouser_xattr, acl, noacl. Default is no option and the file system will be mounted rw. For example, `mount_options: ro` means that, by default, this file system is mounted in read-only mode.

quota Enables quota support. Possible values are yes or no. The default is no.

quota_options If quota is set to yes, it describes options for quota support. The default is: quotaon=ug,iunit=5000,bunit=100,itune=50,btune=50. **Do not use other settings for the moment.**

description A ONE LINE free description of your file system (up to 512 chars). The default is empty string.

If previous keywords are used in the header of the file, before any file system definition (this means before any use of the **fs_name** keyword), they set the new default values which can be locally overloaded for a file system.

fs_name This keyword is the starting point of a file system definition. It is the name of the file system (or the entry in the ldap database).fsname must be defined for each file system.

mount_path The mount-point to use to mount Lustre file system. Same mount-point must not be used for another file system defined in the same model file. Default is /mnt/lustre_<fs_name>.

ost [name=<RegExp>] [node_name=<RegExp>]
[dev=<RegExp>] [size=<RegExp>] [jdev=<RegExp>]
[jsize=<RegExp>] [cfg_status=available|formated]

Specify OSTs to use with this file system, using regular expressions matching their name, node_name, device, size, journal device, journal size or status. At least one field must be specified. If several fields are specified, only OSTs matching all fields of the lines will be chosen. You can use as many OST lines as you need. **At least one OST line must be defined for each file system.**

mdt [name=<RegExp>] [node_name=<RegExp>]
[dev=<RegExp>] [size=<RegExp>] [jdev=<RegExp>]
[jsize=<RegExp>] [cfg_status=available|formated]

Specify MDT of this file system. It is the same syntax as for the OSTs. If several MDTs match, then the first will be used.

Note Only one **MDT** line must be defined for each file system.

7.5.6.3 Extended Model Files (.xmf)

The purpose of the extended model files is to maintain a strict description of a file system. It is planned to replace xml files with this format. They have exactly the same syntax as previous model files, except that the OSTs/MDTs are strictly described and each OST/MDT line MUST point to one and only one OST/MDT of the **lustre_ost** and **lustre_mdt** tables. They can be used in place of lmf files. They are automatically generated in **LUSTRE_CONFIG_DIR** when you use **lustre_util install**, **update** or **rescue** commands.

7.5.6.4 Lustre Model Sample File

There follows a model file which describes two file systems fs1 and fs2, on a cluster with nodes called ns<XX>. Information about OSTs and MDTs can be found using lustre_ost_dba list and lustre_mdt_dba list if a cluster database is present, or in **/etc/lustre/storage.conf** if no cluster database is present.

```
#####################################################
# Firstly, the new default values for the 2
# file systems are defined

# To prevent failover
failover: no

# Set block-size to 4096 for mdt
mdt_mkfs_options: -b 4096

# Set block-size to 4096 for osts
ost_mkfs_options: -b 4096

# Network is elan
nettype: elan

# New mount options
ost_mount_options: extents,mballoc

#####################################################
# First file system : fs1

# File system name is fs1
fs_name: fs1

# mount-point of this file system will be /mnt/lustre1
# instead of the default /mnt/lustre_fs1
mount_path: /mnt/lustre1

# To specify osts hosted by nodes with names ending by odd numbers, with device
        names ending from 2 to 4
ost: node_name=ns.*[1,3,5,7,9] dev=.*[2-4]

# To specify the ost named ost_ns10.ddn1.6
ost: name=ost_ns10.ddn1.6
```

```
# The mdt will be the first hosted by ns12 with a name ending with a 3
mdt: node_name=ns12 name=.*3

#####################################################
# Second file system : fs2

# File system name is fs2
fs_name: fs2

# mount-point of this file system will be /mnt/lustre2
# instead of the default /mnt/lustre_fs2
mount_path: /mnt/lustre2

# To specify osts hosted by nodes with name ending with even numbers, with device
       names ending with 1,2,3 and 5
ost: node_name=ns.*[2,4,6,8,0] dev=.*[1-3,5]

# To specify the mdt named mdt_ns13.ddn12.31
mdt: name=mdt_ns13.ddn12.31

# To specify the generic_client to be fs2_client instead of
# clientelan
generic_client: fs2_client
```

7.6 Installing and Managing Lustre File Systems

lustre_util is the tool used to install, enable, disable, mount and unmount, one or more Lustre file systems from an administration node.

7.6.1 Installing Lustre File Systems using lustre_util

To install lustre file systems, the following tasks must be performed:

1 Use **lustre_util install** command to install the file system.

2 Use **lustre_util start** command to enable the file system.

3 Use **lustre_util mount** command to mount file systems on client nodes.

7.6.2 Removing Lustre File Systems using lustre_util

To uninstall lustre file systems, the following tasks must be performed:

1 Use **lustre_util umount** command to unmount file systems on client nodes.

2 Use **lustre_util stop** command to disable the file systems.

3 Use **lustre_util remove** command to remove the file system.

7.6.3 lustre_util Actions and Options

SYNOPSIS

lustre_util set_cfg [-n <I/O nodes list > | -p <I/O nodes rms partition>]

lustre_util install -f < lmf or xmf path > [--kfeof]

lustre_util update -f < lmf or xmf path > [--kfeof]

lustre_util fsck -f < fs_name | all >

lustre_util chk_dev -f < lmf, xmf files, or fs_name | all >

lustre_util rescue -f < fs_name | all >

lustre_util start -f < fs_name | all >

lustre_util tune_servers -f < fs_name | all >

lustre_util mount -f < fs_name | all > -n <nodes|recover|all>| -p <rms_partition> --mount <[+]opt1,opt2,...>

lustre_util umount -f < fs_name | all > -n <nodes|all> | -p <rms_partition>

lustre_util status [-f < fs_name | all >] [-n <nodes|all> | -p <rms_partition>]

lustre_util fs_status [-f < fs_name | all >]

lustre_util mnt_status [-f < fs_name | all >] [-n <nodes|all> | -p <rms_partition>]

lustre_util stop -f < fs_name | all >

lustre_util remove -f < fs_name | all >

lustre_util info -f < lmf, xmf files or fs_name | all >

lustre_util short_info -f < lmf, xmf files or fs_name | all >

lustre_util lfsck -f < fs_name | all > -n <node> -d <shared_directory>

lustre_util build_mdt_db -f < fs_name | all > -n <node> -d <directory>

lustre_util build_ost_db -f < fs_name | all > -n <node> -d <directory>

lustre_util distribute_coherency -f < fs_name | all > -n <node> -d <directory>

lustre_util scan_storage

lustre_util check_storage

lustre_util show_tuning

lustre_util show_cfg

lustre_util show_conf

lustre_util list

ACTIONS

set_cfg	Copies **/etc/lustre/lustre.cfg** to OSS and MDS nodes.
install	Checks if file systems can be installed, and then install them.
update	Updates settings of an installed file system that do not require reformatting of **previously** formatted OSTs/MDT (New OSTs, different network type, etc).
fsck	Runs e2fsck on the OST/MDT. The file system must be offline.
chk_dev	Check the devices and their links on I/O nodes.
rescue	Makes a file system usable again by formatting OSTs that are assumed to be NOT correctly formatted.

tune_servers	Set the I/O schedulers of file systems devices regarding **lustre.cfg** content. Apply the server related tunings of **tuning.conf** on OSS/MDS.
start	Makes installed file systems available for mounting.
mount	Mounts file systems on specified nodes.
umount	Unmounts file systems on specified nodes.
fs_status	Updates and prints OSS and MDS status information.
mnt_status	Updates and prints information regarding client nodes.
status	Does **fs_status** AND **mnt_status**.
stop	Disables file systems.
remove	Removes file systems.
info	Prints information (mdt,ost,path,etc.) about file systems.
short_info	Prints information (mdt,ost,path,etc.) about file systems, but sort OSTs by nodes.
Lfsck	Builds mdt,osts lfsck database and distributes coherency checking of a Lustre file system.
build_mdt_db	Build mdt lfsck database (first step of lfsck).
build_ost_db	Build osts lfsck database (second step of lfsck).
distribute_coherency	Distributes coherency checking of a Lustre file system (third step of lfsck).
scan_storage	Wizard to help configure **storage.conf**
check_storage	Check the consistency of the storage.conf or tables lustre_ost/lustre_mdt.
show_tuning	Display tuning parameter (from tuning.conf).
show_cfg	Display lustre.cfg variable.
show_conf	Display the lustre_util parameter.
list	Prints name of installed file systems or those file systems which failed to be installed.

OPTIONS

-f < file system path > File systems to work with, this can be:
- For an install and update: File system path MUST lead to an lmf file (lustre model file) or an xmf file (extended model file).
- For other operations that require the -f option, the file system path MUST be only the name of an installed (or attempted to be installed) file system.
- "all" stands for all installed file systems.

-n < nodes > **-n <nodes_list>**
Applies the command to the nodes list using pdsh syntax: name[x-y,z],...,namek.

-n all.
For mount, stands for "all clients which have mounted this fs as least one time". For umount, stands for "all clients which currently mount this fs".

-n recover .
For mount, stands for "all clients which are assumed to mount this fs". The main purpose of recover is to mount Lustre clients after a cluster emergency stop (main failure). Clients will be mounted with the same options as their previous mount.

--mount <[+]opt1,opt2,...> This allows mount options to be specified. For example, if +bar is specified for a file system which has foo as a default mount option, mount will be run on the client with -o foo,bar options. If only bar is specified (without +), mount will be run with -o bar options.

-p <rms_partition> Applies the command to the configured nodes of this running rms partition.

-F Forces commands execution even though this may be dangerous (no user acknowledgement is asked for).

-t <time_in_second> Sets the limit on the amount of time a command is allowed to execute. 0 means no timeout.

-u <user> User name to use to log onto the nodes instead of root.

--fanout Number of ssh connections allowed to run at the same time, default is 128.

--kfeof Stands for "keep formatting even on failure". Without this option, lustre_util returns as soon as the first failure is detected while formatting. With this option, lustre_util returns only when all the formatting attempts return for all devices. This can be useful when formatting a large pool of devices. This way you can check the health of all the devices in one shot, and you do not have to reformat devices that succeed in being formatted in a previous pass (using lustre_util update).

-V Verbose output.

-v Print version and exits.

-h Print help and exits.

set_cfg: Distributing /etc/lustre/lustre.cfg

This file must be copied on every OSS and MDS nodes. You can do it using the **set_cfg** command:

```
lustre_util set_cfg [ -n <I/O nodes list > | -p <I/O nodes rms
partition> ]
```

If no node parameter is specified, this command copies **/etc/lustre/lustre.cfg** of the Management Node on the nodes that host OST and/or MDT. If nodes are specified, **lustre.cfg** will be only copied on those nodes. If LUSTRE_SNMP is set to "yes", and if the variable **disable_chkconfig_for_ldap = no**, snmp server will be enabled on (selected) I/O nodes. If LUSTRE_LDAP_URL is set to a server address, this server will be enabled.

info: Printing Information about File system

```
lustre_util info -f < lmf, xmf files or fs_name | all >
```

This command will print information about the file system descriptor you specify. If you specify only a file system name, this fs must be installed and information will be retrieved from the cluster database.

short_info: Printing Information about a File system

```
lustre_util short_info -f < lmf, xmf files or fs_name | all >
```

Same purpose as the previous command but displays OST sorted by nodes.

install: Installing a lustre File system

```
lustre_util install -f <lmf or xmf path> -V [ --kfeof ]
```

This command formats the storage devices and performs operations required to install the file system such as loading file systems information into **ldap** database and/or cluster database. If **-F** is used, no user acknowledge is required. If **-F** is not specified, user must enter "yes" to go on if a file system with the same name is already installed. An **xmf** file is also automatically generated in LUSTRE_CONFIG_DIR for each file system.

Note This operation is quite long, -V (be verbose) option is recommended.

start: Enabling a Lustre File system

```
lustre_util start -f fs_name -V
```

This command enables a file system and makes it available for mounting (online). Use of -V option (be verbose) is recommended.

mount: Mounting Lustre File system

```
lustre_util mount -f fs_name -n <nodes|all|recover> | -p
<rms_partition>
[ --mount < [+]options> ]
```

This command will mount the file system on specified nodes using the mount-path defined in the model file. If this mount-path does not exist, it is automatically created. It is an error if this path is already used to mount another file system. If --mount is not specified, fs will be mounted with options defined in model file by mount_options. If you use --mount with a parameter which starts with +, fs will be mounted with default options AND with those you give to --mount. If the parameter does not start with +, fs will be mounted with only those you give to --mount.

umount: Unmounting Lustre File system

```
lustre_util umount -f fs_name -n <nodes|all> | -p <rms_partition>
```

This command unmounts the file system on specified nodes. You can use the **-n all** option if you want to unmount the file system everywhere it is mounted. If umount fails because some processes have their working directories in the mount-path, use umount again with -F option, in order to kill such processes before the umount operation.

stop: Disabling a Lustre File system

```
lustre_util stop -f fs_name
```

This command disables a file system. It will not be available for mounting any more (offline).

set_iosched: Set the I/O Schedulers of File system Devices

```
lustre_util set_iosched -f < fs_name | all >
```

The main purpose of **set_iosched** is to be used as call-back when migration occurs and to set the I/O schedulers on the nodes where lustre services are restarted. You do not have to use it directly as **lustre_util start** sets the I/O schedulers automatically.

remove: Removing a Lustre File system

```
lustre_util remove -f fs_name
```

This command totally removes the file system. All data will be lost. If **-F** is used, the action is done directly without any need of a user acknowledgement.
If **-F** is not used, the user is prompted and must answer explicitly "yes".

fs_status: Updating File system Status and Printing File system Information regarding OSS and MDS

```
lustre_util fs_status [ -f fs_name ]
```

This command updates the status of OSTs, MDTs, and file systems. If no file system parameters are provided, all installed file systems are checked. The output appears as follows:

```
FILE SYSTEM STATUS

file system   Config status   Running status   Number of clts migration
tv2ost8       installed       offline          0            0 OSTs migrated
tv2fs1      installed       online           3            0 OSTs migrated
tv2fs2      installed       online           4            0 OSTs migrated
```

The **config status** can take one of the following values:

not installed	**fs** is not installed (it should never be visible).
loaded but not installed	**fs** information is in a database but **lustre_util install** failed.
Formatting	**lustre_util install** is running.
checking	**lustre_util fsck** is running.
installed	fs is correctly installed.
not usable	**fs** is not correctly installed, because some devices failed to be formatted or **fsck** failed to repair some devices.

The **Running status** can take one of the following values:

offline	**fs** is correctly stopped.
Starting	**lustre_util** start is running.
online	**fs** is started and OSTs/MDT are healthy.
not correctly started	**fs** failed to start, some OSTs or MDT may be offline or unhealthy.
CRITICAL	**fs** started, but for unknown reasons, some OSTs or MDT may be offline or unhealthy.

WARNING	fs is started, but OSS or MDS may not be reachable and their states cannot be checked (IB or elan can work).
stopping	**lustre_util stop** is running.
not correctly stopped	**fs** failed to stop, some OSTs or MDT are still online or are in an unhealthy state.

mnt_status: Updating Clients Status and printing File system Information regarding Clients

```
lustre_util mnt_status [ -f fs_name ] [-n <nodes|all> | -p
<rms_partition> ]
```

This command checks if the file system is correctly mounted or unmounted on specified nodes. If no node is specified, **mnt_status** gives the status of all client nodes that work with this file system. If no file system parameter is provided, all installed file systems are checked. The output looks similar to the following:

```
CLIENT STATUS

file system   Correctly     should be mounted but   Correctly unmounted
              mounted       are not
tv2ost8       None                                    tv5
tv2fs1      tv[0-2]
tv2fs2      tv[0-2,5]     tv[3-4]
```

status: Updating Status of Servers and Clients, printing File system Information regarding Servers and Clients

```
lustre_util status [ -f fs_name ] [ -n <nodes|all> | -p
<rms_partition>  ]
```

This command performs a **fs_status** AND a **mnt_status** operation.

fsck: running e2fsck on OSTs and MDT

```
lustre_util fsck -f fs_name
```

This command runs e2fsck on OSTs and MDT. It reports if some devices have been repaired, if some nodes need to be rebooted, and also if some devices have unrecoverable errors. This command should be applied to offline file systems.

chk_dev: Check Devices and their links on I/O Nodes

```
lustre_util chk_dev  -f < lmf, xmf files or fs_name | all >
```

This command checks devices information on file systems I/O nodes:

- If the device exists.
- If the device is managed by **stormap**, it checks if device is up or down.
- If size in MBs is the expected size.

lfsck: Builds mdt,osts lfsck Database and distributes Coherency Checking of a Lustre File system

```
lustre_util lfsck -f < fs_name | all > -n <node> -d <shared_directory>
```

<node> is a client which can mount the file system, but the **fs MUST NOT** be mounted when you start to use **lfsck.**

<shared_directory> is a shared directory where the **lfsck** database files will be placed. The I/O nodes and the client node must have read/write access to this directory using the same path.

Note The database **lfsck** files can be large, depending on the number of files in the file system (10GB or more for millions of files), so ensure there is enough space in the shared directory before using **lfsck**.

lfsck is to be used **ONLY** when unrecoverable errors have been found on OST devices or when OSTs have been reformatted. It attempts to correct problems such as:

- Inode exists but has missing objects = dangling inode. This normally happens if there was a problem with an OST.
- Inode is missing but OST has unreferenced objects = orphan object. This normally happens if there was a problem with the MDS
- Multiple inodes reference the same objects. This can happen if there was corruption on the MDS, or if the MDS storage is cached and loses some but not all of its writes.

After using **lustre_util lfsck**, you should check **lost+found** in the mountpoint of client.

Using **lfsck** is the same as using **build_mdt_db**, followed by **build_ost_db**, and then **distribute_coherency**.

build_mdt_db, build_ost_db, distribute_coherency : step by step lfsck

```
lustre_util build_mdt_db -f < fs_name | all > -n <node> -d <directory>
lustre_util build_ost_db -f < fs_name | all > -n <node> -d <directory>
lustre_util distribute_coherency  -f < fs_name | all > -n <node> -d <directory>
```

These options are to be used:

To restart an **lfsck** operation which has failed, avoiding the need to restart the process from the beginning. **Lustre_util** will provide information regarding which options should be used and when.

If the directory is not a shared directory and there is a need to copy database files, **lustre_util** will provide information regarding which files should be copied and where.

These operations should be done in done in the following order: **build_mdt_db**, then **build_ost_db**, and then **distribute_coherency**.

update: Update File systems already Installed

```
lustre_util update -f fs_name -V
```

This command allows you to update an **ALREADY INSTALLED** and offline file system with new settings (that do not require a reformatting of the **ALREADY FORMATED** devices):

- **stripe_count**
- **nettype**
- **generic_client**
- **failover**
- **mdt_mount_options**
- **ost_mount_options**
- **cluster_id**
- **mount_path**
- **quota**
- **quota_options**
- **description**
- **mount_options**
- **ost** (new OST can be added, previous OSTs must also be included and do not forget that their **cfg_status** should be currently "formatted". OSTs that currently have their **cfg_status** set to "format_failed" may be removed).

Update is done by updating the model file or the corresponding extended model file with the new settings. The following settings MUST be the same:

- **mdt**(mdt line of model file must lead to the same mdt, do not forget that the **cfg_status** of the mdt should be currently "formatted")
- **ost** that were previously part of the file system and that currently have their **cfg_status** set to "formatted".
- **mdt_mkfs_options**
- **mdt_inode_size**
- **ost_mkfs_options**
- **ost_inode_size**
- **fs_name**

An update operation should only be done on a file system which has been stopped correctly.

If High Availability is in use and if the OSTs are distributed on 2 OSSs that are mutually the failover node of each other then all OSTs must be on their primary location otherwise the update will take a long time.

Once the model file is updated, run:

```
lustre_util update -f <path to modified lmf/xmf>.
```

New OSTs will be formatted, new automatically generated xmf file will be copied to the right place, and mdt will be updated (write_conf). Only OSTs that have their **cfg_status** set to "format_failed" before the update may be removed.

Removing correctly formatted OSTs of a file system can cause data loss, Lustre_util will not allow this to be done.

Update can also be used after the installation of a new release of Lustre, if the underlying way of storing information on the **MDT** has changed.

rescue: Try to make the Installed File system Work again

```
lustre_util rescue -f fs_name -V
```

This command can be used on installed file systems which have stopped:

- If the update failed (may be because new OSTs cannot be formatted)
- If fsck detects devices with unrecoverable errors
- Or for other reasons.

This command checks which OSTs have been successfully formatted and formats those that are assumed to be not correctly formatted. Theoretically, the file system should be usable again, **but data may be lost**.

check_storage : Checking Consistency of storage.conf or lustre_ost/lustre_mdt Tables

```
lustre_util check_storage
```

The main purpose of this option is to check if **storage.conf** has been correctly completed by the administrator. It should not be necessary to use this if a cluster database is used, however, this option can be available if required.

show_tuning: Display the Tuning Parameters

```
lustre_util show_tuning
```

Display the tuning parameters according to the content of **/etc/lustre/tuning.conf**.

show_cfg: Display lustre.cfg Variable

```
lustre_util show_cfg
```

Display lustre.cfg variable.

show_conf: Display lustre_util Configuration

```
lustre_util show_conf
```

Display lustre_util configuration, according to the content of **/etc/lustre/lustre_util.conf**.

list: Gives the List of Installed File systems

```
lustre_util list
```

This command prints the name of the file systems which are installed, even if their installation is not yet complete.

Note An example of the complete process to create and install a Lustre file system is described in the *Installation and Configuration Guide*.

7.6.4 lustre_util Configuration File /etc/lustre/lustre_util.conf

This file contains some additional settings for **lustre_util.** The following values are set by default:

```
ssh_connect_timeout=20
```

Timeout in seconds given to the **connect_timeout** parameter of SSH.

```
install_timeout=0
```

Timeout in seconds for install, update and rescue operations and can be overwritten by the **-t** option.

```
start_timeout=0
```

Timeout in **s** for the start operation and can be overwritten by the **-t** option.

```
mount_timeout=60
```

Timeout in **s** for the mount operation and can be overwritten by the **-t** option.

```
umount_timeout=60
```

Timeout in **s** for the umount operation and can be overwritten by **-t** option.

```
stop_timeout=0
```

Timeout in **s** for the stop operation and can be overwritten by the **-t** option.

```
status_timeout=30
```

Timeout in **s** for **status, fs_status, mnt_status** operation and can be overwritten by the **-t** option.

```
set_ioscheds_timeout=60
```

Timeout in s for setting I/O schedulers on I/O nodes (in start and tune_servers operation), can be overloaded by -t option.

```
set_tuning_timeout=60
```

Timeout in s for applying tuning parameters on I/O nodes (in start, tune_servers and mount operation), can be overloaded by -t option.

```
disable_nagios=no [yes]
```

yes will disable the update of the nagios pipe by **lustre_util.**

```
disable_chkconfig_for_ldap=yes [no]
```

yes will disable the **chkconfig** of **ldap** service in the **set_cfg** operation, **no** will allow this operation. It should be set to yes if administration node is an HA node.

```
use_stormap_for_chk_dev=yes [no]
```

If **yes**, **lustre_util** will check health of devices using **stormap -l**. It should only be set to no if stormap is not installed on I/O nodes. It is not a problem if devices you are using are not managed by **stormap.**

```
allow_loop_devices=no [yes]
```

Unless you explicitly want to use loop device, this should be set to no. This way, it prevents lconf to create huge loop devices in /dev/ directory when some LUNS disappear.

```
check_only_mounted_nodes_on_mnt_status=no [yes]
```

If set to yes, only nodes that are assumed to mount a file system will be checked on **status** and **mnt_status** operation.

```
default_fanout=128
```

Number of **ssh** connexions allowed to run at the same time. Can be overloaded using - -fanout option.

7.6.5 Lustre Tuning File /etc/lustre/tuning.conf

This file contains tuning parameters. The syntax is the following:

"<string>" <file> <target> [<delay>] [<file systems>]

"<string>"	String to write in file, it can contain spaces, MUST be between double-quotes.
<file>	Full path to the file where string will be written. Globbing is allowed. 2 macros can be used: **${mdt}** stands for the name of the **mdt** of the file system. **${ost}** stands for the name of ALL the **osts** (one line will be generated for each ost).
<target>	A string composed of the **OSS, MDS**, or **CLT**, separated by semicolons. **OSS**, **MDS** and **CLT** can be followed by a nodes list (**pdsh syntax**) using colon.
<delay>	A time in ms that we have to wait before continuing setting tuning parameters on a node. This is an optional argument, and the default is 0 ms.
<file system>	A list of file system separated with semicolons. This is an optionnal argument, and the default is to allow this tuning for every file systems.

For OSS and MDS, tuning parameters are set when a file system is started. For Clients, tuning parameters are set when the file system is mounted, for example:

- **"1" proc/sys/lnet/panic_on_lbug OSS;MDS;CLT**
  This line will enable panic on **lbug** on **ALL** types of node for all file systems by running **echo "1" >proc/sys/lnet/panic_on_lbug** on all nodes.
- **"0" /proc/sys/lnet/panic_on_lbug OSS:ns[5-6];MDS:ns3 fs1;fs2**
  This line will disable panic on **lbug**:
  - on ns5 and ns6, if they are used as an OSS of **fs1** and/or **fs2,**
  - on ns3, if it is used as MDS of **fs1** and/or **fs2.**

String, file and target can be aliased using the following syntax:
alias <name>=<content>

alias can be declared anywhere in the file, but it also acts on the **WHOLE** file, not only on the lines that follow the declaration.

When you use **alias** on a string, the alias must also be in double quotes.

Example:

A **tuning.conf** example file is shown below:

```
####  ALIAS DECLARATION ###################################
```

```
alias health_check=/proc/fs/lustre/health_check
alias panic_on_lbug=/proc/sys/lnet/panic_on_lbug
alias ping_osc=/proc/fs/lustre/osc/*${ost}*/ping
alias debug=/proc/sys/lnet/debug

#### TUNING PARAMETER ##################################

"1"                   ping_osc        CLT

"0"                   panic_on_lbug   CLT

"0"                   panic_on_lbug   OSS;MDS

"524288"              debug           OSS;MDS;CLT
```

7.6.6 Lustre File system Reconfiguration

This procedure allows you to change the distribution of the Lustre services which are defined on the I/O nodes, without having to re-deploy (which involves configuring the DDN storage systems and High Availability). The file systems involved in the new distribution are stopped; the others continue to be operational.

The following example describes how to stop the `fs1` and `fs2` file systems.

1. If needed save the data of the `fs1` and `fs2` file systems.

2. Unmount the `fs1` and `fs2` file systems:

```
lustre_util umount -f fs1 -n all [-F]
lustre_util umount -f fs2 -n all [-F]
```

3. Stop the `fs1` and `fs2` file systems:

```
lustre_util stop -f fs1 [-F]
lustre_util stop -f fs2 [-F]
```

4. Remove the `fs1` and `fs2` file systems:

```
lustre_util remove -f fs1
lustre_util remove -f fs2
```

5. Make the required modifications in the models associated with the file systems. In our example `fs1` and `fs2` are grouped together in only one `fs3` file system.

6. Configure the new `fs3` file system (this operation erases the `fs1` and `fs2` file systems data).

```
lustre_util install -f /etc/lustre/model/fs3.lmf
```

7. Start the new `fs3` file system:

```
lustre_util start -f fs3
```

8. Mount the new `fs3` file system:

```
lustre_util mount -f fs3 -p p2
```

9. If needed, restore the saved data.

7.6.7 Using Quotas with Lustre File Systems

7.6.7.1 Quota Settings in Model Files

Quotas are enabled by setting "quota" to "yes" in lmf file:

```
quota: yes
```

The default quota options are as follows:

```
quota_options: quotaon=ug,iunit=5000,bunit=100,itune=50,btune=50
```

quotaon=<u|g|ug> Enable quota for user|group|user and group.

iunit=<number of inodes> **iunit** is the granularity of inodes quotas. Inodes are acquired and released by a slice of inuit. iunit is a int type (>0), the default value in Lustre is 5000 inodes.

bunit=<size in MB> **bunit** is the granularity of block quotas. Blocks are acquired and released by a slice of bunit MBs on each OSTs. bunit is expressed in MBs (>0), the default value in Lustre is 100 MBs.

itune=<percentage> **itune** sets the threshold to release and acquire iunit inodes. For example, if a user/group owns n*iunit+m inodes, iunit inodes will be acquired for this user as soon as m goes above itune*iunit/100. If a user/group owns n*iunit-m inodes, iunit inodes will be released for this user/group as soon as m goes above itune*iunit/100. itune is a int type (100 > itune > 0), the default value in Lustre is 50.

btune=<percentage> **btune** sets the threshold to release and acquire bunit block MBs for each OST. For instance, if a user/group owns n*bunit+m MB on one OST, bunit MBs will be acquired on this OST for this user/group as soon as m goes above btune*bunit/100. If a user/group owns n*bunit-m MBs on one OST, bunit MBs will be released on this OST for this user/group as soon as m goes above btune*bunit/100 MB. btune is a int type (100 > btune > 0), the default value in Lustre is 50.

7.6.7.2 Starting Quota: lfs Quotacheck

Once the file system is installed, started and mounted, run the following command on a client:

```
lfs quotacheck -<quotaon parameter> <mount_point>
```

This means that if quota_options are as follows:

`quotaon=ug,iunit=5000,bunit=100,itune=50,btune=50 and mountpoint is /mnt/lustre,`

then it will be necessary to run:

```
lfs quotacheck -ug /mnt/lustre
```

The time taken by **quotacheck** depends on the size of the biggest device used by the file system as OST or MDT. On average, it takes 160s for a 1TB OST/MDT check.

7.6.7.3 Setting the Limits: lfs Setquota

lfs setquota sets limits on blocks and files.

```
lfs setquota [-u|-g] <name> <block-softlimit> <block-hardlimit>
<inode-softlimit> <inode-hardlimit> <mount_point>
```

block-softlimit and **block-hardlimit** are expressed in kB.

Inode-softlimit and **inode-hardlimit** are expressed in number of inodes.

Limits on blocks/inodes MUST be greater than **bunit/iunit**. This means, for example, **bunit=100MB, block-softlimit** and **block-hardlimit** must be greater than 102400kB. If you have **iunit=5000**, **inode-softlimit** and **inode-hardlimit** must be greater than 5000.

Limits on blocks must be greater than the number of OST * bunit. This means, for example, if there are 9 OSTs and **bunit**=100 MBs, **block-softlimit** and **block-hardlimit** must be greater than 9*100*1024=921600 kBs.

For example:

```
lfs setquota -u bob 900000 1000000 5000 10000 /mnt/lustre
```

will set a **block-softlimit** to **900MB**, **block-hardlimit** to **1GB**, **inode-softlimit** to **5000**, **inode-hardlimit** to **10000** for user **testfs**, for a lustre file system mounted on /mnt/lustre.

```
lfs setquota -g dba 900000 1000000 5000 10000 /mnt/lustre
```

The command above will implement the same settings for all users of group **dba.**

Restrictions

- At present, soft limits are not supported in Lustre. So set **block-softlimit** and **inode-softlimit** to 0.
- It is strongly recommended to run **setquota** on a Lustre file system which is not busy. Otherwise an incorrect **block- hardlimit** value may be set.

7.6.7.4 Updating/Rescuing a File system with Quota enabled

If a file system is rescued, quota will have to be enabled again using the command below.

```
lfs quotacheck -<quotaon parameter> <mount_point>
```

If a file system is updated and new OSTs are not added the following command will have to be run again:

```
lfs quotacheck -<quotaon parameter> <mount_point>
```

If a file system is updated and **new OSTs are added** then the **fs** will have to be updated, started and mounted and then the following command run:

```
lfs quotacheck -<quotaon parameter> <mount_point>
```

For ***ALL*** groups and users, all the limits may be set to 0 with the following command:

```
lfs setquota -u <user> 0 0 0 0 <mount_point>

lfs setquota -g <group> 0 0 0 0 <mount_point>
```

For ***ALL*** groups and users, the limits may be set to their former values with the following command.

```
lfs setquota [-u|-g] <name> <block-softlimit> <block-hardlimit>
<inode-softlimit> <inode-hardlimit> <mount_point>
```

7.7 Monitoring Lustre System

Status information about the Lustre file system and I/O nodes is kept up to date in the ClusterDB by the Lustre management tools.

Using this information and that collected by performance daemons, the **Bull System Manager - HPC Edition** supervision tool offers items specific to the Lustre system allowing the health and performance to be monitored from the management station – see the chapter on monitoring for more details.

7.7.1 Lustre System Health Supervision

7.7.1.1 The all status Map view

This includes global status indicators which provide the administrator with information about the global I/O system availability.

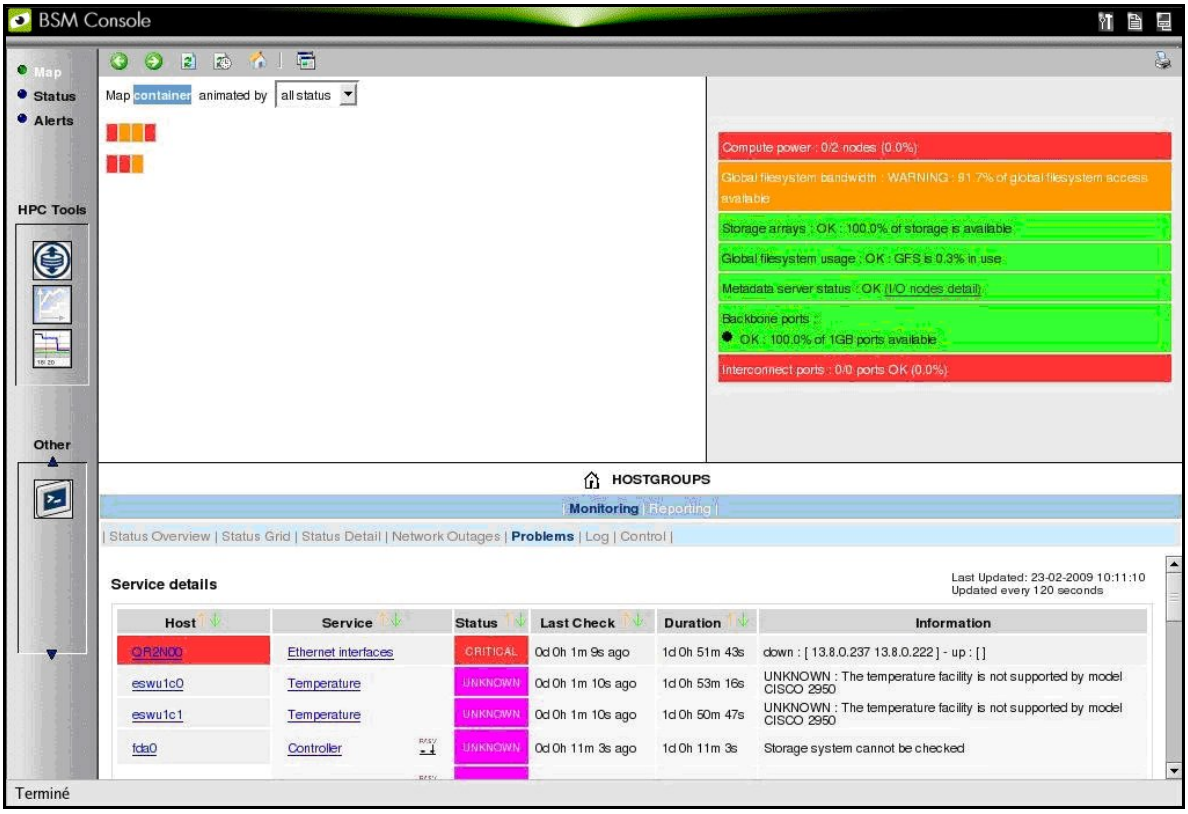


Figure 7-1. **Bull System Manager** - Map view

System Availability Indicators are located at the right top of the topological view and provides a status to the Administrator at a glance. These include:

Available Global File System Bandwidth as a Percentage

This is indicated as a percentage of I/O nodes available. An I/O node is fully available if it has its three Quadrics rails and its height fibre links up and if its Lustre status is OK. If not, a degradation factor is applied as follows:

- **cancel** the node if Lustre is not OK
- **apply** a 30% factor of degradation per quadrics rail missing
- **apply** a 12% factor of degradation per fibre link missing

Available Storage Arrays as a Percentage

The ratio of running storage appliances against the total number is indicated.

Global File System Usage

This gives the current usage rate of the **Lustre** system for all the Lustre file systems together.

MDS Migration Alert

If **High Availability** is configured, this alerts the administrator to a **MDS** failover migration. The Lustre system then no longer has the High-Availability status.

7.7.1.2 File systems Health Monitoring

This is done by the script **/usr/bin/lustre_fs_nagios**. It checks the state of each OSTs/MDTs, and sets the status of the file systems into the ClusterDB according to whether they are online or offline. This script is called every 15 min on the Management Node using **/etc/cron.d/lustre_fs_nagios.cron**, which is automatically installed and enabled by lustre_utils RPM.

lustre_fs_nagios should not be used online by the administrator; however, it can be used to force a refresh of **nagios** lustre file system status entry.

7.7.1.3 The lustre_check Tool

The **lustre_check** tool keeps the I/O node availability information up to date in the **Cluster Database**. It runs on the Management Node, scheduled by a **cron** every 15 min.

When called, it checks the I/O nodes and collects network and storage information. This information is stored for each node in the **lustre_io_node** table of the database, where it is scanned regularly by the supervision tools.

The **lustre_check** tool is unlikely to be used on-line by the Administrator; however, it can be used to force a refresh of the Cluster database information and to get a node by node status instantly.

7.7.2 Lustre File system Indicator

Within Bull System Manager the **Nagios** service plug-ins include a plug to monitor the health for the Lustre file system.

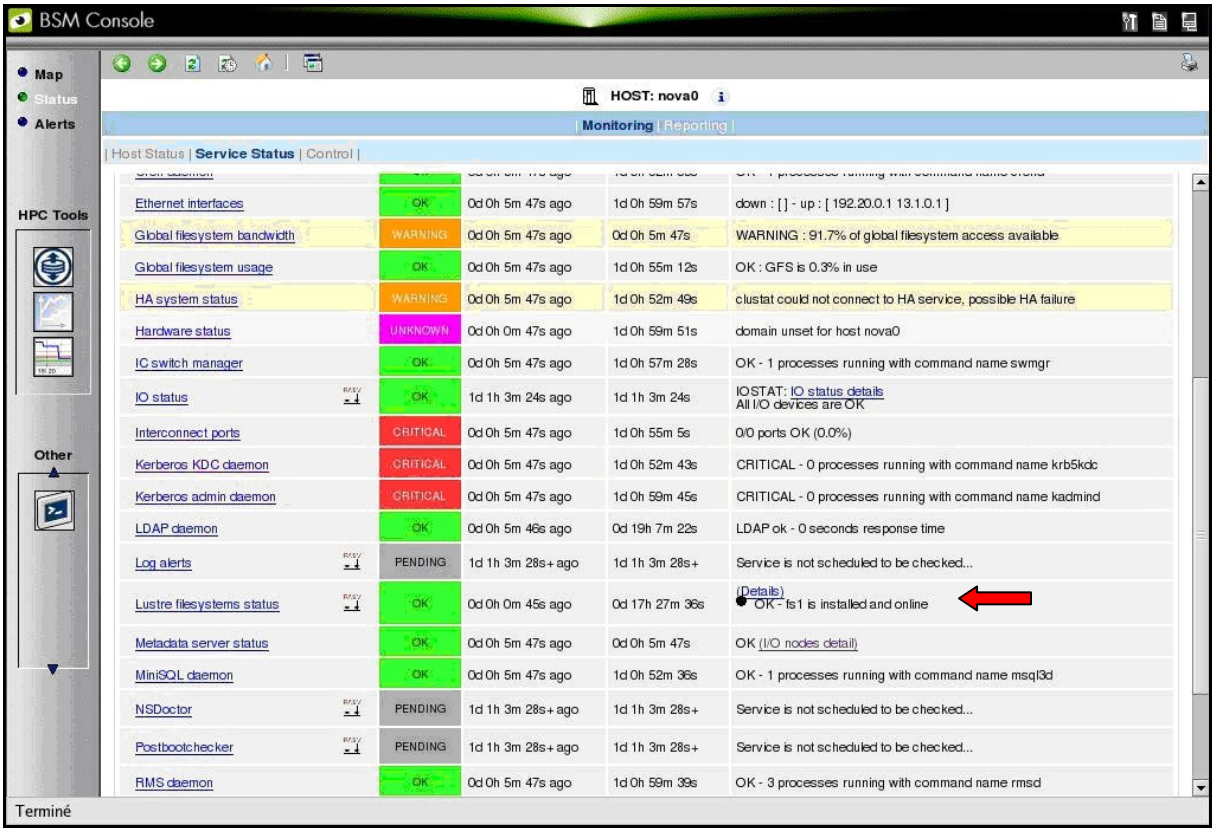


Figure 7-2. NovaScale Lustre FileSystems Status

The **Lustre** file system indicator relates to the Lustre file systems health as a whole. Clicking on the info link will displays a detailed status for each file system running.

Lustre Management Node Web Interface

With a web browser, you can easily check the Lustre file system status using the following URL: http://<mangement node>/lustre

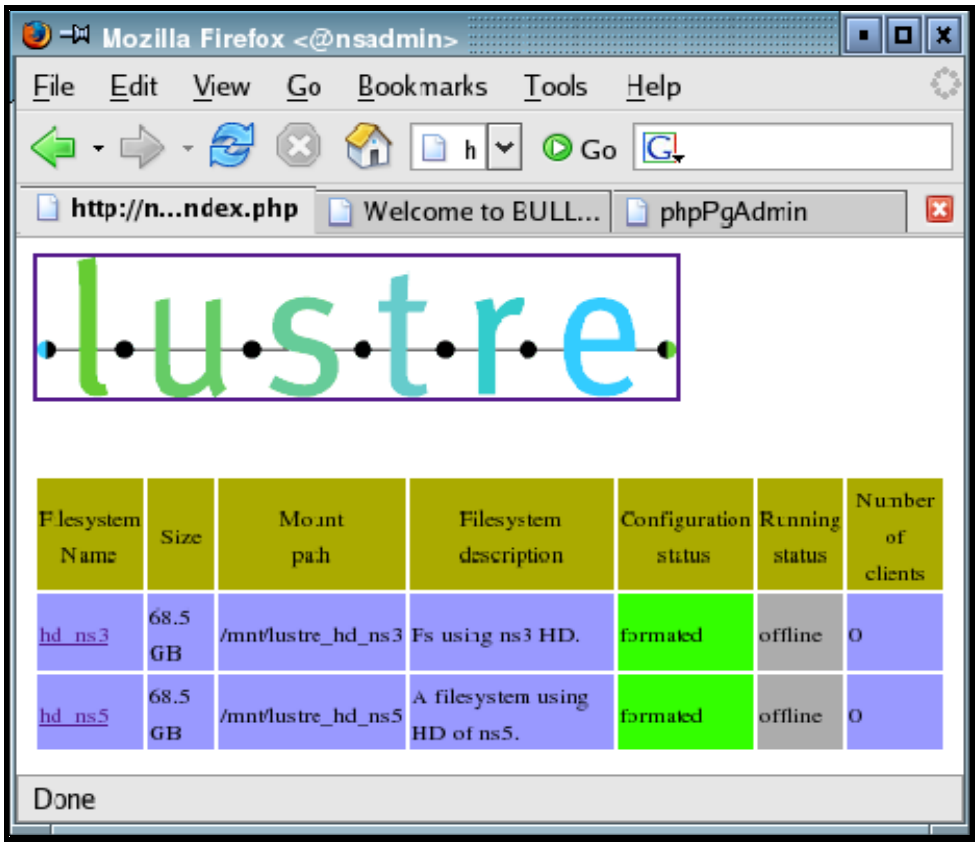


Figure 7-3. Lustre Management Node web interface

By clicking on the file system name, you can get details about the file system, using an interface that allows you to sort OSTs by name, active node, primary node, secondary node, device size, journal device, Config status, status or migration status.

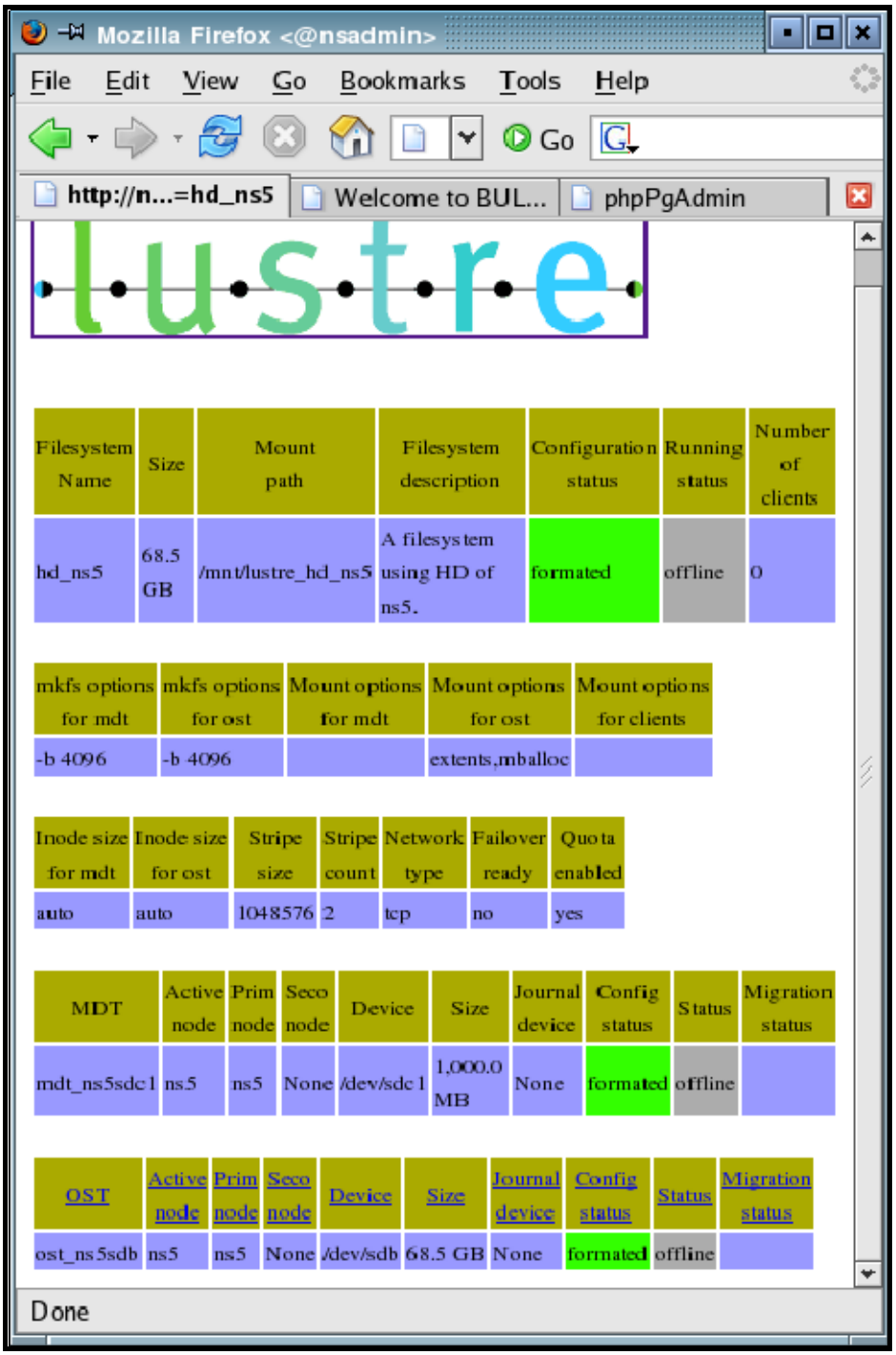


Figure 7-4. Detailed view of Lustre file systems

7.7.3 Lustre System Performance Supervision

7.7.3.1 Group Performance Views

By clicking on the Group performance button in the Bull System Manager console the administrator is provided with an at-a-glance view of the transfer rates of the Lustre system for the file systems all together. The information period can be specified.

Clicking on the compiled view will display a dispatched view giving the performance rates node by node for the same time period.

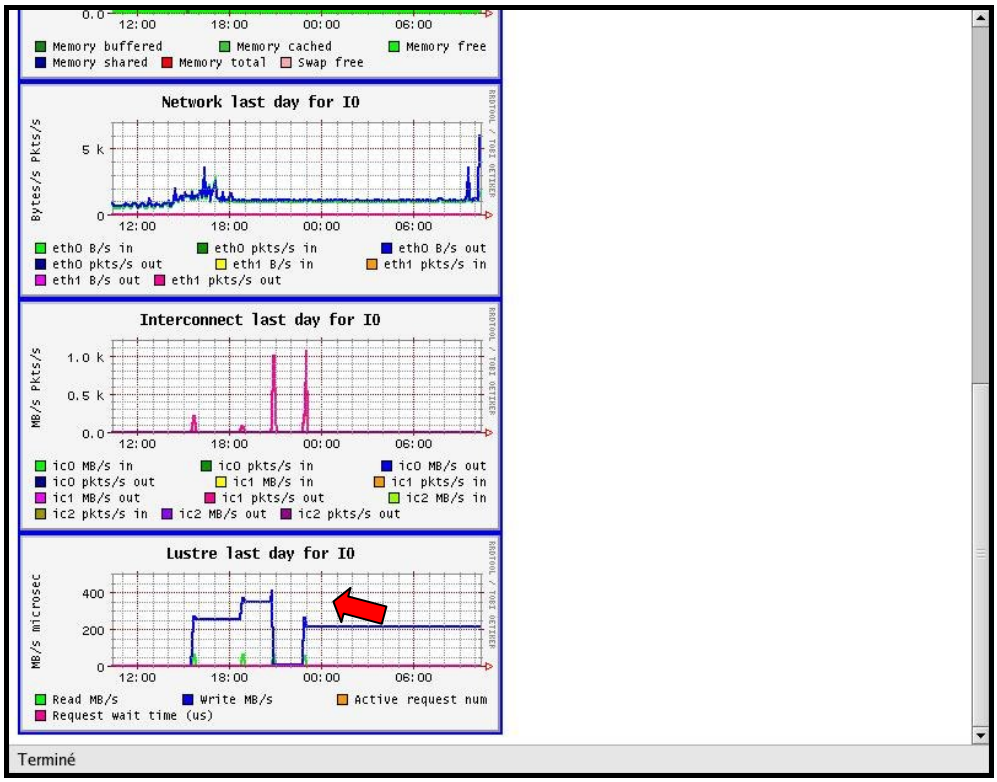


Figure 7-5. Group performance global view pop up window

Figure 7-6. Dispatched performance view pop up window

7.7.3.2 Node Performance Views

Views related to Lustre system local transfer and filling rates are available for each I/O node from the Global Performance view in the Bull System Manager Console.

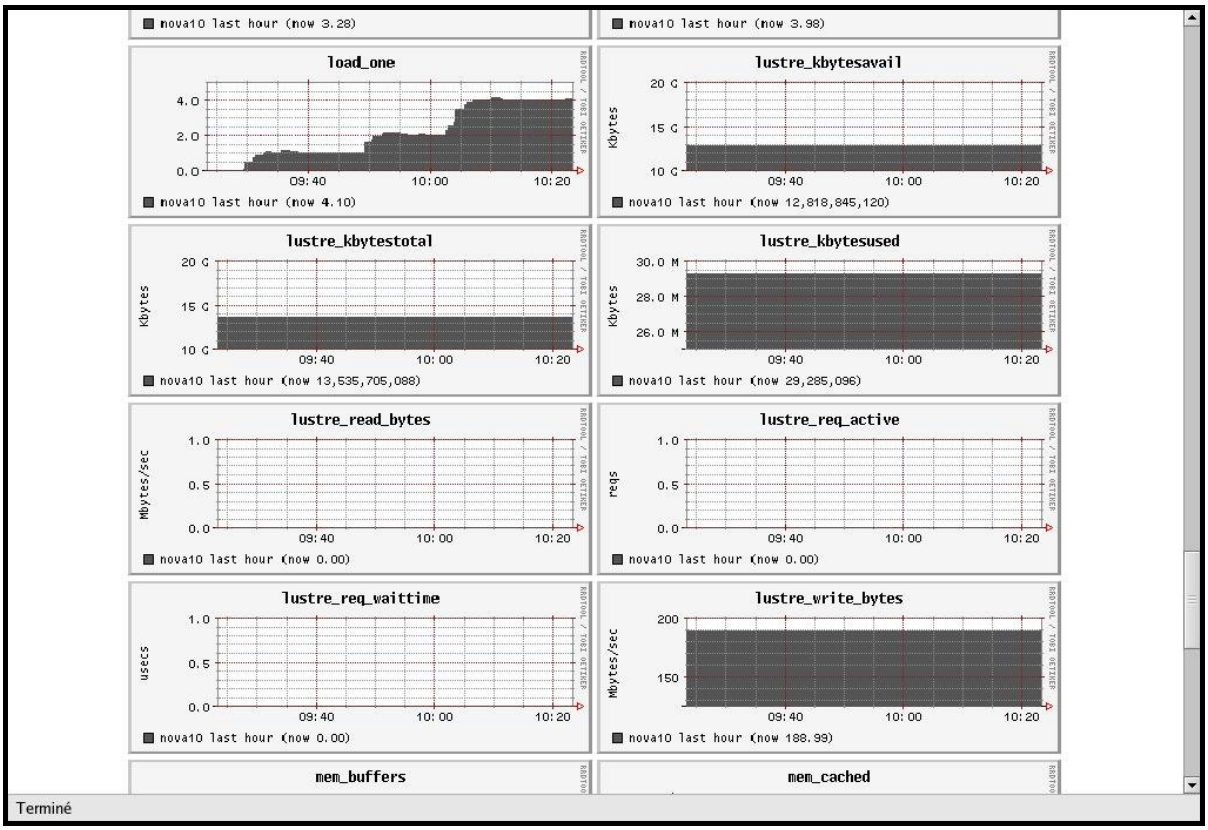


Figure 7-7. Global performance view pop up window

Chapter 8. SLURM Resource Manager

Merely grouping together several machines on a network is not enough to constitute a real cluster. Resource Management software is required to optimize the throughput within the cluster according to specific scheduling policies.

A **resource manager** is used to allocate resources, to find out the status of resources, and to collect task execution information. From this information the scheduling policy can be applied. Bull extreme computing platforms use **SLURM,** an open-source, scalable resource manager.

This chapter describes the following topics:

- 8.1 *Resource Management with SLURM*
- 8.2 *SLURM Configuration*
- 8.3 *Administrating Cluster Activity with SLURM*

8.1 Resource Management with SLURM

8.1.1 SLURM Key Functions

As a cluster resource manager, SLURM has three key functions. Firstly, it allocates exclusive and/or non-exclusive access to resources (compute nodes) to users for some duration of time so they can perform work. Secondly, it provides a framework for starting, executing, and monitoring work (normally a parallel job) on the set of allocated nodes. Finally, it arbitrates conflicting requests for resources by managing a queue of pending work.

Users interact with SLURM using various command line utilities:

- **SRUN** to submit a job for execution.
- **SBCAST** to transmit a file to all nodes running a job.
- **SCANCEL** to terminate a pending or running job.
- **SQUEUE** to monitor job queues.
- **SINFO** to monitor partition and the overall system state.
- **SACCTMGR** to view and modify SLURM account information. Used with the **slurmdbd** daemon
- **SACCT** to display data for all jobs and job steps in the SLURM accounting log.
- **SBATCH** for submitting a batch script to SLURM
- **SALLOC** for allocating resources for a SLURM job
- **SATTACH** to attach to a running SLURM job step.
- **STRIGGER** used to set, get or clear SLURM event triggers.
- **SVIEW** used to display SLURM state information graphically. Requires an Xwindows capable display.

See The man pages for the commands above for more information.

System administrators perform privileged operations through an additional command line utility, **SCONTROL**.

The central controller daemon, **SLURMCTLD**, maintains the global state and directs operations. Compute nodes simply run a **SLURMD** daemon (similar to a remote shell daemon) to export control to **SLURM**.

SLURM supports resource management across a single cluster.

SLURM is not a sophisticated batch system. In fact, it was expressly designed to provide high-performance parallel job management while leaving scheduling decisions to an external entity. Its default scheduler implements **First-In First-Out** (FIFO). A scheduler entity can establish a job's initial priority through a plug-in.

An external scheduler may also submit, signal, and terminate jobs as well as reorder the queue of pending jobs via the API.

8.1.2 SLURM Components

SLURM consists of two types of daemons and various command-line user utilities. The relationships between these components are illustrated in the following diagram:

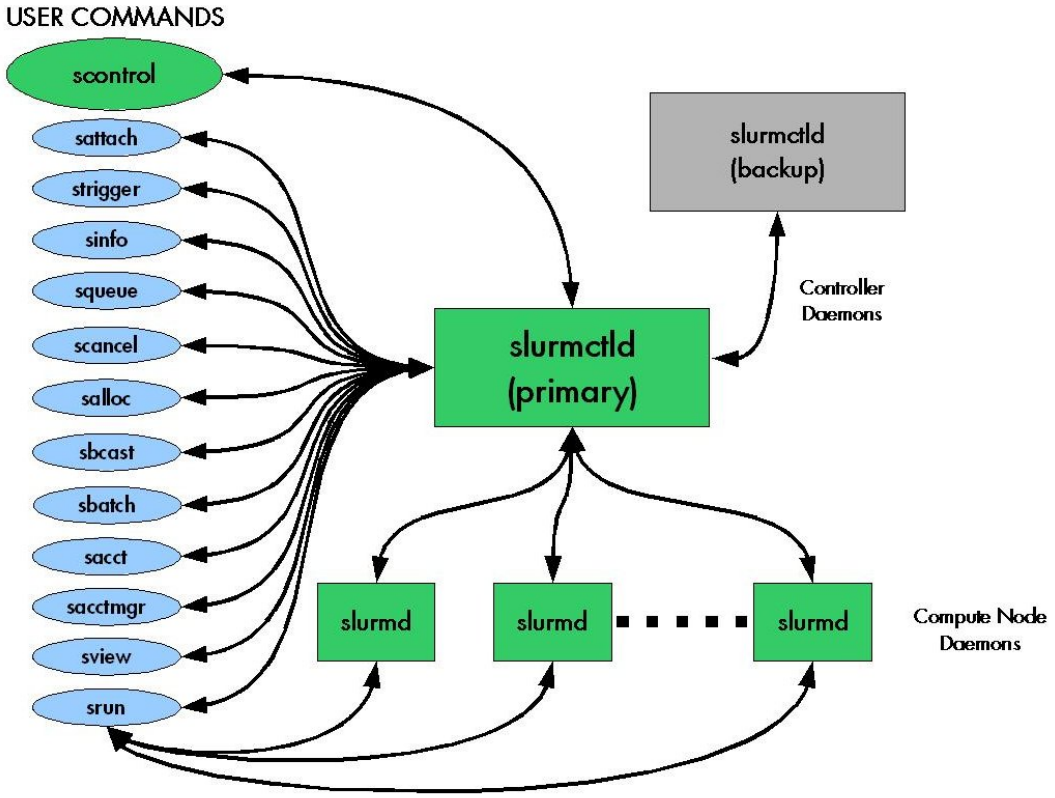


Figure 8-1. SLURM Simplified Architecture

8.1.3 SLURM Daemons

8.1.3.1 SLURMCTLD

The central control daemon for **SLURM** is called **SLURMCTLD**. SLURMCTLD is *multi* – threaded; thus, some threads can handle problems without delaying services to normal jobs that are also running and need attention. **SLURMCTLD** runs on a single management node (with a fail-over spare copy elsewhere for safety), reads the **SLURM** configuration file, and maintains state information on:

- Nodes (the basic compute resource)
- Partitions (sets of nodes)
- Jobs (or resource allocations to run jobs for a time period)
- Job steps (parallel tasks within a job).

The **SLURMCTLD** daemon in turn consists of three software subsystems, each with a specific role:

Software Subsystem	Role Description
Node Manager	Monitors the state and configuration of each node in the cluster. It receives state-change messages from each compute node's SLURMD daemon asynchronously, and it also actively polls these daemons periodically for status reports.
Partition Manager	Groups nodes into disjoint sets (partitions) and assigns job limits and access controls to each partition. The partition manager also allocates nodes to jobs (at the request of the Job Manager) based on job and partition properties. SCONTROL is the (privileged) user utility that can alter partition properties.
Job Manager	Accepts job requests (from SRUN or a metabatch system), places them in a priority-ordered queue, and reviews that queue periodically or when any state change might allow a new job to start. Resources are allocated to qualifying jobs and that information transfers to (SLURMD on) the relevant nodes so the job can execute. When all nodes assigned to a job report that their work is done, the Job Manager revises its records and reviews the pending-job queue again.

Table 8-1. Role Descriptions for SLURMCTLD Software Subsystems

The following figure illustrates these roles of the SLURM Software Subsystems.

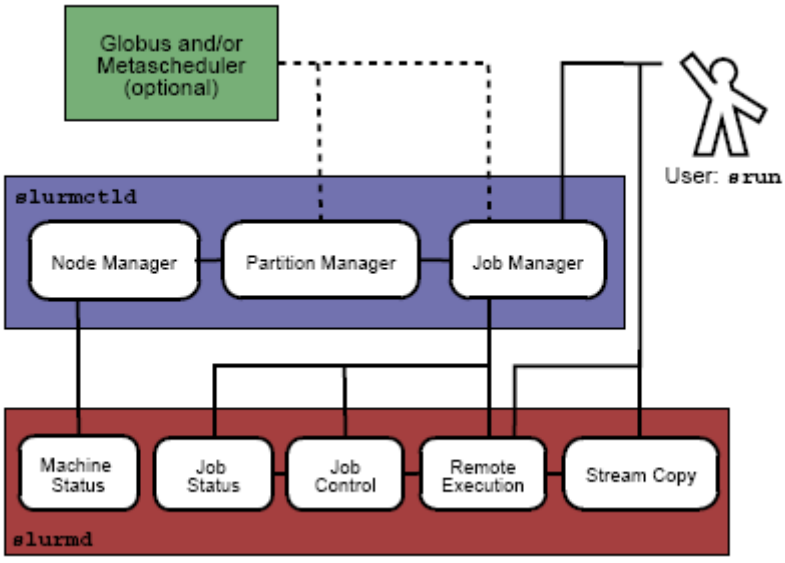


Figure 8-2. SLURM Architecture - Subsystems

8.1.3.2 SLURMD

The **SLURMD** daemon runs on all the compute nodes of each cluster that **SLURM** manages and performs the lowest level work of resource management. Like **SLURMCTLD** (previous subsection), **SLURMD** is multi-threaded for efficiency; but, unlike SLURMCTLD, it runs with root privileges (so it can initiate jobs on behalf of other users).

SLURMD carries out five key tasks and has five corresponding subsystems. These subsystems are described in the following table.

SLURMD Subsystem	Description of Key Tasks
Machine Status	Responds to **SLURMCTLD** requests for machine state information and sends asynchronous reports of state changes to help with queue control.
Job Status	Responds to **SLURMCTLD** requests for job state information and sends asynchronous reports of state changes to help with queue control.
Remote Execution	Starts, monitors, and cleans up after a set of processes (usually shared by a parallel job), as decided by **SLURMCTLD** (or by direct user intervention). This can often involve many changes to process-limit, environment-variable, working-directory, and user-id.
Stream Copy Service	Handles all **STDERR**, **STDIN**, and **STDOUT** for remote tasks. This may involve redirection, and it always involves locally buffering job output to avoid blocking local tasks.
Job Control	Propagates signals and job-termination requests to any SLURM-managed processes (often interacting with the Remote Execution subsystem).

Table 8-2. SLURMD Subsystems and Key Tasks

8.1.4 Scheduler Types

The system administrator for each machine can configure **SLURM** to invoke one of several alternative local job schedulers. To determine which scheduler SLURM is currently invoking on any machine, execute the following command:

```
scontrol show config |grep SchedulerType
```

where the returned string will have one of the values described in the following table.

Returned String Value	Description
builtin	A first-in-first-out scheduler. SLURM executes jobs strictly in the order in which they were submitted (for each resource partition), unless those jobs have different priorities. Even if resources become available to start a specific job, SLURM will wait until there is no previously-submitted job pending (which sometimes confuses impatient job submitters). This is the default.
backfill	Modifies strict FIFO scheduling to take advantage of resource islands that may appear as earlier jobs complete. **SLURM** will start jobs submitted later out of order when resources become available, *and* if doing so does not delay the execution time in place for any earlier-submitted job. To increase the job's chances of benefiting from such backfill scheduling: (1) Specify reasonable time limits (the default is the same time limit for all jobs in the partition, which may be too large), and (2) Avoid requiring or excluding specific nodes by name.
wiki	Uses the Maui Scheduler, with a sophisticated set of internal scheduling algorithms. This choice can be configured in several ways to optimize job throughput. Details are posted on a support web site at the following URL: http://supercluster.org/maui
gang	Gang scheduling involves time-slicing for parallel jobs. Jobs that share resources in the same partition will be suspended and resumed in turn so that all jobs make progress. Usually these will be threads belonging to the same process, but they may also be from different processes. Gang scheduling is used so that if two threads or processes communicate with each other, they will be ready to communicate at the same time. The **slurm.conf** parameter, **SchedulerTimeSlice**, controls the duration of the gang scheduler time slices.
hold	Hold scheduling places all new jobs in a file. If the file exists it will hold all the jobs otherwise SLURM defaults to the built-in **FIFO** as described in the **builtin** section.

Table 8-3. SLURM Scheduler Types

8.2 SLURM Configuration

The SLURM configuration file, **slurm.conf**, is an ASCII file that describes the following:

- General SLURM configuration information
- The nodes to be managed
- Information about how those nodes are grouped into partitions
- Various scheduling parameters associated with those partitions.

The **SLURM** configuration file includes a wide variety of parameters. This configuration file must be available on each node of the cluster.

See The **slurm.conf** man page for a full description of the SLURM configuration parameters.

The **slurm.conf** file should define at least the configuration parameters as defined in the examples provided and any additional ones that are required. Any text following a '**#**' is considered a comment. The keywords in the file are not case sensitive, although the argument usually is (e.g., "**SlurmUser=slurm**" might be specified as "**slurmuser=slurm**"). Port numbers to be used for communications are specified as well as various timer values.

A description of the nodes and their grouping into partitions is required. A simple node range expression may be used to specify a range of nodes to avoid building a configuration file with a large numbers of entries. The node range expression can contain one pair of square brackets with a sequence of comma separated numbers and/or ranges of numbers separated by a "-" (e.g. "linux[0-64,128]", or "lx[15,18,32-33]").

Node names can have up to three name specifications: **NodeName** is the name used by all **SLURM** tools when referring to the node, **NodeAddr** is the name or IP address SLURM uses to communicate with the node, and **NodeHostname** is the name returned by the */bin/hostname -s* command. Only **NodeName** is required (the others default to the same name), although supporting all three parameters provides complete control over the naming and addressing the nodes.

Nodes can be in more than one partition, with each partition having different constraints (permitted users, time limits, job size limits, etc.). Each partition can thus be considered a separate queue. Partition and node specifications use node range expressions to identify nodes in a concise fashion. An annotated example configuration file for SLURM is provided with this distribution in **/etc/slurm/slurm.conf.example**. Edit this configuration file to suit the needs of the user cluster, and then copy it to **/etc/slurm/slurm.conf**.

8.2.1 Configuration Parameters

Refer to the **slurm.conf** man page, using the command below, for configuration details, options, parameter descriptions, and configuration file examples.

Example:

```
$ man slurm.conf
```

8.2.2 SCONTROL – Managing the SLURM Configuration

SCONTROL manages available nodes (for example, by "draining" jobs from a node or partition to prepare it for servicing). It is also used to manage the **SLURM** configuration and the properties assigned to nodes, node partitions and other SLURM-controlled system features.

Note Most **SCONTROL** options and commands can only be used by System Administrators. Some **SCONTROL** commands *report* useful configuration information or manage job *checkpoints*, and any user can benefit from invoking them appropriately.

NAME

SCONTROL - Used to view and modify SLURM configuration and state.

SYNOPSIS

```
SCONTROL   [OPTIONS...] [COMMAND...]
```

DESCRIPTION

SCONTROL is used to view or modify the SLURM configuration including: job, job step, node, partition, and overall system configuration. Most of the commands can only be executed by user root. If an attempt to view or modify configuration information is made by an unauthorized user, an error message will be printed and the requested action will not occur. If no command is entered on the execute line, SCONTROL will operate in an interactive mode and prompt for input. It will continue prompting for input and executing commands until explicitly terminated. If a command is entered on the execute line, SCONTROL will execute that command and terminate. All commands and options are case-insensitive, although node names and partition names are case-sensitive (node names "LX" and "lx" are distinct). Commands can be abbreviated to the extent that the specification is unique.

OPTIONS

For options, examples and details please refer to the man page.

Example:

```
$ man scontrol
```

8.2.3 Pam_Slurm Module Configuration

This section describes how to use the **pam_slurm** module. This module restricts access to Compute Nodes in a cluster where Simple Linux Utility for Resource Management (SLURM) is in use. Access is granted to root, any user with a SLURM-launched job currently running on the node, or any user who has allocated resources on the node according to the SLURM database.

Use of this module is recommended on any Compute Node where it is desirable to limit access to just those users who are currently scheduled to run jobs.

For **/etc/pam.d/** style configurations where modules reside in **/lib/security/**, add the following line to the PAM configuration file for the appropriate service(s) (eg, **/etc/pam.d/system-auth**):

```
account    required     /lib/security/pam_slurm.so
```

If it is necessary to always allow access for an administrative group (e.g., wheel), stack the **pam_access** module ahead of **pam_slurm**:

```
account    sufficient   /lib/security/pam_access.so
account    required     /lib/security/pam_slurm.so
```

Then edit the **pam_access** configuration file (**/etc/security/access.conf**):

```
+:wheel:ALL
-:ALL:ALL
```

When access is denied because the user does not have an active job running on the node, an error message is returned to the application:

```
Access denied: user foo (uid=1313) has no active jobs.
```

This message can be suppressed by specifying the **no_warn** argument in the PAM configuration file.

8.2.4 Installing and Configuring Munge for SLURM Authentication (MNGT)

This software component is required if the authentication method for the communication between the SLURM components is munge (where **AuthType=auth/munge**). On most platforms, the munged daemon does not require root privileges. If possible, the daemon must be run as a non-privileged user. This can be controlled by the **init** script as detailed in the *Starting the Daemon* section below.

See http://home.gna.org/munge/ for additional information about munge software

By default, the **munged** daemon uses the following system directories:

- **/etc/munge/**
  This directory contains the daemon's secret key. The recommended permissions for it are 0700.
- **/var/lib/munge/**
  This directory contains the daemon's PRNG seed file. It is also where the daemon creates pipes for authenticating clients via file-descriptor-passing. If the file-descriptor-passing authentication method is being used, this directory must allow execute permissions for all; however, it must not expose read permissions. The recommended permissions for it are 0711.

- /var/log/munge/
  This directory contains the daemon's log file. The recommended permissions for it are 0700.
- /var/run/munge/
  This directory contains the **Unix** domain socket for clients to communicate with the daemon. It also contains the daemon's **PID** file. This directory must allow execute permissions for all. The recommended permissions for it are 0755.

These directories must be owned by the user that the munged daemon will run as. They cannot allow write permissions for group or other (unless the sticky-bit is set). In addition, all of their parent directories in the path up to the root directory must be owned by either root or the user that the munged daemon will run as. None of them can allow write permissions for group or other (unless the sticky-bit is set).

8.2.4.1 Creating a Secret Key

A security realm encompasses a group of hosts having common users and groups. It is defined by a shared cryptographic key. Credentials are valid only within a security realm. All munged daemons within a security realm must possess the same secret key.

By default, the secret key resides in **/etc/munge/munge.key.** This location can be overridden using the munged command-line, or via the **init** script as detailed in the *Starting the Daemon* section below.

A secret key can be created using a variety of methods:

- Use random data from **/dev/random** or **/dev/urandom**:

```
$ dd if=/dev/random bs=1 count=1024 >/etc/munge/munge.key
```

or

```
$ dd if=/dev/urandom bs=1 count=1024 >/etc/munge/munge.key
```

- Enter the hash of a password:

```
$ echo -n "foo" | sha1sum | cut -d' ' -f1 >/etc/munge/munge.key
```

- Enter a password directly (not recommended):

```
$ echo "foo" >/etc/munge/munge.key
```

This file must be given 0400 permissions and owned by the user that the munged daemon will run as.

8.2.4.2 Starting the Daemon

Start the daemon by using the init script (**/etc/init.d/munge start**). The init script sources /etc/sysconfig/munge, if present, to set the variables recognized by the script.

The **OPTIONS** variable passes additional command-line options to the daemon; for example, this can be used to override the location of the secret key (--key-file) or set the number of worker threads (--num-threads). If the init script is invoked by root, the USER variable causes the daemon to execute under the specified username; the 'daemon' user is used by default.

8.2.4.3 Testing the Installation

Perform the following steps to verify that the software has been properly installed and configured:

1. Generate a credential on **stdout**:

```
$ munge -n
```

2. Check if a credential can be decoded locally:

```
$ munge -n | unmunge
```

3. Run a quick benchmark:

```
$ remunge
```

If problems are encountered, verify that the munged daemon is running (**/etc/init.d/munge status**). Also, check the log file (**/var/log/munge/munged.log**) or try running the daemon in the foreground (**/usr/sbin/munged --foreground**).

Some error conditions can be overridden by forcing the daemon (**/usr/sbin/munged --force**).

8.3 Administrating Cluster Activity with SLURM

SLURM consists of two types of daemons.

- **SLURMCTLD** is sometimes called the "controller" daemon. It orchestrates **SLURM** activities, including queuing of job, monitoring node states, and allocating resources (nodes) to jobs. There is an optional backup controller that automatically assumes control in the event that the primary controller fails. The primary controller resumes control when it is restored to service. The controller saves its state to disk whenever there is a change. This state can be recovered by the controller at startup time. State changes are saved so that jobs and other states can be preserved when the controller moves (to or from a backup controller) or is restarted.

  Note that files and directories used by **SLURMCTLD** must be readable or writable by the user **SlurmUser** (the SLURM configuration files must be readable; the log file directory and state save directory must be writable).

- The **SLURMD** daemon executes on all Compute nodes. It resembles a remote shell daemon which exports control to SLURM. Because SLURMD initiates and manages user jobs, it must execute as the user **root**.

8.3.1 Starting the Daemons

The SLURM daemons are initiated at node startup time, provided by the **/etc/init.d/slurm** script. If needed, the **/etc/init.d/slurm** script can be used to check the status of the daemon, **start**, **startclean** or **stop** the daemon on the node.

Once a valid configuration has been set up and installed, the **SLURM** controller, **SLURMCTLD**, should be started on the primary and backup control machines, and the **SLURM** compute node daemon, **SLURMD**, should be started on each compute server. The **SLURMD** daemons need to run as root for production use, but may be run as a user for testing purposes (obviously no jobs should be running as any other user in the configuration). The SLURM controller, **SLURMCTLD**, must be run as the configured **SlurmUser** (see the configuration file).

For testing purposes it may be prudent to start by just running **SLURMCTLD** and **SLURMD** on one node. By default, they execute in the background. Use the **-D** option for each daemon to execute them in the foreground and logging will be done to the terminal. The **-v** option will log events in more detail with more **v**'s increasing the level of detail (e.g. **-vvvvv**). One window can be used to execute **slurmctld -D -vvvvv**, whilst **slurmd -D –vvvvv** is executed in a second window. Errors such as *'Connection refused'* or *'Node X not responding'* may be seen when one daemon is operative and the other is being started. However, the daemons can be started in any order and proper communications will be established once both daemons complete initialization. A third window can be used to execute commands such as, **srun -N1 /bin/hostname**, to confirm functionality.

Another important option for the daemons is **-c** to clear the previous state information. Without the **-c** option, the daemons will restore any previously saved state information: node state, job state, etc. With the **-c** option all previously running jobs will be purged and the node state will be restored to the values specified in the configuration file. This means that a node configured down manually using the **SCONTROL** command will be returned to service unless also noted as being down in the configuration file. In practice, **SLURM** restarts with preservation consistently.

The **/etc/init.d/slurm** script can be used to **start**, **startclean** or **stop** the daemons for the node on which it is being executed.

8.3.2 SLURMCTLD (Controller Daemon)

NAME

SLURMCTLD - The central management daemon of SLURM.

SYNOPSIS

slurmctld [OPTIONS...]

DESCRIPTION

SLURMCTLD is the central management daemon of SLURM. It monitors all other SLURM daemons and resources, accepts work (jobs), and allocates resources to those jobs. Given the critical functionality of SLURMCTLD, there may be a backup server to assume these functions in the event that the primary server fails.

OPTIONS

-c

Clear all previous SLURMCTLD states from its last checkpoint. If not specified, previously running jobs will be preserved along with the state of **DOWN**, **DRAINED** and **DRAINING** nodes and the associated reason field for those nodes.

-D

Debug mode. Execute SLURMCTLD in the foreground with logging to **stdout**.

-f <file>

Read configuration from the specified file. See NOTE under ENVIRONMENT VARIABLES below.

-h

Help; print a brief summary of command options.

-L <file>

Write log messages to the specified file.

-v

Verbose operation. Using more than one v (e.g., -vv, -vvv, -vvvv, etc.) increases verbosity.

-V

Print version information and exit.

ENVIRONMENT VARIABLES

The following environment variables can be used to override settings compiled into **SLURMCTLD**.

SLURM_CONF

The location of the SLURM configuration file. This is overridden by explicitly naming a configuration file in the command line.

Note It may be useful to experiment with different **SLURMCTLD**-specific configuration parameters using a distinct configuration file (e.g. timeouts). However, this special configuration file will not be used by the **SLURMD** daemon or the **SLURM** programs, unless each of them is specifically told to use it. To modify communication ports, the location of the temporary file system, or other parameters used by other **SLURM** components, change the common configuration file, **slurm.conf**.

8.3.3 SLURMD (Compute Node Daemon)

NAME

SLURMD - The Compute Node daemon for SLURM.

SYNOPSIS

slurmd [OPTIONS...]

DESCRIPTION

SLURMD is the compute node daemon of SLURM. It monitors all tasks running on the compute node, accepts work (tasks), launches tasks, and kills running tasks upon request.

OPTIONS

-c

Clear system locks as needed. This may be required if SLURMD terminated abnormally.

-D

Run **SLURMD** in the foreground. Error and debug messages will be copied to **stderr**.

-M

Lock **SLURMD** pages into system memory using **mlockall** to disable paging of the SLURMD process. This may help in cases where nodes are marked DOWN during periods of heavy swap activity. If the **mlockall** system call is not available, an error will be printed to the log and **SLURMD** will continue as normal.

-h

Help; print a brief summary of command options.

-f <file>

Read configuration from the specified file. See NOTES below.

-L <file>

Write log messages to the specified file.

-v

Verbose operation. Using more than one v (e.g., -vv, -vvv, -vvvv, etc.) increases verbosity.

-V

Print version information and exit.

ENVIRONMENT VARIABLES

The following environment variables can be used to override settings compiled into **SLURMD**.

SLURM_CONF

The location of the **SLURM** configuration file. This is overridden by explicitly naming a configuration file on the command line.

Note It may be useful to experiment with different **SLURMD**-specific configuration parameters using a distinct configuration file (e.g. timeouts). However, this special configuration file will not be used by the **SLURMD** daemon or the **SLURM** programs, unless each of them is specifically told to use it. To modify communication ports, the location of the temporary file system, or other parameters used by other SLURM components, change the common configuration file, **slurm.conf**.

8.3.4 Node Selection

The node selection mechanism used by SLURM is controlled by the **SelectType** configuration parameter. If you want to execute multiple jobs per node, but apportion the processors, memory and other resources, the ***cons_res*** (consumable resources) plug-in is recommended. If you tend to dedicate entire nodes to jobs, the ***linear*** plug-in is recommended.

8.3.5 Logging

SLURM uses the **syslog** function to record events. It uses a range of importance levels for these messages. Be certain that your system's **syslog** functionality is operational.

8.3.6 Corefile Format

SLURM is designed to support generating a variety of core file formats for application codes that fail (see the ***--core*** option of the ***srun*** command).

8.3.7 Security

Unique job credential keys for each site should be created using the **openssl** program **openssl must be used (not ssh-keygen) to construct these keys.** An example of how to do this is shown below.

Specify file names that match the values of **JobCredentialPrivateKey** and **JobCredentialPublicCertificate** in the configuration file. The **JobCredentialPrivateKey** file must be readable only by **SlurmUser**. The **JobCredentialPublicCertificate** file must be readable by all users. Both files must be available on all nodes in the cluster. These keys are used by **slurmctld** to construct a job credential, which is sent to **srun** and then forwarded to **slurmd** to initiate job steps.

```
> openssl genrsa -out /path/to/private/key 1024
> openssl rsa -in /path/to/private/key -pubout -out /path/to/public/key
```

8.3.8 SLURM Cluster Administration Examples

SCONTROL may be used to print all system information and modify most of it.

Only a few examples are shown below. Please see the **SCONTROL** man page for full details. The commands and options are all case insensitive.

- Print detailed state of all jobs in the system.

```
adev0: scontrol
scontrol: show job
```

```
JobId=475 UserId=bob(6885) Name=sleep JobState=COMPLETED
  Priority=4294901286 Partition=batch BatchFlag=0
  AllocNode:Sid=adevi:21432 TimeLimit=UNLIMITED
  StartTime=03/19-12:53:41 EndTime=03/19-12:53:59
  NodeList=adev8 NodeListIndecies=-1
  ReqProcs=0 MinNodes=0 Shared=0 Contiguous=0
  MinProcs=0 MinMemory=0 Features=(null) MinTmpDisk=0
  ReqNodeList=(null) ReqNodeListIndecies=-1

JobId=476 UserId=bob(6885) Name=sleep JobState=RUNNING
  Priority=4294901285 Partition=batch BatchFlag=0
  AllocNode:Sid=adevi:21432 TimeLimit=UNLIMITED
  StartTime=03/19-12:54:01 EndTime=NONE
  NodeList=adev8 NodeListIndecies=8,8,-1
  ReqProcs=0 MinNodes=0 Shared=0 Contiguous=0
  MinProcs=0 MinMemory=0 Features=(null) MinTmpDisk=0
  ReqNodeList=(null) ReqNodeListIndecies=-1
```

- Print the detailed state of job 477 and change its priority to zero. A priority of zero prevents a job from being initiated (it is held in "pending" state).

```
adev0: scontrol
scontrol: show job 477
```

```
JobId=477 UserId=bob(6885) Name=sleep JobState=PENDING
  Priority=4294901286 Partition=batch BatchFlag=0
  more data removed....
scontrol: update JobId=477 Priority=0
```

- Print the state of node adev13 and drain it. To drain a node, specify a new state of DRAIN, DRAINED, or DRAINING. SLURM will automatically set it to the appropriate value of either DRAINING or DRAINED depending on whether the node is allocated or not. Return it to service later.

```
adev0: scontrol
scontrol: show node adev13
```

```
NodeName=adev13 State=ALLOCATED CPUs=2 RealMemory=3448 TmpDisk=32000
  Weight=16 Partition=debug Features=(null)
scontrol: update NodeName=adev13 State=DRAIN
scontrol: show node adev13
NodeName=adev13 State=DRAINING CPUs=2 RealMemory=3448 TmpDisk=32000
  Weight=16 Partition=debug Features=(null)
scontrol: quit
Later
adev0: scontrol
scontrol: show node adev13
NodeName=adev13 State=DRAINED CPUs=2 RealMemory=3448 TmpDisk=32000
  Weight=16 Partition=debug Features=(null)
scontrol: update NodeName=adev13 State=IDLE
```

- Reconfigure all SLURM daemons on all nodes. This should be done after changing the SLURM configuration file.

```
adev0: scontrol reconfig
```

- Print the current SLURM configuration. This also reports if the primary and secondary controllers (slurmctld daemons) are responding. To just see the state of the controllers, use the command ping.

```
adev0: scontrol show config
```

```
Configuration data as of 03/19-13:04:12
AuthType     = auth/munge
BackupAddr    = eadevj
BackupController = adevj
ControlAddr    = eadevi
ControlMachine  = adevi
Epilog      = (null)
FastSchedule   = 1
FirstJobId    = 1
InactiveLimit   = 0
JobCompLoc    = /var/tmp/jette/slurm.job.log
JobCompType    = jobcomp/filetxt
JobCredPrivateKey = /etc/slurm/slurm.key
JobCredPublicKey = /etc/slurm/slurm.cert
KillWait     = 30
```

```
MaxJobCnt     = 2000
MinJobAge     = 300
PluginDir     = /usr/lib/slurm
Prolog      = (null)
ReturnToService  = 1
SchedulerAuth   = (null)
SchedulerPort   = 65534
SchedulerType   = sched/backfill
SlurmUser     = slurm(97)
SlurmctldDebug  = 4
SlurmctldLogFile = /tmp/slurmctld.log
SlurmctldPidFile = /tmp/slurmctld.pid
SlurmctldPort   = 7002
SlurmctldTimeout = 300
SlurmdDebug    = 65534
SlurmdLogFile   = /tmp/slurmd.log
SlurmdPidFile   = /tmp/slurmd.pid
SlurmdPort    = 7003
SlurmdSpoolDir  = /tmp/slurmd
SlurmdTimeout   = 300
TreeWidth     = 50
SLURM_CONFIG_FILE = /etc/slurm/slurm.conf
StateSaveLocation = /usr/local/tmp/slurm/adev
SwitchType    = switch/elan
TmpFS       = /tmp
WaitTime     = 0

Slurmctld(primary/backup) at adevi/adevj are UP/UP
```

- Shutdown all SLURM daemons on all nodes.

```
adev0: scontrol shutdown
```

Chapter 9. PBS Professional Batch Manager

PBS Professional is the professional version of the Portable Batch System (PBS), a flexible resource and workload management system, originally developed to manage aerospace computing resources at NASA.

PBS is a distributed workload management system which has three primary roles:

Queuing
The collecting together of jobs or tasks to be run on a computer. Users submit tasks or jobs to the resource management system which places them in a queue until the system is ready to run them.

Scheduling
The process of selecting which jobs to run, where and when, according to predetermined policies. Sites balance competing needs and goals on the system(s) in order to maximize the efficient use of resources (both computer time and people time).

Monitoring
The act of tracking and reserving system resources and enforcing usage policy. This covers both user-level and system-level monitoring, as well as monitoring the jobs that are being run. Tools are provided to help the human monitoring of the PBS system as well.

See The **PBS Professional** *Administrator's Guide* (on the **PBS Professional CD-ROM** delivered for clusters which use **PBS Professional**) for more detailed information on using PBS PRO, including descriptions of some of the different configurations possible, with examples, plus descriptions of the PBS PRO Administrator commands.

This chapter describes some specific details which apply to Bull extreme computing clusters.

9.1 Pre-requisites

SLURM should not run on the same clusters as PBS Professional. If necessary deactivate SLURM by running the command chkconfig -– level 345 slurm off on the Management Node and on all the Compute Nodes.

- The root user, administrator, should have direct access to all the Compute Nodes from the Management Node, and vice versa, without having to use a password. **ssh** is used to protect this access, see *section 2.3.1.* in this manual for more information.

9.2 Post Installation checks

The **/etc/pbs.conf** file will have been created automatically during the installation of PBS PRO. This will contain the **PBS_EXEC** path (**/opt/pbs/default**) and the **PBS_HOME** directory (`/var/spool/PBS` by default).

See The *Configuring Administration Software* step in the *Installation and Configuration Guide* for more information about the installation and configuration of **PBS Pro**.

9.2.1 Checking the status of the PBS daemons

Run the following command on the Management and Compute Node to check the status of the PBS daemons

```
/etc/init.d/pbs status
```

On the Management Node output similar to that below should appear:

```
pbs_server is pid xxxx
pbs_sched is pid xxxx
```

On the Computes Nodes output similar to that below should appear:

```
pbs_mom is pid xxxx
```

9.2.2 Adding a Node to the Initial Cluster Configuration

Use the **qmgr** option, as below, to add a Compute Node to the list of Compute Nodes for a cluster:

```
/opt/pbs/default/bin/qmgr -c "create node <node_name>"
```

Use the follow command to verify that the node has been created and added to the Compute Node list:

```
/opt/pbs/default/bin/pbsnodes -a
```

9.3 Useful Commands for PBS Professional

The following commands, which are in the **/opt/pbs/default/bin** directory, may be used to test that **PBS Professional** is up and running correctly:

pbsnodes –a
Used to display the status of the nodes in cluster

qsub
Used to submit a job

qdel
Used to delete a job

qstat
Used to display the job, queue and server status

tracejob
Used to extract job info from the log files

See The **PBS Professional** *Administrator's Guide* and *User's Guide* included on the PBS Pro CD ROM for more detailed information on these and on other commands.

9.4 PBS GridWorks Analytics

The **GridWorks Analytics** feature uses a parser to collect information from the PBS Professional server node (normally this is the cluster Management Node). The Application Server installed on a Login Node shows the information stored in the analytics database, either graphically or in the form of tables. These reports can be used to analyse and improve the performance of **PBS Professional** on the cluster, and to troubleshoot configuration problems.

9.4.1 Viewing PBS GridWorks Analytics Data

GWA is accessed by opening an Internet browser session and typing in the following **URL** http://[webserverhost]:port/GWAWeb

The **webserverhost** is the host (Login Node) where the **GWA** web server is installed, and the **port** is the port for the **GWA** web server.

By default, everyone who has created a job in PBS is registered as a **Normal** user within **GWA** with the same password as their login name. Users may be placed in a particular user groups to control which data they have access to.

See
- The PBS Professional documentation for details on how to use this feature.
- The *Installation and Configuration Guide* for details on how to install and configure **GWA**.

9.5 Essential Configuration Settings for bullx cluster suite

This section describes some essential configuration settings which are required to ensure that **PBS PRO** runs smoothly on bullx cluster suite.

9.5.1 MPIBull2 and PBS Professional for all clusters (InfiniBand and Ethernet)

To use **MPIBull2** with **PBS Professional** run the following commands on both the Management Node and on all the Compute Nodes:

```
cd /opt/pbs/default/bin

pbsrun_wrap /opt/mpi/mpibull2-<version>/bin/mpirun pbsrun.mpich2
```

This will give output similar to that below:

```
pbsrun_wrap: EXECUTED: "mv /opt/mpi/mpibull2-<version>/bin/mpirun
/opt/mpi/mpibull2-1.2.1-4.t/bin/mpirun.actual"
pbsrun_wrap: EXECUTED: "cp /opt/pbs/default/bin/pbsrun
/opt/pbs/default/bin/pbsrun.mpich2"
pbsrun_wrap: EXECUTED: "chmod 755 /opt/pbs/default/bin/pbsrun.mpich2"
pbsrun_wrap: EXECUTED: "ln -s /opt/pbs/default/bin/pbsrun.mpich2
/opt/mpi/mpibull2-<version>/bin/mpirun"
pbsrun_wrap: EXECUTED: "ln -s /opt/mpi/mpibull2-
<version>/bin/mpirun.actual/opt/pbs/default/lib/MPI/pbsrun.mpich2.link"
pbsrun_wrap:EXECUTED:"chmod644/opt/pbs/default/lib/MPI/pbsrun.mpich2.init"
```

Chapter 10. 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:

- 10.1 *Launching Bull System Manager - HPC* Edition
- 10.2 *Access Rights*
- 10.3 *Hosts, Services and Contacts for Nagios*
- 10.4 *Using Bull System Manager - HPC* Edition
- 10.5 *Map Button*
- 10.6 *Status Button*
- 10.8 *Alerts Button*
- 10.9 *Storage Overview*
- 10.10 *Shell*
- 10.11 *Monitoring the Performance - Ganglia Statistics*
- 10.12 *Group Performance View*
- 10.13 *Global Performance View*
- 10.14 *Configuring and Modifying Nagios Services*
- 10.15 *General Nagios Services*
- 10.16 *Management Node Nagios Services*
- 10.17 *Ethernet Switch Services*
- 10.18 *Cool Cabinet Door Services*

10.1 Launching Bull System Manager - HPC Edition

Note The cluster database (**ClusterDB**) must be running before monitoring is started. See the chapter on *Cluster Data Base Management*.

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**

10.2 Access Rights

10.2.1 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 10-1*, the Administrator is able to enter host and service commands, whereas an ordinary user will only be able to consult the interface.

10.2.2 Standard User Access Rights

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

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

10.2.3 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.

10.3 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 **Bull System Manager – HPC Edition** 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.

10.4 Using Bull System Manager - HPC Edition

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

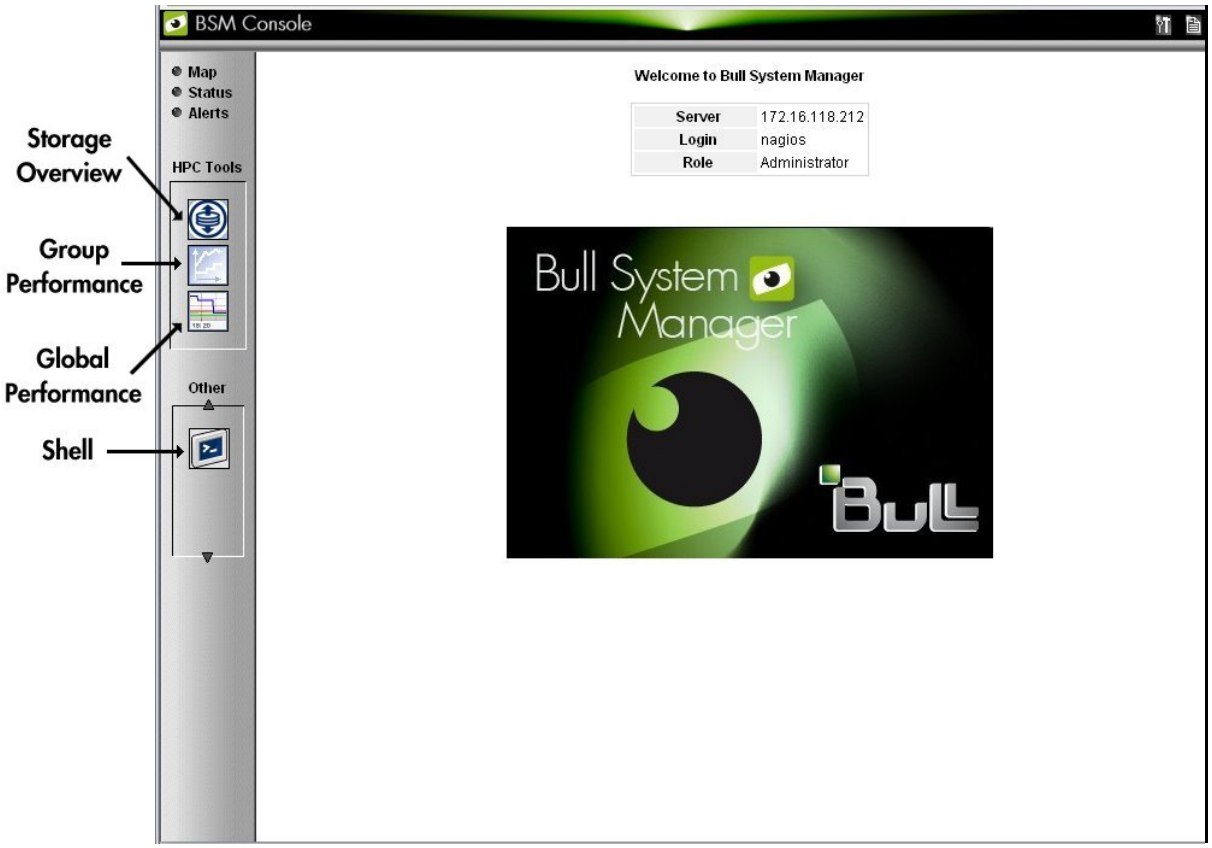


Figure 10-1. Bull System Manager - HPC Edition opening view

10.4.1 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 10-2* leads to the **Rack View** in *Figure 10-3*, which in turns leads to the more detailed **Services** view in *Figure 10-5*, for the host selected in the **Rack View**.

10.5 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.

10.5.1 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 10-2*.

Figure 10-2. **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 10-3*. This displays additional information, including its physical position and the services which are in a non-OK state.

10.5.2 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 10-3*.

Clicking on a component displays a detailed view for it.

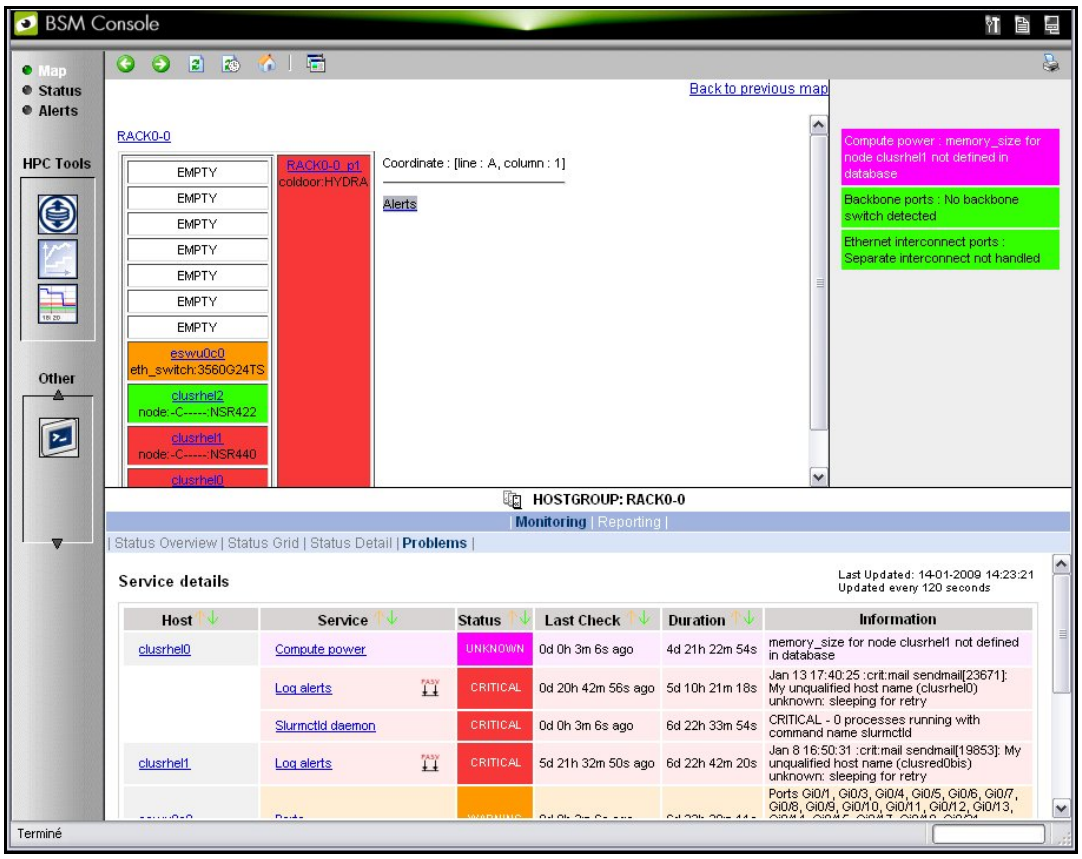


Figure 10-3. **Rack view** with the **Problems** window at the bottom

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 10-5*.

10.5.3 bullx blade map view

For nodes which include **bullx** blades the **Rack View** when selected will open the **bullx blade map view** - see *Figure 10-4*. This displays the individual blades in the **bullx blade chassis**. The Hardware Manager (**CMC**), shown as **hwn0** below, is displayed, and the **Ethernet/InfiniBand** switches are shown between the two rows of blades. Clicking on an individual blade will open the **Host Services View** - *Figure 10-5*, the same as for any other node.

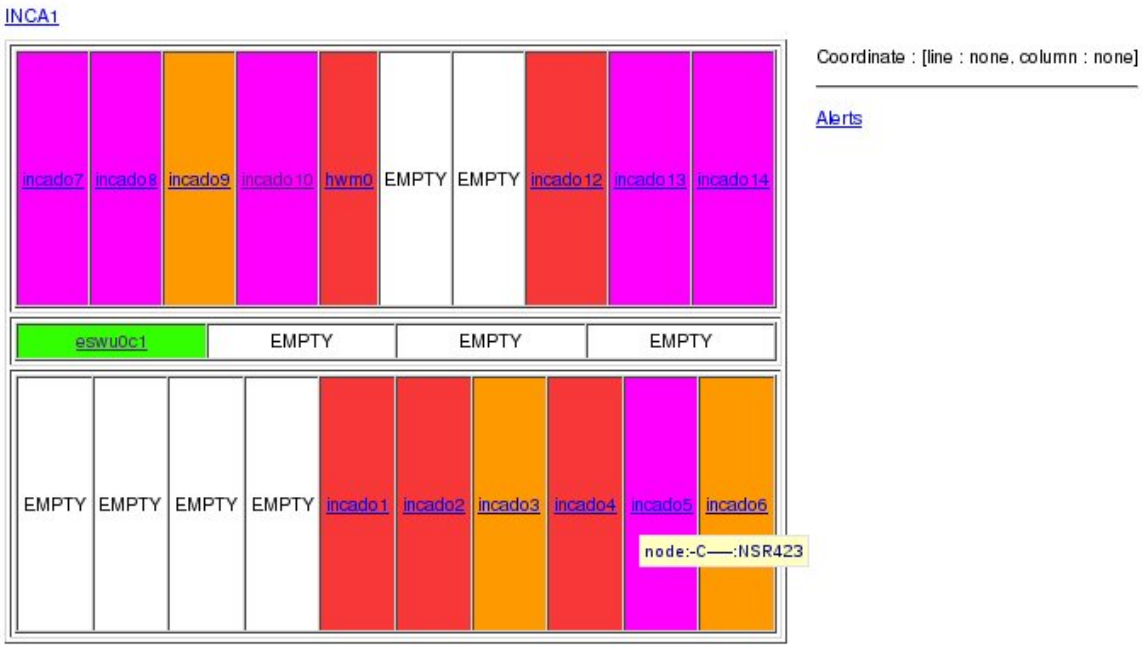


Figure 10-4. bullx blade map view

10.5.4 Host Services detailed View

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

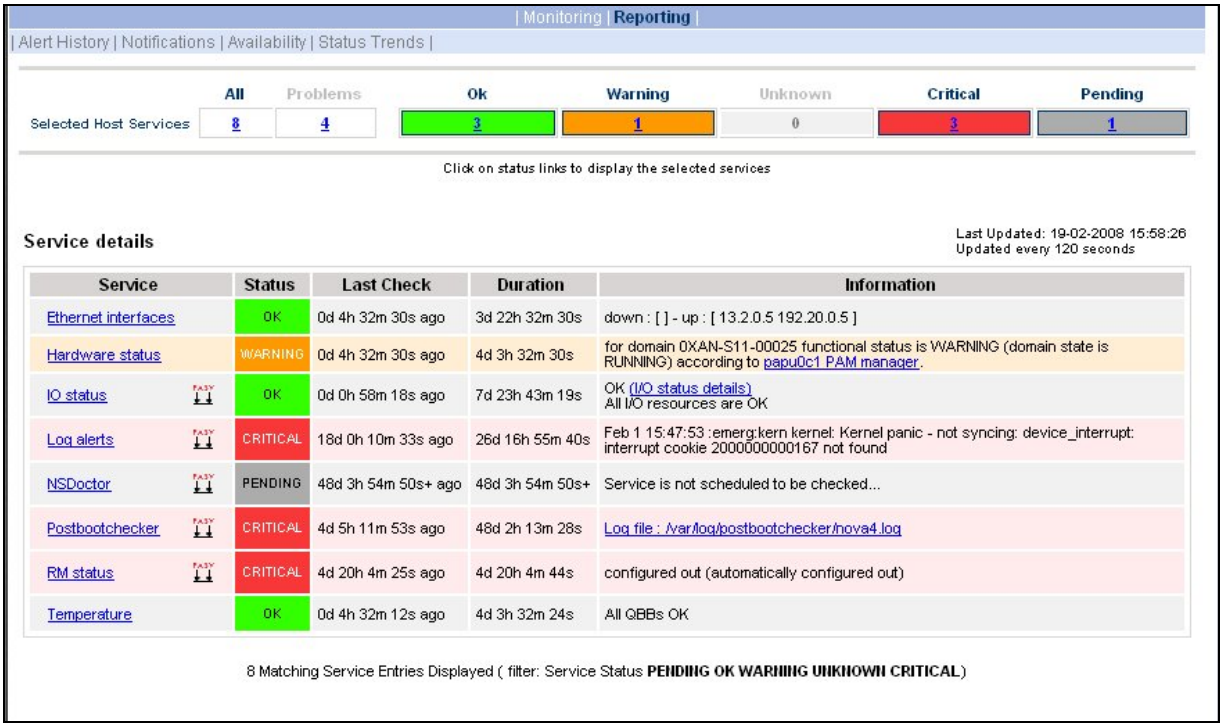


Figure 10-5. Host Service details

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

10.5.5 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 10-5*.

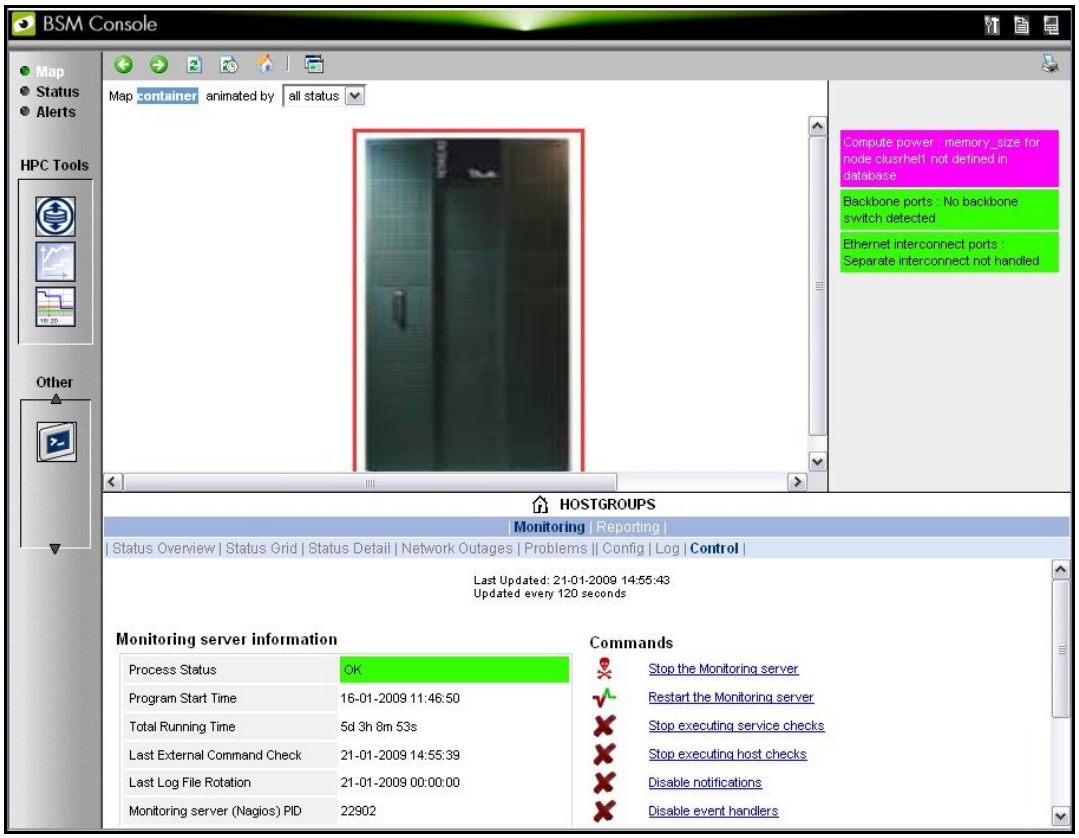


Figure 10-6. **Monitoring Control** Window

10.5.6 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.

10.6 Status Button

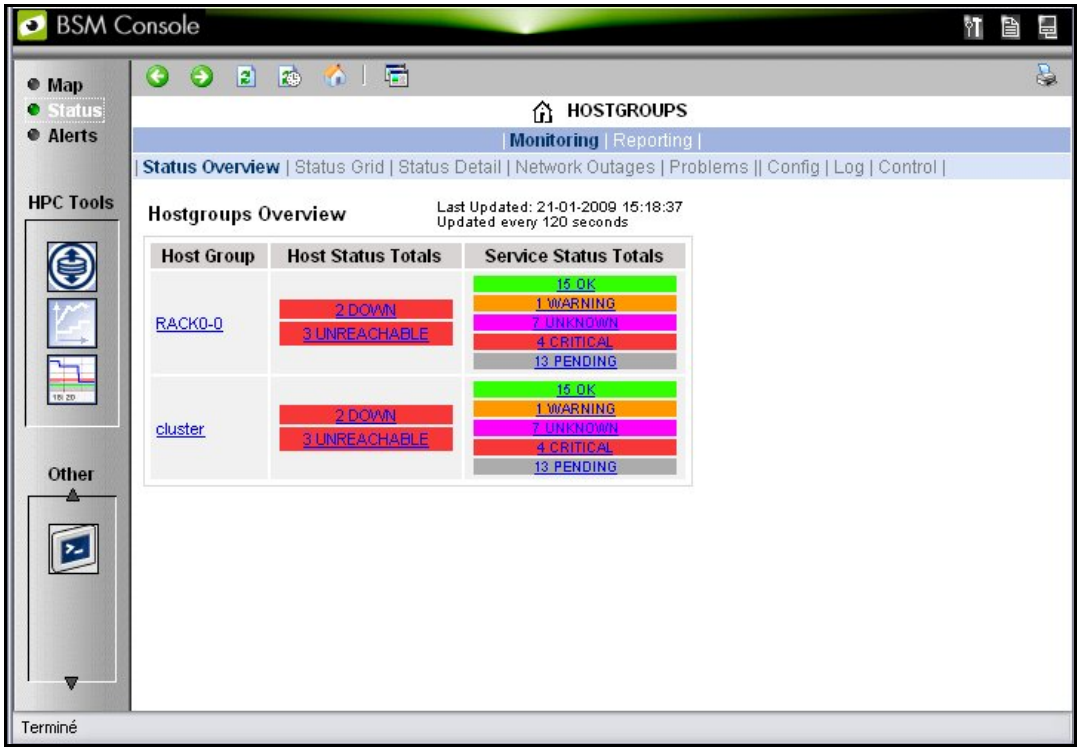


Figure 10-7. **Status Overview** screen

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 10-7. 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.

10.7 Log Window

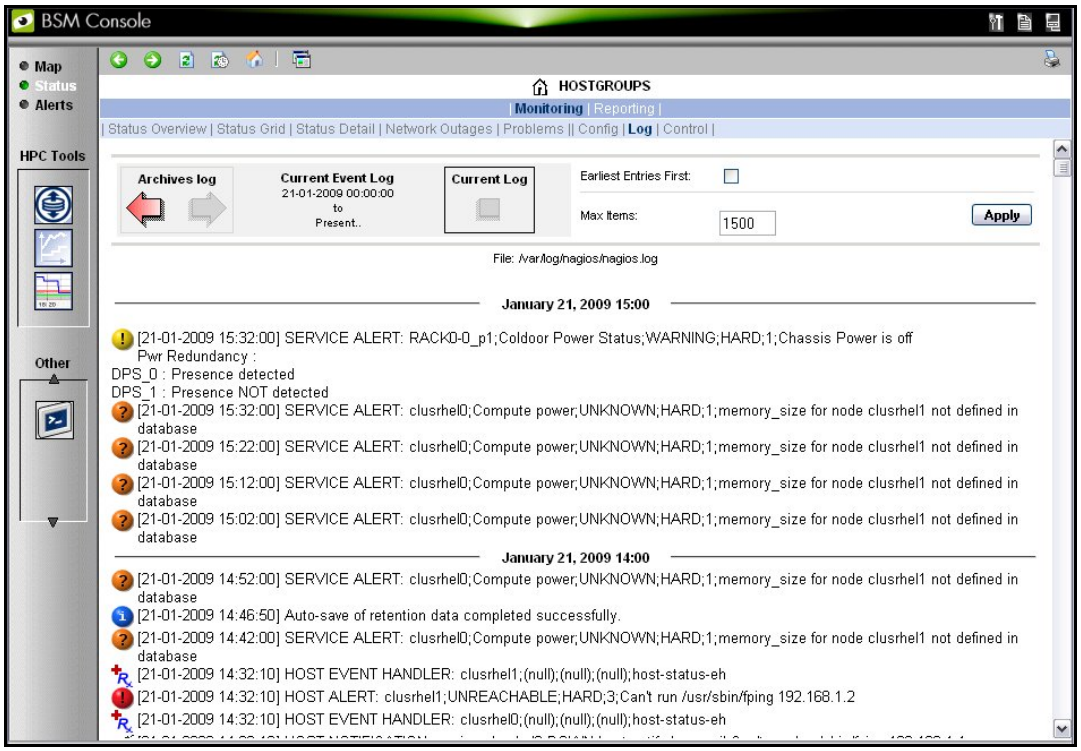


Figure 10-8. Monitoring - Log Window

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 10-8*.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.

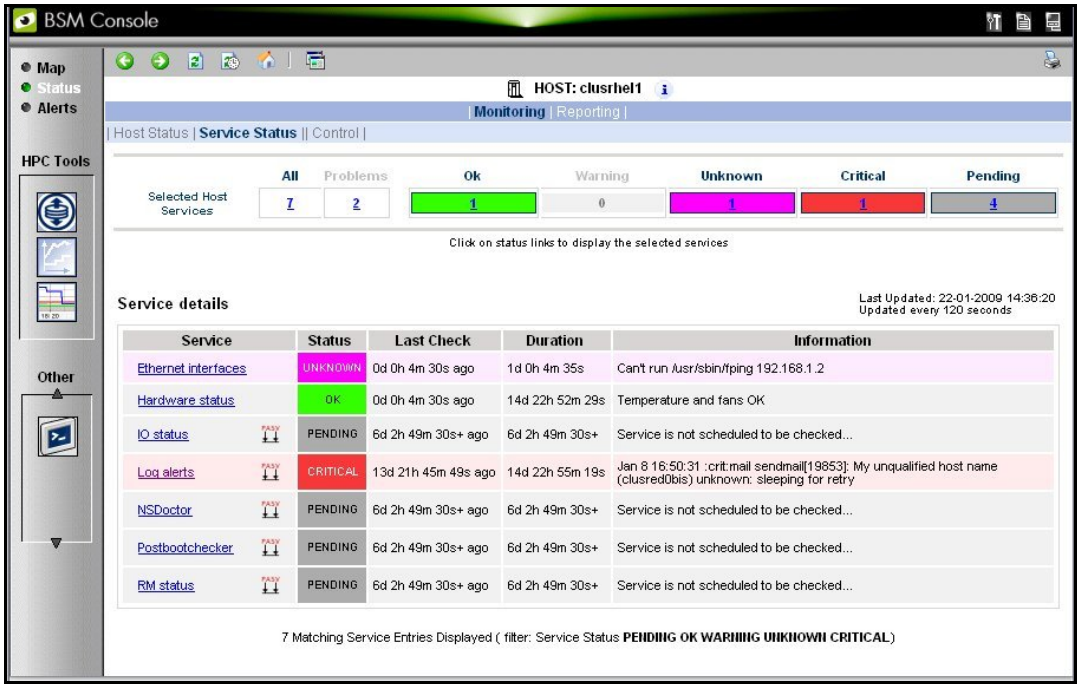


Figure 10-9. Monitoring Service Status window for a host with the Log Alerts link highlighted.

10.8 Alerts Button

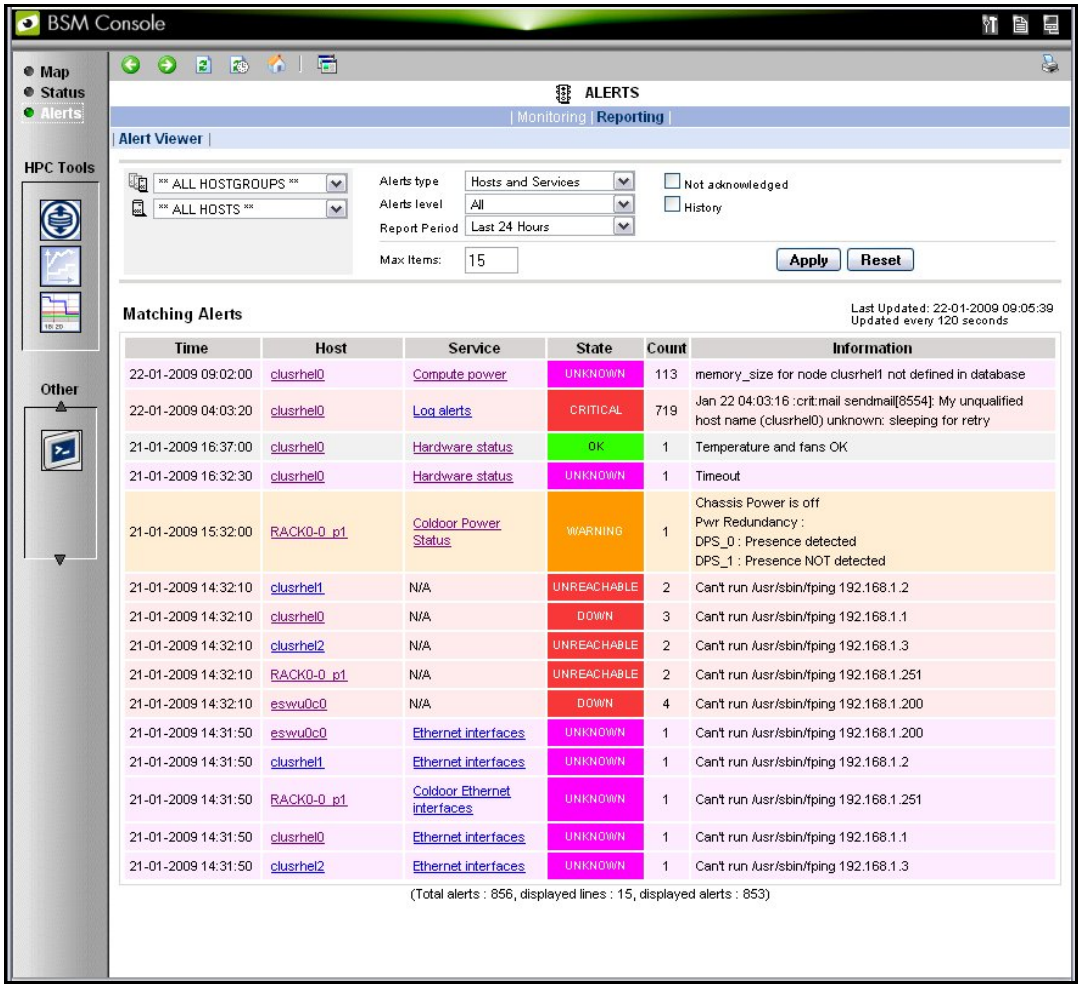


Figure 10-10. **Alert Window** showing the different alert states

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 10-10.*

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.

10.8.1 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.

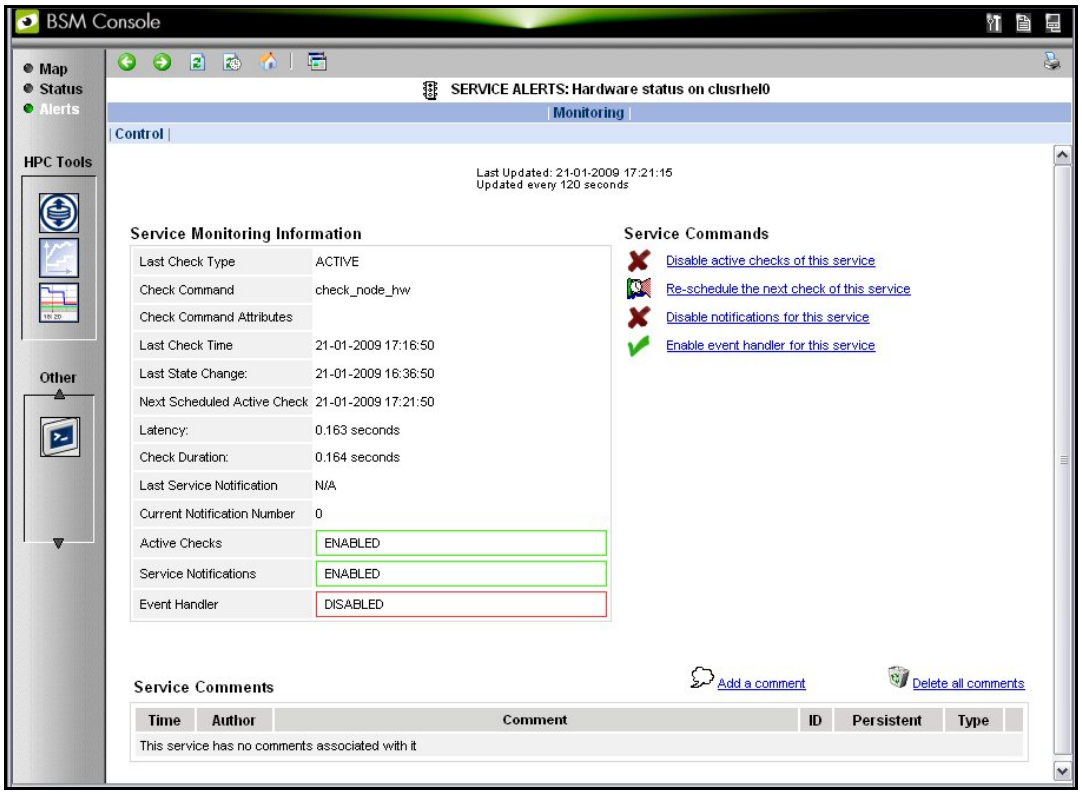


Figure 10-11. **Monitoring Control** Window used to set **Active Checks** for a Service

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 10-10* - 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.

10.8.2 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.

10.8.3 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**).

10.8.4 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 10-11* 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.

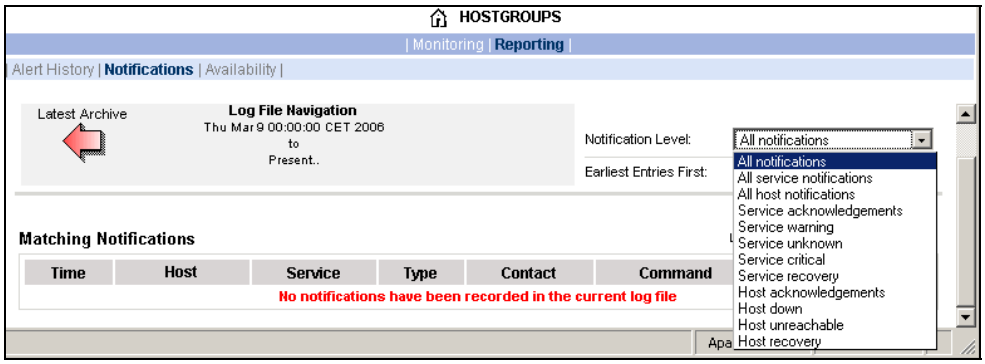


Figure 10-12. Hostgroups Reporting Notifications Window showing the Notification Levels

10.8.5 Acknowledgments

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

10.8.6 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.

10.8.7 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
```

10.8.8 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.

10.8.9 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.

10.8.10 Comments

Users of a particular host or service can post comments from the **Monitoring Control** window - see *Figure 10-11*

10.9 Storage Overview

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

Figure 10-13. Storage overview window

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

See *Chapter 6 – Storage Device Management* for information on **Bull System Manager - HPC Edition** and storage views.

10.10 Shell

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

10.11 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 10-1. 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.

10.12 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.

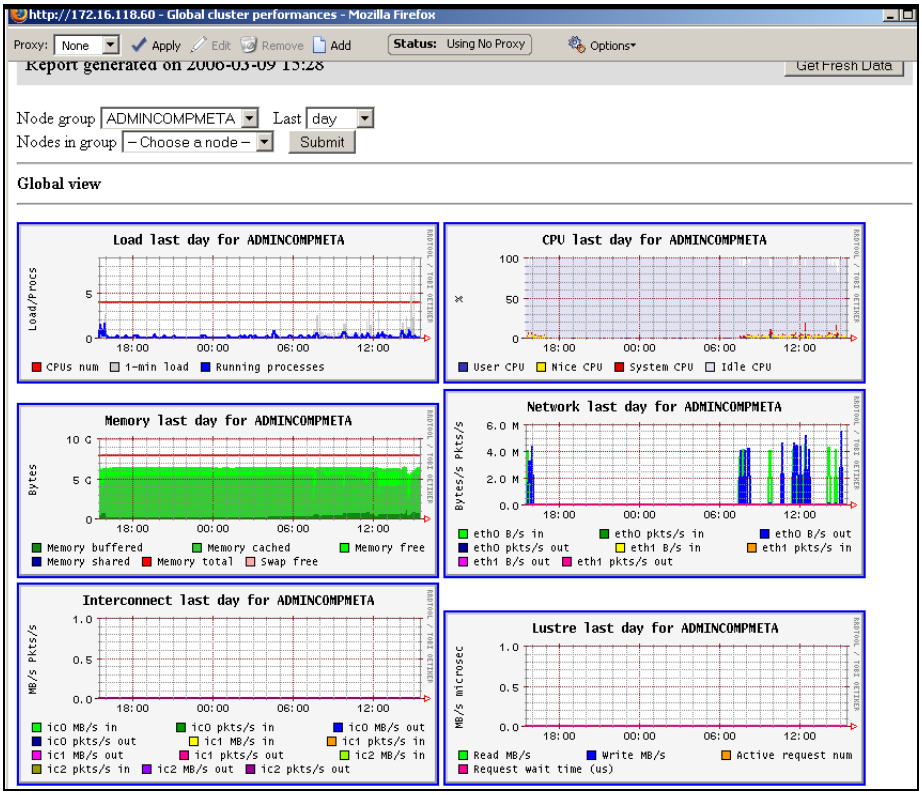


Figure 10-14. Group Performance view

10.13 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.

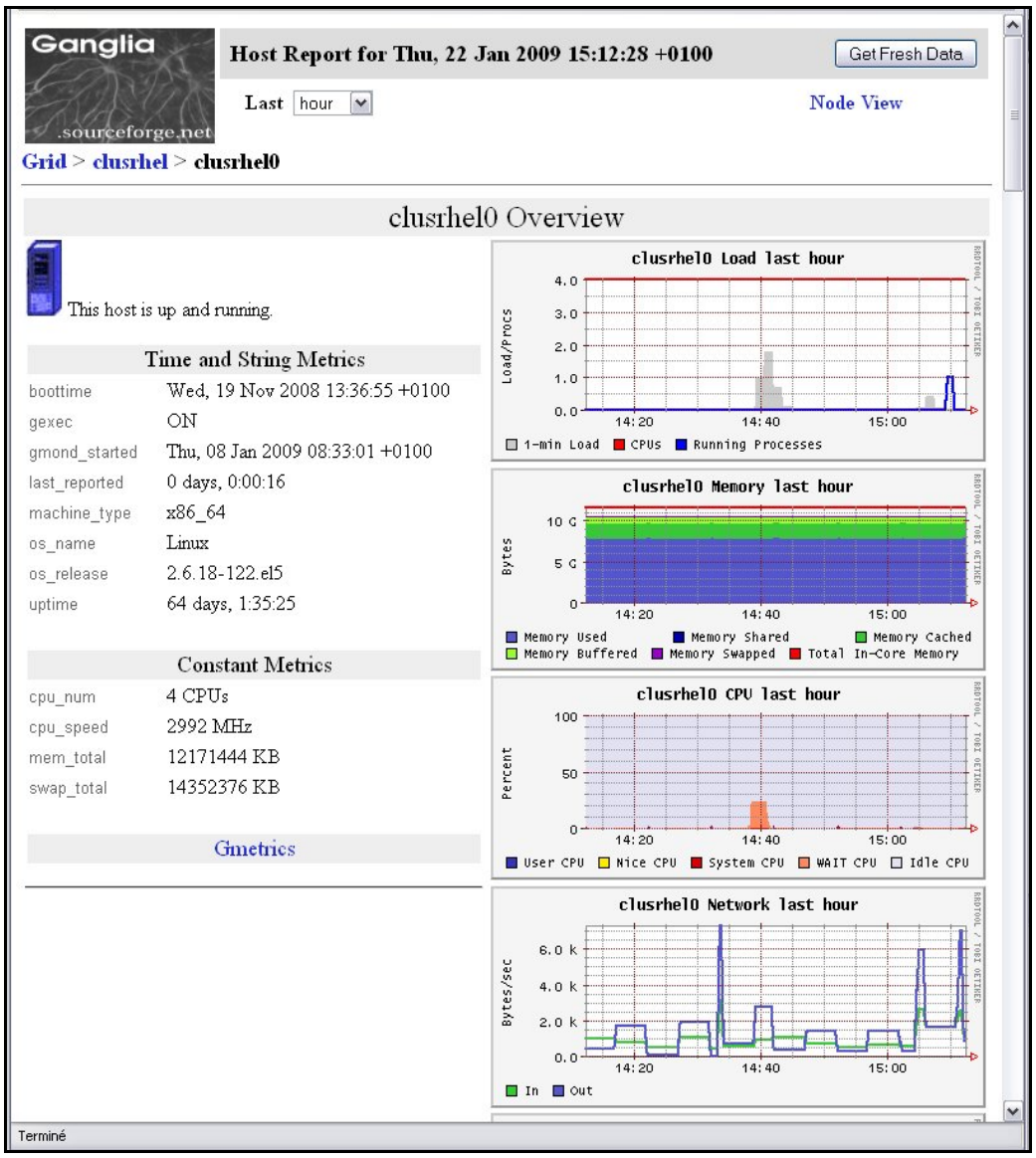


Figure 10-15. Global overview for a host (top screen)

More detailed views are shown by scrolling the window down – see Figure 10-16.

Figure 10-16. Detailed monitoring view for a host (bottom half of screen displayed in Figure 10-15)

10.13.1 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)
```

10.13.2 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**.

10.14 Configuring and Modifying Nagios Services

10.14.1 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
```

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.

10.14.2 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.

10.14.2.1 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.

10.14.3 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.

10.14.4 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**).

10.14.5 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*.

10.15 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.

10.15.1 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**.

10.15.2 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.

10.15.3 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.

10.15.4 Alert Log

The **Log alerts** passive service displays the last alarm raised by system log for the machine – see *Section 10.7* . 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.

10.15.5 I/O Status

The I/O status reports the global status of HBA, disks and LUNs on a cluster node.
Refer to Chapter 6, section *Monitoring Node I/O Status* for more information.

10.15.6 Postbootchecker

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

10.16 Management Node Nagios Services

These services are available on the Management Node only.

10.16.1 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.

10.16.2 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.

10.16.3 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.

10.16.4 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.

10.16.5 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 10-2.

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 – see section 3.3.6 in this guide for more information.

10.16.6 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 10-2.

10.16.7 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 10-2.

10.16.8 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 10-2.

10.16.9 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 10-2.

10.16.10 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 10-2.

10.16.11 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**.

10.16.12 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.

10.16.13 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.

10.16.14 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.

10.16.15 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).

10.16.16 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.

10.16.17 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 8-2.
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 *Maintenance Guide* for more information regarding the IBS tool.

10.16.18 CMC Health

This active service uses the **check_cmc.pl** plug-in to check if the temperatures and the fans are running correctly for **bullx** blades. It remains at the **OK** alert level while the data is in the right range, but switches to **CRITICAL** if the data is out of the right range. The **WARNING** alert level is displayed if no data is available.

10.17 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.

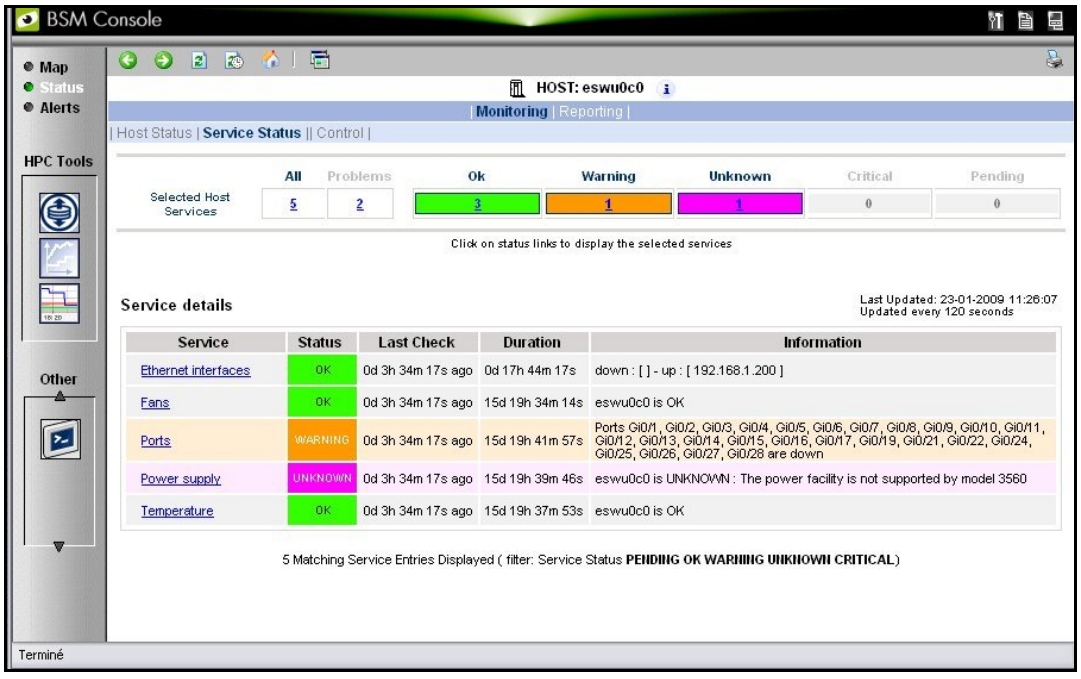


Figure 10-17. **Ethernet Switch** services

10.17.1 Ethernet Interfaces

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

10.17.2 Fans

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

10.17.3 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.

10.17.4 Power supply

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

10.17.5 Temperature

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

10.18 Cool Cabinet Door Services

Bull has developed a set of **Nagios** services to monitor the Cool Cabinet Door used to regulate the temperature for racks of servers. These are as follows:

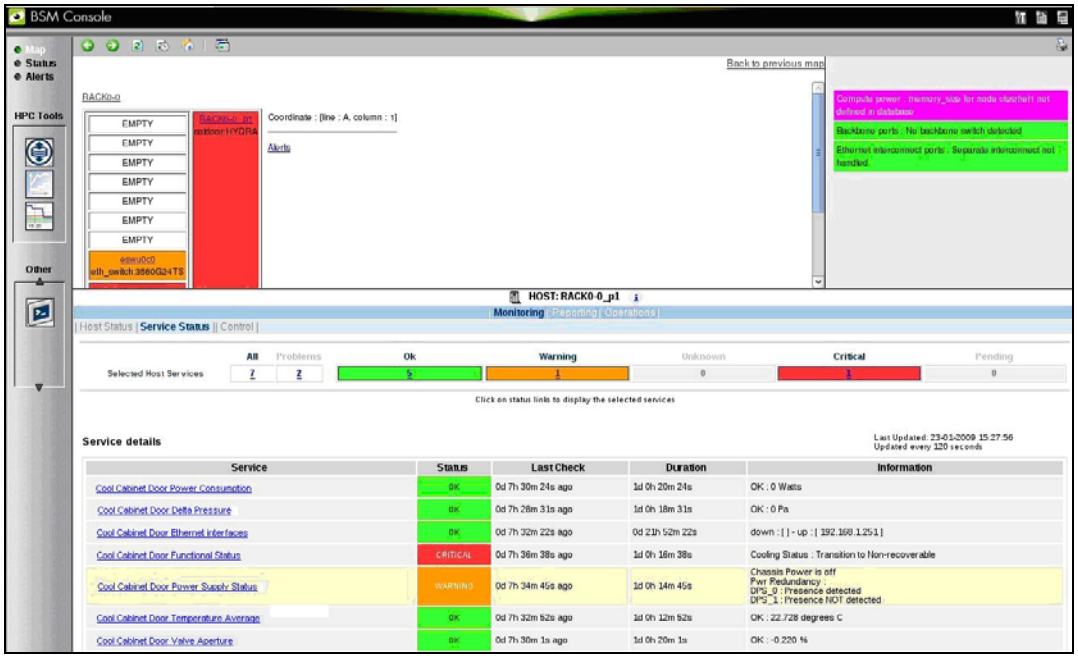


Figure 10-18. Cool Cabinet Door Services

10.18.1 Cool Cabinet Door Functional Status

The **Cool Cabinet Door Functional Status** service monitors the overall status of the cooled water door, with regard to temperature and water pressure through a **BSM CLI** (IPMI). The status for the service will be displayed as **CRITICAL,** if the value returned is **Transition to Critical** or **Transition to Non-recoverable**. The status for the service will be displayed as **WARNING** when the value returned is **Transition to Non-Critical**.

10.18.2 Cool Cabinet Door Power Consumption

The **Cool Cabinet Door Power Consumption** service monitors the power being used by the Cool Cabinet door through a **BSM CLI (IPMI)**. The status for the service will displayed as **CRITICAL,** if the value returned is ABOVE the **Upper-Critical** or **Upper Non-Recoverable** values, or if the value is BELOW the **Lower-Critical** or **Lower Non-Recoverable** values. A **WARNING** status is displayed, if the value or ABOVE the **Upper Non-Critical** value or is BELOW the **Lower Non-Critical** value.

10.18.3 Cool Cabinet Door Delta Pressure

The **Cool Cabinet Door Delta Pressure** service monitors the difference in water pressure between the water circulating inside the door and the water from the external water source through a **BSM CLI (IPMI)**. The status for the service will be displayed as **CRITICAL,** if the value is ABOVE the **Upper-Critical** or **Upper Non-Recoverable** values, or if the value is

BELOW the **Lower-Critical** or **Lower Non-Recoverable** values. It displays the **WARNING** state if the value is BELOW the **Lower Non-Critical** value or ABOVE the **Upper Non-Critical** value.

10.18.4 Cool Cabinet Door Ethernet Interfaces

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

10.18.5 Cool Cabinet Door Power Supply Status

The **Cool Cabinet Door Power Supply Status** service monitors the status of the power supply units, including the back-up units for the Cool Cabinet Door.

10.18.6 Cool Cabinet Door Temperature Average

The **Cool Cabinet Door Temperature Average** service monitors the average temperature of the water circulating within the Cool Cabinet Door through a **BSM CLI** (IPMI). The status for the service will be displayed as **CRITICAL,** if the value is ABOVE the **Upper-Critical** or **Upper Non-Recoverable** values, or if the value is BELOW the **Lower-Critical** or **Lower Non-Recoverable** values. It displays the **WARNING** state if the value is BELOW the **Lower Non-Critical** value or ABOVE the **Upper Non-Critical** value.

10.18.7 Cool Cabinet Door Valve Aperture

The **Cool Cabinet Door Valve Aperture** service monitors the degree (expressed as a percentage) for the valve opening for the water inlet through a **BSM CLI** (IPMI). The status for the service will be displayed as **CRITICAL,** if the value is ABOVE the **Upper-Critical** or **Upper Non-Recoverable** values, or if the value is BELOW the **Lower-Critical** or **Lower Non-Recoverable** values. It displays the **WARNING** state if the value is BELOW the **Lower Non-Critical** value or ABOVE the **Upper Non-Critical** value.

See The *Bull Cool Cabinet* documentation, listed in the *Bibliography* in the *Preface*, for more information regarding the Cool Cabinet Door.

Glossary and Acronyms

A

ABI
Application Binary Interface

ACL
Access Control List

ACT
Administration Configuration Tool

ANL
Argonne National Laboratory (MPICH2)

API
Application Programmer Interface

ARP
Address Resolution Protocol

ASIC
Application Specific Integrated Circuit

B

BAS
Bull Advanced Server

BIOS
Basic Input Output System

Blade
Thin server that is inserted in a blade chassis

BLACS
Basic Linear Algebra Communication Subprograms

BLAS
Basic Linear Algebra Subprograms

BMC
Baseboard Management Controller

BSBR
Bull System Backup Restore

BSM
Bull System Manager

C

CGI
Common Gateway Interface

CLI
Command Line Interface

ClusterDB
Cluster Database

CLM
Cluster Management

CMC
Chassis Management Controller

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

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

CUBLAS
CUDA™ BLAS

CUDA™
Compute Unified Device Architecture

CUFFT
CUDA™ Fast Fourier Transform

CVS
Concurrent Versions System

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

D

DDN
Data Direct Networks

DDR
Double Data Rate

DHCP
Dynamic Host Configuration Protocol

DLID
Destination Local Indentifier

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

DSO
Dynamic Shared Object

E

EBP
End Bad Packet Delimiter

ECT
Embedded Configuration Tool

EIP
Encapsulated IP

EPM
Errors per Million

EULA
End User License Agreement (Microsoft)

F

FDA
Fibre Disk Array

FFT
Fast Fourier Transform

FFTW
Fastest Fourier Transform in the West

FRU
Field Replaceable Unit

FTP
File Transfer Protocol

G

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

GCC
GNU C Compiler

GDB
Gnu Debugger

GFS
Global File System

GMP
GNU Multiprecision Library

GID
Group ID

GNU
GNU's Not Unix

GPL
General Public License

GPT
GUID Partition Table

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

GSL
GNU Scientific Library

GT/s
Giga transfers per second

GUI
Graphical User Interface

GUID
Globally Unique Identifier

H

HBA
Host Bus Adapter

HCA
Host Channel Adapter

HDD
Hard Disk Drive

HoQ
Head of Queue

HPC
High Performance Computing

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

I

IB
InfiniBand

IBTA
InfiniBand Trade Association

ICC
Intel C Compiler

IDE
Integrated Device Electronics

IFORT
Intel® Fortran Compiler

IMB
Intel MPI Benchmarks

INCA
Integrated Cluster Architecture:
Bull Blade platform

IOC
Input/Output Board Compact with 6 PCI Slots

IPMI
Intelligent Platform Management Interface

IPO
Interprocedural Optimization

IPoIB
Internet Protocol over InfiniBand

IPR
IP Router

iSM
Storage Manager (FDA storage systems)

ISV
Independent Software Vendor

K

KDC
Key Distribution Centre

KSIS
Utility for Image Building and Deployment

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

L

LAN
Local Area Network

LAPACK
Linear Algebra PACKage

LDAP
Lightweight Directory Access Protocol

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

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

LOV
Logical Object Volume

LSF
Load Sharing Facility

LUN
Logical Unit Number

LVM
Logical Volume Manager

LVS
Linux Virtual Server

M

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

MAD
Management Datagram

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

MDS
MetaData Server

MDT
MetaData Target

MFT
Mellanox Firmware Tools

MIB
Management Information Base

MKL
Maths Kernel Library

MPD
MPI Process Daemons

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

MPI
Message Passing Interface

MTBF
Mean Time Between Failures

MTU
Maximum Transmission Unit

N

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

NETCDF
Network Common Data Form

NFS
Network File System

NIC
Network Interface Card

NIS
Network Information Service

NS
NovaScale

NTP
Network Time Protocol

NUMA
Non Uniform Memory Access

NVRAM
Non Volatile Random Access Memory

O

OFA
Open Fabrics Alliance

OFED
Open Fabrics Enterprise Distribution

OPMA
Open Platform Management Architecture

OpenSM
Open Subnet Manager

OpenIB
Open InfiniBand

OpenSSH
Open Source implementation of the SSH protocol

OSC
Object Storage Client

OSS
Object Storage Server

OST
Object Storage Target

P

PAM
Platform Administration and Maintenance Software

PAPI
Performance Application Programming Interface

PBLAS
Parallel Basic Linear Algebra Subprograms

PBS
Portable Batch System

PCI
Peripheral Component Interconnect (Intel)

PDSH
Parallel Distributed Shell

PDU
Power Distribution Unit

PETSc
Portable, Extensible Toolkit for Scientific Computation

PGAPACK
Parallel Genetic Algorithm Package

PM
Performance Manager

Platform Management

PMI
Process Management Interface

PMU
Performance Monitoring Unit

pNETCDF
Parallel NetCDF (Network Common Data Form)

PVFS
Parallel Virtual File System

Q

QDR
Quad Data Rate

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

R

RAID
Redundant Array of Independent Disks

RDMA
Remote Direct Memory Access

ROM
Read Only Memory

RPC
Remote Procedure Call

RPM
RPM Package Manager

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

S

SA
Subnet Agent

SAFTE
SCSI Accessible Fault Tolerant Enclosures

SAN
Storage Area Network

SCALAPACK
SCALable Linear Algebra PACKage

SCSI
Small Computer System Interface

SCIPORT
Portable implementation of CRAY SCILIB

SDP
Socket Direct Protocol

SDPOIB
Sockets Direct Protocol over Infiniband

SDR
Sensor Data Record

Single Data Rate

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

SEL
System Event Log

SIOH
Server Input/Output Hub

SIS
System Installation Suite

SL
Service Level

SL2VL
Service Level to Virtual Lane

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

SM
Subnet Manager

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

SNMP
Simple Network Management Protocol

SOL
Serial Over LAN

SPOF
Single Point of Failure

SSH
Secure Shell

Syslog-ng
System Log New Generation

T

TCL
Tool Command Language

TCP
Transmission Control Protocol

TFTP
Trivial File Transfer Protocol

TGT
Ticket-Granting Ticket

U

UDP
User Datagram Protocol

UID
User ID

ULP
Upper Layer Protocol

USB
Universal Serial Bus

UTC
Coordinated Universal Time

V

VCRC
Variant Cyclic Redundancy Check

VDM
Voltaire Device Manager

VFM
Voltaire Fabric Manager

VGA
Video Graphic Adapter

VL
Virtual Lane

VLAN
Virtual Local Area Network

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

W

WWPN
World–Wide Port Name

X

XFS
eXtended File System

XHPC
Xeon High Performance Computing

XIB
Xeon InfiniBand

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

Index

/

/etc/krb5.conf, 5-2

/etc/lustre/storage.conf file, 7-8

/etc/nagios/contactgroups.cfg, 10-16

/etc/nagios/contacts.cfg, 10-16

/etc/nagios/snmptargets.cfg, 10-16

/etc/nsmhpc/nsmhpc.conf, 10-16

/etc/storageadmin/storframwork.conf, 6-34

/var/kerberos/krb5kdc/kadm5.acl, 5-4

/var/kerberos/krb5kdc/kdc.conf, 5-3

/var/log/postgres/pgsql, 3-26

/var/log/synchro.log file, 3-6

A

administrator
- postgres (ClusterDB), 3-2
- root, 2-2

authorized_keys2 file, 2-3

B

Backbone ports available alert, 10-25

Batch Management, 9-1

Bull System Manager - HPC Edition, 10-1
- Acknowledgements, 10-15
- Active checks, 10-13
- Alert definition, 10-14
- Alert levels, 10-13
- Alert types, 10-12
- All status map view, 10-6
- bullx blade map view, 10-7
- Changing passwords, 10-3
- Comments, 10-16
- Ganglia, 10-18
- Global Performance view, 10-19
- Group Performance view, 10-18
- Management node Nagios Services
- Map button, 10-6
- Monitoring performance, 10-18
- Nagios Alert log, 10-23
- Nagios Ethernet interfaces, 10-23
- Nagios IO Status, 10-23
- Nagios logs, 10-11
- Nagios plug-ins, 10-23
- Nagios postbootchecker, 10-23
- Nagios Services, 10-21
- Passive checks, 10-14
- Ping Map view, 10-9
- Rack view, 10-7
- Shell button, 10-18
- SNMP Alerts, 10-16
- Status Button, 10-11
- Storage overview, 10-17
- User password, 10-3

C

chkconfig command, 2-1

ClusterDB
- administrator (postgres), 3-2
- ChangeOwnerProperties, 3-3
- cluster features, 3-7
- Commands, 3-2
- dbmCluster command, 3-7
- dbmConfig, 3-5
- dbmDiskArray command, 3-22
- dbmEthernet command, 3-15
- dbmFiberChannel command, 3-20
- dbmHwManager command, 3-11
- dbmIconnect command, 3-16
- dbmNode command, 3-8
- dbmSerial command, 3-18
- dbmServices command, 3-21
- Description, 3-1
- managing groups, 3-12
- monitoring, 10-24
- PostgreSQL tools, 3-24
- requisite, 10-2
- save and restore, 3-24
- template files, 3-7

ClusterDB tables
- ADMIN table, 3-49
- AVAILABILITY table, 3-52
- CLUSTER table, 3-29
- CLUSTER_IPV table, 3-32
- CONFIG_CANDIDATE table, 3-50
- CONFIG_STATUS table, 3-51
- da_cfg_model table, 3-41

da_controller table, 3-38
da_enclosure table, 3-37
da_ethernet_port table, 3-39
da_fan table, 3-39
da_fc_port table, 3-38
da_io_path table, 3-40
da_iocell_component table, 3-40
da_power_fan table, 3-40
da_power_port table, 3-41
da_power_supply table, 3-39
da_serial_port table, 3-38
da_temperature_sensor table, 3-40
disk_array table, 3-37
disk_slot table, 3-37
ETH_EXTRALINK table, 3-34
ETH_SWITCH table, 3-30
ETH_VLAN table, 3-32
FC_NW table, 3-32
FC_SWITCH table, 3-33
GROUP_NODE table, 3-51
HWMANAGER table, 3-48
IC_BOARD table, 3-46
IC_NW table, 3-30
IC_SWITCH table, 3-31
IP_NW table, 3-29
IPOIB table, 3-46
Lustre_fs table, 3-54
Lustre_IO_node table, 3-55
Lustre_MDT table, 3-55
Lustre_mount table, 3-56
Lustre_OST table, 3-55
MSG_SYSLOG table, 3-51
Node table, 3-44
Node_image table, 3-45
Node_profile table, 3-45
PORTSERVER table, 3-32
RACK table, 3-50
RACK_PORT table, 3-50
SDPOIB table, 3-46, 3-47
SERIAL_NW table, 3-31
SERVICES table, 3-51
TALIM table, 3-34

Commands
ChangeOwnerProperties, 3-3
chkconfig, 2-1
dbmCluster, 3-7
dbmConfig, 3-5
dbmDiskArray, 3-22
dbmEthernet, 3-15
dbmFiberChannel, 3-20
dbmGroup, 3-12
dbmHwManager, 3-11
dbmIconnect, 3-16
dbmNode, 3-8
dbmSerial, 3-18
dbmServices, 3-21
dbmTalim, 3-17
ddn_ set_up_date_time, 6-24
ddn_admin, 6-23
ddn_check, 6-24
ddn_conchk, 6-24
ddn_firmup, 6-25
ddn_init, 6-24
ddn_stat, 6-23
dshbak, 2-4
iorefmgmt, 6-5
kadmin, 5-3
lfs quotacheck, 7-38
lfs setquota, 7-38
lsiodev, 6-4
lustre_investigate, 7-16
lustre_tables_dba, 7-6
lustre_util, 7-22
nec_admin, 6-22
passwd, 2-2
pbsnodes, 9-3
pdcp, 2-4
pdsh, 2-4
qdel, 9-3
qstat, 9-3
qsub, 9-3
stormodelctl, 6-31
storstat, 6-2, 6-17
Tracejob, 9-3
useradd, 2-2

connectivity status, 6-12

contact groups
adding, 10-16

contacts
adding, 10-16

controller status, 6-11

D

dbmCluster command, 3-7

dbmConfig command, 3-5

dbmDiskArray command, 3-22

dbmEthernet command, 3-15

dbmFiberChannel command, 3-20
dbmGroup command, 3-12
dbmHwManager command, 3-11
dbmIconnect command, 3-16
dbmNode command, 3-8
dbmSerial command, 3-18
dbmServices command, 3-21
dbmTalim command, 3-17
DDN commands, 6-23
ddn_ set_up_date_time command, 6-24
ddn_admin command, 6-23
ddn_check command, 6-24
ddn_conchk command, 6-24
ddn_firmup command, 6-25
ddn_init command, 6-24
ddn_stat command, 6-23
deploying software *See* Ksis
distributed shell, 2-4
distribution
  changing, 4-1
  updating, 4-1
distribution software, 4-1
dropdb command, 3-25
dshbak command, 2-4

F

fan status, 6-10
file system
  parallel, 7-1
  striping, 7-1
files
  /etc/lustre/storage.conf, 7-8
  /etc/nagios/contactgroups.cfg, 10-16
  /etc/nagios/contacts.cfg, 10-16
  /etc/nagios/snmptargets.cfg, 10-16
  /etc/nsmhpc/nsmhpc.conf, 10-16
  /var/log/synchro.log, 3-6
  authorized_keys2, 2-3
  genders, 2-5
  id_dsa.pub, 2-3
  kadm5.acl, 5-4
  lustre.cfg, 7-13
  lustre_util.conf, 7-33
  res_rpm_qsnetmpi, 2-8
  storframework.conf, 6-34
  template.model, 6-29
  tuning.conf, 7-35

G

Ganglia
  data categories, 10-19
Ganglia
  Bull System Manager - HPC Edition, 10-1
genders file, 2-5
groups of nodes, 3-13

H

HDD status, 6-10
HPC Toolkit, 1-2

I

id_dsa.pub file, 2-3
image
  list, 3-8
InfiniBand links available, 10-26
iorefmgmt command, 6-5

J

JobCredentialPrivateKey, 8-16
JobCredentialPublicCertificate, 8-16

K

Kerberos, 1-2, 2-4, 5-1
  Access Control List, 5-4
  Admin Daemon, 5-4
  configuration files, 5-2
  database, 5-3
  Host principal, 5-5
  kadmin command, 5-3
  KDC, 5-1
  package, 5-2
  SSH, 5-7
  TGT ticket, 5-7

Kerberos admin daemon, 10-26

Kerberos KDC daemon, 10-25

Ksis
builddatanode command, 4-13
check command, 4-4, 4-12
checkdiff command, 4-6, 4-12
checks database, 4-6
client node, 4-3
command file, 4-6
command options, 4-7
create commands, 4-8
delete command, 4-9
deploy command, 4-9
help command, 4-7
image server, 4-1, 4-3
import command, 4-12
Ksis server, 4-1
list command, 4-9
nodelist command, 4-10
nodeRange, 4-7
overview, 1-2, 4-1
reference node, 4-3
reference/golden image, 4-1, 4-2
undeploy command, 4-9

L

LDAP daemon, 10-26

linux user, 2-2

LOV (Logical Object Volume), 7-2

lsiodev command, 6-4

Lustre, 7-2
administrator tasks, 7-3
Bull System Manager monitoring, 7-40
Creating File systems, 7-17
database, 7-6
Extended model file, 7-20
Installing Lustre file systems, 7-22
lfs quotacheck, 7-38
load_storage.sh, 7-12
lustre.cfg file, 7-13
lustre_check tool, 7-41
lustre_investigate command, 7-16
lustre_storage_config.sh, 7-9
lustre_util, 7-22
lustre_util.conf file, 7-33
Management Node interface, 7-43
model file, 7-18
Monitoring, 7-40
Nagios filesystem indicator, 7-42
networks, 7-13
NovaScale Group Performance view, 7-44
NovaScale Node Performance view, 7-46
planning, 7-4
Quota settings, 7-37
Rescuing a file system, 7-39
Services, 7-16
Setting limits, 7-38
striping, 7-5
system limitations, 7-5
tuning.conf file, 7-35

Lustre filesystems access, 10-26

Bull System Manager, 7-40

lustre.cfg file, 7-13

M

maintenance tools, 2-9

MDS (MetaData Server), 7-2

MDT (MetaData Target), 7-2

MetaData Server migration alert, 10-25

MiniSQL daemon, 10-24

model
file, 6-29
storage system configuration, 6-28

monitoring the cluster, 10-1

N

Nagios
Contact groups, 10-4
Hosts, 10-8
Services, 10-4, 10-8

Nagios
Bull System Manager - HPC Edition, 10-1

Nagios Management node plug-ins
ClusterDB, 10-24
Cron Daemon, 10-24
MiniSQL Daemon, 10-24

Nagios plug-ins
Backbone ports available, 10-25
Ethernet Switch services, 10-28, 10-30
HA system status, 10-25
InfiniBand links available, 10-26

Kerberos admin daemon, 10-26
Kerberos KDC daemon, 10-25
LDAP daemon, 10-26
Lustre filesystems access, 10-26
Metadata servers, 10-25
NFS filesystems access, 10-26

NameSpace, 7-2

nec_admin command, 6-22

nec_admin.conf file, 6-22

NFS filesystems access, 10-26

node list, 3-8

O

oid2name command, 3-25

OpenSSH, 2-3

oppensl, 8-16

OSC (Object Storage Client), 7-2

OSS (Object Storage Server), 7-2

OST (Object Storage Target), 7-2

P

parallel commands, 2-4

passwd command, 2-2

password
user, 2-2

PBS Professional, 1-2

PBS Professional Batch Manager, 9-1
Commands, 9-3
Daemons, 9-2
Ethernet, 9-4
GridWorks Analytics, 9-3
InfiniBand, 9-4
MPIBull2, 9-4

pdcp command, 2-4

pdsh, 1-2

pdsh command, 2-4

pg_dump command, 3-24

pg_restore command, 3-24

phpPgAdmin, 3-24

pipeline (data), 7-4

postbootchecker, 10-23

postgres user, 3-2

PostgreSQL, 3-24

power supply status, 6-10

predefined groups, 3-13

psql command, 3-24

R

res_rpm_qsnetmpi file, 2-8

resource management, 1-2, 8-1

root user, 2-2

rsh, 2-4

S

security
Kerberos, 5-1
policies, 2-3

service
list, 2-1
star), 2-1

shell
distributed, 2-4
kerberos, 2-4
pdsh, 2-4
rsh, 2-4
ssh, 2-4

SLURM, 1-2, 8-1
Draining a node, 8-17
Functions, 8-2
NodeAddr, 8-7
NodeHostname, 8-7
NodeName, 8-7
pam_slurm, 8-8
sacct command, 8-2
sacctmgr command, 8-2
salloc command, 8-2
sattach command, 8-2
sbatch command, 8-2
scancel command, 8-2
SCONTROL, 8-2, 8-8
Scontrol examples, 8-16
SelectType configuration parameter, 8-15
sinfo command, 8-2
slurm.conf, 8-7

SLURMCTLD Controller daemon, 8-12, 8-13
SLURMCTLD daemon, 8-2, 8-3
SLURMD, 8-2, 8-5
SLURMD Compute node daemon, 8-12, 8-14
squeue command, 8-2
srun command, 8-2
strigger command, 8-2
sview command, 8-2

SLURM and openssl, 8-16

SLURM and Security, 8-16

SLURM and syslogr, 8-15

SNMP trap
response to alert, 10-16

software distribution, 4-1

software update, 4-1

ssh, 2-4
setting up, 2-3

storage device
configuration deployment, 6-3
configuration files, 6-33
configuration planning, 6-27
management services, 6-2
managing, 6-1
monitoring, using Nagios, 6-8

stormodelctl command, 6-31

storstat command, 6-2, 6-17

system image, 3-8

system status, 6-12

T

temperature status, 6-11

template.model file, 6-29

U

user
create, 2-2
password, 2-2

useradd command, 2-2

V

view
inventory of storage systems and components, 6-15
storage, 6-13
storage tactical overview, 6-13

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

REFERENCE
86 A2 20FA 02