

BDUHPC- HPC CLUSTER
Bharathidasan University
Trichy
USER MANUAL



Submitted by:



System Integration Team
Wipro InfoTech
Bangalore, India

Confidentiality

This document is a confidential document of Wipro Ltd. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording or otherwise, without the written permission of Wipro Infotech. This document includes confidential information related to Wipro and shall not be distributed to any persons other than those mentioned in the distribution list herein.

Major Stake Holders

Sponsor	Bharathidasan University
Sponsor Organization	Bharathidasan University
Project Manager(s)	Mr. Arun Chaudhary - Wipro Mr. Satish Kumar – Wipro
Project Contact(s)	Mr. Satish Kumar - Wipro Mr. Upamanyu Ghosh – Wipro
Email Address	satish.vaddi1@wipro.com upamanyu.ghosh@wipro.com
File Name	BDUHPC HPC CLUSTER User MANUAL

Revision History

Version	Date	Author	Comment
V1.0	22-September-2015	Upamanyu Ghosh	Original Version

Distribution

Company/Dept.	Name	Function in Project
Bharathidasan University, Trichy	S.Madhavan	Customer
WIPRO	Mr. Arun Choudhary Mr. Satish Kumar	Project Manager

Table of Contents

1.Introduction.....	4
1.1 Cluster Architecture Overview.....	5
1.2 Grid Architecture Overview	6
1.3 Parallel File System Overview	7
2. System Overview.....	9
2.1 Setup	9
2.2 Cluster Description.....	11
2.3 Schematic Diagram	12
2.4 Server Specification Details	13
3. Software Packages	15
4. HPC.....	16
4.1 Man Made Cluster	16
4.2 Configure Storage V3700	17
5. Cluster Management	18
5.1 NIS	18
6. Scheduler - SLURM	19
6.1 SLURM Environment	20
6.2 SLURM Job Submission using Command line.....	26
6.3 Job Monitoring	28
7. How to connect Hpc cluster.....	35
8. Wipro HPC Portal	37
9. Cluster Status Monitoring Ganglia	43
10. OS Basic Commands	46
11. Call logging procedure with Wipro.....	47

1. Introduction:

High Performance Computing

HPC is broadly defined as the technology that is used to provide solutions to problems that need:

- ☐ Significant computational power
 - ☐ To quickly access and process large amounts of data
 - ☐ To operate interdownly across a geographically distributed network

The goal of HPC is to reduce the execution time of a compute intensive or data intensive application

Area includes:

- ☐ High Performance Compute Clusters
- ☐ Deployment
- ☐ Management
- ☐ High end Interconnects
- ☐ Job schedulers
- ☐ Compiler and performance libraries
- ☐ Application porting & optimization
- ☐ Parallel file system

This document summarizes the Hardware, OS, Tools, Execution Command and management / maintenance Commands HPC Cluster.

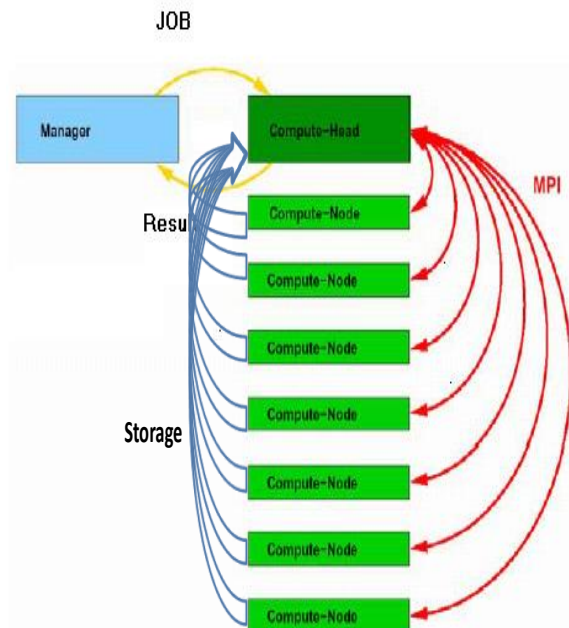
This document is meant to be for experienced personals with a firm understanding and working knowledge of HPC and its components.

This is a specific description of how Cluster is setup and includes not only details the compute aspects, but also the handling and troubleshooting minor defects that may arise while in use, in future.

This document describes in details the hardware and software configurations of Linux computing clusters installed for running applications.

1.1 Cluster Architecture Overview

A Cluster comprises of multiple compute servers connected together over a high speed interconnect. A Large mesh is submitted to the Master Server, wherein it is broken into smaller sub-domains and each sub-domain is submitted to a different Processor for computing. The JOB Decomposition is done within the application (Cluster Version), it is migrated to various Processors using the System Middleware and Low latency interconnects.



Key Advantages

- Massive computing power
- Dedicated Memory to Processor Performance
- Scalability to hundreds of Processors
- Excellent Price to Performance

1.2 Grid Architecture Overview

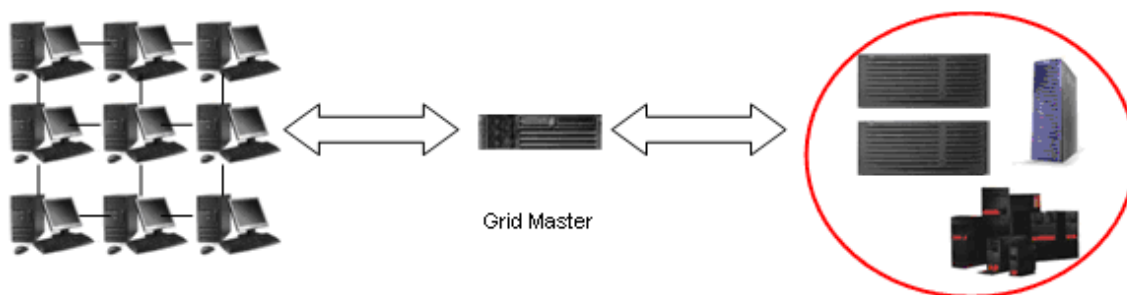
The Grid Engine provides policy-based workload management and dynamic provisioning of application workloads. It creates a grid of network-connected servers, workstations, and desktops; provides user access to the grid; and provides administrative and management interfaces.

The Utilization of Computing Resources and investments in software, hardware and man hours are generally utilized to the extent of 25-40% for a very well managed organization without the use of Workload Management Software. The Use of Workload Management Software provides an immediate benefit by enhancing the overall utilization to nearly 75% or more, thereby almost doubling your organization's ROI. Computing tasks or jobs are distributed across the grid in accordance with resource requirements for the job, user requests for the job, and administrative/managerial policies.

Usage accounting data is stored and made available so that it is possible to determine what resources were used in the execution of a job, and for whom the job was run. Through the pooling of departmental resources into larger enterprise grids, multiple users, teams, and departments can share common resources while

working on different projects with different goals and schedules, providing maximum resource availability to all users in a flexible, policy-based environment. Productivity can be dramatically increased compared to pre-grid approaches.

Through the ability to incorporate hundreds or thousands of Unix/Windows desktops and servers using a Grid Engine, grids will enable enterprises to increase significantly the utilization of compute resources for increased productivity in more powerful grids which are easier to install, operate and manage.



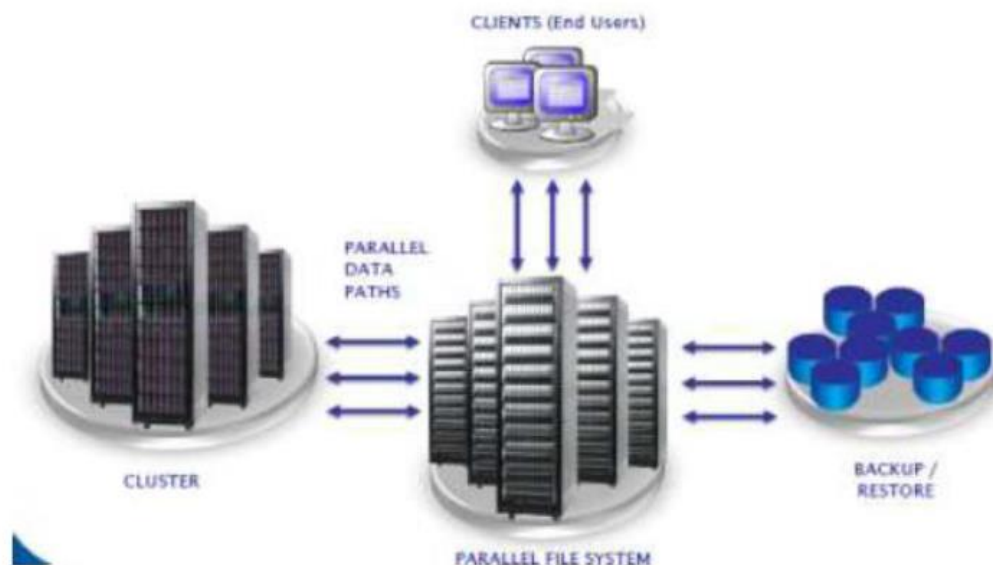
1.3 Parallel file system Overview

In general, a parallel file system is one in which data blocks are striped, in parallel, across multiple storage devices on multiple storage servers. This is similar to network link aggregation in which the I/O is spread across several network connections in parallel, each packet taking a different link path from the previous. Parallel file systems place data blocks from files on more than one server and more than one storage device

The leading data access protocol for batch computing is currently Network File System (NFS), but even with bigger, faster, more expensive Network Attached Storage (NAS) hardware available, batch processing seems to have an insatiable appetite for I/O operations per second.

As the available servers and available storage devices are increased, throughput can easily be doubled or tripled, given enough network connections on the clients. As a client application

requests data I/O, each sequential block request potentially can be going to an entirely different server or storage device.



Advantages:

1. Linear scaling
2. Extreme BW & I/O
3. Single namespace
4. Ease of management
5. Simple installation
6. Lower cost

2. System Overview

2.1 Setup :

Master Node: 1 x IBM X3550M4 1U Rack server with dual Intel Xeon E5-2670 V2 10c 2.5 GHz processors, 128 GB of Memory, 4 x 300 GB 10 K RPM SAS hard disk drives, redundant Power supplies and GigE and FDR Infiniband ports.

Compute Node : 10 x IBM x3550 M4 1U server nodes with dual Intel Xeon E5-2670 V2 10c 2.5 GHz processors, 64 GB of Memory, 2 x 500 GB 7.2 K RPM SATA hard disk drives, redundant Power supplies, Infiniband HBA and GigE ports.

Storage Nodes : 2 x IBM X3650 M4 2U Rack server with dual Intel E5-2660 V2 10c 2.2 GHz processors, 64 GB of Memory, 2 x 500 GB 7.2 K rpm SATA hard disk drives, redundant Power supplies, Dual port FC adapters, Infiniband HBA and GigE ports.

Storage : 18 TB IBM Storwize V3700 configured with 2 storage controllers, Redundant Power supply units, with 7 x 4TB 7.2K RPM NL-SAS Disks.

Infiniband : 1 x Mellanox SX6036 FDR InfiniBand Switch 36 QSFP non blocking ports , dual Management Modules.

Ethernet : 1 x Edge-Core ECS4110-52T 1GbE Switch for Cluster Management

KVM Switch : 1 x Exprezo KVM Switch.

RACK : 1 x 42U IBM RACK with 4 Horizontal PDUs.

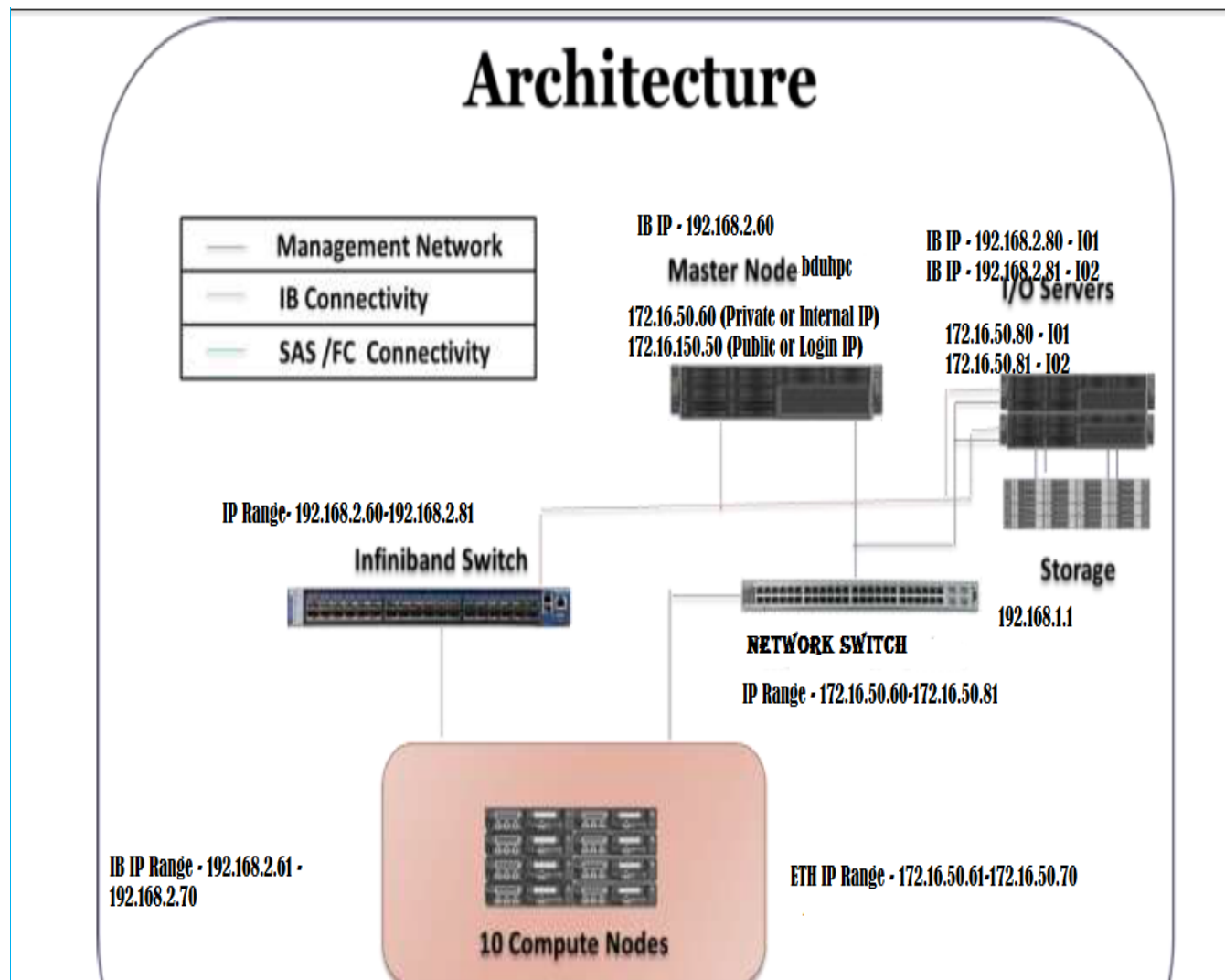
Setup Information

Type of Node	Quantity	CPU	Memory
Master (IBM X3550 M4)	1	2 X Intel Xeon E5-2670 V2 10c 2.5GHz	128 GB
Compute Node (IBM X3550 M4)	10	2 X Intel Xeon E5-2670 V2 10c 2.5 GHz	64 GB
Storage Nodes (IBM X3650 M4)	2	2 X Intel Xeon E5-2660 V2 10c 2.2 GHz	64 GB
Storage	1 Set	NIL	NIL
Mellanox SX6036 FDR IB Switch	1	NIL	NIL
48 port Network Switch	1	NIL	NIL
42U Rack	1	NIL	NIL
KVM Switch	1	NIL	NIL

Storage Information :

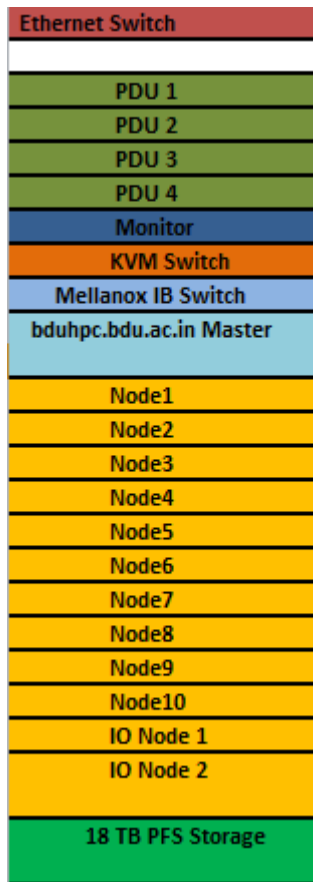
Purpose	Usable Capacity	Type of Disk	No. of Disks	RAID Layout
OST	18 TB	4TB 7.2K RPM NL-SAS Disks	7	RAID 5

2.2 Cluster Description :



- BDU Users login to bduhpc cluster via Master Node

2.3 Schematic Diagram :



BDU Cluster is built of Man Made cluster having CentOS-6.5 operating system!

2.4 Server Specification Details :

Information of all the nodes and their configurations:

Details	Master node	Compute nodes	IO nodes
CPU Type	Intel Xeon E5-2670v2 @2.5 GHz	Intel Xeon E5-2670v2 @2.5 GHz	Intel Xeon E5-2660v2 @2.2 GHz
GPU Type	Not Applicable	Not Applicable	Not Applicable
No of CPUs/ Cores	2/10	2/10	2/10
No of GPU / node	Not Applicable	Not Applicable	Not Applicable
RAM	128GB DDR3	64GB DDR3	64GB DDR3
HDD	4 x300GB SAS hard disk	2 x 500GB SATA hard disk	2 x 500GB SATA hard disk
OS	CENTOS 6.5 64 bit	CENTOS 6.5 64 bit	CENTOS 6.5 64 bit

Partition Information

Partitions	Master Node	Compute Node
/	630 GB	260 GB
Swap	104 GB	104 GB
/boot	500 MB	500 MB
/boot/efi	500 MB	500 MB
/storage	6.1 TB (Mounted from IONode1)	6.1 TB (Mounted from IONode1)
/home	10 TB (Mounted from IONode1)	10 TB (Mounted from IONode1)
/opt	2 TB (Mounted from IONode1)	2 TB (Mounted from IONode1)
/var	100 GB	100 GB

IP of HPC Servers:

HOST NAME	IP / Subnet – eth2 (PRIVATE IP)	IP / Subnet – eth3 (PUBLIC IP)	Infiniband Network IP
Master : bduhpc.bdu.ac.in	172.16.50.60/ 255.255.0.0	172.16.150.50/255.255.0.0	192.168.2.60
Compute Nodes: (Node1-Node10)	172.16.50.61 - 172.16.50.70	NA	192.168.2.61 – 192.168.2.70
IO Nodes : (IONode1-IONode2)	172.16.50.80 – 172.16.50.81	NA	192.168.2.80 – 192.168.2.81

/etc/hosts file :

[root@bduhpc ~]# cat /etc/hosts

```
#####Network IP Lists#####
172.16.50.60      bduhpc      bduhpc.bdu.ac.in
172.16.50.61      node1
172.16.50.62      node2
172.16.50.63      node3
172.16.50.64      node4
172.16.50.65      node5
172.16.50.66      node6
172.16.50.67      node7
172.16.50.68      node8
172.16.50.69      node9
172.16.50.70      node10
172.16.50.80      ionode1
172.16.50.81      ionode2
```

```
##### IB Network #####
192.168.2.60      bduhpc-ib
192.168.2.61      node1-ib
192.168.2.62      node2-ib
192.168.2.63      node3-ib
192.168.2.64      node4-ib
192.168.2.65      node5-ib
192.168.2.66      node6-ib
192.168.2.67      node7-ib
192.168.2.68      node8-ib
192.168.2.69      node9-ib
192.168.2.70      node10-ib
192.168.2.80      ionode1-ib
192.168.2.81      ionode2-ib
```

3. Software Packages

The below are the software packages installed:

- i. Ganglia - 3.5
Installed Path = /etc/ganglia
Location = Master (bduhpc)
- ii. SLURM 14.11.7
Installed Path = /opt/slurm
Location = Master (bduhpc)
- iii. Open Mpi 1.8
Installed Path = /opt/openmpi-1.8.8
Location = Master (bduhpc)
- iv. Mvapich2.2
Installed Path = /opt/mvapich2
Location = Master (bduhpc)
- v. Intel Compiler 2015
Installed Path = /opt/intel
Location = Master (bduhpc)
- vi. Namd-2.9
Installed Path = /home/test/namd/NAMD_2.9_Linux-x86_64
Location = Master (bduhpc)
- vii. Fftw 3.2.1 (library)
Installed Path = /usr/bin/fftw-3.2.1
Location = Master (bduhpc)
- viii. Mellanox OFED-3.5-2

4. HPC

4.1 Man Made Cluster :

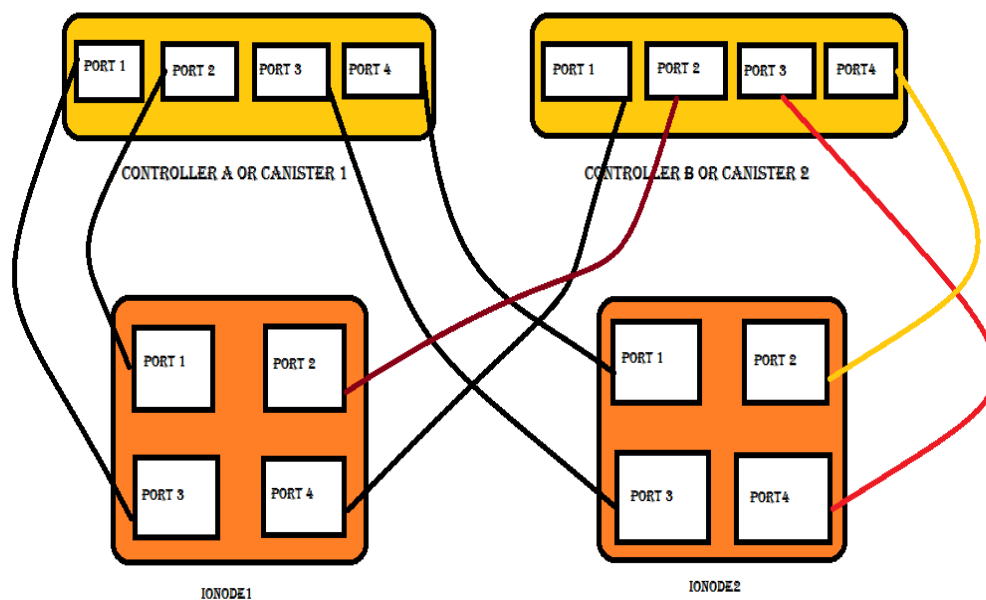
In Man Made Cluster we have to first configure RAID level(0 or 1) on both Master,IO Nodes & Compute Nodes. It is an easy & basic method Cluster Suite for installation of OS.

Internal disk partitioning

The following table details the internal disk partitioning configuration for the Management Servers (Master Server):

sda – 835 GB Raid 5				
/boot/efi	/boot	swap	/	/var
sda1	sda2	sda4	sda3	sda5
500 MB	500 MB	104 GB	630 GB	100GB
Ext4	Ext4	swap	Ext4	Ext4

4.2 Configure Storage V3700



FC Connectivity:-

1. Controller-1 or Canister-1 has 2 OUT FC port is connected to IONode1 IN FC port.
2. Controller-1 or Canister-1 has 2 OUT FC port is connected to IONode2 IN FC port.
3. Controller-2 or Canister-2 has 2 OUT FC port is connected to IONode1 IN FC port.
4. Controller-2 or Canister-2 has 2 OUT FC port is connected to IONode2 IN FC port.

5. Cluster Management

5.1 Network Information Services (NIS) :

Checking NIS:

Run the below command to check if NIS is working on master node or not:

```
#ypwhich  
bduhpc.bdu.ac.in
```

If you don't get the same please run the below commands:

```
#domainname  
#service ypbinding restart
```

Now check the “ypwhich” command once again. You will get the answer.

Now on the compute node if you don't get “ypwhich” output please run the below command:

```
#domainname  
#service ypbinding restart  
./cluster-fork service ypbinding status
```

Eg:

```
[root@bduhpc Desktop]# ./dsh allnodes "/etc/init.d/ypbinding status"
```

```
ypbinding (pid 8780) is running...  
node1 done  
ypbinding (pid 8285) is running...  
node2 done  
ypbinding (pid 8650) is running...  
node3 done  
ypbinding (pid 8391) is running...  
node4 done  
ypbinding (pid 8521) is running...  
node5 done  
ypbinding (pid 8874) is running...  
node6 done
```

6. Scheduler – SLURM

SLURM Overview:

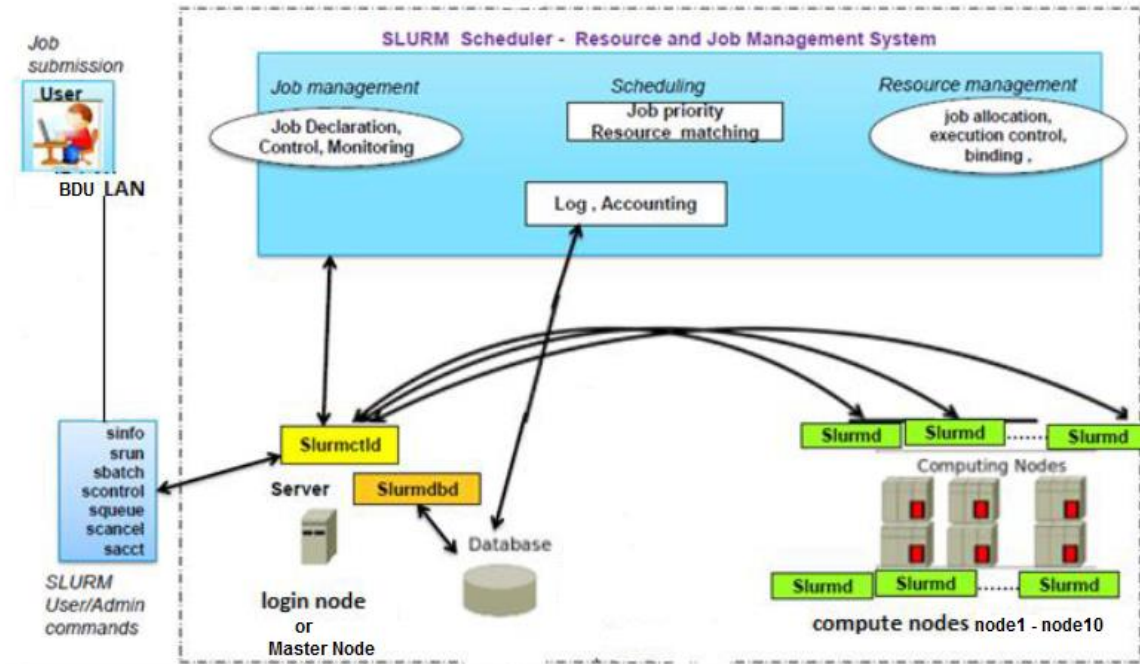
The **Simple Linux Utility for Resource Management (SLURM)** is an open source, fault-tolerant and highly scalable cluster management and job scheduling system for large and small Linux clusters. SLURM requires no kernel modifications for its operation and is relatively self-contained. As a cluster resource manager, SLURM has three key functions.

- First, it allocates exclusive and/or non-exclusive access to resources (compute nodes) to users for some duration of time so they can perform work.
- Second, it provides a framework for starting, executing, and monitoring work (normally a parallel job) on the set of allocated nodes.
- Finally, it arbitrates contention for resources by managing a queue of pending work.

SLURM Architecture:

The SLURM architecture has three daemons

- SLURM controller daemon (slurmctld)
 - Runs on masternode (login server),
 - backup slurmctld runs on master2 server
- SLURM daemon (slurmd)
 - Runs on all compute nodes (n1 to n108)
- SLURM database daemon (slurmdbd)
 - Uses mysql database and stores in shared filesystem
- SLURM has a centralized manager slurm controller daemon “**slurmctld**” to monitor resources and work. There may also be a backup manager (slurm backup controller daemon) on failover server to assume those responsibilities in the event of failure.



Each compute server (node) has a “**slurmd**” daemon, which can be compared to a remote shell: it waits for work to be allocated, and executes (ie run the job), and return the job status to the slurmd, and waits for more work to be assigned from controller daemon. The slurmd daemons provide fault-tolerant hierarchical communications.

There is an optional “**slurmdbd**” (Slurm DataBase Daemon) which can be used to record accounting information for multiple Slurm-managed clusters in a single database. The database used to store the data is mysql

SLURM Terminology

- **Partition:** SLURM groups the compute nodes for the resource allocation and is called as partitions. It can also be referred as **queues**. Users will request a specific partition for the resource allocation of their job. The default partition in this cluster is “**debug**”

6.1 Slurm Environment

- Installation and Version Details

The SLURM Job Scheduler installed details: -

Installed Path	Installed Directory [root@bduhpc ~]# cd /opt/slurm/ bin/ etc/ include/lib/ sbin/ share/ state/ tmp/
Installed Version	# sinfo -V [root@bduhpc ~]# sinfo -V SLURM 14.11.7

Submission Host	# hostname -s [root@bduhpc ~]# hostname -s bduhpc
User Details	#id [wipro@bduhpc ~]# id uid=610(ccms) gid=611(ccms) groups=611(ccms)

The user will submit jobs from the **master server – bduhpc**

- **View the cluster status**

The status of the Cluster, compute nodes and partitions can be viewed with the sinfo command.

- sinfo [options]

```
test@bduhpc:~
[test@bduhpc ~]$ sbatch namd.sh
Submitted batch job 186
[test@bduhpc ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
CPU*      up    infinite    2   alloc node[1-2]
CPU*      up    infinite    8   idle  node[3-10]
[test@bduhpc ~]$
```

- sinfo --summarize

```

[~] test@bduhpc ~]$
[~] test@bduhpc ~]$
[~] test@bduhpc ~]$
[~] test@bduhpc ~]$
[~] test@bduhpc ~]$
[~] test@bduhpc ~]$ sinfo --summarize
PARTITION AVAIL  TIMELIMIT  NODES (A/I/O/T)  NODELIST
CPU*      up      infinite      2/8/0/10  node[1-10]
[~] test@bduhpc ~]$

```

sinfo command - output field details	
Output field	Description
PARTITION	Name of a partition. the suffix "*" identifies the default partition of this cluster
AVAIL	Partition state: up or down up/down : indicates the partition is available for the users to submit their job or not
TIMELIMIT	Maximum run time limit for any user job in days-hours:minutes:seconds. infinite indicates no runtime limit is set for partition job time limit.

NODES	Total no of compute nodes in the partition
STATE	Refers to the node status whether it is idle, allocated, down etc The different node status details is available in the appendix
NODELIST	Names of compute nodes associated with this partition/queue
NODES (A/I/O/T)	Total no. of nodes state details - " allocated/idle/other/total " for this partition
CPUS	Count of CPUs (processors) on each node.
CPUS (A/I/O/T)	Total no. of CPUs details - " allocated/idle/other/total " for this partition
S:C:T	Count of sockets (S), cores (C), and threads (T) on these nodes.
MEMORY	Size of real memory in megabytes on these nodes
TMP_DISK	Size of temporary disk space in megabytes on these nodes
WEIGHT	Scheduling weight of the nodes
FEATURES	Features associated with the nodes (like gpu, intel etc)
REASON	Gives the information to identify why the node is unavailable

SLURM Commands :-

Man pages exist for all SLURM daemons, commands, and API functions. The command option --help also provides a brief summary of options. Note that the command options are all case insensitive.

sbatch :- is used to submit a job script for later execution. The script will typically contain one or more srun commands to launch parallel tasks.

sinfo :- reports the state of partitions and nodes managed by SLURM. It has a wide variety of filtering, sorting, and formatting options.

srun :- is used to submit a job for execution or initiate job steps in real time. srun has a wide variety of options to specify resource requirements, including: minimum and maximum node count, processor count, specific nodes to use or not use, and specific node characteristics (so much memory, disk space, certain required features, etc.). A job can contain multiple job steps executing sequentially or in parallel on independent or shared nodes within the job's node allocation.

scancel :- is used to cancel a pending or running job or job step. It can also be used to send an arbitrary signal to all processes associated with a running job or job step.

Scontrol :- is the administrative tool used to view and/or modify SLURM state. Note that many scontrol commands can only be executed as user root.

Sview :- is a graphical user interface to get and update state information for jobs, partitions, and nodes managed by SLURM.

smap :- reports state information for jobs, partitions, and nodes managed by SLURM, but graphically displays the information to reflect network topology.

sacct :- is used to report job or job step accounting information about active or completed jobs.

salloc :- is used to allocate resources for a job in real time. Typically this is used to allocate resources and spawn a shell. The shell is then used to execute srun commands to launch parallel tasks.

sattach :- is used to attach standard input, output, and error plus signal capabilities to a currently running job or job step. One can attach to and detach from jobs multiple times.

sbcast :- is used to transfer a file from local disk to local disk on the nodes allocated to a job. This can be used to effectively use diskless compute nodes or provide improved performance relative to a shared file system.

squeue :- reports the state of jobs or job steps. It has a wide variety of filtering, sorting, and formatting options. By default, it reports the running jobs in priority order and then the pending jobs in priority order.

strigger :- is used to set, get or view event triggers. Event triggers include things such as nodes going down or jobs approaching their time limit.

View the partition/queue

To view detailed partition/queue information, the below command can be used –

- **Command to check partition status in slurm :**
`scontrol show partititon <partitionname>`

```
PartitionName=CPU
  AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
  AllocNodes=ALL Default=YES
  DefaultTime=NONE DisableRootJobs=NO GraceTime=0 Hidden=NO
  MaxNodes=UNLIMITED MaxTime=UNLIMITED MinNodes=1 LLN=NO
MaxCPUsPerNode=UNLIMITED
  Nodes=node[1-10]
  Priority=1 RootOnly=NO ReqResv=NO Shared=NO PreemptMode=OFF
  State=UP TotalCPUs=200 TotalNodes=10 SelectTypeParameters=N/A

  DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED
```

- **Command to check node status in slurm :**

`scontrol show node`

```
NodeName=node1 Arch=x86_64 CoresPerSocket=1
  CPUAlloc=0 CPUErr=0 CPUTot=20 CPULoad=0.02 Features=(null)
  Gres=(null)
  NodeAddr=node1 NodeHostName=node1 Version=14.11
  OS=Linux RealMemory=1 AllocMem=0 Sockets=20 Boards=1
  State=IDLE ThreadsPerCore=1 TmpDisk=0 Weight=1
  BootTime=2015-09-01T12:32:42 SlurmdStartTime=2015-09-01T16:40:50
  CurrentWatts=0 LowestJoules=0 ConsumedJoules=0
  ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
```


NodeName=node2 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUErr=0 CPUTot=20 CPULoad=0.00 Features=(null)
Gres=(null)
NodeAddr=node2 NodeHostName=node2 Version=14.11
OS=Linux RealMemory=1 AllocMem=0 Sockets=20 Boards=1
State=IDLE ThreadsPerCore=1 TmpDisk=0 Weight=1
BootTime=2015-09-01T12:32:36 SlurmdStartTime=2015-09-01T16:41:04
CurrentWatts=0 LowestJoules=0 ConsumedJoules=0
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s

NodeName=node3 CoresPerSocket=1
CPUAlloc=0 CPUErr=0 CPUTot=20 CPULoad=N/A Features=(null)
Gres=(null)
NodeAddr=node3 NodeHostName=node3 Version=(null)
RealMemory=1 AllocMem=0 Sockets=20 Boards=1
State=DOWN* ThreadsPerCore=1 TmpDisk=0 Weight=1
BootTime=None SlurmdStartTime=None
CurrentWatts=0 LowestJoules=0 ConsumedJoules=0
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=Not responding [root@2015-09-01T16:49:40]

NodeName=node4 CoresPerSocket=1
CPUAlloc=0 CPUErr=0 CPUTot=20 CPULoad=N/A Features=(null)
Gres=(null)
NodeAddr=node4 NodeHostName=node4 Version=(null)
RealMemory=1 AllocMem=0 Sockets=20 Boards=1
State=DOWN* ThreadsPerCore=1 TmpDisk=0 Weight=1
BootTime=None SlurmdStartTime=None
CurrentWatts=0 LowestJoules=0 ConsumedJoules=0
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=Not responding [root@2015-09-01T16:49:40]

NodeName=node5 CoresPerSocket=1
CPUAlloc=0 CPUErr=0 CPUTot=20 CPULoad=N/A Features=(null)
Gres=(null)
NodeAddr=node5 NodeHostName=node5 Version=(null)
RealMemory=1 AllocMem=0 Sockets=20 Boards=1
State=Alloc ThreadsPerCore=1 TmpDisk=0 Weight=1
BootTime=None SlurmdStartTime=None
CurrentWatts=0 LowestJoules=0 ConsumedJoules=0
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=Not responding [root@2015-09-01T16:49:40]

NodeName=node6 CoresPerSocket=1
CPUAlloc=0 CPUErr=0 CPUTot=20 CPULoad=N/A Features=(null)
Gres=(null)
NodeAddr=node6 NodeHostName=node6 Version=(null)
RealMemory=1 AllocMem=0 Sockets=20 Boards=1
State=Alloc ThreadsPerCore=1 TmpDisk=0 Weight=1
BootTime=None SlurmdStartTime=None
CurrentWatts=0 LowestJoules=0 ConsumedJoules=0

```
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s  
Reason=Not responding [root@2015-09-01T16:49:41]
```

```
NodeName=node7 CoresPerSocket=1  
CPUAlloc=0 CPUErr=0 CPUTot=20 CPULoad=N/A Features=(null)  
Gres=(null)  
NodeAddr=node7 NodeHostName=node7 Version=(null)  
RealMemory=1 AllocMem=0 Sockets=20 Boards=1  
State=Alloc ThreadsPerCore=1 TmpDisk=0 Weight=1  
BootTime=None SlurmdStartTime=None  
CurrentWatts=0 LowestJoules=0 ConsumedJoules=0  
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s  
Reason=Not responding [root@2015-09-01T16:49:41]
```

```
NodeName=node8 CoresPerSocket=1  
CPUAlloc=0 CPUErr=0 CPUTot=20 CPULoad=N/A Features=(null)  
Gres=(null)  
NodeAddr=node8 NodeHostName=node8 Version=(null)  
RealMemory=1 AllocMem=0 Sockets=20 Boards=1  
State=Alloc ThreadsPerCore=1 TmpDisk=0 Weight=1  
BootTime=None SlurmdStartTime=None  
CurrentWatts=0 LowestJoules=0 ConsumedJoules=0  
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s  
Reason=Not responding [root@2015-09-01T16:49:41]
```

```
NodeName=node9 CoresPerSocket=1  
CPUAlloc=0 CPUErr=0 CPUTot=20 CPULoad=N/A Features=(null)  
Gres=(null)  
NodeAddr=node9 NodeHostName=node9 Version=(null)  
RealMemory=1 AllocMem=0 Sockets=20 Boards=1  
State=Idle ThreadsPerCore=1 TmpDisk=0 Weight=1  
BootTime=None SlurmdStartTime=None  
CurrentWatts=0 LowestJoules=0 ConsumedJoules=0  
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s  
Reason=Not responding [root@2015-09-01T16:49:41]
```

```
NodeName=node10 CoresPerSocket=1  
CPUAlloc=0 CPUErr=0 CPUTot=20 CPULoad=N/A Features=(null)  
Gres=(null)  
NodeAddr=node10 NodeHostName=node10 Version=(null)  
RealMemory=1 AllocMem=0 Sockets=20 Boards=1  
State=Idle ThreadsPerCore=1 TmpDisk=0 Weight=1  
BootTime=None SlurmdStartTime=None  
CurrentWatts=0 LowestJoules=0 ConsumedJoules=0  
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s  
Reason=Not responding [root@2015-09-01T16:49:41]
```

6.2 Job Submission command 'sbatch'

eg: sbatch /home/test/submit.sh

- **Monitoring job using the command 'Squeue'**

CommandDescription

i)	scancel	delete/cancel batch jobs
ii)	squeue	to see job status
iii)	sacct	accounting information

Submit Job with following Script for reference :

```
[root@bduhpc]# cat submit.sh
```

```
#!/bin/bash

#SBATCH --job-name=JobName

#SBATCH --output=JobName-%j.out

#SBATCH --error=JobName-%j.err

#SBATCH --partition=PartitionName

#SBATCH --nodes=NumberNodes

#SBATCH --ntasks-per-node=NumberTasksPerNode

# below your job commands:
```

mpirun your_commands

Job submission Script

The Sample SLURM job submission script –

```
## -----
## SLURM sample Job array submission script example
## -----
## Options
## -J or --job-name jobname
## -o or --output Output file
## -e or --error error file
## -p or --partition partition (queue) name
## --account user's associated account
## --ntasks no. of total cpus
## --ntasks_per_node no. of tasks per node
## --nodes no. of nodes
## -----
## Environment variables
## SLURM_NTASKS - Replaces the --ntasks value
##
```

```
## %j will be replaced by the value of Job id
```

```
## -----
```

```
--
```

```
#!/bin/sh
```

```
#SBATCH --job-name=namd-job
```

```
#SBATCH --partition=cpu
```

```
#SBATCH --output= namd-%J.out
```

```
#SBATCH --error= namd-%J.out
```

```
#SBATCH --workdir=/home/test
```

```
#SBATCH --nodes=4
```

```
#SBATCH --ntasks-per-node=10
```

```
#SBATCH --ntasks=40
```

```
cd /home/test/namd/apoal
```

```
MACHINE_FILE=/home/test/machinefile
```

```
scontrol show hostname
```

```
$SLURM_JOB_NODELIST &> $MACHINE_FILE
```

```
/opt/intel/impi/5.0.3.048/intel64/bin/mpirun -np 16 -machinefile
```

```
$MACHINE_FILE /home/test/namd/NAMD_2.9_Linux-x86_64/namd2 apoal.namd
```

6.3 Job Monitoring

- **Monitor a job**

The status of all jobs can be viewed with the `squeue` command. Without any options all jobs are displayed.

`squeue <options>`

`squeue -l` : This options provides the details of all the running jobs of all the users

- **Delete a job**

`scancel <jobid>`

Environment Variables

SLURM sets environment variables that your running job script can use:

Some of the used environment variables in the submission script –

SLURM Environment Variables	
Output field	Description
SLURM_SUBMIT_DIR	The directory that the job was submitted from
SLURM_JOB_NAME	The name of the job (such as specified with --job-name=)
SLURM_JOB_ID	The unique identifier (job id) for this job
SLURM_JOB_NODELIST	List of node names assigned to the job
SLURM_NTASKS	Number of tasks allocated to the job
SLURM_JOB_CPUS_PER_NODE	Number of CPUs per node available to the job
SLURM_JOB_NUM_NODES	Number of nodes allocated to the job
SLURM_ARRAY_TASK_ID	This tasks's ID in the job array
SLURM_ARRAY_JOB_ID	The master job id for the job array

Filename Pattern

The filename pattern may contain one or more replacement symbols, which are a percent sign "%" followed by a letter (e.g. %j).

Supported replacement symbols are:

SLURM filename pattern	
Pattern	Description
%j	Job allocation number
%N	Node name. Only one file is created, so %N will be replaced by the name of the first node in the job, which is the one that runs the script ie called Batch Host
%u	User name of the user who submitted the job

Below pattern's are specific to array jobs	
%A	Job array's master job allocation number
%a	Job array ID (index) number.

sinfo (Node State)

Different node States

Node States		
NODE State --long format	Node State – Short form	Description
ALLOCATED	ALLOC	The node has been allocated to one or more jobs ALLOCATED+: The node is allocated to one or more active jobs plus one or more jobs are in the process of COMPLETING.
COMPLETING	COMP	All jobs associated with this node are in the process of COMPLETING
DOWN	DOWN	The node is unavailable for submitting jobs. SLURM can automatically place nodes in this state if some failure occurs.
DRAINING / DRAINED	DRAIN	The node is unavailable for submitting jobs. System Administrator has set the node to unavailable state
ERROR	ERR	The node is currently in an error state and unavailable for running any jobs. SLURM can automatically place nodes in this state if some failure occurs.

FAIL	FAIL	The node is expected to fail soon and is unavailable for use, which is been set by the System Administrator FAILING: The node is currently executing a job, but is expected to fail soon
FUTURE	FUTR	The node is currently not fully configured, but expected to be available in future for use

IDLE	IDLE	The node is not allocated to any jobs and is available for submitting jobs
	MAINT	The node is currently in a reservation with a flag value of "maintenance" or is scheduled to be rebooted.
MIXED	MIX	The node has some of its CPUs ALLOCATED while others are IDLE
PERFCTRS		Network Performance Counters associated with this node are in use, rendering this node as not usable for any other jobs
POWER_DOWN		The node is currently powered down and not capable of running any jobs.
POWER_UP		The node is currently in the process of being powered up
RESERVED	RESV	The node is in an advanced reservation and not generally available
UNKNOWN	UNK	The SLURM controller has just started and the node's state has not yet been determined.

queue (Job state)

Different Job States

Job States		
Job State -- Short Form	Long Form	Description
PD	PENDING	Job is awaiting resource allocation
R	RUNNING	Job currently allocated resources
CG	COMPLETING	Job is in the process of completing. Some processes on some nodes may still be active.
CD	COMPLETED	Job has terminated all processes on all nodes
F	FAILED	Job terminated with non-zero exit code or other failure condition

CA	CANCELLED	Job was explicitly cancelled/terminated by the user or system administrator
S	SUSPENDED	Job has been allocated the resources, but execution has been suspended either by the user/administrator or scheduler
PR	PREEMPTED	Job terminated due to preemption
CF	CONFIGURING	Job has been allocated resources, but are waiting for them to become ready for use
TO	TIME OUT	Job terminated upon reaching its time limit
NF	NODE FAILURE	Job terminated due to failure of one or more allocated nodes
BF	BOOT_FAIL	Job terminated due to launch failure, typically due to a hardware failure (e.g. unable to boot the node or block and the job cannot be requeued).

Slurm with fair share:

The fair-share component to a job's priority influences the order in which a user's queued jobs are scheduled to run based on the portion of the computing resources they have been allocated and the resources their jobs have already consumed. The fair-share factor does not involve a fixed allotment, whereby a user's access to a machine is cut off once that allotment is reached.

Instead, the fair-share factor serves to prioritize queued jobs such that those jobs charging accounts that are under-served are scheduled first, while jobs charging accounts that are over-served are scheduled when the machine would otherwise go idle.

SLURM's fair-share factor is a floating point number between 0.0 and 1.0 that reflects the shares of a computing resource that a user has been allocated and the amount of computing resources the user's jobs have consumed. The higher the value, the higher is the placement in the queue of jobs waiting to be scheduled.

Below are the configuration parameters we have applied for fair share implementation in bduhpc cluster slurm.conf file:

```
ControlMachine=bduhpc
ControlAddr=bduhpc
#BackupController=
#BackupAddr=
#
AuthType=auth/munge
CacheGroups=0
JobCredentialPrivateKey=/opt/slurm/etc/myopenssl
JobCredentialPublicCertificate=/opt/slurm/etc/myopensslcert
MpiDefault=none
ProctrackType=proctrack/pgid
ReturnToService=1
SlurmctldPidFile=/var/run/slurmctld.pid
SlurmctldPort=6817
SlurmdPidFile=/var/run/slurmd.pid
SlurmdPort=6818
SlurmdSpoolDir=/var/spool/slurmd
#SlurmUser=root
SlurmdUser=root
StateSaveLocation=/var/spool
SwitchType=switch/none
TaskPlugin=task/none
InactiveLimit=0
KillWait=30
MinJobAge=300
SlurmctldTimeout=120
SlurmdTimeout=300
Waittime=0
FastSchedule=1
SchedulerType=sched/builtin
SchedulerPort=7321
SelectType=select/linear
ClusterName=bduhpc
#AccountingStorageType=accounting_storage/none
AccountingStorageType=accounting_storage/mysql
SlurmctldDebug=3
SlurmdDebug=3
#
# COMPUTE NODES
NodeName=node[1-10] CPUs=20 State=IDLE
PartitionName=CPU Nodes=node[1-10] Default=YES MaxTime=INFINITE
State=UP
```

Command to display fair share:

```
sshare -a
```

For fair share to work first a cluster needs to be added:

- `sacctmgr add cluster bduhpc`

Then users' needs to be added :

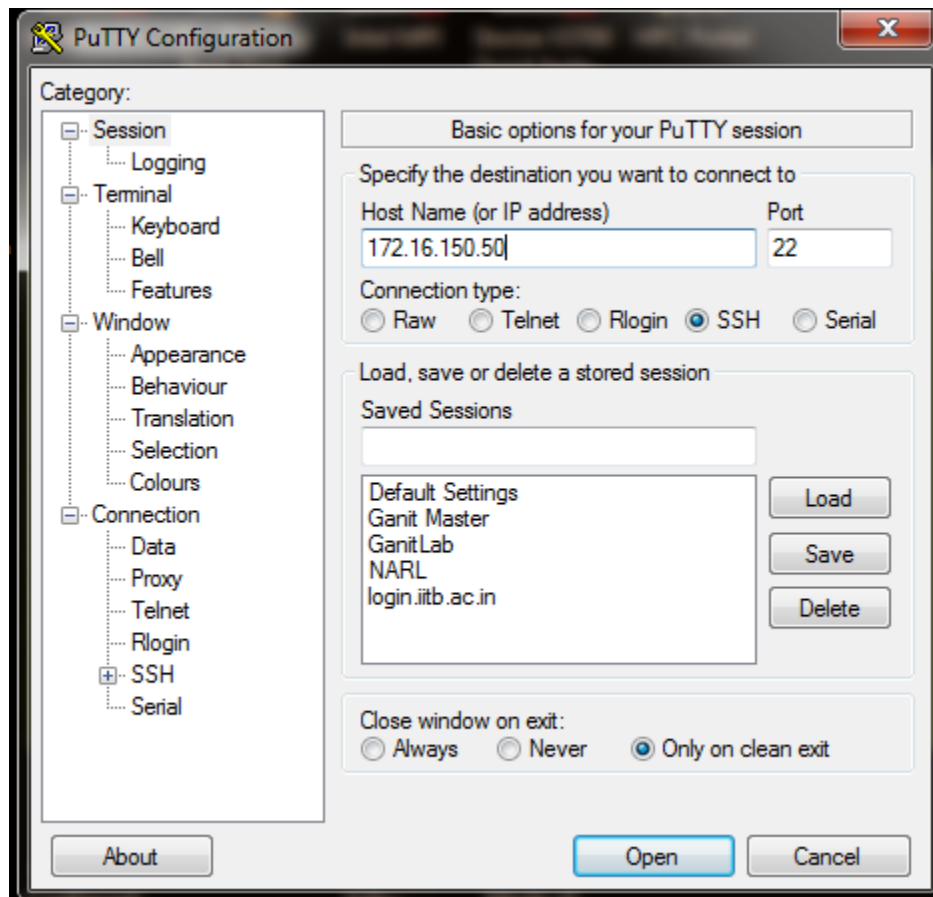
- `sacctmgr add user wipro`

7.How to connect Hpc cluster

Users can able to login from anywhere at BDU LAN, where SSH is enabled. So users can simply do ssh from their systems.

From windows machine user must use putty utility:-

```
ssh 172.16.150.50  
username : xxxxxxxxxxxxxx  
  
password : xxxxxxxxxxxxxx
```



or

From any linux machine users can login directly

```
ssh username@172.16.150.50  
password : xxxxxxxxxxxxxxxx
```

From windows machine user can use putty utility:

Note:- Putty utility is freely available on internet, download putty.exe from internet and run putty.exe

8. Wipro HPC Portal

Wipro developed portal for submitting job through graphical user interface.

Access: <http://172.16.150.50:8080/whpc-portal/pages/Login.faces>

Use cluster root credentials to login.

History

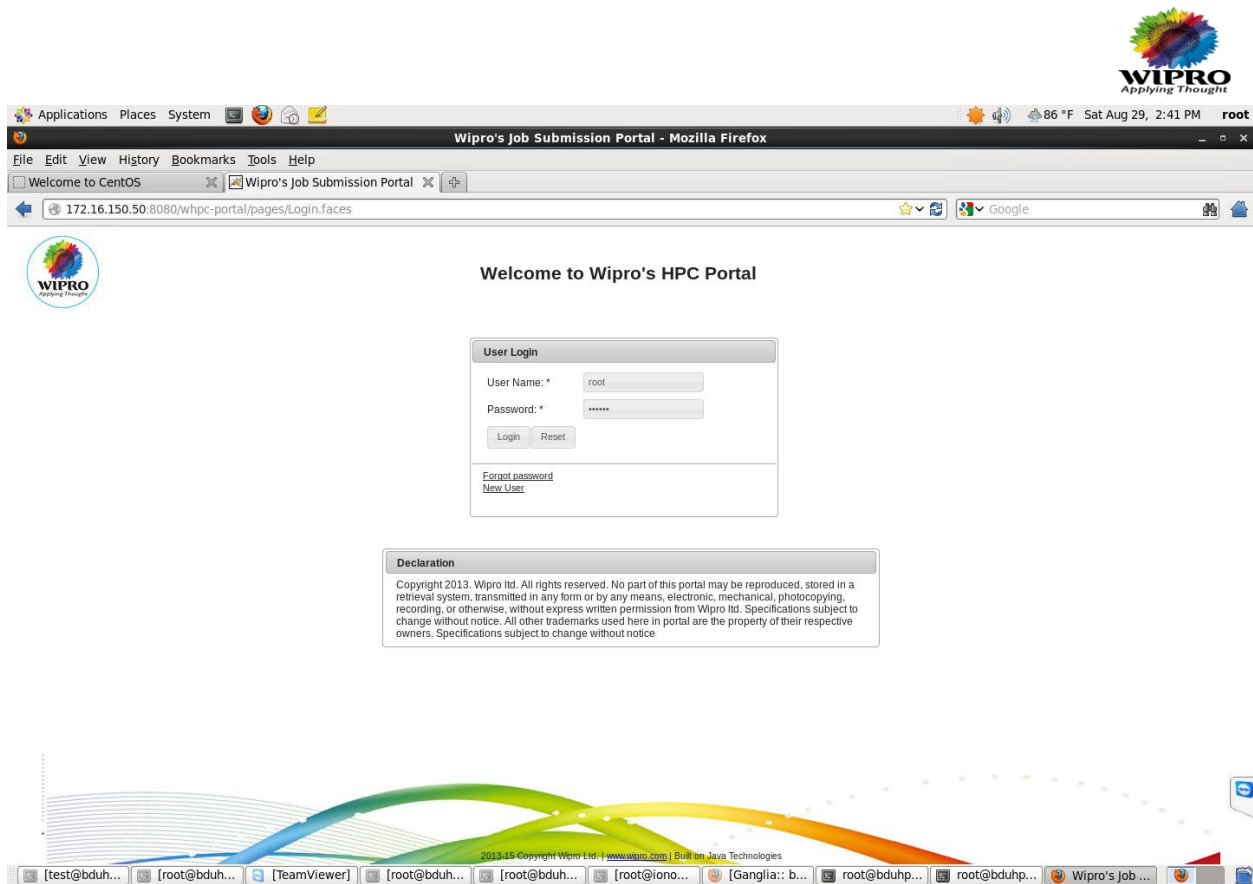
The details of the job which have completed or failed can be viewed here. User can view the details of both 'portal' as well as 'CLI' jobs from here.

The administrator can customize the time frame for which he wants to view the job. The following time frames are available:

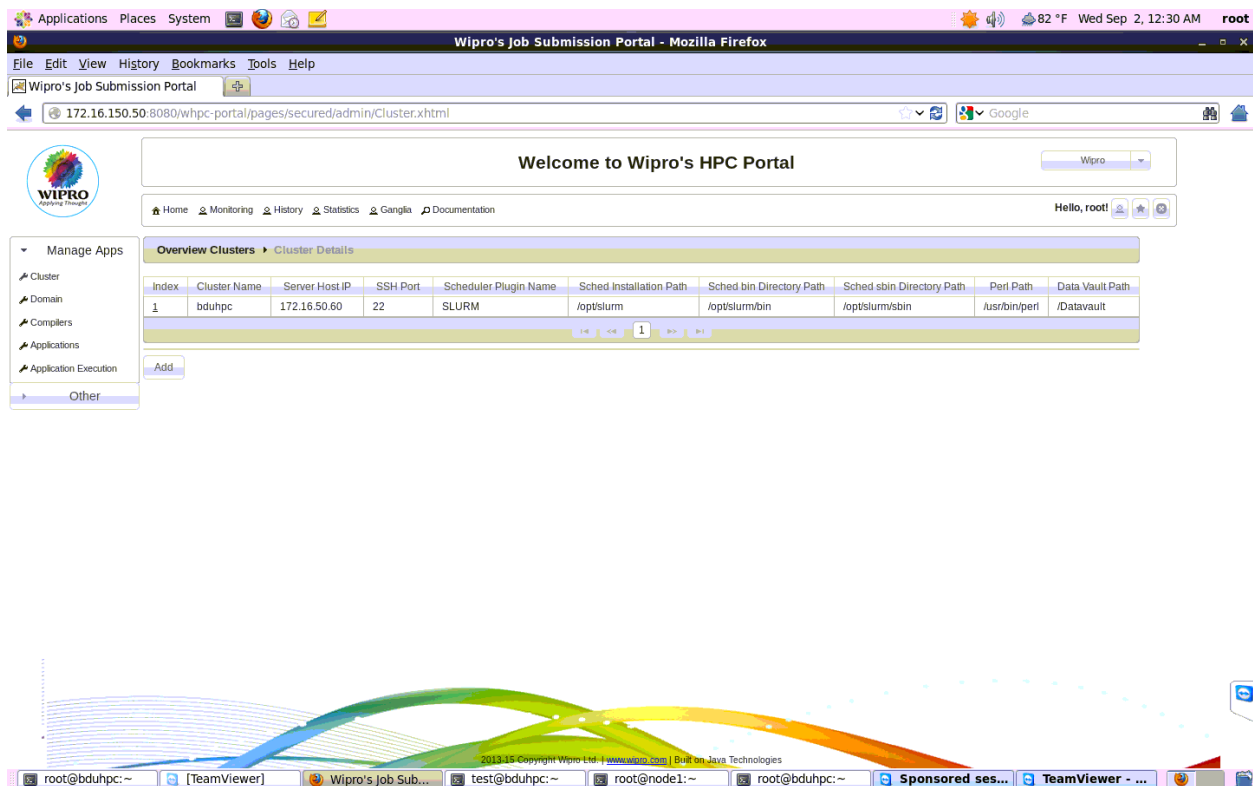
1. Last hour
2. Last day
3. Last week
4. Fortnight
5. Last month
6. Half yearly
7. Last year
8. Customize: As per user need

The user can view the following details:

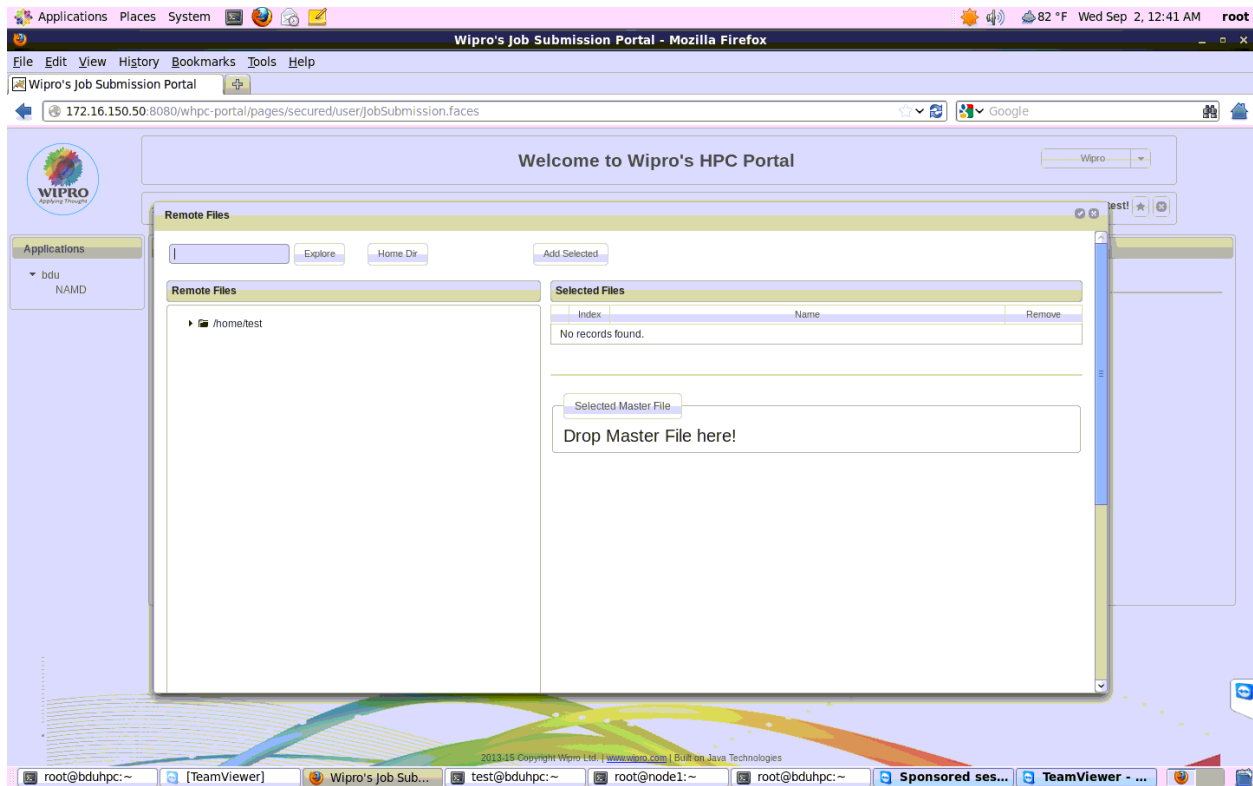
- a. Job Id: The Id of the job
- b. Username: The name of the user who ran the job
- c. State: The status of the job (completed, cancelled or failed)
- d. Queue: The queue in which the job was run.
- e. Nodes: The number of nodes on which the job ran
- f. CPU: No. of CPUs allocated for the job while it ran
- g. Total Time: The total time for which the job ran.
- h. Cluster Name: The name of the cluster in which the job ran.



Wipro Portal Login or Access Page.



After you log in to the portal you can see the applications and click on that application and give the parameters to submit job.



You can drop your Master File.

Applications Places System

Wipro's Job Submission Portal - Mozilla Firefox

Welcome to Wipro's HPC Portal

Home Monitoring History Statistics Ganglia Documentation

Hello, root!

Manage Apps

- Cluster
- Domain
- Compilers
- Applications
- Application Execution

Other

- User
- Notices
- Department

Overview Users User Details

Index	User Name	E-mail Address	Contact Number	Name	Department	User Role	Last Login	Last Logout	Last Session Id
1	root	root@localhost		root root					

Add User

User Role* User

Department* General

User name* test

First name* test

Last name* test

E-mail* test.com

Mobile Number

Date Of Birth*

Add User Reset

2013-15 Copyright Wipro Ltd. | www.wipro.com | Built on Java Technologies

[test@bduh...] [root@bduh...] [TeamViewer] [root@bduh...] [root@bduh...] [root@iono...] [Ganglia: b...] root@bduh... root@bduh... Wipro's Job ...

Adding New User.

Applications Places System

Wipro's Job Submission Portal - Mozilla Firefox

Welcome to Wipro's HPC Portal

Home Monitoring History Statistics Ganglia Documentation

Hello, root!

Manage Apps

- Cluster
- Domain
- Compilers
- Applications
- Application Execution

Other

- User
- Notices
- Department

Overview Users User Details

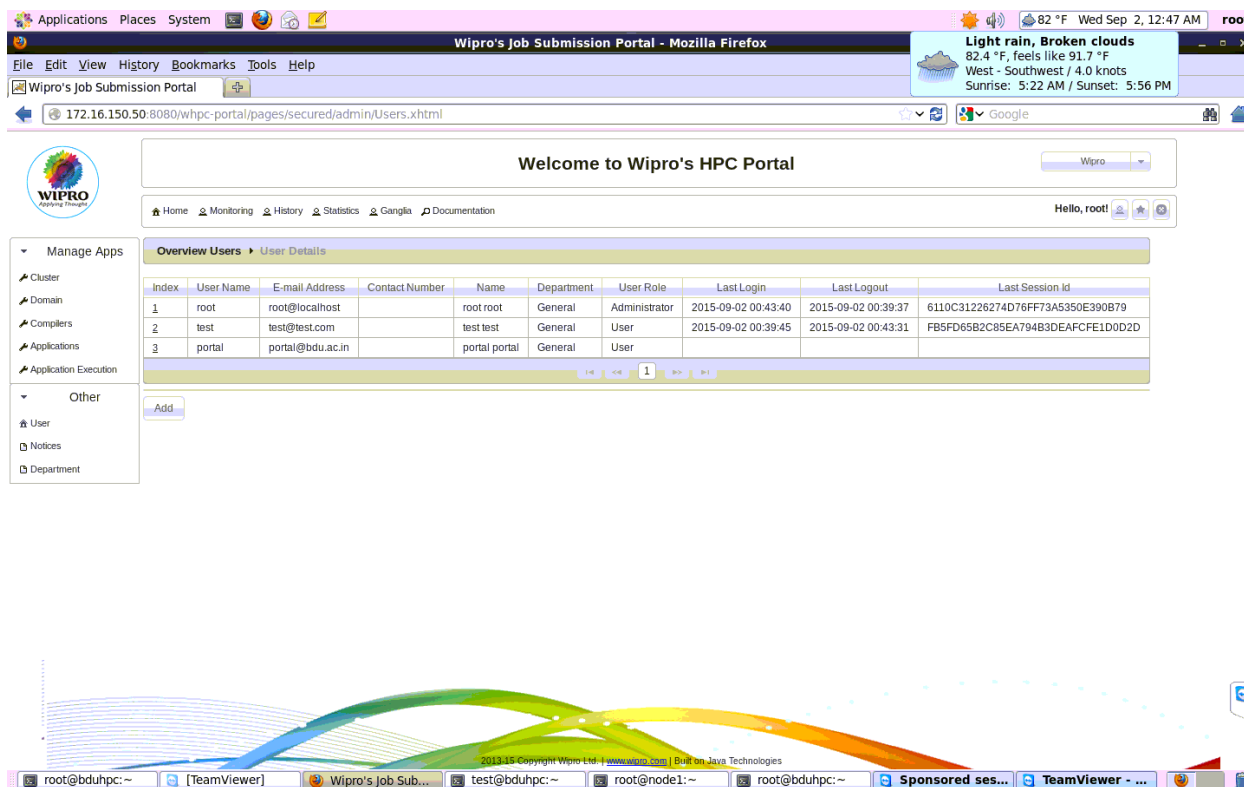
Index	User Name	E-mail Address	Contact Number	Name	Department	User Role	Last Login	Last Logout	Last Session Id
1	root	root@localhost		root root	General	Administrator	2015-09-02 00:43:40	2015-09-02 00:39:37	6110C31226274D76FF73A5350E390B79
2	test	test@test.com		test test	General	User	2015-09-02 00:39:45	2015-09-02 00:43:31	FB5FD65B2C85EA794B3DEAFCFE1D0D2D

Add

2013-15 Copyright Wipro Ltd. | www.wipro.com | Built on Java Technologies

root@bduhpc:~ [TeamViewer] Wipro's Job Sub... test@bduhpc:~ root@node1:~ root@bduhpc:~ Sponsored ses... TeamViewer - ...

User Details Page.

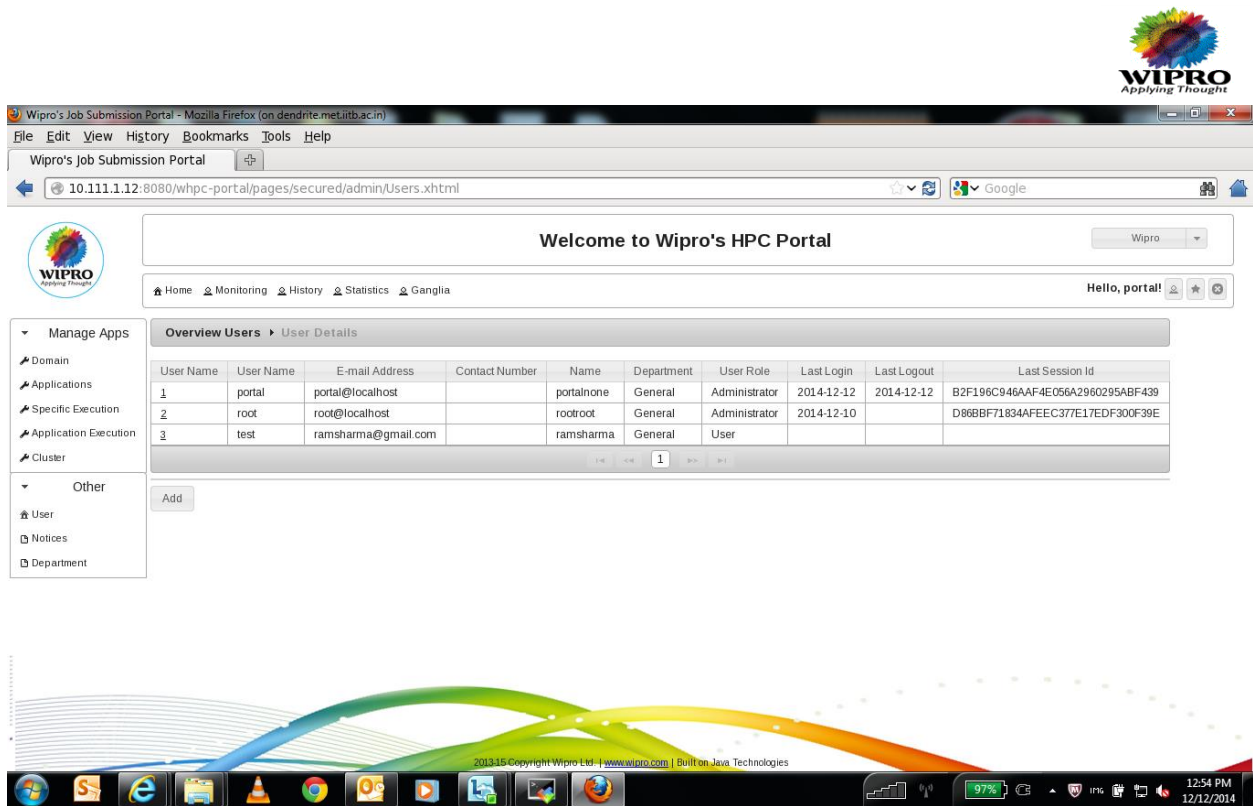


The screenshot shows a web browser window titled "Wipro's Job Submission Portal - Mozilla Firefox". The address bar displays "172.16.150.50:8080/whpc-portal/pages/secured/admin/Users.xhtml". The page content includes a welcome message "Welcome to Wipro's HPC Portal" and a navigation menu with options like Home, Monitoring, History, Statistics, Ganglia, and Documentation. A sidebar on the left contains "Manage Apps" and "Other" sections. The main content area shows "Overview Users" and "User Details" with a table of user information.

Index	User Name	E-mail Address	Contact Number	Name	Department	User Role	Last Login	Last Logout	Last Session Id
1	root	root@localhost		root root	General	Administrator	2015-09-02 00:43:40	2015-09-02 00:39:37	6110C31226274D76FF73A5350E390B79
2	test	test@test.com		test test	General	User	2015-09-02 00:39:45	2015-09-02 00:43:31	FB5FD65B2C85EA794B3DEAFCFE1D0D2D
3	portal	portal@bdu.ac.in		portal portal	General	User			

The bottom of the screenshot shows a taskbar with several open applications, including "root@bduhpc:~", "[TeamViewer]", "Wipro's Job Sub...", "test@bduhpc:~", "root@node1:~", "root@bduhpc:~", "Sponsored ses...", and "TeamViewer - ...".

User Details before Accepting New User Request.



Wipro's Job Submission Portal - Mozilla Firefox (on dendrite.metu.itb.ac.in)

File Edit View History Bookmarks Tools Help

Wipro's Job Submission Portal

10.111.1.12:8080/whpc-portal/pages/secured/admin/Users.xhtml

Google

Wipro

Welcome to Wipro's HPC Portal

Home Monitoring History Statistics Ganglia Hello, portal!

Manage Apps

- Domain
- Applications
- Specific Execution
- Application Execution
- Cluster

Other

- User
- Notices
- Department

Overview Users ▶ User Details

User Name	User Name	E-mail Address	Contact Number	Name	Department	User Role	Last Login	Last Logout	Last Session Id
1	portal	portal@localhost		portalnone	General	Administrator	2014-12-12	2014-12-12	B2F196C946AAF4E056A2960295ABF439
2	root	root@localhost		rootroot	General	Administrator	2014-12-10		D86BBF71834AFEEC377E17EDF300F39E
3	test	ramsharma@gmail.com		ramsharma	General	User			

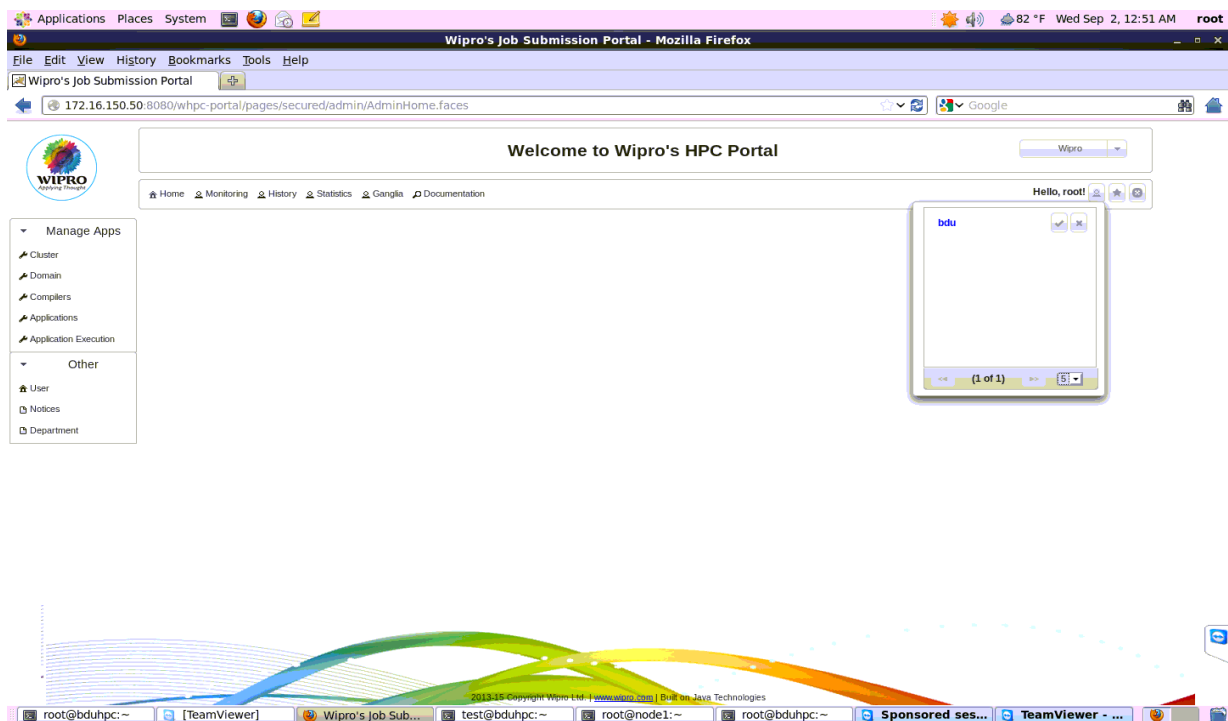
1 of 3

Add

2013-15 Copyright Wipro Ltd. | www.wipro.com | Built on Java Technologies

12:54 PM 12/12/2014

User List after Accepting New User as Test User.



Applications Places System

Wipro's Job Submission Portal - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Wipro's Job Submission Portal

172.16.150.50:8080/whpc-portal/pages/secured/admin/AdminHome.faces

Google

Wipro

Welcome to Wipro's HPC Portal

Home Monitoring History Statistics Ganglia Documentation Hello, root!

Manage Apps

- Cluster
- Domain
- Compilers
- Applications
- Application Execution

Other

- User
- Notices
- Department

test

1 of 1

2013-15 Copyright Wipro Ltd. | www.wipro.com | Built on Java Technologies

root@bduhpc:~ [TeamViewer] Wipro's Job Sub... test@bduhpc:~ root@node1:~ root@bduhpc:~ Sponsored ses... TeamViewer - ...

New User Request Notification

9. Cluster Status Monitoring (Ganglia)

A ganglia is the tool that provides the various information regarding the status of the Cluster. This information can be accessed through a web browser. The below IP address provide the access to this Monitoring tool.

This monitor gathers various metrics such as CPU load, free memory, disk usage, etc. These metrics are sent through the private cluster network and are used by the frontend n to generate the data in a graphical manner

In addition to metric parameters, a heartbeat message from each n is collected by the Ganglia. When certain number of heartbeats from any n is missed, this web page will declare it "dead".

Link : <http://172.16.150.50/ganglia>

The ganglia system is comprised of two unique daemons, a PHP-based web frontend and a few other small utility programs.

Ganglia Monitoring Daemon (gmond)

Ganglia Meta Daemon (gmetad)

Ganglia PHP Web Frontend

Ganglia Configuration:

All the data which is shown is fetched from the files listed in the /proc directory.

You will see the screen as shown below.

On the main page you can monitor cluster parameters like:

- 1) Cluster load last hour/day/year
- 2) Cluster memory last hour/day/year
- 3) Cluster CPU last hour/day/year
- 4) Cluster Network last hour/day/year

By selecting the drop down list on top you can select report for last hour, last day and last year's utilization.

You can also monitor individual n by selecting the n name from the drop down menu as shown below:

For individual hosts, you can see the details of following values:

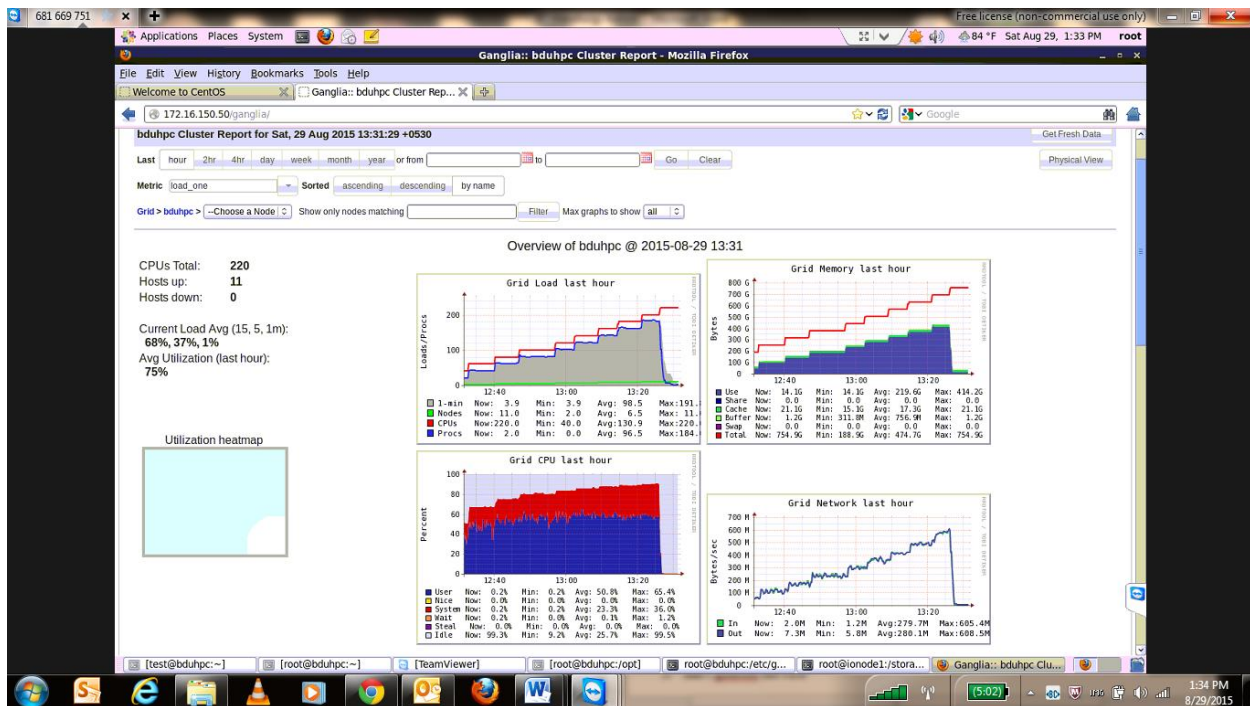
- | | | |
|--------------|---------------|----------------|
| 1) Bytes_in | 5) cpu_system | 9) proc_run |
| 2) Bytes_out | 6) cpu_user | 10) proc_total |
| 3) cpu_idle | 7) disk_free | 11) swp_free |
| 4) cpu_nice | 8) packets_in | |



Compute Nodes Statistics



Stacked Graph



Grid Structure

- i) Total CPU/Cores
- ii) Total Hosts Up
- iii) Total Hosts Down

10. OS Basic commands

ls: list information about the FilesOption:

- l: list one file per line
- t: sort by modification time
- h: print sizes in human readable format -a: list hidden files
- l : list in a table format

#du: estimates file space usage.

#df : report file system disk space usage.

#top : display Linux tasks.

#ps: report a snapshot of the current processes-e: all processes
-f: full

#tail: outputs the last part of files.

#ssh <node name>: used to login to the node. e.g ssh -p <port><user>@<nodename/IP> (ssh root@cn01)

11.HPC Support

Wipro:

Escalation Matrix

Level 1 – hpc-support

Level 2 – Pramod.K.P

Level 3 – A.S.V.S. Sastry

Level 4 – Jigar Halani

Name	Email id	Contact No.
hpc-support	hpc-support@wipro.com	+91-9611186130
Pramod.K.P	pramod.p26@wipro.com	+91-7259028153
A.S.V.S. Sastry	ayyalasomayajula.s46@wipro.com	+91-7259376671
Jigar Halani	Jigar.halani@wipro.com	+91-9902077344

Lenovo Hardware Calls:

rccindia@in.ibm.com

Schrodinger Support:

anirban.banerjee@schrodinger.com

Mob: +91- 9900090022

