Indian Institute of Science Education and Research
Thiruvananthapuram

Padmanabha

High Performance Cluster

User Manual

1. About the cluster Padmanabha

The cluster is having 88 cpu nodes and 3 gpu nodes. The node configurations are

CPU Nodes:

- 2 x Intel Xeon Gold 6132 CPU @ 2.60 GHz, 14 cores, 128 GB RAM
- Intel 100 Gbps OmniPath Interconnect
- Total 88 nodes and 2464 Cores for running MPI, Open MP and Hybrid jobs
- The Node name are node1 to node88.

GPU Nodes:

- 2 x Intel Xeon Gold 6132 CPU @ 2.60GHz 14Cores, 128GB RAM
- Intel 100 Gbps OmniPath Interconnect
- gnode1 having 2 x Nvidia P100 GPU Card and gnode2, gnode3 having 1 x Nvidia P100 GPU Card and each card are installed with necessary drivers and configured to work with slurm job scheduler.
- Total 84 Cores for running GPU jobs
- These nodes are named as gnode1 to gnode3

This cluster also has 500 TB lustre storage which is attached to "/home"

2. Slurm Job Scheduler - CLI

a) Submitting Jobs

There are 4 queues/partitions to submit a job. They are test, cpu, gpu, and long. User needs to create a job script as follows for submitting a job.

For cpu partition (default)

```
#!/bin/bash
#SBATCH -N 1
#SBATCH --ntasks-per-node=28
```

```
#SBATCH --time=120:00:00
#SBATCH --partition=cpu
#SBATCH -o slurm.%N.%j.out # STDOUT
#SBATCH -e slurm.%N.%j.err # STDERR
#SBATCH --mail-user=<username>@iisertvm.ac.in
#SBATCH –mail-type=ALL

<your program>
g16 benzene-mp4.
```

For gpu partition

```
#!/bin/bash
#SBATCH --job-name=newjob
#SBATCH --partition=gpu
#SBATCH --ntasks=1 ##Number of tasks
#SBATCH --cpus-per-task=4 ###Number of cpus per task
#SBATCH --gres=gpu:1
#SBATCH --time=120:00:00
#SBATCH -o slurm.%N.%j.out # STDOUT
#SBATCH -e slurm.%N.%j.err # STDERR
#SBATCH --mail-user=<username>@iisertvm.ac.in

<your program>
g16 benzene-mp4.com
```

For long partition (user has to request for access)

```
#!/bin/bash
#SBATCH --job-name=newjob
#SBATCH --partition=long
#SBATCH --qos=long_qos
#SBATCH --ntasks=1 ##Number of tasks
#SBATCH --cpus-per-task=4 ###Number of cpus per task
#SBATCH --time=120:00:00
#SBATCH -o slurm.%N.%j.out # STDOUT
#SBATCH -e slurm.%N.%j.err # STDERR
#SBATCH --mail-user=<username>@iisertvm.ac.in
```

```
<your program>
g16 benzene-mp4.com
```

Then submit myscript.sh as follows

```
$sbatch myscript.sh
```

The command ‘sbatch’ is to submit job to scheduler and returns a job id. This job id can be used later for monitoring and managing jobs.

Resource	Flag Syntax	Description	Notes
Partition	--partition=cpu	Partition is a queue for jobs.	default value is cpu
time	--time=01:00:00	Time limit for the job.	default value is120Hours.
nodes	--nodes=1	Number of compute nodes for the job.	default is 1
cpus/cores	--ntasks-per-node=8	Corresponds to number of cores on the compute node.	default is 1
resource feature	--gres=gpu:2	Request use of GPUs on compute nodes	default is no feature specified
memory	--mem=131072	Memory limit per compute node for the job. Do not use with mem-per-cpu flag.	memory in 128GB; default limit is 128GM per core
account	--account=group-slurm-account	Users may belong to groups or accounts.	default is the user's primary group.
job name	--job-name="hello_test"	Name of job	default is the JobID
output file	--output=test.out	Name of file for stdout.	default is the JobID
email address	--mail-user=username@iisertvm.ac.in	User's email address	required
access	--exclusive	Exclusive acccess to compute nodes.	default is sharing nodes

b) Monitor the scheduler queue:

```
squeue
```

Above command will display the information about the job in the scheduler queue with its status and other details

```
JOBID PARTITION NAME USER ST TIME NODES NODELIST (REASON)
```

In the above output ST means State of the Job in queue, R means Running, P means Pending, C means Completed and etc

c) Cancel a Job:

```
scancel <jobid>
```

This command will cancel or delete the job

```
scancel -u <username>
```

This command will cancel all the job of user <username>

```
scancel -t PENDING -u <username>
```

This command will cancel all Pending jobs of user <username>

d) Other Scheduler Commands:

```
 scontrol show jobid <jobid>
```

This command will show the full details of the job which is in queue.

```
scontrol show jobid -dd <jobid>
```

This command will show the full details of the job including its job script file which is in the queue.

```
sacct --format=JobID,JobName,MaxRSS,UserCPU,SystemCPU,CpuTime -j
<jobid>
```

This command will show the details of the job which is completed and a day old.

For example:

```
sacct--format=JobID,JobName,MaxRSS,UserCPU,SystemCPU,CpuTime -j 2588

 JobID        JobName      MaxRSS    UserCPU     SystemCPU    CPUTime
---------     ---------    ----------  ----------   ----------    ----------
2588          MA00_5LSt+               09:30:13     02:15.525    09:40:48
2588.batch    batch        4588K       00:00.156    00:00.093    09:40:48
2588.0        pmi_proxy    2020K       09:30:13     02:15.431    00:18:09
```

scontrol hold <jobid>

This command holds the job which is in queue but not running

scontrol resume <jobid>

This command will release the hold job

sstat --format=AvePages,AveRSS,AveVMSize,JobID -j <jobid>

This command will show the statistics of a running job

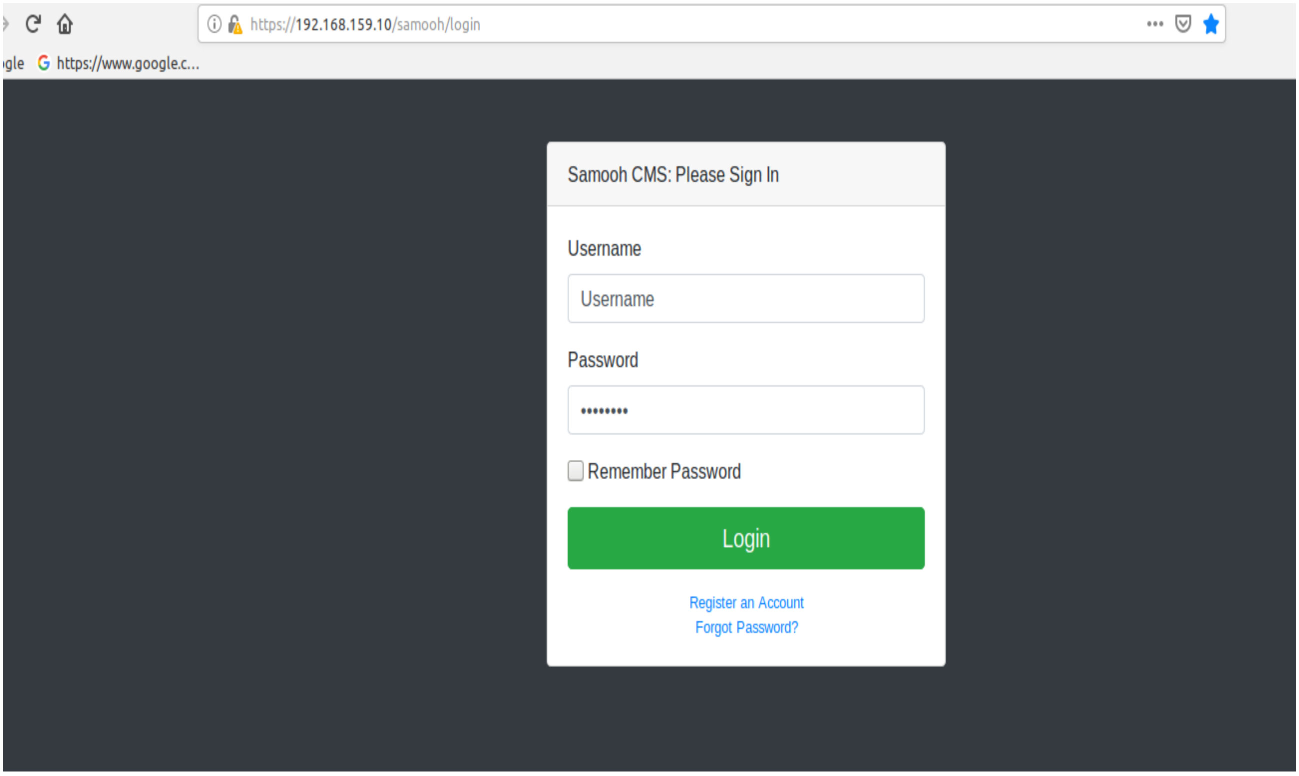
e) Slurm Partition details:

This cluster is partitioned with respect to scheduler to use the resource fairly. The following is the details of the slurm partition information.

S. No	Partition	No. of nodes	No. of cores	Max wall time
1.	cpu	86	2408	120hrs
2.	gpu	3	84 4 NVIDIA P100	120hrs
3.	test	2	56	2hrs
4.	long	86	2408	infinity

3. Samooh CMS – Login

Samooh IP-Address is https://192.168.159.10/samooh/login



https://192.168.159.10/samooh/login

Samooh CMS: Please Sign In

Username

Username

Password

••••••••

□ Remember Password

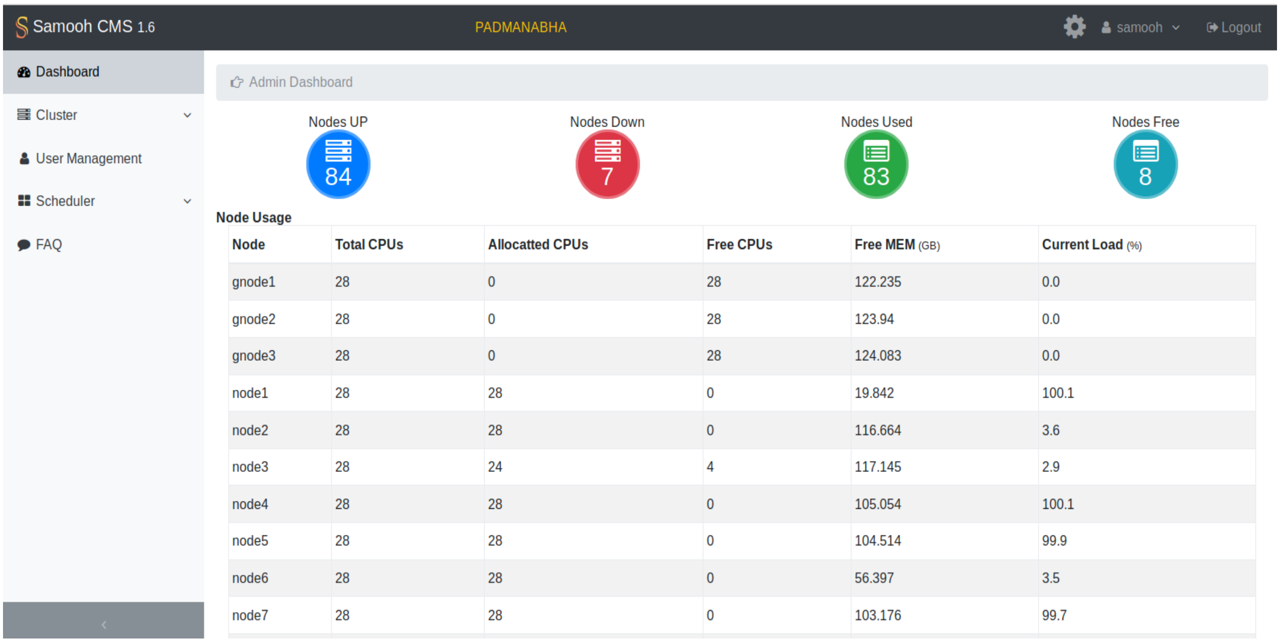
Login

Register an Account

Forgot Password?

a) Job submission portal:

Samooh CMS version 1.6 is web based cluster management suite which provides the job submission and managing portal.



Nodes UP 84

Nodes Down 7

Nodes Used 83

Nodes Free 8

Node Usage

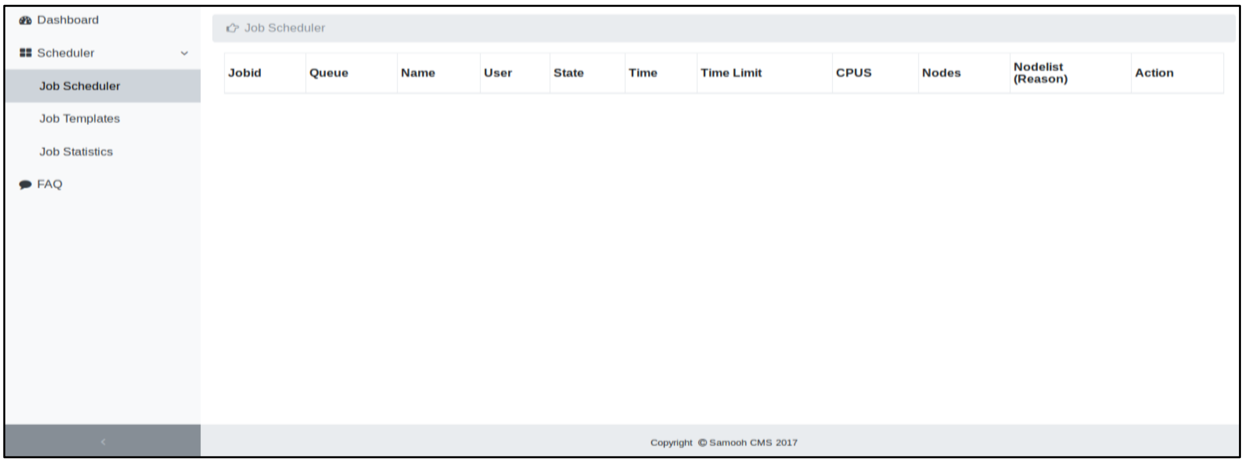
Node	Total CPUs	Allocatted CPUs	Free CPUs	Free MEM (GB)	Current Load (%)
gnode1	28	0	28	122.235	0.0
gnode2	28	0	28	123.94	0.0
gnode3	28	0	28	124.083	0.0
node1	28	28	0	19.842	100.1
node2	28	28	0	116.664	3.6
node3	28	24	4	117.145	2.9
node4	28	28	0	105.054	100.1
node5	28	28	0	104.514	99.9
node6	28	28	0	56.397	3.5
node7	28	28	0	103.176	99.7

b) Dashboard:

Users Dashboard will have Node Usage, Scheduler Partition usage details. Node Usage details include Node name, Total Cpu, Allocated CPU, Free CPU, Free MEM and Current Load of each node. Scheduler Partition usage details includes JobID, Partition Name, Username, Job Name, Job State, Time, Time Limit, CPUs, Nodes, Nodelist, Actions. Actions include Information of job, delete job, hold job and release hold job.

c) Job Scheduler:

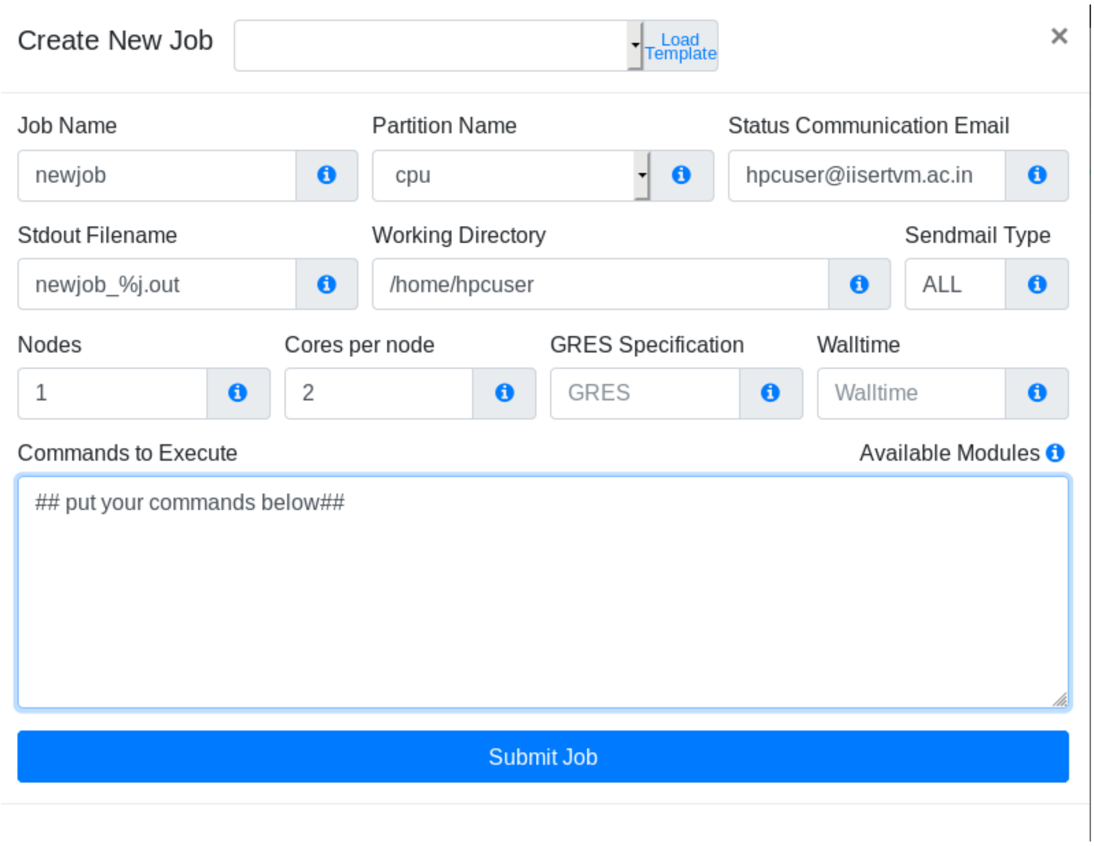
Using this menu user can monitor all their scheduled jobs details like JobID, Partition Name, Username, Job Name, Job State, Time, Time Limit, CPUs, Nodes, Node list. Users also can do the actions include □ to get the information of job, "**X**" delete job, ❚❚ hold job and ▸ release hold job.



Jobid	Queue	Name	User	State	Time	Time Limit	CPUS	Nodes	Nodelist (Reason)	Action

d) Scheduler -> Job Scheduler -> Add Job

Using this menu user can submit new job to users. This will give window to provide necessary details like job name, communication email, stdout file, working directory, mail type, number of nodes, and number of cores, wall time and very importantly job execution commands.



Job Name: newjob

Partition Name: cpu

Status Communication Email: hpcuser@iisertvm.ac.in

Stdout Filename: newjob_%j.out

Working Directory: /home/hpcuser

Sendmail Type: ALL

Nodes: 1

Cores per node: 2

GRES Specification: GRES

Walltime: Walltime

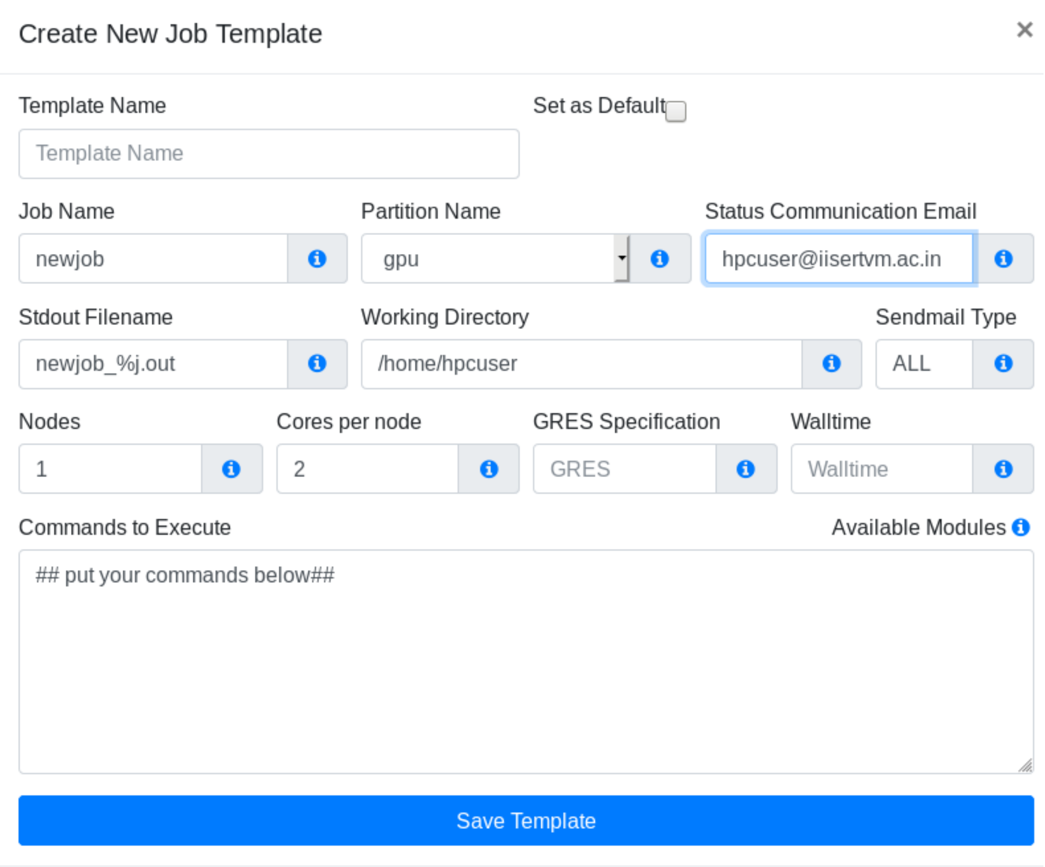
Commands to Execute:

```
## put your commands below##
```

This add new job window has info button at each input, which will give required information about the input field.

e) Job Template:

Job Template is used for saving frequently submitted jobs details to the template show that it can be later used. Due to that its saving lot of re typing of job commands using this menu user can job template for various type of jobs. In this job template user can set important settings of job like job name, communication email, stdout file, working directory, mail type, number of nodes, and number of cores, wall time and very importantly job execution commands.



Create New Job Template

Template Name: Template Name

Set as Default □

Job Name: newjob

Partition Name: gpu

Status Communication Email: hpcuser@iisertvm.ac.in

Stdout Filename: newjob_%j.out

Working Directory: /home/hpcuser

Sendmail Type: ALL

Nodes: 1

Cores per node: 2

GRES Specification: GRES

Walltime: Walltime

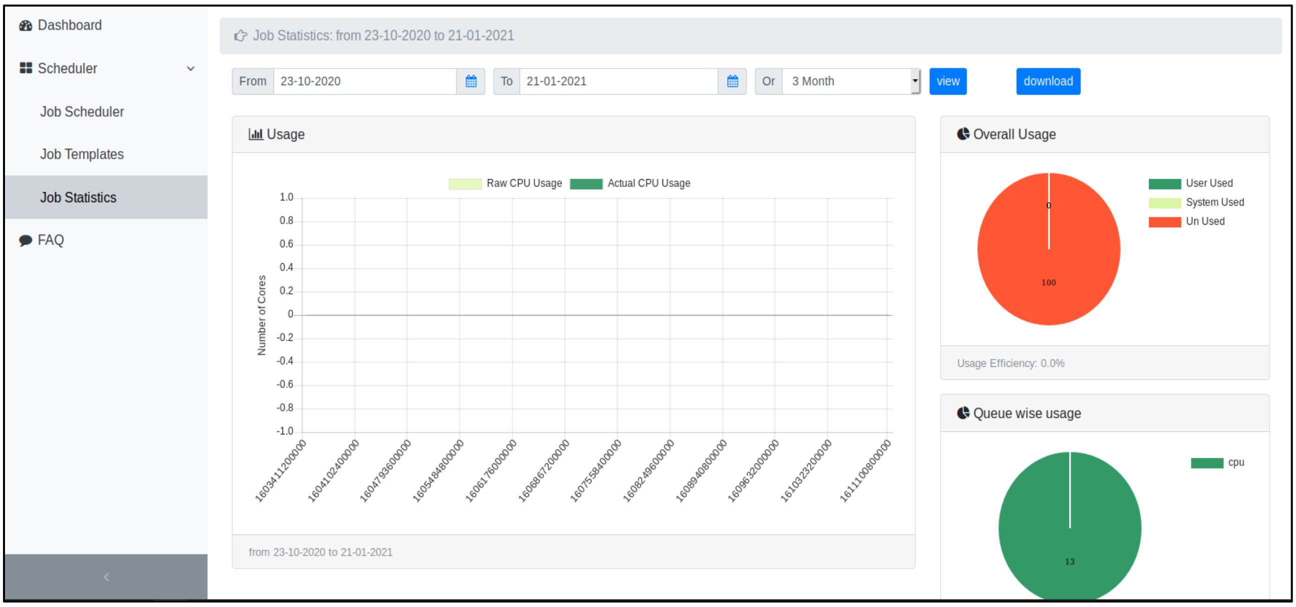
Commands to Execute

Available Modules

```
## put your commands below##
```

f) Job Statistics:

Using this menu user can see their cluster usage statistics as a graph from given date interval. This include Date wise number of core usage, Raw and Actual Usage, Overall Usage, and Queue/Partition wise usage.



Job Statistics: from 23-10-2020 to 21-01-2021

From 23-10-2020 To 21-01-2021 Or 3 Month

Usage Efficiency: 0.0%

from 23-10-2020 to 21-01-2021