

Queuing System in HPCC-SURYA

There are 2 types of queues available in our HPCC-SURYA (1+4)

1. queue1

2. queue2

1. queue1

- * User can submit a **maximum of 2 jobs in running status**, or
- * User can use a **maximum of 48 cores** for their jobs totally.
- * **Minimum core to be used: 24 (ncpus=24)**
- * Maximum wall time: **336:00:00 hrs**

Eg: node=1; ppn = 24 or 48

2. queue2

- * User can submit a **maximum of 2 jobs in running status**, or
- * User can use **maximum of 48 core** for their jobs totally.
- * **Minimum core to be used: 24 (ncpus=24)**
- * Maximum wall time: **336:00:00 hrs**

Eg: node=1; ppn = 24 or 48

Introduction:

In SRMAP HPCC, we are using PBS Pro – Portable Batch System Professional. It is designed to manage the distribution of batch jobs and interactive sessions across the available nodes in the cluster.

Batch System:

The batch system allows the users to submit their jobs by requesting the resources (nodes, processors, memory) that they need. PBS Pro is handling these jobs on a First-Come, First-Served basis.

Login (Linux):

If you are using Linux os, then simply open you're Terminal, type
ssh username@hostname

Example Login page:

```
Last failed login: Mon Dec 19 10:08:03 IST 2022 from 10.1.56.70 on ssh:notty
There were 4 failed login attempts since the last successful login.
Last login: Fri Dec 16 16:42:03 2022 from 10.1.56.62
[ranjit@hpcmaster2 ~]$ cat
```

Submitting a job:

All users must submit your job by using queue system only using qsub.

Example for a script file: qsub.sh

```
-----
#!/bin/bash
#PBS -N fepc7md
#PBS -l nodes=1:ppn=24
#PBS -q queue2
#PBS -j oe
#PBS -V
cd $PBS_O_WORKDIR
cat $PBS_NODEFILE > ./pbsnodelist
CORES=`cat ./pbsnodelist|wc -l`
#source /apps/intel/parallel_studio_xe_2018.4.057/psxevars.sh intel64

module load vasp6
source /opt/intel/oneapi/setvars.sh

cat /home/ranjit/vasp-pseudo/PAW_PBE/Fe/POTCAR > ./POTCAR
#mpirun -np $CORES /home/ranjit/code/vasp.5.4.1/bin/vasp_std > stdout.txt

mpirun -np 24 vasp_std > stdout.txt
-----
```

To check the command under pbs_pro

Use the command >> man pbsnodes

```
#!/bin/bash -Specifies which shell program to use, is mandatory and does not change.
#PBS -N <job name> -Specifies the name of the job that will appear in the job queue.
#PBS -q <queue name> -Specifies that the job should be run in the named queue.
#PBS -l select=X: ncpus=Y: mpiprocs=Z -X = Requesting "n" number of node
-Y = Requesting "n" number of cores for a node.
-Z = It takes number of MPI process per node.
```

#PBS -j oe -Specifies PBS to join standard output and standard error together in the output file.

#PBS -V -Exports Users Environmental Variables to Execution Host.

PBS_O_WORKDIR - Contains the name of the directory from which the user submitted the PBS job.

PBS_NODEFILE -Name of the file that contains a list of the HOSTS provided for the job.

Once job submission script is ready to submit, and then use the command qsub to submit a job to the queuing system.

#qsub your_script.sh

Example: qsub qsub.sh

After submitting a job, an ID will be generating for that particular job. We can see that job ID by using a qstat command.

To Display the available job queue:

qstat -q -It will display all available queue for job.

Deleting Job:

A queuing/running job can be delete by using following command,

#qdel <jobID>

Example:

#qdel 238

(Where 238 is the job ID, that can be obtained from qstat command)

```
[ranjit@hpcmaster2 chk]$ qmgr -c 'p s'
#
# Create queues and set their attributes.
#
#
# Create and define queue workq
#
create queue workq
set queue workq queue_type = Execution
set queue workq enabled = False
set queue workq started = False
#
# Create and define queue queue1
#
create queue queue1
set queue queue1 queue_type = Execution
set queue queue1 max_queued = [u:PBS_GENERIC=1]
set queue queue1 resources_max.walltime = 336:00:00
set queue queue1 acl_group_enable = True
set queue queue1 acl_groups = queue1
set queue queue1 max_run = [u:PBS_GENERIC=2]
```

```
set queue queue1 max_run_res.ncpus = [u:PBS_GENERIC=48]
set queue queue1 enabled = True
set queue queue1 started = True
#
# Create and define queue queue2
#
create queue queue2
set queue queue2 queue_type = Execution
set queue queue2 max_queued = [u:PBS_GENERIC=1]
set queue queue2 resources_max.walltime = 336:00:00
set queue queue2 acl_group_enable = True
set queue queue2 acl_groups = queue1
set queue queue2 acl_groups += queue2
set queue queue2 max_run = [u:PBS_GENERIC=2]
set queue queue2 max_run_res.ncpus = [u:PBS_GENERIC=48]
set queue queue2 enabled = True
set queue queue2 started = True
```