SRM SUPERCOMPUTING CENTER

INTRODUCTION TO HPCC JOB SUBMISSION

Introduction:

In SRM 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:

```
[root@localhost Desktop]# ssh hpcuser@172.16.18.141
hpcuser@172.16.18.141's password:
Last login: Wed May 10 12:10:03 2017 from 10.1.43.100
[hpcuser@hn1-srmhpc03 ~]$
```

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 Pt24Co12Ni19-pseudo
#PBS -q work-02
#PBS -l select=1:ncpus=28:mpiprocs=28
#PBS -j oe
#PBS -V
cd $PBS_O_WORKDIR
cat $PBS_NODEFILE > ./pbsnodelist
CORES=`cat ./pbsnodelist|wc -l`
source /opt/software/intel/parallel_studio_xe_2017.0.035/psxevars.sh  intel64

mpirun -np 28 projwfc.x < Pt31Co12Ni12-PDos.in |tee Pt31Co12Ni12-PDos.out
```

#! /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 **o**utput and standard **e**rror 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.

```
[root@hn1-srmhpc03 ~]# qstat
Job id            Name             User             Time Use S Queue
----------------  ---------------- ----------------  -------- - -----
59.hn1-srmhpc03   vasp             hpcuser           00:00:00 R work-01
60.hn1-srmhpc03   vasp             hpcuser           00:04:39 R work-01
```

(Here 59 and 60 are the job ID)

To Display the available job queue:

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

```
[root@hn1-srmhpc03 ~]# qstat -q

server: hn1-srmhpc03

Queue            Memory CPU Time Walltime Node   Run   Que   Lm  State
---------------- ------ -------- -------- ---- ----- ----- ----  -----
res-4              --      --    720:00:0  --      0     0   --   E R
res-2              --      --    720:00:0  --      0     0   --   E R
workq              --      --       --     --      0     0   --   D S
work-01            --      --    720:00:0  --      1     0   --   E R
                                                ----- -----
                                                    1     0
```

Queuing System:

There are 3 types of queue available,

- res-4
- res-2
- work-01

Their corresponding maximum Walltime will be 30 days for each 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)