Some Very Under Done Instructions for HPC 2013

These instructions are only for a user with some experience. You need proficiency in Linux and parallel programming. Some details for the HPC2013 Cluster are in the other documents. In case of problems write to hpcsupport@iitk.ac.in. You will also be added to a list hpc@lists.iitk.ac.in . There may be some teething problems. For people used to the older cluster please pay attention to I_MPI_FABRICS in the scripts provided.

1. You can login to the login nodes of HPC2013 by **ssh –X <username>@hpc2013.hpc.iitk.ac.in** Your password is the same as that for CC. We have 888 regular nodes with 20 cores (128 GB RAM) and five nodes with high memory (768 GB RAM).
2. **Always, after you logon to the cluster go to the workq by typing**

   qsub –I

   After you have finished, type

   exit

   workq is an interactive queue that places you on a node where you can run commands. **You cannot access /opt/software otherwise.**
3. You can change your password on a CC machine but not on the cluster.
4. The changed password will be effective on the cluster within an hour.
5. We have created a home directory for you which will be initially empty. The path for this directory is /home/<username>/.
6. We have also created a /scratch/<your-name>/ directory that is a temporary directory. There is also a directory /scratch/<your-name>/largefiles/ for storing files larger than 250MB. Do not use the largefiles directory for storing smaller files for performance purposes. The /scratch is faster than /home so you may prefer to write out temporary results and carry out computation here. /scratch contents can be deleted at any point of time and if not in use. All software are in /opt/software.
7. There is no backup and you are responsible for taking regular backup of your area.
8. Please read the structure of the queues given below
9. Please use the cluster in a sensible manner, and follow the rules of engagement, otherwise you may land up causing problems to others.

Currently you can test your programs with the Intel compiler. Here is how.

There is a file in /opt/software/intel/initpaths.

- ➢ To add the intel **32 bit** compilers you have to type
  - ❖ source /opt/software/intel_2019.u0/initpaths **ia32**
- ➢ To add the intel **64 bit** compilers you have to type
  - ❖ source /opt/software/intel_2019.u0/initpaths **intel64**

If you want to do special tuning for trace analyzer then the second argument has to be special but you will have to do your own research on this. The commands above just use the default analyzer. Please read the Intel site documentation for details. **A common mistake is using the programs compiled on one cluster directly on another cluster. You need to recompile programs if clusters are changed.**

After you have **sourced** the files you should compile your programs using the relevant programs such as mpiicc, mpicc etc. Confusion in PATH settings is one of the main sources of error.

Use the following in a file say “test” for submitting programs. Remember to do a chmod 755 to the file. Change this file as per nodes and queue required or for the job name. You can change some variable names such as the name of the queue and job in example file given below. Number of nodes should change with the queue. You should always keep ppn 20 except for “hyperthread” queue where ppn is 40 and workq parallel job where it is 4. The hyperthread queue is an experimental queue and may give better results than normal. If this is the case then please do inform us. Make it less only in exceptional circumstances and do not make it more. Even then restrict yourself to the number of nodes limit in any queue.

Script for running job on small/medium/large queue

Job Script “test”

```
#!/bin/bash
#PBS -N <sample job name>
#PBS -q small
#PBS -l nodes=1:ppn=20
#PBS -j oe
cd $PBS_O_WORKDIR
export I_MPI_MPD_TMPDIR=/scratch/<your_name>
source /opt/software/intel_2019.u0/initpaths intel64
mpirun -machinefile $PBS_NODEFILE -np 20 ./<64bit compiled program>
```

- Now use can submit the job named test using : *qsub test*
  - This will give you a number for your queued job.
  - All output and errors will go to test.<number_you_get>.
- You can check the status of your jobs by using : *qstat –u <your_login_name>*
- You can delete your job by using : *qdel – W force <your job number>*

See man qsub.

A short description of the queues is below. Here workq should be an interactive queue as well as a batch queue and remaining queues are only batch queues. There may be some discrepancy in this functionality as numbers of days etc. are a policy decision.

Queue Name	**Wall-time**	**Max jobs run simultaneously**	**Min cores**	**Max cores**	**Min nodes**	**Max nodes**	**Total nodes**
workq	24 hours 2 hrs CPU time	2 (1 login + 1 testing)	1	6	1	1	4
small	5 days	3 running 1 waiting	20	40	1	2	96
medium	4 days	3 running 1 waiting	40	120	2	6	256
large	4 days	2 running 1 waiting	120	640	6	32	482
hyperthread (each node behaves as if it has 40 cores)	5 days	1 running 1 waiting	40	80	1	2	16
highmem (for large memory jobs)	5 days	1 running 1 waiting	2	20	1	1	5
mini (for jobs of small duration)	2 hours	1 running 1 waiting	20	40	1	2	32
courses	2 hours	1 running 1 waiting	20	40	1	2	All spare

If you do not see /opt/software when you login, it is deliberate. **You must use qsub -I**

- **Script for submitting parallel job on workq**

```
#!/bin/bash
#PBS -N <sample job name>
#PBS -q workq
#PBS -l ncpus=6
#PBS -j oe
cd $PBS_O_WORKDIR
source /opt/software/intel_2019.u0/initpaths intel64
export I_MPI_MPD_TMPDIR=/scratch
mpirun -machinefile $PBS_NODEFILE -n 6  ./<64bit compiled program> | tee <test.txt>
```

- **Script for submitting a SEQUENTIAL JOB**

```
#! /bin/bash
#PBS -l nodes=1:ppn=1
#PBS -N <a_name_for_your_job>
#PBS -q seq
#PBS -j oe
cd $PBS_O_WORKDIR
<fully_qualified_name_of_your_executable>
```

At a time 20 such jobs may be submitted as you are allotted one node for a sequential run.

- **Script for submitting ANSYS FLUENT JOB**

Sample Job Script	Sample Input File "*batch_fluent1.jou*"
#PBS -l nodes=2:ppn=20 #PBS -q mini #PBS -V #PBS -l fluent_user=1 #PBS -l fluent_lic=20 cd $PBS_O_WORKDIR /opt/software/ansys22R1/v221/fluent/bin/fluent -cflush 3d -g -cnf=$PBS_NODEFILE -t40 -i batch_fluent1.jou	file set-batch-opt y y y n file start-transcript output.trn file read-case elbow1.cas.gz solve init init-flow solve it 10 file write-case-data test_%i.gz file stop-transcript exit

- **Script for submitting a parallel MATLAB PCT JOB**

Sample Job Script "matlabr2020a.sh"	Sample Input File "*matlabr2020a_wave.m*"

#!/bin/bash
#PBS -N M2020a
#PBS -l nodes=1:ppn=20
#PBS -q mini
#PBS -j oe
echo "I ran on:"
cd $PBS_O_WORKDIR
matlab -nodisplay -r matlabr2020a_wave -logfile matlabr2020a.log
	p=parpool('local',20);
	parfor i=1:4096
	A(i) = sin(i*2*pi/4096);
	end
	delete(p);
	plot(A)
	print ('-r75','-djpeg','matlabr2020a_plot.jpg')
	quit

For running distributed MATLAB please meet HPCGroup in CC 212.

- **For submitting a GAUSSIAN Job**
  First prepare your input file molecule.com and then type
  submitLinda (For Gaussian 9)
  or
  submitLinda16 (For Gaussian 16)
  and follow the instructions.

The list of all possible scripts would be long but the domain experts, which you as user are supposed to be, should figure it out. You can install software in your own directories and in no-case would you require root privileges for installing the software.

For transferring files from HPC2013 and general warnings

1. Use your favorite sftp program such as winscp sftp etc.
2. Connect to hpc2013.hpc.iitk.ac.in
3. Transfer files
4. exit
5. In no case use the node to which you login for this purpose
6. **Please do not submit jobs on the node to which you logon**