

## Some Very Under Done Instructions for HPC 2010

These instructions are only for a user with some experience. You need proficiency in Linux and parallel programming. HPC 2010 has 2 clusters: a Nehelam based 8 core cluster and a Sandy Bridge (SB) based 16 core cluster. These instructions apply only to the Nehelam based cluster since this is the oldest cluster. Some details for the SB Cluster are in the other document titled Sandy Bridge cluster circulated to you. In case of problems write to [hpcsupport@iitk.ac.in](mailto:hpcsupport@iitk.ac.in) . You will also be added to a list [hpc@lists.iitk.ac.in](mailto:hpc@lists.iitk.ac.in) .

1. You can login to the master node by `ssh -X <username>@hpc2010.hpc.iitk.ac.in` Your password is the same as that for CC. We have 368 nodes with 8 cores.
2. Always, after you logon to the cluster go to the workq by typing  
`qsub -I -X`  
After you have finished, type  
`exit`  
workq is an interactive queue that places you on a node where you can run commands.  
Use `-X` for X windows support.
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. `/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. All software are in `/opt/software`
7. Please read the structure of the queues given below
8. There is no backup and you are responsible for taking regular backup of your area.
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:

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

To run the intel **32 bit** compilers you have to type

```
source /opt/software/intel/initpaths ia32
```

To run the intel **64 bit** compilers you have to type

```
source /opt/software/intel/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. The `source` command sets these paths.

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 8. 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

```
#!/bin/bash
#PBS -N <sample job name>
#PBS -q small
#PBS -l nodes=1:ppn=8
#PBS -j oe
cd $PBS_O_WORKDIR
export I_MPI_FABRICS=shm:tmi
export I_MPI_DEVICE=rdma:OpenIB-cma
export I_MPI_MPD_TMPDIR=/scratch
mpirun -machinefile $PBS_NODEFILE -n 8 ./<64bit compiled program>
```

Now use qsub test to submit the job.

```
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

```
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	walltime	Max jobs run simultaneously	Max cores	Min cores	Max nodes	Min nodes	Total Nodes
workq	2 days (interactive debugging testing) 2 hrs CPU time	3 (1 login + 2 testing) This may change	6	1	6	1	6
small	5 days	1 running 1 waiting	32	2	4	1	59
medium	4 days	1 running 1 waiting	96	32	12	4	100
large	3 days	1 running 1 waiting	256	128	32	16	184
sequential (1 user per node max 36 nodes)	6 days	1 user per node	8 cores	1 core	1	1	8

### Script for running job on WORKQ

```
#!/bin/bash
#PBS -N <sample job name>
#PBS -q workq
#PBS -l ncpus=6
#PBS -j oe
export I_MPI_FABRICS=shm:tmi
export I_MPI_DEVICE=rdma:OpenIB-cma
cd $PBS_O_WORKDIR
export I_MPI_MPD_TMPDIR=/scratch
mpirun -machinefile $PBS_NODEFILE -n 6 ./<64bit compiled program> | tee <test.txt>
```

### Script for running 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>
```

### Script for submitting ANSYS FLUENT JOB

```
# Fluent-PBS Startup Script: (Fields to be edited are marked in RED)
#!/bin/bash
#PBS -l nodes=2:ppn=8
#PBS -l fluent=1
#PBS -l fluent_lic=16
#PBS -q small
#PBS -V
#EXECUTION SEQUENCE
cd $PBS_O_WORKDIR
/opt/software/ansys_inc/v150/fluent/bin/fluent 3ddp -g -cnf=$PBS_NODEFILE -t16 -i
batch_fluent.jou
```

### Script for submitting a parallel MATLAB PCT JOB

```
#!/bin/bash
#PBS -N M11
#PBS -l nodes=1:ppn=8
#PBS -l matlab_user=1
#PBS -l matlab_lic=8
#PBS -q small
#PBS -S /bin/bash
#PBS -j oe
echo "I ran on:"
cd $PBS_O_WORKDIR
cat $PBS_NODEFILE
export PATH=/opt/software/matlab11b/bin:$PATH
matlab -nosplash -nodisplay << !!
matlabpool open local 8
your_program
matlabpool close
!!
```

### **For running a GAUSSIAN Job**

First prepare your input file molecule.com and then type  
submitLinda  
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 running distributed MATLAB please meet HPCSUPPORT in CC 212

### **For transferring files from HPC2010 and general warnings**

1. Use your favorite sftp program such as winscp sftp etc.
2. Connect to hpc2010.hpc.iitk.ac.in
3. Transfer files
4. Exit
5. **Please do not submit jobs on the node to which you logon**