

# Using the IBM eServer Cluster at Cyl

## A Bit About The Cluster

The cluster is composed of:

- Management node (login machine),
- 24 compute nodes, each with dual Intel 5440 2.83GHz Quad core processors and 16GB RAM.
- There is a 53TB GPFS storage array that appears as a locally mounted device on each of the compute nodes.
- The OS on all machines in the cluster is RHEL4.5.
- Access for running jobs is controlled by a queuing system (Torque) with a scheduler (Maui).

## Logging In

The cluster is accessible via ssh from anywhere.

```
ssh username@planck.cyi.ac.cy
```

The default shell is bash. If you wish to use a different shell, or you wish to change some of your environment and you are having difficulties, please contact [hpcsupport@cyi.ac.cy](mailto:hpcsupport@cyi.ac.cy)

## /home and /gpfsFS1

When you login to the system, you will be located in your home directory, /home/username.

Please note that the /home directories are quite small. You also have a directory in /gpfsFS1/username. The /gpfsFS1 partition is a parallel file system that appears locally connected on all the nodes of the cluster, so anything you store there is globally available. It is also quite large (>50TB). This is the area you should use for large input files and to generate output files from your jobs.

**Please do not generate output to /home/username as you will easily fill up the partition and prevent others from being able to work on the cluster.**

## Queuing System

The cluster is a shared resource, used by several researchers simultaneously. To ensure that usage is fair, the cluster resources are managed by a queuing system. We use the Torque queuing system with the Maui scheduler to determine which jobs to run and in what order.

The queuing system requires that you submit your job through a shell-script. There are example shell-scripts for sequential and parallel jobs at the end of this document (with some explanations). If you are having difficulty constructing a shell-script for your job to run properly, please contact [hpcsupport@cyi.ac.cy](mailto:hpcsupport@cyi.ac.cy) for assistance.

Once you have a working script for your job (`myjob.pbs`), you need to submit it to the queuing system. This is done with the **qsub** command:

```
[patrickf@mgt ~]$ qsub myjob.pbs
```

This command should respond with a jobid, such as `6705.mgt`. This is the unique identifier for your job in the queue.

Once you have submitted a job you will want to check on its progress in the queue. Use the **qstat** command for this:

```
[patrickf@mgt ~]$ qstat
```

This command should respond with a table, where you can check on your job's progress through the queuing system.

Job id	Name	User	Time Use	S	Queue
-----	-----	-----	-----	-	-----
6703.mgt	xmecham5	tost	00:00:15	R	dque
6705.mgt	testjob	patrickf	00:01:15	R	dque

Finally, there are times that you may wish to stop a job that doesn't seem to be performing correctly, to prevent it continuing to consume resources. Use the **qdel** command for this:

```
[patrickf@mgt ~]$ qdel 6705.mgt
```

There should be no response to this command, other than to terminate your job and free up the resources. This may take a minute or two to occur. You can only delete your own jobs.

As with most unix/linux based open source applications, these commands can take many options to vary the way they work. These will be documented in the online man-pages, which can be accessed from the command line with, for example `'man qstat'`. The queuing system has many commands, some of which can be used by users. Try `'man -k pbs'` to see a list of man-page topics about queuing system commands.

If you are still having difficulties, please contact [hpcsupport@cyi.ac.cy](mailto:hpcsupport@cyi.ac.cy).

## Compilers and system libraries:

The cluster has the following compiler suites available:

- Gnu gcc compiler suite (gcc, g++, g77), version 3.4.6  
Gnu gcc compiler suite (gcc4, g++4, gfortran), version 4.1.1
- Intel compiler suite (icc, icpc, ifort), version 10.1
- Lahey fortran compiler (lf95), version L8.10a

## MPI

There are several implementations of MPI available on the cluster.

- mvapich, version 1.0.1
- mvapich2, version 1.0.3
- openmpi, version 1.2.6

Compiled versions of these implementations exist for each compiler suite. To access a particular implementation you will need to put the version name in a file called `.mpi-selector` in your home directory, eg.

```
echo openmpi_gcc-1.2.6 > ~/.mpi-selector
```

will give access to openmpi that has been compiled against the gnu compilers.

If you need more help, please contact [hpcsupport@cyi.ac.cy](mailto:hpcsupport@cyi.ac.cy)

## Sample Shellscripts:

Shellscripts for the queuing system take the following form:

- Section 1  
The first part of the shell-script has a series of lines beginning with `#PBS`. These are commands directly to the queuing system and contain information about your job, such as a descriptive short name (`#PBS -N`), instruction to join the queuing system output and error files into one (`#PBS -j oe`) and which queue to submit the job to (`#PBS -q`).  
Most importantly, information about which resources are required for the job is passed to the queuing system (`#PBS -l`). It is essential to request at least the number of processors required and the amount of time to use them for. The queuing system will automatically start your job when these resources are available and it is your turn.  
Information about other resources that can be requested can be found in the `qsub` man-page. These queuing system commands must occur altogether at the beginning of the shell-script
- Section 2  
The second part is just a normal shell-script and can contain whatever commands you wish. These will be executed on the compute node as if they were typed at the command line. You have access to some extra environment variables, provided by the queuing system for this job only. Two of these in particular will be useful:  
`$PBS_NODEFILE` – contains a list of the processors your job has been allocated  
`$PBS_O_WORKDIR` – contains the path to the directory from which your job was submitted

Below are 2 sample shell-scripts for submitting jobs to the cluster. These are very simple examples, just to get you started. If you need more help, please contact [hpcsupport@cyi.ac.cy](mailto:hpcsupport@cyi.ac.cy).

## Sequential job

```
## First Section
## defines a job called 'myjob' that asks for 1 processor, with
## a maximum memory requirement of 1GB, that will run for up to
## 1 hour on the queue dque.
## Any output and error files produced by the queuing system will
## be joined together (useful for debugging problems)

#PBS -N myjob
#PBS -j oe
#PBS -q dque
#PBS -l nodes=1:ppn=1,mem=1GB,walltime=01:00:00

## Second Section
## changes to the directory from where the job was originally
## submitted, defines some useful local variables and then runs
## the executable, redirecting STDIN, STDOUT and STDERR

EXECUTABLE=/path/to/executable
INPUT_DATA=/gpfsFS1/username/path/to/input/data
OUTPUT=/gpfsFS1/username/path/to/output

cd $PBS_O_WORKDIR

$EXECUTABLE < $INPUT_DATA >& $OUTPUT
```

## Parallel job using MPI

```
## First Section
## defines a job called 'myparjob' that asks for 32 processors (4 nodes
## each with 8 processors), with a maximum memory requirement of 32GB
## for the whole job, that will run for up to 6 hours on the queue dque.

#PBS -N myparjob
#PBS -j oe
#PBS -q dque
#PBS -l nodes=4:ppn=8,mem=32GB,walltime=06:00:00

## Second Section
## changes to the directory from where the job was originally
## submitted, works out how many MPI processes to start, defines some
## useful local variables and then runs the executable under mpirun
## (openmpi here), redirecting STDIN, STDOUT and STDERR

EXECUTABLE=/path/to/executable
INPUT_DATA=/gpfsFS1/username/path/to/input/data
OUTPUT=/gpfsFS1/username/path/to/output

cd $PBS_O_WORKDIR
NP=$(cat $PBS_NODEFILE | wc -l)

mpirun -np $NP $EXECUTABLE < $INPUT_DATA >& $OUTPUT
```