HPC Service Introduction

Logging in

In order to access the services, which are Linux based, you will need to have some kind of ssh (secure shell) software installed on the PC or system which you use to access our service. If that system is Linux or Unix based then it is most likely that the ssh and scp commands will be available by default. If you have a Windows based PC you will need extra software such as eXceed or Cygwin or putty to provide this functionality.

There are 3 services, the PC Cluster (cx1), the massively parallel system (cx2) and the very large shared memory system (ax3). Since these systems have very different architectures and are not fully binary compatible, you should log into one of the hosts below to access the service you want to use. For your convenience, the **HOME** directories are shared between the 3 systems. However, if you plan to use both services, it would be wise to have separate directories for the binaries for each. Most users will use the cx1 system. Access to cx2 and ax3 is only granted on demonstration of appropriate use and need. Both cx2 and ax3 are considerably more expensive per CPU than cx1.

For the HPC PC Cluster (cx1):
 ssh to **login.cx1.hpc.ic.ac.uk** using standard college login credentials.
For the massively parallel system SGI Altix ICE (cx2):
 ssh to **login.cx2.hpc.ic.ac.uk**
For the very large shared memory system SGI Altix UV (ax3):
 ssh to **ax3.hpc.ic.ac.uk**

Environment

On the HPC system, there are two file stores available to the user: **HOME** and **WORK** . HOME has a relatively small quota of 15GB and is intended for

storing binaries, source and modest amounts of data. It should not be written to directly by jobs.
WORK is a larger area which is intended for staging files between jobs and for long term data storage. WORK has an initial quota of 150GB.
These areas should be referred to using the environment variables **$HOME** and **$WORK** as their absolute locations are subject to change.
Additionally, jobs running within the queuing system have scratch space available in the directory location stored in the variable **$TMPDIR**. Jobs requiring scratch space at run time should write to **$TMPDIR**.
Jobs running under the queuing system have their working directory set to **$TMPDIR**. This means that temporary and scratch files are on fast disk and are automatically deleted, and problems with multiple jobs interfering with each others' working files are avoided.
You should run your own scripts or commands by either:

1) Installing the commands into a directory which is in your path, or
2) Using the longer pathname to the script, eg. **$HOME/progs/runprog**

Packaged Applications

We use the program **module** to manage the applications that we provide. Module performs the shell environment manipulation (eg: setting PATH, LD_LIBRARY_PATH) . To see a list of supported applications, issue **module avail**:

```
$ module avail

--------------------------- /apps/modules/modulefiles ---------------------------
ant/1.6.1                       hdf5/1.6.5-parallel
ant/1.6.5(default)              hdf5/1.6.5-serial
castep/3.1.1(default)           intel-suite/8.1
cmake/2.2.3(default)            intel-suite/9.0(default)
dl_poly/2.15                    java/ibm-jre-1.4.2
dl_poly/3.05-lfv                java/sun-jdk-1.4.2_05(default)
dl_poly/3.05-vv(default)        lammps/10Nov2005(default)
dot                             lammps/1999
eclipse/3.1.2(default)          module-info
emboss/3.0(default)             modules
fftw/2.1.5-double               mpiblast/1.4.0(default)
fftw/2.1.5-single               namd2/2.6b1(default)
fftw/3.1.1-double               ncbitools/1-3-2006(default)
```

fftw/3.1.1-single	ncbitools/20-10-2004
gamess-uk/7.0-parallel	null
gamess-uk/7.0-serial	nwchem/4.7(default)
gaussian/g03(default)	siesta/1.3(default)
globus/3.2.1	siesta/1.3-parallel
gpt/3.0.1	siesta/1.3-serial
gromacs/3.3(default)	use.own
gulp/1.3.2(default)	vtune/8.0(default)

To load a module, issue **module load** *module-name*. eg:

module load intel-suite loads the intel compiler module marked default

module load intel-suite/8.1 loads the intel compiler version 8.1

To see what modules are presently loaded, issue **module list**.

To get information about a specific module issue **module help** *module-name*

Compiling

The Intel Fortran and C/C++ compilers are available in the module **intel-suite** . The binaries are called ifort and icc . This module also includes the Intel optimized math kernel libraries (mkl), the Intel debugger idb and the vtune performance optimization package.

Most of the packaged applications are compiled against the math kernel libraries (eg **games-uk/7.0-parallel**) and so will need the intel-module loaded as a prerequisite. MPI programs also need the intel-suite module loaded.

The gnu compilers should not be used on either system as the Intel compilers generate faster code.

Queuing System

The HPC system uses the PBS Pro queuing system to manage the execution of jobs on the compute resources. All jobs must be run through the queuing system and not directly from the command line.

To submit a job, the command **qsub** is used. qsub requires an input script which describes the job. Eg:

```
#!/bin/sh
#PBS -l walltime=1:00:00
```

```
#PBS -l mem=600mb
#PBS -l ncpus=4

module load gaussian
g03 < ${HOME}/input
```

The first three lines give information to PBS about the job characteristics, in this case: maximum runtime, memory required, number of CPUs required. The remainder of the script is executed as a normal shell script. In this case, the Gaussian module is loaded and Gaussian run with an input script residing in the user's home directory.

To run an MPI job, use **mpiexec *binary-name***. mpiexec will automatically pick up the process count from PBS.

To run an OpenMP job on ax1, use **ompexec *binary-name***. ompexec will automatically pick up the process count from PBS. On cx1, the binary may be run directly.

NB. For jobs to be run on the PC cluster system, there is a requirement for you to code the node layout in your job submission.
Simply using -l ncpus=<m> will request a single node with <m> CPUs.
If you don't specify the number of CPUs, it will default to 1 ie. It is a serial run using only one processor.

For MPI parallel jobs on the cluster, most of the nodes have 4 CPUs per node, you should request CPUS in chunks of the nodesize (eg 12), using this format of select directive:
#PBS –l select=<m>:ncpus=n
Where <m> is the number of nodes you will need, each with n CPUs (ie. m x n cpus in total)

Eg: for a 48 CPU job, using 12 cpu nodes, you should code:
#PBS –l select=4:ncpus=12

For 72 CPUS, code:
#PBS –l select=6:ncpus=12

The largest parallel jobs normally run on cx1 are 96 CPUs, which is 8 nodes with 12 cores.

You may need to add a mem component to this line, this should be the memory requirement per node, eg. 3200mb, which gives 800mb per CPU
#PBS –l select=2:ncpus=12:mem=3200mb

For smaller parallel jobs, we have nodes with up to 16 cores, so smaller parallel runs can be made very efficiently on single nodes. Eg:
For 8 cpus, code
#PBS –l select=1:ncpus=8
For 12 cpus, code

#PBS –l select=1:ncpus=12
For 16 cpus , code
#PBS –l select=1:ncpus=16

If the submission is successful qsub returns the ID of the job.

Eg:

$ qsub input
160.ax1

In this case, the standard out and standard error are directed by PBS to the files input.o160 and input.e160 (more generally *input-name*.[o|e]*job-id*)

Note: there is no need to specify an explicit queue name as an argument to qsub- PBS will automatically decide the most appropriate queue for the job based on the resource requirements given in the '#PBS –l' directives. Directly submitting to a named queue will always fail.

The command **qdel *job-id*** should be used to abort a queued or running job.

The command **qstat** shows the list of all jobs running or queued.

All PBS commands have man pages giving further information.

Cluster issues

When using the cluster, you should note the following:

- The MPI libraries must be explicitly loaded with **module load mpi**
- User environment variables will not be accessible by a job running under PBS
- ***All jobs must be run under the queuing system***. Do not log directly into nodes!