IT Services – CSF Quick Start Guide

Authors: *Jonathan Boyle, George Leaver*
Version: *1.3*
Date: *September 2012*

1. Syntax Used in the Instructions

Text in **`bold monospace font`** represents commands to be typed in exactly. Be careful to copy all spaces and characters correctly, and to enter commands on a single line.
Text in ***Bold-Italics*** should be replaced as described.

2. Unix/Linux operating systems

Wiki page: wiki.rcs.manchester.ac.uk/community/UNIX

The CSF uses the Linux operating system, which is a Unix-like operating system.

3. To connect to the Computational Shared Facility (CSF)

Wiki page: wiki.rcs.manchester.ac.uk/community/csf/UserInfo

1. Ensure your training desktop is using the **Linux** operating system
2. Open a terminal window
3. Enter the following line at the terminal window prompt, replacing ***UserID*** with your University id (used to log in to your email, for example), and press enter
   `ssh -X` ***`UserID`*****`@csf.itservices.manchester.ac.uk`**
   Note: **`X`** is uppercase X.
4. If asked "Are you sure you want to continue connecting (yes/no)?" enter **`yes`**
5. Enter your password when asked. If you have not previously been given a CSF password this will be provided by your tutor. Note: non-CSF users will be given an account that remains active during the current round of training courses; these will be closed on 14th December 2012.
6. Enter **`hostname`** to ensure you have connected to the CSF. The response should be **login2**, if you do not see this please tell your tutor.
7. If this is the first time you have connected to the CSF you **must** change your password by entering **`yppasswd`** and following the instructions.
   IMPORTANT: Make sure you remember your new password; you will need it for today and for any other training courses that uses the CSF. If you forget your password:
   - If on a training event please tell your tutor
   - At other times please email the CSF helpdesk: its-research@manchester.ac.uk

4. Modules

Wiki page: wiki.rcs.manchester.ac.uk/community/csf/Software/Modules

Modules are used on the CSF to set your environment and to access resources and programs (enter **`man module`** for more info). Useful module commands are:

`module avail`	see all available modules
`module list`	see your currently loaded modules
`module load` ***ModuleName***	load a module called ***ModuleName***
`module unload` ***ModuleName***	unload a module called ***ModuleName***

5. To download files required for training

To install files required for training enter
`module load training/training_material`

Running this command will install files for all our training courses in a directory (called training) in your home folder. You can run this command again to replace any missing files without overwriting files you have edited.

6. Compiling

Wiki page: wiki.rcs.manchester.ac.uk/community/csf/Software/Applications/Compilersintel

Note: our training environment uses two 12-core Intel Xeon E5-2640 (2.5GHz) *Sandy Bridge* compute nodes, each with 64GB ram. We use the Intel compilers to compile Fortran and C/C+ programs. MPI training uses the module for the standard (i.e. non-InfiniBand) network on the CSF. Alternative options can be found on the CSF wiki page.

The steps to compiling are
1. Load the necessary module
2. Run the compiler

The following table lists the modules and compile commands used in our training.

Language	**Parallelisation**	***ModuleName***	***CompileCommand***
Fortran	None	**`compilers/intel/fortran/12.0.5`**	**`ifort`**
	OpenMP	**`compilers/intel/fortran/12.0.5`**	**`ifort -openmp`**
	MPI	**`mpi/intel-12.0/openmpi/1.6`**	**`mpif90`**
C/C++	None	**`compilers/intel/c/12.0.5`**	**`icc`**
	OpenMP	**`compilers/intel/c/12.0.5`**	**`icc -openmp`**
	MPI	**`mpi/intel-12.0/openmpi/1.6`**	**`mpicc`** or **`mpi++`**

Loading the module

Enter the following, where ***ModuleName*** is replaced by text from the above table:
`module load` ***ModuleName***

Running the compiler

In general to compile enter:

CompileCommand SourceFile* `-o` *Executable

where:

CompileCommand	is replaced by text from the above table
SourceFile	is replaced by the name of your source code with the correct file extension, e.g. `.f90` for Fortran `.c` for C
Executable	is replaced by the name of the executable you want to create, and **must** follow `-o`

Optimisation can be controlled using the `-O`***n*** flag, where `O` is uppercase letter O (not zero) and ***n*** is a number representing optimisation level between 0 (no optimisation) and 3 (aggressive optimisation). For example to compile without optimisation (as required in MPI and OpenMP courses) add the `-O0` flag, i.e. enter:

CompileCommand SourceFile* `-O0 -o` *Executable

Some training courses use the **`make`** utility as an alternative method for compilation. If this is the case you will be instructed by your tutor.

7. To submit jobs to the CSF compute nodes

Wiki page: wiki.rcs.manchester.ac.uk/community/csf/UserInfo/sge

When you log in to the CSF you are connected to the *login node*. This is a lightweight node used for tasks such as compiling code and managing files. You should **not** run your executables on this node though - it doesn't have many cores or much memory. Instead, you are going to make use of the much more powerful *compute nodes* in the CSF. You do this by making a request to the SGE batch system to run your executables. This will run your code on suitable compute nodes, according to the number of cores and other resources you request. If many other jobs are running you may have to wait until suitable compute nodes become available, but the queue system manages this fairly. The following sections describe how to submit your jobs to the SGE batch system (from the login node).

7.1. Batch jobs

From the login node, jobs are submitted using the **`qsub`** command to send jobs to the SGE batch system. This will select appropriate compute nodes to run on. **`qsub`** uses various flags to set job parameters (enter **`man qsub`** for more info). The flags can be given in a *job script* file, which is read by **`qsub`** or can be given as command line flags. It is also possible to specify some options as command line flags and some in a job script.

The most common flags you will use are:

-cwd to run the job in the current directory
-l (lowercase letter l) to use specific resources, e.g. **-l course** to use compute nodes reserved for training
-N to set job name
-o (lowercase letter o) to set file name for the standard output stream
-V (uppercase letter V) to export the current environment to the compute node
-pe to set the parallel environment and number of cores (not required for single core jobs) e.g.
-pe smp.pe *n* for shared memory jobs using ***n*** cores (suitable for OpenMP code)
-pe orte.pe *n* for distributed memory jobs using ***n*** cores (suitable for MPI code)

By default standard output and error streams appear in files named **jobname.ojobID** and **jobname.ejobID** respectively, where **jobID** is a unique number given the job by SGE when you submit your job to the batch system. By default **jobname** is the name of the script file or command. Alternative job names can be specified using the **-N** flag, and alternative output and error file names can be specified using **-o** and **-e** flags (see below for flag descriptions).

After submission **qstat** can be used to monitor progress, and **qdel *jobID*** used to delete jobs.

Jobs can be submitted from temporary course accounts outside of training events, after making sure the **-l course** flag is removed. Such jobs will receive lower priority in the CSF queues.

Method 1 – using a job script file

This method uses a text file in which lines starting with **#$** (followed by a space) set the flags. After listing SGE flags all commands to be run are listed. For example, the following script submits a serial (i.e. single core) job to run an executable called ***MyExecutable*** in the current directory. If this script is saved to a file called ***QsubScript*** the job can be submitted by entering **qsub *QsubScript*** at the prompt.

```
#$ -V
#$ -cwd
#$ -o OutputFileName
#$ -l course

./MyExecutable
```

Note: the line **#$ -l course** ensures jobs run on compute nodes reserved for training events. Remove this flag when submitting jobs outside training events.

To run shared memory jobs add

```
#$ -pe smp.pe n
```

where ***n*** is replaced by the number of cores up to a maximum of 12.

For for distributed memory (MPI) jobs use

```
#$ -pe orte.pe n
```

where ***n*** is replaced by the number of cores up to a maximum of 24 for training events.

Note: that when running jobs outside of the training environment (if you have full CFS access) you are encouraged to use **`orte-12-ib.pe`** ***n*** where n is a multiple of 12. This will give better performance than **`orte.pe`** due to use of faster InfiniBand networking. In this case you would also have to load the **`mpi/intel-12.0/openmpi/1.6-ib`** module.

Method 2 – enter flags at the command line

Simply type **`qsub`** followed by the list of flags and values, ending with the command to be executed. If the command to be executed is a binary executable (i.e., not a job-script as described above) then you should add **`-b y`** to indicate a binary executable.

For example, the script shown above is equivalent to

`qsub -b y -V -cwd -l course -o` ***OutputFileName ./MyExecutable***

7.2. Interactive jobs

Wiki page: http://wiki.rcs.manchester.ac.uk/community/csf/UserInfo/sge/qrsh

Interactive jobs are submitted using **`qrsh`**, for example to request a single core interactive job:

`qrsh -V -cwd -l course` ***CommandToRun***

or to request a distributed memory parallel job that uses mpirun enter:

`qrsh -V -cwd -l course -pe orte.pe` ***n*** **`mpirun -np`** ***n CommandToRun***

replacing ***CommandToRun*** with whatever it is you want to run, and for parallel jobs replacing both ***n*** with the number of processes/cores. Outside training events **`course`** should be replaced by **`inter`** when using qrsh.

8. Post training help

Wiki page: wiki.rcs.manchester.ac.uk/community/csf/UserInfo

If unable to find the information you need on the CSF wiki page, please email the helpdesk: its-research@manchester.ac.uk