

Computer Science (CS) High-Performance Computing (HPC) Cluster

The CS HPC cluster has over 1000 nodes. The cluster is designed to run large scale computing jobs in batch (non-interactive) mode. The cluster offers very limited graphics-based interactive computing services. To efficiently use the cluster, users should set up their code so that it can be run in batch mode. Users who primarily need graphics-based interactive computing should use the smaller Economics Department cluster, ISD services, or desktop computers.

This document describes the hardware and software resources available on the cluster and provides instructions on how to access and use the cluster. It is assumed throughout that users already:

1. Know basic linux shell commands
2. Know how to write a shell script to submit batch jobs the HPC clusters
3. Know how to call the desired software in batch mode

Specifically, these instructions cover:

1. Getting a CS cluster account
2. Connecting to the CS cluster
3. Accessing compute nodes
4. Accessing software
5. Storage on the CS cluster
6. Policies and best practices

If you need help setting up or run into problems, please contact CS support or ECON IT support: 1. CS support: cluster-support@cs.ucl.ac.uk 2. ECON IT support: economics.it@ucl.ac.uk.

Note: Throughout this document there are links to help pages hosted by the CS department and the ECON Wiki. These pages are password protected. To access the CS deparment HPC page you must obtain the username and password either from cluster-support@cs.ucl.ac.uk or from economics.it@ucl.ac.uk.

1. Getting a CS cluster account

1.1. Register with Computer Science (CS) Dept

To use the cluster you will need two accounts:

- + a CS departmental account (to allow remote access)
- + an CS cluster account (to allow access to the cluster).

To obtain both accounts:

1. Fill in a Registration Form: Collect a form from Room 4.20 in the CS dept or directly from Fatima in Economics IT. The key fields to fill on the form are:
 - UCL username
 - Phone Contact for user
 - Any supervisor permissions
 - A signature agreeing to CS terms and conditions of usage.
2. Hand in the form to the CS Helpdesk in room 4.20 in the Engineering building and they will setup your accounts. The CS team will notify you by email when they have done so. This may take between a day and a week depending on their workload.

By default both accounts will have the same username and password.

You can reset one or both passwords either over the phone or in person at the CS helpdesk office.

2. Connecting to the CS cluster

There are three ways to connect to the cluster:

- 1) Connect using ThinLinc (graphical interface)
- 2) Connect using ssh (command line interface)
- 3) Connect using ftp (data transfer)

2.1. Connect using ThinLinc

Connecting using ThinLinc is a two-step process:

1. Connect to a **Computer Science Remote Worker (CSRW)** using your **CS Department Account** username and password.
2. Once connected, connect to the CS Cluster using your **CS Cluster Account** username and password.

2.1.1. Use ThinLinc to connect to a CSRW

This step uses a program called ThinLinc to create a 'remote desktop' session connecting your computer to a Computer Science Remote Worker (CSRW). To do this:

1. Download and install Thinlinc.
2. Login to a CSRW using your **CS Department Account** username and password.

Instructions to download ThinLinc and connect to a CSRW can be found [here](#).

To log in, open the ThinLinc application and enter your username and password from your **CS department account**. At this stage you are logged on to a server in the CS department. The remote desktop environment includes some applications including a web browser, a terminal window, and many others.

2.1.2 Connect to the CS cluster from a CSRW

From the CSRW, open a terminal (i.e. from the Applications drop down menu). Then use ssh to logon to one of the CS Cluster ‘Login Nodes’. The main login nodes are named **vic** and **wise**.

To login to **vic**, at the command line prompt type:

```
ssh -X uctpXXX@vic.cs.ucl.ac.uk
```

Or, to login to **wise**, at the command line prompt type:

```
ssh -X uctpXXX@wise.cs.ucl.ac.uk
```

The ‘-X’ option is required to enable ‘X-forwarding’ of graphics.

There are several different log on nodes, but for your purposes one of **vic** or **wise** should suffice (i.e. if you can’t log on to one for some reason, then try the other).

2.2 Use ssh to connect to CS cluster

For remote access (i.e. from a computer not connected to the UCL network), connecting to the CS cluster is a two step process: 1. Connect to CS department network using your **CS Department Account** username and password. 2. From the CS server, connect to the cluster using your **CS Cluster Account** username and password.

From a computer connected to the UCL network (e.g. a desktop in Drayon House or the Econ HPC), you can skip step 1. and connect directly to the CS cluster directly using ssh.

2.2.1 Use ssh to connect to CS Department network

You can connect to several CS department servers including **tails**, **storm**, or **jet**.

To connect to **tails**, open a command terminal and enter the command:

```
ssh -X uctpXXX@tails.cs.ucl.ac.uk
```

Here,

- The option `-X` allows graphics to be forwarded from the CS cluster to your computer.
- The username “uctpXXX” is that given when you are assigned your CS department account.
- The text after the `@` is the address of the server.

After you type the command, the server will ask for a password. The password should be your CS department account password.

To access `storm` or `jet`, replace `tails` with one of those options.

2.2.2 Connect to the CS cluster from within CS department network

At this point you have logged into a server within the CS department’s network. The next step is to log in to a ‘login’ node on the CS cluster.

You can do this using your username and password for your **CS cluster account**.

At the command line prompt type:

```
ssh -X uctpXXX@vic.cs.ucl.ac.uk
```

Here, * The option `-X` allows graphics to be forwarded from the CS cluster to your computer. * The username “uctpXXX” is your **CS Cluster** username. * The text after the `@` is the address of the login server.

After entering the command, the system will request your password. Enter your **CS Cluster** password.

Alternatively, replace `vic` with `wise`.

There are several different log on nodes, but for your purposes one of `vic` or `wise` should suffice (i.e. if you can’t log on to one for some reason, then try the other).

2.3 Connect to the CS cluster using sftp

To transfer files to the cluster use sftp. You can use any ftp client or you can use the command line. For command line sftp, connecting using sftp is the same as using ssh. Simply replace `ssh` with `sftp`.

For example, to connect to `tails` using sftp:

```
sftp uctpXXX@tails.cs.ucl.ac.uk
```

Then, when prompted enter your password. For sftp instructions, enter `man sftp` at the command prompt or search the web.

3. Accessing compute nodes

The login nodes are used by many users and should never be used for large scale computational work. Instead, after logging in to a login node, you must run your ‘job’ on the the ‘compute nodes’ of the cluster. The cluster uses a ‘job scheduler’ called Sun Grid Engine (SGE). Typical workflow is as follows: 1. Logon to cluster. 2. Transfer data and/or files to cluster (using ftp or email). 3. Edit files or code. 4. Write a script to submit your job to SGE. 5. Submit your job. 6. Monitor job progress if necessary. 7. Download results to your local computer (using ftp or email).

SGE allocates jobs to compute nodes and attempts to minimise congestion across users. Depending on the load on the cluster, the jobs you submit will either be instantly scheduled to a compute node or placed in a queue until the resources requested become available.

SGE does not currently allow for graphical interfaces. For graphics-based-interactive sessions, see Section XXX below.

There are two types of SGE sessions:

- Interactive (command line interactive)
- Non-interactive (batch mode)

Both are discussed in more detail below.

3.1. Interactive sessions (command line only)

To open an interactive session, you need to specify requests for **running time** and **memory**.

To open a session, at the command line type:

```
qrsh -l h_vmem=1.9G,tmem=1.9G,h_rt=8:0:0
```

This will log you into an available node for 8hrs and allow you to use up to 1.9G of memory. In more detail:

- **q**rsh is the login command for an interactive session
- **-l** is a flag for resource requests for the interactive session
- The resource options listed after the **-l** flag are:
 - **h_vmem=XG,tmem=XG** requests X Gb of memory
 - **h_rt= H:M:S** requests that the session run for H hours, M minutes, S seconds

*Note 1: For further command line options for **q**rsh type: **man q**rsh*

User Tips

- The SGE scheduler runs in 5 minutes cycles, so it may take a short while to be allocated a node.
- The smaller the memory requested, the more likely your job will be allocated quickly. There are a limited number of nodes that have access to large amounts of memory.
- For a small job, 2G is likely to be sufficient. For Matlab, request at least 4G.
- If you have need to request a lot of memory (i.e. $X > 2G$), exclude the `h_rt` resource request from your `qssh` command. For example, to request a 14G session type:

```
qssh -l h_vmem=14G,tmem=14G
```

The `qssh` command will log you onto one of the ‘compute nodes’ on the cluster. After logging on, you will need to load and open the software you require. See Section 4 for details.

3.2. Non-interactive sessions

To run a batch job:

1. Write a script detailing what job to run and requesting resources from SGE (e.g. create a text file named `job1.sh`).
2. Submit the script using the command `qsub`:

```
qsub job1.sh
```

Instructions for writing a script can be found at:

shell script for the Sun Grid Engine

The remainder of this section discusses some specific command options you can include in the job script.

Requesting memory and running time

To request memory and running time, add lines like the following to your job script:

```
## -l h_rt=1:10:35 # This line specifies run time of 1 hour, 10 mins and 35 seconds
## -l tmem=1.9G,h_vmem=1.9G # This specifies 1.9 Gigabytes (can also specify M for Megabytes)
```

Without these options, the job will run with default options:

```
## -l h_rt=0:0:30 # 30 mins is default hard run time
## -l tmem=256M,h_vmem=256M # Default is 256MB
```

Requesting a parallel environment

If you are running a parallel job that uses more than one nodes on the cluster, you need to specify the parallel environment:

- **smp**: single node with multiple workers
- **matlabpe2014b**: parallel environment specific to Matlab. CS only support b release each year. Matlab 2015b is yet to be supported.
- **mpi**: Old MPI interface
- **mpich**: New MPI interface
- **orte**: Distributed computing across nodes, but tries to cluster processes on nodes
- **para**: Distributed computing across nodes, no clustering of processes

If no parallel environment is specified, the default is **orte**.

To specify a the parallel environment used by the CS cluster job add the following lines to your shell script. Note the second line is only necessary when using more than one node i.e. default is Default is `#$ -R n`.

```
#$ pe [pe_option] [NumWorkers] # <- SGE option for parallel environment
#$ -R y                        # <- Resource reservation. Useful when lots
```

Below are some example of lines to add to submission scripts.

Example 1: Use a single node with 4 workers in parallel add the following line to you script:

```
#$ pe smpd 4
```

Example 2: To use a single node with 16 workers in parallel add the following line to your script:

```
#$ pe smpd 16
```

Note this may be slow as the programme will wait for node with enough cores to support 16 workers to be free. But it is possible.

Example 3: To use a multiple nodes clustered on few nodes with 20 workers in parallel add the following lines to your script:

```
#$ pe orte 20
#$ -R y
```

Example 4: To use a multiple CPUs arbitrarily distributed over nodes with 30 workers add the following lines to your script:

```
#$ pe para 30
#$ -R y
```

3.3. Checking the status of your jobs

This section shows you how to check the current status of your jobs and the SGE queues. See the Econ Wiki.

Note that `qstat-rn` command is specific to the Economics HPC and is only available on the CS HPC if you have added the `econutils` module to your session. Note that you can do this by adding:

```
module load econutils
```

to your `.bashrc` profile or the command line.

3.4. Deleting jobs

To delete job with job number 123456 type:

```
qdel 123456
```

4. Accessing Software

4.1. Software stack

Once logged on to a compute node, to access software you can use the modules environment.

To see the list of software currently available to load type:

```
module avail
```

The following software is currently available via a module. To see the list of software available using modules:

```
[uctXXX@vic ~]$ module avail
```

```
----- /share/apps/econ/Modules/3.2.10/modulefiles -----
dot                matlab/r2012b          openmpi/gcc/1.10.0
econutils          matlab/r2013b          openmpi/gcc/1.8.1
gcc/5.2.0          matlab/r2014a          openmpi/intel/1.10.0
gcc/6.2.1          matlab/r2014b          openmpi/intel/1.8.1
git/2.8.3          matlab/r2015b          R/3.4.2
intel/composer/2013.1.117 module-info          stata/14
intel/composer/2015.1.133 modules          totalview/8.15.7-6
julia/0.4.5        nag/fl16i25dc          use.own
knitro/10.0.1-z    nag/mb16a23dml
knitro/10.1.2-z    nag/mb16a24dnl
```


4.2. Loading Software

On the CS HPC the user must load software listing dependencies. Below are some details on loading some of the most commonly used programs.

MATLAB

```
module load gcc/5.2.0 nag/mb16a24dnl matlab/r2015b
module load gcc/5.2.0 nag/mb16a24dnl matlab/r2014b
module load gcc/5.2.0 nag/mb16a24dnl matlab/r2014a
module load gcc/5.2.0 nag/mb16a24dnl matlab/r2013b
module load nag/mb16a23dnl matlab/r2012b
```

Stata

```
module load stata
```

Julia

```
module load git/2.8.3 julia
```

or if you have your own version of Julia you will need to only load `git/2.8.3`. That is, type:

```
module load git/2.8.3
```

Note: Julia users should speak to cluster support about getting setup on the CS cluster. Amongst other things, there are file limit, version control, library paths, and non-trivial batch-mode issues to address before job processing can begin.

R

```
module load gcc/5.2.0 R/3.4.2
```

Fortran

For an example demonstrating how to run Fortran see this example

KNITRO

```
module load knitro/10.1.2-z
```

4.3. Running software

Once you have loaded your package you can call them in either batch or interactive mode. For example, once software is loaded,

- to open an interactive stata from terminal from inside CSRW type:
`xstata-mp`
- to open an interactive MATLAB type: `matlab`
- to open an interactive R type: `R`

Note: You might want to add the ‘module load ... commands to your .bashrc file for commonly used software so they load automatically when you log on.

5. Storage

The storage of files is quite different to the setup on the Economics HPC.

5.1. Saving files and making directories

While it is possible to make directories and save files in your home directory on the CS HPC it is important to note that these files **are not backed up**.

The CS department offer backed up storage areas called **project stores**. Unlike your home directories, project stores are designed to handle intensive reading and writing of files during cluster jobs.

Project stores are allocated to individual users and/or multiple user groups on request. To **request a project store** fill in the online storage request form.

5.2. Transferring Files to and from the CS HPC

Any SFTP service can be used to transfer files to and from the CS HPC. Popular SFTP include WinSCP or FileZilla

To move files via an FTP client use `tails.cs.ucl.ac.uk` and log on using your **CS department account** username and password.

The address of Economics storage space from `tails` is:

`/slash/economics/research/[your storage directory name]`

6. Policies and Best Practices

6.1. Cluster Etiquette

CS department cluster etiquette

6.2. Memory Management

Please think about your job's memory usage.

In order for us to make best use of the cluster, we now require you to submit your expected memory usage when you submit a job (if you leave this out, your job will not run).

If your job goes over this requested limit, it will be automatically killed. The more memory you request, the longer your job will wait in the queue. Only request what you think you need.

To request x GB of memory:

```
-l h_vmem=xG , tmem=xG
```

to your **qsub** command. Or, alternatively, add the following line to your **qsub** script:

```
#$ -l h_vmem=xG,tmem=xG
```

Most nodes have between 2 and 8GB per core. The physical machines vary from 4GB to 1TB, but if you want to run a job requiring more than 47.2GB please contact request@cs.ucl.ac.uk.

For the best job throughput please request less than 1.9G. That is:

```
-l tmem=1.9G,h_vmem=1.9G
```

There are only a few machines with more than 64GB of memory. If you request more than 8G you will probably need to add a resource reservation to your job, unless the cluster is very empty.

To add resource reservation to your **qsub** command add:

```
-R y
```

or in your **qsub** script, add:

```
#$ -R y
```

- Although machines have 8, 16, 24, 48, 96, 128, 256 and 1TB of memory, in reality this translates into requestable memory of 7.9, 15.7, 23.5, 62.9, 47.2, 94.4GB. i.e. if you request 16G you will not be able to run on any of the 16G machines and your job will queue for longer.

- Also note in a parallel environment, the amount you request is the `h_vmem` and `tmem` multiplied by the number of cpu's you request.

See link for more details.