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

Through the CS HPC the Economics Deparmtent has access to over 1000 nodes.

Below is description explaining of how members of the Economics faculty can access nodes on the Computer Science's HPC.

If you need help setting up or run into problems please contact CS support: cluster-support@cs.ucl.ac.uk.

Throughout this document there are links to help pages hosted by the CS department. These pages are password protected. To view them you must obtain the username and password either from cluster-support@cs.ucl.ac.uk or from j.mcglynn@ucl.ac.uk.

1. Accessing the CS HPC

Like the Economics department, the CS department has two types of account - a departmental account and a HPC account.

If you only require command line access to the CS HPC, you only need a CS HPC account to gain access via ssh. If, however, you wish to access the CS HPC through a graphical user interface (i.e. remote desktop) you will need both a CS departmental account and a CS HPC account.

This is because the CS HPC graphical interface, the Computer Science Remote Worker (CSRW), requires a CS departmental account to use it (see section 2.1 below for more details).

Note that the CS HPC can only be accessed from the UCL campus. If you would like access from outside UCL you will need contact the CS support team.

1.1 Getting registered with Computer Science (CS) Dept

To get a CS account:

1. Fill in a Registration Form: Collect a from Room 4.20 in CS. 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 for to CS Helpdesk in room 4.20 in Engineering building and they will setup your account. 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.

Once you have CS account, they will set up a CS HPC account. You will then need to set the password either over the phone or in person at CS helpdesk office.

To only apply for a CS HPC account fill in this online form.

Note that if you require both a CS account and CS HPC account you should first get a CS account.

2. Connecting to the CS HPC

To use the CS HPC you first have to log in to a "header" or "log on" node. There are two "header" nodes for Economics department users on the CS HPC:

- wise.cs.ucl.ac.uk
- vic.cs.ucl.ac.uk

There are two ways to access these header nodes:

- Log in and open a remote desktop session on the HPC using CS department's GUI called CSRW
- SSH in directly from your terminal

2.1. Graphical User Interface (GUI) - CSRW, Thinlinc

To use graphical tools on the CS HPC you can use the Computer Science Remote Worker (CSRW).

To use the CSRW you will need a CS account and to have downloaded Thinlinc. Download and use instructions for the CS department's can be found here.

Note: At the time of writing, NX client - the graphical user interface used by the Economics department to access its HPC - is not supported. However, there are plans to support it in the near future.

2.2. Accessing the CS HPC through a terminal

To gain access to the header node called `vic` from the command line using ssh type:

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

where

- The option `-X` allows graphics to be forwarded form the HPC to your computer (as long as you are on site at UCL).
- The username "uctpXXX" is the name of the user. The username should be your UCL username

- 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 you CS HPC Account Password.

Further platform specific SSH logon details can be found below:

- Windows
- Mac OSX
- Linux

3. General Information about the CS HPC

3.1 Storage

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

3.1.1. 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.

3.1.2. Accessing existing files on Economics Dept Server

On the Economics HPC it was possible to directly access files saved on the Economics department server. At the time of writing, this is not possible on the CS HPC. Therefore, if you want to access these files on the CS HPC you will have to transfer them using a SFTP clients (see section 4).

Ideally, you should transfer these files to a project store.

3.2. Software licenses

See Economics Wiki

4. 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

5. Accessing compute nodes: Sun Grid Engine

To ensure your jobs run as quickly as possible the cluster uses the Sun Grid Engine© (SGE) solution to keep track of what resources are available. Depending on the load, the jobs you submit will either be instantly scheduled to a compute node or placed in a queue until the resources requested become available.

5.1. Interactive sessions

This section covers how to submit an interactive sessions along with examples of how to decide when it may be best to use this type of session.

The CS department request that users specify options to limit memory and time logged in when logging onto the CS HPC using an interactive session. As such, to open an interactive session from your terminal once you have logged into the CS HPC, 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 1.9G of memory. In more detail:

- `qrsh` is an alternative interactive session login command
- `-l` is a flag for resource requests of the interactive session
- The resource options listed afted the `-l` flag:
  - `h_vmem=XG, tmem=XG` requests X Gb of memory
  - `h_rt= H:M:S` request that the session run for `H` hours, `M` minutes, `S` seconds

Note 1: From the user's persepective the ***`qrsh`*** *command is an alternative to* ***`qlogin`*** *command currently used on the Economics HPC.*

Note 2: Unlike the Economics HPC there is no distinction between a batch queue and an interactive session queue.

5.2. Non-interactive sessions

This section covers how to submit an interactive sessions along with examples of how to decide when it may be best to use this type of session.

A detailed discussion of how to submit batch jobs with many useful the SGE options can be found on the Economics Wiki.

In addition, like an interactive session, you will have to add lines specifying hard run time and memory requirements. For example, if you submit a a qsub using a shell script you must add:

```
#$ -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 Megabyt
```

The job will run without it if omitted, but with restrictive defaults applied. The defaults are:

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

Note: Users who specify a parallel environment in their submission script please do see section 5.5 for details of parallel environments on the CS HPC.

5.3. 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.

5.4. Deleting jobs

To delete job with job number 123456 type:

```
qdel 123456
```

5.5. SGE Parallel Environments

If you wish to control the parallel environment used by your cluster job there are several parallel environments on the CS HPC:

- `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 that clusters are grouped on nodes
- `para`: Distributed computing across nodes across nodes.

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

Those using the `julia` parallel environment on the Economics HPC should specify `orte` or `para` depending on how you want to instantiate workers.

Specifying a parallel environment

To specify a the parallel environment used by your cluster job add the following lines to your 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]::ASCIIString NumWorkers::Int  # <- 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 distrbuted over nodes with 30 workers add the following lines to your script:

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

6. Accessing Software

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 and is stored in `./share/apps/econ`:

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

----------------- /share/apps/econ/Modules/3.2.10/modulefiles -----------------
dot                         matlab/r2012b             nag/mbl6a24dnl
econutils                   matlab/r2013b             openmpi/gcc/1.10.0
gcc/5.2.0                   matlab/r2014a             openmpi/gcc/1.8.1
git/2.8.3                   matlab/r2014b             openmpi/intel/1.10.0
intel/composer/2013.1.117   matlab/r2015b             openmpi/intel/1.8.1
intel/composer/2015.1.133   module-info               stata/14
julia/0.4.5                 modules                   use.own
knitro/10.0.1-z             nag/mbl6a23dml
```

Unfortunately, we are still working on developing symmetric functionality for the CS HPC as the Economics HPC. In particular, there are some important differences for users migrating from the Economics HPC:

a. Loading modules requires more verbose commands

For example, typing `matlab` on the Economics HPC in an interactive session calls a script which automatically loads the MATLAB together with any dependencies. However, on the CS HPC the user must load software listing dependencies. For example, to load MATLAB r2015b the user must type:

```
module load gcc/5.2.0 nag/mbl6a24dnl matlab/r2015b
```

b. No Graphical User Interface (GUI) or graphics

At the time of writing it is not possible to access the Graphical User Interfaces of the software above in an interactive cluster session. Unfortunately, this is the case even if you specify `-X` flag when logging in.

This includes the GUI of MATLAB, Multi-Processor Stata, and any graphical packages in Julia.

6.2. Loading Software

Below are some details on loading some of the most commonly used programs.

6.2.1. MATLAB

To load (command line only) MATLAB run a command from the list below that corresponds to the version you wish to load.

```
module load gcc/5.2.0 nag/mbl6a24dnl matlab/r2015b
module load gcc/5.2.0 nag/mbl6a24dnl matlab/r2014b
module load gcc/5.2.0 nag/mbl6a24dnl matlab/r2014a
module load gcc/5.2.0 nag/mbl6a24dnl matlab/r2013b
module load nag/mbl6a23dml matlab/r2012b
```

6.2.2. Stata

To load (command line only) Stata-MP

```
module load stata
stata-mp
```

6.2.3. Julia

To open Julia type

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

Note: you will need git to manage packages

In order to use Julia's packages you will need to change the default location of you package directory (or repository) to a project store (see section 3). This is because your home drive, the default location for Julia to store its packages, has file number limit that Julia may hit when it install packages.

The easiest way to manually control the location of your package directory is to to add the following lines to your `.bash_profile`.

```
JULIA_PKGDIR=[PATH_TO_PKGDIR]
export JULIA_PKGDIR
```

Then reload your `.bash_profile`

```
source ~/.bash_profile
```

Then, open Julia and check that your package directory has changed and initialize it. That is, in Julia type:

```
Pkg.dir() # Returns the path of you package directory
Pkg.init() # Initializes the package directory
```

Note: You only need initialize the package directory it once.

6.2.4. Fortran

For an example demonstrating how to run Fortran this example

7. Policies and Best Practices

7.1 Cluster Etiquette

CS department cluster etiquette

7.2 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.