README.md

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 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 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 and interacting with the CS cluster's resources
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. Getting registered with Computer Science (CS) Dept

Like the Economics department, the CS department has two types of account: + a department account; and, + a HPC account.

To get a **CS department account**:

1. Fill in a Registration Form: Collect a form from Room 4.20 in 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 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, the CS department will automatically set up a **CS cluster account** for you. It will have the same user name and password by default.

You can reset the password either over the phone or in person at CS helpdesk office.

2. Connecting to the CS cluster

Connecting to the CS cluster is a two step process.

1. Log in to the CS department computing system using your **CS department account**
2. Once inside, connect to the CS cluster using your **CS cluster account**.

Sections 2.1 and 2.2 below discuss these steps in turn.

2.1. Log in to the CS department computing system

There are two different ways to log into the Computer Science system:

1. Log in using the **Computer Science Remote Worker (CSRW)**, ThinLinc
2. `ssh` in from your terminal

Each is discussed in more detail below.

2.1.1. Remote Desktop Session - CSRW, ThinLinc

To use start a 'remote desktop' session, use the Computer Science Remote Worker (CSRW). The CSRW uses a program called ThinLinc.

- Download and use instructions for the CS department's CSRW can be found here.

Once you have downloaded ThinLinc for your operating system, you will need your CS department account to log in.

To log in, open the ThinLinc application and enter you username and password from your **CS department account**. The next step is to access the CS cluster are described in section 2.2.

2.1.2. Accessing the CS cluster through a terminal

The second option is to `ssh` into the CS computing system at the command line by logging into `tails`. To do this, at the command line type:

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

where

- 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.

2.2. Accessing the CS cluster from within CS department computing system

At this point you have logged into the CS department's computing system using either CSRW as described in 2.1.1 or via `ssh` as described in 2.1.2. The next step is to log onto the CS cluster.

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

If you accessed the CS computer network through ThinLinc, open a terminal (i.e. from the Applications drop down menu). If you have `ssh`'d directly into `tails` continue at the command prompt.

To log into the CS cluster via the `vic` log on node, at the command line prompt type:

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

Alternatively, to log into the CS cluster via the `wise` log on node, at the command line prompt type:

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

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).

3. Accessing compute nodes

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.

There are two types of sessions:

- Interactive
- Non-interactive

Both are discussed in more detail below.

3.1. 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 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.

User Tips

- The SGE scheduler runs in 5 minutes cycles, so it may take a short while to be allocated a node while.
- The smaller the memory the more nodes will be available for your work. So it is your interest not to request more memory than you absolutely need
- If you have want to request a lot of memory (i.e. X > 2G), exclude the `h_rt` resource request from your `qrsh` command. For example, to request a 14G session type:

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

Once logged on, you need to load and open your software. See Section 4 for details.

3.2. Non-interactive sessions

This section covers how to submit an non-interactive session to CS cluster.

As noted in the introductions, these instructions assume users know how to:

1. Write a shell script for the Sun Grid Engine
2. Run parallel jobs using your desired software from the command line.

The remainder of this section discusses the specific commands required to run on the CS cluster.

Requesting memory for parallel jobs

In addition to the example shell scripts, you should add lines specifying hard run time and memory requirements to your shell script. For example, when you submit a job, in the shell script you must add the following SGE flags:

```
#$ -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 of k for kilobytes)
```

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
```

Advanced Settings: Parallel Environment

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

- `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 of memory and/or mult
```

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
```

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:

```
[uctXXXX@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/fll6i25dc             use.own
```

```
knitro/10.0.1-z           nag/mbl6a23dml
knitro/10.1.2-z           nag/mbl6a24dnl
```

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/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
```

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 you 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.