MSc in Statistics
Department of Mathematics
Imperial College London

IMPERIAL

MSc in Statistics -- Short guide for students to using the NextGen Maths Cluster

Last updated: May 2024

1. Overview

The Department of Mathematics at Imperial College has computing resources for running simulations with CPUs or GPUS.

One of these resources is called the **Maths NextGen Compute Cluster** and there are excellent, detailed instructions by Andy Thomas here: http://sysnews.ma.ic.ac.uk/compute-cluster/.

However, there is so much information in these instructions, it may be overwhelming for someone trying to get started quickly. This document is intended to get users up and running quickly, and then those looking for further information are encouraged to look on the website above. The information in these notes is essentially all from the above site, it is just presented a bit differently here.

Although several languages are supported, **these notes will present an example using the R language**. These notes will take someone with no experience of using the cluster through a series of steps to the point where they can submit a simple R programme to the cluster to be executed.

Alternatively, student can approach their supervisor to access Imperials High Performance Computing cluster, https://www.imperial.ac.uk/admin-services/ict/self-service/research-support/rcs/service-offering/hpc/

2. Advantages of using the cluster

The cluster is almost certainly more powerful than your personal computer, and any program you want to execute (called a **job**) can be submitted to a **queue** of programmes that will be executed by the cluster in turn; you can instruct the cluster to send you an email to let you know when your job is completed.

Note that there are computing servers that are solely reserved for MSc in Statistics students: https://sysnews.ma.ic.ac.uk/stats/MSc_compute_servers.html

3. Technical guidance for MSc in Statistics students

Step 1: Get a terminal with SSH

If you are using Linux or Mac OS, then these operating systems already come with a suitable terminal.

If you are using Windows, then download PuTTY from https://www.chiark.greenend.org.uk/~sgtatham/putty/. This is a terminal emulator with an implementation of SSH (Secure Shell Protocol) built-in.

Note that the Windows 10 Powershell supports SSH, but it needs to be added/installed, since it is not installed by default.

Step 2: Connect to the Imperial College network with a VPN

If you are on campus, you should not need to do this, but if you are off campus, you will. There are instructions on the College website for doing this, and the instructions depend on the operating system you are using.

Step 3: Create a public key to use with SSH

You need to create an SSH public/private key pair. In a terminal, first move to your home directory using:

```
cd ~/
```

And then type:

```
ssh-keygen
```

When prompted to enter a passphrase, **just press enter**, i.e. do not enter a passphrase. You will be asked to enter the passphrase a second time (just press enter again).

Now you need to copy your public key to a file called `authorized_keys`. This is done using the following two commands:

```
cd .ssh
```

```
cp id_rsa.pub authorized_keys
```

The first command changes your folder to the hidden `.ssh` folder, and the second command does the copying.

Step 4: Connect to the cluster

The cluster consists of several individual computers. One of those is called **macomp001**. In a terminal, use the `ssh` command to connect to this computer; if your College ID is **abc123** then you would type:

```
ssh abc123@macomp001.ma.ic.ac.uk
```

and press the **enter** key; a prompt will appear asking you to enter your College password (as if you were signing into your email online).

Note that when you type in your password, it will not show on screen (it will seem as if you are not typing anything); this is to protect your password. Press the **enter** key after typing your password.

Step 5: Create a simple program

It is worth learning a few simple terminal commands first.

Create a simple folder using the `mkdir` command:

```
mkdir myfolder
```

and navigate to that folder:

```
cd myfolder
```

Now, open a terminal-based editor such as `Vim` or `nano`, and create a simple script named `myscript.R` that creates some output; for example:

```
makeAndSaveData <- function(n=10, seednum=1, filename="data.csv"){

      set.seed(seednum)

      x <- rnorm(n)

      write.table(x, file=filename, sep=",", row.names=F, col.names=F)

}

makeAndSaveData()
```

Note that if this script were sourced in an R terminal, it would run, because `makeAndSaveData()` is called at the end of the script, and all it does is generate some random numbers and save them in the file data `.csv.`

Try running `myscript.R` by calling the following command in a terminal:

```
Rscript myscript.R
```

If you now use the ls command in the terminal,

```
ls
```

This will show all the files in the folder `myfolder`, which should be

- `myscript.R`, the script you wrote,
- `data.csv`, the data that was created by calling `myscript.R`.

Step 6: Copy a script from your computer to the server

If you have a script on your computer that you would like to copy to your folder on the server, you can do this in the terminal using the **scp** (secure copy) command.

In a terminal, navigate to the folder containing your script (suppose it is named `otherscript.R`) and then type the following:

```
scp otherscript.R abc123@macomp001.ma.ic.ac.uk:~/myfolder/
```

This will copy the file to `myfolder` on the server. Note that the colon `:`, the tilde `~` and the slashes `/` are important.

You will be prompted to enter your password again.

Step 7: Create a helper file to use queue

To submit your script to the cluster to be executed, you need to create a ‘helper file’ that will submit your job to the queue. An example is below, which can be saved as **qjob.sh** in the **same folder as your script**, i.e. `myfolder`:

```
#!/bin/bash

#PBS -N RJOBNAME

#PBS -m bea

#PBS -q standard

cd ${HOME}/myfolder

/usr/bin/R --vanilla < myscript.R > myoutput.out
```

It is important that you use all the symbols correctly. For example, `${HOME}/myfolder` is the folder where the script is saved.

The bottom line tells the server to use **R** (for a different language, you will need to change this line).

The top four lines give the job a name (*RJOBNAME* - you can rename this), and the line *#PBS -m bea* tells the server to **send you an email when your job starts execution AND when it is completed**; this can be very helpful for jobs that could run for several hours.

Step 8: Submit your script to the job queue

If you have done everything as above and you are in the correct folder e.g. myfolder, then simply type: `qsub qjob.sh` to submit your job to the queue.

To check on the status of your job, you can use the command `qstat` or the command `showq`. See http://sysnews.ma.ic.ac.uk/compute-cluster/ for further details.

Step 9: Copy a file from the server to your personal computer

To copy a file (results) from the server to your local machine, it is very similar to Step 5, except the order of commands is reversed. Open a terminal and navigate to the folder to which you want to copy the file. Then type (if your College ID is `abc123`):

```
scp abc123@macomp001.ma.ic.ac.uk:~/myfolder/data.csv ./
```

where `data.csv` in the folder `~/myfolder` on the cluster is the file you wish to copy.

You will be prompted to enter your password again.

Note that the `./` means ‘here’.

4. Final thoughts and tips

- It is better to save any output/results to a file on the server, which you then copy to your local machine and analyse later (seperately), rather than trying to also analyse the results on the server.
- Consider running any job that takes more than 15 minutes on your local machine on the server, especially if you think you may need to repeat a similar experiment later.
- Before you run a job that will take a long time, **first run the same script but for smaller parameters** to make sure the job finishes smoothly. For example, before trying to a script that executes 1000 trials of something, change 1000 to a smaller number such as 5, that will run in a few minutes. If everything runs smoothly, change the parameters back to the values for the larger simulation. There is nothing worse than waiting hours for a job to complete, only to find out it fails because of an error.
- Become familiar with using a terminal-based editor such as nano or vim for basic editing; then if you need to change parameters in your script, you can do this easily.

5. Further information on the cluster

- Statistics computing resources: https://sysnews.ma.ic.ac.uk/stats
- Resources reserved for MSc in Statistics students: https://sysnews.ma.ic.ac.uk/stats/MSc_compute_servers.html
- Statistics HPC cluster: https://sysnews.ma.ic.ac.uk/stats/latest-news_070122.html
- Accessing the cluster from outside the College: https://sysnews.ma.ic.ac.uk/accessing_Maths_systems_from_outside_college.html