Simple parallelisation on hpc

R.J.B. Goudie

March 8, 2017

But R is R is R...?!

R code that runs on your desktop/laptop will (almost) always run on the HPC just fine without *any* changes to the R code whatsoever.

This talk is about using the **standard R code** you already have, but running it lots of times efficiently, for example for

- Simulation studies
- Multiple-chain MCMC
- GWAS-type problems
- etc...

i.e. doing *almost* the same thing lots of times, but rather than doing one after the other, doing all of them simultaneously.

Each task must be completely independent

Notation and caveats

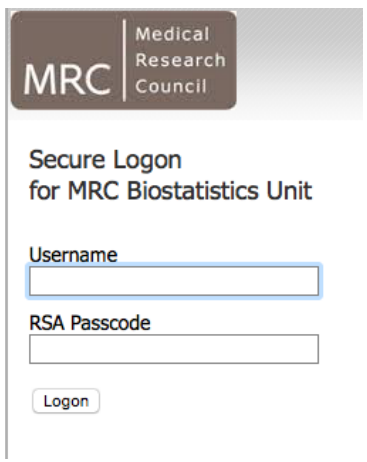
1. Everywhere where you see `abc123` in this talk, replace with your Cambridge CRSID (Raven) username.
2. When I refer to a file as `~/.ssh/config` this means the file `/home/abc123/.ssh/config` i.e. a file called `config` in the `.ssh` folder in your home folder.
3. node = a single one of the many computers that make up the overall HPC, each of which has a processor on it
4. core = a processor has multiple 'cores', each of which can be doing something different
5. Slurm = software that controls the order that jobs are run on the HPC

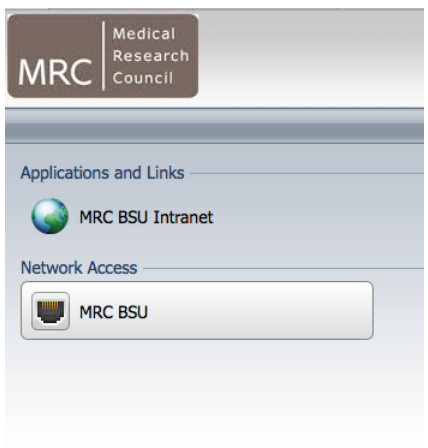
Caveat: I use the HPC from my Mac, and use the command line. I might have made mistakes about the situation if your desktop is Linux or Windows.

Faff 1: SecurID

If you are not using your BSU desktop you won't be able to log in directly. You have to log in first using your RSA SecurID token.

- Use F5. Go to `https://bsulogin.mrc-bsu.cam.ac.uk`, enter your username and token security number, then click "Netowrk access MRC BSU" button. Install the plugin if requested.





- On iOS (iPad and iPhone), download F5 BIG-IP Edge Client[1]
- You can also use SSH Tunnelling, which is a little more faff to set up.

[1] `https://itunes.apple.com/app/id411062210`

Faff 2: remembering/typing the tediously long command to log in

On Linux/Mac, the usual way to log in is

```
ssh abc123@login-mrc-bsu.hpc.cam.ac.uk
```

Instead (from a Linux/Mac), add the following to the bottom of (or create new file) `~/.ssh/config`

```
Host hpc
  User abc123
  HostName login-mrc-bsu.hpc.cam.ac.uk
```

Now to log in, just type

```
ssh hpc
```

On Windows, use PuTTY[2]

[2] http://www.chiark.greenend.org.uk/~sgtatham/putty

Faff 3: having to type your Raven password incessantly

Set up passwordless SSH on Mac/Linux:

1. Check for an existing "ssh key" by checking for a file called `id_rsa.pub`, `id_dsa.pub` or similar in `~/.ssh/` by running `ls -la ~/.ssh/`. If so, skip step 2, & replace `id_rsa` in step 2+3 with your key's name).
2. Generate a key, and create a "passphrase" for the key

```
ssh-keygen -t rsa -b 4096
```

3. Copy the "public key" to the HPC

```
ssh-copy-id -i ~/.ssh/id_rsa.pub abc123@hpc
```

4. Setup an "ssh agent" to avoid typing the passphrase.

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_rsa
```

Faff 4: your internet connection dies, and you lose what you were doing

- Get SSH to say 'hello' from time-to-time. Add the following lines to `~/.ssh/config` on your desktop/laptop

```
ServerAliveInterval 10
ServerAliveCountMax 200
```

- Use GNU `screen`[3] or `tmux`[4]. For screen, when you log in type

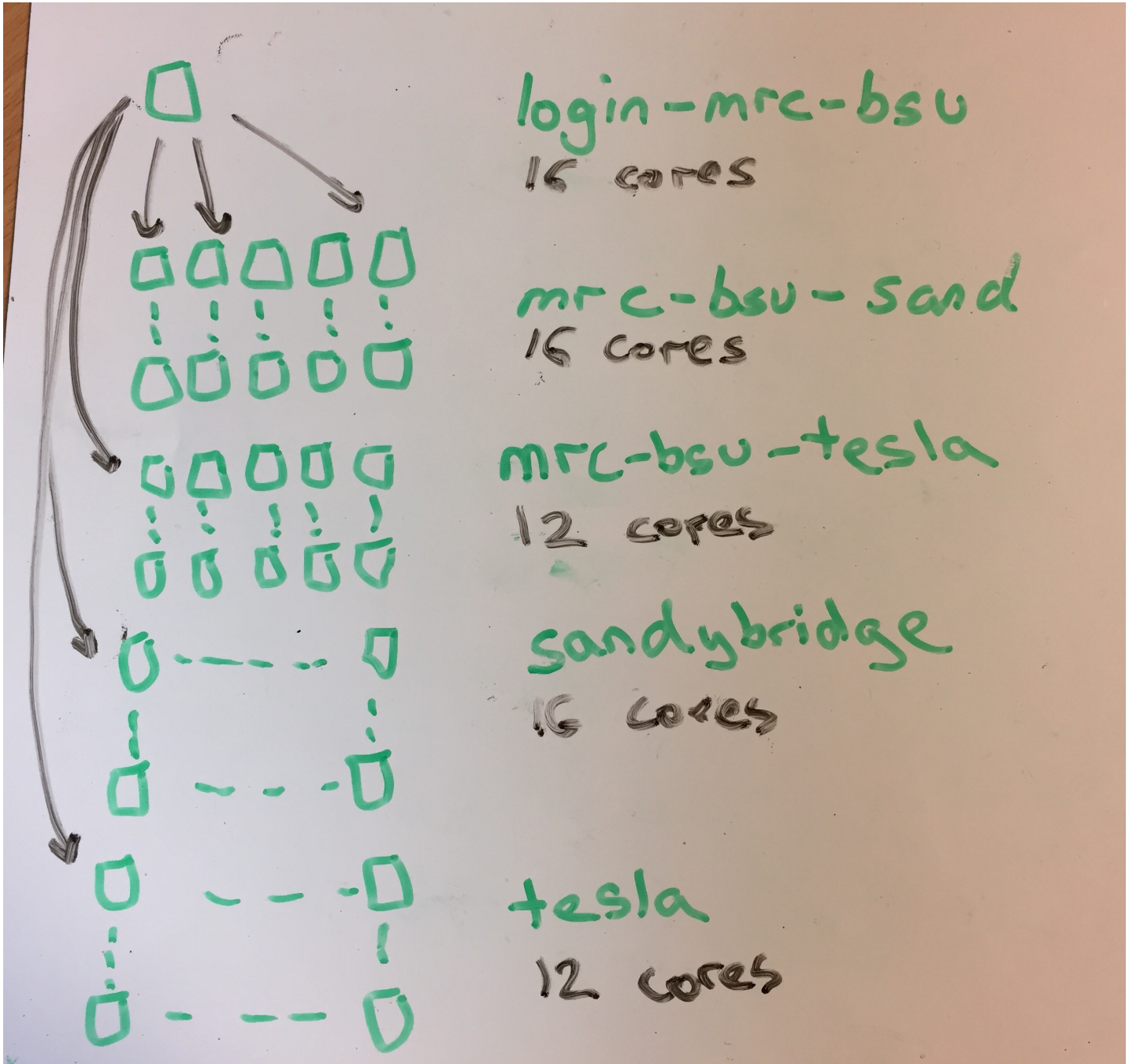
```
screen
```

  This starts a new 'session'. Now if your connection dies, you just log back in with `ssh hpc`, then type `screen -r` and whatever you had open etc is exactly as you left it.

[3] https://www.gnu.org/software/screen/

[4] https://tmux.github.io

Structure of the HPC



Faff 5: load 'modules'

By default, when you log on to the HPC, `R` is not available.

You have to load it using `module`[5] by typing

```
module add R/3.3.2
```

You can view all the software that is available by typing `module avail`.

To avoid having to type this all the time, add the above to the bottom of `~/.bashrc`, e.g.

```
module add R
module add htop
```

[5] `http://modules.sourceforge.net`

Faff 6: having to use the command line

You can partly avoid this.

- You can also use VNC, see instructions[6]
- Or (maybe slower?) change `~/.ssh/config` on your Linux/Mac to

```
Host hpc
  User abc123
  HostName login-mrc-bsu.hpc.cam.ac.uk
  ForwardX11 yes
```

Then once you have logged into the HPC type, for example,

```
module add rstudio
rstudio
```

Do not log in as above, start R, and run intensive calculations – see next slide

[6] http://www.hpc.cam.ac.uk/using-clusters/remote-desktops-and-3d/darwin-turbovnc

login-mrc-bsu, also called the 'frontend node'

This computer `login-mrc-bsu` is what you are using after typing `ssh hpc`

Please don't do anything even vaguely intensive on login-mrc-bsu.

- Don't start `R` or `rstudio` and run your simulation directly like you would on your desktop, unless it is very quick + simple + low memory.
- Even people reading in enormous data/results files or `zip`ping enormous files causes problems from time to time.

Instead: use an 'interactive session'. This is really easy to do. Just type:

```
sintr -A MRC-BSU-SL2 -p mrc-bsu-sand -N1 -t 1:00:00
```

This is then (for 1 hour) your own core on a machine, and you can do whatever you want: including running R, rstudio, zipping 1TB files, ...

How the ‘queue’ works

Roughly first come, first served queue (but not quite).

- ‘Fairness’ is only considered when a slot is available.
- You never get kicked off until your time is up, even if you are using the whole cluster and have been for hours.

Please don’t submit jobs that use 100s of cores for hours, except if you use ‘rate limiting’.

Batch jobs

Two example batch scripts are put in your filespace on the HPC, but they include a line `Do not change` ... which does need to be changed for the mrc-bsu cluster...

Instead, you might find my BSU-specific examples useful:

`https://github.com/rjbgoudie/bsu-cluster`

Running basic 1 core R script

To run a R script called `myrfile.R` for 1 hour on 1 core:

1. Download a submission script, e.g.[7] by clicking 'Raw' then saving the file.
2. Copy `myrfile.R` and `slurm_submit.mrc-bsu-sand` to the HPC. Easiest to use a SFTP program: e.g. Filezilla[8], or lots of other choices for Mac[9] and Linux[10]; WinSCP[11] on Windows.
   Connect to `login-mrc-bsu.hpc.cam.ac.uk` with username `abc123`
3. Log on with `ssh hpc` in a command line window.
4. Submit it:

```
sbatch slurm_submit.mrc-bsu-sand myrfile.R
```

[7] https://github.com/rjbgoudie/bsu-cluster/blob/master/submission-scripts/r-single-core/slurm_submit.mrc-bsu-sand

[8] https://filezilla-project.org

[9] http://apple.stackexchange.com/a/25667

[10] http://askubuntu.com/a/109020

[11] https://winscp.net/

Array jobs

Running the same R file 1000 times is easy: download example[12], or just add the following to submission script (after e.g. the `#SBATCH --ntasks=1` line)

```
#SBATCH --array=1-1000%50
```

The `%50` bit means at most 50 tasks will run simultaneously. This 'rate limiting' helps smooth about the peaks and troughs in demand.
R can find out which of the 1000 tasks it is processing by

```
task_id_string <- Sys.getenv("SLURM_ARRAY_TASK_ID")
task_id <- as.numeric(task_id_string)

seed <- c(425, 3453, ......., 223, 232)[task_id]
set.seed(seed)
rnorm(100)
```

[12] https://github.com/rjbgoudie/bsu-cluster/blob/master/submission-scripts/r-single-core-array/slurm_submit.mrc-bsu-sand

Array jobs admin

Remember to add `task_id` to the filenames of all your output, otherwise they will overwrite each other. For example,

```
filename <- paste0("samples_", task_id, ".rds")
saveRDS(samples, file = filename)
```

Monitoring progress

To see the status of your jobs

```
squeue --user=abc123
```

If you want this to update 'live', use `watch`[13]

```
watch squeue --user=abc123
```

To see a summary of how much of our cluster is being used currently

```
sinfo --partition=mrc-bsu-sand,mrc-bsu-tesla --format="%P %C"
```

I wrote a (trivial) script that presents this marginally more clearly[14]

[13] See `http://www.linfo.org/watch.html` or `https://linux.die.net/man/1/watch`

[14] `https://github.com/rjbgoudie/bsu-cluster/blob/master/bin/bsu-queue-status`

Switching to the bigger University HPC (Darwin/Wilkes)

BSU have 200,000 (?) free hours on Darwin and Wilkes, and anyway Ela tells me money not an issue.

It is very easy to switch, so please do if you are doing something enormous

All you have to do is change 2 lines in your submission script.

For standard nodes ('Darwin'/'sandybridge'), change

1. `#SBATCH -A MRC-BSU-SL2` to `#SBATCH -A MRC-BSU-SL3`
2. `#SBATCH -p mrc-bsu-sand` to `#SBATCH -p sandybridge`

For nodes with GPUs ('Wilkes'/'tesla'), change:

1. `#SBATCH -A MRC-BSU-SL2-GPU` to `#SBATCH -A MRC-BSU-SL3-GPU`
2. `#SBATCH -p mrc-bsu-tesla` to `#SBATCH -p tesla`

Everything else is identical.

Automating more

- Chris Wallace has some Ruby scripts[15]
- There is an R-package called `rslurm`[16], but I've never used it.
- I use my own (completely undocumented) R-package `mngr`[17], which automates running 'workflows' of R files; using `git` to make jobs reproducible; handling multiple `git` branches; shortcuts for dull admin of filing logs etc.

[15] `https://github.com/chr1swallace/slurmer`
[16] `https://cran.r-project.org/package=rslurm`
[17] `https://github.com/rjbgoudie/mngr`

Further help

- The university's documentation[18] for the university-wide HPC, which is basically identical to the BSU cluster.
- For technical documentation, use, for example, `man sbatch` on the command-line. Note the HPC uses Slurm version 14.11.8, which is a couple of years old, so the Slurm web documentation[19] for version 17.02 is occasionally misleading.

[18] `http://www.hpc.cam.ac.uk/using-clusters`

[19] `https://slurm.schedmd.com/man_index.html`