Cluster computing with R using O2

Isabel Fulcher

August 20, 2018

1 Getting started

Once you have an account on O2, you can logon via your terminal window. For windows users, you should download PuTTY to access a terminal window.

- Open terminal on your computer
- To logon to O2, type the following command (you need to be connected to the internet):

  ```
  ssh -l <userid> o2.hms.harvard.edu
  ```

- You will be prompted to enter your password
- To navigate your folders on the server, you can use basic terminal commands. Here are some common ones:
  - `cd` changes your directory so you should specify the path afterwards
  - `cd ..` moves one level up from the current directory/folder you are in
  - `ls` lists the contents of the current directory/folder you are in
  - `mkdir new_folder` creates a new directory/folder called `new_folder`
  - `rm new_folder` deletes the `new_folder` folder (be careful!)
  - A comprehensive list can be found here

2 Working with R on the server

There are two reasons I work with R on the server: (1) to install packages and (2) test code.

- To open R on the cluster, you need to start an interactive job

  ```
  srun --pty -p interactive --mem 4G -t 0-06:00 /bin/bash
  ```

  You can change the specified time and memory

- Once your resources have been allocated, enter the following command

  ```
  module load gcc/6.2.0 R/3.4.1
  ```

  then enter `R` . Note you can also use `R` 3.4.1 or 3.5.1, but you will need to reinstall your packages on each version.

- You should now be in the R environment. You can directly enter `install.packages()` . This will ensure that the package is installed on your server.
- To close out of R, enter `q()`
- More information about using R on O2 can be found here

3 Steps to submit a job

The O2 cluster uses SLURM to submit jobs, so that is what is outlined here. The Odyssey (FAS computing cluster) also uses SLURM, so if you get an account here, you will already be familiar with the syntax. There are other schedulers, such as LSF and SGE, that you may encounter on different clusters.

Step 1. Prepare your R script(s)

Before you submit a job to the cluster, you likely have R code that you want to execute. This code needs to contained in an R script with the following considerations:

- All packages needed for your code must be loaded at the beginning of the script (this is very similar to R markdown). Also, make sure they are installed on your server (see Section 2).
- If you pass any values into your R script (i.e. array jobs), you need to make sure these are defined in your R script
- If you reference other R scripts or data files in your R script, double check the file path!
- At the end of your R script, you should consolidate your results. This may involve reporting one table or multiple different objects. If you are doing the latter, it may be helpful to consolidate this into a list and use `saveRDS()` .
- Note the results will be saved in the server folder where your shell file lives, so if you would prefer it be somewhere else, you need to specify that file path!

Step 2. Create your submission (shell) script

A shell script tells the cluster everything it needs to know to complete your job. Most importantly, it details the name and location of the R script you want to run (from Step 1). It also will contain information on the time and storage required for the job, where to send error messages, and the partition to run it on (short, medium, long). You can use a basic text editor to create your shell scripts. The O2 wiki page gives a detailed overview on the exact layout of these files.

Step 3. Transfer all necessary files to the cluster

Using a file transfer software (i.e. FileZilla), you need to move all of the files necessary for your job from your computer to the server. The first time you use FileZilla, you will need to logon to the server via FileZilla and enter your account information.

- Open FileZilla
- The host name you should enter is: `transfer.rc.hms.harvard.edu`
- The port is 22
- Enter your own username and password

Note: keep track of your file locations and setup a workflow that makes the most sense to you!

Step 4. Run your job

- Log on to O2 and navigate to the folder that contains your shell script (which should also contain your other files for the job)
- Enter the following command to execute your shell script:

```
sbatch myshell.sh
```

- If you are submitting job array, the command will look something like:

```
sbatch --array=1-10 myshell.sh
```

- There are numerous ways to get fancy with submitting jobs. Once you get to this stage, the best solution is to google.

Step 5. Check on your job

- To check on the status of all your currently running jobs (replace userid with your O2 login / ecommons ID):

```
squeue -u <userid>
```

- To check on the status of a specific job (you need to replace jobid with the numerical ID assigned to the job):

```
sacct -j <jobid>
```

- To get some more information on the above, such as time elapsed,

```
sacct -j <jobid> --format=jobid,state,elapsed
```

- To cancel an existing job,

```
scancel <jobid>
```

Step 6. Transfer files back to your computer

When a job is complete, you will get an e-mail notification. Hopefully, the e-mail subject line will read “COMPLETE”, but more often than not, it will say “FAILED”. In the case that your job successfully completed, you should:

- Log on to FileZilla
- Transfer the output files back to your computer
- If your results are spread out over multiple files (happens with array jobs), you will want to consolidate them. I recommend creating another R script that gathers all of your results. This way you can view them all in one place instead of trying to view or copy and paste from multiple files.

If your job failed, then you will likely want to find the reason for this. Sometimes, this will be apparent from the subject line of the e-mail. However, you will often need more information. Fortunately, SLURM creates error files containing details about the error. You can also download these files to your computer and read them in a texteditor. Or you can use the following shortcut to view the files in the terminal,

```
vi <file_name>
```

and then you can use `:q` to close the screen when you are done.

4 Things you may want to look into...

- You can setup a `.bashrc` to create shortcuts for logging on to O2
- Instead of using FileZilla, you can use your GitHub account to transfer files (great for version control)
- You can also pass in arguments from your sbatch command without using an array (or in addition to an array!).
- If you want to run a job in parallel, use the doParallel package in R. This is nice because it will gather all your results for you (i.e. you will not get a bunch of output files like you do for an array job).