1 First time on Cluster

In order to get an account on the cluster, you need to email Jacob Bien who will email the Cornell Center for Advanced Computing (CAC) and ask them to create an account for you on the DSS cluster. Shortly thereafter, you should receive an email with a username and password. The first time you attempt to use the cluster, open a terminal and type:

```
ssh lkm54@linuxlogin2@cac.cornell.edu
```

Enter the password they give you. After entering your password, you will need to create a new one. Follow the directions on-screen for doing this.

2 Changing your password

You can change your password at any time. Make sure you have no open connections to any machines, open a new terminal, and type

```
ssh lkm54@linuxlogin2@cac.cornell.edu
```

you will need to enter your password. To change your password, simply type

```
passwd
```

and follow the on-screen directions. You will need to enter your old password and then the new password. Be sure that your chosen password follows the requirements at `http://www.cac.cornell.edu/wiki/index.php?title=V4_Linux_Cluster#Changing_a_Password_at_Any_Time`.

3 Logging onto DSS Cluster

Once you have logged onto the CAC cluster and changed your password for the first time, you are ready to begin using the DSS cluster. Unless you need to change your password, you no longer need to log into the CAC cluster - you can log onto the DSS cluster directly. To do this, type

```
ssh lkm54@dss@cac.cornell.edu
```

note that

```
ssh lkm54@dss1@cac.cornell.edu
```

will also work. Once logged on, you can submit jobs (see section below).

3.1 Logging onto Specific Nodes

Once you are logged into the DSS cluster, you can log into a specific node. The website `http://dss.cac.cornell.edu/ganglia/` shows you each node on the DSS cluster and the activity on each one. Observe that above each graph on the website is a name for that node (e.g. `compute-1-1.dss`). To log onto this node, simply type

```
ssh compute-1-1
```

The first time logging onto each node, you will get a warning message stating that the node is permanently being added to you list of known hosts. Once logged onto a specific node, you will only be utilizing that node, so you can run programs such as R locally through the terminal without issue.

4 Submitting a Job to the Cluster

1. Begin by creating the R file that you desire to run on your own computer. Make sure that the file is "error free" and compiles. We will call the file HelloAgain.R.

2. Once the file is created, open a terminal and type

   ```
   ssh lkm54@dss.cac.cornell.edu
   ```

   to connect to the DSS cluster - you will need to enter your password.

3. If you would like to make a new directory for the file, you can do that with

   ```
   mkdir ~/hello
   ```

   which will create a folder/directory named hello. You can see all folders on the cluster with ls

4. Now we need to transfer the R script we wrote to the cluster. In a new terminal window, type

   ```
   scp <local path>/HelloAgain.R lkm54@dss.cac.cornell.edu:~/hello
   ```

   scp stands for secure copy protocol and as such, you will again need to enter your password.

5. Now we need to create a shell script. We will do so using emacs. Begin by changing the directory to 'hello' and opening emacs and creating a file named HelloAgain.sh

   ```
   cd ~/hello
   emacs HelloAgain.sh
   ```

   After doing this, the terminal should switch to an emacs editor. Input the following text:

   ```
   cd ~/hello
   R CMD BATCH --no-restore HelloAgain.R
   ```

   Now save the file by pressing Ctrl-X CTRL-S and close emacs by pressing Ctrl-X Ctrl-C

6. Now we can submit the job to the cluster using qsub

```
qsub HelloAgain.sh
```

To check the status of your job, type `qstat` and to kill the job you can type

```
qdel job_id
```

where `job_id` can be found from the list in `qstat`

5 Viewing/Saving/Loading R Output

Now, once the job is complete, we want to view the output. By default, the output from the R terminal is saved as the FileName.Rout (here this is `HelloAgain.Rout`). To view the output of the R terminal, type

```
cat HelloAgain.Rout
```

Any plots generated will be saved to a file called `Rplots.pdf` To view these plots, we first need to download them to our home computer. Open a new terminal window. If we wanted to save the plots to a folder on the desktop called `Rplotstest`, we would type

```
scp lkm54@dss.cac.cornell.edu:~/hello/Rplots.pdf /Users/lucasmentch/Desktop/Rplotstest
```

and we will again need to enter our password. After this, the plots should be saved as a .pdf file in the Rplotstest folder on our home desktop. Note that if you save anything in the R file (e.g. write a table or data frame) then that file will also be saved to your drive on the cluster. You can download and view it in the same way.

Now suppose that we wanted to be able to use a variable from the program in another program on our home computer. For example, I submit an R program to the cluster which creates vectors x and y and I want to be able to use x and y (after they are computed) on my home program. To do this, we need to save the x and y to an r data file, save it on the cluster, and then export it back to our home computer. At the end of the R file, include the line:

```
save(x,y,file='/home/fs01/lkm54/xy.rda')
```

This will save an R data file called "xy" to your cluster folder. You can download it to your home computer as before. Once you have it saved to your home computer, you can double-click on it and R will open and load it. You know have the variables 'x' and 'y' to use as usual, just as if you run the program on your home computer.

6 Viewing your Cluster Files and Folders on your Home Computer

Note that this only works if you are on campus connected to Red Rover - from off campus, you need to install a Cisco client. From on campus, click "Finder" and from the apple menu, select "Go" and then "Connect to Server". For the server address, enter

```
smb://storage01.cac.cornell.edu/lkm54
```

and click connect. Change the user name to "lkm54" and enter your cluster password and click connect. After connected, when you open a finder window, underneath the shared folders there will be a drive called

```
storage01.cac.cornell.edu
```

This is your folder that contains all of your files and folders that are being stored on the network/cluster. Note that this is not your Z drive.

7 Difference between qsub and R CMD

From the head node (i.e. the node you are in once you log into DSS) if you submit a job using `qsub`, it will select a "good" node on which to run the program. Alternatively, you can log into a specific node (see instructions above) and then submit a job using `R CMD`. Do not use `R CMD` from the head node - this will run the job from the head node.

8 Creating your R Library

Note that R packages are not regularly updated on the cluster, so you need to do something in order to be able to use functions in packages in R that may not be in the version of R that is on the cluster. To make your own R library on the cluster, first log onto the css cluster. Once logged in, go to your home directory by typing `cd ~`. Once in your home directory, make a new directory called `Rlibs` by typing

```
mkdir ~/Rlibs
```

Next, we create a file called `.Renviron` and edit it using emacs:

```
emacs .Renviron
```

Once emacs opens the file, type `R_LIBS=~/Rlibs` and then save the file and exit emacs by typing `Ctrl-X Ctrl-S` followed by `Ctrl-X Ctrl-C`. Now, start R by typing `R` at the prompt. Once R opens, type `.libPaths()`. You should see something like `"/home/fs01/lkm54/Rlibs"` appear in the output. Type `q()` to quit R. Now, to install packages that you want to use, open R by typing `R` at the prompt. Once R opens, type `install.packages("packageName")` and this package will automatically be saved in your personal R library.