Using the UTC SimCenter

First, download and connect to the UTC VPN with your UTC User ID and password.

* Instructions:
  + <https://blog.utc.edu/it-knowledge-base/connect-your-windows-computer-to-the-utc-vpn/>
* Download links only:
  + <https://netaccess.utc.edu/vpn/>

Open Windows Command Prompt or Mac Terminal.

* Cheat sheet for Windows Command Prompt:
  + <http://www.cs.columbia.edu/~sedwards/classes/2015/1102-fall/Command%20Prompt%20Cheatsheet.pdf>
* Cheat sheet for Linux/Mac Terminal:
  + <https://www.git-tower.com/blog/command-line-cheat-sheet/>
  + You also will need to use these commands when using the cluster!

Note: It is easier to have two windows open - one working on your personal computer, and one connected to the SimCenter. This way you don’t have to enter and exit frequently.

Connect to the SimCenter server with the following command:

$ ssh user@ts.simcenter.utc.edu

You will be asked for your password. Once you are logged in, you will be in your personal directory. You can check this with the command $ pwd, which should return /home/user. You can see what is in your own directory with the command $ ls. The command $ ll will show you more details about the contents. If you type the beginning of a unique file name, you should be able to hit the tab button and the rest of the file name should appear.

You can use the command $ cd .. to move into the home directory and the command $ ls to see all the directories there. All of the personal directories are password-protected. You can also see who else is using the cluster with the command $ w.

When running jobs, you will use the code and data you have in your personal directory on the cluster. There are several ways to get files into your directory:

* You can write files in your personal directory.
* You can clone repositories from GitHub.
  + From your personal directory on the cluster, use the following code:

$ git clone https://github.com/your-repository-name

* You can copy files from your personal computer.
  + From your personal computer, use the following code:

$ scp filename user@ts.simcenter.utc.edu:./.

Once you have the necessary code and data files in your directory, you will need to create a job file in order to submit your job to the cluster. This will be a .pbs text file, which will contain some specific details about the job.

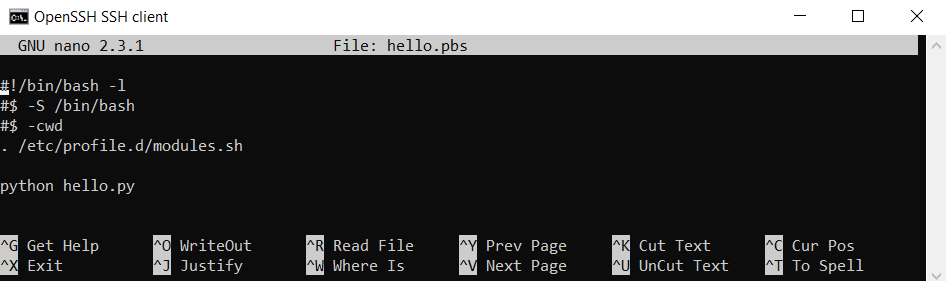
You can create this file with the touch command.

$ touch filename.pbs

Then, you can edit the file with the nano command.

$ nano filename.pbs

This will open a text editing window that will look something like the image below:

There are several arguments that can be used in a .pbs file. Include any arguments you want and the commands you need to run your code. In this example, I am writing a job file called hello.pbs that will run a Python file called hello.py.

Here is a list of arguments that can be used, taken from the SimCenter Wiki:

-S path: Specifies the interpreting shell for the job.

-N name: Specifies the job\_name.

-V : Specifies that all environment variables active within the qsub utility be exported to the context of the job.

-pe <pe> N: Specifies the parallel environment required for the job with N processes launched.

-cwd : Specifies that all environment variables active within the qsub utility be exported to the context of the job.

-e path: Defines or redefines the path used for the standard error stream of the job.

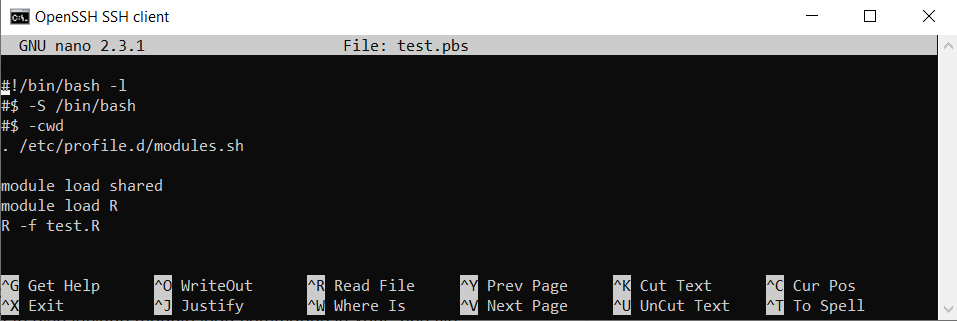
-o path: Defines or redefines the path used for the standard output stream of the job.

-j y|n: Specifies whether or not the standard error stream of the job is merged into the standard output stream.

-l resource=value: Launch the job in a Grid Engine queue meeting the given resource request list.

If you are running a Python file, the code should run with the command python filename.py.

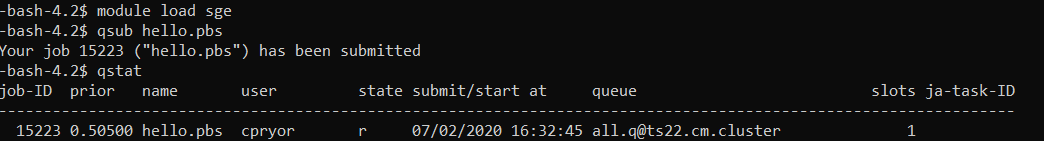
If you are running R code, you will need to also include module load commands in your .pbs file. In the following example, I am writing a job file called test.pbs that will run an R file called test.R.

When you are done editing your job file, exit and make sure your changes are saved.

Load the SGE job scheduler for the cluster and submit the job with the following commands.

$ module load sge

$ qsub filename.pbs

The command $ qstat allows you to check the status of your job. It will let you know if the job is in the queue or if it is running. This will look something like the image below. If it doesn’t return anything, then the job is complete.

Once the job is complete, you should be able to find any output files or error files in your personal directory. You can use the $ cat filename command to see the contents of a file.

**Extra Resources**

* SimCenter Wiki:
  + <https://wiki.simcenter.utc.edu/doku.php?id=ts117:sge>
* Dr. Qin blog on computing environments:
  + <https://hongqinlab.blogspot.com/2020/06/ts-anaconda-qin-log.html>
* Exercises/examples on Dr. Qin’s GitHub:
  + <https://github.com/QinLab/simcenter_clusters_excises>
* PBS Information from University of Minnesota:
  + <https://www.msi.umn.edu/content/job-submission-and-scheduling-pbs-scripts>