Running R Scripts with SharcNet

What is SharcNet and Compute Canada?

SharcNet is a community of 19 academic institutions (one of which is York University) that contribute funding for purchasing supercomputers. Compute Canada is Canada's national high-performance computing (HPC) system. As of March 2022, SHARCNET and Compute Canada have been superseded by the Digital Research Alliance of Canada (The Alliance).

Why are Supercomputers Useful?

Computationally heavy tasks that may require days can be completed in less time using supercomputers. By reducing the time it takes to run simulations, researchers will be able to finalize the results of their study more efficiently.

Initial Important Terms

Term	Definition
Terminal	Computer users often interact with their computers using a graphical user interface (GUI). However, the **terminal** provides computer users with a text-based method to interact with their computers. For instance, rather than clicking a file on my desktop (using the GUI), I can execute commands in the terminal to access a file on my desktop.
Clusters	A set of computers that behave as one unit.
Secure Shell (SSH)	"A cryptographic network protocol for operating network services securely over an unsecured network"

Steps for using SharcNet

Preliminary Steps

- Familiarize yourself with basic commands:
    - Although not mandatory, I highly recommend that new users familiarize themselves with the basics of the command line, as they will use it to log in and run jobs on clusters.
- Popular commands:

Commands	Description
cd	Change directory

cd ..	Go back one directory
pwd	Current directory
ls	Lists all the files in the specified directory
ssh-keygen	Creating an SSH Key
ssh	Secure shell (SSH)
cat DirectoryToPublicKey	Public key
module list	A list of the current modules installed
module spider r	A list of available R versions
module spider r/4.3.1	Information about this specific R module (including other modules that are required)
module load StdEnv2020 StdEnv2023 r/4.3.1	Loading the R module and other required modules
nano FileName.sh	Creating a text editor file
sbatch FileName.sh — running job script	Running the job script
squeue -u YourUserName	Checking the status of your job

- Create a Compute Canada account:
    - Visit the Compute Canada home page to register.

- Students who are interested in using the clusters will require a sponsor (i.e., permission from a professor at one of the funding institutions that has a Compute Canada account).
- Once you have logged in, I recommend verifying the information for your account.

- Open *Terminal*
  - The *Terminal* is where you log in and submit jobs to the cluster.
  - There are different ways that you can access *Terminal*:
    - iOS/Linux → *Terminal* is already installed on the computer
    - Microsoft → must download software containing a *Terminal* (e.g., *R*, *MobaXTerm, etc.*)

- Creating a Secure Shell (SSH) Key (Optional):
  - A secure shell key provides users additional security when logging into a cluster.
    - Once you have created your key, you will be provided with a public and private key.
    - *Note: do NOT use your SSH key on another computer*

  - Steps to create a secure shell key:
    1. Open *Terminal*

    2. Check your existing SSH keys:
       - Existing SSH keys will be located in the .ssh directory on your computer.
       - To navigate through directories on your computer, use the **cd** command.
       - Use the **ls** command to check your existing SSH keys.
       - *Note: do NOT use SSH keys you have no recollection of creating.*

    3. Create a key:
       - Once you are in the SSH directory, you can create a key using the **ssh-keygen** command
       - Different types of keys can be generated. By default, the *RSA* key is generated (unless otherwise specified)
       - Arguments can be used to modify the key:
         - For example, ssh-keygen **-f *NameOfKey*** → This argument allows users to customize the name of their key, which can be helpful for organization purposes.

4. Create a passphrase for your key:
    - I.e., create a password
    - According to Alliance Canada, the passphrase should be 15 characters and include a mixture of numbers and uppercase, lowercase, and special characters.

5. Find your public key:
    - Use the command **cat .ssh/NameOfPublicKey.pub** in your home directory to find your public key.

6. Add the public key to CCDB:
    - Copy the public key
    - After logging into CCDB, click *My Account* > *Manage SSH Keys*
    - Paste the public key into the *SSH Key box*
    - Add key

Logging into a Cluster

- There are different ways to log into a cluster:
    - CCDB Username:
        - Command: **ssh CCDBUsername@Address**
        - Address: the address of the cluster you are interested in logging into.
        - The address for the different clusters can be found via the links provided below:
            - Béluga
            - Narval
            - Cedar
            - Graham
    - SSH Key:
        - Command: **ssh -i path/to/private/key/NameOfPrivateKey CCDBUsername@Address**
        - Address: the address of the cluster you are interested in logging into.
        - *Tip: use the pwd command in the .ssh directory to determine the path to your private key.*

Running R in Login Node

- What are modules?
    - Modules allow users to use different installed programs.
    - Like packages in R, some modules are automatically active when you open the terminal. However, to use some modules (like the R module), you must install the module before initiating a job on the cluster.

- We are interested in loading the R module onto the cluster for this tutorial. However, the general steps below can download any module onto a cluster.

- Steps for running basic R operations within a cluster login node:
  - *Note: you should not run jobs longer than two minutes*
  1. Type '*R'* in the terminal within the cedar cluster
     - The output will display a list of available R versions and the dependency modules

  2. Select which version R version you are interesting in loading on the cluster

     - Run regular commands, including commands like 'install.packages'

Creating a Job Script (for an R Job)

- What is a Job Script?
  - You will be directed to a login node when you log into a cluster. The login node is meant to complete computationally small tasks (e.g., creating a job script, transferring files, etc.); however, computationally heavy tasks are completed on compute nodes.
  - SLURM (i.e., the scheduler) will assign your job to a compute node. However, the scheduler requires basic information about your job to do this.
    - Therefore, the purpose of the job script is to provide the scheduler with the necessary information regarding the resources required to complete your job. Once the required resources are available, the scheduler will assign your job to a compute node.

- Steps for creating a job script:
  - *Note: you can create the job script in any text editor (e.g., RStudio)*
  1. Go to the directory/folder you transferred your script file to:
     - Navigate through the folders in the cluster using the **cd** command

  2. Open a text file in *Terminal*:
     - Command: **nano NameOfJobScript.sh**
     - This command will open a text file editor in your terminal.

  3. Create the Job Script:
     - Example Job Script:

```
#!/bin/bash
#SBATCH --cpus-per-task=32
#SBATCH --mem-per-cpu=4000
#SBATCH --time=03-00:00:00 # DD-HH:MM
#SBATCH --account=def-cribbie
#SBATCH --mail-user=cribbie@yorku.ca
#SBATCH --mail-type=ALL
module load r/4.3.1
Rscript play_7.R
```

- Mandatory lines in job script:
  - Line 1 – each job script must start with the first line
  - Line 6 – you must inform the scheduler to install any modules that are not automatically installed on the cluster
  - Line 7 – you must indicate the name of the file you wish to run on the cluster
- Optional lines (i.e., different pieces of information you can provide the scheduler within your job script regarding your job):

Element	Purpose
time	The approximate time it will take to complete your job (hours:minutes:seconds)
mem-per-cpu	The amount of memory required to complete your job *Note: if this line is not included, the default is 256MB*
cpus-per-task	The amount of cores required to complete your job *Note: if this line is not included, the default is one core*
job-name	The desired name of your job
mail-user **mail-type**	Include to get an email notification of the start and end of the job

Note: you must provide the scheduler with accurate information regarding your job because this can impact how quickly your job gets assigned by the scheduler.

4. Save job script:
   - Save your job script using **control + x + y**

Transferring Files

- What is Globus?
  - Globus is a service that allows researchers to securely transfer files between two "resources."

- Although Globus is recommended for large file transfers, this system is much more user-friendly than alternative methods.

- Steps for using Globus:

1. Set up a Globus account (using Compute Canada login)
    - Visit the Globus page to create an account.

2. Download Globus Connect Personal on your personal computer and create a collection
    - Visit the Globus Connect Personal page to download.
    - Once you have installed Globus Connect Personal, you must create a "collection" on your computer. Think of this collection as a starting and ending point for files to be transferred to and from your computer.
    - *Note: by default, windows users will only have access to their documents directory on Globus. To change this default setting, go to option > access. From here, you can select additional directories that Globus can access.*

3. Select your collection and where you want to store the file (on the cluster)
    - *Note: you must turn on Globus Connect Personal on your computer before selecting your collection.*
    - You will be required to select your personal collection and the collection of the cluster to which you are interested in transferring files.
    - Visit the links below to determine the name of the collection for the desired cluster:
        - Béluga
        - Narval
        - Cedar
        - Graham
    - After granting Globus permission on your computer, you will be able to navigate the folders and files on your computer through Globus. Navigate through your folders and files and select the file you wish to transfer to the cluster.
    - Each cluster contains different types of storage, which differ in various properties (e.g., the amount of storage, whether files are backed up, etc.). I recommend exploring the properties of the storage and transferring the files to the storage that best suits your needs.
        - For information about the different types of storage, please refer to this link.

4. Initiate transfer
    - Click the arrow (on the side of your collection) to initiate the transfer.

        - You will be notified via email once your file has been successfully transferred. You can check that your file has been transferred on *Terminal* by going to the directory containing the file (using the **cd** command) and executing the **ls** command.

Submitting Jobs

- Send your job script to the scheduler using the **sbatch NameOfJobScript.sh** command.
    - *Note: make sure all your files (e.g., job script, R script, etc.) are in the same directory.*
- Once you have sent your job script to the scheduler, you will be provided with a Job ID. Keep note of this Job ID because it will be important for file retrieval later.

Checking Job Progress

- To check the status of your job, use the **squeue -u YourUserName** command.
- Running this line of code will provide information about all the jobs currently pending. The ST column includes information about your job status (e.g., PD → pending).

Viewing Output

- Each output file will be named using a standard format (unless otherwise specified by the user) – slurm-ProjectID.out
    - The output file will be saved in the same folder as your job and project script.
- Once your job has been successfully completed, you should check that the output file has been saved.
    - To do this, you will want to navigate to the folder on the cluster where you have saved your job script and project script file (using the **cd** command). You will then execute the **ls** command, which will list all the files within that folder.
- If your output file is in the folder, you are now ready to transfer the file to your computer using Globus, following the same steps provided above.

Additional Notes

- Whenever you are done with a cluster, end your cluster and terminal session by executing the **exit** command twice.
- When you are not using Globus, you must end your *Globus Connect Personal* session.

Additional Resources

- **An introduction to the command line**
- **Running R script with SharcNet**
- **Using Secure Shell (SSH)**
- **Creating a key pair on iOS and Linux**
- **Information about Modules**
- **Information about Globus**
- **Information about running jobs on SharcNet**
- **Information about SharcNet**
- **Other video resources provided by SharcNet**
- **Additional Information about SharcNet**