

For Dr. Landau's PHYS8602 course

Dr. Shan-Ho Tsai (shtsai@uga.edu)
Georgia Advanced Computing Resource Center - GACRC

January 4, 2023

You will be given a student account on the GACRC's Teaching cluster. Your username will be your UGA MyID and the password is your UGA MyID password. You will be able to access/use this account for the duration of the course. **At the end of the semester your account will be deleted, so make sure to transfer (download) any files that you would like to keep to your local machine/drive by the end of the semester.**

1 Machine characteristics

- Machine name: Teaching cluster
- Login node (node where you login to) name: `teach.gacrc.uga.edu`
- Operating System: 64-bit Linux (CentOS 7.8)
- Types of nodes:
 - The login node is a virtual machine with a 4-core Intel Xeon processor.
 - The compute nodes (nodes where your programs will run) have Intel Xeon processors with 12 cores and 48GB of RAM per node.
 - The cluster also has two interactive nodes, which are identical to the other compute nodes, but they can be used to compile code and run applications that have a graphical front-end, such as graphing programs.
- **NOTE: No jobs should be run on the login node!!!**

2 Where do I find documentation about GACRC and the teaching cluster?

- Webpage: <http://gacrc.uga.edu/>
Provides a general overview of GACRC.
- Wiki: https://wiki.gacrc.uga.edu/wiki/Main_Page
Provides documentation about the teaching cluster, including how to connect to it, how to transfer files, how to compile code, how to run jobs, what software packages are installed, etc.
- Kaltura: <https://kaltura.uga.edu/channel/GACRC/176125031>
GACRC's Kaltura channel provides training videos, including an Intro to Linux Playlist that is available at <https://kaltura.uga.edu/playlist/dedicated/176125031/1-wwkiealj/>

- Training Opportunities: <https://wiki.gacrc.uga.edu/wiki/Training>
GACRC provides free training workshops on how to use computing clusters, how to use Linux, Python programming, etc. You are welcome to register and attend these workshops.

3 How do I access the teaching cluster?

Use Secure Shell (**SSH**) to connect to the login node: **teach.gacrc.uga.edu**. To connect via ssh, you must have an ssh software on your local machine and a connection to the UGA campus network. SSH software is included in recent releases of Unix based operating systems (including Linux and Mac OSX). To connect from a Windows machine, you can use PuTTY (<https://www.putty.org/>). Some information on how to install and configure PuTTY is available at https://wiki.gacrc.uga.edu/wiki/How_to_Install_and_Configure_PuTTY (please use teach.gacrc.uga.edu as the hostname).

To access the teaching cluster you will need your UGA MyID and password, and two-factor authentication with Archpass Duo. For more details on Archpass Duo, please refer to https://eits.uga.edu/access_and_security/infosec/tools/archpass/

If you are connecting from outside of the UGA main campus, you need to connect to the UGA VPN first, then you can connect to the teaching cluster. Information on how to use the VPN is available at https://eits.uga.edu/access_and_security/infosec/tools/vpn/

Detailed information on how to connect to the teaching cluster, including how to set up the SSH client to use two factor authentication is available at <https://wiki.gacrc.uga.edu/wiki/Connecting>

Videos that describe how to connect to the teaching cluster (use the remote host name **teach.gacrc.uga.edu**) are available in our Kaltura channel:

- Connecting to Remote Linux Computers from Windows:
https://kaltura.uga.edu/playlist/dedicated/176125031/1_a6e4voao/1_0zu02g1a
- Connecting to Remote Linux Computers from Mac:
https://kaltura.uga.edu/playlist/dedicated/176125031/1_a6e4voao/1_5voz2lt
- Connecting to Remote Linux Computers from Linux:
https://kaltura.uga.edu/playlist/dedicated/176125031/1_a6e4voao/1_z8tftk87

4 How do I transfer files to/from the teaching cluster?

Users can transfer files between their local machines and the teaching cluster using various programs, such as (a) Filezilla, (b) WinSCP, and (c) a secure copy (scp). To transfer files, you must have a file transfer program installed on your local machine and a connection to the UGA campus network.

All teaching cluster users can access the file transfer node using the hostname txfer.gacrc.uga.edu and using their UGA MyID and the MyID password to authenticate. Two-factor authentication using Archpass Duo is also necessary.

4.1 Using FileZilla

FileZilla (<https://filezilla-project.org/>) is a file transfer program that has a graphical interface with user friendly features, such as “drag and drop” upload and download of files. It is available for Windows, Mac OSX, and Linux operating systems.

In order to use FileZilla to transfer files between a local machine and the teaching cluster, users will need to install the FileZilla software on the local machine. After installation, start FileZilla and follow these steps:

1. Go to File -> Site Manager
2. Add a New Site with the following configuration.

In the General tab select

```
Protocol: SFTP - SSH File Transfer Protocol
Host: sftp://txfer.gacrc.uga.edu
Port: 22
Logon Type: Interactive
User: [your UGA MyID]
```

In the Transfer Settings tab, check the box for "Limit number of simultaneous connections" and set "Maximum number of connections" to 1. If this step is omitted, then you will need to enter your password and Duo authentication for each file that you upload or download. By limiting the number of simultaneous connections to 1, you only need to enter your password and Duo authentication once per session.

With the above settings saved, you should be able to transfer files to txfer using FileZilla. When you open the connection, you will be prompted for your MyID password (unless you use key based ssh). If the password authenticates successfully, then another pop-up window will prompt for Duo. Note that the Duo window looks very similar to the password window, but the text in the box will describe the options you can use for Duo (for example, a push, enter a passcode, etc).

4.2 Using Secure Copy (scp)

Use Secure Copy (**scp**) to transfer files between the teaching cluster and your local machine.

Sample commands to transfer files between the teaching cluster and your local Linux/Mac OSX machine using scp:

To copy a file from your local machine to the teaching cluster, type the following command at a terminal prompt on your local machine:

```
scp filename yourMyID@txfer.gacrc.uga.edu:
```

You will then be prompted for your MyID password and then for ArchPass Duo authentication.

To copy a file from the teaching cluster to your local machine, type the following command at a terminal prompt on your local machine:

```
scp yourMyID@txfer.gacrc.uga.edu:filename .
```

Please note the trailing period (.) in the line above is part of the command.

In the examples above *filename* should be replaced by the name of the file you want to transfer.

For more information on how to transfer files, please see https://wiki.gacrc.uga.edu/wiki/Transferring_Files

5 How do I open X-Windows applications from the teaching cluster on my local machine?

A number of software programs installed on the teaching cluster have X-Window (GUI) front ends. Examples of such applications are some text editors and some graphing programs.

In order to display such X-Window applications on your Windows desktop, your desktop needs to have an X-Window client (or server) running on it. A free X-Window server for Microsoft Windows (10/8/7) is Xming. You can download it from <http://sourceforge.net/projects/xming> and make a default installation. You will need to install the Xming server and the Xming-fonts package. During the installation of Xming, you might want to select the option to create a desktop icon for Xming. When the installation of these two packages is complete, double click on the Xming icon to start the X-Window server (a capital X will appear on your task bar). Now you need to configure your SSH program (such as PuTTY) to allow tunneling of X11 connections. Once that is done, you can SSH into your teaching cluster account and run X-Window applications. The application should appear on your local Windows desktop. Each time you logout and log back into your Windows desktop, you will need to start the Xming Server manually before using PuTTY to connect to your teaching cluster account.

If you connect to the teaching cluster using a Mac OSX machine, make sure the machine has XQuartz installed, then open an XQuartz terminal (not a regular terminal). From the XQuartz terminal prompt, connect to the teaching cluster using

```
ssh -X yourMyID@teach.gacrc.uga.edu
```

where *yourMyID* is your UGA MyID. If you connect to the teaching cluster using a Unix/Linux machine, use the -X option with ssh (see line above) and your teaching cluster X-Window applications should open on your local machine.

For more information, please see https://wiki.gacrc.uga.edu/wiki/Frequently_Asked_Questions

6 Basic Unix/Linux commands:

Here are some of the common Linux commands that you can use on the teaching cluster:

- **ls**: lists all the files in the current directory
- **du -h**: checks how much space each file and subdirectory is using and the total space used by the current directory
- **mkdir *subdir***: makes a new subdirectory called *subdir*
- **cd *subdir***: changes to the subdirectory *subdir*
- **rm *filename***: deletes the file called *filename*
- **cp *filename* *subdir***: copies the file called *filename* to a directory called *subdir*
- **cp *filea* *fileb***: makes a copy of *filea* and call it *fileb*
- **mv *foo* *moo***: renames a file or directory called *foo* to *moo*
- **rmdir *subdir***: deletes an empty directory named *subdir*
- **rm -f -r *subdir***: deletes the directory *subdir* and all its content (files and subdirectories).

A series of short and very easy to follow videos covering the basics of Linux are available in Kaltura (login with your UGA MyID and password) at <https://kaltura.uga.edu/channel/GACRC/176125031>. **Highly recommended.**

7 Creating/editing text, including source code of programs:

If you create files on your Windows/Mac desktop (or laptop) and then copy them to the teaching cluster, please type the following commands on the teaching cluster, to ensure that these files can be interpreted correctly by the Linux OS:

dos2unix *filename* (for files created/edited on Windows)
mac2unix *filename* (for files created/edited on Macs)

Several text editors are available on the teaching cluster, so you can create/edit files directly on the teaching cluster. A user-friendly text editor, available on the teaching cluster is **nano**. To create or edit a file called *program.f*, type the following command at a teaching cluster terminal prompt

nano *program.f*

Other text editors installed on the teaching cluster are **emacs** and **vim**. If interested, you can open several text editors, copy and paste some text into them and try to manipulate the text to see which editor you like best. You can google them to get information on how to use these programs.

8 Plotting figures

You can use these programs on the teaching cluster to create plots:

- **xmgrace**: plotting program with a user-friendly graphical front-end. You can google for “xmgrace tutorial” to find simple tutorials to help you learn how to use it. To use it, first load its module with the command

module load Grace

Then you can start it by typing

xmgrace &

- **gnuplot**: command line plotting program. You can google “gnuplot tutorial” to find a tutorial on how to use gnuplot. To run gnuplot on the teaching cluster, first load its module with the command

module load gnuplot

Then you can start it by typing

gnuplot

Both **xmgrace** and **gnuplot** are available for other platforms, including Mac OSX and Windows, and you might have access to these programs on your local machines as well.

Note: If you plan to use these plotting software interactively with the graphical front-end on the teaching cluster, please first type the command **xqlogin** to get a prompt on an interactive node, and then start these plotting programs.

9 Compiling a Fortran77 (simple, legacy flavor of Fortran) code

Sample Fortran77 code: Let us assume you have a Fortran77 program called *mult.f* with the following content:

```

C      Program mult
      Multiplies two integer numbers
      implicit none
      integer i,j,iprod
      i=3
      j=4
      open(1, file='output.txt')
      iprod=i*j
      write(1,10)i,j,iprod
10     format('The product of ', I2, ' and ', I2, ' is ', I3)
      close(1)
      end

```

This sample program creates a file called *output.txt* and writes the product of the integer numbers 3 and 4 in the file. If the file already exists, it will be overwritten.

To compile this Fortran77 code (called *mult.f*), first start an interactive session with the command:

```
qlogin
```

then load the GCC compiler module by typing the following command at the terminal prompt on the interactive node:

```
module load GCC/8.3.0
```

and then type the following command at the teaching cluster terminal prompt, in the directory where *mult.f* is located:

```
gfortran -O mult.f -o mult.x
```

This command will create an executable named *mult.x* in the same directory. The **-O** option instructs the compiler to generate an executable that includes very basic optimizations.

On the teaching cluster, type **man gfortran** at a terminal prompt (after you typed **module load GCC/8.3.0**) to get the documentation for this compiler.

10 Compiling a C code

Sample C code: Let us assume you have a C program called *mult.c* with the following content:

```

/* Program mult
Multiple two integer numbers */
#include <stdio.h>

int main(void)
{
    int i=3, j=4, iprod;
    FILE *fp;
    fp = fopen("output.txt","w");
    iprod=i*j;
    fprintf(fp, "The product of %d and %d is %d\n", i,j,iprod);
    fclose(fp);
    return 0;
}

```

This sample program creates a file called *output.txt* and writes the product of the integer numbers 3 and 4 in the file. If the file already exists, it will be overwritten.

To compile this C code (called *mult.c*), first start an interactive session with the command:

qlogin

then load the GCC compiler module by typing the following command at the terminal prompt on the interactive node:

module load GCC/8.3.0

and then type the following command at the teaching cluster terminal prompt, in the directory where *mult.c* is located:

gcc -O mult.c -o mult.x

This command will create an executable named *mult.x* in the same directory. The -O option instructs the compiler to generate an executable that includes very basic optimizations.

On the teaching cluster, type **man gcc** at a terminal prompt (after you typed **module load GCC/8.3.0**) to get the documentation for this compiler.

For more information, please see https://wiki.gacrc.uga.edu/wiki/Code_Compilation_on_the_teaching_cluster.

11 Running the executable

Whether the original program was written in Fortran or C language, the procedure to run a compiled executable (e.g. *mult.x*) is the same. Please note that you should **not** run jobs (executables) on the teaching cluster login node (where you login to); instead, you should submit them to the batch queue.

Steps to run a job on the batch queue

1. Create a job submission script file

Create a script file (called e.g. *sub.sh*) for job submission, with the following sample content

```
#!/bin/bash
#SBATCH --partition=fsr8602
#SBATCH --ntasks=1
#SBATCH --time=00:10:00
#SBATCH --mem=2gb
cd ${HOME}/project2
module load GCC/8.3.0
time ./mult.x
```

The contents of this job submission script are:

- Line 1: indicates the shell that will be used to interpret this script file.
- Line 2: indicates the name of the partition that you will be using (fsr8602).
- Line 3: indicates the number of computing cores your job needs (1 core).
- Line 4: indicates the walltime limit for this job (10 minutes).
- Line 5: indicates the memory needed for this job (2GB).

- Line 6: changes the directory to your working directory. In this example it is a subdirectory called *project2* in your home directory (which is referenced as `${HOME}`).
- Line 7: loads the GCC/8.3.0 module used to compile the code.
- Line 8: instructs the computer to run an executable (called *mult.x* in the example), which is located in the directory specified in line 6, and to record the amount of time used by the program.

Lines 1-5 are called header lines and they can be left unchanged for work you do in this course, unless your program needs to use more than 2GB of memory. On line 6 you should change the name of the subdirectory appropriately, on line 7 you should load the same module that you used to compile the code, and on line 8 you should change the name of the program appropriately, but it has to be preceded by `./` (dot-slash) as in the example here.

The partition that you will use in this course is **fsr8602** and the maximum runtime of each job is 10 minutes.

You can use other header lines in your job submission script, for example, to specify a name for the job. For more information, please see https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_the_teaching_cluster

2. Submit the job to the queue

To submit the job to the queue, type the following command at a teaching cluster terminal prompt:

```
sbatch sub.sh
```

If your job is correctly submitted to the queue with this command, you will get the job id number printed on your terminal. The standard output of your job will be stored in a file called *slurm-jobid.out* and the standard error of your job will be written to a file called *slurm-jobid.out*.

3. How to check the status of a job

To check the status of your pending or running jobs, use the command

```
squeue
```

To see more detailed information about a pending or running job, use the command

```
scontrol show job jobid
```

where *jobid* is the ID number assigned to your job when it is submitted and it is the number that appears in the first column of the **squeue** output.

4. How to cancel a job

To cancel a job use the command

```
scancel jobid
```

5. How to get information about a job that finished

To obtain a report with information such as the wall-clock time, the cpu time, etc for a job that finished, use the command

```
sacct-gacrc
```

You can save the output of the **sacct-gacrc** command in a file (e.g. called *summary.log*) with the command

```
sacct-gacrc > summary.log
```

6. Where can I get more information on how to run jobs

For more information, please see https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_the_teaching_cluster.

For more information on how to monitor jobs, please see

https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_the_teaching_cluster

12 Using the interactive node

To use an interactive node, type the command **interact** at a teaching cluster terminal prompt. This will return a terminal prompt on an interactive node, which can be used to test your programs. You can run your interactive applications from that terminal prompt. If you want to use a graphical front-end for a program (e.g. to make plots on the screen), type the command **interact -x11** at a teaching cluster terminal prompt. You can type **interact -h** to see all the options that this command accepts.

13 How to get help

You can request support, ask questions, and report problems with the teaching cluster by pointing your browser at

<https://uga.teamdynamix.com/TDClient/Requests/ServiceDet?ID=25844>

then please click on Request Service to access the online support form.

When you request support, please let us know details of the problem you are experiencing or the question you have.

14 What do I need to do before the online in-class training session

A training session on how to use the teaching cluster will be provided via Zoom, during the class period on Jan. 11, 2023. Before this class, please check that you have an SSH program and the VPN client installed on your computer, as described in Sec. 3 above.

If you are not familiar with the Linux operating system, please watch the Intro to Linux Playlist available at GACRC's Kaltura channel: https://kaltura.uga.edu/playlist/dedicated/176125031/1_uwkicelj/