A Guide to the GT Instructional Cluster

Jeff Epstein

Contents

1 Introduction

1.1 What is the GT Instructional Cluster?

The GT Instructional Cluster Environment (ICE) is a computing resource for providing a first-hand scientific computing experience, including HPC and GPU programming.

In this class, we'll use the hardware and software resources of ICE to write, compile, run, and debug CUDA programs.

1.2 Use of VPN

In order to access ICE from an off-campus location, you must first be connected to the Georgia Tech VPN. The method for setting up the VPN on your computer depends on your operating system; follow the appropriate instructions.

If you are connecting to ICE from a computer on Georgia Tech's campus, including connected to Georgia Tech's wifi, you do not need to use the VPN.

1.3 Logging in via SSH

1.3.1 Password authentication

One way to connect to ICE is with the `ssh` (secure shell) command, which is available on all modern operating systems. If `ssh` is not available on your computer, you can install an alternative, such as PuTTY. Before using `ssh`, ensure that you are connected either to the on-campus network or via the VPN.

To use `ssh`, first open the Terminal program that ships with your operating system. You will be presented with a command prompt similar to the following:

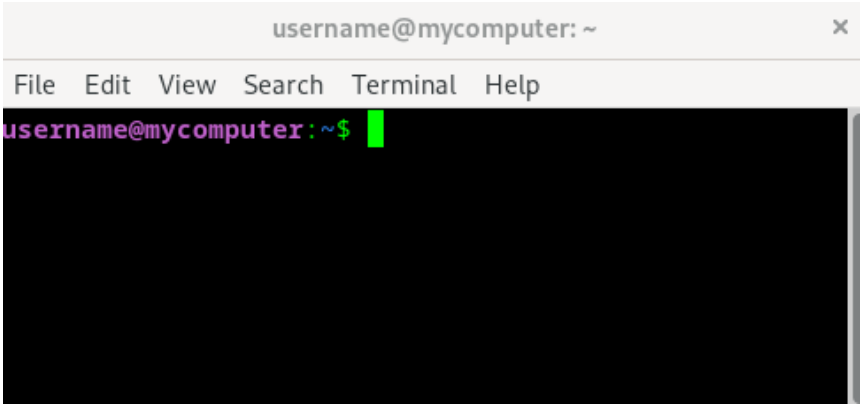

Figure 1: initial view of Terminal

Note that in the above image, the command prompt is in the form `username@mycomputer`, where *mycomputer* is the name of your computer, and *username* is your username on your computer.

From the command prompt, enter the following command, substituting `gtusername` with your GT username:

```
ssh gtusername@login-ice.pace.gatech.edu
```

At this point, you will be prompted to enter your GT password, as shown in the following image. If the password prompt does not appear, make sure that you are either connected to the campus network or connected via the VPN. Note that as you type your password, the characters will not echo to the screen.

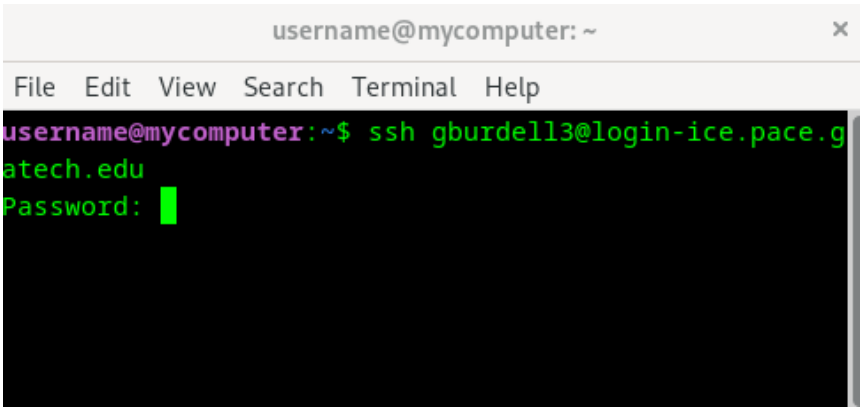

Figure 2: password entry

After you correctly type your password and press enter, you should be connected to the ICE login node. Note that the command prompt has changed to reflect that you've connected to the ICE node with your GT username, as shown in the below image. Commands issued here will be run on the ICE login node.

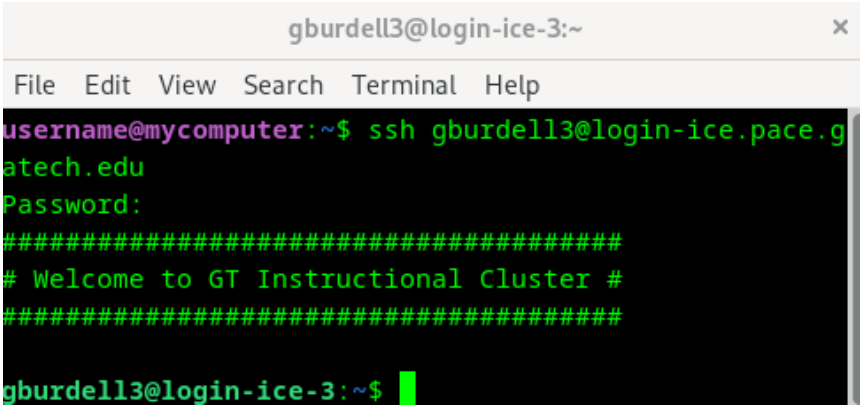

Figure 3: after connecting to ICE login node

From the login node, type `exit` to logout and close the connection.

The rest of this guide assumes that you are able to access the ICE login node and issue commands there.

1.3.2 Key authentication

A more secure and convenient way to log in to ICE is with key authentication. Under this approach, instead of using a password to authenticate your identity, `ssh` will use *key* stored in a file on your computer. The key will be automatically transmitted to the server when you connect.

In order to use key authentication, you must first generate a *key pair*. Use the following command from you own computer:

```
ssh-keygen -t ed25519 -C "gatech-ice" -f ~/.ssh/gatech-ice -P ""
```

After completing, the above command will create two files on your computer, in your `.ssh` directory: `gatech-ice`, which contains your private key, and `gatech-ice.pub`, which contains your public key. The private key is equivalent to your password, and should be kept secret: the content should never leave your computer. The public key must be installed on the computer that should accept your login credentials: in this case, on ICE. In your home directory on ICE, there's a file named `authorized_keys` keys that contains all public keys that are allowed to log in to your account, and it is into this file that we must insert the content of your public key file.

To install the public key on ICE, first open the public key file in your text editor and copy the contents into the clipboard. Log in to ICE using your password, and modify the file `authorized_keys` located in your the `.ssh` directory in your home directory. Paste the content copied from your public key onto a new line in that file and save the changes. Do not change any other lines in the file. If all goes well, you should be able to log out from ICE and log back in without a password.

1.4 Logging in via OnDemand

Another way to connect to ICE is through a remote desktop running in your web browser, termed OnDemand. With this approach, you have access to a full Linux desktop GUI, not just the command line. Before using OnDemand, ensure that you are connected either to the on-campus network or via the VPN.

To use OnDemand, first go to the OnDemand web site in your web browser. From the Interactive Apps menu, select Interactive Desktop.

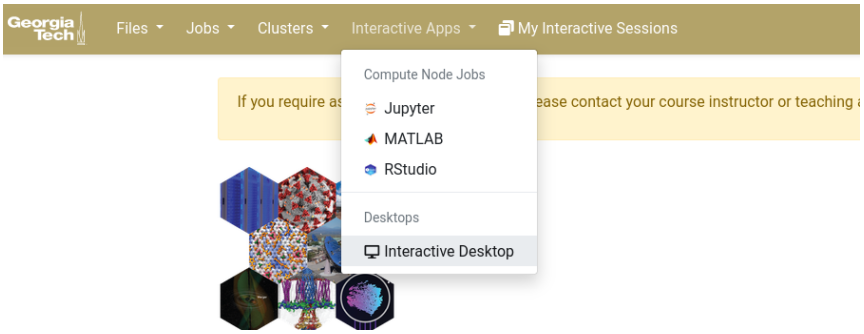

Figure 4: Interactive Desktop menu item

Configure your Interactive Desktop as follows. Be sure to select the maximum duration of your reservation. The Node Type should be set to give you the first available Nvidia GPU. Then click Launch.

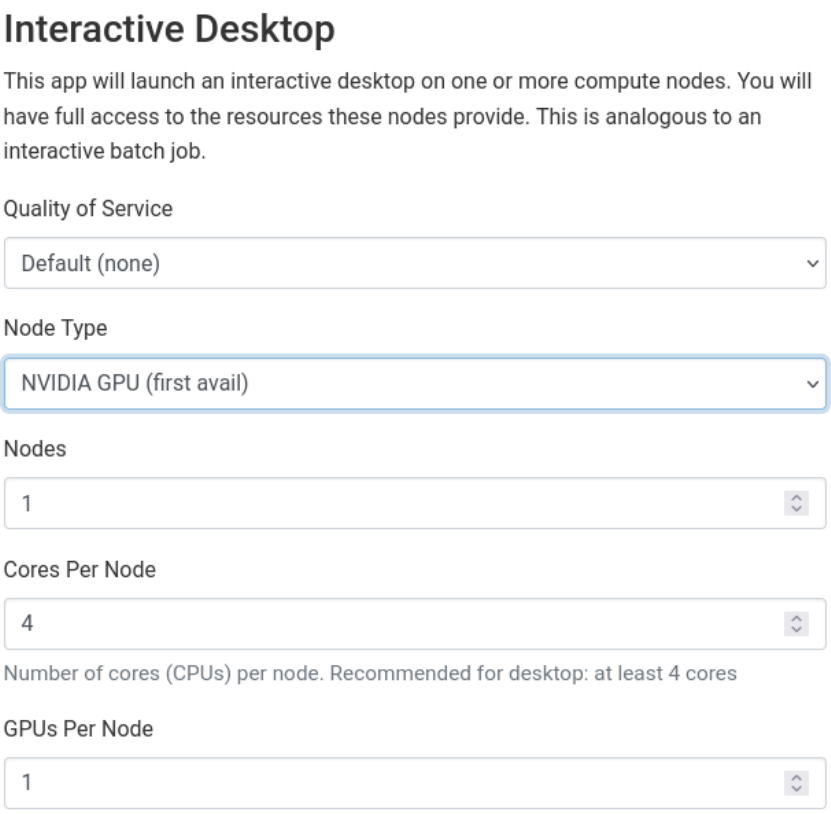

Figure 5: Interactive Desktop configuration

Your request will be queued. You may need to wait a few minutes. When your environment is ready, you'll be prompted to Launch Interactive Desktop.

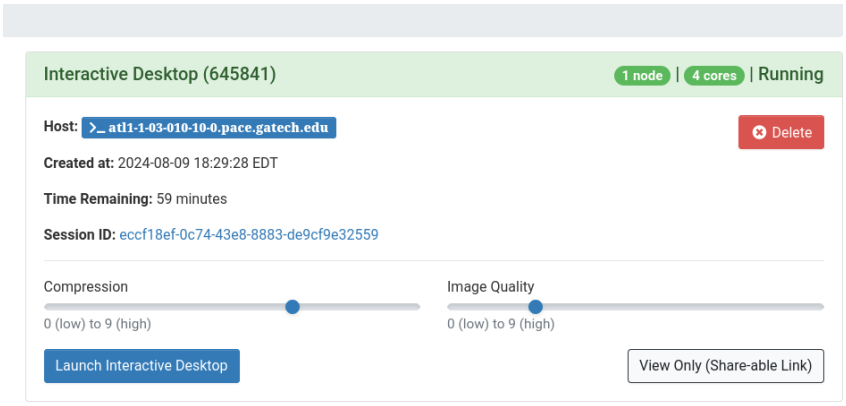

Figure 6: Launch Interactive Desktop

You'll be presented with a Linux desktop.

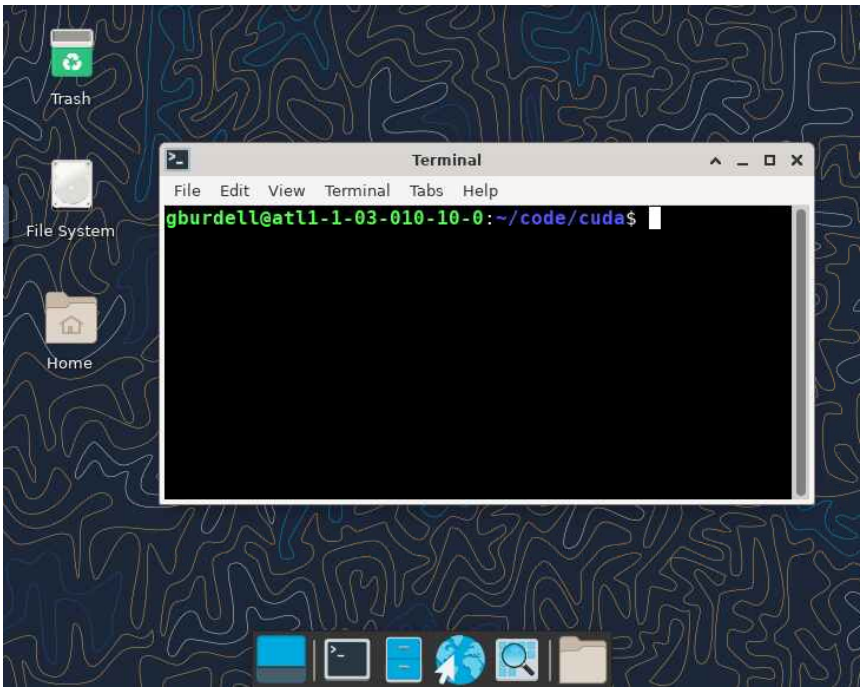

Figure 7: Linux desktop

Once you've logged in, click the Terminal icon to open the local terminal. From here, you can run commands. It's not necessary to use `salloc`, as your GPU is already allocated as part of the environment.

When you've finished using the desktop, please release your system resources by clicking the Delete button on the Interactive Desktop launch screen, or select Logout from the desktop. Changes made to your files in the environment will not be lost.

2 Using the cluster

2.1 Transfering files

The usual way to transfer files to or from the ICE cluster is the `scp` (secure copy) command. In general, the syntax of the command is as follows:

```
scp source destination
```

In the above command, *source* represents the file to be copied, and *destination* represents the place you want to copy it to. The source and the destination can include a username, server, directory, and filename. For example:

- `filename.cpp` refers to a file named `filename.cpp` in the current directory of the computer on which the command is run.
- `ece2000/hw1/filename.cpp` refers to a file named `filename.cpp` in the `ece2000/hw1` directory of the computer on which the command is run.
- `server:filename.cpp` refers to a file named `filename.cpp` on a server named `server`.
- `server:ece2000/hw1/filename.cpp` refers to a file named `filename.cpp` in the `ece2000/hw1` directory on a server named `server`.
- `username@server:filename.cpp` refers to a file named `filename.cpp` on a server named `server`, which we log in to with as `username`.

If `.` (period) is used as the destination, it indicates the file should be saved to current directory. Note that if a source or destination does not contain a colon, it is presumed to refer to a local file.

Transferring files from your computer to the cluster To transfer the file *filename* to the ICE cluster, use this command:

```
scp filename gtusername@login-ice.pace.gatech.edu:
```

Don't forget the colon at the end of the command (*not* preceded by a space). The command will save the file in your ICE home directory, overwriting any file with the same name already present in that location. If you want to save the file to a different location on ICE, give the path or filename after the final colon.

Note that the above command must be run on *your* computer. You won't be able to transfer files from your computer by running `scp` on an ICE node.

Transferring files from the cluster to your computer To transfer the file *filename* from the ICE cluster to your computer, use this command:

```
scp gtusername@login-ice.pace.gatech.edu:filename .
```

Don't forget the period at the end of the command (preceded by a space). The command will save the file in the current directory, overwriting any file with the same name already present in that

location. If you want to save the file to a different location, replace the period with the directory or filename.

Note that the above command must be run on *your* computer. You won't be able to transfer files to your computer by running `scp` on an ICE node.

Transferring multiple files You can use wildcards to copy several files at once. For example, the following command will copy all C++ source code files in the current directory to your ICE `proj1` subdirectory, if it exists.

```
scp *.cpp gtusername@login-ice.pace.gatech.edu:proj1
```

You can use the optional `-r` flag to perform a *recursive* copy, which will include the named directory, as well as all directories and files contained therein. For example, the following command will copy the `src` direcory and all files inside it.

```
scp -r src gtusername@login-ice.pace.gatech.edu:
```

Mount as local Depending on your operating system, you may be able to *mount* your ICE home directory on your own computer. This will allow you to use your favorite IDE or text editor, running on your own computer, to access files stored on ICE.

1. If you're using a recent GNOME-based distribution of Linux, you can mount your ICE home directory by navigating to "Other Locations" in the Nautilus file manager and entering `sftp://login-ice.pace.gatech.edu/` in the "Connect to Server" field. You will be prompted to enter your username and password. Your home directory will then be accessible as a mounted drive on the Nautilus locations panel.

   Alternatively, you can type the command `gio mount sftp://login-ice.pace.gatech.edu/` from the command the line.

2. If you're not using a GNOME-based distribution, you can use `sshfs`. First, install `sshfs` through your package manager; this will probably be a command like:

   ```
   sudo apt-get install sshfs
   ```

   Then, create a mountpoint and use `sshfs` to mount your ICE directory to that mountpoint:

   ```
   mkdir my-ice-dir
   sshfs gburdell3@login-ice.pace.gatech.edu: my-ice-dir
   ```

   To unmount, run:

   ```
   sudo umount my-ice-dir
   ```

2.2 Editing files

Editing files on the cluster You can edit text files and source code using editors already installed on the cluster.

The text editor `nano` is probably the easiest option. You can start `nano` by running the following command from the ICE login node or a computer node:

```
nano filename
```

You should see a screen similar to the following:

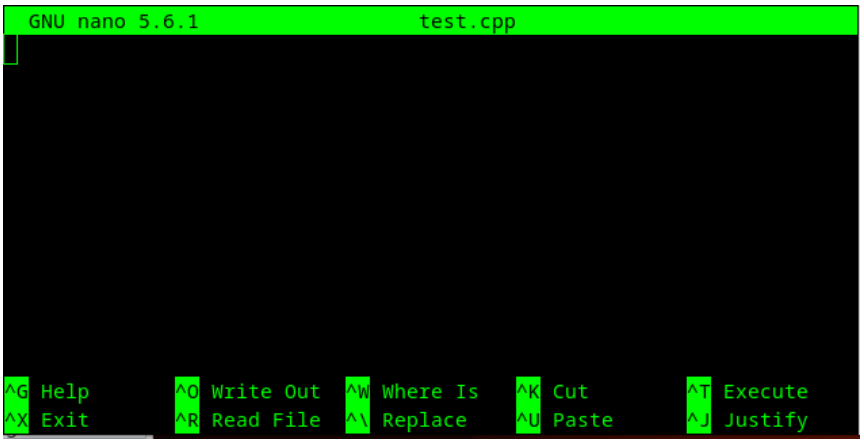

Figure 8: nano

You can use the arrow keys to navigate around the file. Unfortunately, `nano` does not support the use of the mouse. The menu at the bottom of the screen shows other `nano` commands, where the caret character indicates the Ctrl key. For example, you can type Ctrl-X to quit `nano`, Ctrl-O to save the file, Ctrl-K to cut the current line, etc.

Besides `nano`, the cluster also has other popular editors, such as `vi` and `emacs`. You can find tutorials for these tools online.

Editing on your computer Alternatively, you can edit files on your own computer with your editor or IDE of choice. However, you will need to transfer the files to the cluster in order to run and execute them. Therefore, you might use the following procedure:

1. Edit file on your own computer.
2. Save changes.
3. Transfer file to cluster.
4. Compile, run.
5. Repeat.

See the above instructions for info on how to transfer files.

2.3 Compiling CUDA programs

The Nvidia CUDA Toolkit is required to compile and run CUDA programs. It's installed on ICE, but its availability must be explicitly request each time you log in. For this, ICE uses the SLURM workload manager, whose use is described briefly below.

To make the Toolkit accessible use either this command:

```
module load cuda/12.1.1
```

or this:

```
module load nvhpc/24.5
```

After the CUDA Toolit has been made accessible with the `module` command, we can compile our program with the CUDA compiler. The usual command to compile CUDA programs is `nvcc`, which serves as a near drop-in replacement to the familiar `g++` or `gcc` commands. The syntax for invoking the command is mostly identical:

```
nvcc -o outputfile inputfile
```

CUDA source files typically have a `.cu` suffix. To compile the source code in `example.cu` into an executable, use the following command.

```
nvcc -o example example.cu
```

The above command, if successful, will produce an executable file named `example` in the current directory, which you can run by typing `./example`. If running your CUDA program fails with a "no device" error or similar, make sure that a GPU is accessible, as described in the next section.

If you get a `command not found` error when running `nvcc`, make sure you've first run the `module` command given above.

2.4 Running CUDA programs

In order to run a CUDA program, a GPU must be accessible. Because GPUs are a limited resource, they must be requested before use. If you requested a GPU node from OnDemand, then your GPU is already allocated, and no further steps are necessary. If you're connecting to ICE through `ssh`, you need to explicitly request a GPU with a command similar to the following:

```
salloc --gpus=1
```

After running the above command, there may be a pause while the system waits for a GPU to become available. At that point you'll be logged in to a compute node, and then you'll be able to run CUDA programs. You don't need a GPU in order to merely compile CUDA programs.

If you need X11 forwarding for running graphical programs with a GPU, add the `--x11` flag to the above command line.

From a compute node, you can run a CUDA program by typing in its path. For example, to run the program `foobar` in the current directory, just type `./foobar` and press enter.

When you no longer need the GPU, you can type `exit` to release your reservation, leave the compute node, and return to the login node.

The above `salloc` command requests allocation of a GPU until you relinquish it by logging out of the compute node. Alternatively, if you want to request a GPU for running just one command, you can use the following form:

```
srun --gpus=1 --pty your-command-here
```

Please note that the load of the ICE cluster may vary. At busy times, particularly at the end of the semester or before due dates, there may be significant wait for an available GPU. Delays in allocation are your responsibility, and extensions will not be granted due to cluster availability issues. Please plan ahead.