

# How To Use Hampshire's Supercomputer

Andrew Zito

## 1 What is the Supercomputer?

Hampshire's supercomputer (actually called a "high performance cluster") is a bunch of computers all interconnected. Each one of these computers is more powerful than your laptop (in different ways – some have a lot of active memory, or RAM, some have really high processor speeds, etc.) These computers are called "blades" or "nodes."

The blades are managed by a software called "Tractor." Tractor's job is to take a "job," or piece of code to be run by the cluster, decide which blade to run it on, and send it to that blade. For example, if Lee runs an intensive evolutionary algorithm program, it will get sent to a relatively powerful blade. On the other hand, if a student runs a fairly minor task they need for some class, it will get sent to a much less powerful blade.

At the conceptual center of the cluster is the "head node." The head node is like the main lobby or headquarters of the cluster. It's the first place you end up when you access the cluster. All the other blades are tied to the head node, and it is from here that Tractor sends out jobs.

## 2 Basic Terminal Commands

These are some super basic commands you *really* should know before using the cluster. They only work when input from a Linux or OSX terminal. If you use Windows, don't freak out – the Cluster uses Linux, and once you are inside the cluster these commands will work (we'll get to how to actually get inside later).

- `ls`  
Shows the files in your current location.
- `cat [filename]`  
Shows the contents of a (typically text) file.
- `cd [folder]`  
Changes your location to the specified folder (folders are also called directories).

- `pwd`  
Shows your current location.
- `vi [filename]`  
Uses VIM, a text editor, to edit a file. If `[filename]` does not exist, VIM will create it.

VIM Commands (to be used from inside a VIM session, after you input `vi [filename]`):

- `I`  
Pressing this key will enter “insert” mode, in which you can edit files.
- `ESC`  
Pressing `ESC` will *exit* insert mode and allow you to type further commands. See below.
- `:wq`  
Exits and saves.
- `:q!`  
Exits and discards changes.

### 3 Step One: Access the Cluster

To access the cluster, the very first thing you must do is email Lee and/or Josiah Erikson. Josiah will email you back with a username and password which you can use to access the cluster.

Once you have your username, you will use SSH to actually get into the cluster. SSH is basically just a way to remotely access a computer. All you’re doing when you SSH into the cluster is opening a connection to the head node.

To do this, you first need to open a terminal or command prompt. On Windows, you can type **Windows key + R**, and in the window that opens type `cmd`. On Mac, go to your **Applications** folder and find **Terminal**. If you are on Linux, you should already know how to do this (but if you don’t, try **Ctrl + Alt + T**).

In your terminal or command prompt window, type:

```
ssh [username]@fly.hampshire.edu
```

and hit **Enter**. Make sure to replace `[username]` with the username Josiah sent you (it should just be the preface of your Hampshire email).

*Very important note: if you are using Windows, and you get an error like: ‘ssh’ is not recognized as an.... then you probably don’t have SSH software installed. Scroll to the bottom of this section for further instructions.*

You should be prompted for a password. You won’t be able to see what you are typing – don’t panic! That’s for security, just like how most website logins turn your password into dots. Your typing is still working, so go ahead and input your password and hit **Enter**. You should then see something like this:

```
Last login: Wed Dec 9 14:49:50 2015
gromit131-50.hampshire.edu
Rocks 6.1.1 (Sand Boa)
Profile built 17:09 14-May-2014
```

```
Kickstarted 13:49 14-May-2014
```

```
-----
[username@fly ~]$
```

Congratulations! You are now inside the cluster’s head node. Anything you input from now on will be sent as a command to the head node.

*Hello confused Windows users! If you aren’t a confused Windows user lacking an SSH client (if you successfully accessed the cluster using the instructions above) then skip this and move on to Step Two!*

So, why did your computer say ‘ssh’ is not recognized...? Because to use SSH you need to have an SSH “client” installed. This is basically just a piece of software. Unfortunately, unlike Linux and Mac, Windows does not come with a built-in SSH client.

To obtain an SSH client, go here (<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>), scroll down, and download “A Windows Installer for everything except PuTTYtel”. Run the installer and follow its instructions. You have now installed PuTTY, the most popular SSH client for Windows, but you’re not quite done.

Run PuTTY. A window called “PuTTY Configuration” will open. Near the top of the window should be a text box titled “Host Name (or IP address)”. In this box, type `fly.hampshire.edu`. You *do not* need to change *anything* else. Now click **Open**.

An alert will pop up, saying a bunch of things. It’s just telling you that this is the first time you’ve connected to this; click **Yes**. A command prompt window will open, with the message `login as:` Type in the username Josiah gave you. You will then be prompted for a password, so type in your password and press enter. You won’t see anything as you are typing; this is just a security feature, don’t worry!

You should see something like:

```
Last login: Wed Dec 9 14:49:50 2015
gromit131-50.hampshire.edu
Rocks 6.1.1 (Sand Boa)
Profile built 17:09 14-May-2014

Kickstarted 13:49 14-May-2014
```

```
-----
[username@fly ~]$
```

Congratulations, you have successfully accessed the cluster!

## 4 Step Two: Uploading Your Files

On the cluster, your username has its own storage space. In file path terms it is `/home/username`. Now, presumably, whatever code you want to run is stored in one or more files. In order to copy these files into your cluster storage space, so that the cluster can access them and run your code, there are two things you can do. The first and “right” way is to use SSH file transfer:

*Very important note: Meet me at the bottom of the section again, Windows users. If the `ssh` command didn't work for you, neither will this. Also, if the instructions at the bottom of the section don't work, try the second method described below.*

1. Open a terminal/command prompt window. (Do not use the one that is SSHed into the cluster).
2. Type in `scp [file path] [username]@fly.hampshire.edu:/home/[username]` and hit **Enter**. Replace `[username]` with your username and `[file path]` with the path of your file. A file path is just where the file is located on your computer, so on Linux/OSX it might look like `/home/username/my.file` or on Windows it might look like `C:\Users\Username\my.file`. If you are having trouble with this, try using the `cd` command to move into the folder where your file is, and then you can use *just* the filename.
3. That's it. Repeat for multiple files.

If, for whatever reason, this is proving troublesome (especially if you are on Windows and the instructions at the bottom of this section aren't working either), you can try the second method, which is a bit “hacky,” but works on any operating system and is in some ways, simpler.

1. Copy the code from your file.
2. Go to your terminal or command prompt that is SSHed into the cluster.
3. Type in `vi [filename]` and hit `Enter`, replacing `[filename]` with whatever you want to call your file (include an extension if you want one, like `vi filename.py`).
4. Your terminal or command prompt should go blank-ish, with some blue squiggles on the side and few things at the bottom. Ignore this. All you need to do is press `I`.
5. Right click (on Windows, you may need to right click on the bar at the top of the window rather than directly in the command prompt) and paste in your code (on Windows, try `Edit > Paste` in the right click menu).
6. Press `ESC`, and then type `:wq`.
7. Hit `Enter`. Ta-da! Your code is on the cluster, in a file called whatever you input for `[filename]`.

Repeat this method for multiple files.

*Hello again confused Windows users! And again, if you are not in this category, move on to Step Three.*

We're going to have to use PuTTY to do this.

1. Open a command prompt window (don't use the one that is already SSHed into the cluster).
2. Try typing *just* `pscp`. If you get `'pscp' is not recognized...` then move directly to the next step. If you get something else much longer, then skip to step 4.
3. Use `cd` to navigate to your PuTTY install folder. By default this is `C:\Program Files (x86)\PuTTY` (so you would type `cd C:\Program Files (x86)\PuTTY`).
4. Type `pscp [file path] [username]@fly.hampshire.edu:/home/[username]`. Replace `[username]` with your username and `[file path]` with the path of your file. The file path is just the location of your file on your computer, so something like `C:\Users\Username\my.file`. If you are having trouble with file paths, try using the `cd` command to move into the folder where your file is, and then you can *just* use the file name (but *don't* do this if you followed step 3).
5. That's it. Repeat for multiple files.

## 5 Step Three: Run Your Code

Almost there! By the way, if you want to see the files you uploaded in the last section, you can use the `ls` command (in your SSHed terminal or command prompt window).

To run your code, we're going to pass a terminal command to Tractor. You *could* send Tractor something as simple as `ls` (although that would be rather a waste of a supercomputer).

First, copy-paste this command into your SSHed terminal/command prompt (DON'T hit enter): `/opt/pixar/tractor-blade-1.7.2/python/bin/python2.6 /opt/pixar/tractor-blade-1.7.2/tractor --engine=fly:8000 -c`

Now, the next part is going to vary according to what you are trying to do. Let's say you are trying to run a python file. In that case, you will append to the commands above:

```
/opt/python/bin/python [mypythonfile]
```

Overall, it will look like this:

```
/opt/pixar/tractor-blade-1.7.2/python/bin/python2.6
/opt/pixar/tractor-blade-1.7.2/tractor-spool.py --engine=fly:8000 -c
/opt/python/bin/python [mypythonfile]
```

If you are trying to do something else, you will have to change the part you append. What you need to know is that everything after `-c` is a terminal command, just like `ls` or `cd`. So, if you wanted to, say, run C++ code, you would need to look up the terminal commands for that. The example command above simply tells the computer to run `[mypythonfile]` using python.

## 6 Step Four: After You Run It

Presumably your code has some sort of output. It is possible to access this output from the cluster's monitoring site (which we'll get to in just a second), but a better way is to save your output to a file, either at the end of the program or periodically as it runs. How to do this will depend on the language you are coding in. The file you write to will appear in the same place as all your other files on the cluster (`/home/[username]`), the first place you end up when you SSH in. You can view the data in this file using the `cat` command, or by downloading it to your computer and opening it in your preferred text editor.

To download it, we're going to use `scp` again (or `pscp` if you're on Windows with PuTTY, and remember to `cd` into the PuTTY install folder if you need to). Use the same command we used to earlier to send files to the cluster, and simply reverse the order of the files. So, from

a terminal or command line *not* SSHed into the cluster, you would input something like `scp [username]@fly.hampshire.edu:/home/[username]/myoutputfile [file path]`. Replace `[file path]` with wherever you want to put the file on your computer.

If your output is small, or you just don't feel like writing to a file, you can output it to the console. For example, in Python you could use `print()`. However, this output won't actually appear in your window. You will need to go to the cluster's monitoring site ( <http://fly.hampshire.edu:8000/tractor/tv/>, and input your username (it doesn't need a password).

Apart from viewing console output, this site is somewhere you will want to visit to check up on your job. It tells you if your job is still running, when it was started, how long it has been running, and many other stats that you might need to know. To see console output, and also to see errors (if your job does produce an error) click on the job in the main window. Over on the right side you should see a pane that takes up the whole vertical length of the site, and it should have a little box in it. If your job is running, the box will be green, if it's done, the box will be gray, and if it errored, the box will be red. Double click the box, and it will show you, in the main window, console output from your job (which will include errors, if they happened).