

Running HPL on Sooner

Note: Things that you should type are in the **computer boldface** font.

1. Before doing this, you **MUST** have already installed HPL.

2. Go into your NCSI2010 directory:

```
cd ~/NCSI2010
```

3. Check to make sure that you're in your NCSI2010 directory:

```
pwd
```

4. Copy the directory named HPL_exercise from Henry's NCSI2010 directory:

```
cp -r ~hneeman/NCSI2010/HPL_exercise .
```

5. You should now have your own copy of the HPL_exercise directory, as a subdirectory of your NCSI2010 directory. Check to make sure that you do:

```
ls
```

Note that this command is lower case L followed by lower case S (that is, "ell ess" which is short for "list"), **NOT** "one ess."

You should see a list of files and subdirectories, one of which should be:

```
HPL_exercise
```

6. Change directory into your HPL_exercise directory, like this:

```
cd HPL_exercise
```

7. Make sure that you're in your HPL_exercise directory, like this:

```
pwd
```

8. See what's in this directory, like this:

```
ls
```

You should see some subdirectories, such as HPL_0001p.

9. Go into the first such subdirectory:

```
cd HPL_0001p1t
```

10. Using your preferred Unix text editor (for example, nano, pico, vi, emacs), edit the batch script file hpl_0001p1t.bsub.

In particular:

(a) Change yourusername to your user name.

(b) Change youremailaddress@yourinstitution.edu to your e-mail address.

11. While you're editing the batch script file, carefully read its contents.

12. Also examine the file named HPL_0001p1t.dat, which contains the input parameters for this run.

13. Once you're done editing and reading, submit the `hpl_0001p1t.bsub` batch script, using the `bsub` command as described in the batch script file you just read.
14. Check the status of your batch job using the `bjobs` command, as described in the batch script file. You may need to check its status repeatedly.
15. You can start submitting other batch jobs (below) while waiting for this one to finish.
16. Once the batch job finishes (which you'll know because it no longer shows up when you do the `bjobs` command, and which may take quite a while), find out which files have been created most recently in your current working directory:

```
ls -ltr
```

This command is lower case L, lower case S, space, hyphen, lower case L, lower case T, lower case R (that is, “ell ess space hyphen ell tee are”), meaning “list with long listing (lots of information), with files listed ordered by time, with the most recent at the bottom”.

17. Examine the contents of the `stdout` and `stderr` files created by the batch job.
18. What percentage of the theoretical peak of the hardware you're running on did you achieve?
Hint: These chips are 2.0 GHz, and can perform up to 4 Floating point OPerations per second (FLOPs) per core.
HPL reports speeds in GFLOPs (“gigaflops,” meaning billions of floating point operations per second).

19. Go into your `HPL_0001p2t` directory:

```
cd ../HPL_0001p2t
```

20. Edit your `hpl_0001p2t.bsub` batch script file.
In this file, notice that we've changed the number of threads to 2 but kept the number of MPI processes at 1.
21. Also examine the file named `HPL_0001p2t.dat`, which contains the input parameters for this run.

How does this run's parameters differ from the previous? Why?

22. Submit this batch job using the `bsub` command.
23. Monitor its progress using the `bjobs` command.
24. When it completes, find its `stdout` and `stderr` files, and examine them.

How does this run differ from the previous run?

Is the output what you expected? **Why or why not?**

25. Do the same sequence of steps (#16 - #20) with `hpl_0002p1t.bsub`.

In this file, notice that we've changed the number of MPI processes to 2 but kept the number of threads per process at 1. So this is the opposite approach from the previous.

How does this run's parameters differ from the previous? **Why?**

26. Now, create new directories with names like

```
HPL_0001p4t  
hpl_0004p4t
```

and figure out how to change them to do the appropriate run.

Specifically, in the `HPL_whatever.dat` file, you'll want to change the following values: P, Q, Ns.

For the various values of N, we recommend doing the following:

a. Calculate:

```
sqrt(0.75 * nodes * 16 * 1024 * 1024 * 1024 / 8)
```

(Explanation: Each compute node on Sooner has 16 GB of RAM; each double precision value is 8 bytes; you want to use only part of the RAM, because the operating system needs part of it, and you really really don't want to use swap disk.)

b. Find the nearest multiple of $256 * 3 * 5$ that is just below the value calculated.

c. Then divide by 2, 4 and 8 to get the 4 N values needed.

Explain why this would be a good idea.

27. Why shouldn't you edit these files on your Windows PC, and then upload them to Sooner?

In principle you can, but in practice it's a **TERRIBLE IDEA**.

Here's why:

In Windows, almost all text editors embed hidden special characters (for formatting and so on) in the text file that you're editing, and also they express carriage returns differently than in Unix (including Linux).

Furthermore, in some of these files (especially batch script files and makefiles), **where** the carriage returns occur is **EXTREMELY IMPORTANT** (and in the case of makefiles, so too with tabs as opposed to spaces), and the text editors in Windows cannot be relied on to keep those in the proper places.

So, if you edit a file in Windows, there's no guarantee that, when you upload it to a Unix machine (including Linux), it'll be usable.

There's a command that can fix some (but not all) of these problems:

```
dos2unix filename
```

This is a very handy command, but it can't fix everything, so you take **HUGE RISKS** if you choose to edit a file in Windows and then try to use it in Unix.