Guide: How to use the NCSU HPC facility

1. Log into the machine: Use SSH to log into the machine with your UnityID and password. If you are using Windows, you may have PuTTy to log in. If you are using UNIX/Linux/Mac computers, you can log in by the command below:

```
ssh <UnityID>@login.hpc.ncsu.edu
```

2. Download the class code by executing: (or replace the URL with your own git repo)

```
git clone https://github.ncsu.edu/atmughra/ECE-506-406-Projects.git
```

3. Use a secure `scp` to transfer files to the hpc machine from your own computer. If you are using Windows, you may use the SFTP for ExpanDrive or WinSCP. For other systems, you can directly use the following command:

```
scp <src_path>/file
```

`<UnityID>@login.hpc.ncsu.edu:<dest_path>/file` 4. To compile your code with

OpenMP enabled use the following switch with gcc: `-fopenmp`

5. Run your program: The remote machine uses the Load Sharing Facility (LSF) to handle the program running requests. You can find useful commands on OIT's HPC website: http://www.ncsu.edu/itd/hpc/Documents/usefulLSF.php. Here are a summary of commands that are useful for this assignment:

a) Set OpenMP environment variable to indicate the number of threads:

```
setenv OMP_NUM_THREADS <number>
```

b) Submit your request:

```
bsub -W 2:00 -n 4 -q shared_memory -o std_out -e std_err "./program_name_here"
```

where

"-W 2:00" indicates that the running time limit of your program is 2 hours;
"-n 4"means that you request 4 processors to run your program;
"-q shared_memory" specifies the running queue;
"-o std_out" & "-e std_err" redirect the output and error streams to the files named as std_out and std_err.

c) Check the status of your running programs: (For short programs the program will likely already be done by the time you finish)

```
bjobs
```

If you need further assistance, please feel free to ask the TAs, or post on Piazza.