MPI

You will be using the **turing.cs.wit.edu** cluster for this lab. You already have logins for this cluster. Your username is your wit email without the @wit.edu. You can decide which ssh and scp client you would like to use – the one you used previously in Prof Carpenter's class is fine. Putty and WinScp also work well.

The `turing` cluster has 10 physical processors each with 4 threads for a total of 40 virtual processors. When you use MPI the np option on the command line specifies the number of **processes** – this value can be greater than 40.

The following links provides a quick reference to some useful resources, but you can use any other resources that you find on the internet:

- Vi: http://ryanstutorials.net/linuxtutorial/cheatsheetvi.php
- Vi: https://www.tutorialspoint.com/unix/unix-vi-editor.htm
- MPI: https://computing.llnl.gov/tutorials/mpi/#Exercise1
- Torque: https://kb.iu.edu/d/avgl
- Torque: https://csc.cnsi.ucsb.edu/docs/example-scripts-running-jobs
- Linux: https://www.pcsuggest.com/basic-linux-commands/
- Linux: https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners

Initial Test and Setup

Hello World: Test the hello world program done in class on the cluster. Create a file `hello-world.c` and enter the code found on the slides.

Compile:

```
mpicc -o hello_world hello-world.c
```

This creates an executable called hello_world.

Running from command line without batch queuing:

Execute:

```
mpirun –np 1 ./hello_world
```

Executes a single process (np is 1).

```
mpirun –np 4 ./hello_world
```

Executes 4 processes (np is 4).

The OS schedules the processes on the available processors. If available, it schedules one process per processor. In the case where there are other users, the OS can schedule the processes on fewer processors and interleave the execution. For example, if 8 processes are created (np = 8), but only 4 processing nodes are available, the OS will schedule 2 processes per processor.

Using Torque for batch queuing:
The Torque Script

helloScript	**comments**
#!/bin/bash	Use the bash shell
#PBS -S /bin/bash	
#PBS -o pbs_hello_out.dat	Call the output file pbs_hello_out.dat
#PBS -j oe	Join the output and error files
#PBS -l nodes=1:ppn=40	1 node with 40 processors per node
#PBS -M rawlinsm@wit.edu	Send emails to this address
#PBS -m be	Send an email when the job **b**egins and **e**nds
cd $PBS_O_WORKDIR	Change to the current directory
mpirun -np 40 -machinefile $PBS_NODEFILE ./hello_world	Run 40 processes

Type the helloScript (without the comments). Reminder: to ensure that your script is executable use the following command or equivalent:

```
chmod 777 helloScript
```

Submit the job to the Torque scheduler

```
qsub helloScript
```

Open `pbs_hello_out.dat` to view the results.

Refer to https://kb.iu.edu/d/avgl for more details.

If you request that your processes are run on 10 processors, for example, the scheduler will not start your job until 10 processors are available and your job will have exclusive use of all 10 processors. The trade-off is that your job may not start immediately and may have to wait.

Note that `turing` has been set up as a single node with 40 processors therefore you must use

```
#PBS -l nodes=1:ppn=<n>
```

Where n can vary from 1 to 40. Setting n greater than 40 will result in an error. The number of processes (np in the mpirun command) can exceed 40.