CS 5334 Spring 2016 Name _____
Shirley Moore, Instructor
In-class Activity April 7

MPI Programming

For this activity, we will use some of the example codes from the LLNL MPI Tutorial at https://computing.llnl.gov/tutorials/mpi/ . We will compile and run the codes on the Stampede supercomputer at TACC.

1. a. Compile the mpi_hello.c example and run it on two different nodes on Stampede. See the Stampede User Guide for information on how to compile the code and submit the batch job.

b. Copy mpi_hello.c to mpi_helloBsend.c. Edit your new **helloBsend** source file and modify it to do the following - after the master task has printed the number of tasks, but before MPI_Finalize. Compile and run as before.

- Have each task determine a unique partner task to send/receive with. One easy way to do this:
- if (taskid < numtasks/2) then partner = numtasks/2 + taskid
  else if (taskid >= numtasks/2) then partner = taskid - numtasks/2

- Each task sends its partner a single integer message: its taskid
- Each task receives from its partner a single integer message: the partner's taskid
- For confirmation, after the send/receive, each task prints something like "Task ## is partner with ##" where ## is the taskid of the task and its partner.

c. Copy your **helloBsend** source file to a new **helloNBsend** source file. Then convert the blocking routines to non-blocking routines. Compile and run as before.

2. Compile and run the mpi_wave example. Is the communication safe? If not, modify the code that it is safe.

3. a. Compile and run the mpi_heat2D example. Fix the communication so that it is safe if needed.

b. Analyze the communication requirements for the parallel heat2D algorithm for

1) the 1D data decomposition in the mpi_heat2D.c implementation

2) assuming a 2D data decomposition

c. Now consider a 3D heat problem. Analyze the communication requirements for 1D, 2D, and 3D data decompositions.