programming assignment 5 - MPI

due: Tues 4/19 (during class)

Ohio Supercomputer Center

MVAPICH2, an open source standard MPI implementation, is installed and available on the Oakley cluster. Use the *module list* command to verify it is loaded, and *module load mvapich2* if necessary. To allocate a proper node and begin an interactive computing session, use: **qsub -I -l walltime=0:59:00 -l nodes=1:ppn=12** (qsub -"capital eye" -"little ell" walltime . . . -"little ell" nodes . . .).

The qsub command usually takes from 5 to 30 seconds to execute. Note, however, that OSC is a shared resource, and your wait will very with current system load. We are guests on the OSC cluster, so please practice good citizenship. You can compile your MPI programs on the "login nodes." Only start a qsub batch **when you are ready to run your program.** Please "exit" from qsub once your test is complete. If you need to make more than very minor changes to your source, exit qsub, edit as required, and then start a new qsub batch session to re-test.

The MPI compiler is *mpicc*, which uses the same options as the gcc C compiler.

To execute MPI programs on Oakley first start an interactive qsub session, and then use *mpiexec -np 2 a.out*, where a.out is your compiled program name. The "-np 2" option will specify the 2 processes required for this assignment.

assignment 5

ping-pong: Determine the time taken and bandwidth achieved while performing repeated point-to-point communication transfers of buffers containing double-precision floating-point numbers between a pair of processors.

Structure the body of your program as follows:

```
  process 0
barrier
start timer
repeat iterations times
{
  send(buf, process1)
  receive(buf, process1)
  send(buf, process1)
  receive(buf, process1)
}
stop timer
time = totaltime/(4*iterations)
bandwidth = sizeof(double)*msg_size/time
```

```
  process 1
barrier
start timer
repeat iterations times
{
  send(buf, process0)
  receive(buf, process0)
  send(buf, process0)
  receive(buf, process0)
}
stop timer
time = totaltime/(4*iterations)
bandwidth = sizeof(double)*msg_size/time
```

Implement your ping-pong code using standard MPI blocking primitives, MPI_Send and MPI_Recv.

(cont'd. next page ...)

Note the run-time of each test and calculate the bandwidth achieved over 1,000,000 iterations for each of the following message sizes (number of doubles): 32, 256, 512, 1024 and 2048.

Your program should complete all tests in a single invocation, and should not require running the program multiple times. Use no command line parameters.

submitting results

- Submit your MPI source file following the general submission guidelines for the previous labs, with the following specifics:
  - name your program file <lastname>_<firstname>_lab5.c
  - provide a single make file called "makefile" that will name your executable "a.out".
- Submit your printed report during class on Tues. 4/19. Include in your report your name and section (9:35am or 12:45pm) along with your timing and bandwidth results and any observations you made.