Homework 7

Part A:

Write an MPI program where each process prints "Hello World", the processor each process is running on (use some form of the "hostname" command) , and the rank (pid) of the process. Run this on the ITaP Scholar queue on two nodes. Each nodes has 20 cores, so you should run a total of 40 processes.

What to turn in: You should turn in a zip file called <your last name>.zip. When unzipped it should create a directory called <your last name> containing your code and your output. Your output can either be a screen shot, what you capture from using the Unix/Linux *script* command or the program output directed into another file.

Information on how to run jobs using the scholar queue is available on the course web page.

Part B:

Write a plain old sequential C program that takes as input the number of processors (numP), an array size and prints the elements of the array of the given size that would appear on each processor p, $0 \leq p < numP$. For block, cyclic and block-cyclic for processor p your output should look like:

p: [l:u:s]

where l is the lowest element (array index) on the processor, u is the highest element (array index) on the processor and s is the distance between elements. Thus for processor p=0 and p=2 with a cyclic distribution on numP=7 processors of an array of size 50 the output for p would be

0: [0:49:7]
2: [2:44:7]

For block-cyclic, you can just print the elements on each processor.

What to turn in: You should turn in a zip file called <your last name>.zip. When unzipped it should create two sub-directories, *partA* and *partB*.

partA will contain your code and the output .o and .e files returned by qsub.
partB will your code and the output of running it with 8 processors, a chunk size of 5 for block-cyclic, and an array size of 200, and the output of running it with 7 processors, a chunk size of 3 for block-cyclic, and an array size of 101.