EECE5640
High Performance Computing
Homework 4
* Submit your work on Canvas in a single zip file.

1. (30) Develop a simple MPI program using C/C++ and OpenMPI that uses at least 2 nodes on Discovery and utilizes at least 16 processes on each node (a minimum of 32 processes in total). You are suggested to use the sample batch script provided on Canvas for specifying your OpenMPI configuration and running your program.
   a.) Start with an integer variable that you will pass to each process, where process 1 prints the value, increments the value by 1, and sends it to process 2. Process 2 prints the value, then increments the value by 1 and sends it to process 3. Repeat this for all 32 processes. When performing printing, print both the integer value, as well as identify which process is printing, and on which node this process is running on.
   b.) Next, continuing the printing, but once the value gets to 32, decrement the value, and continue to print out the value until the decremented value is zero.

2. (40) Develop a parallel histogramming program using C/C++ and OpenMPI. A histogram is used to summarize the distribution of data values in a data set. The most common form of histogramming splits the data range into equal-sized bins. For each bin, the number of data values in the data set that falls into that class are totaled. Your input to this program will be integers in the range 1-1,000,000 (use a random number generator that first generates the numbers first). Your input data set should contain 2 million integers. You will vary the number of bins. You should have as many OpenMPI processes as bins. You are suggested to use the sample batch script provided on Canvas for specifying your OpenMPI configuration and running your program. **Make sure to use the express partition, versus the short partition.**

   a.) Assume there are 100 bins. Perform binning across nodes and processes using OpenMPI, and then perform a reduction on the lead node, combining your partial results. Run this on 2 and 4 nodes on Discovery. Your program should print out the number of values that fall into each bin. Compare the performance between running this on 2 and 4 nodes. Comment on the differences.

   b.) For this part, assume you have 20 bins. Perform binning on each process using OpenMPI, and then perform a reduction on the lead node, combining your partial results. Run this on 2 and 4 nodes on Discovery. Your program should print out the number of values that fall into each bin. Compare the performance between running this on 2 and 4 nodes. Comment on the differences.

   c.) Compare the performance measured in parts a.) and b.). Try to explain why one is faster than the other and run additional experiments to support your claims.

3. (30) Let's revisit problem 2 from homework 3 (accelerating matmul.c). Now you have available a GPU to accelerate this code. Come up with an efficient CUDA implementation of the matmul code and compare it's performance to the performance you were able to get on a CPU. Discuss the strong scaling performance of your CUDA code (plot this). Also, run this and two different NVIDIA GPUs on the Discovery cluster and discuss the performance differences that you observe.

4. (Extra) Part of your weekly reading included a paper titled "MPI on Millions of Cores." Given that this paper was published in 2010 (12 years ago), can you comment on what changes have occurred since 2010 that could positively and/or negatively impact our ability to fully exploit parallelism on millions of cores? Many of the papers today discuss exascale computing. Select a recent paper on exascale-computing and compare/contrast the barriers identified in the two papers that impact our ability to achieve these milestones.

   This problem is worth 25 points of extra credit for the undergraduates and PlusOne students in the class and 15 points of extra credit for the graduate students.

* Written answers to the questions should be included in your homework 4 write-up in pdf format. You should include your C/C++ programs and the README file in the zip file submitted.