EECE5640
High Performance Computing
Homework 5
*Submit your work on Turnitin on Blackboard

1. (50) Develop a parallel histogramming program using C/C++ and MPI. 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 called classes. For each bin, the number of data values in the data set that fall into that class are totaled. Your input to this program will be integers in the range 1-1000 (use a random number generator). Your input data set should contain N integers, where N can be varied.

   Perform binning on each node using MPI, and then perform a reduction on node 0 to combine your partial results. Run this on 4 and 8 nodes on Discovery. Your program should print out the values that fell into each class, in ascending class order.

   a. For your binning, use each node to produce a histogram for (1/#classes)th of the input data values. Experiment with varying the number of classes and different values for N, and plot your performance results.
   b. For your binning, use each node to only produce values for a single class of the input data values. Experiment with varying the number of classes and different values for N, and plot your performance results.
   c. Compare the performance results for a and b, and attempt to explain any differences.

   *Answers to this question should be included in your homework write-up in pdf format, should include a C program submitted on Blackboard, and submitted through Turnitin.

2. (30) Develop a parallel CUDA program to implement part a.) in Problem 1. Run on a single GPU.
   a. Compare the performance of your CUDA implementation with your MPI implementation.

   *Answers to this question should be included in your homework write-up in pdf format, should include a C program submitted on Blackboard, and submitted through Turnitin.

3. (20) Read the Pascal whitepaper provided, and then identify the key features that were introduced in the Pascal P100 architecture, as compared against the Volta V100 architecture (make sure to identify the source for the information you obtained on the V100). Please do not just repeat what you read in the Pascal whitepaper, go into more detail on each of the features you identify.

   *Answers to this question should be included in your homework write-up in pdf format.

4. (30 points extra credit for both UG and Grad) Develop a parallel edge detector in CUDA (you can pick which edge detector you want to use). We provide you with 2-D gray-scale image to use as your input and you can use OpenCV to read the image (see help on using OpenCV). We provide you with code that can help you read the image. Save the output of your CUDA program in a file. The help file provides details on how to access the image on Discovery, and how to read the file. Include a picture of your input and output image in the pdf write-up. Evaluate the performance of your CUDA code using nvprof.

   *Answers to this question should be included in your homework write-up in pdf format, should include a C program submitted on Blackboard, and submitted through Turnitin.