ECE5640 High Performance Computing Spring 2016

Homework 5: Due Wednesday, March 23 at midnight

Programs should be run on the Discovery cluster. You should turn in a pdf report that answers the questions below, and one zip file that contains code and bash scripts for running your code.

1. (40 pts) Implement vector addition in CUDA. Your code should run on the discovery cluster and should add two vectors containing float values and store the result in a third vector. Use a random number generator to initialize your input vectors. Run the random number generator on the *host* and the vector addition on the GPU.

   (a) Run your code on vectors of size: 10^4, 10^6 and 10^8. Report your run times.

   (b) Run your code on vectors of size: 2^{12}, 2^{18}, and 2^{24}. Report your run times.

   (c) Comment on the relative efficiency of the different runs. Are powers of two or powers of ten more efficient? Other observations?

2. (60 pts) Implement a CUDA program that takes a vector of floats and returns the minimum value of that vector.

   (a) Run your code on vectors of size: 10^4, 10^6 and 10^8. Report your run times.

   (b) Run your code on vectors of size: 2^{12}, 2^{18}, and 2^{24}. Report your run times.

   (c) Comment on the relative efficiency of the different runs. Are powers of two or powers of ten more efficient? Other observations?

   (d) Compare finding the minimum to adding two vectors. Which is more efficient on the GPU and why? Assume initial vector values are initialized on the host and the result of your kernel computation is communicated back to the host.

   (e) Vector addition requires no communication between threads. Finding the minimum requires communication. Does this effect efficiency? Explain.