Michael Gibson

CS 462

11/22/2023

I tested my code by first giving it small inputs with the file input. This file contained a small assortment of numbers which it would find the mean of. Since the array was small, it was easy for me to check to see if the program was right. I then needed to find a way to get larger input files, so I created a small program that simply wrote out “1000000” and the a for loop starting at 1 and ending at 1000000, printing out each number into the input file. I then used that file to test my program as well. I compared this to a program that did not run cuda. Despite using multiple threads parallelizing, the cpu was still faster even up to 1 million.

I learned how to use cuda effectively by adding many elements together. I was tempted to use something like atomic, as it would be an easier process, but I figure that that would be no better than the cpu if not worse simply because each process would be waiting on a single process to finish. I decided that creating operations to decide pairs of values per thread would be the best idea. This can be seen in the comments for sum.

The mean proved more difficult, but once I figured it out, variance was a cake walk. I figured out what I should do initially with mean, knowing that I needed to take halves of the entire array and add parallel. Eventually I stumbled on using a scope to keep track of how far along I was with finishing the addition.