Lab03 Report

Meisi LI(ml6095)

The machine used cuda5.cims.nyu.edu

In my code, it is going to setup GPU and have a error handling to check the CUDA. Transfer the numbers from host to device and check the first number in the array with others. When it is smaller than it compared number, the first will change to the larger number. And then, transfer this array to host and the first number in this array the largest number in this array.

How to pick the block and grid sizes/dimensions

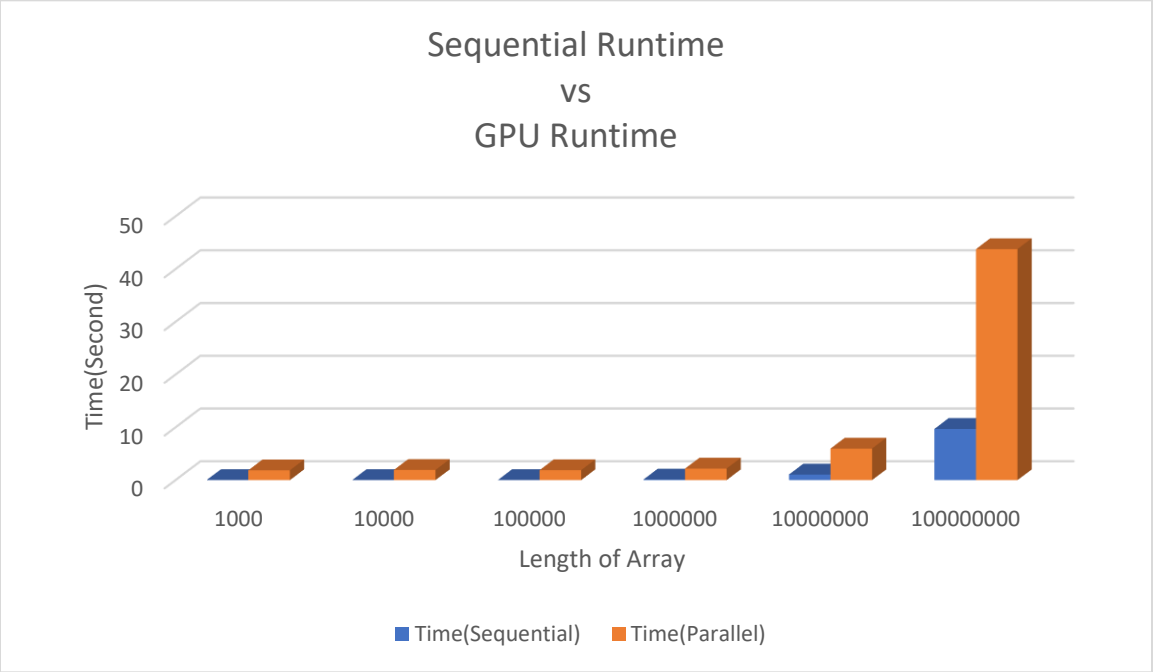
The CUDA compute servers on CIMS limits each block with 1,024 threads per block. Also, it is 32 multiple of the warp size.

The length of random array numbers is n. Thus, the block dimension are 32 or n. For the grid dimensions, it is ceil(n / 32);

Command line

nvcc -g -arch=compute_50 -code=sm_50 -o maxgpu maxgpu.cu

Graph



The behavior in the graph

From above runtime graph, the trend of both version are increasing. And GPU version costs more time to sequential version. I think it could be better. In my code, the time complexity is O(n) but it is slower than sequential version. When the data transfer from host to device and then transfer back to host, it might spend lots time and this is the essential reason of what happened in my code.

However, the speedup is increasing when we increase the length of numbers. From the graph, the GPU version has obvious speedup when the GPU handles the larger data. Because the performance improvement on computational power will replace the cost of data transfer.