Brian Laccone
CS 475
5/29/2019

Project 6

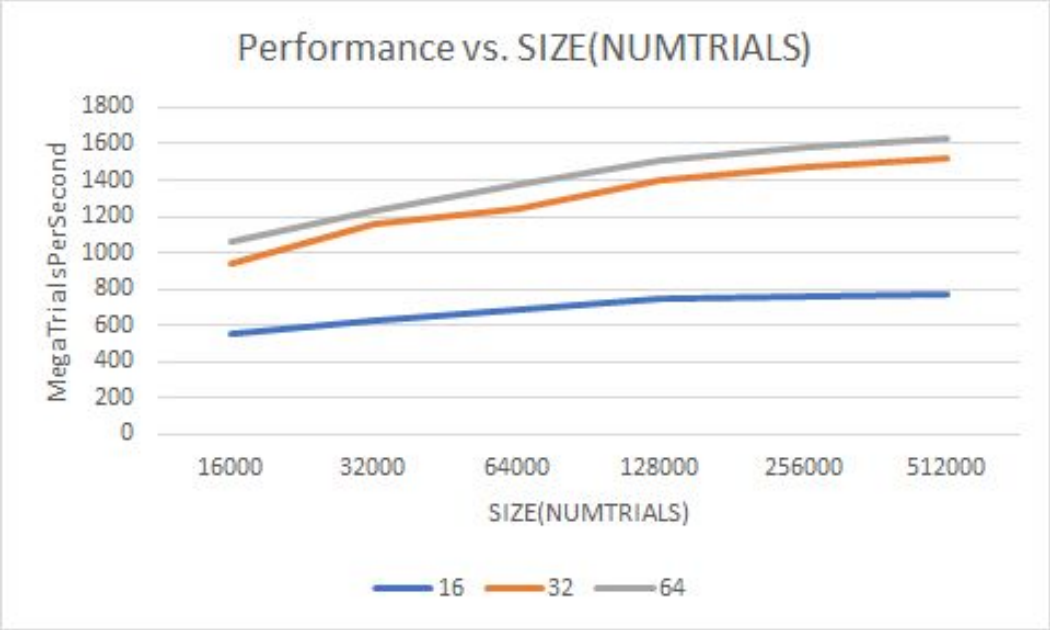
1. I ran this project on my own computer. I have a NVIDIA GeForce GTX 1080.
2. Show table and graphs:

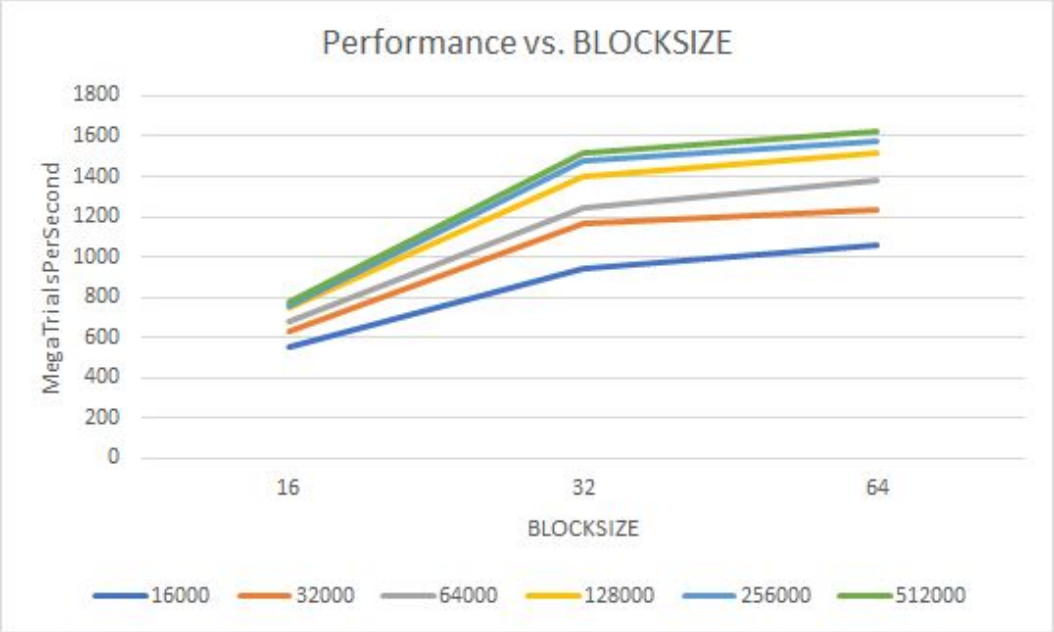
Table

(Probability that the beam hits the plate is ~42%)

	16	32	64
16000	553.86	938.28	1056.66
32000	627.95	1163.44	1232.66
64000	683.51	1249	1379.43
128000	743.47	1403.71	1513.34
256000	762	1475.45	1578.24
512000	773.85	1521	1624.48

Graphs


Performance vs. SIZE(NUMTRIALS)
MegaTrialsPerSecond
1800
1600
1400
1200
1000
800
600
400
200
0
16000
32000
64000
128000
256000
512000
SIZE(NUMTRIALS)
16
32
64


Performance vs. BLOCKSIZE
MegaTrialsPerSecond
1800
1600
1400
1200
1000
800
600
400
200
0
16
32
64
BLOCKSIZE
16000
32000
64000
128000
256000
512000

3. The pattern that I am observing from the graphs is that as the BLOCKSIZE and SIZE grew the performance grew as well. The blocksize of 16 was pretty far behind the blocksizes of 32 and 64. The increase from 32 blocksize to 64 blocksize was a very small increase but it was still an increase in performance. When comparing SIZE(NUMTRIALS) vs performance, the higher the size reached the higher the performance was. 16k was the lowest performance and 512k was the highest.

4. I believe that the pattern looks this way because as the array size increases more calculations are need to execute the program. This gives the GPU more opportunity to take advantage of it's multiple work groups and distribute work efficiently. It is faster to work inside of blocks instead of having to work across different blocks. I believe that the patterns of the graphs are very close to how I thought the outcome would be. I knew from last weeks project that my GPU's 32 and 64 blocksizes were going to be very close performance wise. I'm not sure why my 32 and 64 are very similar in performance. I think that if the SIZE was much larger then I may see a difference in 32 and 64 but with the sizes given for this project they were similar. I also had a feeling that 16 would be far behind the 32 and 64.

5. I believe that the BLOCKSIZE of 16 was much worse than the 32 and 64 because it is faster to compute within blocks than computing across different blocks. This means that having higher blocksizes, such as 32 and 64, increases the speed because computing done inside the blocks is faster instead of across the blocks as is the case with the blocksize of 16. The GPU is leaving performance on the table by having low blocksizes.

6. This means that GPU parallel computing is a field that is absolutely needed if a program needs faster run times. It is especially needed because you can't just stick in numbers that may seem correct and run with it. A lot of tests need to be done to determine which blocksizes are best for the program at hand. This does mean that programs can become faster with proper programming but it can be hard to determine which is the best blocksize compared to the array size being computed. Too small or too large of a blocksize can cause performance decreases. There is not a "one size fits all" when it comes to GPU computing and each program and each computer it is being run on will different performances but it is positive to realize that with proper GPU parallel programming, increased performances can be achieved.