Project #5 – CUDA: Monte Carlo Simulation

Young-Joon Park

parky8@oregonstate.edu

1. Tell what machine you ran this on

The simulation was run locally on a desktop PC with a Ryzen 5600 and an RTX 2080.

2. What do you think this new probability is?

Ntrials	Blocksize	MegaTrials/sec	% Probability
1024	8	34.2612	24.41
1024	32	33.8624	25.98
1024	64	22.7111	26.46
1024	128	61.5385	26.27
4096	8	229.3907	27.2
4096	32	239.2523	26.93
4096	64	225.3521	26.05
4096	128	225.7496	27.05
16384	8	795.031	26.88
16384	32	942.9098	26.93
16384	64	516.6499	26.88
16384	128	340.4255	26.89
65536	8	1134.626	26.88
65536	32	1020.429	26.71
65536	64	595.0029	26.91
65536	128	1148.626	27.01
262144	8	1462.335	26.93
262144	32	1710.587	26.94
262144	64	3880.625	26.78
262144	128	5204.574	26.89
1048576	8	2602.287	26.88
1048576	32	4717.535	26.86
1048576	64	4610.017	26.85
1048576	128	7949.539	26.91
2097152	8	2980.942	26.86
2097152	32	6594.486	26.88
2097152	64	6246.879	26.85
2097152	128	8790.878	26.87

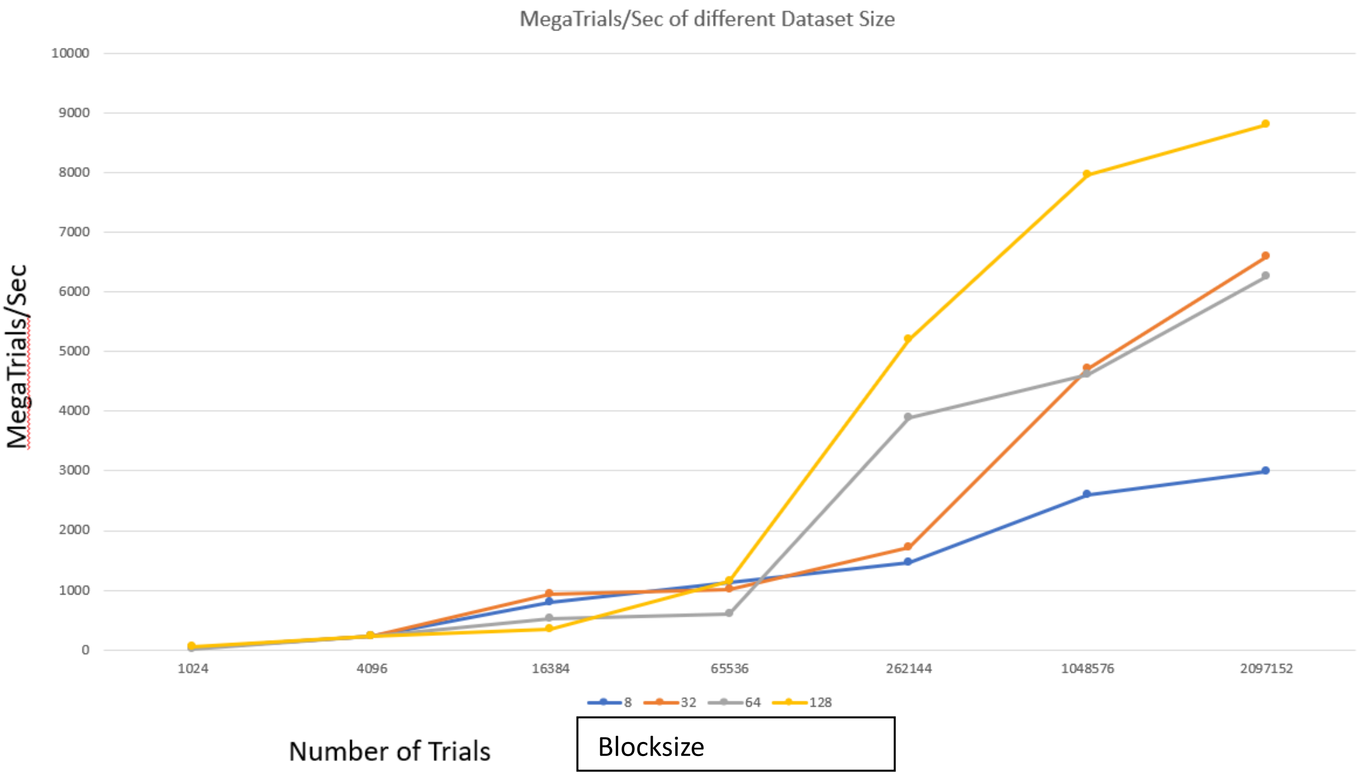
The probability is about 26.9 to 27%.

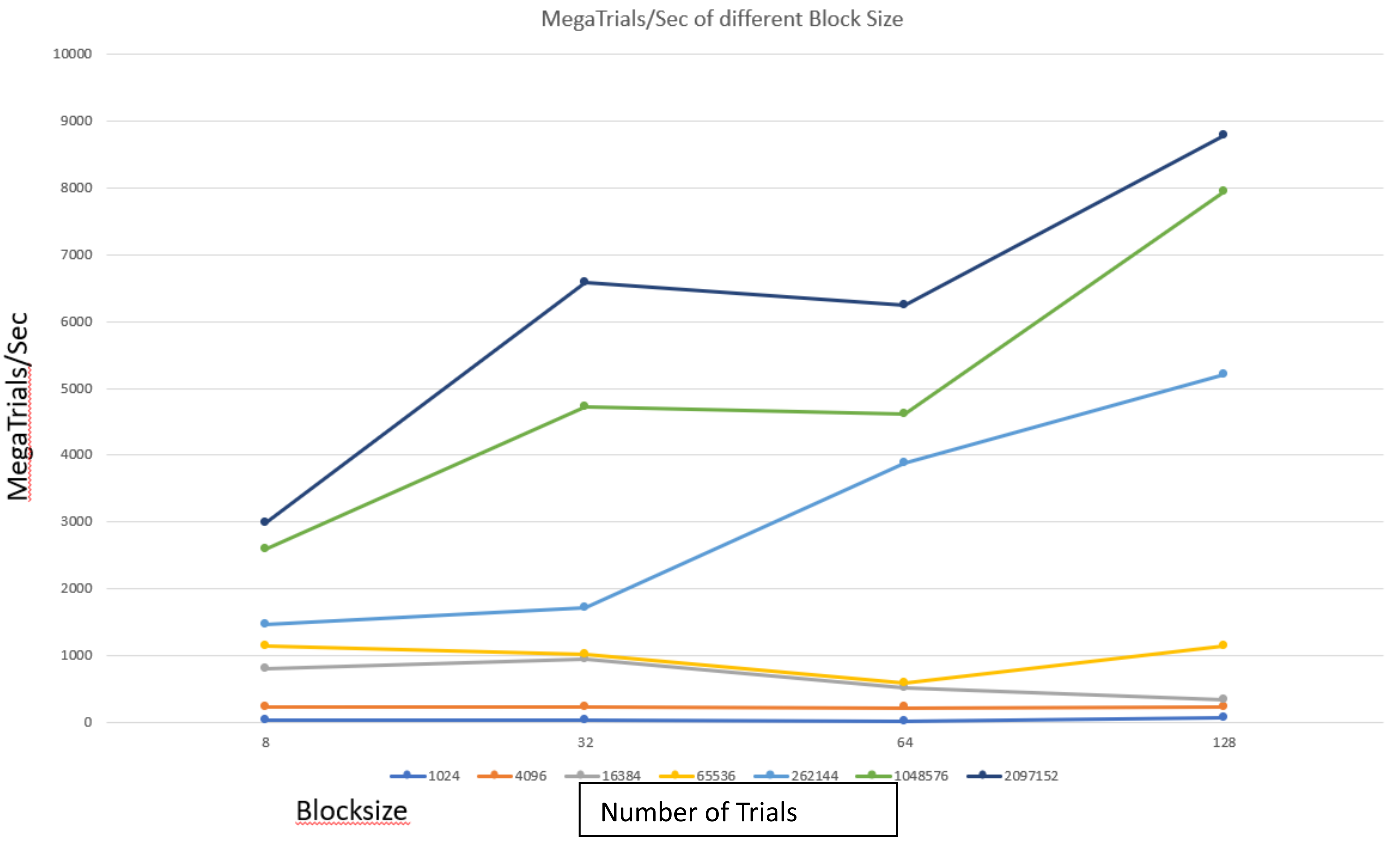
3. Show the table and the two graphs

Table:

	1024	4096	16384	65536	262144	1048576	2097152
8	34.2612	229.3907	795.031	1134.626	1462.3349	2602.2872	2980.9416
32	33.8624	239.2523	942.9098	1020.4285	1710.5868	4717.5354	6594.486
64	22.7111	225.3521	516.6499	595.0029	3880.6253	4610.0168	6246.8785
128	61.5385	225.7496	340.4255	1148.6259	5204.5744	7949.5387	8790.8784

Graphs:





4. What patterns are you seeing in the performance curves?

32 and 64 block sizes show similar performance (in relative terms, 64 is marginally better), while 128 show much better performance compared to 32 and 64. Block size 8 shows very bad performance compared to others.

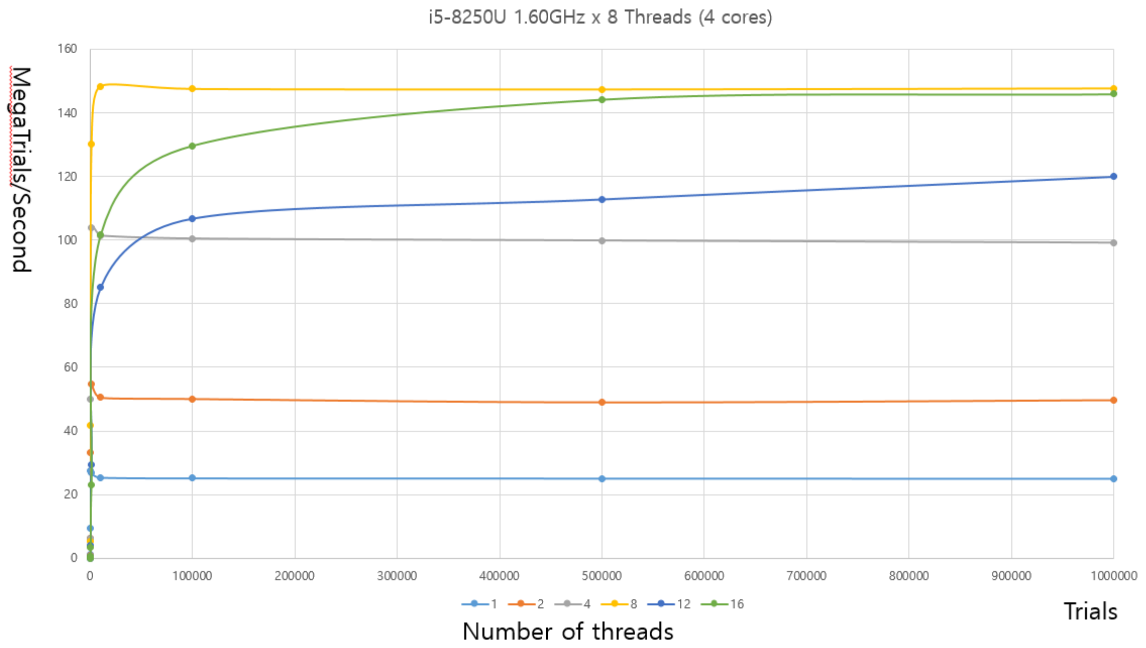
5. Why do you think the patterns look this way?

CUDA works based on increments of blocks of threads called warps. Each warp consists of 32 threads. This means a block size of 8 leads to wasted threads as 24 threads in a warp are idle while only 8 are working. For 32 and 64, there is not as much as a performance difference compared to 128, most likely because the overhead required to set up CUDA (setting up and making sure the registers of the threads in each block are filled up) overshadows the performance gains from utilizing only one or two warps, leading to small performance increase as opposed to 4 warps working which leads to ample time for CUDA to set things up as part of its overhead. This results in moderate performance gains in the jump from 1 to 2 warps, as compared to the big gain from 2 to 4 warps.

6. Why is a BLOCKSIZE of 8 so much worse than the others?

As stated above, CUDA performs in increments of warps and a warp consists of 32 threads. Only utilizing 8 threads of 32 threads in a warp leads to 75% of a warp being idle, resulting in the bad performance as shown in the experiment.

7. How do these performance results compare with what you got in Project #1? Why?



For reference, here is the graph from project 1. Granted that project 1 was performed on my much worse laptop, one can see that the number of MegaTrials/sec is much higher when using CUDA. This is because it harnesses the power of the specialized threads in the GPU, as opposed to solely relying on threads on the CPU.

8. What does this mean for what you can do with GPU parallel computing?

Under certain conditions in which one's program can utilize GPU parallel computing, using it can lead to extreme performance gains.