

Project #5

CUDA: Monte Carlo Simulation

PangFa Chou | choupa@oregonstate.edu

1. The simulation was run on a rabbit server.

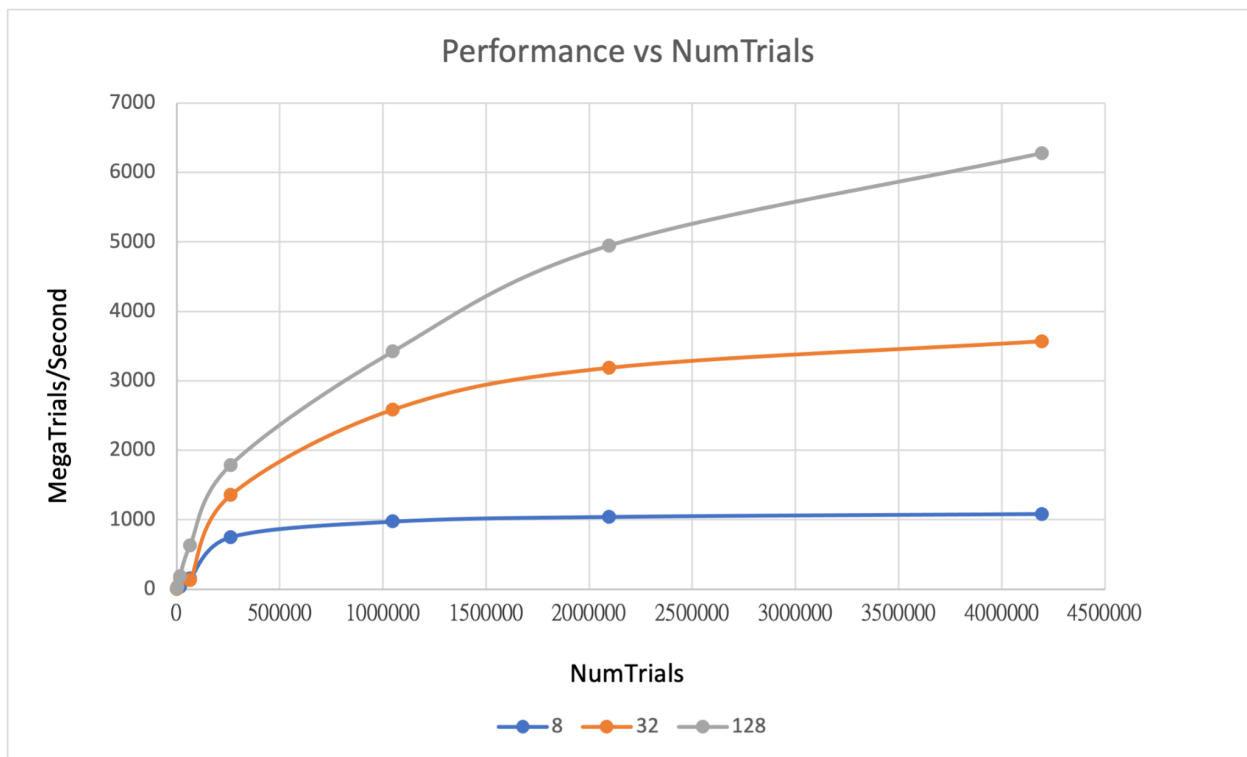
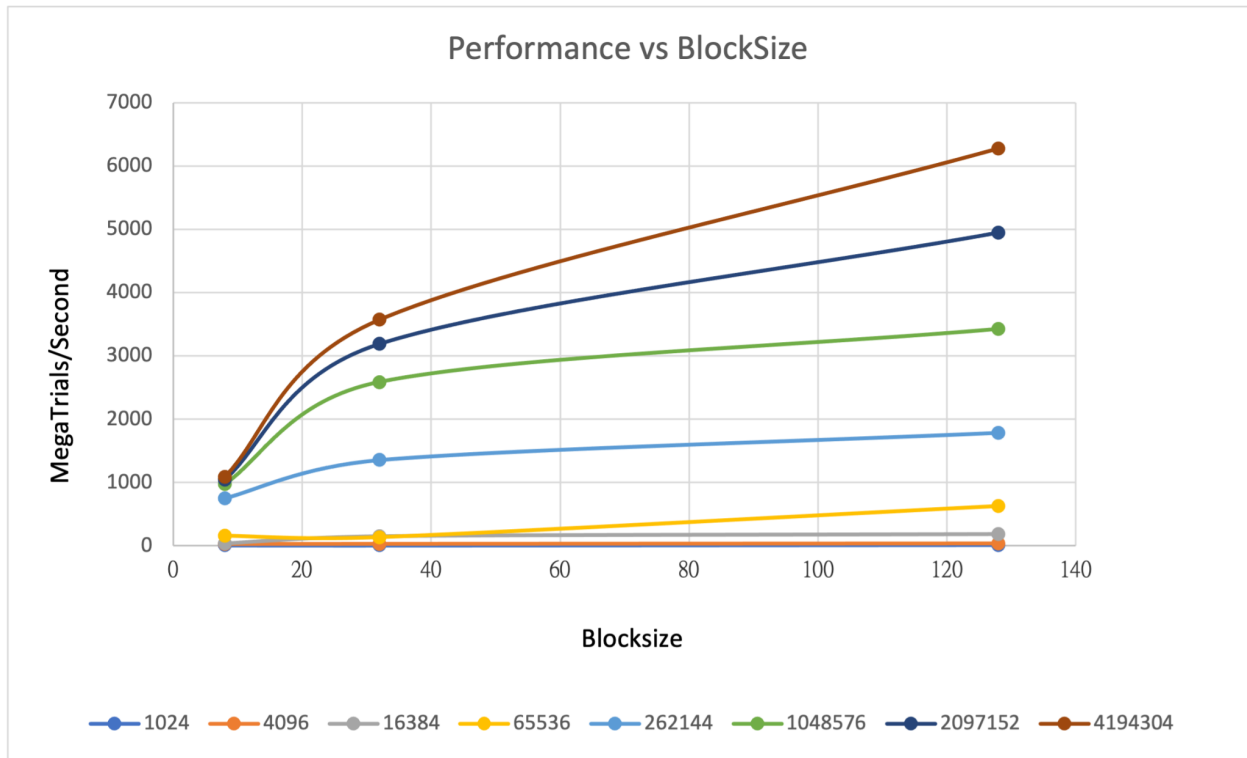
(GPU Device 0: "NVIDIA GeForce GTX TITAN Black" with compute capability 3.5)

2. The result of the program can be presented as following tables and graphs:

Table:

Number of Trials	BlockSize	MegaTrials/Second	Probability
1024	8	8.784	22.36%
1024	32	7.9661	22.66%
1024	128	12.0346	22.56%
4096	8	26.8569	21.66%
4096	32	29.405	22.80%
4096	128	37.5037	22.56%
16384	8	38.016	23.00%
16384	32	151.9288	22.43%
16384	128	187.0661	21.94%
65536	8	157.9029	22.64%
65536	32	135.0478	22.37%
65536	128	632.8801	22.48%
262144	8	751.6976	22.44%
262144	32	1356.0668	22.36%
262144	128	1788.2558	22.59%
1048576	8	975.4994	22.44%
1048576	32	2587.0835	22.57%
1048576	128	3423.3179	22.50%
2097152	8	1043.4174	22.47%
2097152	32	3189.7206	22.51%
2097152	128	4950.2228	22.51%
4194304	8	1085.2937	22.50%
4194304	32	3571.6389	22.47%
4194304	128	6281.0045	22.48%

Graphs:



3. We can see that the performance grows up really quick at first. Then, after they get bigger, the performance gradually approaches a certain value.
4. I think the reason why performance grows up so fast is because the more block size it has, its parallelism is better. Then, when the numtrials is larger, it starts to hit the peak performance so they gradually approach certain values.
5. It is because its block size is smaller. It exposes more parallelism when the blocksize is higher.
6. There is a difference between Project 1 and Project 5. The performance in project 1 grows up and drops at the beginning, and then grows up again. It is because of the false sharing problem, which the higher number of threads may cause worse performance at some point.
7. We should assign more block size in GPU parallel computing because it performs better.