

Project #5: CUDA: Monte Carlo Simulation

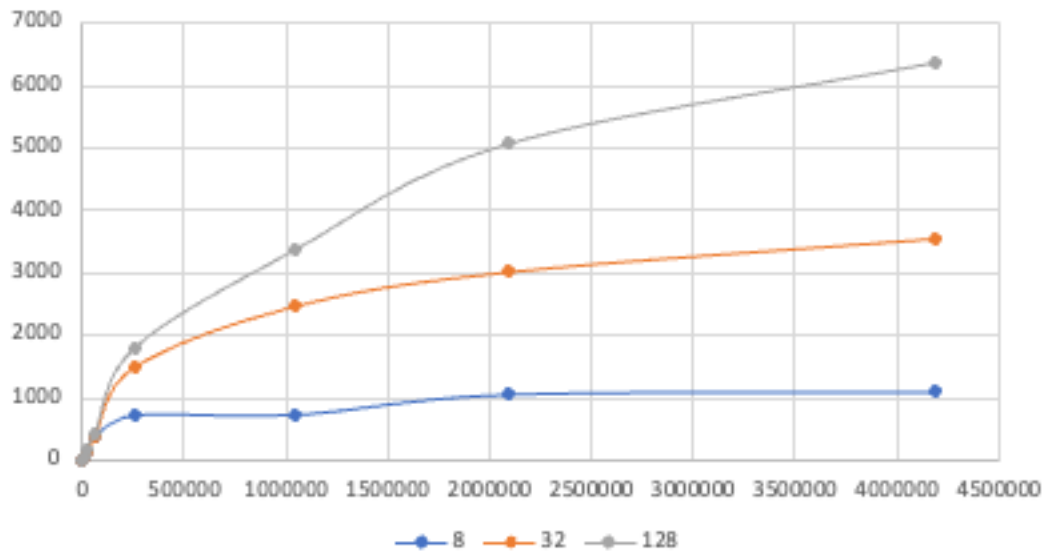
Danlin Song

songdan@oregonstate.edu

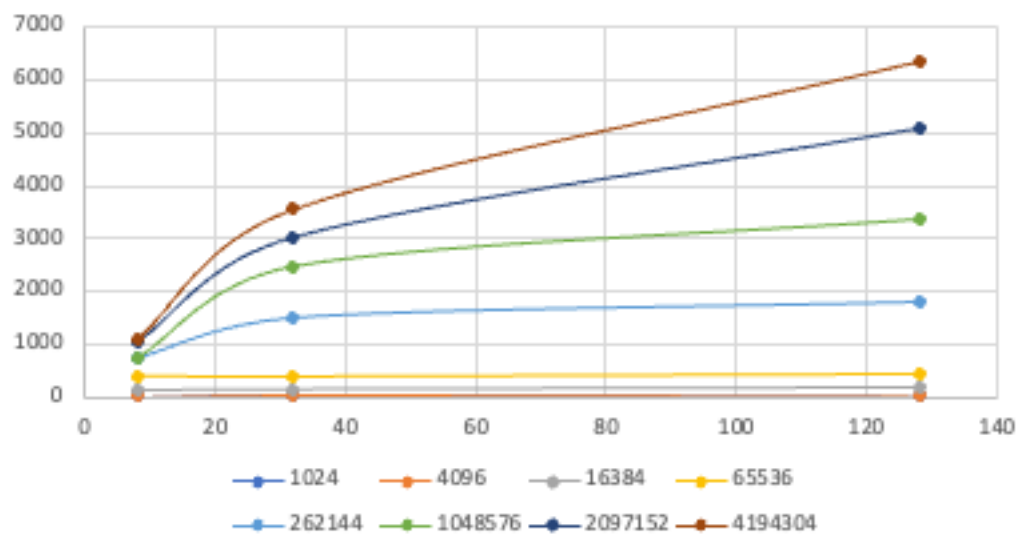
- Tell what machine you ran this on
I ran this project on OSU rabbit server.
GPU Device 0: "NVIDIA GeForce GTX TITAN Black"
- Show the table and the two graphs
Monte Carlo performance table

	1024	4096	16384	65536	262144	1048576	2097152	4194304
8	9.0575	38.4269	115.5495	401.2539	729.15	733.311	1055.3641	1095.4709
32	9.4927	44.0771	132.7801	393.9977	1507.8225	2477.1696	3026.3679	3562.1263
128	9.2113	40.6091	164.1026	443.5781	1812.7904	3373.9702	5070.0912	6368.283

Performance vs. NUMTRIALS



Performance vs. BLOCKSIZE



- What patterns are you seeing in the performance curves?
As both the block size and the number of trials increase, performance will improve, and this growth will be fast at first, then level off, and should eventually reach a steady state.

- Why do you think the patterns look this way?
GPUs are better at handling large data volumes because they can split the data into chunks and process them in parallel. data on the GPU is parallel, so performance increases as the size of the data set increases.
- Why is a BLOCKSIZE of 8 so much worse than the others?
When the block size is 8, it will cause the GPU to generate more blocks for parallel operation, and the performance will be reduced when the number of blocks increases compared to other large capacity blocks
- How do these performance results compare with what you got in Project #1? Why?
The GPU performs much better than project1 because the CPU has a much smaller core and cache than the GPU. GPU processes faster than cpu if the number is larger, but when the data is smaller, GPU is slower than CPU because GPU still needs to split the data into blocks for parallel processing, which takes a lot of unnecessary time.
- What does this mean for the proper use of GPU parallel computing?
GPUs can be used to perform large-scale data parallel operations, such as image rendering and graphics calculations.