CS 475 - Project 5: CUDA: Monte Carlo Simulation

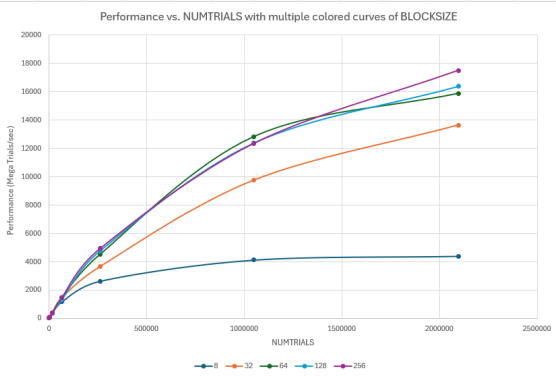
Name: Luke Scovel
Email: scovell@oregonstate.edu
Project Number: 5
Project Name: CUDA: Monte Carlo Simulation
Machine: `submit-b.hpc.engr.oregonstate.edu`

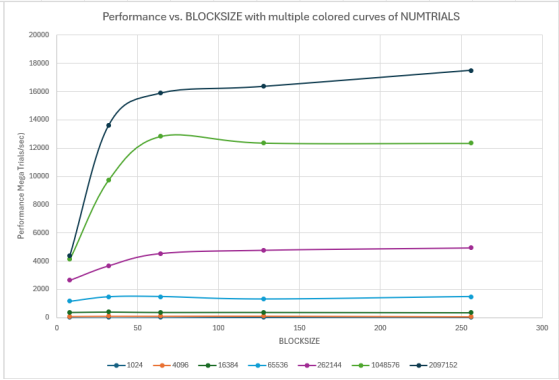
Performance Analysis

Performance Data

Trials	BlockSize	MegaTrails/sec	Probability
1024	8	24.39	8.11
1024	32	24.39	6.74
1024	64	25	8.69
1024	128	26.32	10.94
1024	256	24.39	9.86
4096	8	90.91	8.91
4096	32	108.11	8.54
4096	64	111.11	8.89
4096	128	114.29	8.62
4096	256	70.18	9.23
16384	8	380.95	8.84
16384	32	410.26	9.16
16384	64	380.95	8.65
16384	128	380.95	8.62
16384	256	355.56	8.95
65536	8	1166.95	8.68
65536	32	1472.32	8.9
65536	64	1489.45	8.62
65536	128	1323.85	8.97
65536	256	1488.37	8.78
262144	8	2640.88	8.66
262144	32	3673.54	8.82
262144	64	4546.06	8.73
262144	128	4782.25	8.78
262144	256	4946.86	8.74
1048576	8	4134.24	8.77
1048576	32	9752.38	8.74
1048576	64	12840.12	8.81
1048576	128	12374.62	8.76
1048576	256	12351.3	8.74
2097152	8	4404.6	8.74
2097152	32	13633.45	8.75
2097152	64	15895.22	8.78
2097152	128	16375.81	8.75
2097152	256	17508.95	8.75

	8	32	64	128	256
1024	24.39	24.39	25	26.32	24.39
4096	90.91	108.11	111.11	114.29	70.18
16384	380.95	410.26	380.95	380.95	355.56
65536	1166.95	1472.32	1489.45	1323.85	1488.37
262144	2640.88	3673.54	4546.06	4782.25	4946.86
1048576	4134.24	9752.38	12840.12	12374.62	12351.3
2097152	4404.6	13633.45	15895.22	16375.81	17508.95





Probability Analysis

The average probability of hitting the castle in this simulation is approximately 8.75% due to the different random paramaters.

Performance Patterns

1. As the number of trials increases, the MegaTrials/Second also increases substantially, from ~25 MT/s at 1,024 trials to ~17,500 MT/s at 2,097,152 trials. This shows that the GPU becomes more efficient when processing larger workloads.

2. Larger block sizes perform better than smaller ones as the problem size increases. At the largest trial size (2,097,152), performance ranges from 4,404 MT/s with block size 8 to 17,508 MT/s with block size 256.

3. The performance improvement from increasing block size becomes less significant at extremely large trial size since the GPU hits a limit at hoe much more it can process in parallel and threads are in groups of 32.

Why Block Size 8 Is Worse

1. CUDA GPUs put threads into groups of 32, leading to wasted computational resources.

2. Small block sizes result in more blocks but don't efficiently utilize the GPU's thread-level parallelism capabilities.

3. There's a fixed overhead associated with launching blocks. With many small blocks, this overhead becomes significant compared to the actual computation time.

Vs. Project #1

In Project #1, the best performance achieved was approximately 84-91 MegaTrials/Second. This CUDA implementation reaches over 17,500 MegaTrials/Second, a massive improvement in performance due to the incredible number of cores a GPU has, their efficiency at calculations, memory bandwidth, and simply the fact that they are design for parallel processing.

What you can do with GPU parallel computing

1. Monte Carlo and other similar scenarios are ideal for GPU acceleration due to their parallel nature.
2. GPUs excel at problems that can be divided into thousands or millions of independent tasks.
3. GPUs can provide massively better performance than CPUs for certain types of calculations, especially those involving large datasets.