Project5

CUDA: Monte Carlo Simulation

Name: Yi Chiun Chang
Email: changyic@oregonstate.edu

1. Tell what machine you ran this on

CyberDuck(rabbit.engr.oregonstate.edu & submit-a.engr.oregonstate.edu)

2. What do you think this new probability is?

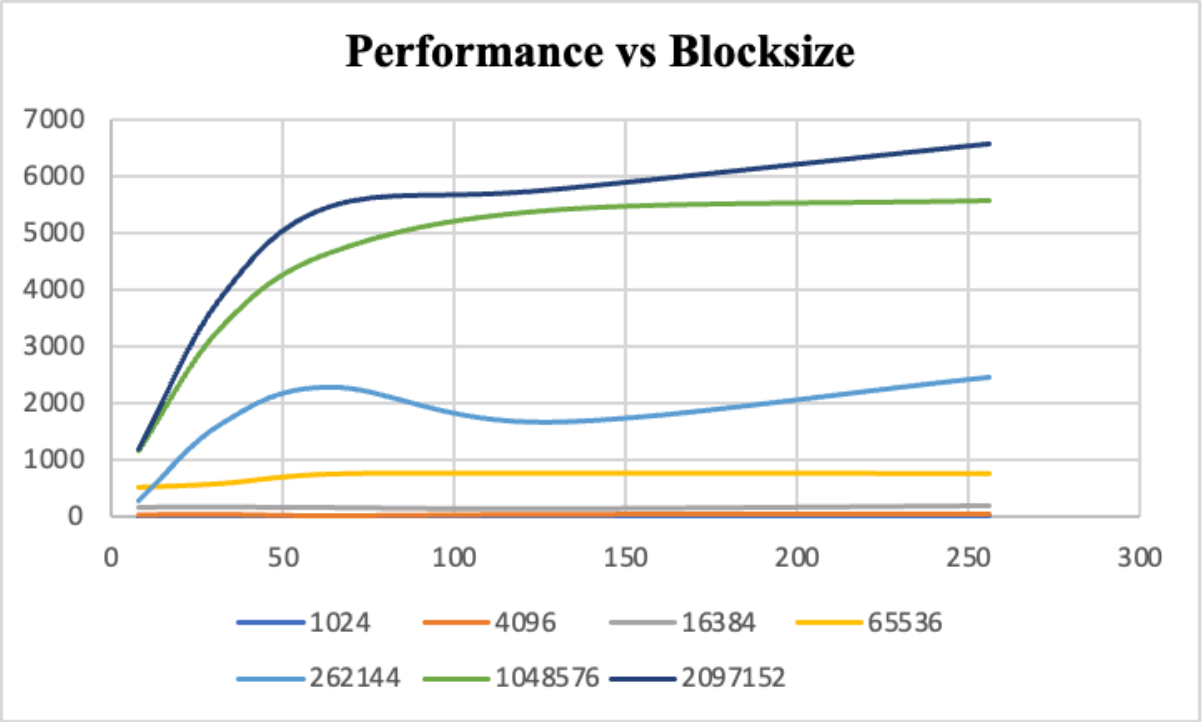
The average probability based on the Monte Carlo simulation results, is approximately 83.79%.

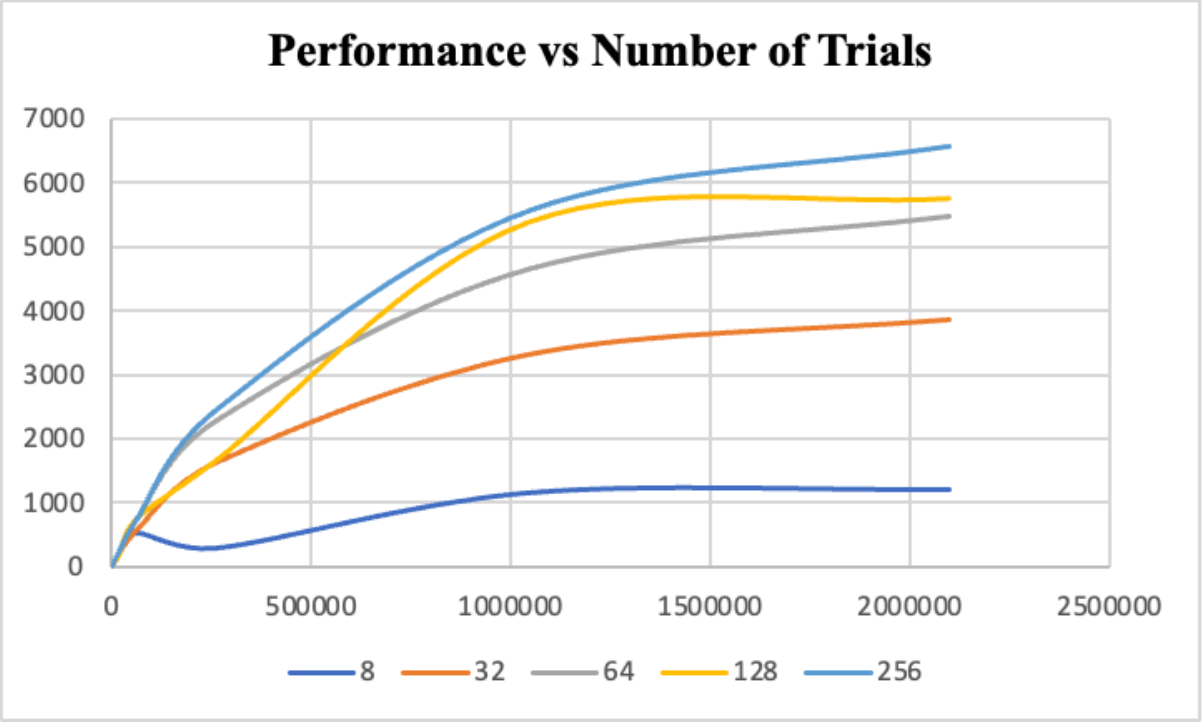
3. Show the rectangular table and the two graphs

Table:

NUMTRIALS	BLOCKSIZE	megaTrialsPerSecond	probability
1024	8	8.0869	86.23
1024	32	13.1687	84.67
1024	64	12.0937	81.84
1024	128	12.4224	82.62
1024	256	13.2231	85.35
4096	8	47.6013	82.89
4096	32	50.0782	82.89
4096	64	41.926	83.74
4096	128	50.3739	84.45
4096	256	52.459	82.96
16384	8	182.0768	83.64
16384	32	185.2388	84.14
16384	64	176.0055	83.21
16384	128	155.3398	83.9
16384	256	208.8091	83.42
65536	8	523.7852	83.88
65536	32	590.5421	83.78
65536	64	760.7727	83.77
65536	128	773.4139	83.81
65536	256	771.9563	83.95
262144	8	289.1328	83.71
262144	32	1631.5476	83.79
262144	64	2279.3545	83.9
262144	128	1664.7023	83.92
262144	256	2451.2268	83.93
1048576	8	1151.5726	83.76
1048576	32	3324.0008	83.8
1048576	64	4649.922	83.8
1048576	128	5395.6859	83.86
1048576	256	5568.0545	83.84
2097152	8	1200.9969	83.83
2097152	32	3863.4677	83.84
2097152	64	5465.8884	83.84
2097152	128	5756.8517	83.83
2097152	256	6564.7599	83.81

Graphs:





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

The performance of the Monte Carlo simulation improves with increasing blocksize and number of trials, exhibiting a steep initial rise followed by diminishing returns. Larger blocksizes consistently deliver higher performance, particularly for larger trial sizes, but the efficiency gain plateaus beyond

an optimal point. Higher trial sizes benefit more from increased blocksize, maximizing GPU resource utilization. However, performance gains slow down and may slightly decrease at the highest blocksizes and trial sizes, indicating a saturation point. Optimal performance is achieved with a balanced selection of blocksize and trial size, beyond which additional increases yield minimal improvements.

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

The patterns reflect the balance between computational efficiency and resource saturation, where increasing blocksize and trial size initially enhance performance by better utilizing GPU resources, but eventually lead to diminishing returns as the system approaches its maximum capacity.

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

A BLOCKSIZE of 8 performs significantly worse than larger blocksizes because it underutilizes the GPU's parallel processing capabilities. GPUs are optimized for handling large numbers of threads simultaneously, and a smaller blocksize fails to leverage this advantage, resulting in lower computational efficiency and increased overhead per thread. This inefficiency is particularly evident as the number of trials increases, where the disparity in performance becomes more pronounced compared to larger blocksizes that better match the GPU's architecture and maximize resource utilization.

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

The performance results in Project #5 are generally higher compared to Project #1, showing a similar initial abrupt increase followed by a table as the number of trials increases. This difference in performance can be attributed to variations in kernel implementation, workload characteristics, and parallel efficiency. Project #5 may have benefited from optimizations and more efficient utilization of GPU resources, leading to better overall performance.

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

This means that to fully leverage GPU parallel computing, it is essential to optimize the configuration by using sufficiently large blocksizes and trial sizes that align with the GPU's architecture, maximizing its parallel processing capabilities and minimizing overhead, thereby achieving significant performance gains for computationally intensive tasks.