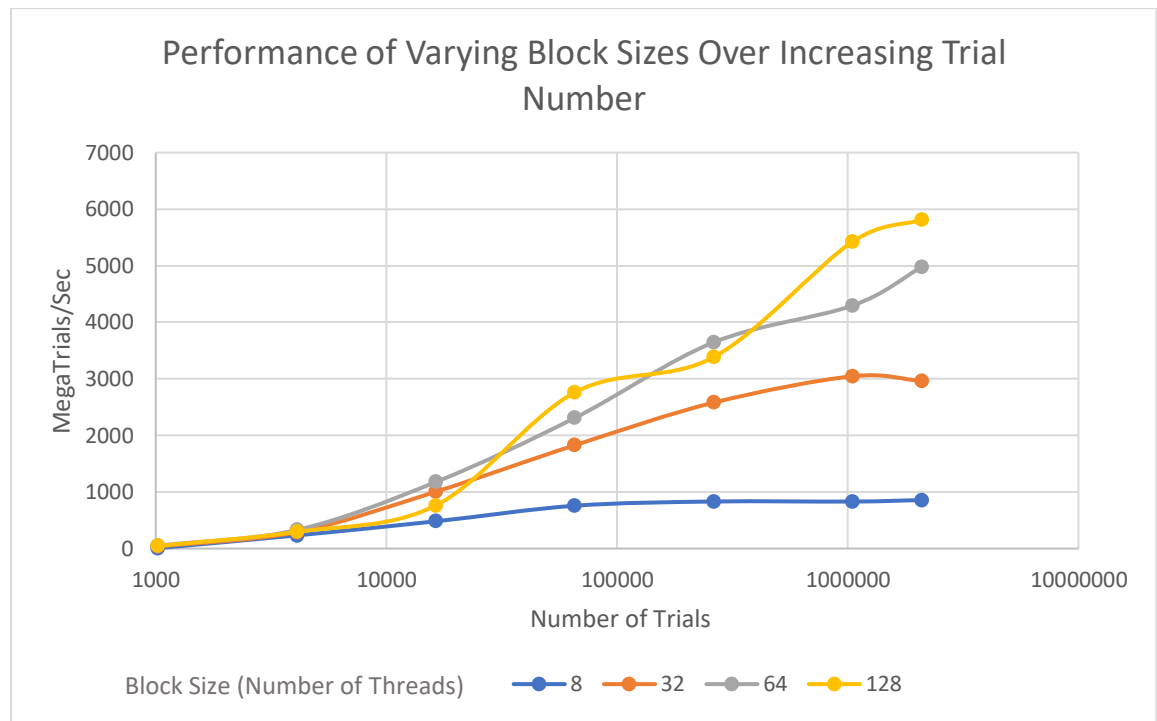
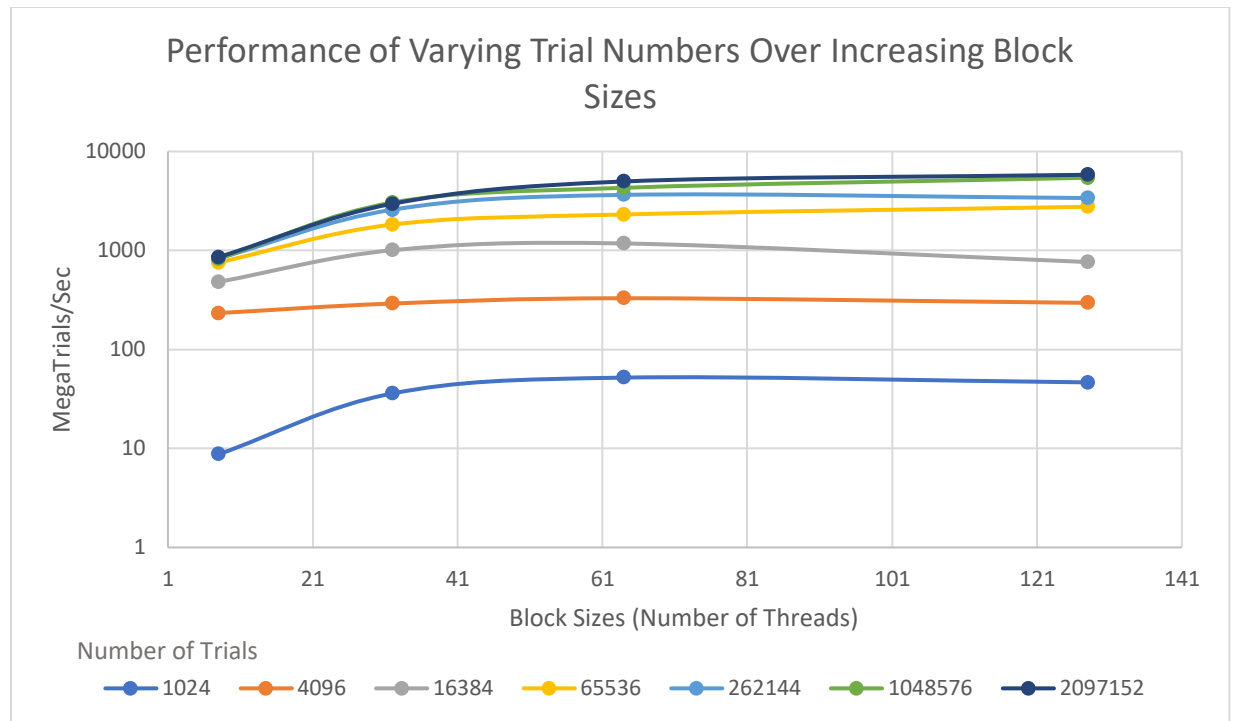


Project 5 Analysis

1. **Tell what machine you ran this on.** I ran this on rabbit.
2. **What do you think this new probability is?** Based on the average of all runs with a trial size of 2097152, the probability is approximately 26.88%.
3. **Show the table and the two graphs:**

		Number of Trials						
		1024	4096	16384	65536	262144	1048576	2097152
Block Size	8	8.7408	233.1512	482.1092	757.1164	831.8441	830.9792	857.3522
	32	35.9551	290.9091	1005.894	1826.9402	2580.1575	3045.9192	2962.7485
	64	51.9481	329.0488	1177.0115	2308.9064	3645.7497	4296.3157	4978.4261
	128	46.5116	295.612	764.1791	2760.1078	3383.7257	5425.1657	5814.0524





4. **What patterns are you seeing in the performance curves?** As the number of trials increase, the performance scales logarithmically. The same logarithmic scaling occurs with increasing thread counts, however, the overall performance of lower number of trials is far less effective than larger trial numbers. As seen in the second graph the max performance of trial number = 1024 never surpasses 100 MegaTrials/Sec, whereas the performance of trial numbers greater than 262144 surpasses 5,000 MegaTrials/Sec. Most likely due to reaching the GPU's full processing capabilities as the data set size increases.
5. **Why do you think the patterns look this way?** The patterns in the performance curves can be attributed to the interaction between the number of trials and the block size. As the number of trials increases, the MegaTrials/Second generally tends to increase, indicating improved performance. However, the rate of increase slows down as the number of trials becomes larger. Smaller block size values tend to result in lower MegaTrials/Second values compared to larger block size values. This can be due to factors like increased overhead relative to the processing.
6. **Why is a BLOCKSIZE of 8 so much worse than the others?** This can be due to factors like increased overhead relative to the processing. Also, as mentioned before, this can be because it is too small, leading to suboptimal GPU utilization. Smaller blocksize values may cause underutilization of GPU resources, as the GPU threads have to wait more frequently for memory accesses, resulting in lower performance.
7. **How do these performance results compare with what you got in Project #1?**
Why? It appears that the MegaTrials/Second values in the this project's data tend to be much higher compared to Project #1 (only reaching just under 200 MegaTrials/Sec

compared to GPUs 6000), especially for larger numbers of trials and threads. This improvement in performance can be attributed to various factors, such as optimizations in the code, improvements in hardware, or better utilization of GPU resources. GPUs can have much higher performance in parallelism than parallelizing CPUs since GPUs hardware is designed for parallelism in simple computations such as this Monte Carlo simulation making it a much better option in this case. It's important to note that a higher number of trials and threads generally provide more opportunities for parallelism, which can lead to improved performance in GPU parallel programming.

8. **What does this mean for what you can do with GPU parallel computing?** The results indicate that GPU parallel computing can significantly improve performance by leveraging parallelism in the GPU architecture. With appropriate tuning of parameters such as the number of trials and blocksize, the MegaTrials/Second values can be increased, leading to faster computation.