Gretel Rajamoney

rajamong@oregonstate.edu

Project #5

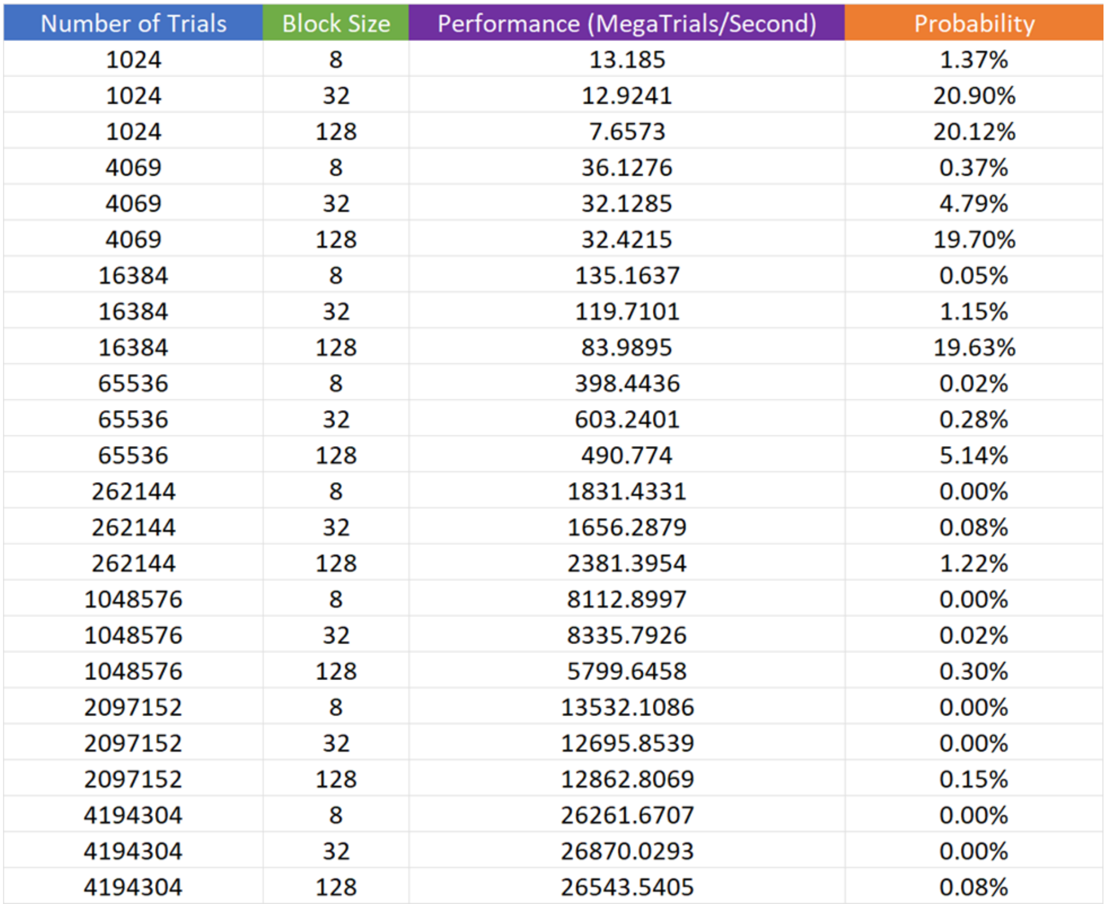
Project Questions:

1. What machine did you run this on?
   = I ran my program on my Windows machine on Visual Studio Code utilizing the engineering server rabbit.engr.oregonstate.edu. To run my program in the terminal, I inputted the following lines of code:

```
chmod u+x proj05.sh
sh proj05.sh >& proj05.csv
```

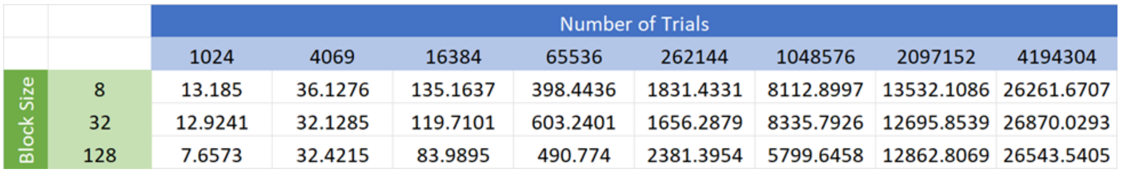
2. Show the table and the two graphs?

Results Table:



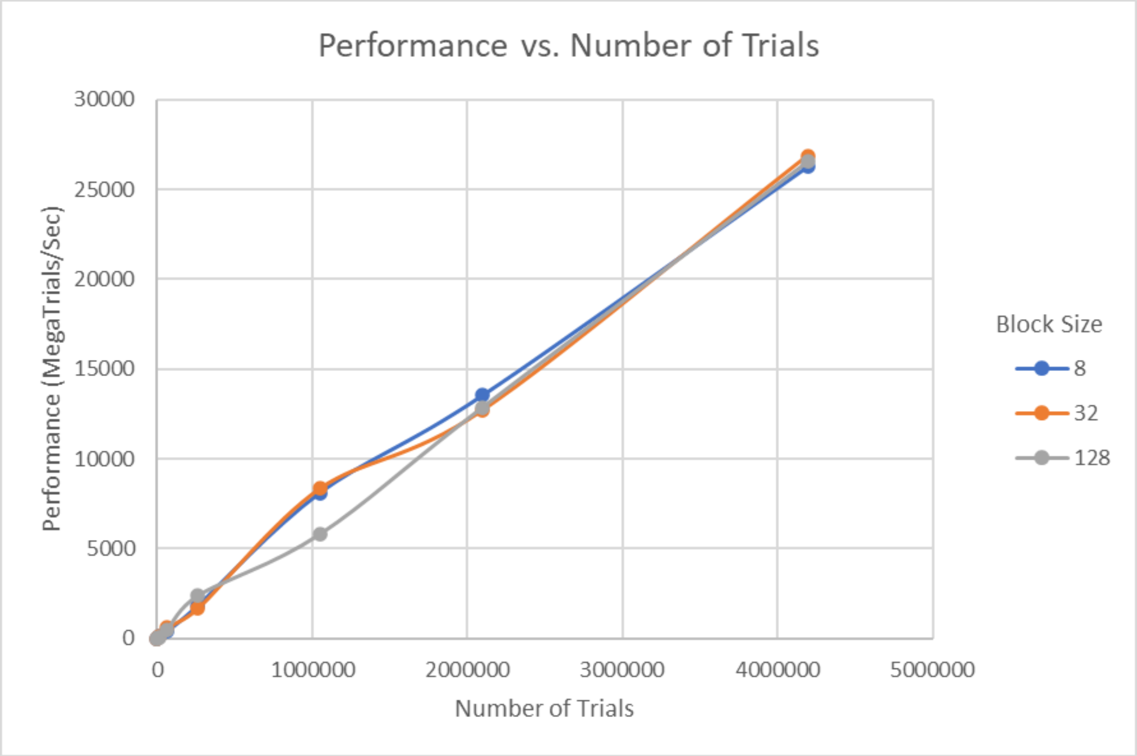
Number of Trials	Block Size	Performance (MegaTrials/Second)	Probability
1024	8	13.185	1.37%
1024	32	12.9241	20.90%
1024	128	7.6573	20.12%
4069	8	36.1276	0.37%
4069	32	32.1285	4.79%
4069	128	32.4215	19.70%
16384	8	135.1637	0.05%
16384	32	119.7101	1.15%
16384	128	83.9895	19.63%
65536	8	398.4436	0.02%
65536	32	603.2401	0.28%
65536	128	490.774	5.14%
262144	8	1831.4331	0.00%
262144	32	1656.2879	0.08%
262144	128	2381.3954	1.22%
1048576	8	8112.8997	0.00%
1048576	32	8335.7926	0.02%
1048576	128	5799.6458	0.30%
2097152	8	13532.1086	0.00%
2097152	32	12695.8539	0.00%
2097152	128	12862.8069	0.15%
4194304	8	26261.6707	0.00%
4194304	32	26870.0293	0.00%
4194304	128	26543.5405	0.08%

Pivot Table of Performance in MegaTrials/Second:

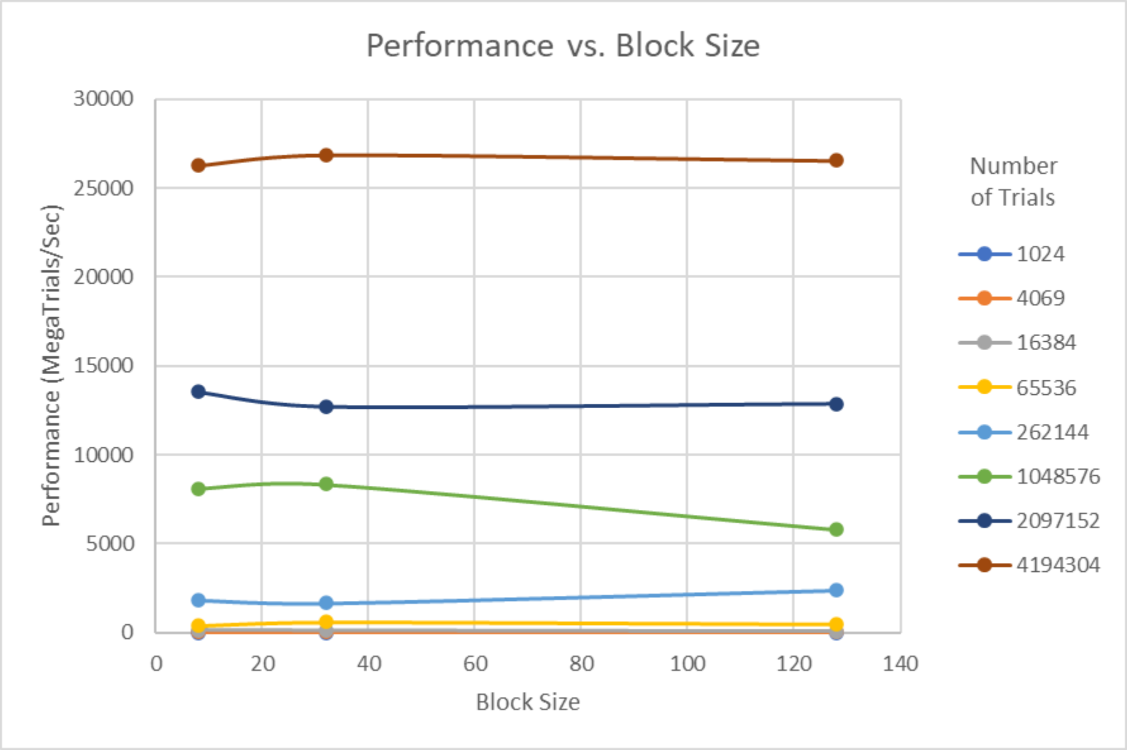


		Number of Trials							
		1024	4069	16384	65536	262144	1048576	2097152	4194304
Block Size	8	13.185	36.1276	135.1637	398.4436	1831.4331	8112.8997	13532.1086	26261.6707
	32	12.9241	32.1285	119.7101	603.2401	1656.2879	8335.7926	12695.8539	26870.0293
	128	7.6573	32.4215	83.9895	490.774	2381.3954	5799.6458	12862.8069	26543.5405

Graph of Performance vs. Number of Trials:



Graph of Performance vs. Block Size:



3. What patterns are you seeing in the performance curves?
   = In the graph displaying ‘Performance vs. Number of Trials’, it can easily be seen that as the number of trials increases, the performance also increases. The relationship between number of trials and performance is positively correlated. Since the lines appear to be overlapping within the graph, it is difficult to notice any differences regarding block size affect the performance of the program. In the graph displaying ‘Performance vs. Block Size’, all the curves appear to be mainly linear with a few decreases and increases in performance throughout. Through seeing each curve representing a different number of trials, it can be easily seen that a higher number of trials leads to a significantly better performance. When the number of trials is set to 419304 trials, the performance is significantly greater than the rest of the curves present on the graph.

4. Why do you think the patterns look this way?
   = It is incredibly noticeable the impact that increasing the number of trials has on the performance of the program. This is because the increase in trial size allows the program to receive a larger dataset for the program's GPU to analyze. The increase in performance due to the increase in block size is due to the fact that a higher number of readily available threads significantly reduces time which results in higher efficiencies, hence the higher performance.

5. Why is a BLOCKSIZE of 8 so much worse than the others?
   = When the block size is 8, the performance is in general worse in comparison to the block sizes of 32 and 128 due to computing capacity. Since CUDA works with warps, it essentially utilizes 32 thread units to compute the program. The block size of 32 factors perfectly when divided by a warp, and a block size of 128 factors perfectly into 4 warps. On the contrary, a block size of 8 results in only one-quarter filled warps, this means that the 32 thread carrying capacity is not being efficiently utilized. This inefficiency is the contributing factor as to why the block size of 8 is significantly worse in comparison to block sizes of 32 and 128.

6. How do these performance results compare with what you got in Project #1? Why?
   In Project #1, my performances were their best when the number of trials were at its largest at 1000000 trials, and my number of threads was at its largest at 32 threads. These results compare very similarly to the results of this project because as the number of threads and the number of trials increase, the performance of the program also increases as well. Although Project #5 operates using the CUDA GPU and the unit of warps as opposed to Project #1 where it utilizes the CPU and the unit of threads, there is a positive relationship with these

units and program performance in MegaTrials per Second. In my results from Project #1, when I had a number of threads set to 32 and a number of trials set to 1000000, the performance of my program was 286.1852 MegaTrials/Second. A close comparison to these values from Project #5 can be a block size of 32 and a number of trials set to 1048576, which resulted in a performance of 8335.7926 MegaTrials/Second. This comparison between the 286.1852 MegaTrials/Second we got from Project #1 and the 8335.7926 MegaTrials/Second we get from Project #5, we can make the reasonable conclusion that the CUDA GPU performs significantly better when compared to the CPU.

7. What does this mean for the proper use of GPU parallel computing?
   = In order to properly use the GPU for parallel computing, it is important that we understand how block size and number of trials impacts the overall performance of the program. Although we have made the conclusion that increasing block size increases the performance of the program, we must ensure that we set the block size to be evenly factorable with 32. Without setting our block size to be everly factorable with 32, we will be inefficiently utilizing our carrying capacity that the GPU provides us with, essentially executing incomplete warps. We also found that as the number of trials increases, the performance of the program when using a GPU also increases because the system is created in order to parse large datasets. Therefore in order to properly use the GPU for parallel computing, we should use it for large trial sizes containing large datasets and proper block sizes that correctly fill warps.