Gretel Rajamoney

rajamong@oregonstate.edu

Project #5

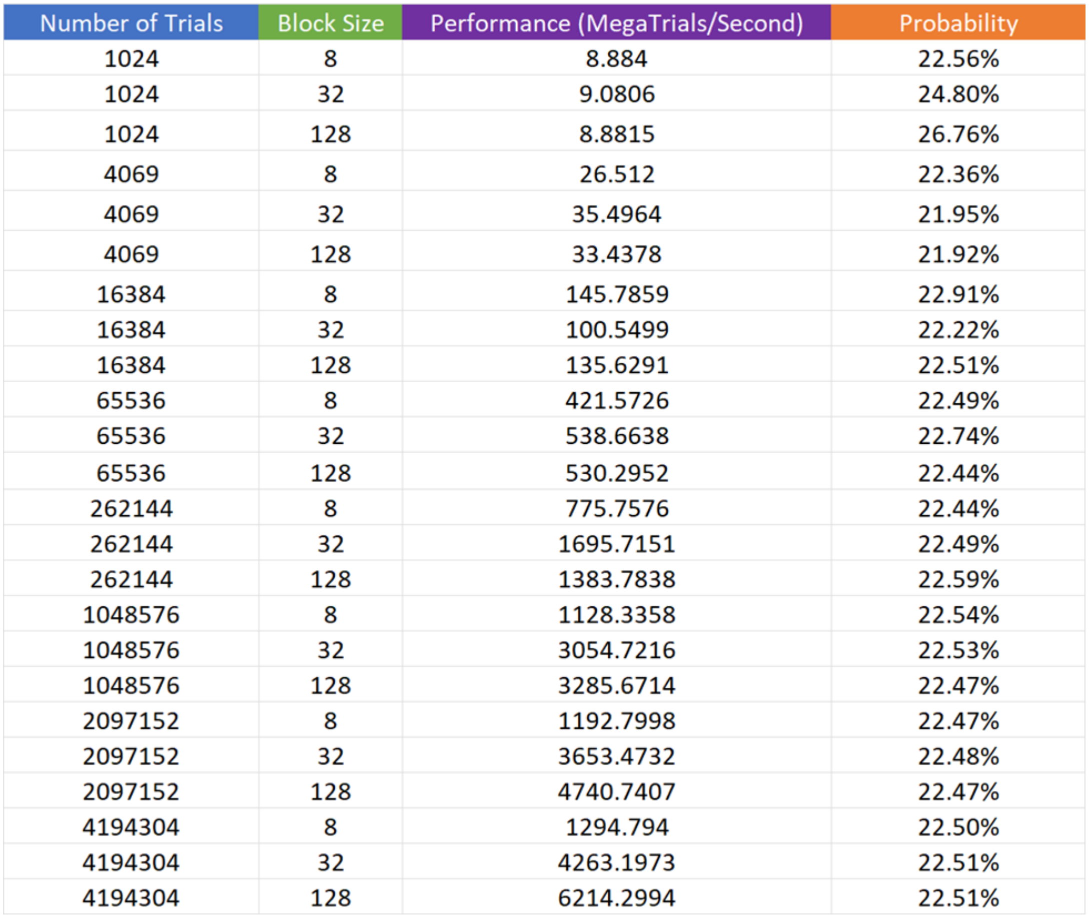
Project Questions:

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
   ```

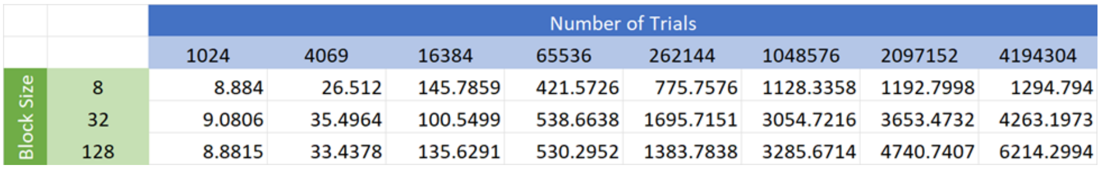
2. Show the table and the two graphs?

   Results Table:



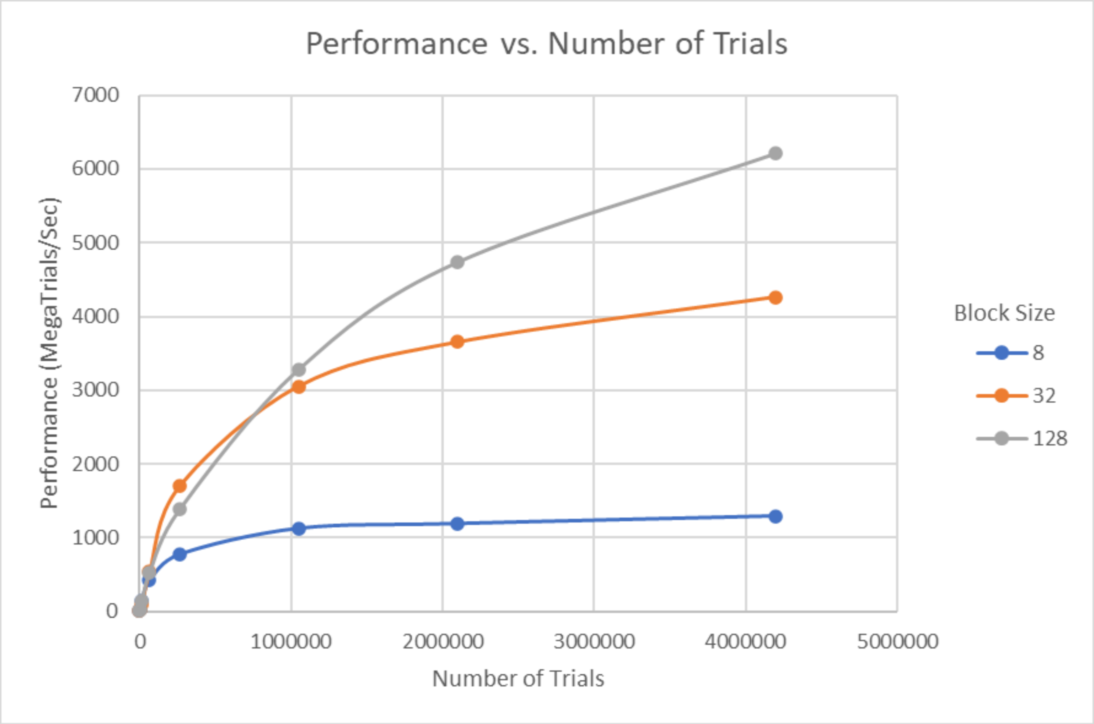
Number of Trials	Block Size	Performance (MegaTrials/Second)	Probability
1024	8	8.884	22.56%
1024	32	9.0806	24.80%
1024	128	8.8815	26.76%
4069	8	26.512	22.36%
4069	32	35.4964	21.95%
4069	128	33.4378	21.92%
16384	8	145.7859	22.91%
16384	32	100.5499	22.22%
16384	128	135.6291	22.51%
65536	8	421.5726	22.49%
65536	32	538.6638	22.74%
65536	128	530.2952	22.44%
262144	8	775.7576	22.44%
262144	32	1695.7151	22.49%
262144	128	1383.7838	22.59%
1048576	8	1128.3358	22.54%
1048576	32	3054.7216	22.53%
1048576	128	3285.6714	22.47%
2097152	8	1192.7998	22.47%
2097152	32	3653.4732	22.48%
2097152	128	4740.7407	22.47%
4194304	8	1294.794	22.50%
4194304	32	4263.1973	22.51%
4194304	128	6214.2994	22.51%

Pivot Table of Performance in MegaTrials/Second:

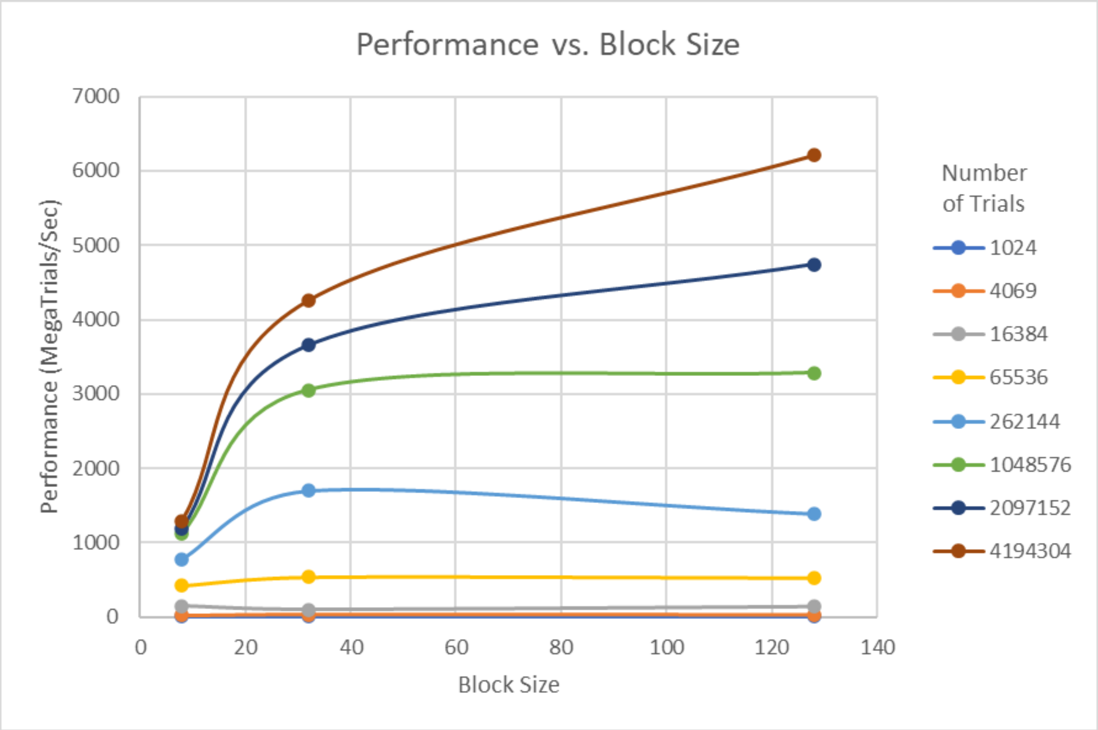


		Number of Trials							
		1024	4069	16384	65536	262144	1048576	2097152	4194304
Block Size	8	8.884	26.512	145.7859	421.5726	775.7576	1128.3358	1192.7998	1294.794
	32	9.0806	35.4964	100.5499	538.6638	1695.7151	3054.7216	3653.4732	4263.1973
	128	8.8815	33.4378	135.6291	530.2952	1383.7838	3285.6714	4740.7407	6214.2994

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 both directly and positively correlated. Block size also appears to play a major role in the performance of the program, the larger the block size the better the program performs overall. In the graph displaying ‘Performance vs. Block Size’, all the curves appear to be rapidly increasing between the block size intervals of 8 to 32, but then leveling out as it reaches the block size of 128. 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 and block size 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. Block size plays a role in this pattern because CUDA works with 32-thread warps, meaning that better utilized carrying capacity between each instruction will result in a better performing program. A block size of 8 is a quarter utilized carrying capacity, a block size of 32 is a fully utilized carrying capacity, and a block size of 128 is a 4 warp sized carrying capacity. These carrying capacities respective to each block size explain why the patterns appear the way that they do.

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 into one unit 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 performs significantly worse in comparison to performances of 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 3054.7216 MegaTrials/Second. This comparison between the 286.1852 MegaTrials/Second we got from Project #1 and the 3054.7216 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.