CS 575

Project 5

CUDA: Monte Carlo Simulation

Submitted By: Aman Pandita

ONID: panditaa@oregonstate.edu

1. **Tell what machine you ran this on?**

   Rabbit (rabbit.engr.oregonstate.edu)

2. **Show the table and the two graphs?**

Fig: Simulation Data

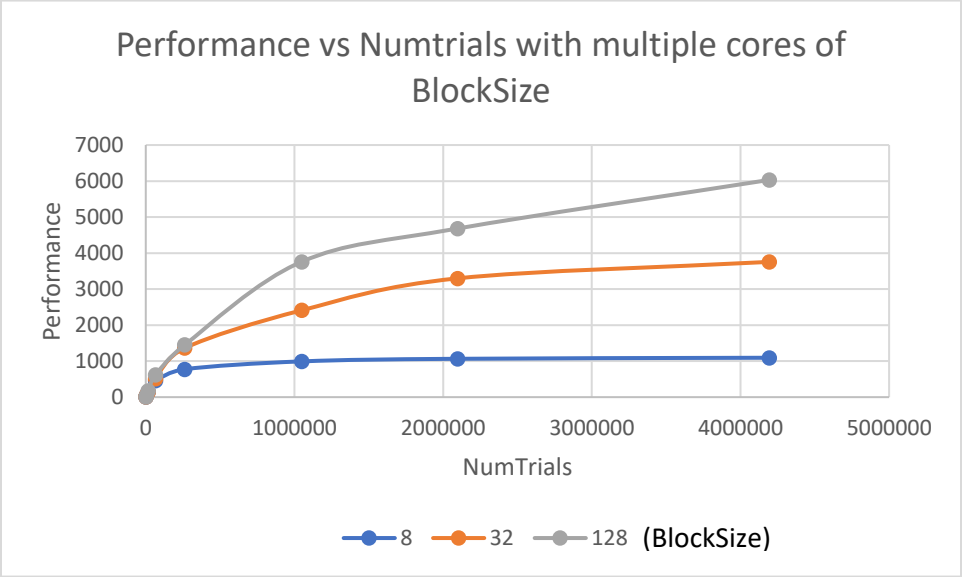
Numtrials	BlockSize	megaTrialsPerSecond	Probability
1024	8	9.2673	23.73%
1024	32	7.7201	23.73%
1024	128	7.6573	21.88%
4096	8	36.148	22.31%
4096	32	40	21.88%
4096	128	37.8698	22.61%
16384	8	147.9341	22.55%
16384	32	142.6184	22.77%
16384	128	171.9852	22.03%
65536	8	456.735	22.88%
65536	32	514.4436	22.47%
65536	128	619.2924	22.57%
262144	8	774.5839	22.30%
262144	32	1364.1965	22.59%
262144	128	1451.4528	22.43%
1048576	8	992.9095	22.49%
1048576	32	2411.0073	22.55%
1048576	128	3759.9541	22.53%
2097152	8	1065.1064	22.48%
2097152	32	3298.5707	22.52%
2097152	128	4681.8117	22.54%
4194304	8	1093.9897	22.53%
4194304	32	3755.7523	22.49%
4194304	128	6031.5677	22.51%

From the above table, my observation of the actual correct probability is around 22%

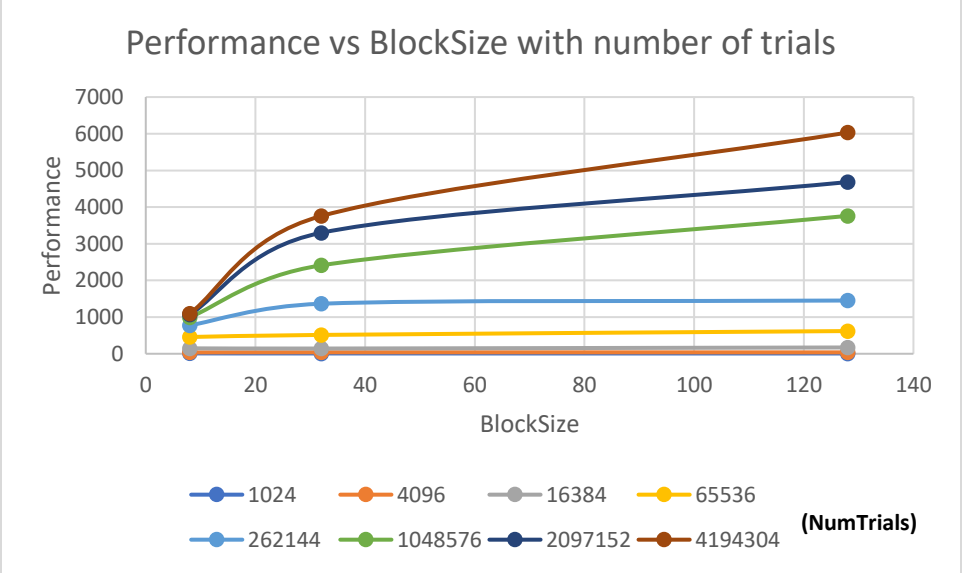
Fig: Pivot Table:

	1024	4096	16384	65536	262144	1048576	2097152	4194304
8	9.2673	36.148	147.9341	456.735	774.5839	992.9095	1065.106	1093.99
32	7.7201	40	142.6184	514.4436	1364.197	2411.007	3298.571	3755.752
128	7.6573	37.8698	171.9852	619.2924	1451.453	3759.954	4681.812	6031.568

- **Graph for Performance vs Numtrials with multiple cores of BlockSize**



- **Graph for Performance vs BlockSize with number of trials**



3. **What patterns are you seeing in the performance curves?**

   a) Performance vs NumTrials with multiple cores of BlockSize

   We can see that the performance increases as the number of trials increase. But we can see that initially the performance increase is very rapid but as the number of trials increase, the performance drops. One more thing that we can conclude is that the block size increase impacts the performance by increasing it when the number of trials are more.

   B) Performance vs BlockSize with Numtrials

   We can see that the performance increases as the BlockSize increases when the number of trials is high. At a smaller number of trials, there isn't much change even when the block size is changed.

4. **Why do you think the patterns look this way?**

   As the scheduler needs to arrange a greater number of blocks for each processor, but the BlockSize remains the same when the data set is increased. This means that the work done by the scheduler will be more. Due to this the calculations take more time and performance improvement also decreases. The solution to this problem is to increase the BlockSize for better performance.

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

   This might be due to the fact that 32 and 128 are multiples of 32, this allows them to fully utilize and improve the efficiency greatly. We can fill minimum 32 threads in the system here, but in the case of 8, we are leaving 24 threads idle without any work so the overall performance will be impacted hugely. The same is observed from the graphs.

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

   There is a great difference in the Project #1 and Project #5 graphical representations as in Project #1, CPU threads were used instead of GPU which are larger.

7. **What does this mean for the proper use of GPU parallel computing?**

   To maximize the performance, we can use GPU parallel computing as it reacts to increase in data sets positively. We have seen that the performance improves when the data set is increased. To make sure that each block size gets a complete queue, data selection is based on the multiple of 32. Moreover, performance doesn't improve any further if the block size is maxed out.