**Name**: Shreya Palit
**Email**: palits@oregonstate.edu
**Project Name:** CUDA: Monte Carlo Simulation

## CS 575 - Project #5

1. Tell what machine you ran this on

I ran this on the DGX server.

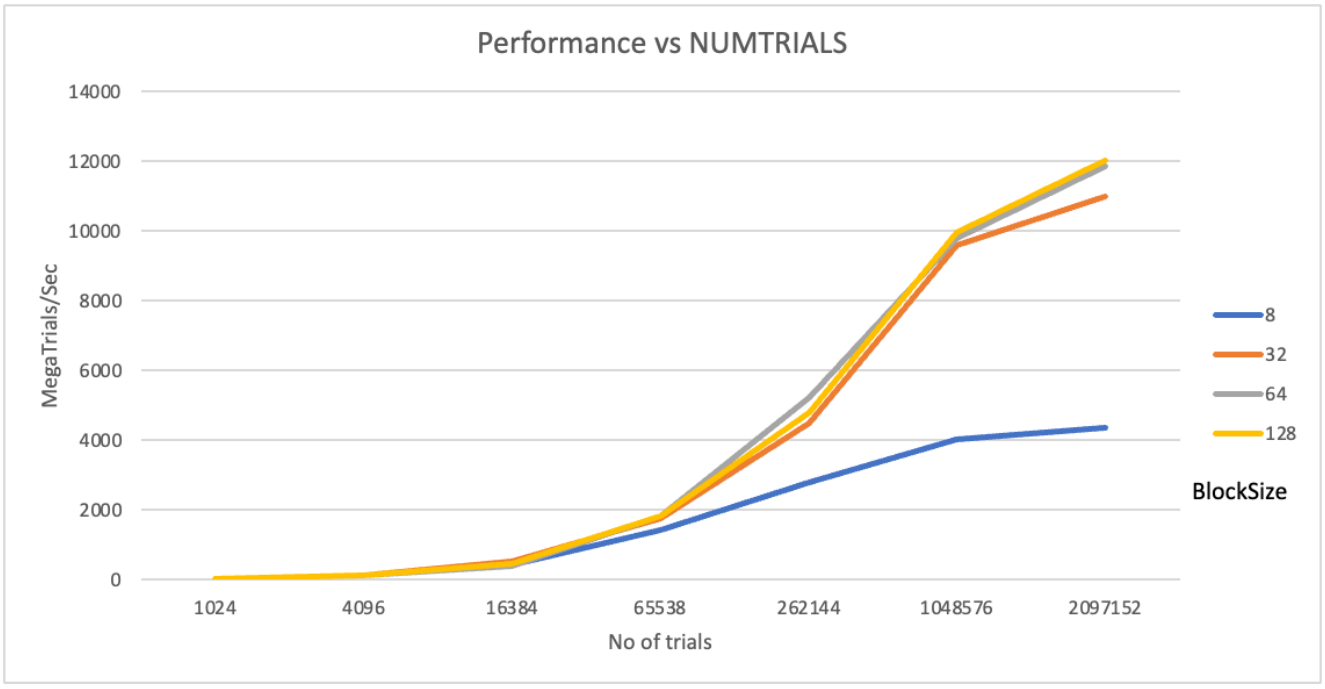
2. What do you think this new probability is?

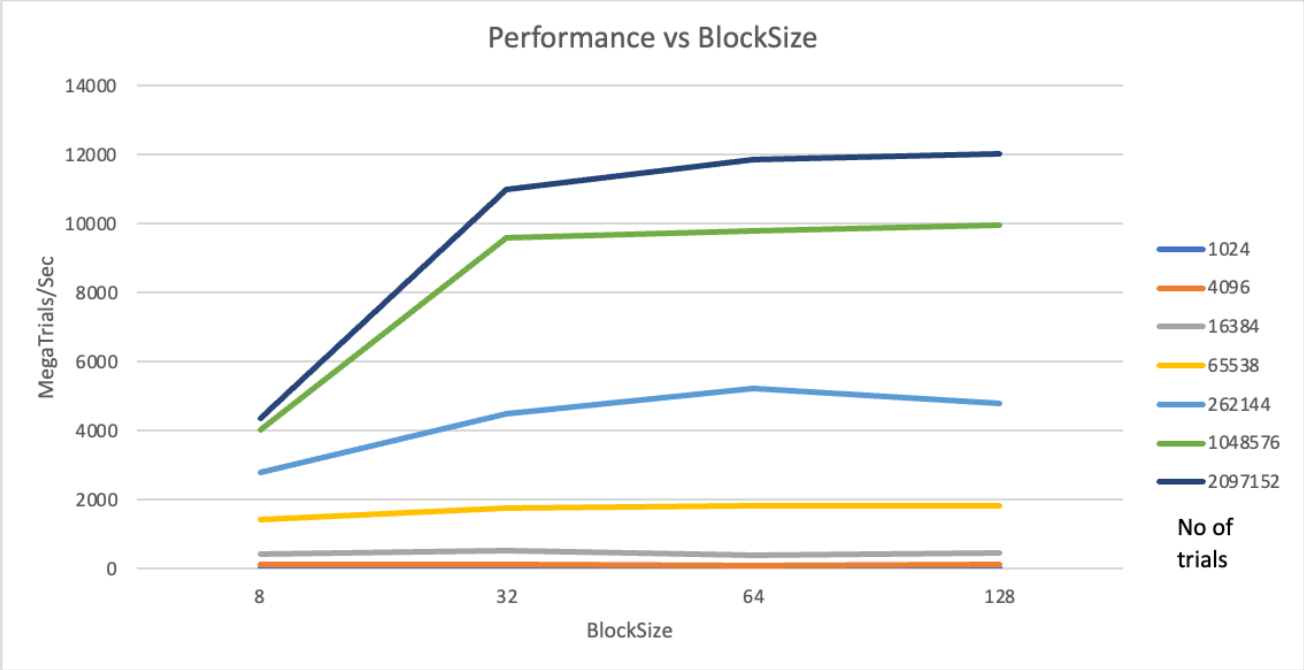
The average of the new probability is coming out to **74.74%**.

3. Show the table and the two graphs

| Number of Trials | BlockSize | MegaTrials/Second | Probability |
|---|---|---|---|
| 1024 | 8 | 31.25 | 75.59 |
| 1024 | 32 | 31.25 | 76.27 |
| 1024 | 64 | 31.25 | 72.75 |
| 1024 | 128 | 31.25 | 73.54 |
| 4096 | 8 | 121.21 | 75.66 |
| 4096 | 32 | 125 | 74.9 |
| 4096 | 64 | 117.65 | 75.02 |
| 4096 | 128 | 137.93 | 74.76 |
| 16384 | 8 | 432.43 | 75.2 |
| 16384 | 32 | 516.13 | 74.71 |
| 16384 | 64 | 390.24 | 74.57 |
| 16384 | 128 | 457.14 | 74.69 |
| 65538 | 8 | 1424.2 | 74.53 |
| 65538 | 32 | 1744.46 | 74.58 |
| 65538 | 64 | 1836.77 | 74.63 |
| 65538 | 128 | 1833.48 | 74.85 |
| 262144 | 8 | 2785.45 | 74.73 |
| 262144 | 32 | 4501.1 | 74.67 |

| | | | |
|---|---|---|---|
| 262144 | 64 | 5207.88 | 74.69 |
| 262144 | 128 | 4787.84 | 74.76 |
| 1048576 | 8 | 4024.56 | 74.66 |
| 1048576 | 32 | 9598.13 | 74.73 |
| 1048576 | 64 | 9790.26 | 74.77 |
| 1048576 | 128 | 9968.97 | 74.78 |
| 2097152 | 8 | 4348.48 | 74.67 |
| 2097152 | 32 | 11003.36 | 74.74 |
| 2097152 | 64 | 11866.02 | 74.67 |
| 2097152 | 128 | 12038.21 | 74.69 |






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

In the above graphs, I am seeing a significant performance increase as the number of blocks increases. Even though there have been around 2 million trials, the performance is continually growing. The maximum performance of the program executing on the GPU (using DGX) appears to be for the 128 blocks (which is the highest in this case) for 2097152 trials.

In the Performance vs NumTrials graph it can be seen that the curves for blocksize 64 and 128 almost overlap for all the different no of trials. In the same graph it can also be seen that after 65538 trials, the performance increases steeply for all block sizes.

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

The reason for patterns to look this way is because this code is utilizing GPU, which can handle enormous data sets. Once the block size is appropriate for the data amount, more data usually means higher efficiency. In addition, for parallel computing, many data sets can be collected at the same time and stored in different blocks just to ensure higher efficiency. In other words, if block size increases then the performance will also increase. Hence the graphs look like the way they do - an increase in block size is making the curves go up (increase in performance).

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

Blocksize 8 indicates that there are only 8 threads per block. When utilizing GPU, more threads per block means greater performance. Here, the lowest number of blocksize is 8 so this is the reason why the performance is significantly lower than the others.

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

Since this project uses CUDA to compute some functions, the GPU permits a large number of threads to do the calculation. This means threads can execute instructions concurrently which means an efficient use of GPU. On the other hand, for Project #1, we used OpenMP which just relied on available CPU cores which is not as efficient as using GPU. Hence the performance in Project #5 is significantly greater than the performance in Project #1.

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

This project gave me the understanding that the GPU is capable of handling large amounts of data as it will be more efficient. If I have to deal with lots of trials then I will use GPU parallel computing to ensure efficient performance.