# Project 1

## Monte Carlo Experiment

**Submitted By: Aman Pandita**

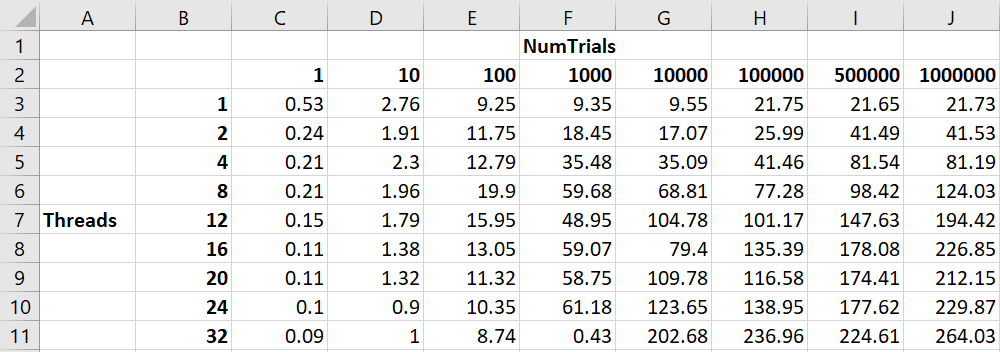
**OSU ID: 934-456-235**

**ONID:** [**panditaa@oregonstate.edu**](mailto:panditaa@oregonstate.edu)

**The results of the execution of the program are as follows:**

|  |  |  |  |
| --- | --- | --- | --- |
| Threads | NumTrials | Probability | Max Performance |
| 1 | 1 | 0 | 0.53 |
| 1 | 10 | 20 | 2.76 |
| 1 | 100 | 26 | 9.25 |
| 1 | 1000 | 30.8 | 9.35 |
| 1 | 10000 | 28.82 | 9.55 |
| 1 | 100000 | 29.33 | 21.75 |
| 1 | 500000 | 29.07 | 21.65 |
| 1 | 1000000 | 29 | 21.73 |
| 2 | 1 | 0 | 0.24 |
| 2 | 10 | 30 | 1.91 |
| 2 | 100 | 26 | 11.75 |
| 2 | 1000 | 29.8 | 18.45 |
| 2 | 10000 | 29.13 | 17.07 |
| 2 | 100000 | 29.08 | 25.99 |
| 2 | 500000 | 29.11 | 41.49 |
| 2 | 1000000 | 29.03 | 41.53 |
| 4 | 1 | 100 | 0.21 |
| 4 | 10 | 30 | 2.3 |
| 4 | 100 | 31 | 12.79 |
| 4 | 1000 | 31 | 35.48 |
| 4 | 10000 | 30.01 | 35.09 |
| 4 | 100000 | 29.14 | 41.46 |
| 4 | 500000 | 29.15 | 81.54 |
| 4 | 1000000 | 29.1 | 81.19 |
| 8 | 1 | 0 | 0.21 |
| 8 | 10 | 20 | 1.96 |
| 8 | 100 | 28 | 19.9 |
| 8 | 1000 | 28.3 | 59.68 |
| 8 | 10000 | 28.88 | 68.81 |
| 8 | 100000 | 29.06 | 77.28 |
| 8 | 500000 | 29.08 | 98.42 |
| 8 | 1000000 | 29.12 | 124.03 |
| 12 | 1 | 0 | 0.15 |
| 12 | 10 | 50 | 1.79 |
| 12 | 100 | 28 | 15.95 |
| 12 | 1000 | 27.8 | 48.95 |
| 12 | 10000 | 29.14 | 104.78 |
| 12 | 100000 | 29.1 | 101.17 |
| 12 | 500000 | 28.95 | 147.63 |
| 12 | 1000000 | 29.06 | 194.42 |
| 16 | 1 | 0 | 0.11 |
| 16 | 10 | 30 | 1.38 |
| 16 | 100 | 28 | 13.05 |
| 16 | 1000 | 29.1 | 59.07 |
| 16 | 10000 | 29.47 | 79.4 |
| 16 | 100000 | 29.21 | 135.39 |
| 16 | 500000 | 29.26 | 178.08 |
| 16 | 1000000 | 29.21 | 226.85 |
| 20 | 1 | 100 | 0.11 |
| 20 | 10 | 20 | 1.32 |
| 20 | 100 | 26 | 11.32 |
| 20 | 1000 | 30.7 | 58.75 |
| 20 | 10000 | 29.57 | 109.78 |
| 20 | 100000 | 29.24 | 116.58 |
| 20 | 500000 | 29.18 | 174.41 |
| 20 | 1000000 | 29.23 | 212.15 |
| 24 | 1 | 0 | 0.1 |
| 24 | 10 | 20 | 0.9 |
| 24 | 100 | 26 | 10.35 |
| 24 | 1000 | 30.5 | 61.18 |
| 24 | 10000 | 29.14 | 123.65 |
| 24 | 100000 | 28.97 | 138.95 |
| 24 | 500000 | 29 | 177.62 |
| 24 | 1000000 | 29.04 | 229.87 |
| 32 | 1 | 100 | 0.09 |
| 32 | 10 | 30 | 1 |
| 32 | 100 | 40 | 8.74 |
| 32 | 1000 | 29.6 | 0.43 |
| 32 | 10000 | 29.61 | 202.68 |
| 32 | 100000 | 29.06 | 236.96 |
| 32 | 500000 | 29.23 | 224.61 |
| 32 | 1000000 | 29.16 | 264.03 |

**Pivot Table:**

****

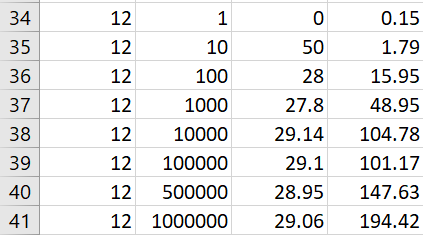
**Graph for Performance vs Number of Trials:**

We can see that initially as the Number of trials were less, the performance was also less. But as the requirement increased, the performance also increased.

**Graph for Performance vs Number of Cores:**

We can see that as the number of cores increased for the number of trials, the performance increased by a good margin. But we can observe a drop in performance for an instance in every case, this is because the GPU isn’t dedicated to only this task. Hence, other task also require the GPU which dips the performance.

**Probability:**

****

We can see in the above example that the probability got stagnated around 29 after performing higher number of numTrials. This justifies that the highest probability of this approximately 29.

**Parallel Fraction:**

To compute the Parallel fraction, the first process is to calculate speedup,

*Speedup(S) = Performance with 32 threads / Performance with 1 thread*

*Speedup(S) = 264.03/21.73 = 12.15048*

Now, Calculating the Parallel Fraction

***Fp(n,S) = n(1-1/S) / (n-1) = 32(1-1/12.15048)/(32-1) = ~*0.94730190932**