Parallel Programming CS-575

Project-1

MonteCarlo Simulation

Submitted by

Akhil Sai Chintala

ONID: chintala

1) Data After Execution:

For the execution of MonteCarlo's program, I have taken the number of threads as 1,2,4,8,12,16,20, and the number of trials are 1,10,100, 1000,100000,500000,1000000.
For the threads and number of trials the below table represents the probability and maximum performance.

Threads	Numtries	Probability	Performance
1	1	0	0.65
1	10	30	3.58
1	100	24	9.22
1	1000	27.9	10.13
1	10000	28.75	9.58
1	100000	29.09	21.79
1	500000	29.11	21.81
1	1000000	29.13	21.67
2	1	0	0.24
2	10	30	2.74
2	100	28	7.62
2	1000	29.6	18.41
2	10000	28.62	17
2	100000	29.1	28.56
2	500000	29	41.49
2	1000000	29.04	41.57
4	1	100	0.26
4	10	10	2.45
4	100	33	11.26
4	1000	30.4	21.74
4	10000	30.03	33
4	100000	29.37	44.03
4	500000	29.16	81.58
4	1000000	29.15	81.13
8	1	0	0.23
8	10	40	1.94
8	100	24	19.25
8	1000	28.7	59.58
8	10000	29.33	68.78
8	100000	29.09	72.8
8	500000	29.22	113.9
8	1000000	29.15	140.21

12	1	100	0.16
12	10	20	1.67
12	100	19	14.95
12	1000	29.4	76.75
12	10000	29.42	69.43
12	100000	29.19	100.6
12	500000	29.02	147.52
12	1000000	29.03	187.02
16	1	0	0.12
16	10	20	1.31
16	100	30	11.29
16	1000	30.3	58.08
16	10000	29.47	90.73
16	100000	29.1	92.94
16	500000	29.17	180.53
16	1000000	29.05	227.34
20	1	0	0.12
20	10	20	1.25
20	100	27	10.43
20	1000	30	65.48
20	10000	29.7	103.76
20	100000	29.14	114.96
20	500000	29.2	157.15
20	1000000	29.08	213.33

Converting the above table into pivot table is represented as below, where columns are represented with the number of trials and the rows are represented with the number of threads with the corresponding values of maximum performance.

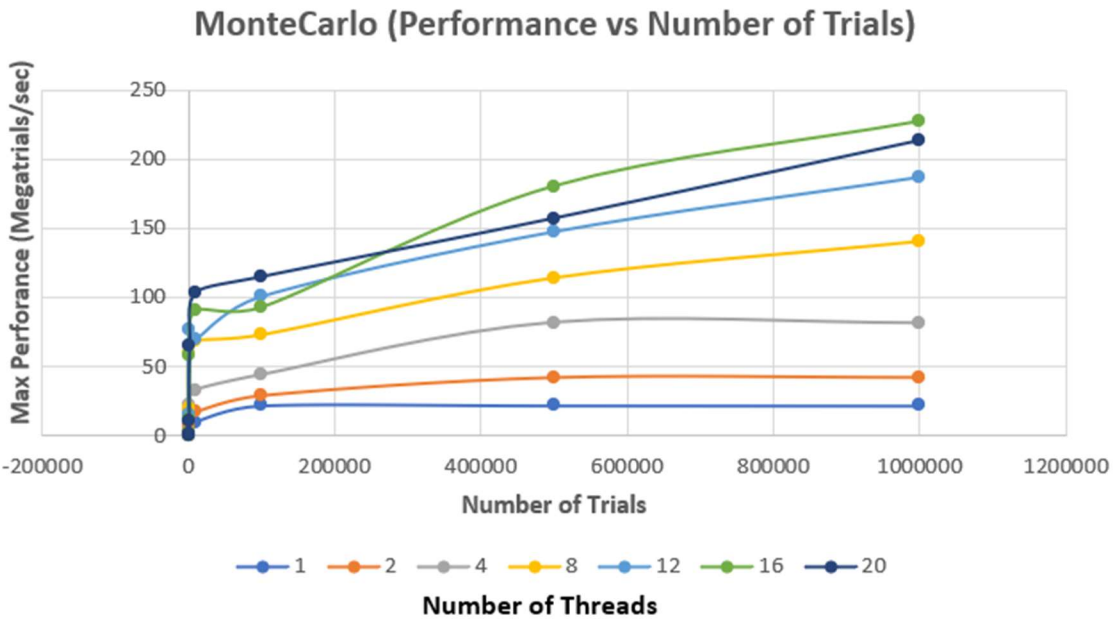
Pivot Table:

Threads\NumTries	1	10	100	1000	10000	100000	500000	1000000
1	0.65	3.58	9.22	10.13	9.58	21.79	21.81	21.67
2	0.24	2.74	7.62	18.41	17	28.56	41.49	41.57
4	0.26	2.45	11.26	21.74	33	44.03	81.58	81.13
8	0.23	1.94	19.25	59.58	68.78	72.8	113.9	140.21
12	0.16	1.67	14.95	76.75	69.43	100.6	147.52	187.02
16	0.12	1.31	11.29	58.08	90.73	92.94	180.53	227.34
20	0.12	1.25	10.43	65.48	103.76	114.96	157.15	213.33

2) Graphs:

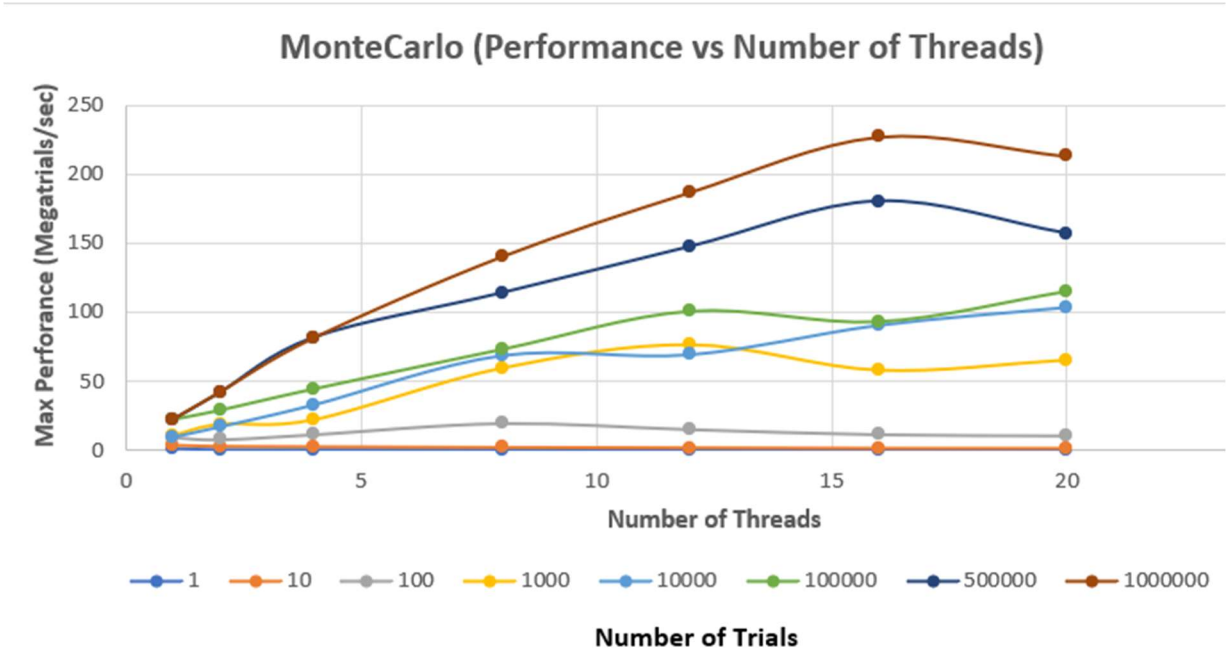
a) Performance versus the number of Monte Carlo trials, with the colored lines being the number of OpenMP threads.

By using the above pivot table mentioned in (1), the graph for performance vs number of trials is represented as below,
In the initial stage the performance is very low because of the lack of the amount of work done in the initial stages, where the number of trials were also less.



b) Performance versus the number of OpenMP threads, with the colored lines being the number of Monte Carlo Trials.

By using the above pivot table mentioned in (1), the graph for performance vs number of threads is represented as below,
Here, the performance was increased by increase in threads because, increase in data leads to increase in performance.



3) Probability:

After going through the recorded observations of the probability in table (1). (Output of the program).
By selecting the probabilities of the thread 20 for all number of trials. Initially, the probability for 1 trial is 0, then the probability for 10 trials is 20. Similarly, as the number of trials were increasing, the probability gets stabilized at 29, and for the maximum number of trials the probability of snowball hitting the truck is 29.08%.

Hence the final probability is 29.08%.

4) Parallel Function (Fp):

To calculate the parallel fraction (Fp), speedup needs to be calculated first.
The Speedup formula is derived as follows,

$$\text{Speedup(S)} = \frac{\text{Performance of 20 threads (Maximum)}}{\text{Performance of 1 thread}}$$

(I am considering the performance for the maximum numtrials which is 1000000).

= 213.33/21.67
= 9.845

Speedup = 9.845

Parallel Fraction Fp is calculated using speedup as below,

Fp = (n/n-1) * (1-1/s)

n-> maximum number of threads
s-> Speedup

Fp = (20/20-1) * (1-1/9.845)

= 0.945

Parallel Fraction (Fp) = 0.945