Shawn Berg
bergsha@oregonstate.edu
April 16th, 2019
CS 475

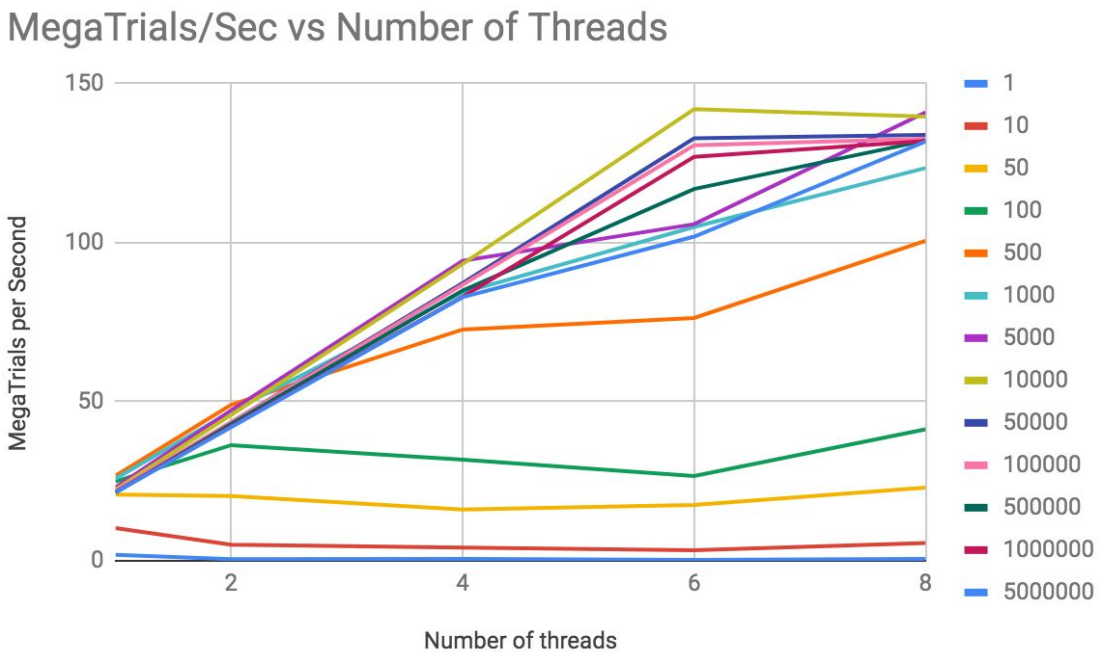
Project 1
Monte Carlo Simulation
Writeup

Table (Performance as a function of Threads vs Number of Trials)

	1	10	50	100	500	1000	5000	10000	50000	100000	500000	1000000	5000000
1	1.87	10.25	20.83	24.91	26.73	25.77	23.02	22.36	21.72	21.92	21.37	21.55	21.45
2	0.49	5.05	20.33	36.34	49.05	46.93	47.27	45.88	43.56	43.51	42.98	42.29	41.99
4	0.56	4.14	16.07	31.81	72.71	84.34	94.31	93.33	87.29	86.78	84.93	83.01	82.86
6	0.31	3.33	17.56	26.65	76.35	104.95	105.82	142	132.83	130.64	116.91	127.01	101.93
8	0.54	5.57	22.99	41.34	100.58	123.49	141.02	139.62	133.92	132.69	132.11	131.98	131.84

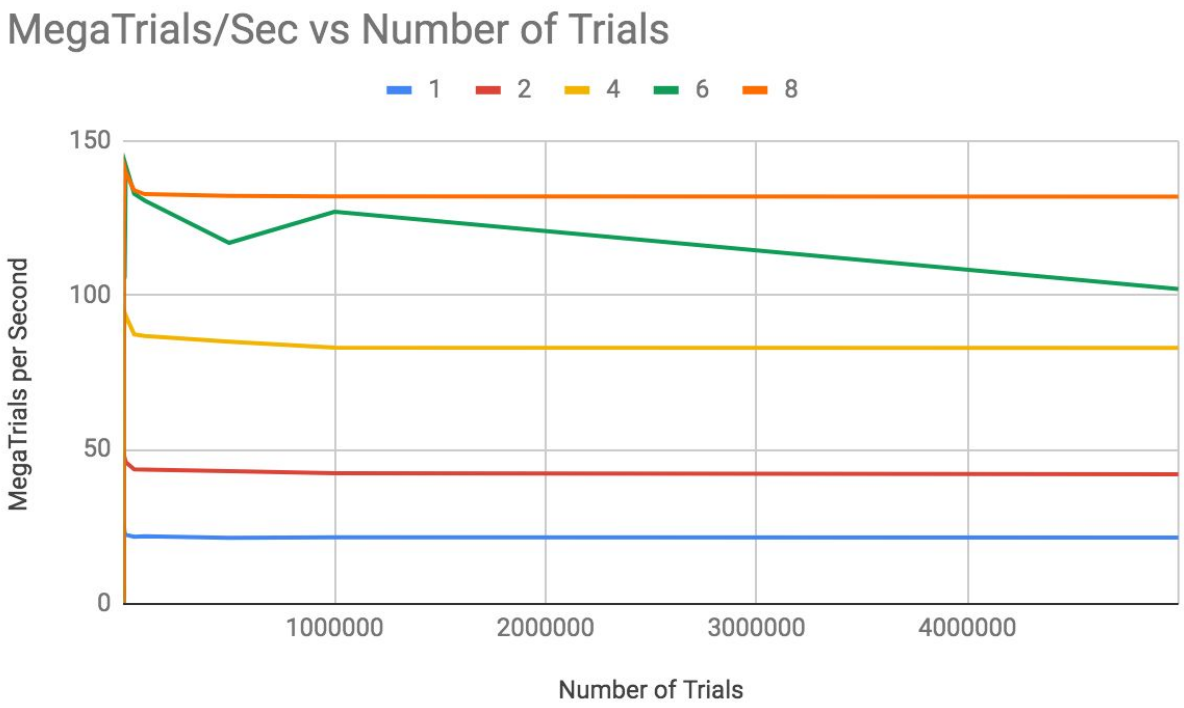
Units = MegaTrials/Sec

Graph 1 (MegaTrials per Second vs Number of Threads)



Each line represents number of trials

Graph 2 (MegaTrials per Second vs Number of Trials)



Each line represents number of threads.

Estimated Probability from run with most trials: 0.1902

Speedup = P_8 / P_1 = 131.84/21.45 = 6.15

Parallel Fraction = (8/7) * (1 - (1/6.15)) = 0.957

Additional Commentary

This assignment was much simpler from an environment perspective thanks to the work I had completed on the previous assignment. I spent the majority of this assignment formatting my compilation script and the main C++ code to format the output to an easily importable .csv file. This extra effort upfront made creating the necessary tables and charts simple and will surely pay dividends for future assignments.

The rest of my time was spent reading and understanding the code for the Monte Carlo simulation, finishing the implementation, and printing out the necessary information.

Analyzing the results from the Monte Carlo simulation, you see a clear normalization of the probability as the number of trials increases. This would be expected behavior, as the probability is calculated by taking the number of hits on the plate and dividing by the number of trials. As the number of trials increases, you're going to see a more accurate number for

probability. Also as expected, there is a general increase in performance with an increase in threads. However, as you can see in the MegaTrials/Sec vs Number of Threads chart, this increase in performance doesn't have an effect until > 100 trials are performed. Anything less than that and the increase in thread count has minimal effect on the performance of the algorithm.