Project1

OpenMP: Monte Carlo Simulation

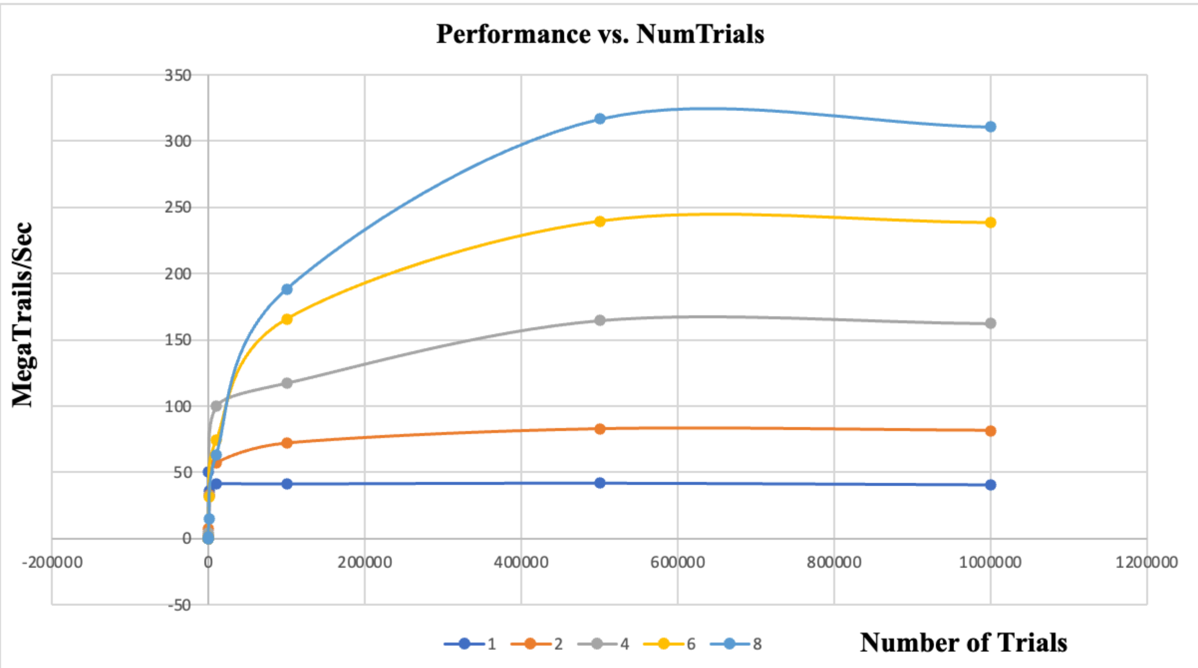
Name: Yi Chiun Chang

Email: changyic@oregonstate.edu

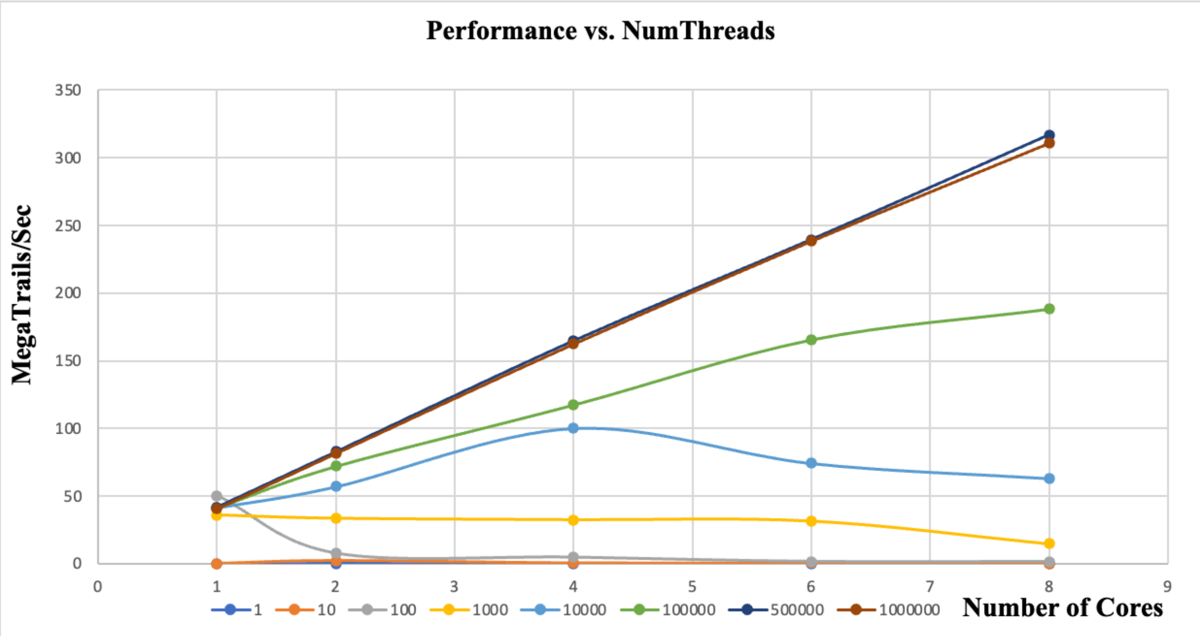
Data table of the performance numbers as a function of Threads and Trials.

	1	10	100	1000	10000	100000	500000	1000000
1	0.0813	0.22	50.03	35.72	41.15	41.17	41.74	40.49
2	0.17	2	7.69	33.33	56.82	71.89	82.64	81.43
4	0.05	0.32	4.76	32.26	100	117.37	164.58	162.26
6	0.02	0.24	1.61	31.25	74.07	165.56	239.35	238.27
8	0.01	0.18	1.59	14.71	62.89	188.32	316.66	310.66

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



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



3. Probability

	1000000
1	57.01
2	56.97
4	57.00
6	56.99
8	56.99
Avg	56.992

This consistency suggests that the probability does not significantly depend on the number of threads, which indicates that the operation's outcome is not being impacted by the level of parallelism in use. Given that the average of the probabilities is 56.992, we can say that the 'actual' probability, assuming these measurements are accurate and precise, is very close to this average. In a practical sense, the difference between 56.992 and any of the individual measurements is negligible. Therefore, you could reliably expect that the probability of whatever event or success rate these numbers are tracking is approximately 56.992%.

4. Computation of the Parallel Fraction

Based on the maximum of trials, the highest and lowest performances are 310.66 and 40.49.

S = 310.66/40.49 = 7.67

Float Fp = (8./7.)*(1. - (1./S)) = 0.993

5. Commentary

We know that Amdahl's Law can also give us the maximum possible speedup.

max Speedup = $\frac{1}{1-Fp} = \frac{1}{1-0.993}$ = 143.

Given the parallel fraction of 0.993, the theoretical maximum speedup achievable if you could utilize hundreds of cores approaches approximately 143. This means that under ideal conditions with infinite resources, the program could potentially run about 143 times faster than the single-threaded version.

When interpreting the performance behavior as the number of threads increases, you would typically expect performance to improve due to parallel processing gains. However, it's common to see diminishing returns as overhead from thread management and synchronization begins to outweigh the benefits of additional threads, a phenomenon predicted by Amdahl's Law. The variability in performance might stabilize as trials increase, providing a more accurate measure of average performance. At very high thread counts, performance could degrade, indicative of contention, synchronization issues, or hitting the physical limits of the hardware. Overall, the data reflects both the power and the constraints of parallel computing, where adding more processing units leads to faster computations up to a point, beyond which the architecture and problem's inherent parallelizability limit improvements.