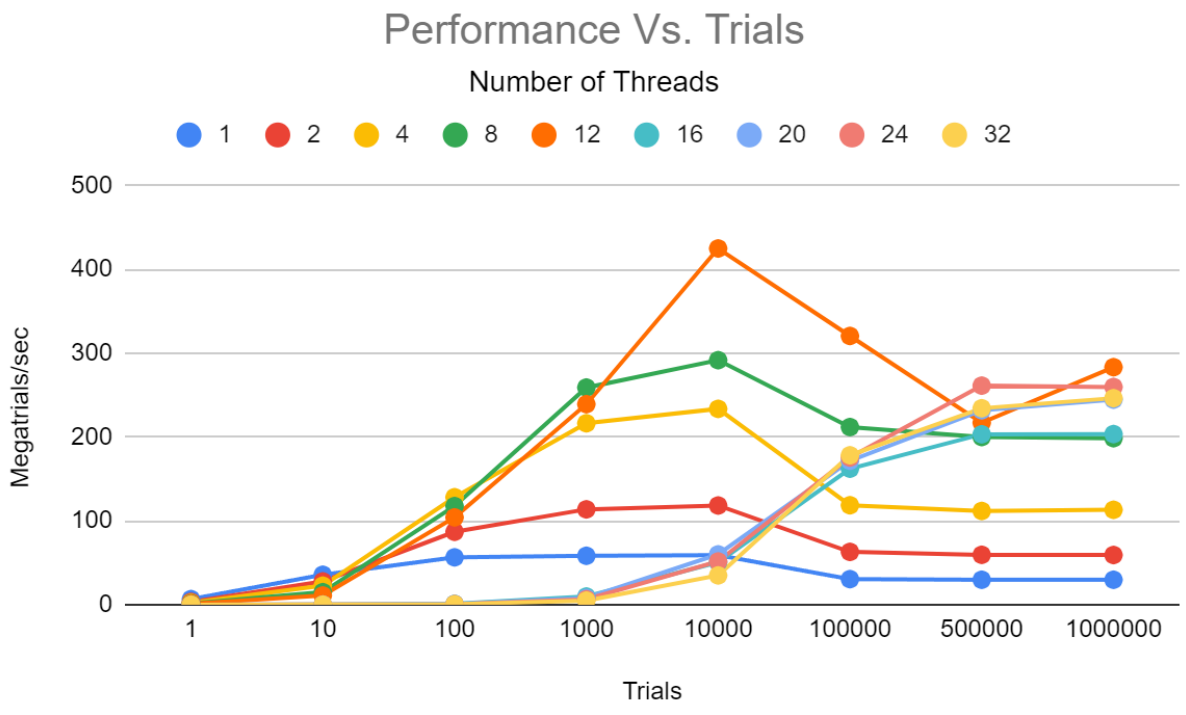
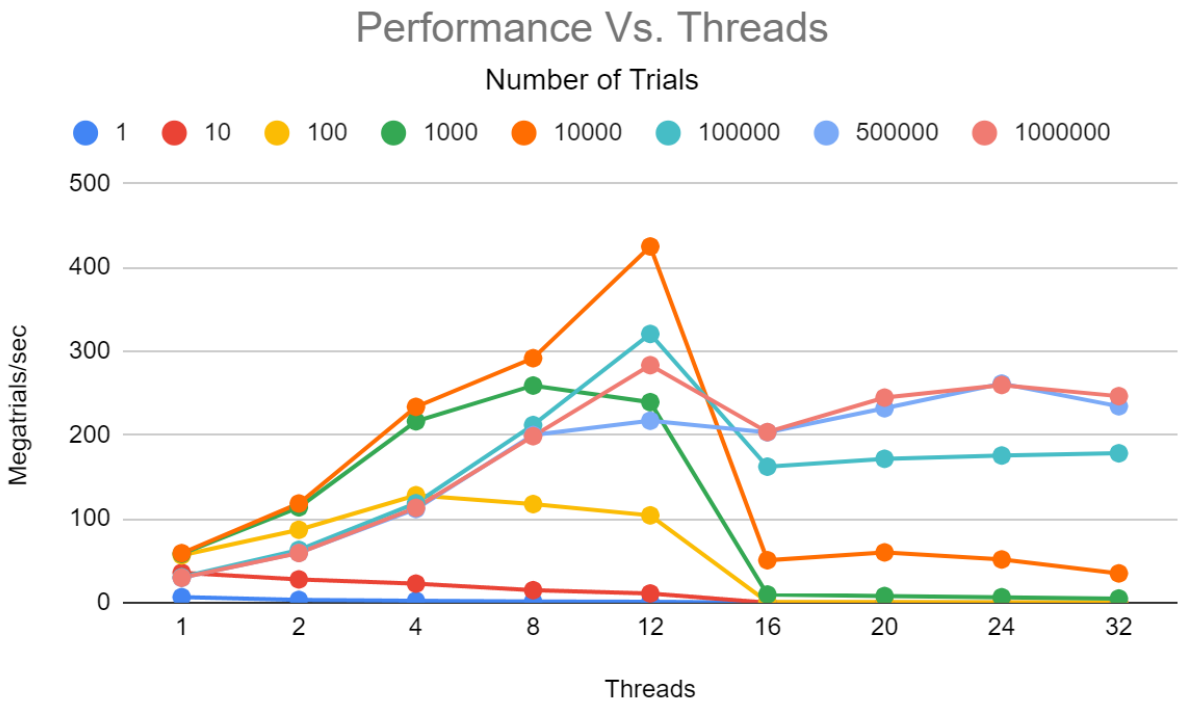
CS 475 Parallel Programming: Project 1 - Monte Carlo Problem

Conner Rhea
rheac@oregonstate.edu

SUM of Performance	*Trials*								
Threads	1	10	100	1000	10000	100000	500000	1000000	Grand Total
1	6.666666431	35.71428638	56.49717568	58.34305714	59.10864168	30.50208878	29.73829114	29.74125495	306.3114622
2	3.333333215	27.77777767	86.95652153	113.765643	118.3431952	63.00077743	59.49208041	59.27218499	531.9415135
4	2.272727231	22.72727231	128.2051293	216.4502178	233.6448597	118.654787	111.8229012	113.2155363	946.9934308
8	1.470588298	14.92537396	117.6470599	259.067358	291.7152857	211.9272666	200.0230427	198.6390839	1295.415059
12	1.136363689	10.9890112	104.1666642	239.2344487	424.9893759	320.6155819	217.0654231	283.3587648	1601.555633
16	0.009533797312	0.05403915677	0.9783778499	9.751245722	50.66112772	162.2691721	203.2931043	203.5658218	630.5824225
20	0.007480550567	0.07050694493	0.7580351728	7.934555785	60.0132029	171.676767	231.9622662	244.6534056	717.0762202
24	0.006590611015	0.06942564166	0.5608839531	6.525242903	51.58069025	175.6018754	261.3556413	259.7257556	755.4261056
32	0.002741754174	0.05096060746	0.4809981674	4.961794184	34.95525727	178.2528017	234.3585583	246.4215889	699.4847009





Link to Full Spreadsheet: CS 475 Project 1 Sheet

Parallel Fraction:

Speedup = 1 thread exec / N thread exec = N thread perf / 1 thread perf

S_4 = 113.22 / 29.74 = 3.81

S_{12} = 283.36 / 29.74 = 9.53

S_{20} = 244.65 / 29.74 = 8.23

S_{32} = 246.42 / 29.74 = 8.29

Max Speedup = 9.53

Avg. Speedup = 7.47

$Fp = (n / (n - 1)) * (1 - (1 / S))$

$Fp_4 = (4 / 3) * (1 - (1 / 3.81)) = 0.9834$

$Fp_{12} = (12 / 11) * (1 - (1 / 9.53)) = 0.9765$

$Fp_{20} = (20 / 19) * (1 - (1 / 8.23)) = 0.9247$

$Fp_{32} = (32 / 31) * (1 - (1 / 8.29)) = 0.9077$

Max Parallel Fraction = 0.9834
Avg. Parallel Fraction = 0.9481

Estimated Probability:
Disregarding the single trial data, almost every runs probability falls squarely into the 20% range of success, sugessting that the overall probability lies close to 20% chance for the top piece to fit into the bottom.

Graph Notes:
It seems that performance begins to fall off heavily after 12 cores. My CPU is a 6 core CPU from AMD, which suggests that this task begins to receive greatly diminishing returns after 2 threads per core. I would hazard a guess that each core handling more than 2 threads is causing the swapping between threads to negatively affect performance as it scales upward, though it slightly levels out as it climbs again.