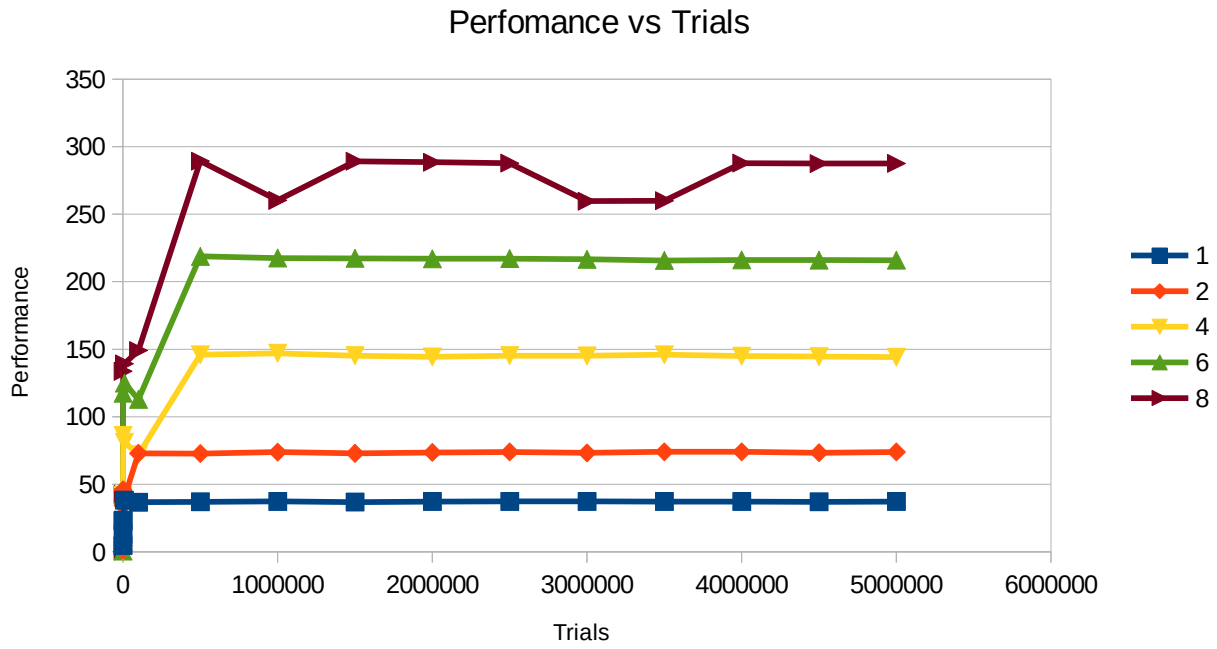
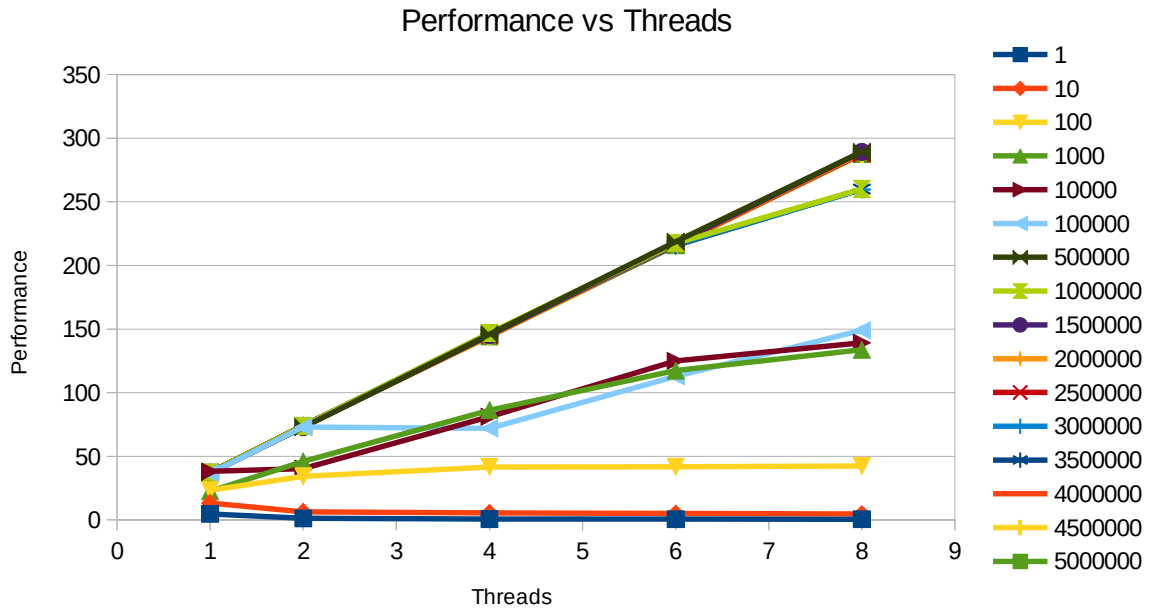
Jose Garay
garayj@oregonstate.edu
Project 1
OpenMP: Monte Carlo Simulation

	1	10	100	1000	10000	100000	500000	1000000	1500000	2000000	2500000	3000000	3500000	4000000	4500000	5000000
1	4.76	13.33	23.36	22.56	38.17	36.75	37.01	37.31	36.8	37.19	37.32	37.3	37.25	37.24	37.02	37.24
2	1.25	6.41	34.25	45.91	40.34	72.96	72.79	73.84	72.94	73.57	73.96	73.26	74.02	74.02	73.36	73.96
4	0.73	5.49	41.67	86.28	81.1	71.9	145.92	146.95	145.22	144.3	145.25	145.14	145.92	144.91	144.51	144.16
6	0.61	5.05	41.84	117.37	125.02	112.88	218.74	217.47	217.26	217	217.09	216.64	215.69	216.08	216.02	215.96
8	0.56	4.59	42.37	133.69	139.18	149.09	289.29	260.19	289.22	288.45	287.79	259.65	260.01	287.76	287.51	287.62
Speedup	11.76%	34.43%	181.38%	592.60%	364.63%	405.69%	781.65%	697.37%	785.92%	775.61%	771.14%	696.11%	698.01%	772.72%	776.63%	772.34%
Fp	-857.14%	-217.62%	51.28%	95.00%	82.94%	86.11%	99.66%	97.90%	99.74%	99.55%	99.47%	97.87%	97.91%	99.50%	99.57%	99.49%





I had my program print out the probability print out the probability from each run, and it looks like there's about a 13% probability that the laser would hit the circle and bounce in such a way that would hit the infinite plate.

For 5000000 trials:
So my Speedup is (287.62 / 37.24) = 7.72.
Using the formula (n/(n-1))(1-1/Speedup), my Fp should be:
(8/7)(1-(1/7.72)) * 100 = 99.48%

I also went ahead and calculated the Fp and speedup for 8 v 1 threads in x trials. It's pretty amazing to see Fp and speedup get better and better when trials go for longer. Another thing I really found interesting was looking at the very low end of these trials where one thread is much faster than the rest presumable due to overhead of setting up OpenMP and mostly sequential code.