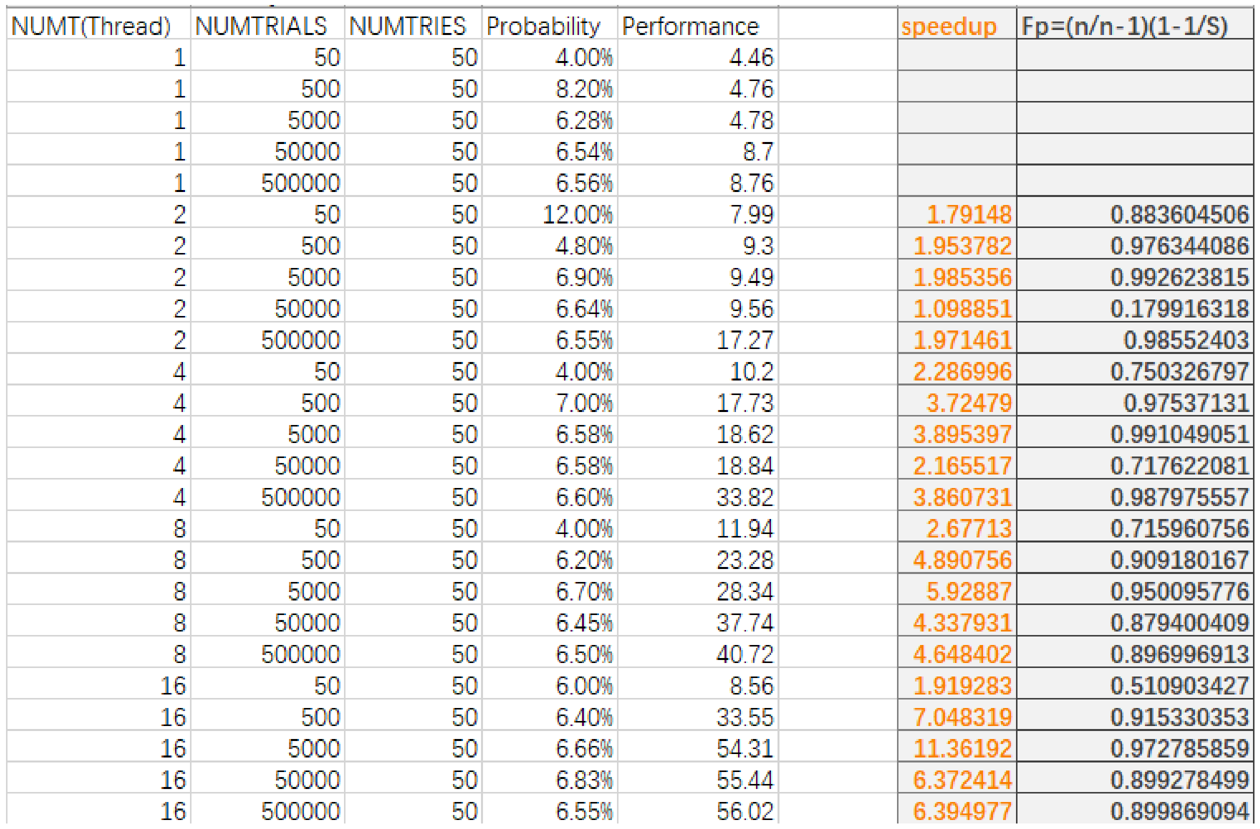
CS 575
Project #1
OpenMP: Monte Carlo Simulation

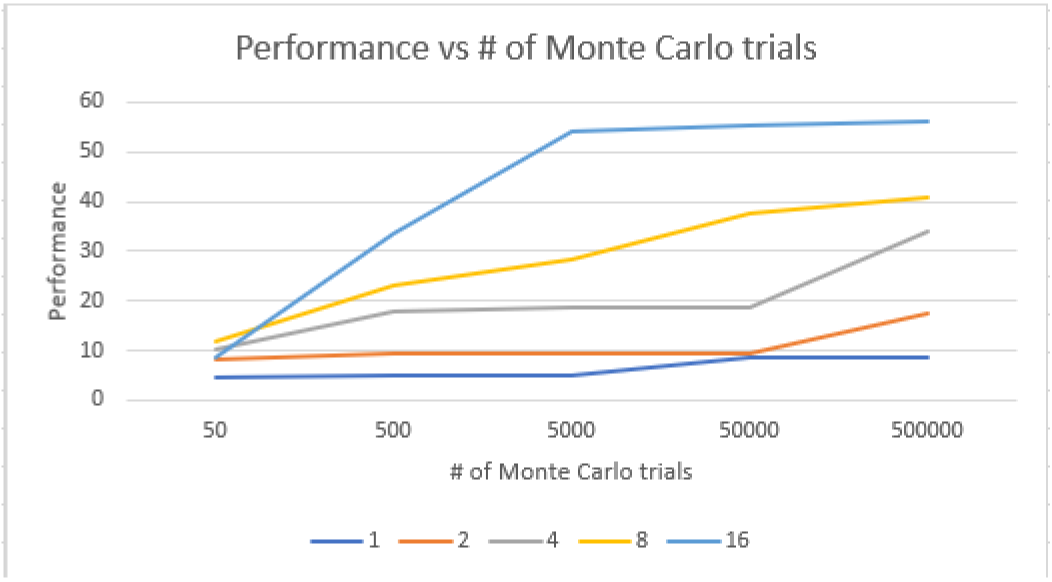
Xuming Wu
wuxum@oregonstate.edu

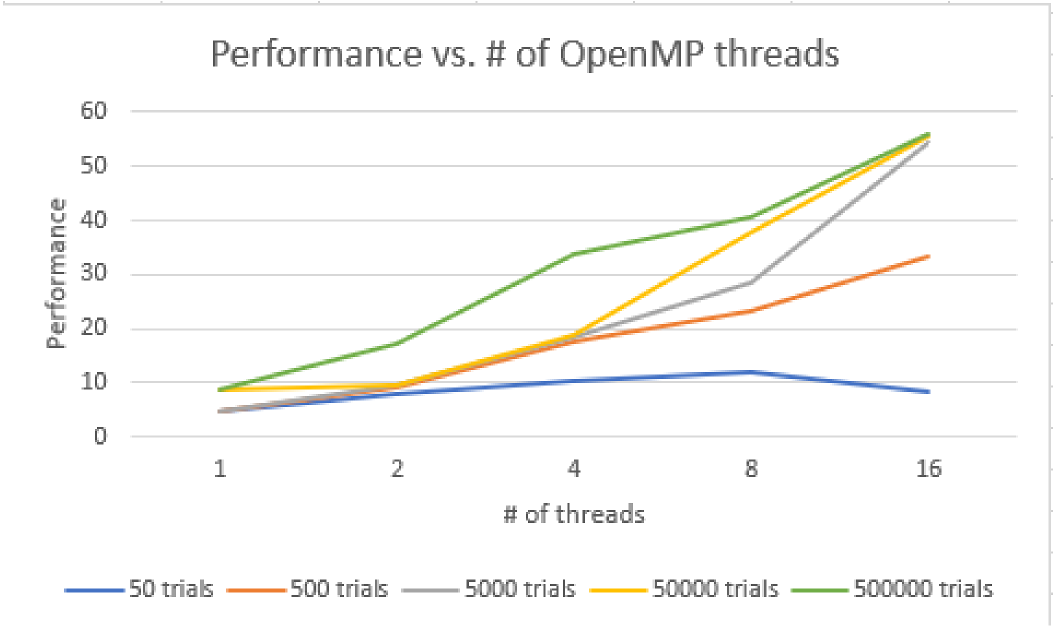
Code was run on the flip server. Got result as:



NUMT(Thread)	NUMTRIALS	NUMTRIES	Probability	Performance		speedup	Fp=(n/n-1)(1-1/S)
1	50	50	4.00%	4.46			
1	500	50	8.20%	4.76			
1	5000	50	6.28%	4.78			
1	50000	50	6.54%	8.7			
1	500000	50	6.56%	8.76			
2	50	50	12.00%	7.99		1.79148	0.883604506
2	500	50	4.80%	9.3		1.953782	0.976344086
2	5000	50	6.90%	9.49		1.985356	0.992623815
2	50000	50	6.64%	9.56		1.098851	0.179916318
2	500000	50	6.55%	17.27		1.971461	0.98552403
4	50	50	4.00%	10.2		2.286996	0.750326797
4	500	50	7.00%	17.73		3.72479	0.97537131
4	5000	50	6.58%	18.62		3.895397	0.991049051
4	50000	50	6.58%	18.84		2.165517	0.717622081
4	500000	50	6.60%	33.82		3.860731	0.987975557
8	50	50	4.00%	11.94		2.67713	0.715960756
8	500	50	6.20%	23.28		4.890756	0.909180167
8	5000	50	6.70%	28.34		5.92887	0.950095776
8	50000	50	6.45%	37.74		4.337931	0.879400409
8	500000	50	6.50%	40.72		4.648402	0.896996913
16	50	50	6.00%	8.56		1.919283	0.510903427
16	500	50	6.40%	33.55		7.048319	0.915330353
16	5000	50	6.66%	54.31		11.36192	0.972785859
16	50000	50	6.83%	55.44		6.372414	0.899278499
16	500000	50	6.55%	56.02		6.394977	0.899869094

We can get the Performance vs. the number of Monte Carlo trials graph and the Performance vs. the number OpenMP threads graph as:





From the results, we can estimate that the actual probability is about 6.55%.
Pick the parallel fraction of cases with maximum Monte Carlo trials, we have Fp:

2 threads:	Fp = 0.98552403
4 threads:	Fp = 0.987975557
8 threads:	Fp = 0.896996913
16 threads:	Fp = 0.899869094