OpenMP: Monte Carlo Simulation

Rui Gao

gaorui@oregonstate.edu

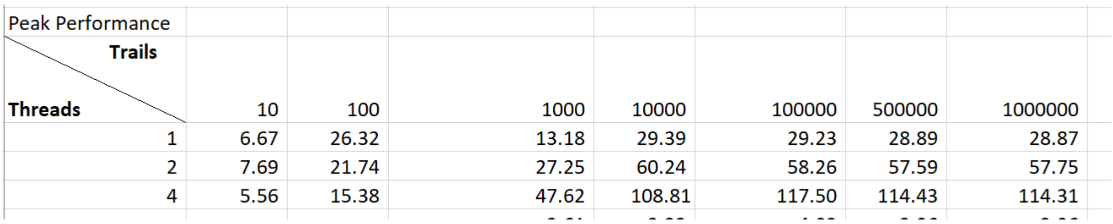
Project 1

CS 575

The probability is around 0.3%. That’s too amateur… I don’t know, maybe I am wrong.

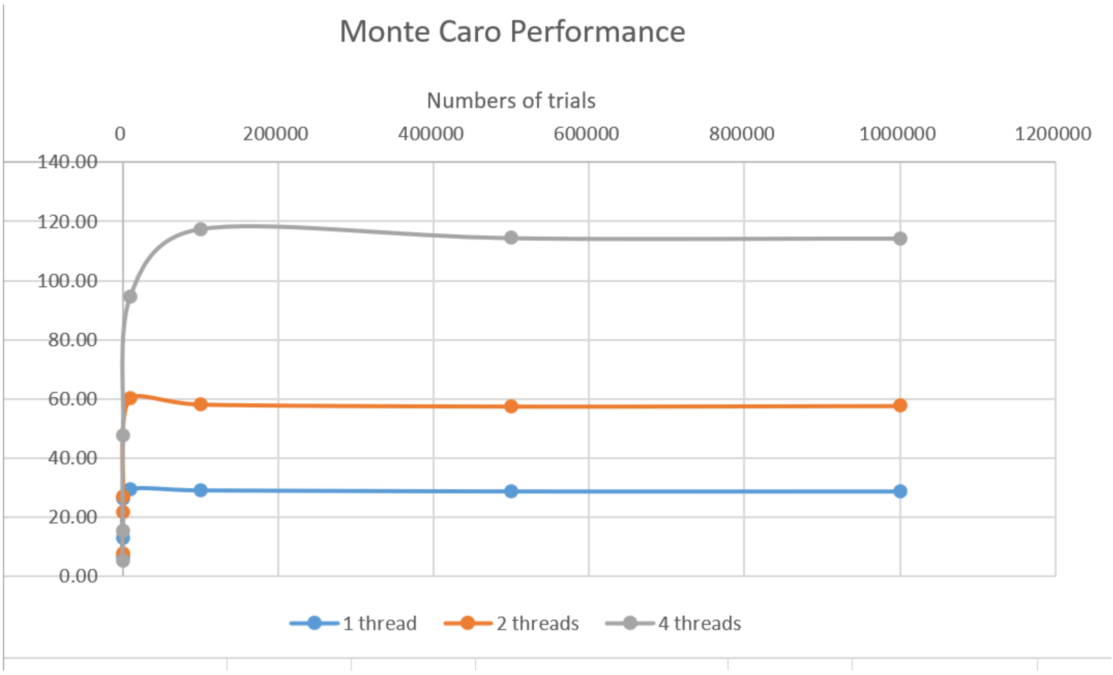
Threads	Trials							
	10		100		1,000		10,000	
	Pb.		Pb.		Pb.		Pb.	
1	0		0		0.1		0.25	
2	0.1		0		0.1		0.25	
4	0.1		0		0.1		0.25	
	100,000				500,000		1,000,000	
	Pb.				Pb.		Pb.	
1	0.29				0.31		0.3	
2	0.29				0.31		0.3	
4	0.29				0.31		0.3	

Pb. = the probability of hitting the castle (%), Peak Performance (MegaMults/Sec).



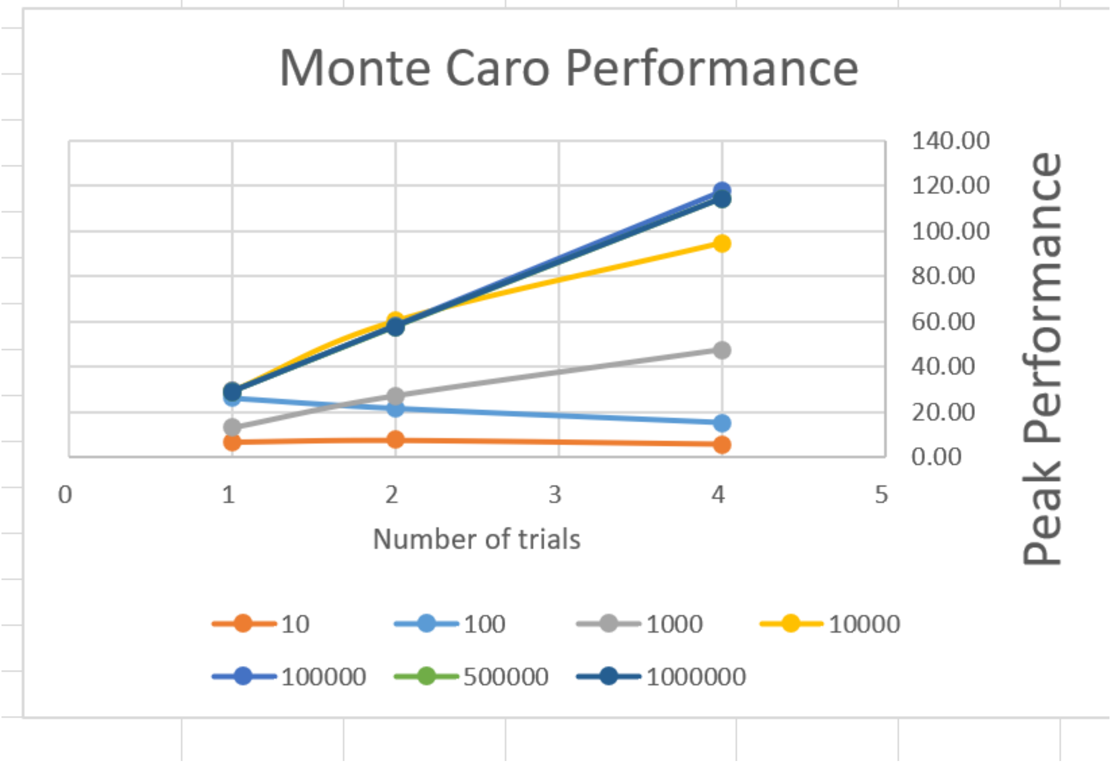
Peak Performance							
Trails Threads	10	100	1000	10000	100000	500000	1000000
1	6.67	26.32	13.18	29.39	29.23	28.89	28.87
2	7.69	21.74	27.25	60.24	58.26	57.59	57.75
4	5.56	15.38	47.62	108.81	117.50	114.43	114.31

Graph of performance vs. number of trials

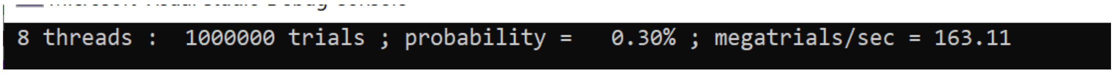


After 100,000 tries, the performance doesn't increase with the trials.

Graph of performance vs. number of threads

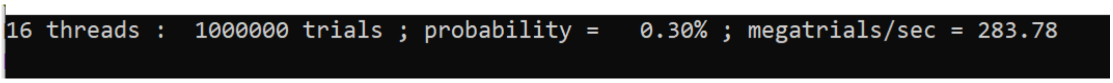


I also tried 8 threads for one time, they are almost 1.4 times.



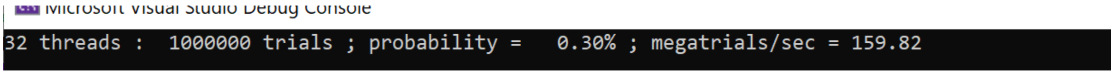
```
8 threads :  1000000 trials ; probability =   0.30% ; megatrials/sec = 163.11
```

16 threads, they are almost 1.7 times.



```
16 threads :  1000000 trials ; probability =   0.30% ; megatrials/sec = 283.78
```

But when it's 32 threads,



```
32 threads :  1000000 trials ; probability =   0.30% ; megatrials/sec = 159.82
```

So, it will keep increasing until we use 32 threads, which is more than 16 that is the physical/real threads the 4900hs has.

Fp, the Parallel Fraction.

S = (Performance with four threads) / (Performance with one thread)

Fp = (4./3.)*(1. - (1./S))

Fp_1,000 = (4 / 3) * (1 – 1/3.61) = 0.964

Fp_10,000 = (4 / 3) * (1 – 1/3.22) =0.919

Fp_100,000 = (4 / 3) * (1 – 1/4.01) = 1.000

Fp_500,000 = (4 / 3) * (1 – 1/3.96) = 0.997

Fp_1,000,000 = (4 / 3) * (1 – 1/3.96) = 0.997

As the result, the Fp = Avg (Fp above) = 0.975