Project #1 OpenMP: Monte Carlo Simulation

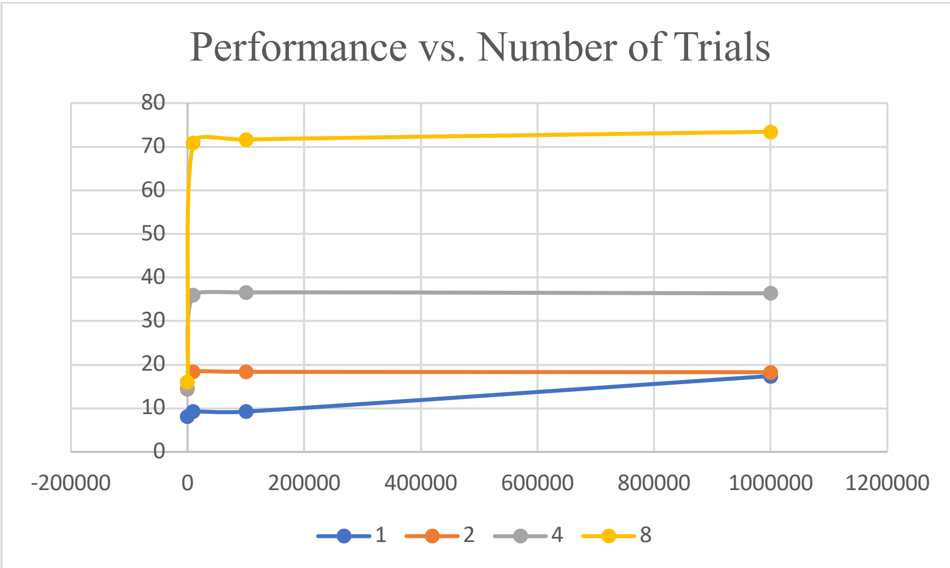
Qingxiao Yuan

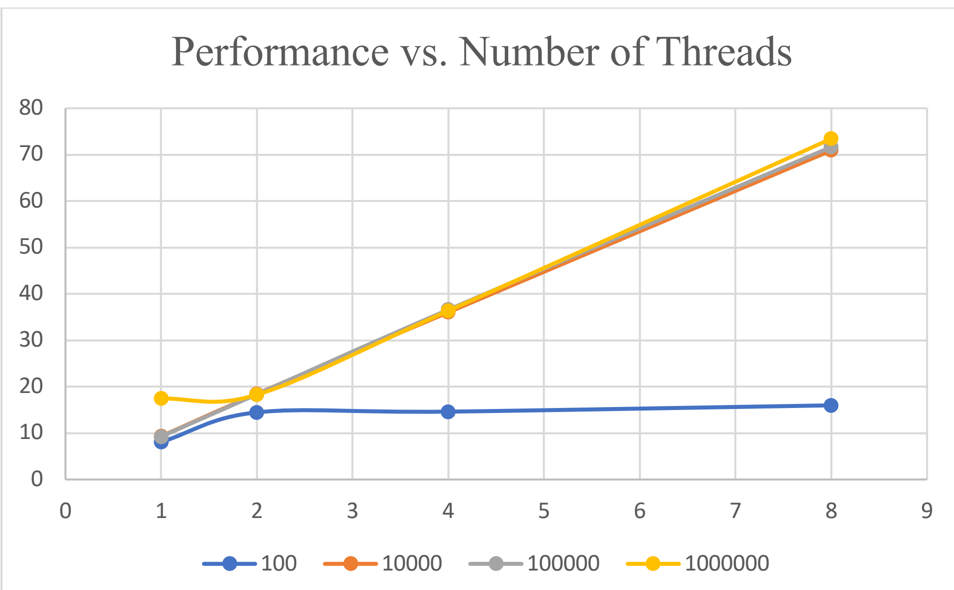
yuanqi@oregonstate.edu

To begin with, I run program 1 on flip2 of OSU.

1.The tables and related two graphs are showing below,

number of threads	number of trials	probability	MegaTrials PerSecond
1	100	0.2	8.0775
1	10000	0.1858	9.3101
1	100000	0.1907	9.237
1	1000000	0.19	17.4084
2	100	0.16	14.4636
2	10000	0.1895	18.4564
2	100000	0.1912	18.3616
2	1000000	0.1898	18.2784
4	100	0.14	14.6242
4	10000	0.1878	36.0372
4	100000	0.1917	36.5908
4	1000000	0.1903	36.3662
8	100	0.16	15.9795
8	10000	0.1935	70.9195
8	100000	0.1915	71.6563
8	1000000	0.1898	73.4394





In the ‘Performance vs. Numbers of Trials’ chart, the vertical axis indicates the number of trials, the horizontal axis indicates the performance using ‘Mega Trials Per Second’ as a unit and the lines indicates the number of threads.

In the ‘Performance vs. Number of Threads’ chart, the vertical axis indicates the number of threads, the horizontal axis indicates the performance using 'Mega Trials Per Second' as a unit and the lines indicates the number of trials.

When the number of threads increases, the performance increases and the speedup is less than n (the number of threads). The probability calculation process of each trial has nothing to do with each other, so the speed will increase using parallel programming. But there are some fractions of the total operation that are inherently sequential and can't be parallelized. So the speedup for n threads will always be less than n.

When the number of trials is small, the performance will increase rapidly. When the number of trials increases after 100,000, the performance will keep stable. As the number of trials increases, the proportion of overhead in parallel programming is getting smaller and the speed will be faster. But there is still an upper bound of the computing capacity of CPU and the speed will not increase without limit.

2. Explanation about the probability(Chose the largest one). When the number of trials is 1,000,000 and the number of threads is 8, the probability is 0.1898. I think the actual probability is the probability of $x \in (0.0, 1.0)$ and $y > x$ and $r^2 < x^2 + y^2$. The possibility is not the largest. But I think it nearly 0.19 form the total threads in the maximum tries. I'm not sure the reason. But I guess it's because of the fluctuate of server flip2. The load is following:

15:28:55 up 20 days, 7:01, 69 users, load average: 1.00, 0.88, 0.95

3. Compute Fp, the parallel Fraction.

The law is:

$$F_p = \frac{n}{n-1}\left(1 - \frac{1}{Speedup}\right) = \frac{n}{n-1}\left(1 - \frac{MegaTrialsPerSec_1}{MegaTrialsPerSec_n}\right)$$

To see the result clear, I rearranged the table. It shows below:

	100	10000	100000	1000000
1	8.0775	9.3101	9.237	17.4084
2	14.4636	18.4564	18.3616	18.2784
4	14.6242	36.0372	36.5908	36.3662
8	15.9795	70.9195	71.6563	73.4394

When the number of trials is 100,

$$F_2 = \frac{2}{2-1}\left(1 - \frac{8.0775}{14.4636}\right) = 0.8831$$

$$F_4 = \frac{4}{4-1}\left(1 - \frac{8.0775}{14.6242}\right) = 0.5969$$

$$F_8 = \frac{8}{8-1}\left(1 - \frac{8.0775}{15.9795}\right) = 0.5652$$

When the number of trials is 10,000,

$$F_2 = \frac{2}{2-1}\left(1 - \frac{9.3101}{18.4564}\right) = 0.9911$$

$$F_4 = \frac{4}{4-1}\left(1 - \frac{9.3101}{36.0372}\right) = 0.9889$$

$$F_8 = \frac{8}{8-1}\left(1 - \frac{9.3101}{70.9195}\right) = 0.9928$$

When the number of trials is 100,000,

$$F_2 = \frac{2}{2-1}\left(1 - \frac{9.2370}{18.3616}\right) = 0.9939$$

$$F_4 = \frac{4}{4-1}\left(1 - \frac{9.2370}{36.5908}\right) = 0.9967$$

$$F_8 = \frac{8}{8-1}\left(1 - \frac{9.2370}{71.6563}\right) = 0.9955$$

When the number of trials is 1,000,000,

$$F_2 = \frac{2}{2-1}\left(1 - \frac{17.4084}{18.2784}\right) = 0.0952$$

$$F_4 = \frac{4}{4-1}\left(1 - \frac{17.4084}{36.3662}\right) = 0.6951$$

$$F_8 = \frac{8}{8-1}\left(1 - \frac{17.4084}{73.4394}\right) = 0.8719$$