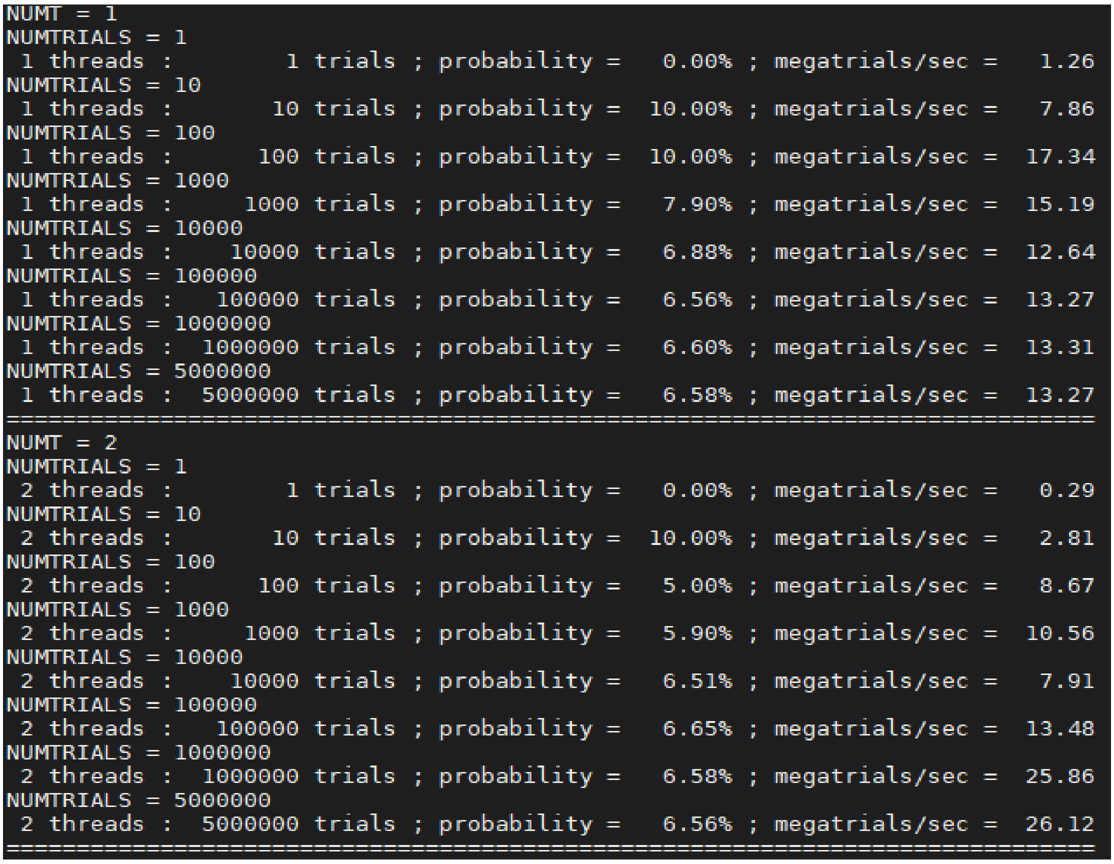
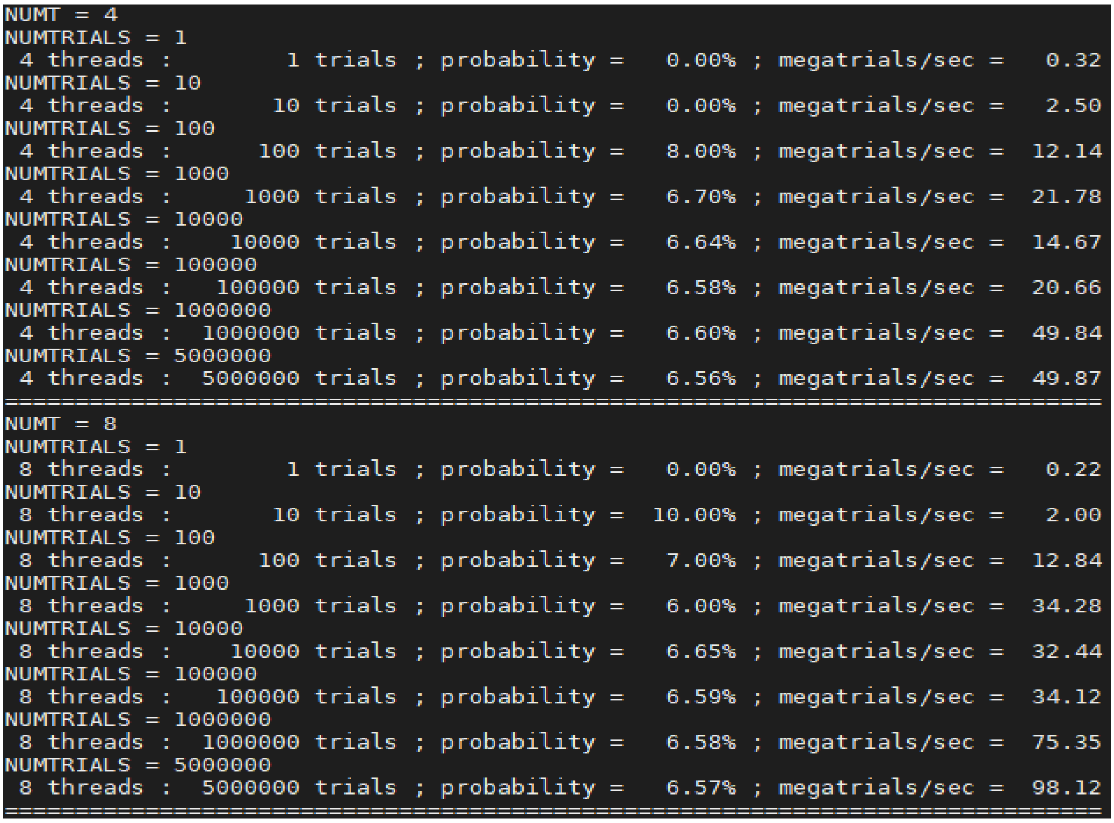
Name: Wei-Chen Kao

Project Name: OpenMP: Monte Carlo Simulation



```
NUMT = 1
NUMTRIALS = 1
 1 threads :        1 trials ; probability =   0.00% ; megatrials/sec =   1.26
NUMTRIALS = 10
 1 threads :       10 trials ; probability =  10.00% ; megatrials/sec =   7.86
NUMTRIALS = 100
 1 threads :      100 trials ; probability =  10.00% ; megatrials/sec =  17.34
NUMTRIALS = 1000
 1 threads :     1000 trials ; probability =   7.90% ; megatrials/sec =  15.19
NUMTRIALS = 10000
 1 threads :    10000 trials ; probability =   6.88% ; megatrials/sec =  12.64
NUMTRIALS = 100000
 1 threads :   100000 trials ; probability =   6.56% ; megatrials/sec =  13.27
NUMTRIALS = 1000000
 1 threads :  1000000 trials ; probability =   6.60% ; megatrials/sec =  13.31
NUMTRIALS = 5000000
 1 threads :  5000000 trials ; probability =   6.58% ; megatrials/sec =  13.27
==============================================================================
NUMT = 2
NUMTRIALS = 1
 2 threads :        1 trials ; probability =   0.00% ; megatrials/sec =   0.29
NUMTRIALS = 10
 2 threads :       10 trials ; probability =  10.00% ; megatrials/sec =   2.81
NUMTRIALS = 100
 2 threads :      100 trials ; probability =   5.00% ; megatrials/sec =   8.67
NUMTRIALS = 1000
 2 threads :     1000 trials ; probability =   5.90% ; megatrials/sec =  10.56
NUMTRIALS = 10000
 2 threads :    10000 trials ; probability =   6.51% ; megatrials/sec =   7.91
NUMTRIALS = 100000
 2 threads :   100000 trials ; probability =   6.65% ; megatrials/sec =  13.48
NUMTRIALS = 1000000
 2 threads :  1000000 trials ; probability =   6.58% ; megatrials/sec =  25.86
NUMTRIALS = 5000000
 2 threads :  5000000 trials ; probability =   6.56% ; megatrials/sec =  26.12
==============================================================================
```

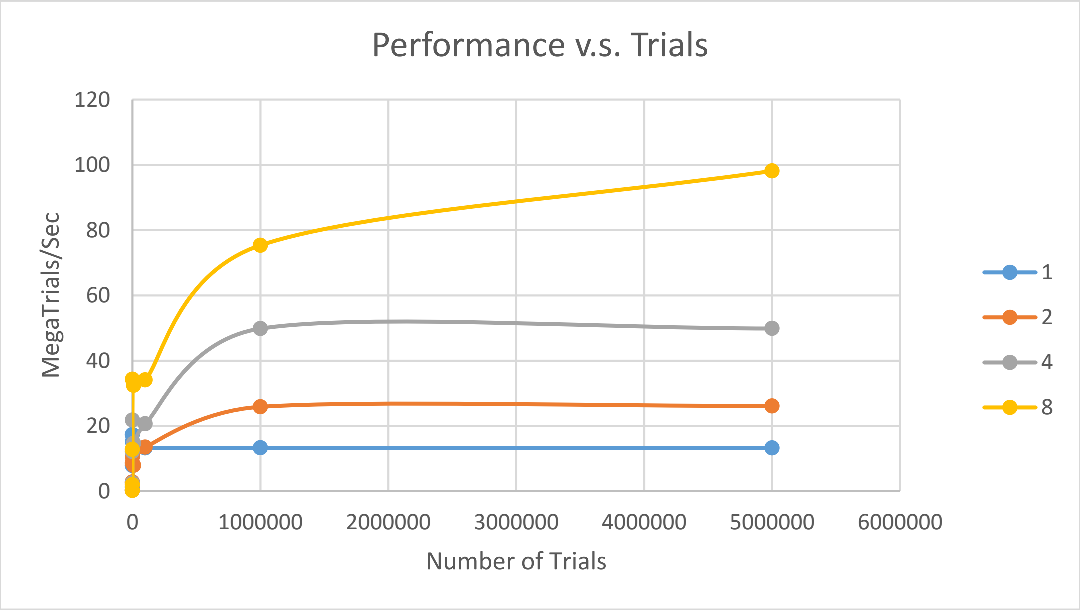


```
NUMT = 4
NUMTRIALS = 1
 4 threads :        1 trials ; probability =   0.00% ; megatrials/sec =   0.32
NUMTRIALS = 10
 4 threads :       10 trials ; probability =   0.00% ; megatrials/sec =   2.50
NUMTRIALS = 100
 4 threads :      100 trials ; probability =   8.00% ; megatrials/sec =  12.14
NUMTRIALS = 1000
 4 threads :     1000 trials ; probability =   6.70% ; megatrials/sec =  21.78
NUMTRIALS = 10000
 4 threads :    10000 trials ; probability =   6.64% ; megatrials/sec =  14.67
NUMTRIALS = 100000
 4 threads :   100000 trials ; probability =   6.58% ; megatrials/sec =  20.66
NUMTRIALS = 1000000
 4 threads :  1000000 trials ; probability =   6.60% ; megatrials/sec =  49.84
NUMTRIALS = 5000000
 4 threads :  5000000 trials ; probability =   6.56% ; megatrials/sec =  49.87
==============================================================================
NUMT = 8
NUMTRIALS = 1
 8 threads :        1 trials ; probability =   0.00% ; megatrials/sec =   0.22
NUMTRIALS = 10
 8 threads :       10 trials ; probability =  10.00% ; megatrials/sec =   2.00
NUMTRIALS = 100
 8 threads :      100 trials ; probability =   7.00% ; megatrials/sec =  12.84
NUMTRIALS = 1000
 8 threads :     1000 trials ; probability =   6.00% ; megatrials/sec =  34.28
NUMTRIALS = 10000
 8 threads :    10000 trials ; probability =   6.65% ; megatrials/sec =  32.44
NUMTRIALS = 100000
 8 threads :   100000 trials ; probability =   6.59% ; megatrials/sec =  34.12
NUMTRIALS = 1000000
 8 threads :  1000000 trials ; probability =   6.58% ; megatrials/sec =  75.35
NUMTRIALS = 5000000
 8 threads :  5000000 trials ; probability =   6.57% ; megatrials/sec =  98.12
==============================================================================
```

The above two screenshots are run by the script, and it shows probability and performance by using 4 kinds of threads and 8 kinds of trials. The following form is generated by the above results.

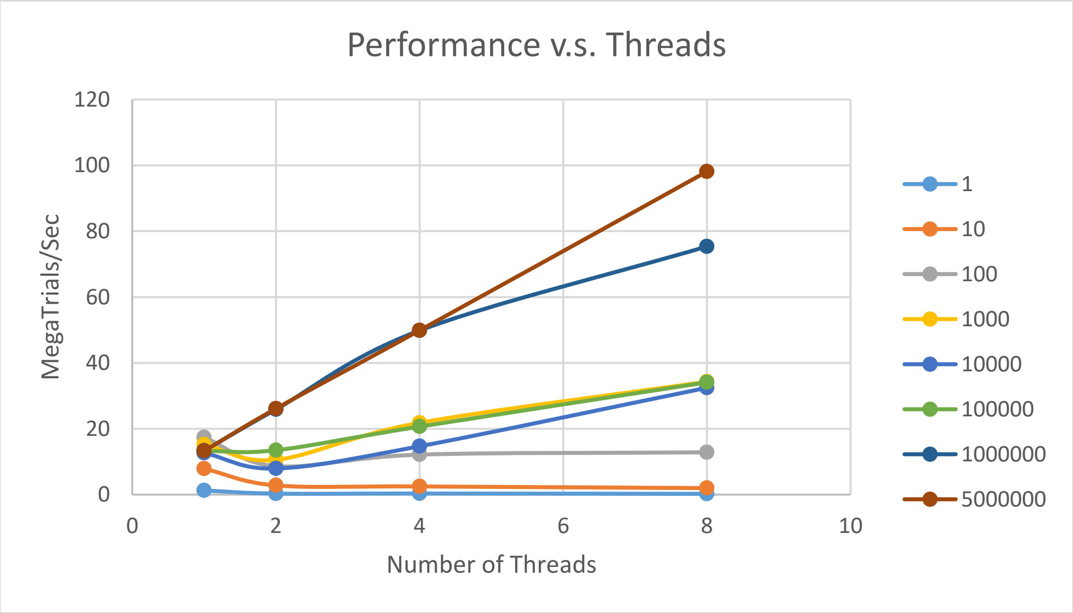
Trials & Threads									
	1	10	100	1000	10000	100000	1000000	5000000	Trials
1	1.26	7.86	17.34	15.19	12.64	13.27	13.31	13.27	
2	0.29	2.81	8.67	10.56	7.91	13.48	25.86	26.12	
4	0.32	2.5	12.14	21.78	14.67	20.66	49.84	49.87	
8	0.22	2	12.84	34.28	32.44	34.12	75.35	98.12	
Threads									megatrials/sec

When looking at the trials, especially in 1 trial and 10 trials, we can see that the probabilities are 0% or 10 %. The reason is that trials (data) are too less. For example, if we occasionally hit the castle but only with 1 trial. The probability of hitting the castle will become 100%. Also, when we have more threads in 1 trial and 10 trials, the performance are getting worse. One of the reason I assume is that the whole process doesn't need to take much time, but assign the works to each of the threads will result in a terrible performance. When the trials getting bigger, we can observe that the performance and the numbers of threads almost get a positive correlation.



The label on the right hand side presents the number of threads

One thing we have to notice is the performance won't infinitely increase when the trials getting bigger. It will gradually reach to maximal performance but the increment will get slower.



The label on the right hand side presents the number of trials

To determine the probability of hitting the castle, I will choose the highest trials with the most trials which is 5000000 (five million) trials with 8 threads. Its probability is 6.57% and its performance is 98.12 MegaTrials/sec. The reason I choose this one is because the more the data, the more accurate the probability will be.

Speedup value is 98.12/13.27 = 7.3941 (8 threads performance/ 1 thread performance)
Based on this performance, the parallel fraction I get is
8/7 * (1- (1/7.3941)) = 0.9882