

# CUDA Monte Carlo

Rui Gao

gaorui@oregonstate.edu

Project 5

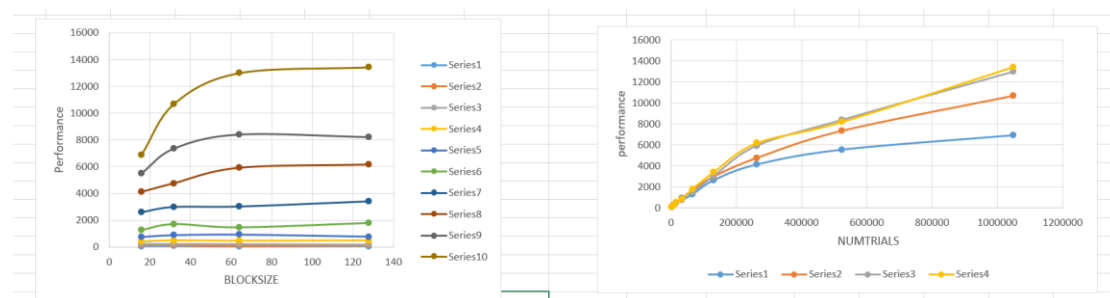
CS 575

## 1. Tell what machine you ran this on

DGX

## 2. Show the table and the two graphs

2	Peak Performance NUMTRIALS  BLOCKSIZEs											
3		2048	4096	8192	16384	32768	65536	131072	262144	524288	1048576	
4		16	52.6316	121.2121	242.4242	432.4324	732.9993	1258.7584	2612.245	4098.0488	5514.6415	6884.0337
5		32	62.5	121.2121	242.4242	484.8485	885.8132	1698.1758	2985.4228	4746.2341	7347.0853	10685.734
6		64	35.0877	117.6471	235.2941	470.5882	933.4549	1457.6513	3020.649	5906.2727	8380.5627	12982.5679
7		128	60.6061	121.2121	205.1282	484.8485	763.6093	1780.8695	3390.7285	6154.7708	8191.9996	13424.0068
8												
9	Probability(%)											
10		16	7.324	6.812	6.592	6.964	7.01	7.121	7.002	7.085	7.127	7.098
11		32	6.396	7.91	7.312	6.909	7.22	7.1	7.109	7.067	7.124	7.099
12		64	7.08	6.152	7.69	6.622	7.043	7.15	7.143	7.176	7.051	7.14
13		128	7.568	6.665	6.75	7.159	6.924	6.96	7.134	7.189	7.082	7.085
14												



## 3. What patterns are you seeing in the performance curves?

As the number of NMTRAILS and BLOCKSIZE increase, the performance increase.

## 4. Why do you think the patterns look this way?

Because there has more threads to handle the jobs, so the ability increased.

Because there has more jobs to do, so more cores had been assigned jobs to do.

**5. Why is a BLOCKSIZE of 16 so much worse than the others?**

Because there has less machines to do the jobs.

**6. How do these performance results compare with what you got in Project #1? Why?**

Much faster in Project5, because GPU has more computing power, I guess. In addition, I calculated wrong in project#1

**7. What does this mean for the proper use of GPU parallel computing?**

We'd better use GPU for simple but heavy amount computing missions.