CS-575 Parallel Programming

Spring 2022

Name: Akhil Sai Chintala

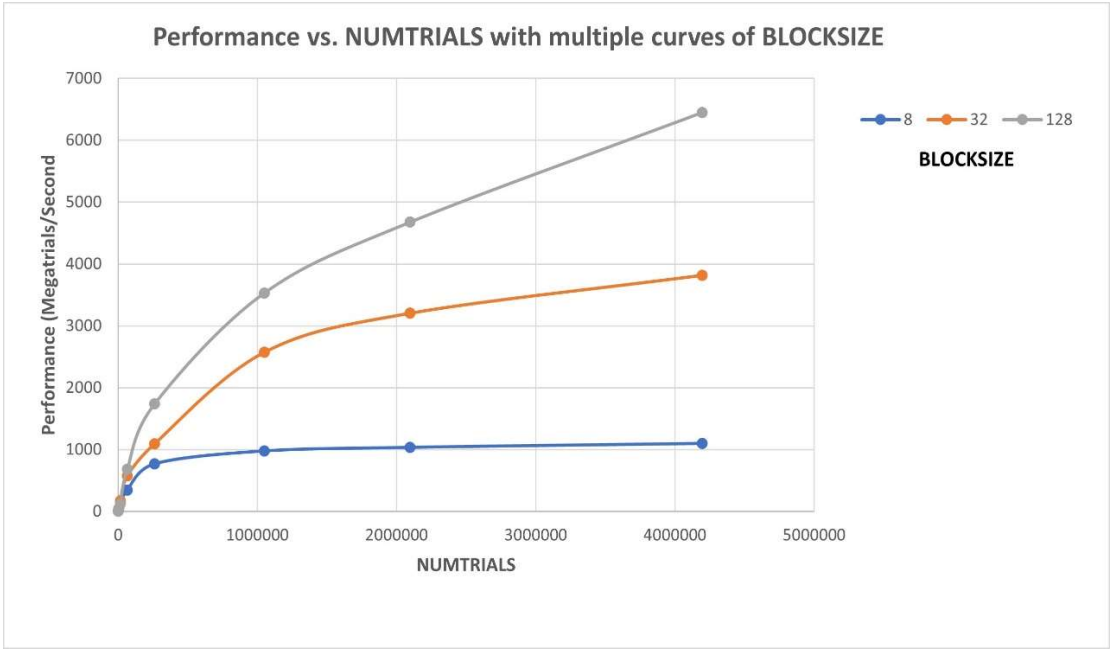
Email: chintala@oregonstate.edu

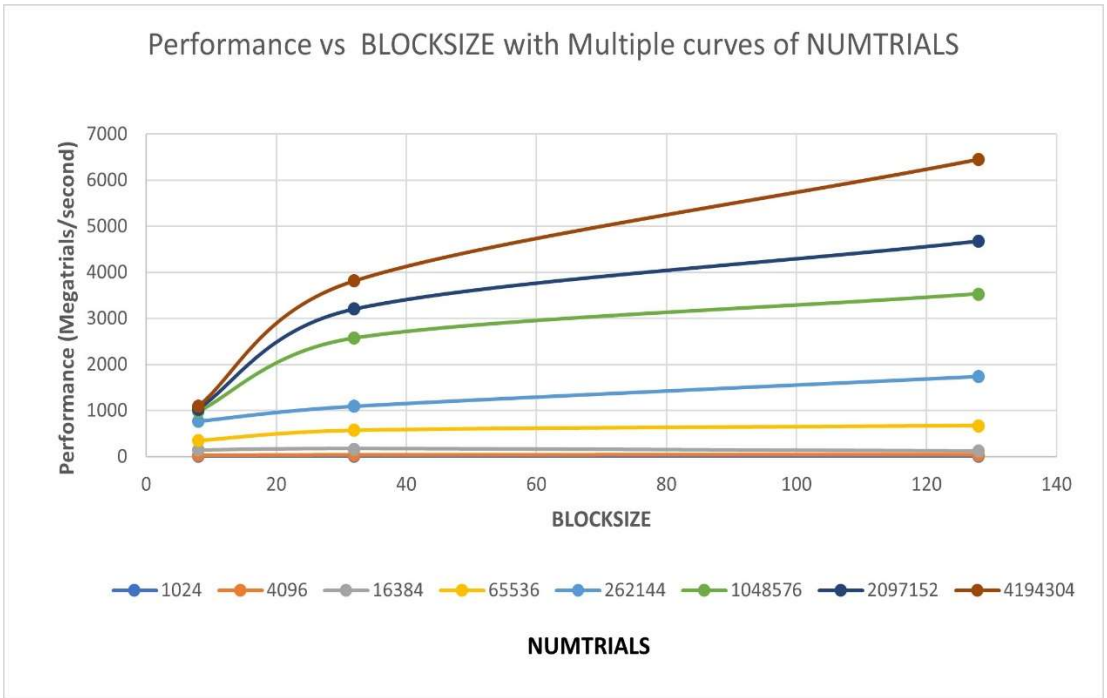
Project Name: #Project-5

1. **Tell what machine you ran this on?**
   I have executed the project-5 code on the rabbit server "rabbit.engr.oregonstate.edu".

2. **Show the table and the two graphs**

NUMTRIALS	BLOCKSIZE	megaTrialsPerSecond	probability
1024	8	9.8826	21.58
1024	32	8.0767	21.09
1024	128	9.0498	20.51
4096	8	35.8142	22.39
4096	32	40.5963	23.05
4096	128	44.062	22.49
16384	8	140.1204	22.73
16384	32	171.4094	22.92
16384	128	120.0751	22.43
65536	8	347.2952	22.52
65536	32	573.8302	22.61
65536	128	680.851	22.29
262144	8	770.7216	22.35
262144	32	1094.7481	22.53
262144	128	1739.6476	22.42
1048576	8	977.8574	22.57
1048576	32	2571.0474	22.54
1048576	128	3529.8935	22.49
2097152	8	1035.2911	22.45
2097152	32	3204.2243	22.47
2097152	128	4679.1375	22.51
4194304	8	1098.8783	22.49
4194304	32	3817.5569	22.48
4194304	128	6451.0288	22.49


Performance vs. NUMTRIALS with multiple curves of BLOCKSIZE
Performance (Megatrials/Second
7000
6000
5000
4000
3000
2000
1000
0
0
1000000
2000000
3000000
4000000
5000000
NUMTRIALS
8
32
128
BLOCKSIZE


Performance vs BLOCKSIZE with Multiple curves of NUMTRIALS
Performance (Megatrials/second)
7000
6000
5000
4000
3000
2000
1000
0
0
20
40
60
80
100
120
140
BLOCKSIZE
1024
4096
16384
65536
262144
1048576
2097152
4194304
NUMTRIALS

3. **What patterns are you seeing in the performance curves?**
   In the graphs,

   a) **Performance vs BLOCKSIZE:**
   Performance of 1024 and 4096 Numtrials remains constant throughout the execution whereas, 16384 and 65536 Numtrials are little bit improved in performance. Remaining all the Numtrials performance is increased more as the number of blocks increase as shown in the graph.

   b) **Performance vs NUMTRIALS:**
   Performance of BlockSize 8 increased till few numtrials and remained stagnant till the end of the execution. The others BlockSizes 32 and 128 increases as the number of trials increase and the 128 blocksize reaches the maximum among the others. So, when the number of trials increases performance also increases.

4. **Why do you think the patterns look this way?**
   As the number of trials increases the number of blocks should be assigned by each processor. The size of the block remains same, and the data is increased which results in more work and more performance. As size of the block increases, thread size also increases which results in better performance.

5. **Why is a BLOCKSIZE of 8 so much worse than the others?**
   When compared to the other two blocks, the performance of blocksize 8 is much worse which is clearly seen in the graphs. This is because, the size is less than 32 and the other two blocks are 32 and 128 which are multiples of 32 where the performance can be maximized with full utilization.

6. **How do these performance results compare with what you got in Project #1? Why?**
   By comparing the performance of this with Project #1 the current performance is much greater than the project #1. In this project we are using CUDA which is developed by NVDIA which have its own GPU's used to speed up the computing performance.

7. **What does this mean for the proper use of GPU parallel computing?**
   GPU Parallel computing is properly used when the performance is maximum. The larger data we have the better performances i.e., threads can be fully utilized without making them idle. The data selection is selected based on multiples of 32 so that each block size gets a complete queue at a time. The block size is limited to certain maximum, which if the block size reaches maximum the performance doesn't improve.