Swetha Jayapathy
Student ID : 934041047
Instructor : Mike Bailey
CS575: Introduction to Parallel Programming
May 20, 2020

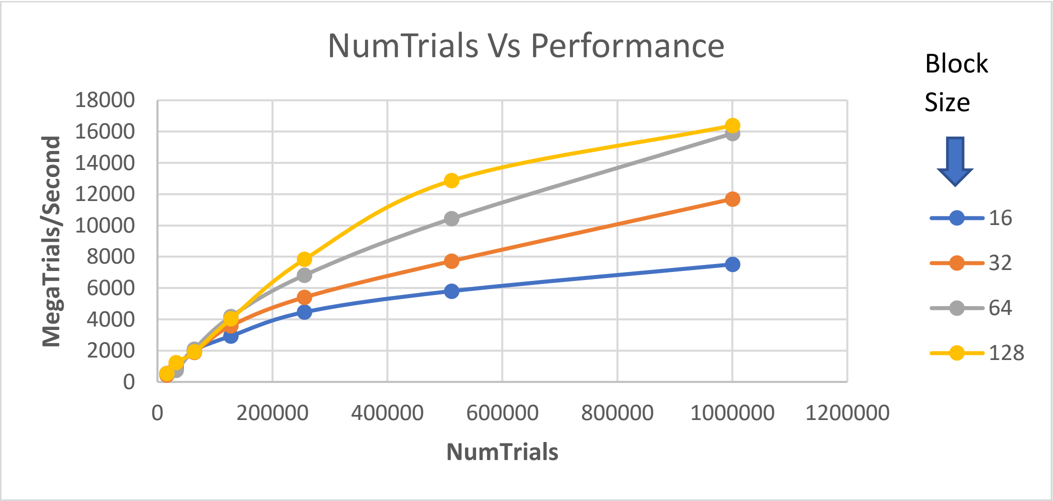
Project #05

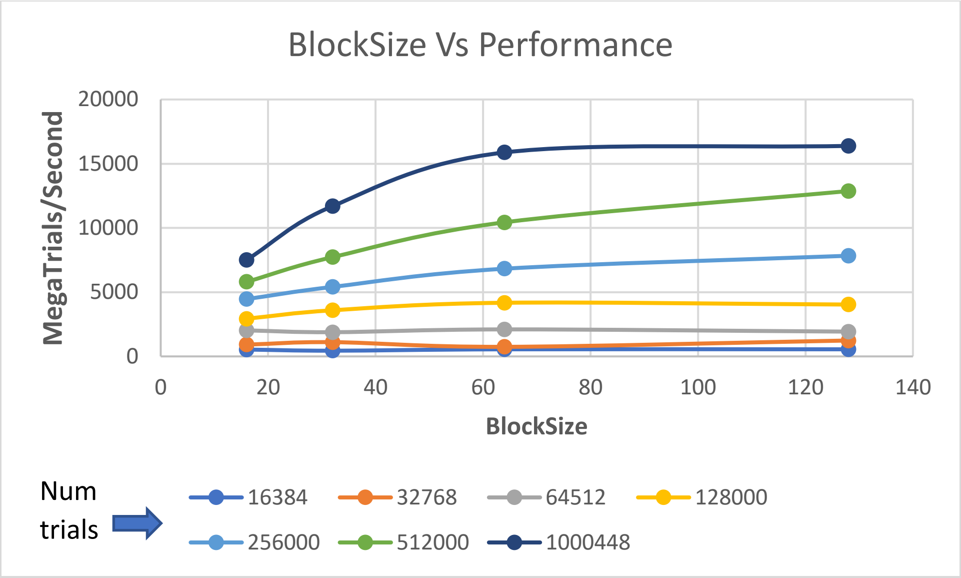
1. Tell what machine you ran this on

I ran my code on DGX system.

2. Show the table and the two graphs

BlockSize/Numtrials	16	32	64	128
16384	533.3333	444.4444	551.7241	551.7241
32768	915.9213	1109.426	744.7273	1236.715
64512	2026.131	1884.112	2102.19	1929.187
128000	2934.703	3590.664	4175.365	4036.327
256000	4461.796	5409.06	6825.939	7835.456
512000	5805.515	7725.737	10437.05	12872.08
1000448	7515.385	11696.22	15878.11	16385.74





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

   It could be seen from the first graph that as the Numtrials increases, the performance also increases. From Graph 2, it can be observed that as the Block size increases, for lower value of Numtrials , there is only a slight performance increase and then it flattens out. But for larger Numtrials, it can be seen that there is a significant performance enhancement initially and then it remains.

   With block size at 64, most of them reaches saturation point.

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

   It can be seen that with large data size (Numtrials), the performance increases as the GPU gets to utilize its whole potential as the data size increases. Occupancy is the ratio of active warps to the maximum number warps which can be used. Since occupancy increases with large data size, the GPU gets to use its full capacity and hence the increase in performance.

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

   Block size of 16 is worse than others as the minimum that a GPU can handle is 32 which is 1 warp. So, when data is lower than this, the performance gets affected. There needs to be a bunch of warps to work on so something is always ready to run. But here 16 is much lesser than 32 and hence this low performance. This is a case of low occupancy, having too few eligible warps.

   Each block should have at least one warp active at a time, where the warp size is 32. Hence, 32 threads can be active at once. With larger block size, we can get multiple warps where warp scheduler can be used for swap between warps when one warp stalls out and hence we can get a greater performance.

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

   When this is compared to the results of Project 1, it can be observed that there is a huge performance increase with respect to using GPU rather than simple Multithreading using OpenMP. This is due to the high level of parallelism available in GPU compared to OpenMP. GPU has more capacity than openMP in terms of the number of threads.

   Also, it can be observed that in Project 1 - the performance of all the threads reaches a saturation level at a much earlier stage (at 100000 #trials). This means that even for large amount of data, the performance remains same. But with GPU, we can get a much higher performance when the data size is increased.

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

   Each block should have atleast one warp active at a time, where the warp size is 32. Hence, 32 threads can be active at once. With larger block size, we can get multiple warps where warp scheduler can be used for swap between warps when one warp stalls out.
   For proper usage of GPU, its whole capacity is to be utilized. Occupancy is the ratio of active warps to the maximum number of active warps. Low occupancy results in poor efficiency as there are not enough eligible warps to hide the latency. When the occupancy is at the sufficient level to hide latency, increasing it may lead to lowering the performance due to the reduction in resources per thread.
  GPU’s is best for arithmetic operations and calculations as it has a heavy parallel architecture.