GPU — Architecture & Programming
Assignment 1

Anirudhan Rajagopalan — ajr619@nyu.edu

October 11, 2016

1 Q3. Bar chart

The below chart shows the performance of GPU and CPU versions.

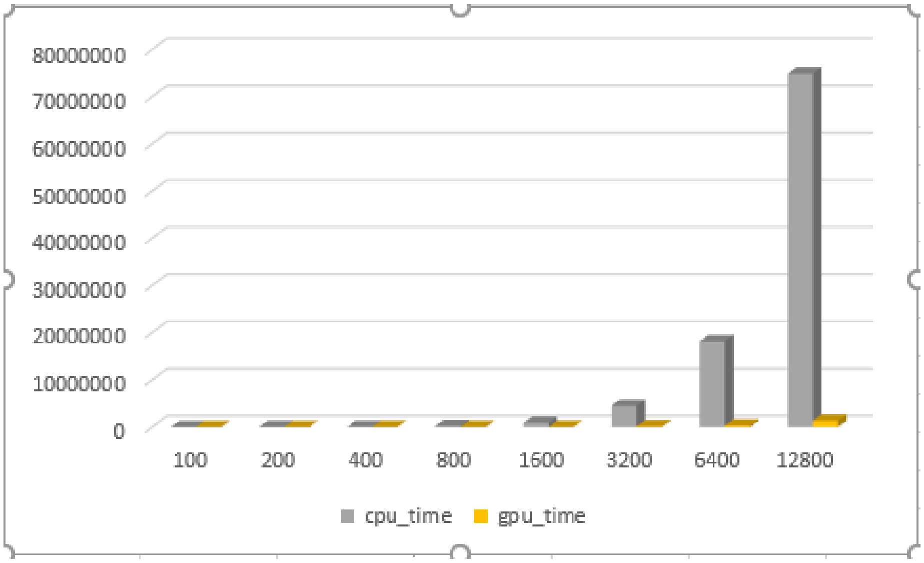


Figure 1: GPU and CPU performance.

2 Q4. Observations

The image 2 shows the performance of the GPU and CPU functions.

2.1 When GPU usage is more beneficial

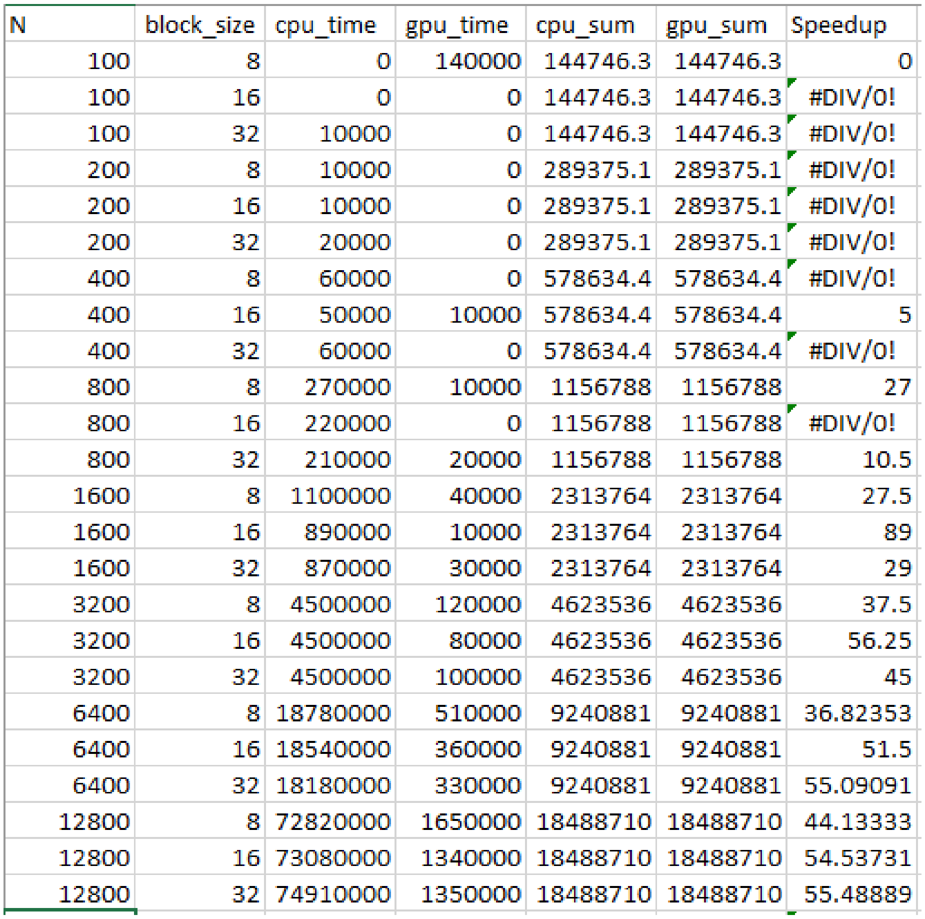
GPU usage is generally beneficial when the size of our playground is bigger. For smaller size playground (from 32 to 200), the performance of CPU is better than that of GPU. This is mainly because we need to transfer data across GPU and CPU for doing our calculations and also the GPU overhead.

2.2 Lowest Speedup

The lowest speedup (almost in -ve) is observed for N = 100 and block size 32. As explained above, because of the overhead of copying the data (which is the slowest operation in many GPUs) we have poor performance.

2.3 Highest Speedup

The hightest speedup is observed for N = 12800 and block size of 32. This gives the highest performance as we are using max noȯf threads per block (1024) and



N	block_size	cpu_time	gpu_time	cpu_sum	gpu_sum	Speedup
100	8	0	140000	144746.3	144746.3	0
100	16	0	0	144746.3	144746.3	#DIV/0!
100	32	10000	0	144746.3	144746.3	#DIV/0!
200	8	10000	0	289375.1	289375.1	#DIV/0!
200	16	10000	0	289375.1	289375.1	#DIV/0!
200	32	20000	0	289375.1	289375.1	#DIV/0!
400	8	60000	0	578634.4	578634.4	#DIV/0!
400	16	50000	10000	578634.4	578634.4	5
400	32	60000	0	578634.4	578634.4	#DIV/0!
800	8	270000	10000	1156788	1156788	27
800	16	220000	0	1156788	1156788	#DIV/0!
800	32	210000	20000	1156788	1156788	10.5
1600	8	1100000	40000	2313764	2313764	27.5
1600	16	890000	10000	2313764	2313764	89
1600	32	870000	30000	2313764	2313764	29
3200	8	4500000	120000	4623536	4623536	37.5
3200	16	4500000	80000	4623536	4623536	56.25
3200	32	4500000	100000	4623536	4623536	45
6400	8	18780000	510000	9240881	9240881	36.82353
6400	16	18540000	360000	9240881	9240881	51.5
6400	32	18180000	330000	9240881	9240881	55.09091
12800	8	72820000	1650000	18488710	18488710	44.13333
12800	16	73080000	1340000	18488710	18488710	54.53731
12800	32	74910000	1350000	18488710	18488710	55.48889

Figure 2: Figure showing performance of GPU and CPU functions for various values of N and various block width.

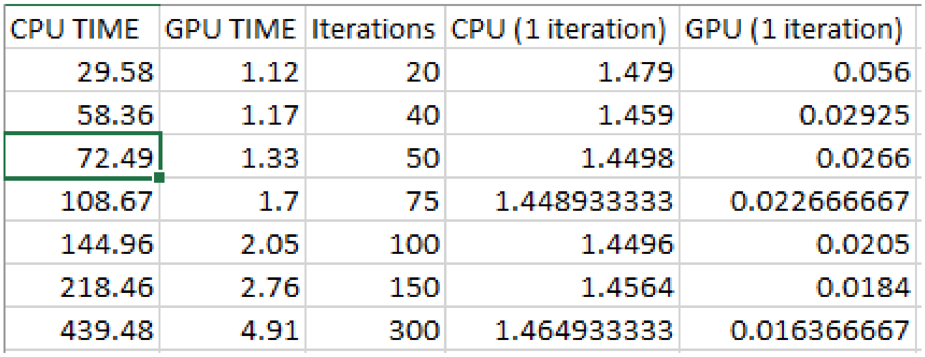
also we are making full utilization of the parallel GPUs.

3 Q5. Effect of number of iterations

As explained in the section above, we ran a lot of experiments similar to searching a grid of three dimensions. One dimension for N, the other for block size and the third dimension for iterations.

1. N = 100, 200, 400, 800, 1600, 3200, 6400, 12800
2. BLOCK_WIDTH = 8×8, 16×16, 32×32
3. Iterations = 20, 40, 50, 75, 100, 150, 300

We took our best performing (GPU performing) values of N (12800) and block size (32) based on the speedup ratio for all these experiments and tried to run experiments by varying the number of iterations. The values we got are summarized in Figure 3



CPU TIME	GPU TIME	Iterations	CPU (1 iteration)	GPU (1 iteration)
29.58	1.12	20	1.479	0.056
58.36	1.17	40	1.459	0.02925
72.49	1.33	50	1.4498	0.0266
108.67	1.7	75	1.448933333	0.022666667
144.96	2.05	100	1.4496	0.0205
218.46	2.76	150	1.4564	0.0184
439.48	4.91	300	1.464933333	0.016366667

Figure 3: Effect of iterations on cpu and gpu performance. N = 12800 and block size = 32×32.

As you can see, the utilization of GPU goes up as we add more number of iterations to GPU. This is because GPU will be able to schedule the warps more efficiently as there are more operations to be done. Thus the execution effeciency goes up.