

# CS698L Assignment-5

Hritvik Taneja(160300)

November 23, 2019

## Problem 2

	Run-1	Run-2	Run-3	Run-4
Serial(SIZE = 4096)	6192.69 msec	6028.99 msec	6211.61 msec	6001.25 msec
Parallel(SIZE = 4096)	157.844 msec	159.41 msec	180.665 msec	157.502 msec
Parallel(SIZE = 4097)	156.913 msec	148.321 msec	153.31 msec	148.99 msec
SpeedUp(SIZE = 4096)	39.23	37.82	34.50	38.10

\*This includes cudaMemcpy time as well

Since  $j$  goes from 0 to  $\text{SIZE}-2$ (both inclusive), and all the  $j$  iterations run in parallel. Hence, for  $\text{SIZE} = 4096$  there will be divergence in the last wrap of the last block in the grid, whereas this is not the case when  $\text{SIZE} = 4097$ , so that is why  $\text{SIZE} = 4097$  performs better.

### Problem 3

SIZE = 1024	Run-1	Run-2	Run-3	Run-4
Serial	11718.9 msec	9890.89 msec	9716.82 msec	11119.9 msec
Parallel	7.72022 msec	7.91219 msec	7.77123 msec	7.65194 msec

\*This includes cudaMemcpy time as well and TILE-SIZE = 32

Current initialization of  $A[i][j] = 0.25 * i * j$ , but if we change this to  $A[i][j] = 0.25 * \text{random}()$ , we will see a lot of differences between the CPU and the GPU version, this is because of the precision of double, which behaves differently in GPU.

## Problem 4

SIZE = 512	Run-1	Run-2	Run-3	Run-4
Serial	564.417 msec	543.935 msec	546.25 msec	593.389 msec
Parallel(Kernel-1)	1.75206 msec	1.80266 msec	1.76582 msec	1.72083 msec
Parallel(Kernel-2)	1.32989 msec	2.18528 msec	1.33222 msec	1.31155 msec

\*This includes cudaMemcpy time as well and TILE-SIZE = 16

SIZE = 1024	Run-1	Run-2	Run-3	Run-4
Serial	5070.77 msec	4939.17 msec	5138.53 msec	5189.07 msec
Parallel(Kernel-1)	9.52877 msec	9.52394 msec	9.70864 msec	10.8619 msec
Parallel(Kernel-2)	7.41491 msec	7.33402 msec	10.8109 msec	8.5520 msec

\*This includes cudaMemcpy time as well and TILE-SIZE = 16

SIZE = 2048	Run-1	Run-2	Run-3	Run-4
Serial	58189.6 msec	57808.6 msec	58307.2 msec	68033 msec
Parallel(Kernel-1)	62.9573 msec	63.8941 msec	81.1943 msec	83.4262 msec
Parallel(Kernel-2)	46.8607 msec	51.7905 msec	48.1859 msec	46.3817 msec

\*This includes cudaMemcpy time as well and TILE-SIZE = 16

SIZE = 4096	Run-1	Run-2	Run-3	Run-4
Serial	1.02533e+06 msec	-	-	-
Parallel(Kernel-1)	570.155 msec	613.754 msec	635.421 msec	673.252 msec
Parallel(Kernel-2)	343.824 msec	447.729 msec	453.418 msec	449.945 msec

\*This includes cudaMemcpy time as well and TILE-SIZE = 16. Since the serial version was taking too much time, so only calculated for Run-1

So, as SIZE increases the performance gap between Kernel-1 and Kernel-2 increases.

SIZE = 4096	Run-1	Run-2	Run-3	Run-4
Parallel(Kernel-2)[TILE-SIZE = 8]	643.001 msec	674.459 msec	553.967 msec	644.381 msec
Parallel(Kernel-2)[TILE-SIZE = 16]	453.235 msec	451.901 msec	452.607 msec	392.899 msec
Parallel(Kernel-2)[TILE-SIZE = 32]	398.101 msec	399.528 msec	401.324 msec	375.026 msec

\*This includes cudaMemcpy time as well.

Increasing tile size improves performance.