

# Vector Reduction

-Rutvij Girish Karkhanis

## Item: 1

Item 2 is implemented in vectorReduce.cu, Item 4 is implemented in atomicadd.cu, and Item 5 is implemented in atomicadd\_if.cu. All the files are included in the tar file.

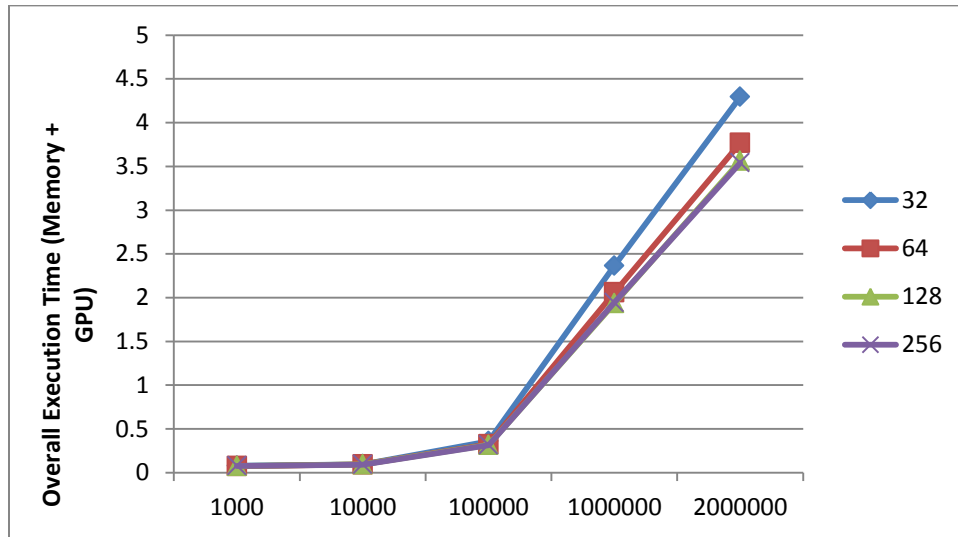
## Item: 2

The vector reduction was run on GPU5 (ecee-gpu5.colorado.edu) with varying block size and input size. The finding has been tabulated as below.

Input Size	Blocksize	GPU Execution Time	Memory Transfer Time	CPU Time (to add partial sums)	Overall Execution Time (Memory + GPU Execution)
1000	32	0.035000	0.045000	0.000000	0.080000
10000	32	0.037000	0.058000	0.001000	0.096000
100000	32	0.137000	0.223000	0.005000	0.360000
1000000	32	0.671000	1.694000	0.051000	2.365000
2000000	32	1.079000	3.214000	0.096000	4.293000
1000	64	0.034000	0.039000	0.000000	0.074000
10000	64	0.036000	0.058000	0.001000	0.094000
100000	64	0.117000	0.207000	0.002000	0.324000
1000000	64	0.445000	1.615000	0.025000	2.060000
2000000	64	0.687000	3.079000	0.049000	3.767000
1000	128	0.034000	0.041000	0.000000	0.075000
10000	128	0.036000	0.058000	0.000000	0.094000
100000	128	0.112000	0.205000	0.001000	0.317000
1000000	128	0.367000	0.596000	0.011000	1.936000
2000000	128	0.530000	3.032000	0.025000	3.563000
1000	256	0.034000	0.043000	0.000000	0.077000
10000	256	0.035000	0.056000	0.001000	0.092000
100000	256	0.112000	0.203000	0.001000	0.315000
1000000	256	0.382000	1.561000	0.006000	1.944000
2000000	256	0.545000	2.995000	0.012000	3.540000

Above table shows different execution times for varying block and input sizes. It is seen that the GPU execution and memory time increases when input size is increased. However, when the block size is increased the GPU execution and memory copy time slightly reduces. Additionally the CPU execution time significantly reduces with the increase in the block size.

**Item: 3**



From the above graph it is inferred that with the increase in the block size, the overall execution time decreases. This difference is more significant for larger input sizes.

**Item: 4**

Input Size	Blocksize	Previous total execution (CPU+GPU)	Total execution (atomic support in GPU)	Speedup
1000	32	0.035000	0.034000	1.029411
10000	32	0.038000	0.039000	0.974358
100000	32	0.142000	0.145000	0.979310
1000000	32	0.722000	0.692000	1.043352
2000000	32	1.175000	1.162000	1.011187
1000	64	0.034000	0.035000	0.971428
10000	64	0.037000	0.040000	0.925000
100000	64	0.119000	0.155000	0.767741
1000000	64	0.470000	0.691000	0.680173
2000000	64	0.736000	1.161000	0.633936
1000	128	0.034000	0.035000	0.971428
10000	128	0.036000	0.039000	0.923076
100000	128	0.113000	0.144000	0.784722
1000000	128	0.378000	0.709000	0.533145
2000000	128	0.555000	1.160000	0.478448

1000	256	0.034000	0.035000	0.971428
10000	256	0.036000	0.039000	0.923076
100000	256	0.113000	0.145000	0.779310
1000000	256	0.388000	1.154000	0.336222
2000000	256	0.557000	1.161000	0.479758

**Item: 5**

Input Size	GPU Overall Execution Time (blocksize=32) with the if statement present	GPU Overall Execution Time (blocksize=32) without the if statement present	Percentage different in performance
1024	0.078000	0.077000	1.29
4096	0.089000	0.086000	3.40
16384	0.114000	0.114000	0.00
262144	0.870000	0.838000	3.81
1048576	2.392000	2.360000	1.35

From the above table it is observed that by removing the overhead created by “if” statement increases the performance of the GPU. In this case, we are passing multiples of the block size. Hence the statement can be removed to achieve faster execution.