**Name: Rohan Jhaveri**

## Assignment 2: GPU (CUDA) Vector Reduction

[Item 1] – .cu file submitted

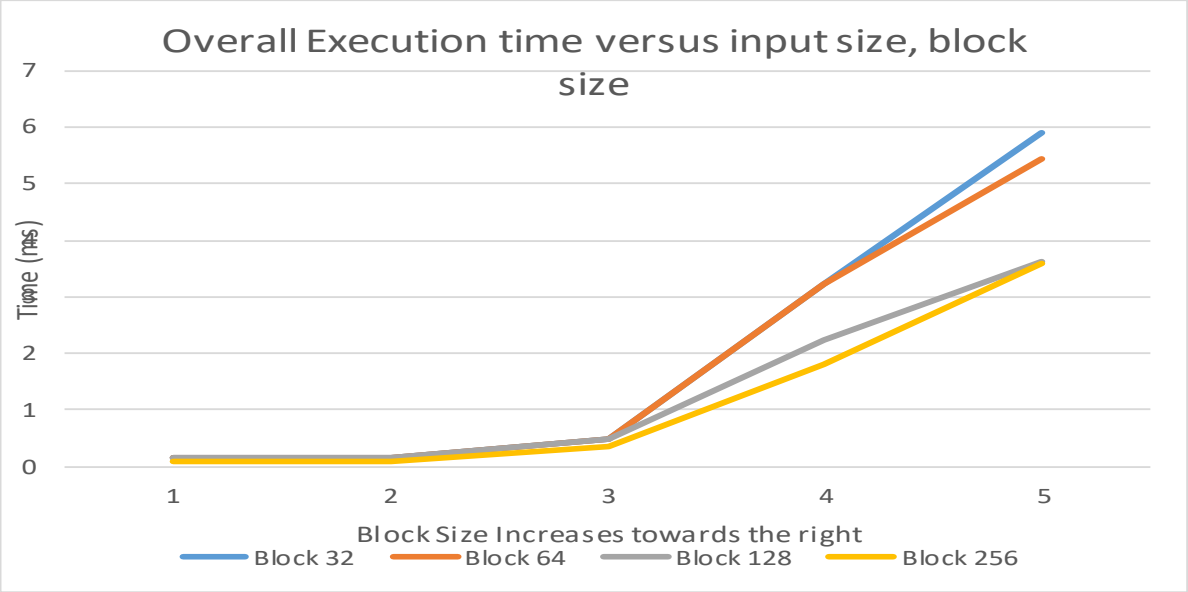
[Item 2] - **Provide the execution time and memory transfer time for the following data sizes:**

| Input Size | Blocksize | GPU Execution Time | Memory Transfer Time | CPU Time (to add partial sums) | Overall Execution Time (Memory Transfer Time + GPU Execution Time +CPU Time) |
|---|---|---|---|---|---|
| 1000 | 32 | 0.064000 | 0.077000 | 0.000000 | 0.141000 |
| 10000 | 32 | 0.068000 | 0.099000 | 0.001000 | 0.168000 |
| 100000 | 32 | 0.151000 | 0.330000 | 0.005000 | 0.486000 |
| 1000000 | 32 | 0.781000 | 2.387000 | 0.046000 | 3.214000 |
| 2000000 | 32 | 1.362000 | 4.435000 | 0.091000 | 5.888000 |
| 1000 | 64 | 0.070000 | 0.077000 | 0.000000 | 0.147000 |
| 10000 | 64 | 0.069000 | 0.094000 | 0.001000 | 0.164000 |
| 100000 | 64 | 0.150000 | 0.323000 | 0.005000 | 0.478000 |
| 1000000 | 64 | 0.782000 | 2.386000 | 0.045000 | 3.213000 |
| 2000000 | 64 | 1.361000 | 4.443000 | 0.091000 | 5.895000 |
| 1000 | 128 | 0.064000 | 0.078000 | 0.000000 | 0.142000 |
| 10000 | 128 | 0.072000 | 0.097000 | 0.001000 | 0.170000 |
| 100000 | 128 | 0.150000 | 0.326000 | 0.005000 | 0.481000 |
| 1000000 | 128 | 0.782000 | 2.396000 | 0.046000 | 2.224000 |
| 2000000 | 128 | 1.362000 | 4.439000 | 0.090000 | 3.62800 |
| 1000 | 256 | 0.065000 | 0.076000 | 0.001000 | 0.09520 |
| 10000 | 256 | 0.068000 | 0.094000 | 0.001000 | 0.10030 |
| 100000 | 256 | 0.150000 | 0.328000 | 0.005000 | 0.35000 |
| 1000000 | 256 | 0.781000 | 2.391000 | 0.046000 | 1.8000 |
| 2000000 | 256 | 1.360000 | 4.546000 | 0.095000 | 3.5870 |

**Name: Rohan Jhaveri**

**Assignment 2: GPU (CUDA) Vector Reduction**

[Item 3] – **Overall Execution time versus input size, block size**




[Item 4] – **Results with Atomic Add:**

| Input Size | Blocksize | Previous total execution (CPU+GPU) | Total execution (atomic support in GPU) | Speedup |
|---|---|---|---|---|
| 1000 | 32 | 0.141000 | 0.142000 | 0.9x |
| 10000 | 32 | 0.168000 | 0.166000 | 1.01x |
| 100000 | 32 | 0.486000 | 0.483000 | 1.001x |
| 1000000 | 32 | 3.214000 | 3.218000 | 1.0 |
| 2000000 | 32 | 5.888000 | 5.916000 | 0.98x |
| 1000 | 64 | 0.147000 | 0.142000 | 1.035x |
| 10000 | 64 | 0.164000 | 0.167000 | 0.98x |
| 100000 | 64 | 0.478000 | 0.482000 | 0.97x |
| 1000000 | 64 | 3.213000 | 3.217000 | 1x |
| 2000000 | 64 | 5.895000 | 5.907000 | 0.98x |
| 1000 | 128 | 0.142000 | 0.141000 | 1.07x |
| 10000 | 128 | 0.170000 | 0.173000 | 0.982x |
| 100000 | 128 | 0.481000 | 0.486000 | 0.98x |
| 1000000 | 128 | 3.224000 | 3.216000 | 1.03x |
| 2000000 | 128 | 5.891000 | 5.901000 | 0.99x |
| 1000 | 256 | 0.142000 | 0.141000 | 1.07x |
| 10000 | 256 | 0.163000 | 0.170000 | 0.95x |
| 100000 | 256 | 0.483000 | 0.490000 | 0.98x |
| 1000000 | 256 | 3.218000 | 3.224000 | 0.98x |
| 2000000 | 256 | 6.001000 | 5.943000 | 1.01x |

**Name: Rohan Jhaveri**

**Assignment 2: GPU (CUDA) Vector Reduction**

The atomic add tends to speed up the overall execution by a really small margin or do the keep the time same for most of the block sizes. As we increase the data size, the number of blocks increase causing the addition of the block outputs by atomics to take more time and that is why it does not improve on the previous gpu performance by a whole lot.

**[Item 5] – Results without the if statement.**

```
 if (globalid < N) {
… // work
 }
```

| 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 | 0.064000 | 0.063000 | 1.5% |
| 4096 | 0.066000 | 0.066000 | 0 |
| 16384 | 0.076000 | 0.073000 | 3.94% |
| 262144 | 0.364000 | 0.360000 | 1.09% |
| 1048576 | 0.844000 | 0.824000 | 2.36% |

Here the GPU shows an improvement of about 3% on an average when the input sizes split exactly by the block size.