

CMPE-214

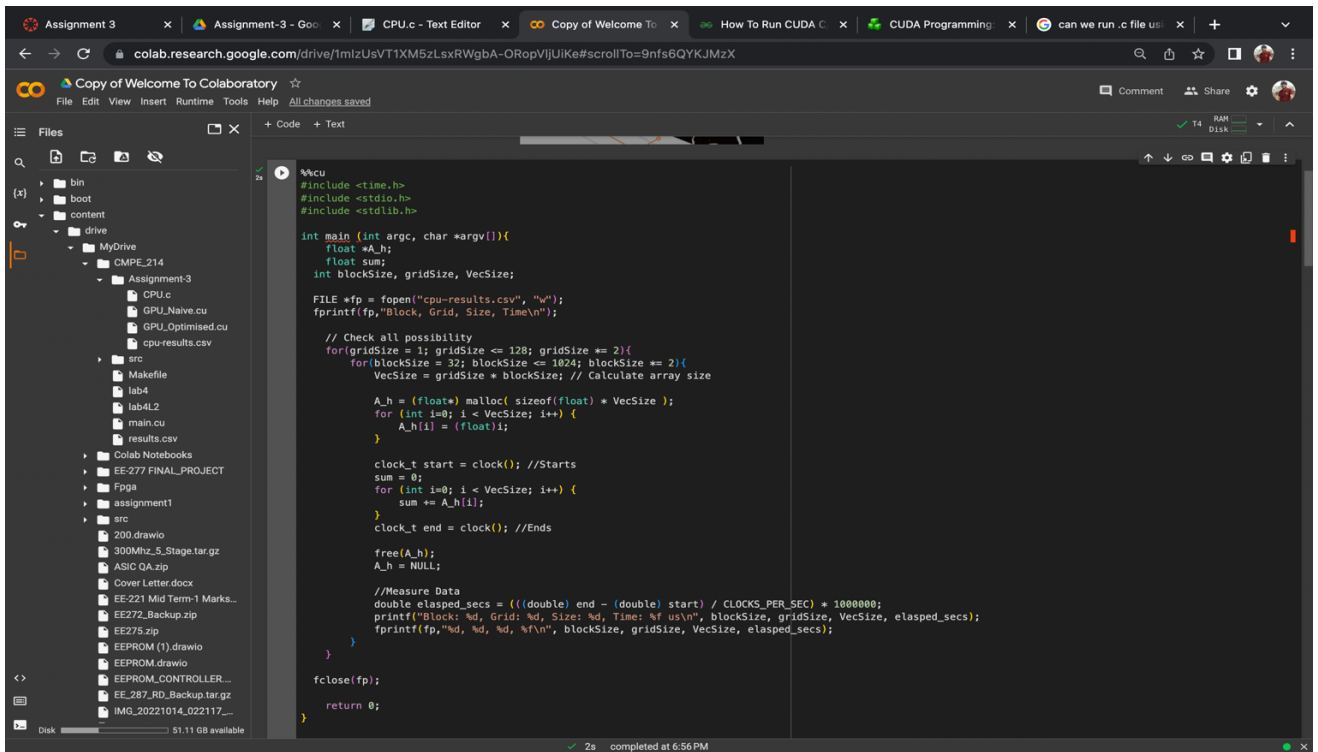
Assignment-3

Implement, test, and compare the performance of Reduction with the following configurations:

- 1. Serial (CPU)**
- 2. No optimization (naive GPU method)**
- 3. GPU with optimized thread organization (as indicated in the hint)**

You can use simple patterns to initialize your input array and verify the GPU results with the CPU result. Also, use different grid and block size combinations to observe the performance difference. For example, you can choose grid sizes of 1, 2, 4, 8, ..., 128, and for each of them use block size 32, 64, 128, ..., 1024. Then based on the execution time measurement (which should exclude the data copying part and be on the GPU side for better accuracy), draw a figure and explain your observations.

CPU:



```
%cu
#include <time.h>
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char *argv[]){
    float *A_h;
    float sum;
    int blockSize, gridSize, VecSize;

    FILE *fp = fopen("cpu-results.csv", "w");
    fprintf(fp, "Block, Grid, Size, Time\n");

    // Check all possibility
    for(gridSize = 1; gridSize <= 128; gridSize += 2){
        for(blockSize = 32; blockSize <= 1024; blockSize += 2){
            VecSize = gridSize * blockSize; // Calculate array size

            A_h = (float*) malloc( sizeof(float) * VecSize );
            for (int i=0; i < VecSize; i++) {
                A_h[i] = (float)i;
            }

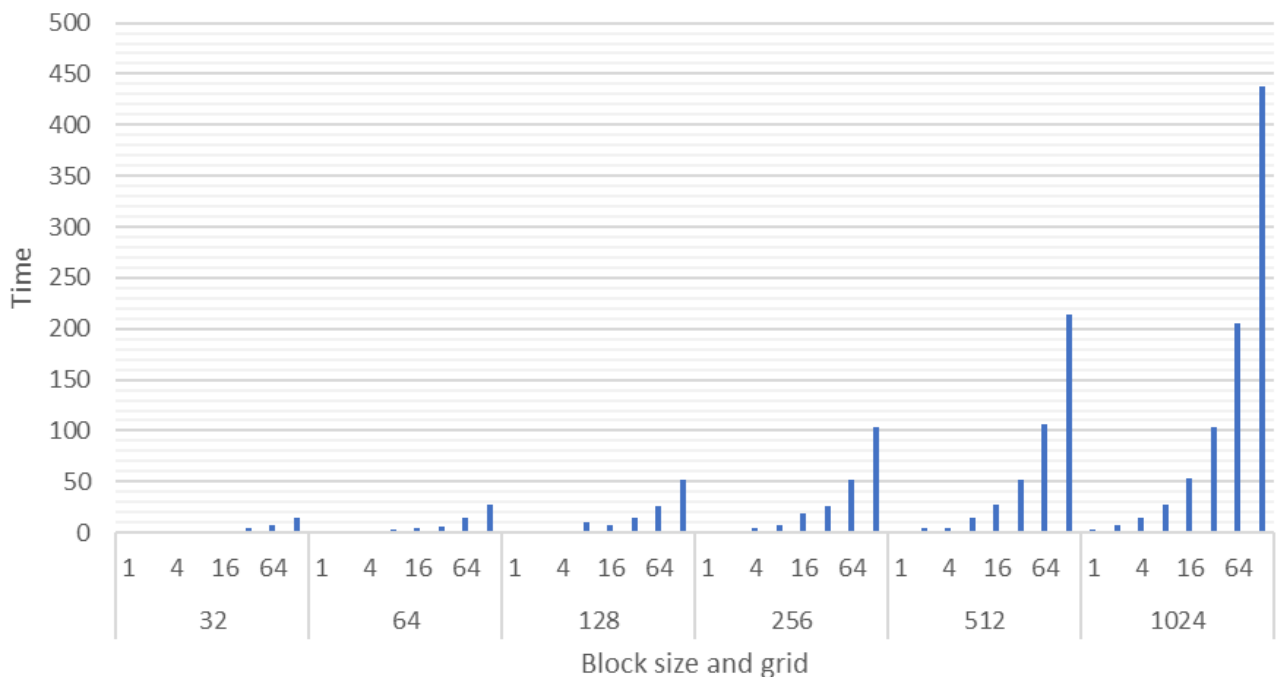
            clock_t start = clock(); //Starts
            sum = 0;
            for (int i=0; i < VecSize; i++) {
                sum += A_h[i];
            }
            clock_t end = clock(); //Ends

            free(A_h);
            A_h = NULL;

            //Measure Data
            double elapsed_secs = ((double) end - (double) start) / (CLOCKS_PER_SEC * 1000000);
            printf("Block: %d, Grid: %d, Size: %d, Time: %f us\n", blockSize, gridSize, VecSize, elapsed_secs);
            fprintf(fp, "%d, %d, %d, %f\n", blockSize, gridSize, VecSize, elapsed_secs);
        }
    }

    fclose(fp);
    return 0;
}
```

Sum of Time



Block ▾ Grid ▾

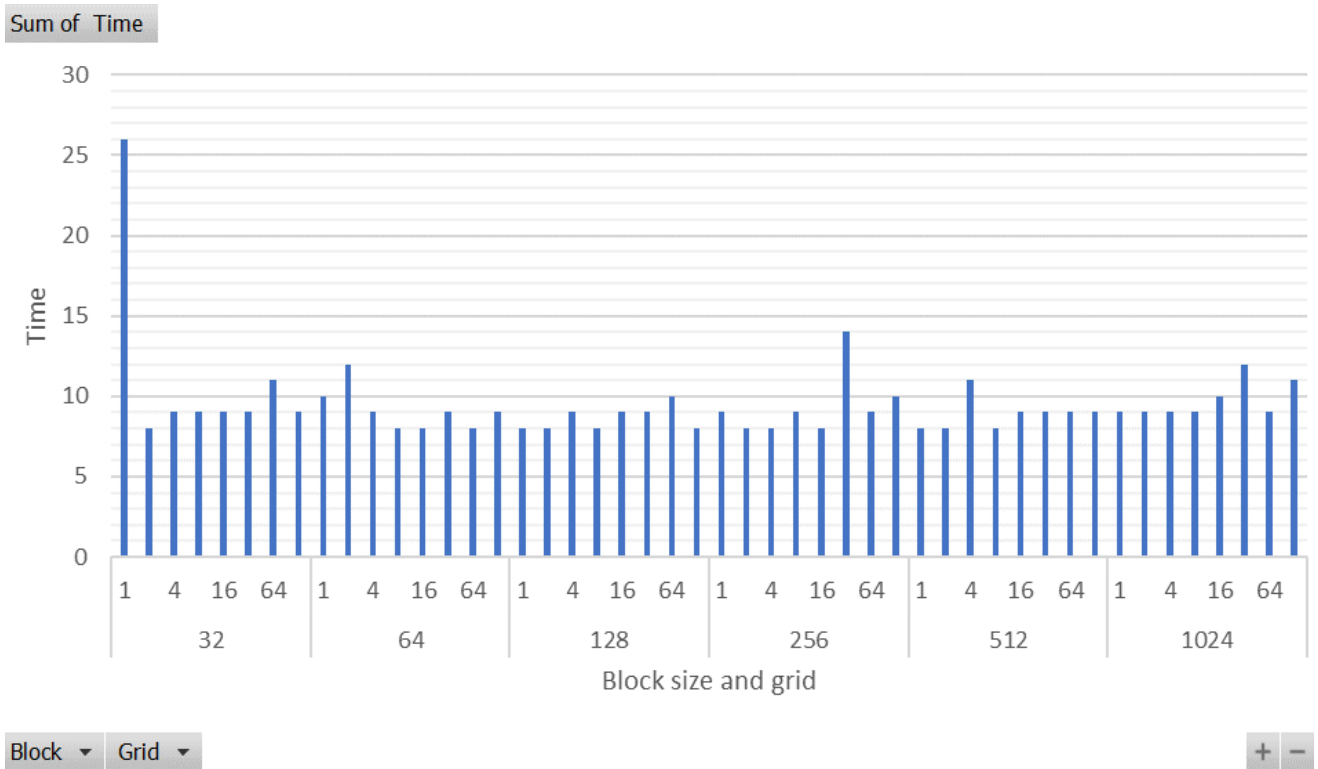
+ -

Observation:

When the grid size is kept constant and the block size is changed, the execution time increases exponentially.

GPU Naïve Results:

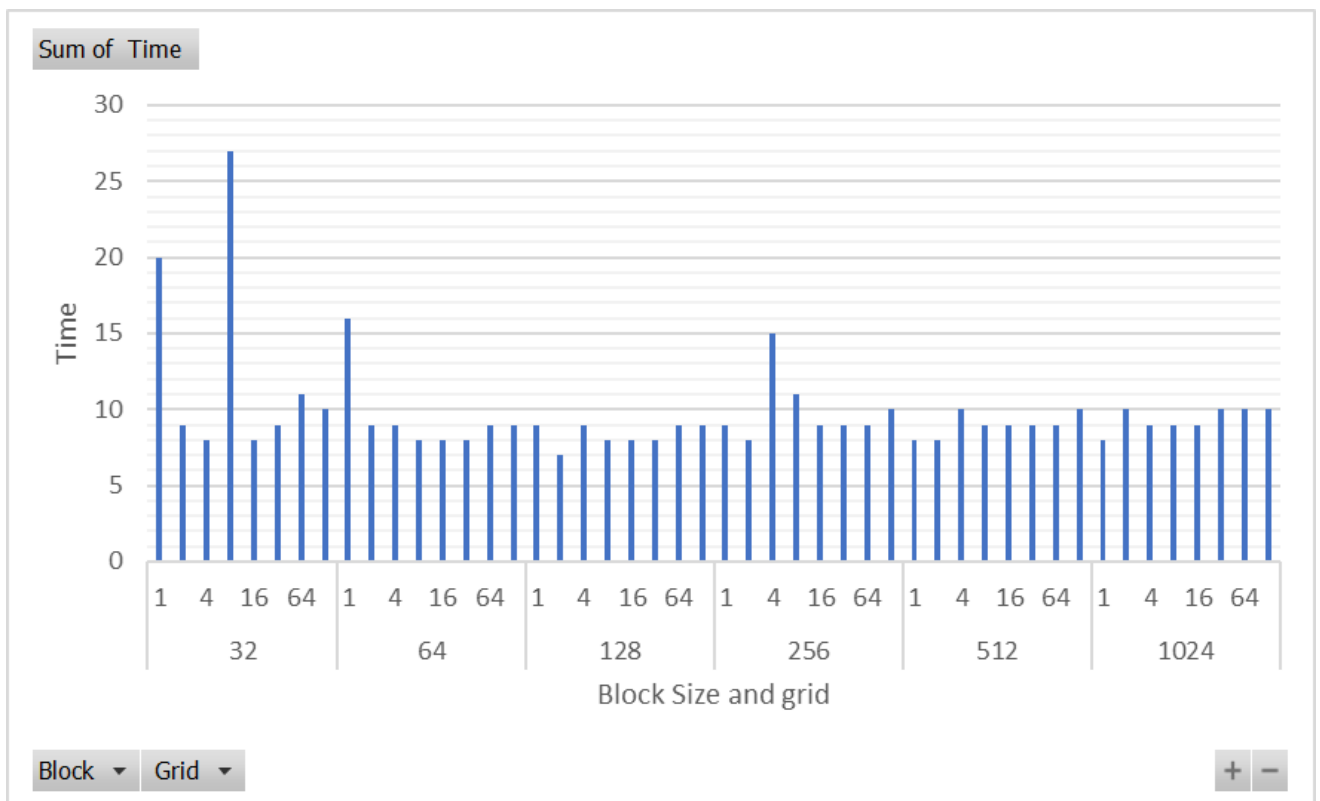
[illegible]



Observation:

When the grid size is maintained and the block size is changed, the time taken for execution almost remains the same.

GPU Optimization Results:

[illegible]

Observation:

When the grid size is maintained and the block size is changed, the time taken for execution almost remains the same.