Introduction to CUDA and OpenCL

Lab 3

Kacper Kapuściak

Introduction

On the third laboratory about implementing and measuring algorithms both on integer and floating point numbers. These algorithms were:

1. Matrix addition
2. Hadamard product
3. Dyadic product

I don’t provide the last one because I couldn’t figure out how to implement Dyadic product.

All operations were executed on Nvidia RTX 2060 graphics card.

Measurements

- Matrix addition of integer numbers on 1D grid

number of elements	CPU [µs]	GPU [µs]
100	3,2	328,1111111
10000	220,9	340,3
1000000	25621,7	3998,7
100000000	2735136,2	647255

- Matrix addition of float numbers on 1D grid

number of elements	CPU [µs]	GPU [µs]
100	4	279,5555556
10000	208,6	326,4
1000000	23047,1	3981,6
100000000	2534469,9	480443,2

- Matrix addition of integer numbers on 2D grid

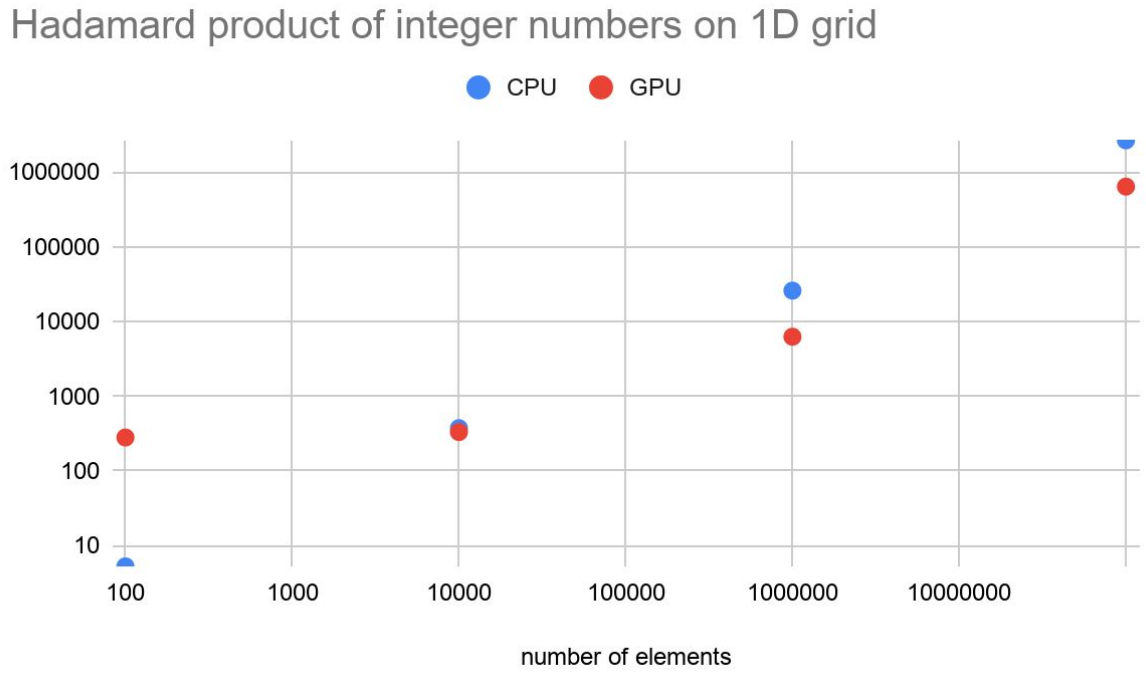
number of elements	CPU [µs]	GPU [µs]
100	3,2	282,5555556
10000	220,9	321,1
1000000	25621,7	3830,5
100000000	2735136,2	471492,5

- Matrix addition of float numbers on 2D grid

number of elements	CPU [µs]	GPU [µs]
100	4	264,3333333
10000	208,6	412,1
1000000	23047,1	3878,5
100000000	2534469,9	476060

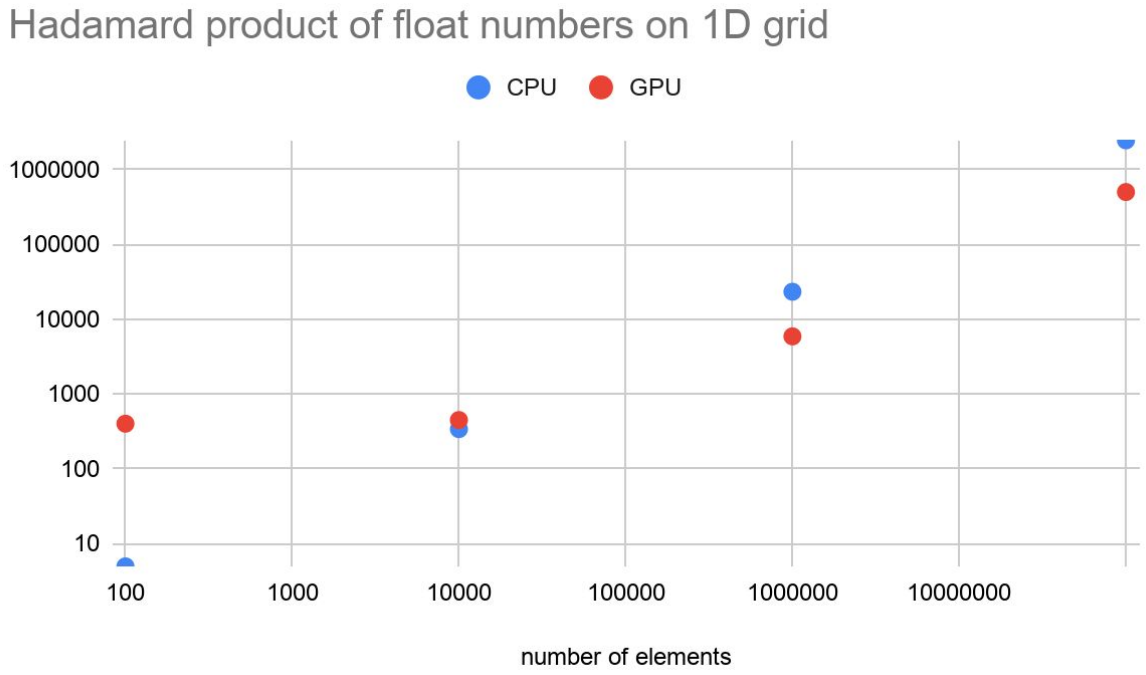
- Hadamarad product of integer numbers on 1D grid

number of elements	CPU [µs]	GPU [µs]
100	5,3	281,3333333
10000	374,9	332,7
1000000	25954,6	6275,3
100000000	2664056,2	640178,6



- Hadamarad product of float numbers on 1D grid

number of elements	CPU [µs]	GPU [µs]
100	5	402,5555556
10000	341,1	450,5
1000000	23301,9	5905,4
100000000	2451691,2	499974,5



- Hadamarad product of integer numbers on 2D grid

number of elements	CPU [µs]	GPU [µs]
100	5,3	398,2222222
10000	374,9	451,2
1000000	25954,6	5315,6
100000000	2664056,2	640712

- Hadamarad product of float numbers on 2D grid

number of elements	CPU [µs]	GPU [µs]
100	5,3	281,2222222
10000	374,9	323,2
1000000	25954,6	3807,3
100000000	2664056,2	509089,2

Conclusion

As we can see in my measurements when the number of elements is small making calculations on GPU is less efficient than on the CPU. That's not new for us, we've learned that in the report from our last laboratory. It's like that because sending data to and from graphics card takes a considerable amount of time. But when number of elements in the matrix is huge i.e. more than million we can see that the GPU starts to take the lead. For 100000000 and simple matrix addition the difference can be as big as 2 seconds! It's also interesting that addition and multiplication floating point numbers, both on CPU and GPU takes comparably less amount of time.