AGH University of Science and Technology

Final project report

Authors: Pedro Alexandre Simões dos Reis,
Iñaki Urrutia Sánchez

Subject: Introduction to CUDA and OpenCL
Year: 2019/2020

Contents

1 Introduction 2

1.1 How to use this library 2

1.2 About inverse matrix and dot product 2

2 Vector addition and subtraction 3

3 Matrix addition and subtraction 4

4 Matrix multiplication 5

1 Introduction

The goal of this report is to compare the behaviour and performance of the CPU and GPU using a library that supports the main matrix operations.

1.1 How to use this library

Please, find the instructions of the makefile of this project in the README.md file in this repository

1.2 About inverse matrix and dot product

Since inverse matrix and dot product operations do not work properly on large matrix, we do not have data enough to compare the CPU and GPU implementations.

2 Vector addition and subtraction

We found that the CPU implementation has a similar performance in small amounts of data, but as we increase the vector size, the GPU version becomes much faster than the CPU version.

Respect on the implementation of the vector subtraction functions, the behaviour is the same as it is in the addition version, because we are performing the same operation but with the opposite sign.

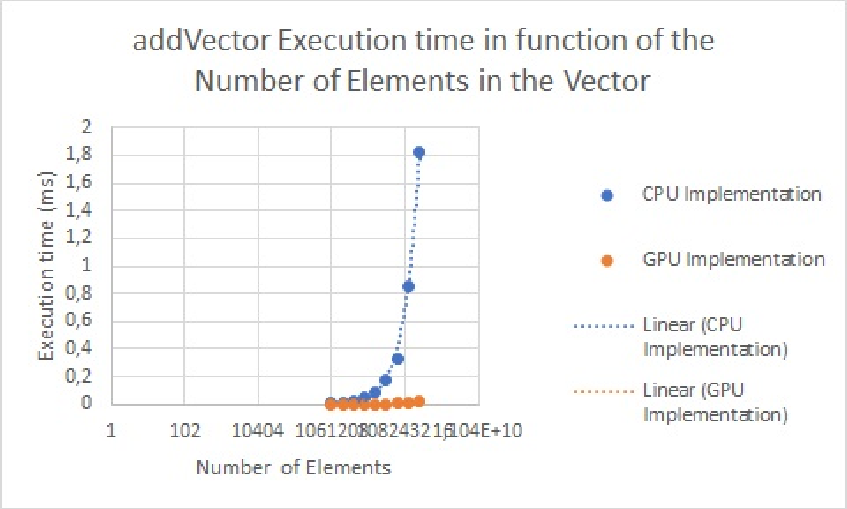

Figure 1

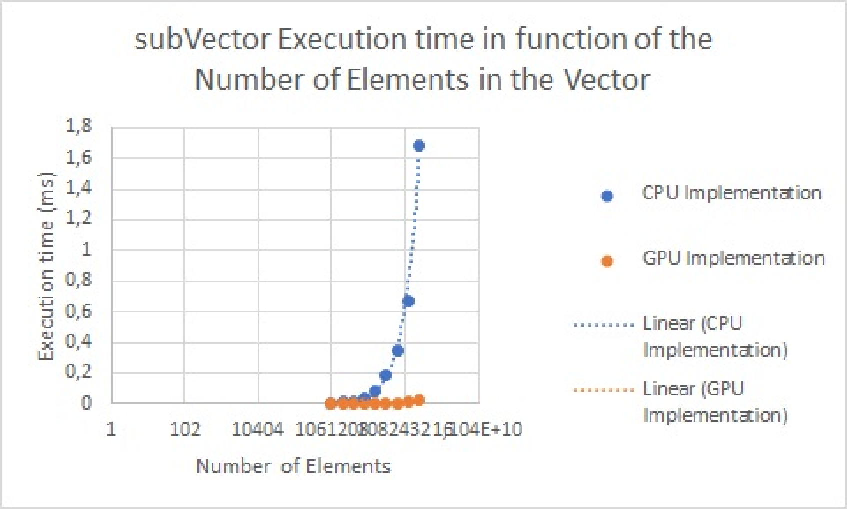

Figure 2

3 Matrix addition and subtraction

In case of the implementation of the operations of addition and subtraction of matrices, we transform each matrix into a single dimensional vector, with the goal of being able to apply the corresponding operation (subtraction/addition of vectors). Once we have the result, we undo the transformation to get the final matrix. As we use this technique, the performance is quite similar to the single dimensional operations we studied above.

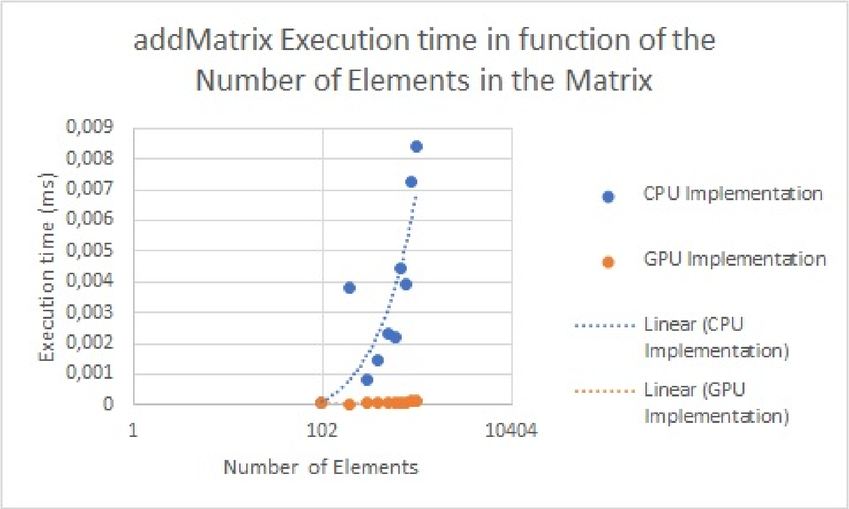

Figure 3

a

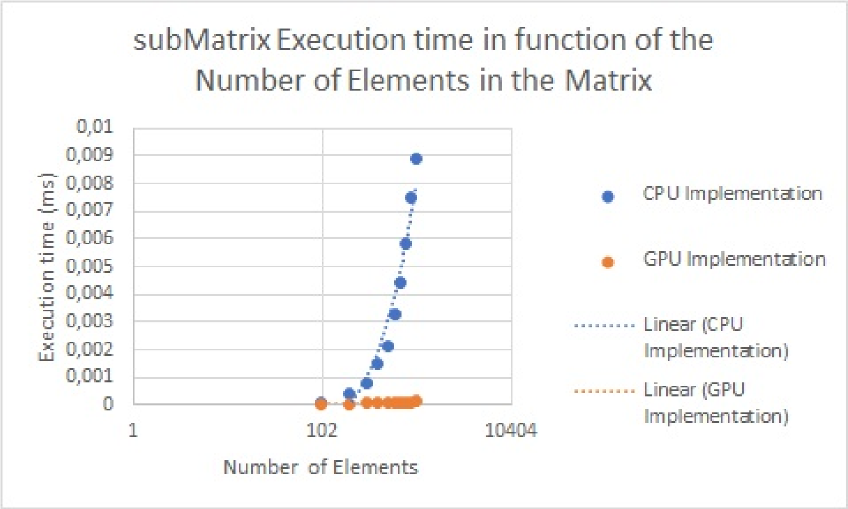

Figure 4

4 Matrix multiplication

Regarding the matrix multiplication operation, we see that the performance gap between the CPU's and GPU's version starts with smaller size of data. This means that the GPU implementation is worth it to use even on multiplication of small matrix, since we are gaining performance almost in every situation.

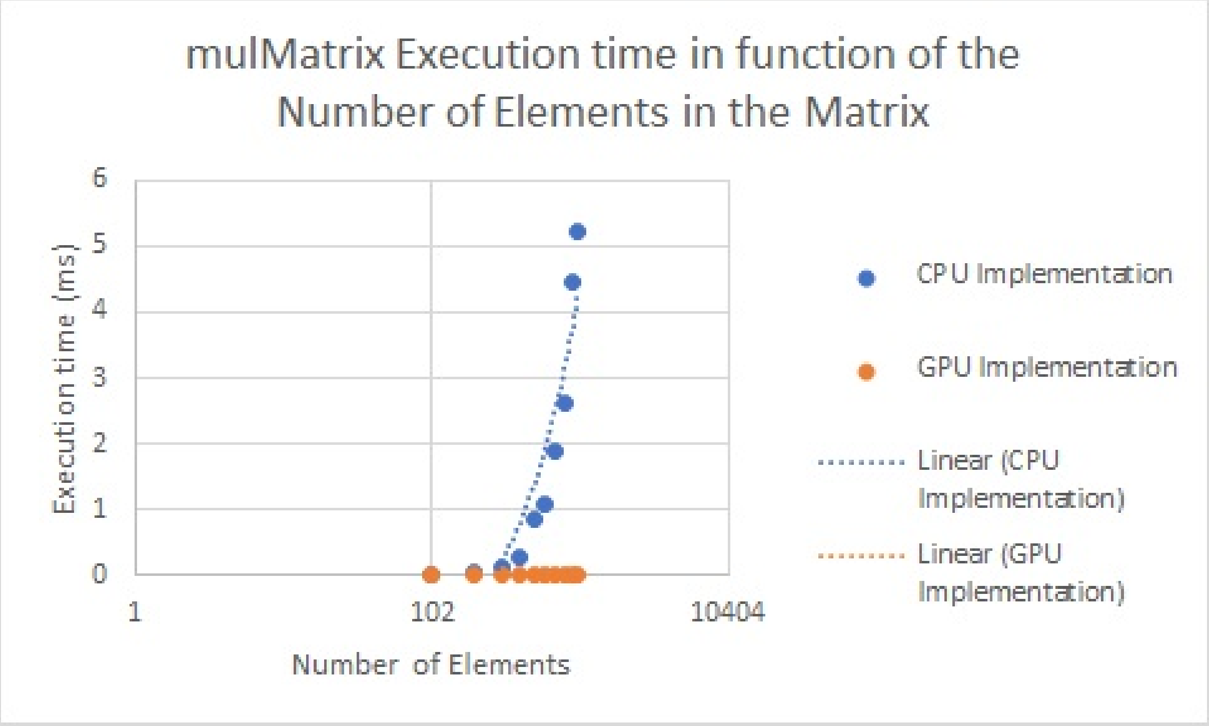

Figure 5