AGH University of Science and Technology

Laboratory report 7

Author: Pedro Alexandre Simões dos Reis

Subject: Introduction to CUDA and OpenCLYear: 2019/2020

Contents

1 Comparison between OpenCL and CUDA

1.1 Introduction

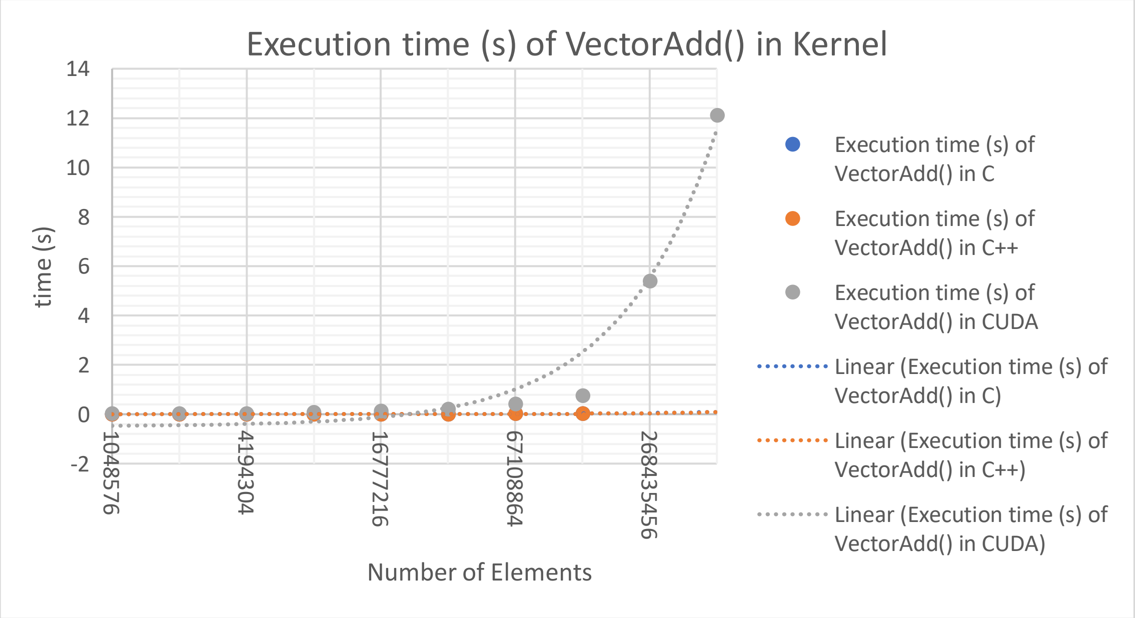
In this section will be discussed the performance of OpenCL compared with the performance of CUDA. To do this comparison it will be used an implementation of the Vector Addition problem, using OpenCL with both C and C++ as base programming languages and CUDA for compare the total execution time of the program as a whole and the execution time in the device.

1.2 Comparison between OpenCL and CUDA based on Kernel Execution Times

After running the implementation of the Vector Addition in C and C++ using OpenCL library and in CUDA, it was measured the execution time of the Kernels in each implementation.

Table 1. Kernel Execution Times of OpenCL in C and C++ and CUDA

Number of Elements	C	C++	CUDA
1048576	0,000326	0	0,01043
2097152	0,000636	0	0,01991
4194304	0,000939	0,001	0,034986
8388608	0,001716	0,002	0,063762
16777216	0,003099	0,003	0,119059
33554432	0,005831	0,005	0,213755
67108864	0,01367	0,011	0,405865
134217728	0,021723	0,021	0,759524
268435456			5,401798
536870912			12,11599



Graphic 1. Kernel Execution Times of OpenCL in C and C++ and CUDA

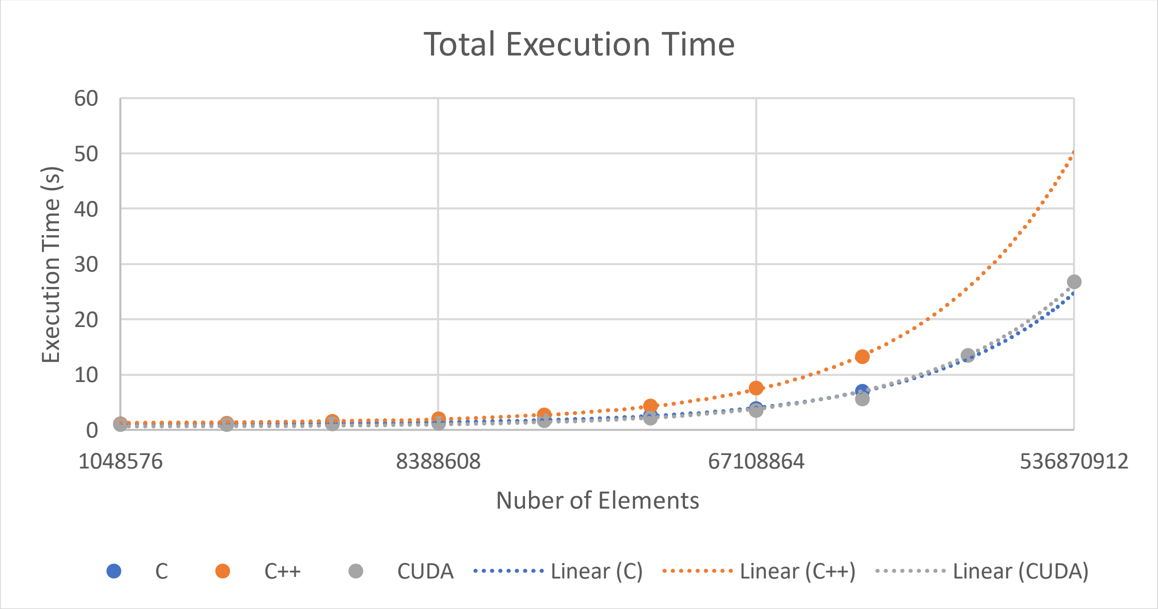
As it can be observed in both Table 1 and Graph 1, the Kernel execution times using OpenCL are smaller than the ones observed with CUDA. It is also possible to observe that when the number of elements in the vector increases the kernel execution time of the CUDA code increases linearly in a faster pace than the others.

1.3 Comparison between OpenCL and CUDA based on Total Execution Times

It was ran the code ran in the previous subsection but this time it was measured the total execution time.

Table 2. Total Execution times of OpenCL in C and C++ and CUDA

Number of Elements	C	C++	CUDA
1048576	1,058	1,087	0,973
2097152	1,066	1,232	1,013
4194304	1,223	1,525	1,084
8388608	1,432	2,019	1,217
16777216	1,766	2,768	1,625
33554432	2,561	4,345	2,094
67108864	3,919	7,52	3,476
1,34E+08	6,958	13,285	5,605
2,68E+08	Out of memmory	Out of memmory	13,434
5,37E+08	Out of memmory	Out of memmory	26,747



Graphic 2. Total execution times of OpenCL in C and C++ and CUDA

As it can be observed in both Table 2 and Graph 2, the total execution time of CUDA is lower than the others, even though the execution time of the C code with OpenCL is very close. This difference of the execution is more notorious when the number of elements in the vector increases.

1.4 Comparison between OpenCL and CUDA based on Kernel Execution Times with multiple Kernel Calls

In this case it was implemented some code in C and C++ using the OpenCL library and in CUDA to sum multiple vectors as indicated bellow:

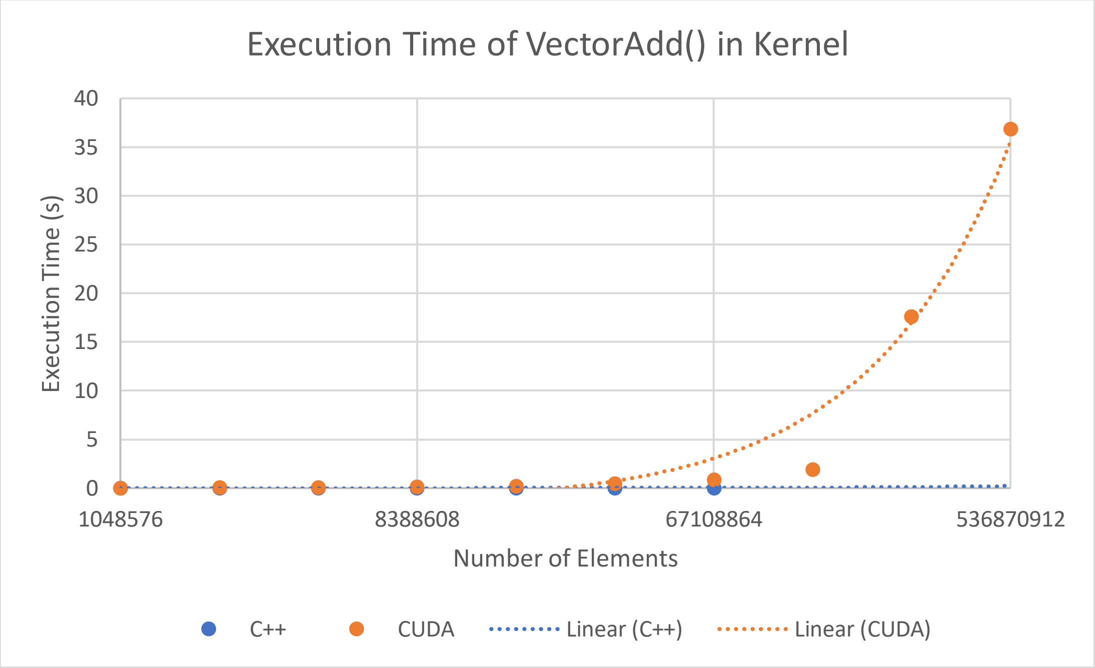
$$\vec{c} = \vec{a} + \vec{b}$$

$$\vec{d} = \vec{c} + \vec{e}$$

$$\vec{f} = \vec{d} + \vec{g}$$

Table 3. Total multi-kernel Execution times of OpenCL in C and C++ and CUDA

Number of Elements	**C++**	**CUDA**
1048576	0,005	0,022852
2097152	0,005	0,042504
4194304	0,007	0,08103
8388608	0,007	0,147924
16777216	0,008	0,231025
33554432	0,016	0,473548
67108864	0,032	0,881832
1,34E+08	Out of memory	1,889098
2,68E+08	Out of memmory	17,60611
5,37E+08	Out of memmory	36,86254



Graphic 3. Total multi-kernel Execution times of OpenCL in C and C++ and CUDA

As observed in Table 3 and Graphic 3, the kernel execution time in CUDA is higher then the C++ equivalent, as expected since the CUDA kernel take longer to execute.