



AGH

**AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY**

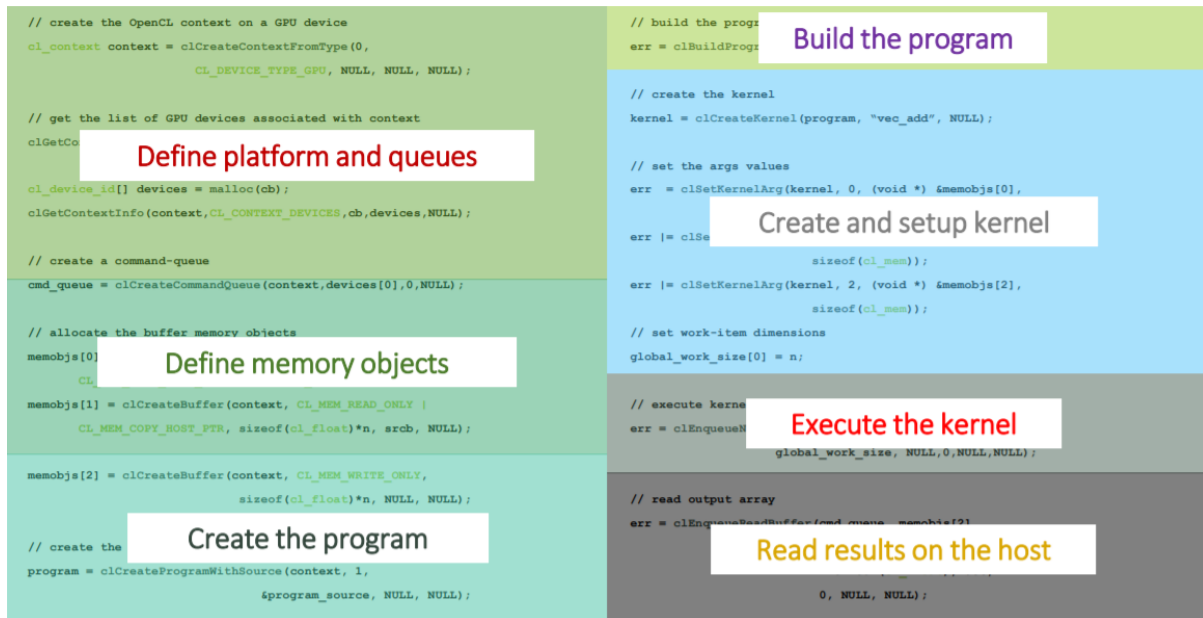
Introduction to CUDA and Open CL

Lab 6

Wiktor Żychowicz

1. OpenCL.

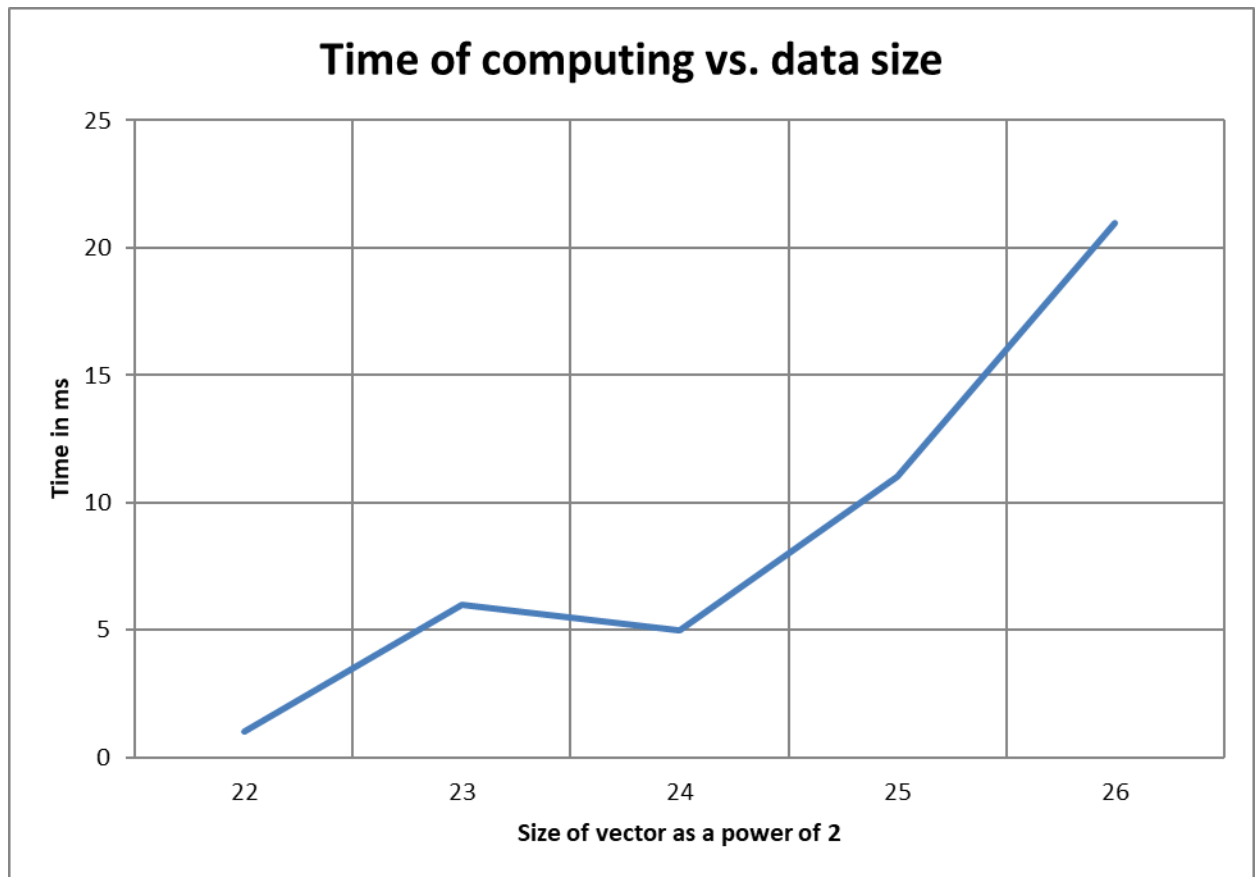
Open Computing Language is a tool which allows to use all compute devices in a system to accelerate your calculations. It can take advantage of CPUs ,GPUs, APUs and many others. Furthermore in principal adjustments in your source code prior different systems configurations are keeping as small as possible . This is biggest advantage of OpenCL but also it dictates high level of abstraction. Writing a OpenCL program reminds more of a instruction how to compile and execute code more than only write a simple program. Thanks to that we achieve universality and high percent of resource usage but we sacrifice opportunity to optimize our software in hardware specific way.



Picture 1: Source code of a typical OpenCL program with highlighted few additional parts required by higher abstraction level .

2. Vector adding.

Source code of discussed program can be obtained from GitHub(https://github.com/Sightster/CUDA_Gr_5) . It add few vectors together in a chain fashion. Vast majority of the code is dedicated to define a management layer for the application. However when we compile and run our code can be executed on any GPU (without wondering about things like a producer company). We can also switch to or add new types of compute devices in comparatively fast and easy way. Chart below represent how program perform:



Picture 2: Vectors add performance.

As we can see Vector adding scale in massive parallel fashion which cannot be achieved on a CPU without using multiple threads. This prove usage of a GPU. Chart is somehow distorted what indicate that data management is still a main problem when using more than one compute device.

3. OpenCL vs CUDA.

Finally we should compare this two technologies. Main difference from user perspective is that OpenCL supports wider range of hardware accelerators and does not force you to use NVIDIA environment (software and hardware wise). On the other hand CUDA has better support in terms of development tools (you get IDE implementation and graphical profiler).

In terms of performance CUDA is faster and give you vastly more opportunity to optimize your applications but you can not use any other accelerator than NVIDIA's GPU. Writing CUDA program is also easier because there is no need to define so big management layer. Developer can focus on a compute part of the task.