

Tommi Järvenpää
2553157
Hours used: Approximatetly 4.5

Assignment 1 Report

During the assignment, I installed the following:

- Microsoft Visual C++ (MSVC) C/C++ compiler
- Visual Studio 2019 Integrated Development Environment (IDE) (<https://visualstudio.microsoft.com/vs/>)
- Intel OpenCL Software Development Kit (SDK) (<https://software.intel.com/content/www/us/en/develop/tools/opencl-sdk.html>)
- Intel OpenCL Central Processing Unit (CPU) Runtime (<https://software.intel.com/content/www/us/en/develop/articles/opencl-drivers.html#proc-graph-section>)
- clinfo (<https://github.com/Oblomov/clinfo>)

The above software was selected because my computer has an Intel CPU and an integrated Intel Graphics Processing Unit (GPU). While my computer has an NVIDIA GPU, I opted to just use the integrated one, so I don't have to install the CUDA SDK. The 2019 version of Visual Studio was already installed, so I decided to use it instead of installing an older version.

Description of installed tools:

MSVC [1] is Microsoft's compiler for C and C++. It was originally released with the name Microsoft C/C++ in 1993. While it previously was its own product, it now comes bundled with Visual Studio installations. It provides tools for code debugging and development.

Visual Studio is an customizable IDE created by Microsoft. It support the development of applications for Windows, Mac, Android, and web [2]. It also allows developers to debug their code and use Git with the IDE.

As stated in the Assignment description, an SDK had to be installed so OpenCL programs could be created. Since my computer has an Intel CPU, I installed Intel's OpenCL SDK [3]. The SDK supports the development, debugging, and resource analysis of OpenCL applications. In addition to enabling the usage of the CPU with OpenCL, it also allows Intel GPUs to be used with OpenCL as well. The OpenCL runtime for the CPU also had to be installed so everything would work properly. The GPU runtime seemed to already exist on my computer without any additional installations. The SDK also made some modifications to Visual Studio. For example, it added an empty example OpenCL project and an example CPU and GPU project.

In an attempt to make sure that the CPU and GPU are correctly set up, clinfo [4] was installed. It is a command-line application, which enumerates all OpenCL capable devices on the computer.

After installing all of the above software, I built the CPU and GPU example OpenCL project provided by Intel's SDK. Both of them ran correctly. Interestingly, the GPU code also worked for my NVIDIA GPU, even though I have never explicitly installed the CUDA SDK.

OpenCL:

OpenCL [5, 6] is an open standard for parallel programming with all kinds of heterogeneous systems. It was originally developed by Apple, but now the developers are the Kronos Group. It uses different devices, i.e., CPUs and GPUs for computation. OpenCL programs are created with C, and the functions ran on the devices are called kernel functions. It allows parallel computation to

be used on the devices, and has a four-level memory hierarchy. This hierarchy consists of global, read-only, local, and per-element memory.

References:

- [1] Microsoft Visual C++. URL: https://en.wikipedia.org/wiki/Microsoft_Visual_C%2B%2B
- [2] Visual Studio. URL: <https://visualstudio.microsoft.com/>
- [3] Intel SDK For OpenCL.
URL: <https://software.intel.com/content/www/us/en/develop/tools/oneapi-sdk.html>
- [4] Github – clinfo. URL: <https://github.com/Oblomov/clinfo>
- [5] OpenCL Overview. URL: <https://www.khronos.org/opencl/>
- [6] OpenCL. URL: <https://en.wikipedia.org/wiki/OpenCL>