GPU Computing Course – Jan Lemeire

Lesson 1: The Power of GPUs

History of GPGPUs

A GPU is a graphics processing unit. Its main purpose is to show images on the screen and to carry out the necessary computations to do so. During the last 20 years the computational power of GPUs has increased enormously, and a number of people started to look at the possibility to use the *GPU for general purpose computations*, hence the term **GPGPU**. Initially it was necessary to cast these general-purpose problems into a graphic framework i.e. it was necessary to write the code using for example OpenGL or DirectX, APIs used to program computer graphics. This approach was very cumbersome, and efforts were made to design and implement ways to ease the creation of general-purpose programs for the GPU.

In November 2006 the first commercially available framework for general purpose computing appeared: **CUDA** from NVIDIA. CUDA stands for ``Compute Unified Device Architecture'' and presents itself as a set of extensions to the C programming language and a runtime library. CUDA is proprietary and can only be used with NVIDIA GPUs.

OpenCL is an open standard that provides a framework for writing programs that execute across heterogeneous platforms. A heterogeneous platform is basically defined as a platform that contains more than one computing device, for example a CPU and a GPU. OpenCL was initially developed by Apple, but it has become an open standard that is maintained by the non-profit technology consortium Khronos Group, that also maintains OpenGL. Version 1.0 of OpenCL was released in 2008. At the moment of writing, OpenCL is at version 2.1 and there are several commercial implementations available from for example NVIDIA, AMD, Intel and IBM.

Computational performance

While a CPU can only execute a few computations at the same time, a GPU has a lot of **Scalar Processes** (SPs), called **Processing Elements** (PEs) in OpenCL and called CUDA Cores by Nvidia. The (theoretical) peak performance can then be calculated by multiplying this number with the clock frequency:

```
Operations per second = PEs x clock frequency          (1)
```

These PEs refer to Single Precision (**SP**) computational units.

A GPU is a collection of **Compute Units** (CUs), called **MultiProcessors** (MPs) or **Streaming Multiprocessors** (SMs) by Nvidia. The layout of a CU depends on its **architecture**. The number of PEs for a CU is fixed for each architecture. The performance can thus also be calculated as:

```
Operations per second = CUs x PEperCU x clock frequency     (2)
```

Native transcendental functions (like cos or pow) are executed on a special **SFU** unit. Also double precision data is calculated by specific **DP** units. With the number of such units on a CU, the peak performance can be calculated by Equation 2.

The website https://www.techpowerup.com/gpu-specs/ provides you the numbers for your GPU.

The following table lists the *characteristics of a CU* for the Nvidia architectures. Since the exact number of CUs depends on the specific GPU, the peak performance cannot be mentioned here. We also mention the theoretical RAM bandwidth and the estimated SP completion latency (discussed later)

Architecture	Clock freq MHz	PEs per CU	SFU	DP	RAM band-width (GBs)	Λ_{SP}
Tesla		8	?	-		24
Fermi	1147	32	8	-	144	18
Kepler	1032	192	32	64	86	9
Maxwell	1058	128	32			6
Pascal	1506	128	32	64	192	6
Turing		64	8	?		