Introduction to GPU programming: When and how to use GPU-acceleration?

Mirko Myllykoski
mirkom@cs.umu.se

Department of Computing Science / HPC2N
Umeå University

5 November 2019

SNIC UMEÅ UNIVERSITY HPC2N

Course description

- GPU-acceleration has been shown to provide significant performance benefits in many different applications.
- However, for a novice, or even for a moderately experienced scientist or programmer, **it is not always clear which applications could potentially benefit from GPU-acceleration and which do not**.
- For example, a Nvidia V100 GPU can perform artificial intelligence (AI) related computations in a fraction of the time it takes a regular CPU to perform the same computations but ill-informed OpenACC compiler pragma can actually make a code run slower.

Course description

Questions to answer:

- Why is this?
- When should one invest time in GPU-acceleration?
- How much speedup can be expected with a given application?

Purpose:

- The main goal of this one day course is to *start answering these questions*.
- The course also
  - covers the *basics of GPU programming* and
  - aims to provide the necessary *information for avoiding the most common pitfalls*.

Requirements: The course does not require any existing GPU programming knowledge but basic understanding of the C language is required for the hands-ons.

Course outline

- Introduction to HPC2N and Kebnekaise (Birgitte)

Course outline

- Introduction to HPC2N and Kebnekaise (Birgitte)
- GPU hardware and CUDA basics
  - Hello world, CUDA cores, threads, thread blocks, kernels, memory spaces (global and shared), memory transfers, streams, ...

Course outline

- Introduction to HPC2N and Kebnekaise (Birgitte)
- GPU hardware and CUDA basics
  - Hello world, CUDA cores, threads, thread blocks, kernels, memory spaces (global and shared), memory transfers, streams, ...
- Where is my performance?
  - Flops, bandwidth, arithmetical intensity, roofline model, things not to do, ...