P. Hadjidoukas, C. Papadimitriou
ETH Zentrum, HG E5/G3
CH-8092 Zürich

Set 8 - GPUS

Issued: April 24, 2017

Question 1: GPU - Hello World

We will do a little vector addition to get started on the GPU.

a) Get access to a computer with NVIDIA CUDA. If your laptop/machine has a NVIDIA GPU you may install the CUDA toolkit on your own machine[1].

b) Complete the `Makefile` in `code08/q1/` to compile the given code with the NVIDIA CUDA compiler.

   For now the program will perform a vector addition on the CPU and do nothing on the GPU. It will then compare the results of the CPU and the GPU. As you do nothing on the GPU, running the program will cause many error messages.

c) Complete the empty CUDA kernel in `vector_addition.cu` and add the kernel call in the `main()` function to perform a vector addition on the GPU.

   Run the program on a computer with an NVIDIA GPU. The error messages should be gone.

   Note: Beware of out-of-bounds accesses.

Question 2: Diffusion on GPUs

In this exercise we will port the well known 2D diffusion using stencils on GPUs.

a) Implement a kernel for the propagation using only global memory. You may copy the serial CPU code `diffusion2d_serial.cpp` we provided in `code08/q2/` to `diffusion2d_cuda.cu` as a skeleton. The function `PropagateDensity()` should call the CUDA kernel instead of doing the computation. Adapt the other functions of the class to minimize the data transfers between the CPU and the GPU.

   Hint: `threadIdx` and `BlockIdx` offer more than one dimension.

b) Check your results by comparing to the serial code and report the achieved speed compared to the given OpenMP implementation `diffusion2d_openmp.cpp`.

[1] `https://developer.nvidia.com/cuda-downloads`