CUDA exercises

Dorothea vom Bruch

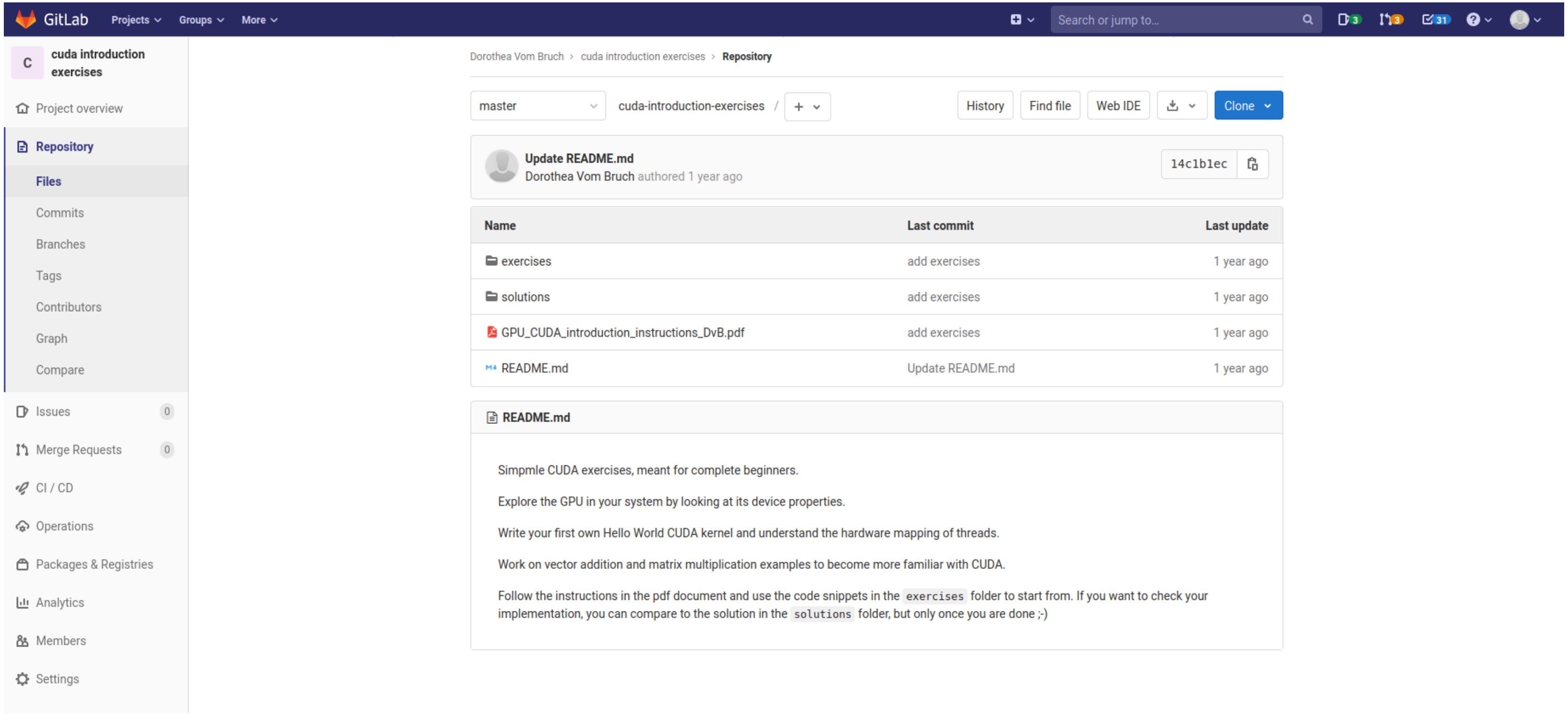
GDR-InF Annual workshop

GPU Hands-on project

October 2020

Where to find them

https://gitlab.cern.ch/dovombru/cuda-introduction-exercises



Update README.md
Dorothea Vom Bruch authored 1 year ago
14c1b1ec

Name	Last commit	Last update
exercises	add exercises	1 year ago
solutions	add exercises	1 year ago
GPU_CUDA_introduction_instructions_DvB.pdf	add exercises	1 year ago
README.md	Update README.md	1 year ago

README.md

Simpmle CUDA exercises, meant for complete beginners.

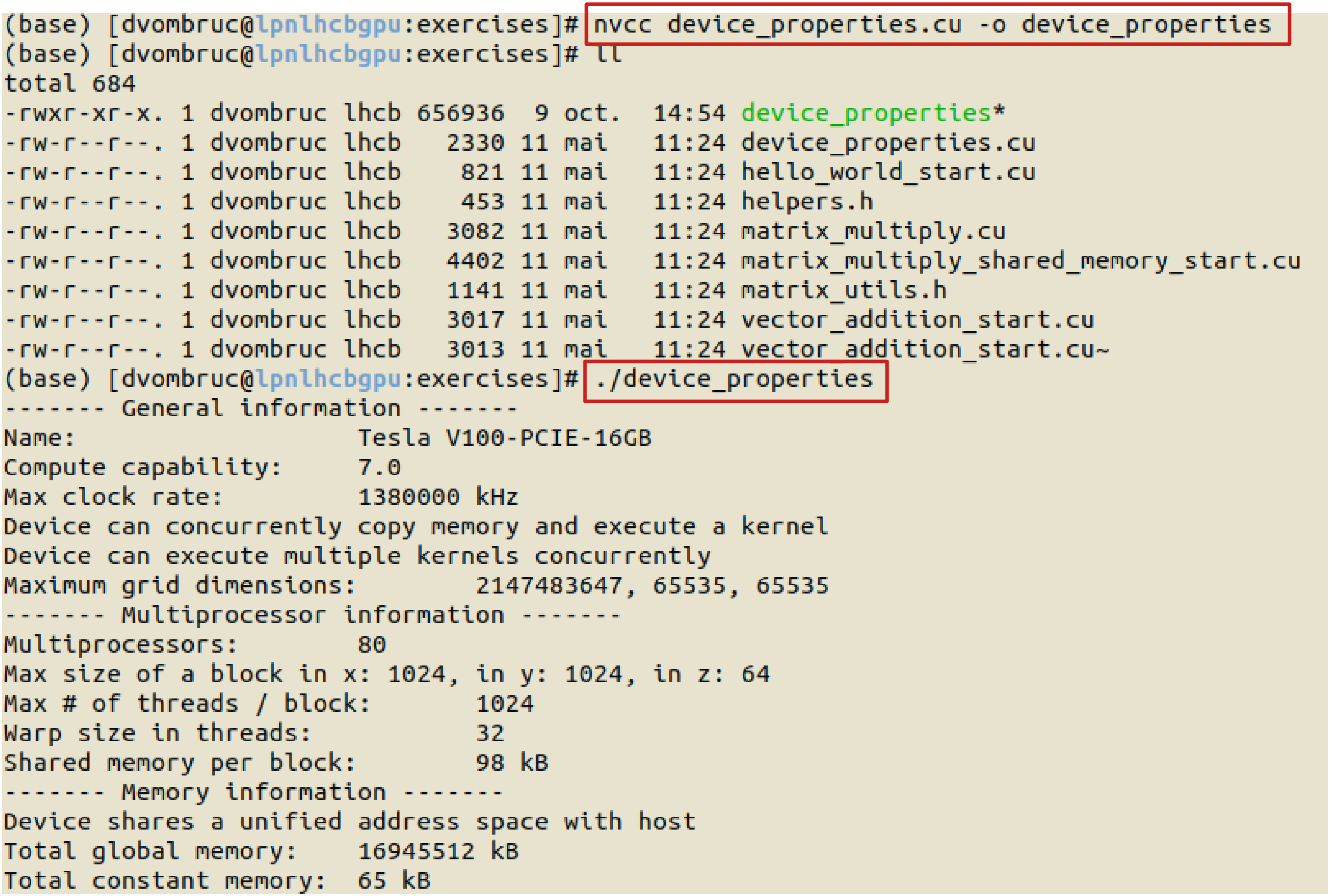
Explore the GPU in your system by looking at its device properties.

Write your first own Hello World CUDA kernel and understand the hardware mapping of threads.

Work on vector addition and matrix multiplication examples to become more familiar with CUDA.

Follow the instructions in the pdf document and use the code snippets in the `exercises` folder to start from. If you want to check your implementation, you can compare to the solution in the `solutions` folder, but only once you are done ;-)

How to compile and run a CUDA program



```
(base) [dvombruc@lpnlhcbgpu:exercises]# nvcc device_properties.cu -o device_properties
(base) [dvombruc@lpnlhcbgpu:exercises]# ll
total 684
-rwxr-xr-x. 1 dvombruc lhcb 656936  9 oct.  14:54 device_properties*
-rw-r--r--. 1 dvombruc lhcb   2330 11 mai   11:24 device_properties.cu
-rw-r--r--. 1 dvombruc lhcb    821 11 mai   11:24 hello_world_start.cu
-rw-r--r--. 1 dvombruc lhcb    453 11 mai   11:24 helpers.h
-rw-r--r--. 1 dvombruc lhcb   3082 11 mai   11:24 matrix_multiply.cu
-rw-r--r--. 1 dvombruc lhcb   4402 11 mai   11:24 matrix_multiply_shared_memory_start.cu
-rw-r--r--. 1 dvombruc lhcb   1141 11 mai   11:24 matrix_utils.h
-rw-r--r--. 1 dvombruc lhcb   3017 11 mai   11:24 vector_addition_start.cu
-rw-r--r--. 1 dvombruc lhcb   3013 11 mai   11:24 vector_addition_start.cu~
(base) [dvombruc@lpnlhcbgpu:exercises]# ./device_properties
------- General information -------
Name:                   Tesla V100-PCIE-16GB
Compute capability:     7.0
Max clock rate:         1380000 kHz
Device can concurrently copy memory and execute a kernel
Device can execute multiple kernels concurrently
Maximum grid dimensions:        2147483647, 65535, 65535
------- Multiprocessor information -------
Multiprocessors:        80
Max size of a block in x: 1024, in y: 1024, in z: 64
Max # of threads / block:       1024
Warp size in threads:           32
Shared memory per block:        98 kB
------- Memory information -------
Device shares a unified address space with host
Total global memory:    16945512 kB
Total constant memory:  65 kB
```

Exercises

- Run nvidia-smi on your machine: What GPUs are available?
- Run device query on your machine: What are the specs of your GPUs?
    - You can use the device properties script within the exercises gitlab repo, if you don’t want to compile the CUDA samples

Follow the instructions in the PDF of the gitlab repo:

- The “exercises” directory contains code snippets with commented instructions to follow
- Your first Hello World CUDA program → get a feeling for parallelization
- Vector addition → do actual work in parallel
- Matrix multiplication → slightly more advanced
    - Suggest to try the single thread version, multiple threads & multiple threads and blocks
    - The shared memory option is more advanced

- Note: when measuring the speed of your program:
    - Make sure you are the only one using the GPU
    - You can check with nvidia-smi
    - If you know there are multiple users on the same GPU, talk to each other :)