Lesson 06: alpaka Workflow

Based on CMake

- Starting tomorrow, the lectures will become more detailed
- We will start to do actual work with alpaka
- alpaka is usually used with an up-to-date CMake (>= 3.15)
- All examples in the upcoming lectures will require CMake >= 3.15
Lesson 06: alpaka Workflow

No CMake available?

- Special stand-alone headers available. Example for CUDA GPUs:
  ```cpp
  #include <alpaka/standalone/GpuCudaRt.hpp>
  ```

- Beware: **You** will have to guarantee for all dependencies, compatibilities and compiler-specific flags!
Lesson 06: alpaka Workflow

alpaka and cupla

- alpaka: *Abstraction Library for Parallel Kernel Acceleration*
  - [https://github.com/alpaka-group/alpaka](https://github.com/alpaka-group/alpaka)
  - Focus of the lectures

- cupla: *C++ User Interface for the Platform Independent Library Alpaka*
  - [https://github.com/alpaka-group/cupla](https://github.com/alpaka-group/cupla)
  - Thin layer over alpaka
  - CUDA-style API, intended for porting existing CUDA codes
  - Not covered here
Lesson 06: Lecture Workflow

Slides and examples

- All slides will be uploaded after the corresponding lecture took place
- We will upload the slides to here: https://github.com/alpaka-group/alpaka-workshop-slides
- The lecture sessions are recorded. We will send around the links to the videos once we obtained them from the video conference system.
- Examples can be found here: https://github.com/alpaka-group/alpaka-workshop-examples
Lesson 06: Lecture Workflow

Cheat sheet

• A cheat sheet / FAQ is available here: https://github.com/alpaka-group/alpaka-workshop-slides/tree/master/cheatsheet