alpaka Parallel Programming – Online Tutorial

Lecture 10 – The alpaka Programming Model

Lesson 12: Introducing Parallelism

CASUS
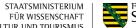CENTER FOR ADVANCED
SYSTEMS UNDERSTANDING

www.casus.science

Lesson 12: Introducing Parallelism

Executing Hello World

- In your `build` directory, run the generated executable:

  `./helloWorld`

- Windows: Look at the `Release` folder!

Lesson 12: Introducing Parallelism

Results

- Example output:

```
Hello, World from alpaka thread 2!
Hello, World from alpaka thread 3!
Hello, World from alpaka thread 4!
Hello, World from alpaka thread 7!
Hello, World from alpaka thread 0!
Hello, World from alpaka thread 1!
Hello, World from alpaka thread 6!
Hello, World from alpaka thread 5!
```

- Unspecified thread order

Lesson 12: Introducing Parallelism

Executing Hello World

- Run the generated executable again:
  ```
  ./helloWorld
  ```

Lesson 12: Introducing Parallelism

Results

- Example output:

```
Hello, World from alpaka thread 2!
Hello, World from alpaka thread 3!
Hello, World from alpaka thread 5!
Hello, World from alpaka thread 6!
Hello, World from alpaka thread 7!
Hello, World from alpaka thread 4!
Hello, World from alpaka thread 0!
Hello, World from alpaka thread 1!
```

- Different thread order but we did not change the code!

Lesson 12: Introducing Parallelism

Thread parallelism

- alpaka spawns a user-specified number of Threads

- Threads are executed in parallel
  → Threads can run at the same time

- Thread scheduling is nondeterministic (to the user)
  → Order of access to shared resources (memory, input/output, …) not specified