Overview
PRACE OpenACC Course 2021

27-29 October 2021 | Andreas Herten | Forschungszentrum Jülich, Jülich Supercomputing Centre

JÜLICH Forschungszentrum
JÜLICH SUPERCOMPUTING CENTRE

Jülich Supercomputing Centre

- Forschungszentrum Jülich
- Part of Institute for Advanced Simulation (IAS), Gauss Centre for Supercomputing (GCS), PRACE, …
- Operates supercomputers and connected infrastructure
- Researches in next-gen supercomputers
- Supports applications leveraging machines
- Supercomputers
  - JUWELS Cluster, JUWELS Booster
  - JURECA/JURECA DC, JURECA Booster
  - DEEP
  - JUSUF

Jülich Supercomputing Centre

- Forschungszentrum Jülich
- Part of Institute for Advanced Simulation (IAS), Gauss Centre for Supercomputing (GCS), PRACE, …
- Operates supercomputers and connected infrastructure
- Researches in next-gen supercomputers
- Supports applications leveraging machines
- Supercomputers
  - **JUWELS Cluster, JUWELS Booster**
  - **JURECA/JURECA DC**, JURECA Booster
  - **DEEP**
  - **JUSUF**

OpenACC Course

About

- Since 2014
- Usually a course in meat-space
- Interactive course – many hands-ons 👐
- There are other *many-core* courses
  - CUDA: April 2021
  - → See JSC training program

Tutors

Kaveh Haghighi-Mood
Application-Oriented Technology Development, JSC

Andreas Herten
NVIDIA Application Lab at Jülich, JSC

Jiri Kraus
NVIDIA Application Lab at Jülich, NVIDIA

Thorsten Hater
High Performance Computing in Neuroscience, JSC

Markus Hrywniak
NVIDIA Application Lab at Jülich, NVIDIA

Timetable

- Every day from 9:00 until 13:00
- 30 minute break around 10:45

Day 1 Basics
- Welcome
- GPU Introduction
- OpenACC Introduction

Day 2 Tools, Interoperability
- Debugging
- Profiling, Performance Optimization
- Interoperability

Day 3 Multi-GPU

More Technicalities

- Supercomputer for this course: **JUWELS Booster**
- Infrastructure for tasks
  - Already done for *Profound Proverb*
  - Jupyter-JSC: `https://jupyter-jsc.fz-juelich.de`
  - Project: `training2124`
  - Remember to source the environment (tasks rely on environment variables): `source $PROJECT_training2124/env.sh`
- Tasks
  - Sorted by session
  - Solutions are always given, you decide how long you tinker before peaking into solutions (Hint: The longer, the more benefit you will get from this course!)
  - Re-sync once might be needed now, call `jsc-material-sync`

Let’s Get Started!

Questions?

Let’s Get Started!

Questions!

My favorite programming language is:

C	C++	Fortran	Python
Java	JavaScript	Julia	Haskell
Go	Rust	Bash	Assembly

I’ve used OpenMP before

Yes	No

I’ve used MPI before

Yes	No

I’ve used a GPU before

Yes	No

I programmed a GPU before

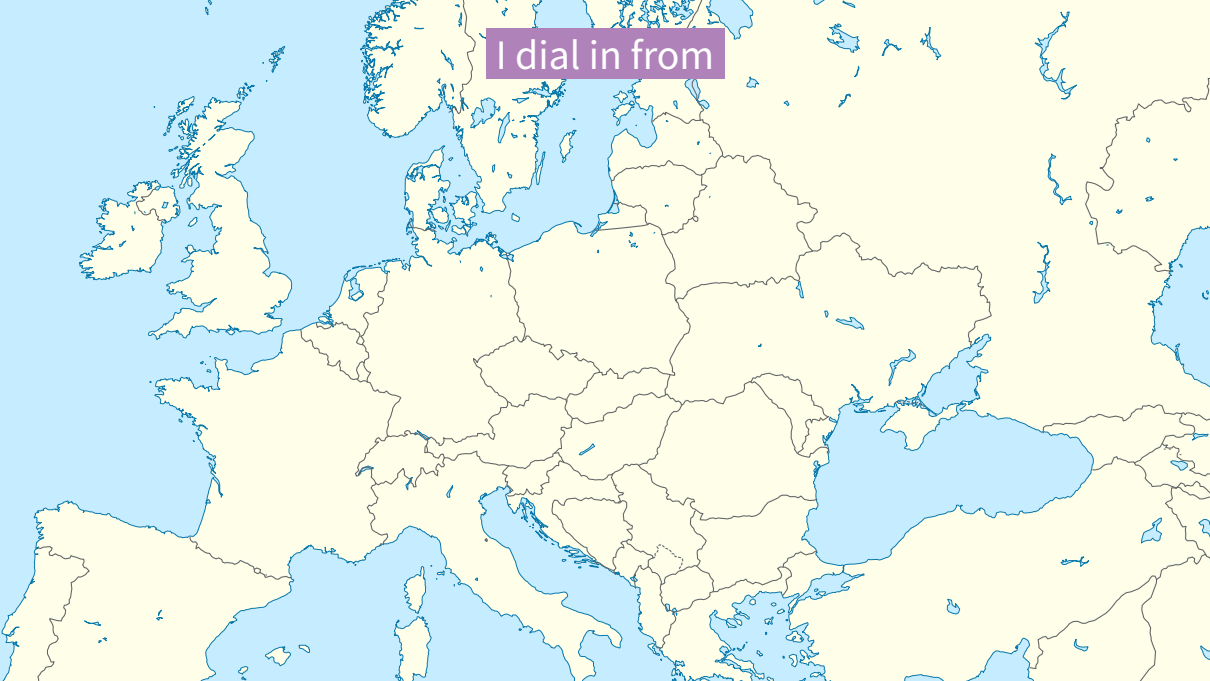
Yes	No

I programmed a GPU before

Other SYCL, HIP, OpenGL, …	OpenACC	No
OpenCL	CUDA	

I currently work from home

Yes	No


I dial in from

I dial in with

Linux	macOS
Windows	other