



Overview JSC OpenACC Course 2024

29-31 October | Andreas Herten | Forschungszentrum Jülich, Jülich Supercomputing Centre

Jülich Supercomputing Centre

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

Jülich Supercomputing Centre

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

OpenACC Course

About

- Since 2014
- Interactive course – many hands-ons 
- There are other *many-core* courses
 - CUDA: 2025
 - → See JSC [training program](#)

Tutors



Kaveh Haghghi-Mood

Accelerating Devices
Lab, JSC



Andreas Herten

Accelerating Devices
Lab, 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 Interoperability, Multi-GPU

More Technicalities

- Supercomputer for this course: **JURECA-DC**
- Infrastructure for tasks
 - Already done for Poem
 - Jupyter-JSC: <https://jupyter-jsc.fz-juelich.de>
 - Project: training2440
 - Remember to source the environment (tasks rely on environment variables):
`source $PROJECT_training2440/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	
OpenCL	CUDA	No

I currently work from home

Yes

No



I dial in from

Course Conclusion

Course Wrap-Up

Adventure Across OpenACC Land

- GPU Introduction
- OpenACC Introduction
- Debugging
- Profiling, Performance Optimization
- Interoperability
- Multi-GPU

Course Wrap-Up

Adventure Across OpenACC Land

- GPU Introduction
- OpenACC Introduction
- Debugging
- Profiling, Performance Optimization
- Interoperability
- Multi-GPU
- Download material via Jupyter
`tar -czf prace-openaccourse-2024.tar.gz -C $HOME GPU-Course`

Course Wrap-Up

Adventure Across OpenACC Land

- GPU Introduction
- OpenACC Introduction
- Debugging
- Profiling, Performance Optimization
- Interoperability
- Multi-GPU
- Download material via Jupyter
`tar -czf prace-openaccourse-2024.tar.gz -C $HOME GPU-Course`

Course Wrap-Up

Adventure Across OpenACC Land

- GPU Introduction
 - OpenACC Introduction
 - Debugging
 - Profiling, Performance Optimization
 - Interoperability
 - Multi-GPU
-
- Download material via Jupyter
`tar -czf prace-openaccourse-2024.tar.gz -C $HOME GPU-Course`
 - Certificates will be sent