

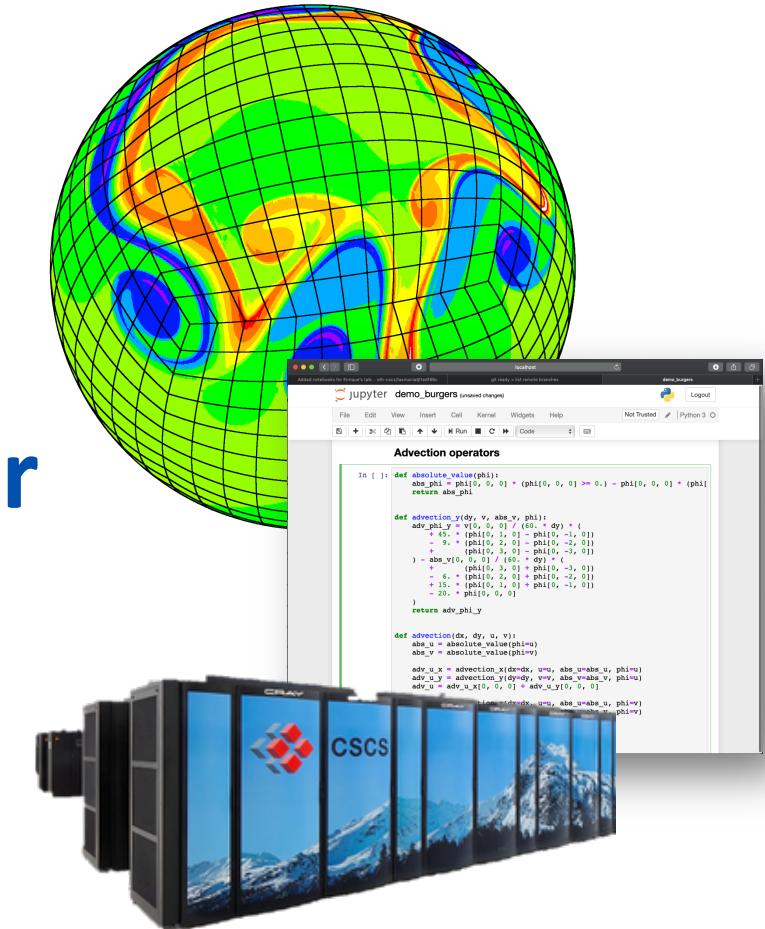
High Performance Computing for Weather and Climate (HPC4WC)

Content: Administrative

Lecturers: Oliver Fuhrer, Tobias Wicky, Stefano Ubbiali

Block course 701-1270-00L

Summer 2020



Nice to meet you!



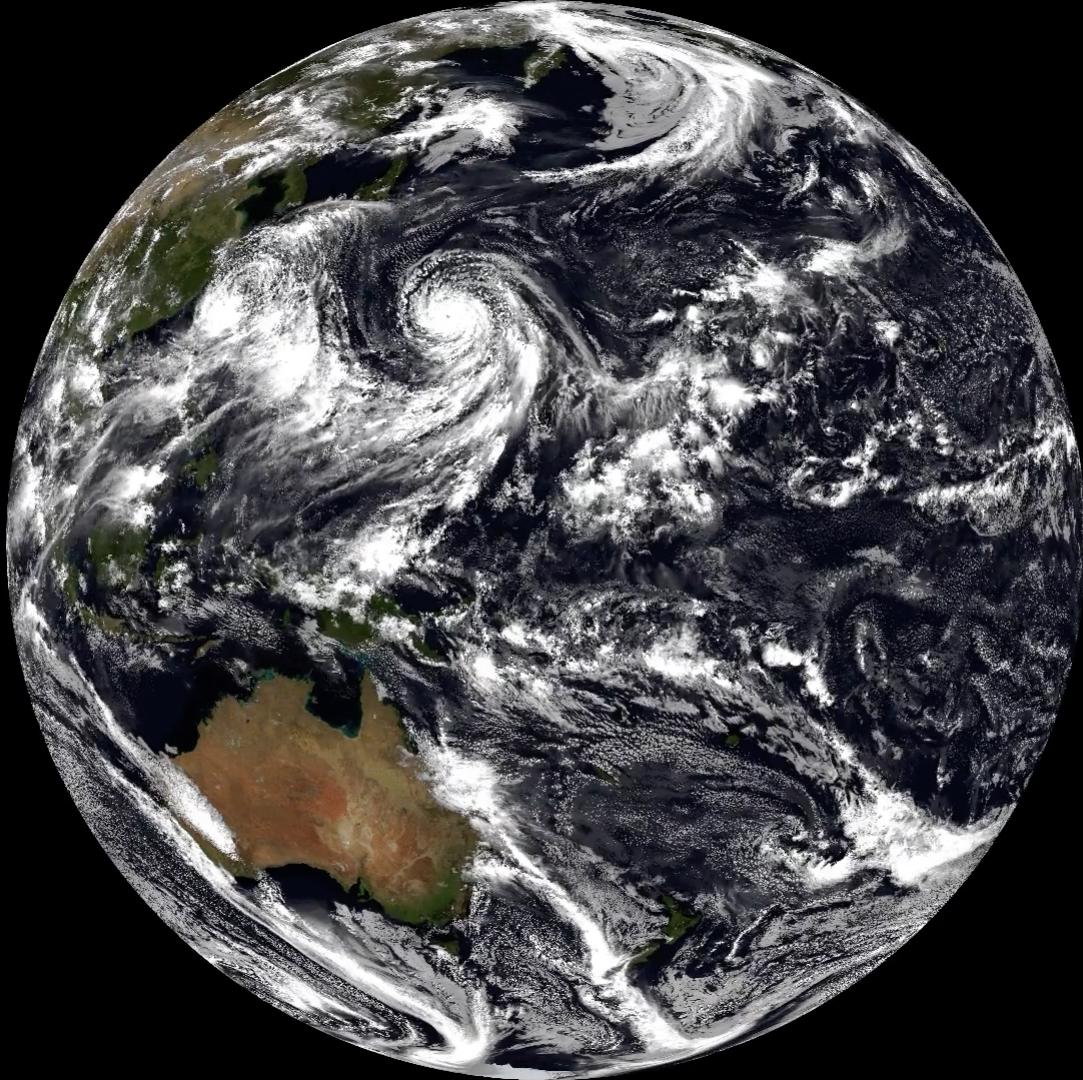
Oliver Fuhrer
oliver.fuhrer@ethz.ch



Tobias Wicky
twicki12@gmail.com



Stefano Ubbiali
subbiali@phys.ethz.ch



2016-08-11 18:00Z
258 Forecast Hours
FV3 3km

Visualization
Xi Chen@FV3 team
[Introduction](#) 3

Goals of course

- Understand high performance computing concepts relevant for weather and climate simulations
- Able to work with weather and climate simulation codes that run on large supercomputers

Approach

**“ I Hear and I Forget,
I See and I Remember,
I Do and I Understand**

(chinese proverb)

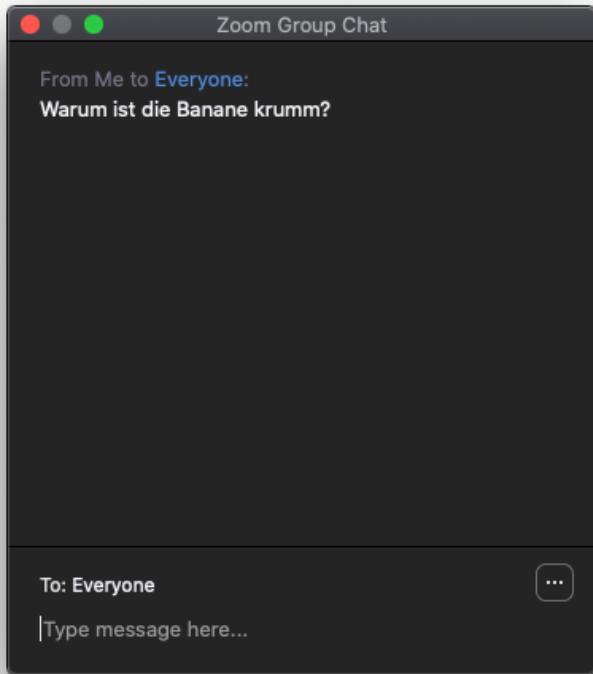
- Lectures that explain concepts and give context (hear).
- Demonstrations of the concepts being applied (see).
- Practical exercises and a work project (do).

Questions, please!

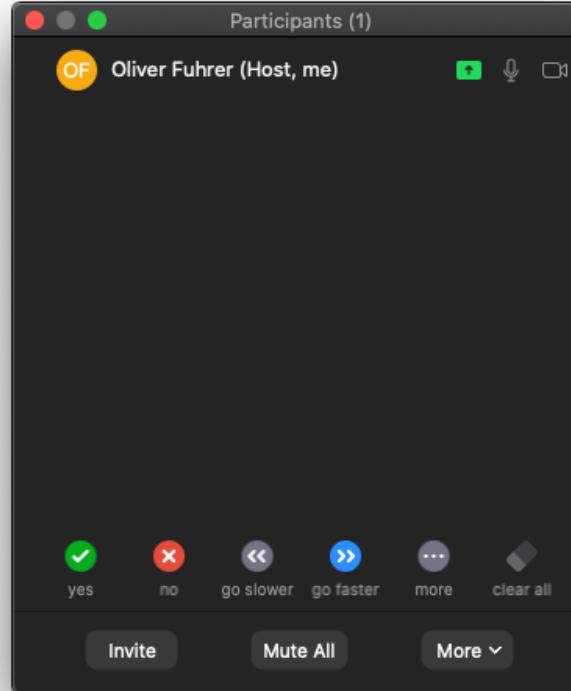


Zoom Interactivity

Chat



Answer



Schedule

Monday	Motivation, stencil computations, memory hierarchy, lab environment	08:15 – 12:00	Morning session
Tuesday	Shared memory parallelism, OpenMP, performance metrics	08:15	Check-in (Zoom)
		12:00 – 13:30	Lunch break
Wednesday	Distributed memory parallelism, domain-decomposition and halo-updates	13:30 – 17:30	Afternoon session
Thursday	Hardware trends in supercomputing, GPU computing	13:30	Check-in (Zoom)
Friday	High-level programming, domain-specific languages, wrapup	17:00	Check-in (Zoom)

Zoom lectures will be recorded and made available!

Currently registered students

MSc/BSc Environment Science (USYS)	11
MSc/BSc Computational Sciences (MATH)	11
MSc Atmosphere and Climate (ERDW)	8
PhD students (various departments)	6
MSc/BSc Environmental Engineering (BAUG)	3
MSc/BSc Physics (PHYS)	2
MSc/BSc Computer Sciences (INFK)	1
MSc Math / Applied Math / Statistics (MATH)	1
MSc Env Engineering (MAVT)	1
Auditor	1
Total	45

Prerequisites

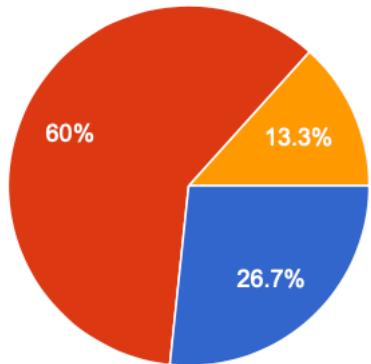
- **Fundamentals of numerical analysis and atmospheric modeling**
 - Basic partial differential calculus and finite difference methods.
 - e.g. ETH course “[Numerical methods in environmental physics](#)” or “[Numerical modeling of weather and climate](#)”
- **Experience in a programming language (C/C++, Fortran, Python, ...)**
 - We will read and write [Fortran](#) and [Python](#) in this course.
- **Experience using command line interfaces in *nix environments (e.g., Unix, Linux)**
 - Familiar with work in the [command line shell](#) and the most commonly used shell commands.
 - Can logon to linux system via ssh and can work remotely on that system.
 - We will work on the [Piz Daint supercomputer](#) at the [Swiss National Supercomputing Center \(CSCS\)](#) in Lugano in this course.

If you think this course might not be suitable for you, contact us!

Questionnaire

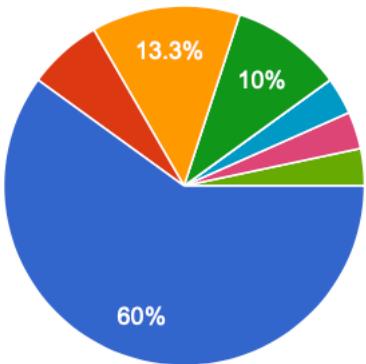
(based on 30 replies received by 6/6/20)

PDEs on computer



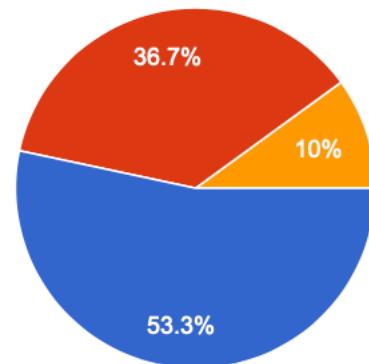
- Yes, regularly.
- Yes, a few times.
- Never.

Programming language



- Python
- Fortran
- C / C++
- Modern C++
- Java
- None
- R
- MATLAB

Linux / unix



- It's my preferred work environment.
- I have basic experience.
- It's new to me.

Practicalities

- All course material on GitHub repository (slides, notebooks, codes, ...) <https://github.com/oführer/HPC4WC/>
- Questions related to course in dedicated Slack workspace https://join.slack.com/t/hpc4wc/shared_invite/zt-dqj7kgqt-2Rkw4Y0Khg64XUDJQgPaew
 - Possible to use video and screen sharing.
 - Generally, try to use public channels for questions since others probably have the same questions.
- Lectures and check-ins are via Zoom (Link shared before lecture via Email or Slack)

How to earn credits (3 ECTS)

- **Attend the block course** (and participate actively!)
- **Work project**
 - Choose group and topic
 - Hand in working source code and report (max. 10 pages)
 - Projects will be graded
 - **Deadline: 7. August 2020**
- Credits are awarded if course attended and grade of work project ≥ 4.0
- Same rules apply for BSc, MSc and PhD students

Work project

- **Work in groups of 2-3** (individual projects will not be accepted)
 - Programming is not a solitary art!
- **Topics will be presented on Friday**
 - If you prefer to choose your own, you are required to discuss with us beforehand
 - Each project must have a software development and performance evaluation part.
- **Grading**
 - 25% correctness (compiles & runs, results correct, no bugs)
 - 25% quality (structure, clean code, comments, naming, tests, error handling)
 - 25% performance (depending on work project)
 - 25% report (maximum 10 pages)
- **Use Slack workspace for questions**

Lab exercises

- Swiss National Supercomputing Centre <https://www.cscs.ch/>
- Piz Daint supercomputer (Europe's largest supercomputer)



CSCS Accounts

- Send us a direct message on Slack to get your user name and password.
- **Do not share you login / pwd with anybody else.** Accounts with suspicious activities will be close down by CSCS immediately.
- **Change your password** immediately upon your first login to CSCS using the kpasswd command in a Terminal.
- We have a shared quota of 3000 node hours for using the CSCS supercomputers for this block course.
 - Do not launch jobs with more than 1 node without checking with us first.
 - Do not leave your JupyterHub Server running if you don't need it.
- **Do not contact CSCS** if you have trouble. Use the Slack workspace to get your issues resolved.
- Take a look at the [**CSCS Code of Conduct**](#)

JupyterHub

- Lab exercises will all be conducted on <https://jupyter.csccs.ch/>, the JupyterHub portal of CSCS.
- Interactive development and computing environment.
- If things get stuck or go wrong, it's always possible to "Stop Server" and "Launch Server" again.
- Jupyter notebooks auto-save and almost certainly no work will be lost.

1. Increase to 10 hours

2. Click to expand

3. Enter "course_ethz" here

4. Delete the "1" here

5. Click to launch

Node Type: GPU

Nodes: 1

Duration (hr): 10

Queue: Dedicated Queue (Max. 4 Nodes)

Project Id (leave empty for default):

Advanced Reservation: course_ethz

JupyterLab Version: 1.1.1

Start IPyParallel Cluster with MPI Support? No

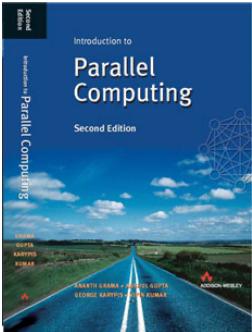
Start Distributed Dask Cluster? No

MPI Processes Per Node (default: one process per virtual core): 1

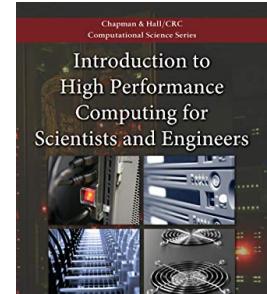
Dask Tasks Per Node (default: one task per node): 1

Launch JupyterLab

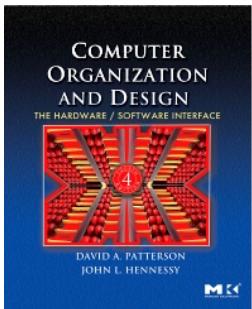
Literature & Links



Introduction to High Performance Computing for Scientists and Engineers, G. Hager and G. Wellein, CRC Press, 2011
[\(available online at ETH\)](#)



Parallel Computing, A. Grama, A. Gupta, G. Karypis, V. Kumar
[\(available free online\)](#)



Parallel Programming in MPI and OpenMP, V. Eijkhout
[\(Link to course\)](#)



Computer Organization and Design, D.H. Patterson and J.L. Hennessy [\(available online at ETH\)](#)