703308/703309 VO+PS High-Performance Computing
Where are the slides?

https://github.com/philippgs/uibk_hpc_23

or

https://tinyurl.com/UIBKHPC23

703308/703309 VO+PS High-Performance Computing
Introduction & Administrative Stuff

Philipp Gschwandtner

Organizational stuff

- lecturer information
  - Philipp (Gschwandtner (PhD))
  - philipp.gschwandtner@uibk.ac.at
  - philgs
  - room 2W05, ICT building
  - no fixed office hours
    (send an e-mail, I'm quite responsive…most of the time)
- dates and location
  - see lfu:online for exact dates
  - generally:
    - lecture every Wednesday 11:15-12:45 in SR 12
    - proseminar every Tuesday 08:15-09:45 in RR 14

More organizational stuff

- prerequisites
  - interest in parallel hardware, parallel programing and high performance computing
  - lecture: very little beyond that
  - proseminar: + programming in C/C++
- language
  - English
- content
  - general concepts of parallel programming and its intricacies
    - concepts apply to almost all parallel programming models
    - as an example, we will mainly discuss MPI
    - there are countless others (OpenMP, OpenCL, CUDA, TBB, Cilk, Pthreads, C++ STL, Charm++, X10, PGAS, …)

Grading: Lecture

- no mandatory attendance
  - Note: not everything I say will be on the slides…
- single, written exam on January 31st 2024
  - multiple exercises with multiple points
  - standard grading scheme, ≥ 50 % for positive grade
  - Don‘t memorize the slides, understand the content!

Grading: Proseminar

- weekly assignments, published on GitHub
  - https://github.com/philippgs/uibk_hpc_23
- teamwork is permitted and encouraged
  - 3 people max. per team
  - **every** team member must be able to present and discuss solution
- solutions must be handed in via OLAT until Monday 17:00!
  - solutions **must work** on the LCC3 cluster
  - copying solutions (e.g. off the Internet) is acceptable **if properly cited and understood**
  - grade is 50 % solutions, 50 % presentations/discussion – both must be ≥ 50 %!

Interaction & Feedback

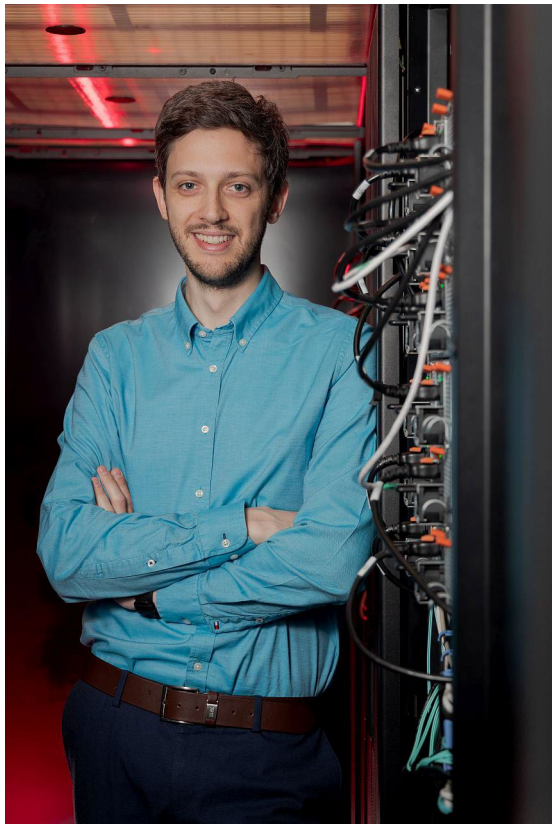
- Platform for interaction outside of the PS/VO
  - https://discord.gg/f2QhzDPAH
- Anonymous Feedback possible via Google Form linked in OLAT

Literature

- www.internet.com
  - MPI: A Message-Passing Interface Standard 4.0
    (PDF available via https://www.mpi-forum.org/, hardcover of v3.1 available)
  - Stackoverflow
  - Google
  - …
- old school: Printed books
  - Let me know and I will look up some references…

Who is this speaker, what does he do besides teaching?

- Deputy Head of Research Center HPC (Forschungszentrum Hochleistungsrechnen), University of Innsbruck (https://dps.uibk.ac.at/~philipp, https://uibk.ac.at/fz-hpc)
  - Studied computer science @ UIBK, focus on parallel programming, benchmarking and tuning
- Research interests in and around HPC
  - Measurement/optimization/modeling of performance, energy, efficiency, …
  - APIs, programming models, runtime systems, compilers, ...
- Aid researchers at UIBK in developing and optimizing parallel applications
  - (Co-)Writing HPC-focused project proposals
  - Making stuff go faster
  - Giving training course on how to make stuff go faster (OpenMP, MPI, etc.)
- **Supervising exciting master theses!**



Extremely professional-looking photo of Philipp Gschwandtner (© Andreas Friedle)

What are we all doing here?

- discuss key concepts of parallel computing
  - hardware **and** software aspects
  - multiple non-functional aspects – there's more than just speed
  - portability, usability, maintainability, sustainability
- we still need to actually do some concrete work
  - (mostly) MPI for implementing and evaluating distributed-memory parallelism concepts
  - we'll use ~~LCC2~~LCC3 for running experiments

What are we going to discuss?

- crash course on hardware and programming models
- introduction to MPI (and a bit of other APIs/models such as OpenMP?)
- tons of generic concepts at the example of MPI (and others) programs
  - metrics: performance, efficiency, scalability
  - problem partitioning, scheduling and load balancing
  - parallel program classification and characteristics
  - programmer productivity, debugging, profiling
  - …

Hints (not only) for this course

- choose a suitable source code editor / IDE and choose it wisely!
- get acquainted with your toolchain
  - debuggers, version control (git), etc.
- use common sense and sanity checks!

Questions?

Image sources

- LCC2: https://www.uibk.ac.at/zid/systeme/hpc-systeme/leo3/
- Sandbox: http://www.googblogs.com/open-sourcing-sandboxed-api/