703308 VO High-Performance Computing WS2022/2023 Introduction & Administrative Stuff

Philipp Gschwandtner

Where are the slides?

https://github.com/philippgs/uibk_hpc_22

or

https://tinyurl.com/UIBKHPC22

Organizational stuff

- **lecturer information**
  - Philipp (Gschwandtner (PhD))
  - philipp.gschwandtner@uibk.ac.at
  - philgs#1930
  - 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 Thursday
      08:15-10:00 in HSB 8
    - proseminar every Tuesday
      08:15-10:00 in RR 15

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-ish?

- **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 February 2nd 2023**

  ▸ multiple exercises with multiple points

  ▸ standard grading scheme, ≥ 50 % for positive grade

  ▸ Covid-19-specific: could be in-person, could be online ¯_(ツ)_/¯

  ▸ Don't memorize the slides, understand the content!

Grading: Proseminar

- weekly assignments, published on GitHub
  - https://github.com/philippgs/uibk_hpc_22

- 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 LCC2 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

▸ **Recommend some platform for interaction outside of the PS/VO**

　　▸ Discord server, anyone?

　　▸ OLAT boards are also possible

　　▸ Alternatives?

▸ **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…

What do I do when I am not teaching?

- Senior Scientist and Deputy Head at Research Center HPC (Forschungszentrum Hochleistungsrechnen)
  - www.uibk.ac.at/fz-hpc
  - aid researchers at UIBK in developing and optimizing parallel applications
  - formerly Distributed and Parallel Systems Group (DPS), https://dps.uibk.ac.at

- research interests in and around HPC
  - measurement/optimization/modeling of performance, energy, efficiency, …
  - APIs, programming models, runtime systems, compilers, …
  - interested in master thesis topics?

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 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/lcc/hardware/
- Sandbox: http://www.googblogs.com/open-sourcing-sandboxed-api/