Parallel Computing
„Einführung in paralleles Rechnen“

The Formalities, the General Plan

Q&A

General plan

- Wednesdays 13:00 – 15:00 (Informatik Hörsaal, HERE): Lectures (Träff).

Mandatory

2 out of 3 for final grade

- Three exams (2+1), Fridays 21.4, 26.5, 16.6 (Hörsäle INF & EI 9 & HS 18), 9:00 – 11:00
- (Optional) Exercises&Programming projects

Mandatory

Mandatory means:
We think it is a good idea to visit the lectures regularly, collect the information, talk to your fellow students, … (your responsibility: if other things are more important, consider whether to attend…)

Attendance is not policed, not enforced by any means…

But: The material for the course is that which is covered in the lectures. If you cannot come to some particular lecture, it's your own responsibility to check up on the material (read the slides, talk to your friends…). It is no excuse that "I did not attend the lecture"

Same rules for everybody, no exceptions, no special cases, …

Administration

Sign up via TISS (until 13.3.2023)

Sign off via TISS if you do not want to complete the VU (until 24.4.2023)

All information via TISS and TUWEL (!)

Course material (slides) and exercise/project hand-in via TUWEL

Check regularly!

This VU consists of

- Lectures
- “Getting account” exercise
- Exams, in person (2 out of 3 counts towards final grade)
- Home exercises and programming project: Theoretical stuff from lecture, programming project in C with OpenMP&MPI to get working experience with a real, modern (small) high-performance cluster-system
- Self-study

auf Deutsch

- Read, think, solve, program, experiment, learn…

Exercises/projects are OPTIONAL.
Solutions handed in via TUWEL (groups of 2-3 allowed), commented/graded offline, plenary feedback

ECTS Breakdown

- Lectures: 1.5 ECTS
- Study: 1.5 ECTS
- Exercises (theory, and implementation, test, benchmarking): 3 ECTS

In hours to invest…

- Lectures: 12x2h = 24h
- Other online-stuff: 2x2h = 4h
- Exams: 3x3h = 9h
- Self-study: 41h
- Home exercises: 24h
- Programming parts: 2x24h = 48h

Total: 150h = 6 ECTS

Give feedback by the end of the course

Detailed plan: Lectures (Wednesdays)

1.3: Intro, motivation. Basics (I)
8.3: Principles (II): time, work, cost, speed-up, Amdahl
15.3: Principles (III): Models, PRAM
22.3: Examples & Algorithms
29.3: Shared-memory systems, (p)threads (1st exam preview)

Easter, 3.4 - 14.4

19.4: OpenMP
26.4: OpenMP
3.5: OpenMP
10.5: Distributed memory systems
17.5: MPI
24.5: MPI
31.5: MPI
7.6: Slack
15.6: Emergency slack

Attendance is mand

PC will be over by the end of the semester, no hangover

Detailed plan: Exams (three Fridays) 9:00 – 11:00

Mandatory (2 best out of 3 count)

21.4: Basics (Work&Time, PRAM, Problems&Algorithms)

26.5: Shared-memory, OpenMP (plus questions related to first exam)

16.6: Distributed memory, MPI (plus questions related to first two exams)

Mandatory to sign up in TISS in advance

Exams: Mode and procedure

Exams are individual, in lecture hall, pen-and-pencil, no aids (no books, no script, no notes, no mobile phone, no tablet)

Sign-up in TISS mandatory: Careful with choice of lecture hall

X independent, multiple-choice problems (in each exam), Yi questions (for each problem), for a total maximum of
$\sum_{X \text{ problems}} \sum_{Yi \text{ questions}} 1$ points

Points:
$\sum_{X \text{ problems}} \sum_{Yi \text{ questions}}$(correct +1, no answer/blank 0, wrong -1)

The exams cover material directly from the lecture.

- Exam 1: PRAM model and algorithms, work, time, parallel speed-up, efficiency, Amdahl's law
- Exam 2: Shared-memory basics, OpenMP, plus questions on earlier material
- Exam 3: Distributed memory basics, MPI, plus questions on earlier material

Exams: Grading

Of the three exams, **the best two (2) out of three (3) will count**. The total grade will follow from the average percentage points of the two best exams.

Adjustment/scaling of points may be done after each exam (depending on overall performance)

Individual feedback (ca. 5 minutes) will be offered ("Einsichtnahme"). Dates TBA, sign-up for slot in TISS

Detailed plan: Exercises and programming projects

Optional. Can be done in groups of 1, 2, 3

22.3 – 29.3: Basics, PRAM algorithms

26.4 – 10.5: Shared-memory, OpenMP

24.5 – 7.6: Distributed memory, MPI

But: Group sign-up in TUWEL required

Optional

Exercises&programming projects: Content and grading

More detailed exercises on the material from the lectures (“theory”, on paper) as preparation for the exams.
Concrete programming projects in C with OpenMP and MPI.
Roughly 1/3 dry exercises, 1/3 OpenMP, 1/3 MPI

Programming projects will be checked semi-automatically, and correctness and efficiency matter! Follow the specifications!

We give feedback and points for the exercises/programming projects

Exercies and project will not count towards final grade!

System access for programming projects

The programming projects will be done on a real (small) high-performance cluster which is operated like a compute center: batch mode via scheduler (slurm)

Login to the system via ssh, key needed, separate exercise sheet

How-to-do will be described in the exercise sheet, read carefully!

System access needed, by all group members, for completing the programming projects successfully

Exercises&programming projects: Hand-in and groups

All hand-ins via TUWEL before deadline, on-time

Deadlines are fixed, and will not be changed at discretion. Same deadline for everyone, **no exceptions**, no late or special hand-ins.

Watch out for the deadlines, be aware!

Exercises&programming projects: Groups

Exercises and programming projects can be done in groups of up to three (3).

Group registration necessary via TUWEL

For group hand-in's: One upload suffices.

If more than one solution is uploaded, they must be identical. If not, weakest one counts. Group's responsibility to make sure at least one solution is handed-in!

Disclaimer:

Do not cheat, do not plagiarize

Exercises and projects are optional. There is simply no point in handing in something generated or copied from elsewhere…

We offer feedback (as good as we can)

Overall grading

Two best exams, percentage of exam points for each averaged (may be adjusted)

Grading scheme:

1 (“sehr gut”):	90-100%
2 (“gut”):	75-90%
3 (“befriedigend”):	60-75%
4 (“genügend”):	50-60%

TUWEL Fora, feedback

- TUWEL Forum for finding group partners
- TUWEL Forum for discussion

The Parallel Computing team (professors, tutors) will answer questions regarding the content of the lecture once a week (Thursday or Friday)

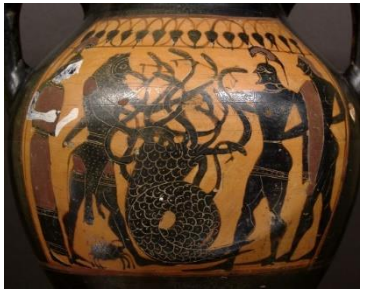
No guarantee that a particular question will be answered!

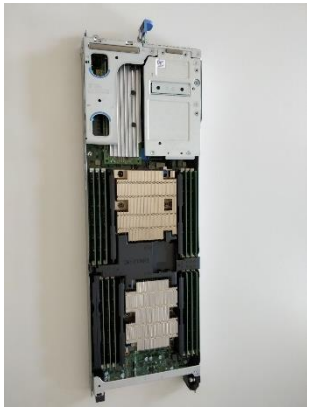
Talk to your friends and colleagues first, before posting questions to Forum

System

Programs can be developed at home on own computer (prerequisites: C compiler, OpenMP, MPI), but final test and benchmarking must be done on our (TU Wien) systems

- “Hydra”: 36-node x 32-core Intel Skylake-OmniPath cluster





Serverroom, Favoritenstr. 9-11

Programming exercises (projects), system

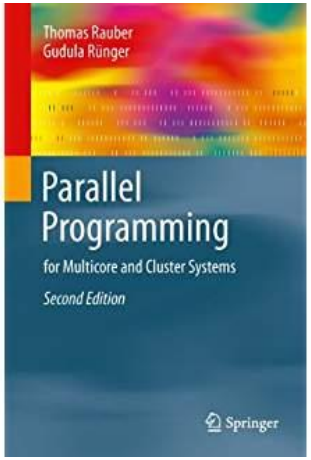
The compute cluster is a rare, limited resource (1 cluster, 300+ users). We run it like a compute center…

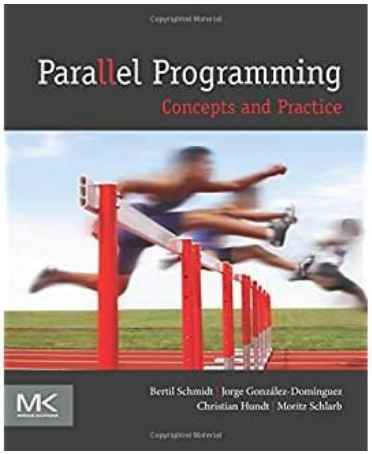
Use your resources with care, start early with the programming parts

Literature

- Slides, script, course material

- T. Rauber, G. Rünger: Parallel Programming for Multicore and Cluster Systems. 2nd Ed., Springer, 2013 (auch auf Deutsch)
- B. Schmidt et al.: Parallel Programming. Concepts and Practice. Morgan-Kaufmann, 2018.

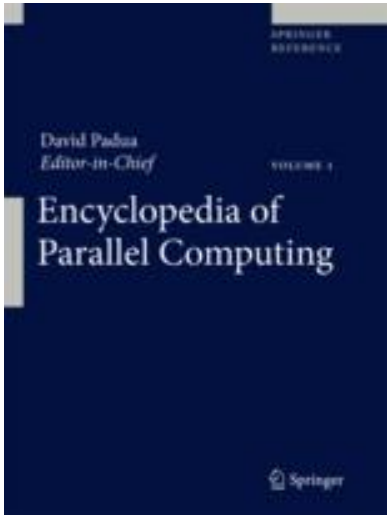


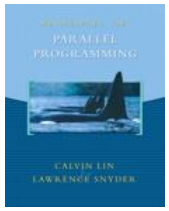


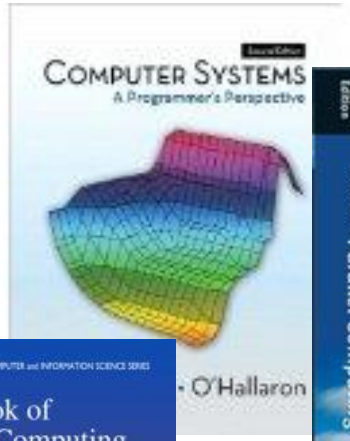
Additional Literature

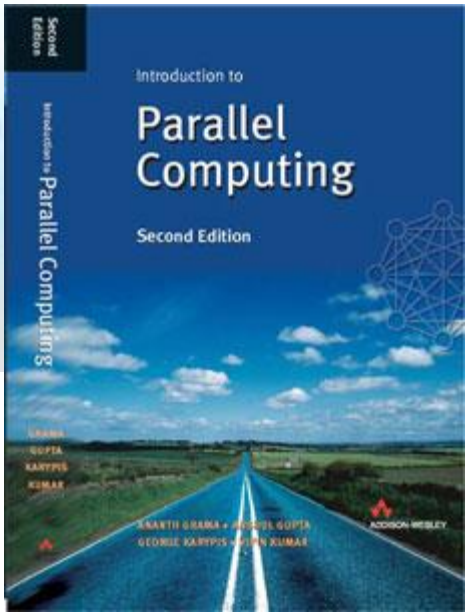
- Grama, Gupta, Karypis, Kumar: Introduction to Parallel Computing. Second edition. Pearson 2003
- Michael J. Quinn: Parallel Programming in C with MPI and OpenMP. McGraw-Hill, 2004
- Calvin Lin, Lawrence Snyder: Principles of parallel programming. Addison-Wesley, 2008
- Peter Pacheco: An introduction to parallel programming. Morgan Kaufmann, 2011

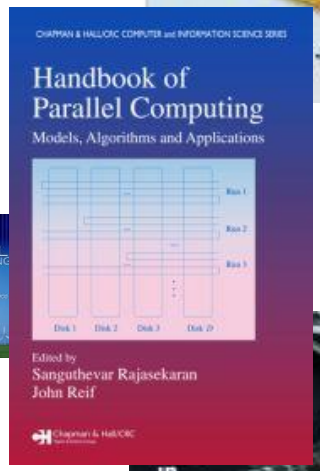
- Randal E. Bryant, David R. O'Hallaron: Computer Systems. Prentice-Hall, 2011


David Padua
Editor-in-Chief
Encyclopedia of
Parallel Computing
Springer


PARALLEL
PROGRAMMING
CALVIN LIN
LAWRENCE SNYDER

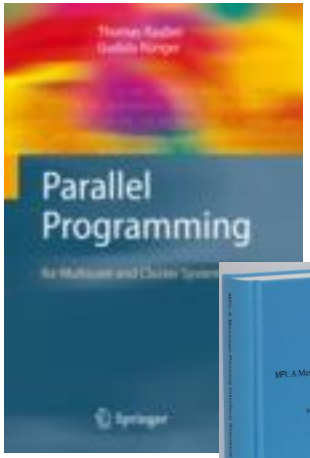

COMPUTER SYSTEMS
A Programmer's Perspective
O'Hallaron

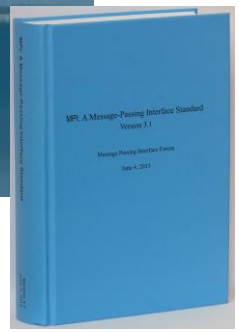

Second Edition
Introduction to
Parallel
Computing
Second Edition
Introduction to Parallel Computing
ADDISON-WESLEY


CHAPMAN & HALL/CRC COMPUTER and INFORMATION SCIENCE SERIES
Handbook of
Parallel Computing
Models, Algorithms and Applications
Edited by
Sanguthevar Rajasekaran
John Reif
Chapman & Hall/CRC


PARALLEL PROGRAMMING
in C with MPI and OpenMP
Michael J. Quinn


PARALLEL
PROGRAMMING


Parallel
Programming
Springer

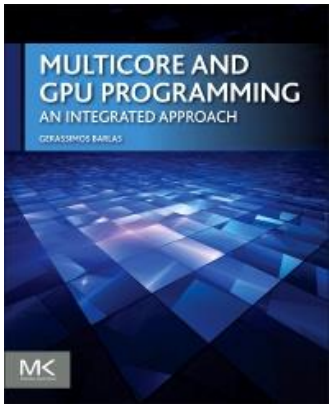

MPI: A Message-Passing Interface Standard
Version 3.1

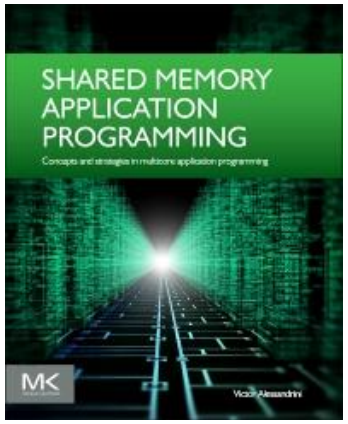

Using MPI
Portable Parallel Programming
with the Message-Passing Interface
second edition

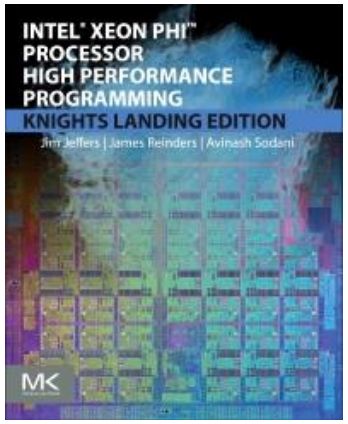

in
OpenMP

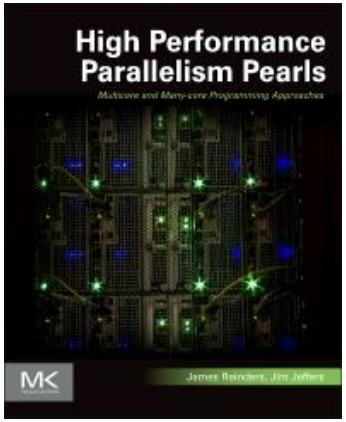

OpenMP

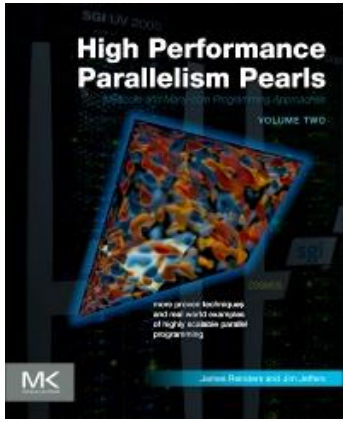

Programming Massively
Parallel Processors
A Hands-on Approach

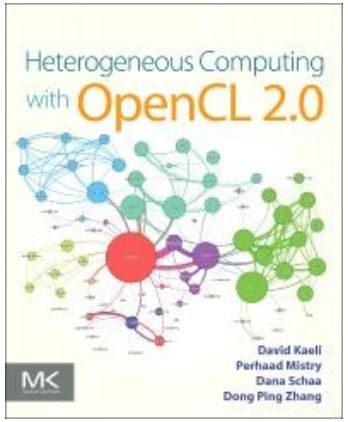

MULTICORE AND
GPU PROGRAMMING
AN INTEGRATED APPROACH
GERASSIMOS BARLAS


SHARED MEMORY
APPLICATION
PROGRAMMING


INTEL® XEON PHI™
PROCESSOR
HIGH PERFORMANCE
PROGRAMMING
KNIGHTS LANDING EDITION
Jim Jeffers | James Reinders | Avinash Sodani


High Performance
Parallelism Pearls
James Reinders, Jim Jeffers


High Performance
Parallelism Pearls
VOLUME TWO


Heterogeneous Computing
with OpenCL 2.0
David Kaeli
Perhaad Mistry
Dana Schaa
Dong Ping Zhang

Follow-up

Bachelor thesis (Träff, Hunold):

- All aspects of parallel computing (implementations, benchmarking, applications, algorithms, models, …)

Master:

- High Performance Computing, VU, 4.5ECTS
- Advanced Multiprocessor Programming, VU, 4.5 ECTS
- Parallel Algorithms, VU, 3 ECTS
- Projects, 6+6, ECTS
- Seminars in Software Engineering, Algorithms, Theoretical computer science, computer engineering, 3 ECTS
- Master Thesis, 30 ECTS