Introduction to Parallel Computing

Jesper Larsson Träff, Angelos Papatriantafyllou
{traff, papatriantafyllou}@par.tuwien.ac.at

Parallel Computing, 184-5
 Favoritenstrasse 16, 3. Stock

Sprechstunde: Per email-appointment
Parallel Computing

Parallel computers are around and everywhere
(it was not always like that)

How to use them?
• Efficiently?
• In practice?

How do they look?

Algorithms, Languages, Interfaces,
(Applications)

Architecture, Models
This VU: Introduction to parallel computing

• Introduction: aims, motivation, basics, history (Amdahl's Law, Moore's „Law“, ...)

• Shared memory parallel computing
  • Concrete language: OpenMP, pthreads, Cilk

• Distributed memory parallel computing
  • Concrete interface: MPI (Message-Passing Interface)

• New architectures, new languages (GPU, CUDA, OpenCL)
  • Other languages, paradigms

Theory and PRACTICE
Introduction to parallel computing

Focus on Principles: parallel algorithms, (architectures), languages and interfaces

Standard, paradigmatic, actual, much-used languages and interfaces (MPI, OpenMP, pthreads/C threads, Cilk)

Lot's of approches, languages, interfaces that will not be treated - but possible to follow up later: bachelor-thesis, project, master-thesis, master lectures, seminars. See us!
Prerequisites (≥3rd Semester, STEOP)

- C/C++, Fortran (Java) programming skills
- Operating systems
- Algorithms & Data structures
- Computer architecture
- ...

Interest in solving problems faster on these machines...
Lectures, exercises, projects

- Monday, **10:00-12:00** MANDATORY
- Occasionally: Tuesday, **10:00-12:00** (also MANDATORY)

FreiHaus Hörsaal 7
(except 13.10: Hauptgebäude HS7)
Tuesday: Argentinierstrasse

Project work: **ON YOUR OWN** - there will be **Q&A sessions** (Tuesday slots)

*Can start early, complete before end of lecture, discussion/examination at end of semester (late January, early February)*
Lectures, exercises, projects

Capacity?

Lecture was originally planned for 40+ students...

Lecture halls??? Overlap with “Gesellschaftswiss. Grundlagen der Informatik”... (sorry)
Requirements, credit (4 hours/week, 6ECTS)

• Lecture attendance **MANDATORY**
• Active participation during lectures
• Presentation of project work (exam) **MANDATORY**
• Hand-in of project work **MANDATORY**:
  1. Short write-up
  2. Program code
  3. Results

Practical project work: *should* be done in groups of 2

**NOTE:**
See us ("Sprechstunde") in case of problems with schedule
(unable to finish project in time)
Requirements, credit (4 hours/week, 6ECTS)

- Lecture attendance **MANDATORY**
- Active participation during lectures
- Presentation of project work (exam) **MANDATORY**
- Hand-in of project work **MANDATORY**:
  1. Short write-up
  2. Program code
  3. Results

Practical project work: should be done in groups of 2

**GRADE:**
Based on project presentation and hand-in
Requirements, credit (4 hours/week, 6ECTS)

- Lecture attendance **MANDATORY**
- Active participation during lectures
- Presentation of project work (exam) **MANDATORY**
- Hand-in of project work **MANDATORY**:
  1. Short write-up
  2. Program code
  3. Results

Practical project work: should be done in groups of 2

**NOTE:**
Solutions to project exercises can possibly be found somewhere.
Don’t cheat yourself!!
Requirements, credit (4 hours/week, 6ECTS)

• Lecture attendance **MANDATORY**
• Active participation during lectures
• Presentation of project work (exam) **MANDATORY**
• Hand-in of project work **MANDATORY**:
  1. Short write-up
  2. Program code
  3. Results

Practical project work: should be done in groups of 2

**NOTE:**
Solutions to project exercises can possibly be found somewhere. **Don’t cheat us:** be open about what you took from others, plagiarism will automatically result in grade 6 (fail)!
Grading: **DON'T OPTIMIZE**

Active participation in lectures...

Written project solution, quality of programs (correctness, performance, readability), oral explanation, knowledge of course material

Each project will consist of 3 parts; deliberately, not everything is said explicitly (but enough should be said)

Very roughly:
• **1-2**: All parts solved, performance/speed-up achieved, everything can be explained
• **2-3**: 2 out of 3...
• **Fail**: Less than 1 out of tree
Grading: **DON'T OPTIMIZE**

Groups of two:
Stand or fall as a group, ideally both get same grade

**Means:**
Both group members should have contributed and feel responsible for all parts of the solutions
ECTS breakdown

- Planning, intro („Vorbesprechung“): 1h
- Lectures: 15 x 2h = 30h
- Preparation: 45h
- OpenMP: 20h
- Cilk: 20h
- MPI: 20h
- Write-up: 10h
- Presentation, including preparation: 4h

Total: 150h = 6ECTS
Project exercises

- Programming exercises using the main three languages/interfaces covered in the lecture (OpenMP/pthreads, Cilk, MPI). Each exercise will explore the same problem in all three paradigms

- Tentatively: Select 1 (or 2) out of 4

- Focus on achieving and documenting improved performance (good benchmarking)
- Correctness first!

- (Some) room for creativity
Project exercises, credit

Document solution with code and short report

**Code:** readable, compilable, correct

**Report:**
- **IN ENGLISH** (as far as possible)
- State problem, hypothesis, explain (briefly) solution, implementation details and issues, state of solution (correct, what works, what not), testing and benchmarking approach, document performance (plots or tables)
- Compare/comment on paradigms
- 8-15 pages per exercise, including performance plots

**Project exercises in groups of two**
Schedule - TENTATIVE

6.10: Planning, overview („Vorbesprechung“)
13.10: Motivation, concepts
20.10: Example problems: merging, prefix-sums
21.10: Projects presentation (IMPORTANT!)
27.10: Shared memory architectures & programming
3.11: pthreads, C11
10.11: OpenMP
17.11: NO LECTURE (do project work)
24.11: OpenMP, Cilk
1.12: Cilk
8.12:
15.12: Distributed memory architectures & programming, MPI
22.12
12.1: MPI
19.1: Other architectures and interfaces
26.1:

2.2: Project hand-in

WS14/15 © Jesper Larsson Träff
Schedule - TENTATIVE

Idea:
All basics, and all 3 interfaces (OpenMP, Cilk, MPI) covered before Christmas (so: some Tuesdays may be necessary)
January: Other architectures and interfaces; project work

Project hand-in: 2.2.2015
Exams: 9-13.2.2015
Literature, course material

Slides in English - will be made available at

www.par.tuwien.ac.at/teach/WS13/ParComp.html

Look here and TISS for information (cancelled lectures, change of plans, ...). Will try to keep up to date, timely (but lectures will not be ready much in advance...)

No script; slides should be enough for doing the project work, additional material can be found easily
Organizational

This year: will (try to) use TUWEL
• Forming the groups
• Getting accounts
• Your discussions?
• Uploading code/reports

Register in groups of 2 now (until 31.10.14)!
Literature: general


• 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

Literature: general


Literature: OpenMP, MPI, CUDA

Chandra, Dagum et al.: Parallel Programming in OpenMP. Morgan Kaufmann, 2001

Barbara Chapman, Gabriele Jost, Ruud van der Pas: Using OpenMP. MIT, 2008


William Gropp, Ewing Lusk, Anthony Skjellum: Using MPI. MIT, 1999

Parallel Programming

Computer Systems
A Programmer's Perspective

Introduction to Parallel Computing
Second Edition

Parallel Programming in OpenMP

Using MPI: Portable Parallel Programming with the Message-Passing Interface
Second Edition

Programming Massively Parallel Processors
A Hands-on Approach
Systems, hardware

OpenMP, Cilk

48-core AMD-based shared-memory cluster, „Saturn“

MPI

36-node InfiniBand AMD-based 2x8 core cluster = 576 processor cores, „Jupiter“

Access via ssh (instructions to follow), program at home/TU. No actual lab
Saturn: AMD-based, shared-memory system

Jupiter: small InfiniBand cluster, AMD processors
Other systems at TU Wien parallel computing

• Pluto: 16-core Ivy-bridge system + 2xNVidia K20x GPU + 2xIntel Xeon-Phi 60-core accelerator
• Mars: 80-core Intel Westmere system, 1TB shared memory
• Ceres: 64-core Oracle/Fujitsu shared memory system, Sparc-based with HW-support for 512 threads, 1TB shared memory

Research systems for bachelor, master and PhD work...
Research Group Parallel computing

Some information at www.par.tuwien.ac.at

Favoritenstrasse 16, 3rd floor
Next to U1, Taubstummengasse, exit Floragasse
Research Group Parallel computing

Exam (Early February) in Favoritenstrasse 16, 3rd floor, HK 03 20

Contact:
Use lecture first, TUWEL second, contact us per email (for questions, appointment):

traff@par.tuwien.ac.at
papatriantafyllou@par.tuwien.ac.at
Our themes

1. HPC languages, interfaces – design, algorithmic support and implementation (MPI, PGAS)
2. Interfaces for multi-core parallel computing – algorithmic support and implementation: task-parallel models, lock- and wait-free data structures
3. Parallel algorithms
4. Scheduling in theory and practice
5. Communication networks (routing), memory-hierarchy
6. Experimental parallel computing – benchmarking, validation, reproducibility
7. (Heterogeneous parallel computing: interfaces, autotuning, scheduling)
Parallel computing
TU Wien parallel computing

Algorithms

Programming interfaces

Architectures

Applications
Bachelor:

VU Parallel Computing

Master:

VU Parallel Algorithms
• PRAM
• Network algorithms

VU Advanced Multiprocessor Programming
• Programming models, lock-free algorithms and data structures

VU High Performance Computing

SE Topics in Parallel Programming Models, Algorithms, Architectures