
Parallel Processing

Winter Term 2025/26

Roland Wismüller
Universität Siegen
roland.wismueller@uni-siegen.de
Tel.: 0271/740-4050, Büro: H-B 8404

Stand: October 21, 2025

Parallel Processing

Winter Term 2025/26

0 Organisation



- ➔ Studies in Computer Science, Techn. Univ. Munich
 - ➔ Ph.D. in 1994, state doctorate in 2001
- ➔ Since 2004 Prof. for Operating Systems and Distributed Systems
- ➔ **Research:** Secure component based systems; Using generative AI for teaching; Parallel and distributed systems
- ➔ Head of Examination Board Computer Science
- ➔ **E-mail:** roland.wismueller@uni-siegen.de
- ➔ **Tel.:** 0271/740-4050
- ➔ **Room:** H-B 8404
- ➔ **Office Hour:** Mo., 14:15-15:15



Andreas Hoffmann

andreas.hoffmann@uni-...

0271/740-4047

H-B 8405

- ➔ E-assessment and e-labs
- ➔ IT security
- ➔ Web technologies
- ➔ Mobile applications



Felix Breitweiser

felix.breitweiser@uni-...

0271/740-4719

H-B 8406

- ➔ Operating systems
- ➔ Programming languages
- ➔ Virtual machines



Sven Jacobs

sven.jacobs@uni-...

0271/740-2533

H-B 8407

- ➔ E-assessment and e-labs
- ➔ Generative artificial intelligence
- ➔ Web technologies

Lectures/Labs

- ➔ Rechnernetze I, 6 CP (Bachelor, summer term)
- ➔ Rechnernetze Praktikum, 6 CP (Bachelor, winter term)
- ➔ Rechnernetze II, 6 CP (Master, summer term)

- ➔ Betriebssysteme und nebenläufige Programmierung, 6 CP (Bachelor, summer term)
- ➔ Parallel processing, 6 CP (Master, winter term)
- ➔ Distributed systems, 6 CP (Bachelor, winter term)

Project Groups

- ➔ e.g., secure cooperation of software components
- ➔ e.g., concepts for secure management of Linux-based thin clients

Theses (Bachelor, Master)

- ➔ Topic areas: secure virtual machine, parallel computing, pattern recognition in sensor data, e-assessment, ...

Seminars

- ➔ Topic areas: IT security, programming languages, pattern recognition in sensor data, ...
- ➔ Procedure: block seminar (30 min. talk, 5000 words paper)
- ➔ Master: attend the lecture “Scientific Working” beforehand!
 - ➔ block course end of Feb. / beginning of March



Lecture

- ➔ Mon., 12:15-13:45, H-C 8326
- ➔ on Tue., 14.10., 21.10., 28.10. and 04.11. also in the lab slot:
10:15-11:45, H-C 8326

Practical labs

- ➔ Tue., 10:15-11:45, PC lab room H-C 4111 (and at home)
 - ➔ some exercises rely on software / hardware in the lab room
- ➔ Tutor: Felix Breitweiser (felix.breitweiser@uni-siegen.de)
 - ➔ for questions, help, and discussion of solutions



Information, slides, and announcements

- ➔ See the WWW page for this course
- ➔ <http://www.bs.informatik.uni-siegen.de/lehre/pv/>
- ➔ Annotated slides (PDF) available; maybe slightly modified

Moodle course

- ➔ <https://moodle.uni-siegen.de/course/view.php?id=23366>
- ➔ For students arriving late:
 - ➔ screen casts of the lectures until mid November
- ➔ Screen casts with tutorials on exercises / tools
- ➔ Submission of lab assignments



Learning targets

- ➔ Knowing the basics, techniques, methods, and tools of parallel programming
- ➔ Basic knowledge about parallel computer architectures
- ➔ **Practical knowledge/experience with parallel programming**
- ➔ Knowing and being able to use the most important programming models
- ➔ Knowing about the possibilities, difficulties and limits of parallel processing
- ➔ Being able to identify and select promising strategies for parallelization
- ➔ Focus: high performance computing



Methodology

- ➔ Lecture: Basics
 - ➔ theoretical knowledge about parallel processing
 - ➔ practical introduction to programming environments
 - ➔ “hands-on” tutorials
- ➔ Lab: practical use
 - ➔ **independent programming work (lab + homework!)**
 - ➔ programming language: C++
 - ➔ practical skills and experiences
 - ➔ in addition: raising questions
 - ➔ main task: parallelization of two representative problems
 - ➔ iterative, numerical method (Jacobi, Gauss/Seidel)
 - ➔ combinatoral search (Sokoban)



- ➔ Passing the course requires successful completion of the lab:
 - ➔ i.e., qualified attempt for all mandatory exercises
- ➔ You must register for '4INFMA024-S Coursework Parallel Processing' in unisono **before you can submit a solution!** (do it right now!)
 - ➔ independent of the registration to the course and the lab!
 - ➔ if you cannot complete the course: **deregister** again!

A screenshot of a course registration interface. At the top, a dropdown menu is open, showing a list of course components for '4INFMA024 - Parallel Processing'. The components are:

- 4INFMA024-VG1 - Parallel Processing (Lecture) - rec. S. 1 - Compulsory
- 4INFMA024-VG2 - Parallel Processing (Lab) - rec. S. 1 - Compulsory
- 4INFMA024-S - Coursework Parallel Processing - rec. S. 1 - Compulsory - 2.0 Credits

The third component, '4INFMA024-S', is highlighted in yellow. To the right of this component, there is a button labeled 'Apply' with a mouse cursor hovering over it.



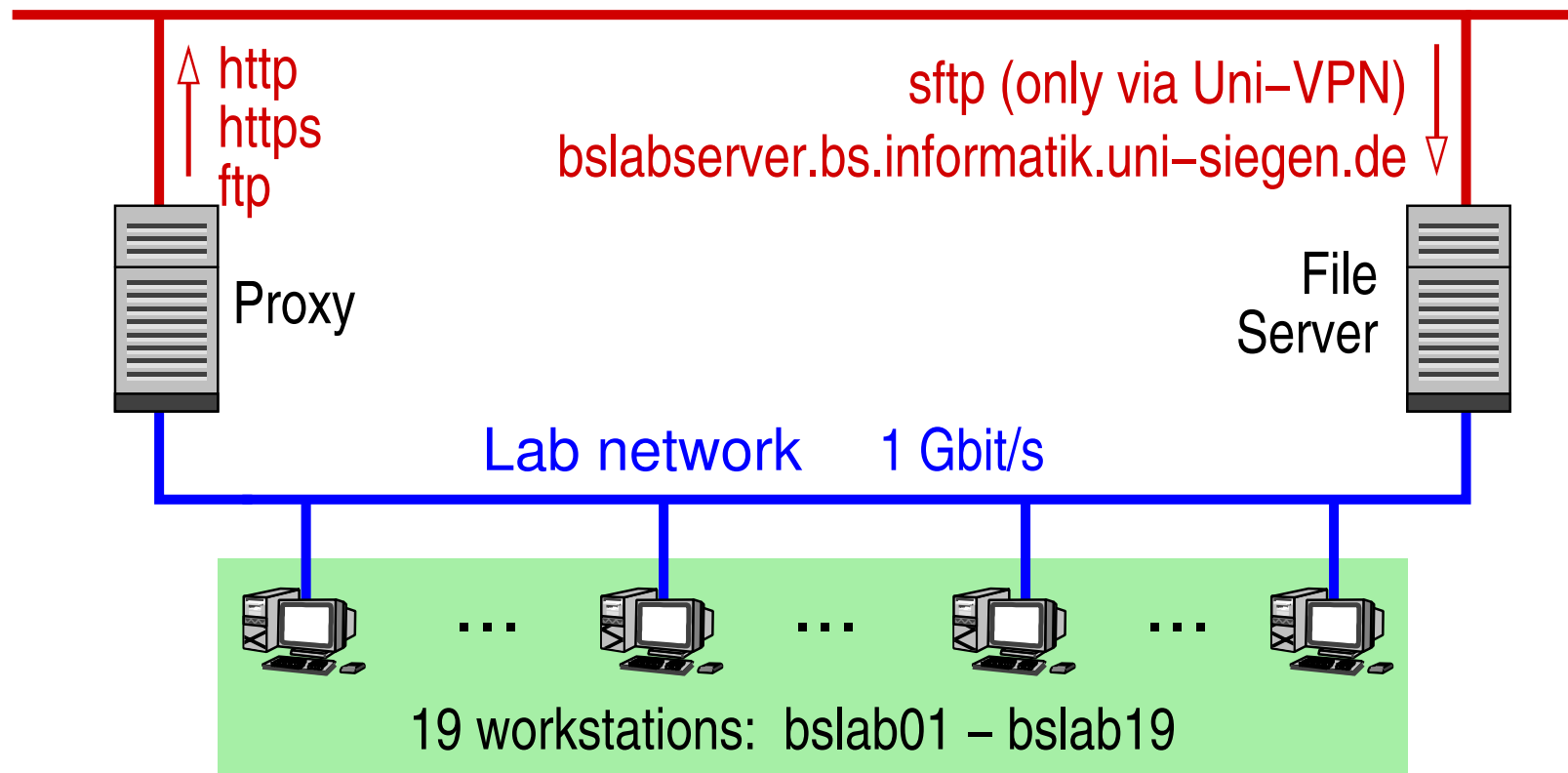
- ➔ Eight exercise sheets
 - ➔ each with 1-2 mandatory exercises of different weight (1-3)
 - ➔ **it'll be hard work!**
 - ➔ 6 LP = 180 hours, that is 60 h presence (lecture + lab), **60 h homework**, 60 h preparation for exam
 - ➔ solve exercises in the lab (H-A 4111) and at home!
 - ➔ solutions must be submitted via **Moodle** in due time
 - ➔ requirement for passing: at least **serious** attempt
 - ➔ codes must at least compile and execute in the lab
 - ➔ sometimes, additional answers / materials are required
 - ➔ **do not copy!** (both will get 0 points)
- ➔ Grading: yes/no
- ➔ You need at least 12 out of 16 possible points to pass



- ➔ Assignments should be done in the lab (H-A 4111), but can / should also be prepared at home
- ➔ Ideally, you have a Linux-PC with the GNU-compiler (g++)
 - ➔ Windows with MSVC will not work for all exercise sheets
- ➔ In addition, you should have MPI installed, preferable MPICH
 - ➔ see <https://www.mpich.org/downloads>
- ➔ User regulations and key card application form:
 - ➔ <http://www.bs.informatik.uni-siegen.de/lehre/pv>
 - ➔ please let me sign the key card application form and then deliver it directly to Mr. Kiel (AR-P 209)
 - ➔ by using the lab computers you accept the user regulations (see web page)
- ➔ **First lab hour: Tue. 11.11., H-A 4111**

- ➔ Linux-PCs, private IP network, but `sftp` access via VPN

Internet / group network (`bs.informatik.uni-siegen.de`)



Intel Core Ultra 7, 8+12 cores, 5.2/4.6 GHz, 32 GiB RAM
+ RTX 4060, 3072 cores, 1.8 GHz, 8 GiB RAM

- ➔ Written examination (60 minutes)
 - ➔ open book exam
 - ➔ electronic exam, computers provided by university
 - ➔ subject matter: lecture and labs!
 - ➔ examination also covers the practical exercises

- ➔ Application via unisono
 - ➔ **at least two weeks before the exam date (hard deadline!)**
 - ➔ exam date is published via unisono and course web page



- ➔ Repetition / Foundations
 - ➔ C/C++ for Java programmers
 - ➔ Threads and synchronisation
 - ➔ C++ threads

- ➔ Basics of Parallel Processing
 - ➔ Motivation, Parallelism
 - ➔ Parallelization and Data Dependences
 - ➔ Parallel Computers
 - ➔ Programming Models
 - ➔ Organisation Forms for Parallel Programs
 - ➔ Design Process
 - ➔ Performance Considerations



- ➔ Parallel Programming with Shared Memory
 - ➔ Basics
 - ➔ OpenMP
- ➔ Parallel Programming with Message Passing
 - ➔ Approach
 - ➔ MPI
- ➔ Optimization Techniques
 - ➔ Cache Optimization
 - ➔ Optimization of Communication



- ➔ Until November, 4th: only lectures (Mon. + Tue.)
 - ➔ no lab (but home work)
- ➔ Then: lectures (Mon.) and lab (Tue. + home work)
- ➔ Last three weeks: only lab (Tue. + home work)
- ➔ Prospective due dates for the assignments:
 - ➔ 11.11.: Exercise sheet 1
 - ➔ ... (see [web page](#))
 - ➔ 03.02.: Exercise sheet 8



- ➔ Currently no recommendation for a all-embracing text book

- ➔ Barry Wilkinson, Michael Allen: *Parallel Programming*. internat. ed, 2. ed., Pearson Education international, 2005.
 - ➔ covers most parts of the lecture, many examples
 - ➔ short references for MPI, PThreads, OpenMP

- ➔ A. Grama, A. Gupta, G. Karypis, V. Kumar: *Introduction to Parallel Computing*, 2nd Edition, Pearson, 2003.
 - ➔ much about design, communication, parallel algorithms

- ➔ Thomas Rauber, Gudula Rünger: *Parallele Programmierung*. 2. Auflage, Springer, 2007.
 - ➔ architecture, programming, run-time analysis, algorithms



- ➔ Theo Ungerer: *Parallelrechner und parallele Programmierung*, Spektrum, Akad. Verl., 1997.
 - ➔ much about parallel hardware and operating systems
 - ➔ also basics of programming (MPI) and compiler techniques
- ➔ Ian Foster: *Designing and Building Parallel Programs*, Addison-Wesley, 1995.
 - ➔ design of parallel programs, case studies, MPI
- ➔ Seyed Roosta: *Parallel Processing and Parallel Algorithms*, Springer, 2000.
 - ➔ mostly algorithms (design, examples)
 - ➔ also many other approaches to parallel programming



- ➔ S. Hoffmann, R. Lienhart: *OpenMP*, Springer, 2008.
 - ➔ handy pocketbook on OpenMP
- ➔ W. Gropp, E. Lusk, A. Skjellum: *Using MPI*, MIT Press, 1994.
 - ➔ the definitive book on MPI
- ➔ D.E. Culler, J.P. Singh: *Parallel Computer Architecture - A Hardware / Software Approach*. Morgan Kaufmann, 1999.
 - ➔ UMA/NUMA systems, cache coherency, memory consistency
- ➔ Michael Wolfe: *Optimizing Supercompilers for Supercomputers*, MIT Press, 1989.
 - ➔ details on parallelizing compilers