Module Description, available in: EN

Parallel and distributed computing

General Information

Number of ECTS Credits

3

Module code

TSM_ProgAlg

Valid for academic year

2026-27

Last modification

2024-09-27

Coordinator of the module

Christoph Stamm (FHNW, christoph.stamm@fhnw.ch)

Explanations regarding the language definitions for each location:

- Instruction is given in the language specified for each location and module execution.
- Documentation is available in the language(s) listed for each location and module execution. If the documentation is in multiple languages, the percentage distributed is indicated (100% = all documentation provided).
- The examination, including both questions and answers, is provided entirely (100%) in the language(s) specified for each location and module execution. The exams are on-site.

	Lausanne			**Lugano**	**Zurich**		
Instruction					**X** E 100%		
Documentation					**X** E 100%		
Examination					**X** E 100%		

Module Category

TSM Technical scientific module

Lessons

2 lecture periods and 1 tutorial period per week

Entry level competences

Prerequisites, previous knowledge

- Procedural programming (C/C++ knowledge is helpful)
- Algorithms and data structures
- Basic notion of complexity analysis and big-O notation
- Basic notions of concurrent programming (Threads)

Brief course description of module objectives and content

The objective of this module is to provide the student with an introduction to parallel computing and algorithms. Students learn to parallelize systems at three different levels: Shared memory systems, distributed memory systems, and heterogeneous shared memory systems. For all three systems, which can be part of a high-performance cluster, students learn the necessary parallelization techniques and a few classical parallel algorithms.

This course includes hands-on work to train students in the application of parallel programming techniques and the programming and analysis of parallel algorithms.

Aims, content, methods

Learning objectives and competencies to be acquired

By the end of the course, students will know:

- the most common heterogeneous parallel hardware infrastructures
- different ways to model and efficiently program these architectures
- how to assess the performance of parallel algorithms
- how to efficiently implement parallel algorithms
- how to choose the right parallel algorithm to solve a specific problem on a specific parallel architecture

Module content with weighting of different components

Introduction (30%)

- Motivation and high-performance cluster model
- Basic concepts in parallel programming in C++
- Asymptotic analysis of parallel programs
- Performance metrics for parallel systems
- Decomposition and mapping techniques for load balancing
- Scalability of parallel systems

Shared memory systems (20%)

- Architectures of widely used multi-core systems
- Cache coherence in multiprocessor systems
- Parallel programming models (OpenMP)
- Parallel algorithms (numerical, sorting, graphs)

Distributed memory systems (30%)

- Architectures of distributed memory systems
- Communications models and communication costs
- Message passing paradigm (MPI)
- Collective operations and their costs
- Distributed algorithms (numerical, sorting)

Heterogeneous shared memory systems (20%)

- GPU architectures
- SIMD programming models (vectorization, SYCL)
- Matrix-vector and matrix-matrix multiplication
- GPU algorithms (numerical, image processing)

Teaching and learning methods

The most important basics are taught in theory lessons in face-to-face classes. In between, students work on the practical exercises at home and exchange information via online channels.

Literature

A. Introduction to Parallel Computing, Trobec, R. et al., Springer, 2018
B. SYCL. Data Parallel C++, Reinders, J., Ashbaugh, B., et al., Apress, 2nd Edt., 2023.

Assessment

Additional performance assessment during the semester

The module does not contain an additional performance assessment during the semester

Basic principle for exams

As a rule, all standard final exams are conducted in written form. For resit exams, lecturers will communicate the exam format (written/oral) together with the exam schedule.

Standard final exam for a module and written resit exam

Kind of exam

Written exam

Duration of exam

120 minutes

Permissible aids

Aids permitted as specified below:

Permissible electronic aids

No electronic aids permitted

Other permissible aids

A handwritten summary of a fixed number of pages given by the lecturer.
Exception: In case of an electronic Moodle exam, adjustments to the permissible aids may occur. Lecturers will announce the final permissible aids prior to the exam session.

Special case: Resit exam as oral exam

Kind of exam

Oral exam

Duration of exam

30 minutes

Permissible aids

No aids permitted