Module Description

Parallel computation and algorithms

General Information

Number of ECTS Credits

3

Abbreviation

TSM_ProgAlg

Version

04.03.2013

Responsible of module

Pierre Kuonen

Language

	Lausanne	Bern	Zürich
Instruction	☐ E X F	☐ D ☐ E ☐ F	X D X E
Documentation	X E X F	☐ D ☐ E ☐ F	☐ D X E
Examination	☐ E X F	☐ D ☐ E ☐ F	☐ D X E

Module category

☐ Fundamental theoretical principles

X Technical/scientific specialization module

☐ Context module

Lessons

☒ 2 lecture periods and 1 tutorial period per week

☐ 2 lecture periods per week

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. The first part of the course will be dedicated to the architectures of parallel infrastructures, the different theoretical models for these architectures and the different programming models and tools for programming such architectures. The second part will be dedicated to the study of a number of classical parallel algorithms. This course includes practical work to train the student in the use of parallel computing.

Aims, content, methods

Learning objectives and acquired competencies

At the end of the course the student knows:

- The most common heterogeneous parallel hardware infrastructures
- The different ways to model and efficiently program these architectures
- How to choose the proper parallel algorithm to write an application for solving a specific problem on a specific architecture
- How to efficiently program this application
- How to assess the performance of this application

Contents of module with emphasis on teaching content

- Introduction
  - Different architectures of parallel infrastructures
  - Communications models and communication costs
  - Performance metrics for parallel systems
  - Scalability of parallel systems
- Heterogeneous shared memory systems
  - Architecture of widely used multi-core systems with attached accelerators (e.g. GPUs)
  - Parallel programming models (OpenCL, OpenMP)
- Distributed memory systems
  - Communication operations and their costs
  - Message passing paradigm (MPI)

- Distributed object paradigm (POP-C++)

- Parallel algorithms
  - Asymptotic analysis of parallel programs
  - Decomposition techniques
  - Mapping techniques for load balancing
  - Matrix-vector and matrix-matrix multiplication
  - Parallel sorting algorithms
  - Parallel Graph and optimization algorithms

Teaching and learning methods

- This course involves theoretical presentations and practical exercises or laboratories. Some of the exercises or laboratories are programming exercises that can be done at home by accessing a parallel infrastructure made available through the internet.

Prerequisites, previous knowledge, entrance competencies

- Procedural and object oriented programming
- Software engineering (UML or other)
- Basic notion of algorithms

Literature

A. Grama, A. Gupta, G. Karypis and V.Kumar, "Introduction to Parallel Computing," Addison Wesley

Assessment

Certification requirements for final examinations (conditions for attestation)

Some exercises or laboratories could be mandatory.

Written module examination

Duration of exam :	120 minutes
Permissible aids:	A hand-written summary of one A4 page.