Technology
Arts Sciences
TH Köln

Course Manual PPRA

Parallel Programming and Computerarchitektur

Version: 3 | Last Change: 29.04.2022 08:41 | Draft: 2 | Status: Entwurf

General information

Long name	Parallel Programming and Computerarchitektur
Approving CModule	PPRA_BaTIN
Responsible	Prof. Dr. Lothar Thieling Professor Fakultät IME
Valid from	summer semester 2021
Level	Bachelor
Semester in the year	summer semester
Duration	Semester
Hours in self-study	60
ECTS	5
Professors	Lehrbeauftragte(r) / Thieling
Requirements	basic skills in procedural programming basic skills in programming multiple tasks structure and mode of operation of a simple computer basics in digital systems (Automata, Hardware Description Language)
Language	German
Separate final exam	Yes

Literature

Barlas, Gerassimos: Multicore and GPU Programming: An Integrated Approach

Tanenbaum, Goodman: Computerarchitektur, Pearson Studium (Prentice Hall)

Pacheco: Parallel Programming with MPI

Eijkhout: Introduction to High Performance Computing

Final exam

Details	The students should demonstrate the following competencies in a written exam: The students should demonstrate the following skills in a written exam: 1.) Confident handling of basic terms, mechanisms and concepts. 2.) Parallel programming using common design tools (e.g. MPI and CUDA). 3.) Development of problem solutions that are predestined for the use of parallel computer systems.
Minimum standard	At least 50% of the total number of points
Exam Type	EN Klausur

− Lecture / Exercises

Learning goals

Goal type	Description
Knowledge	basics of parallel programming introduction approach/basic idea Data dependencies and synchronization Parallel computer architectures classification MMID SIMD
Knowledge	design of parallel programs development process decomposition pattern completely parallel task parallelism (incl. task pool) divide and conquer pipeline (or general task graph) data parallel (geometric data) recursive data
Knowledge	design of parallel programs design pattern for parallel programming master slave (master worker) fork and join single program multiple data (SPMD) multiple program multiple data (MPMD) map reduce loop parallelism mapping of program structure patterns to decomposition patterns
Knowledge	design of parallel programs performance Metrics speedup amdahl's law efficiency scalability loss of performance load balancing performance measurement
Knowledge	classification of standard libraries with regard to the preceding design options and their use based on design patterns MPI (distributed memory) CUDA (GPU programming)

Special requirements

none

Accompanying material	lecture slides (electronic), set of exercises (electronic), tool chains (compile, link, debug, simulate), set of example program codes
Separate exam	No

Knowledge	computer architectures (according to Von-Neumann) conceptual components to increase performance regarding ... storage processing units GPU (see above) communication protection
Knowledge	implementation of the above concepts in concrete computer architectures IA32e (AMD64) ARM
Knowledge	alternative architectures in addition to von-neuman connection of FPGAs to von veumann architectures veural networks implemented in FPGAs
Skills	The students are able to - discuss the structure, organization and operating principle of computer systems, - analyze the connection between hardware concepts and the effects on the software, to be able to create efficient programs, - to understand and apply the basic principles of design from the understanding of the interactions of technology, computer concepts and applications, - evaluate and compare computer concepts.
Skills	The students are able to - describe architectural features of parallel computers, - evaluate parallel computers, programming paradigms and design patterns and select them for a specific application, - to program parallel computers
Skills	specifying system behavior (derived from text documents)

Expenditure classroom teaching

Type	Attendance (h/Wk.)
Lecture	2
Exercises (whole course)	1

Exercises (shared course)	1
Tutorial (voluntary)	0

− Practical training

Learning goals

Goal type	Description
Skills	refer to "Vorlesung/Übung->Lernziele->Fertigkeiten"
Skills	targeted use of the software development environment
Skills	manage complex tasks as a small team
Skills	Development of more complex solutions to problems in the field of compute/data intensive algorithm, signal processing or artificial intelligence or graphic animation that are specific for the use of parallel computers.

Expenditure classroom teaching

Type	Attendance (h/Wk.)
Practical training	1
Tutorial (voluntary)	0

Special requirements

none

Accompanying material	problem and task description (electronic), tool chaine (compile, link, debug, simulate), set of example-codes, self-study tutorials for the tool chain
Separate exam	No