



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Parallel processing [N1Inf1>PROW]

Course

Field of study

Computing

Year/Semester

3/5

Area of study (specialization)

–

Profile of study

general academic

Level of study

first-cycle

Course offered in

Polish

Form of study

part-time

Requirements

compulsory

Number of hours

Lecture

12

Laboratory classes

12

Other

0

Tutorials

0

Projects/seminars

0

Number of credit points

3,00

Coordinators

dr inż. Rafał Walkowiak

rafal.walkowiak@put.poznan.pl

Lecturers

Prerequisites

The student starting this course should have knowledge of the computer organization, algorithms and data structures, operating systems and programming in the C language.

Course objective

Providing students with basic knowledge of parallel processing, including: models, computing systems, environments and languages, problems and methods of solving them. Developing students' skills in: solving tasks in the field of processing and optimization of processing in a parallel computing system, comparing the effectiveness of parallel processing carried out with the use of various environments and systems. Developing in students awareness of the need to use tools that allow for the assessment of processing efficiency in parallel processing systems (profilers), understanding the principles of their operation.

Course-related learning outcomes

Knowledge:

The student has ordered, theoretically founded knowledge of parallel programming paradigms, parallel computer systems, parallel processing algorithms and their complexity evaluation.

The student has knowledge of the directions of development of architectures of parallel computer systems.

The student knows the methods, techniques and tools used to solve computer science tasks in the field of parallel programming.

Skills:

The student is able to properly plan and perform experiments to evaluate the effectiveness of parallel processing, interpret the results and correctly draw conclusions from them.

The student is able to solve problems in the field of parallel processing using appropriate experimental methods.

The student has the ability to formulate parallel algorithms and their implementation in the OpenMP and CUDA environment.

Social competences:

The student is aware of the importance of the knowledge - regarding both hardware and software of parallel systems - needed in solving problems in the field of parallel processing optimization.

The student understands the need to broaden the knowledge and skills resulting from technological progress in the field of parallel processing equipment.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

The intended learning outcomes in the laboratory are assessed through:

- in-class discussion of the results obtained during the laboratory task,
- in-class defense of the project task reports by students (understanding the issues described in the report is required),
- evaluation of the laboratory project reports by the instructor.

The assessment of understanding and practical application of the topics presented in lectures is carried out in the form of an in-person exam with problem-solving questions involving (for example):

- analysis of the described situation occurring in a parallel system,
- checking the correctness and potential processing efficiency of the given code,
- proposing code in OpenMP or CUDA illustrating the required data processing method,
- calculating parallel processing efficiency measures for the described situation,
- explaining concepts presented in lectures or in supporting materials.

Sample tasks from previous years are made available to students and the correct solutions to the tasks are presented during consultations to students presenting their solutions.

Programme content

The program includes processor cache memory and coherency of cache in multiprocessor systems, spatial and temporal locality of data access, efficiency evaluation of parallel algorithms and parallel systems, parallel programming environments: OpenMP, CUDA.

Course topics

The purpose of this course is to provide information concerning:

- internal parallelism of general purpose single processor computing systems, pipelining and super scalar architecture;
- classifications and examples of parallel systems (multi core systems, GPU);
- processor cache memory and coherency of cache in multiprocessor systems;
- the importance of spatial and temporal locality of data access for processing efficiency;
- basics of efficiency evaluation of parallel algorithms and parallel systems, scalability, Amdahl and Gustafson law;
- parallel processing models (shared memory, message passing, data parallelism);
- parallel algorithms: a general method of algorithm design, problem partition techniques, task assignment methods (divisible task method, task graph embeddings, load balancing);
- parallel programming environments and standards: OpenMP, CUDA;
- tools for parallel processing efficiency evaluation: Nvidia Profiler;
- parallel algorithms for sorting, reduction, stencil computation, matrix multiplication, primary numbers.

The lab-classes are focused on: practical exercises with software implementations (in OpenMP, CUDA

environments) for selected problems in parallel multicore systems and GPU, data sharing in multicore architectures and efficiency evaluation of different approaches.

Teaching methods

Lectures: presentation illustrated with examples on a black board.

Labs: presentation of work and use of parallel environments and tools; discussion of the results and parallel processing efficiency for experiments performed on code prepared by students in various systems.

Bibliography

Basic

1. Wprowadzenie do obliczeń równoległych, Z. Czech, PWN, Warszawa, 2013.

2. Cuda w przykładach: wprowadzenie do ogólnego programowania procesorów GPU, J.Sanders, E.Kandrot, Helion, 2012.

3. Introduction to Parallel Computing, A.Grama, A.Gupta, G.Karypis,V.Kumar, Addison Wesley, 2003

Additional

Specifications and manuals for OpenMP, CUDA.

Breakdown of average student's workload

	Hours	ECTS
Total workload	75	3,00
Classes requiring direct contact with the teacher	26	1,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	49	2,00