COURSE DESCRIPTION CARD - SYLLABUS

Course name
Large Scale Computing [S2Bioinf1>OWS]

Course

Field of study
Bioinformatics

Year/Semester
1/2

Area of study (specialization)
–

Profile of study
general academic

Level of study
second-cycle

Course offered in
polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
15

Laboratory classes
30

Other (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

4,00

Coordinators

dr inż. Rafał Walkowiak
rafal.walkowiak@put.poznan.pl

Lecturers

dr inż. Rafał Walkowiak
rafal.walkowiak@put.poznan.pl

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

Provide students with basic knowledge of parallel and distributed 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 and distributed computing systems, comparing the effectiveness of parallel processing carried out with the use of various environments and equipment.

Course-related learning outcomes

Knowledge:
The student has ordered, theoretically founded knowledge of parallel programming paradigms, parallel adn distributed computer systems and environments, 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 and distributed 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 and environments.
The student has the ability to formulate parallel algorithms and their implementation in the MPI, OpenMP and CUDA environments.
The student is able to cooperate and work in a group by carrying out projects and research on the implementation and evaluation of the effectiveness of parallel algorithms.

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:

Learning outcomes presented above are verified as follows:
Formative assessment:
- during lectures is based on answers to questions related to the material covered in previous lectures;
- during laboratory classes is based on assessment of the current progress and the results of the ongoing and finished tasks and projects .
Total assessment:
- verification of assumed learning objectives related to lectures is performed by assessment of the knowledge and skills during a final test consisting of close-ended questions of multiple-choice answers and hand assessed open-ended questions of a problem nature; exemplary test task from previous years are available for students in a form of a list.
- verification of assumed learning objectives related to laboratory classes by evaluation of student's knowledge necessary to prepare and carry out the lab tasks, monitoring students' activities during classes, evaluation of laboratory reports (partly started during classes, finished after them).

Programme content

Internal parallelism of general purpose single processor computing systems, pipelining and super scalar architecture.
Classifications and examples of parallel systems (multi core systems, GPU and multicomputer systems).
Processor cache memory and coherency of cache in multiprocessor systems.
Communication in parallel systems - method of communication (synchronic, asynchronic, buffered, unbuffered), cost of communication, algorithms of group communication.
Basics of efficiency evaluation of parallel algorithms and parallel systems, scalability, Amdahl and Gustafson lows.
Parallel processing models (shared memory, message passing, data paralelism).
Parallel programming environments and standards: OpenMP, CUDA, MPI.
Parallel algorithms for sorting, matrix multiplication, primary numbers search.
The lab-classes are focused on: practical exercises with software implementations (Open MP, CUDA, MPI) for selected problems in a network of computing systems, parallel multicore systems and GPU, efficiency evaluation of different approaches.

Teaching methods

Lectures: presentation illustrated with examples on a black board, homeworks connected to the lecture.
Labs: presentation of work, configuration and use of parallel and distributed 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 graphic cards, OpenMP, MPI, CUDA and Parallel Studio.

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	45	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	55	2,00