KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY

Course title	ECTS code
Parallel programming in Python	13.3.1301

Name of unit administrating study

null

Studies

faculty	field of study		
Faculty of Chemistry	Chemistry	type	second tier studies (MA)
		form	full-time
		specialty	all
		specialization	all

Teaching staff

prof. dr hab. Cezary Czaplewski, profesor uczelni; dr hab. Adam Sieradzan, profesor uczelni; dr hab. Artur Giełdoń

Forms of classes, the realization and number of hours

Forms of classes

Laboratory classes

The realization of activities

classroom instruction

Number of hours

Laboratory classes: 30 hours

ECTS credits

2
laboratory classes – 30 h
student's own work – 10 h
tutorial classes – 10 h
Total: 50 h – 2 ECTS

The academic cycle

2022/2023 summer semester

Type of course

an elective course

Language of instruction

english

Teaching methods

Case studies in computer laboratory

Form and method of assessment and basic criteria for eveluation or examination requirements

Final evaluation

Graded credit

Assessment methods

- completion of the final project (design and programming of a parallel application or parallelization of a serial program)
- completion of all assigned projects during classes in the computer lab
- written report for each assigned project

The basic criteria for evaluation

- correctness of the reports on assigned projects, the final grade of the lab. is based on the partial grades received from each report and presentation of the final project; failure to complete the experimental part means failing the laboratory exercises

Method of verifying required learning outcomes

The method of verifying the acquisition of knowledge:
oral presentation and argumentation during the discussion.
The method of verifying the acquisition of skills: the student solves problems in writing (reports including program codes) or oral (oral answer).
The method of verifying the acquisition of social competences:
observation of the student's behavior during classes and during consultations

Required courses and introductory requirements

A. Formal requirements

Introduction to Python programming

B. Prerequisites

basis of calculus and linear algebra, ability to use the LINUX operating system

Aims of education

Introduction to parallel programming using MPI library in Python. Effectively design and conduct parallel computing.

Course contents

Parallel programming as an essential method in computational chemistry. Types of computer architectures used for parallel computing, shared and distributed memory. Scalability of parallel computing: Amdahl's law. Running parallel tasks on computing clusters - queuing systems. Parallel programs with the use of message passing interface (MPI) library. Initialization and termination of references to MPI libraries in Python programs. Point to point communication: safety and avoiding deadlock. Collective communication. Process groups and messengers. Intergroup communication. Data types and user operators in MPI. Virtual topologies. Libraries in MPI: rules of creation. Evaluation of the effectiveness of parallelization and profiling of parallel programs. MPI extensions (MPI2 and MPI3): MPI-IO, remote memory operations, dynamic process management.

Bibliography of literature

Literature required to pass the course
W. Gropp, E. Lusk, A. Skjellum, Using MPI. Portable Parallel Programming with the Message-Passing Interface, The MIT Press, Cambridge, 1999.
W. Gropp, E. Lusk, R. Thakur, Using MPI-2. Advanced Features of the Message-Passing Interface. The MIT Press, Cambridge, 1999.
MPI for Python https://mpi4py.readthedocs.io/en/stable/

Extracurricular readings
I. Foster, Designing and Building Parallel Programs, Addison Wesley, 1995
M. Snir, S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra, MPI: the Complete Reference, The MIT Press, 1995

The learning outcomes (for the field of study and specialization)

K_W05: has extended knowledge in the field of the specialisation studied

K_W06: applies mathematics to the extent necessary to understand, describe and model chemical processes of extended complexity

K_U02: critically assesses the results of conducted, performed observations and theoretical calculations and discusses errors

K_K01: knows the limitations of her/his own knowledge; understands the need for further education and can inspire other people to do so

K_K06: undertakes research tasks consciously and responsibly, understanding the social aspects of the practical application of the acquired knowledge and skills and the responsibility related to it

Knowledge

Student recognizes and characterizes parallel computer architectures, differentiate parallel libraries and tools for parallel programming, knows function from MPI library

Skills

Student evaluates the usefulness of parallel programming to solve a given problem, runs parallel applications in batch and interactive mode, analyses parallel source codes, creates simple parallel codes using python with MPI library.

Social competence

The student learns the principles of safe, responsible and effective work on supercomputers in computing centers and on local computer clusters.

Contact

cezary.czaplewski@ug.edu.pl