Course Unit Descriptor

Study Programme: Applied Mathematics – Data Science
Course Unit Title: High Performance Computing
Course Unit Code: MDS29
Name of Lecturer(s): Srdjan Skrbic
Type and Level of Studies: Master Academic Degree
Course Status (compulsory/elective): elective
Semester (winter/summer): Winter
Language of instruction: English
Mode of course unit delivery (face-to-face/distance learning): Face-to-face
Number of ECTS Allocated: 5
Prerequisites: None
Course Aims: High performance computing is seen as the only cost-effective method for fast solution of computationally large and data-intensive problems. That is why the objective of this course is to study the principles, tools, and techniques for programming the wide variety of parallel and high performance platforms currently available.
Learning Outcomes: *Minimal:* At the end of the course, it is expected that students understand and show ability to discuss advantages and disadvantages of different high performance architectures and paradigms. The knowledge of parallel programming using the message passing paradigm is a must for every student. *Desirable:* At the end of the course, it is expected that a successful student understand solutions to key problems in high performance computing and shows ability to identify the optimal way of solving particular given problem. From a successful student is also expected to have active knowledge of advanced concepts of parallel programming using the message passing paradigm.
Syllabus: *Theory* At the beginning of the course, introduction to high performance computing including motivation and fields of application is explained. After that, parallel architectures and platforms are examined. In the third part of the course, principles of design of parallel algorithms, decomposition techniques and models of parallel algorithms are studied. The final part of the course is dedicated to details related to parallel programming using the message passing paradigm. *Practice* In the first part of the practical classes, ways to connect computers to a computer cluster or a grid are examined and practically demonstrated. The rest of the practical instruction is spent on mastering practical skills of parallel programming using message passing paradigm through analysis of a number of examples and case studies.
Required Reading: 1. Grama, A., Gupta, A., Karypis, G., Kumar, V.: Introduction to Parallel Computing, 2nd Edition, Addison-Wesley, 2003. 2. Pacheco, P.: An Introduction to Parallel Programming, Morgan Kaufmann, 2011. 3. Karniadakis G, Kirby, R.: Parallel Scientific Computing in C++ and MPI, Cambridge University Press, 2002.

4. Gropp, W., Lusk, E. and Skjellum, A.: Using MPI, MIT Press, 2014.		
Weekly Contact Hours:	**Lectures:** 2	**Practical work:** 2

Teaching Methods:

During theoretical classes, classical methods of teaching with the use of a projector are used to present stated topics. On practical classes, classical methods of teaching with the use of a projector and computers with appropriate software installed are used to practically master the skills of usage of suggested tools. A premise for successful practical classes is the existence of enough computers so that every student may work individually.

Knowledge Assessment (maximum of 100 points):

Pre-exam obligations	points	**Final exam**	points
Active class participation		written exam	
Practical work		oral exam	50
Preliminary exam(s)	50	…….	
Seminar(s)			