Study programme(s): Computer Science (CS)

Level: master

Course title: High Performance Computing (CS708)

Lecturer: Srđan M. Škrbić

Status: elective

ECTS: 6

Requirements: none

Learning objectives

High performance computing is being seen as the only cost-effective method for the 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 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 parallel 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 parallel programming and shows ability to identify the optimal way of solving particular given problem using parallel programming. From a successful student is also expected to have active knowledge of advanced concepts of parallel programming using the message passing paradigm and shared memory parallel programming tools.

Syllabus

Theoretical instruction

At the beginning of the course, introduction to parallel processing 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. Second part of the course is dedicated to shared memory parallel programming using OpenMP. In the final and most important part of the course, students learn how to program computer clusters using the message passing paradigm.

Practical instruction

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 second part is dedicated to practical usage of OpenMP. 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.

Literature

1. Grama, A., Gupta, A., Karypis, G., Kumar, V.: Introduction to Parallel Computing, 2nd Edition, Addison-Wesley, 2003.
2. Pacheco, P.: An Inproduction to Parallel Programming, Morgan Kaufmann, 2011.
3. Karniadakis G, Kirby, R.: Parallel Scientific Computing in C++ and MPI, Cambridge University Press, 2002.
4. Pacheco, P.: A User's guide to MPI, University of San Francisco, 1998.

Weekly teaching load				Other:
Lectures: 2	Exercises: 0	Other forms of teaching: 2	Student research:	

Teaching methodology

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.

Grading method (maximal number of points 100)			
Pre-exam obligations	**points**	**Final exam**	**points**
Two colloquiums	**25, 25**	Oral exam	*50*