Study programme(s): Computer Science

Level: Master of Science

Course title: Distributed Programming

Lecturer: Miloš Savić

Status: elective

ECTS: 6

Requirements: Object-oriented programming 1, Operating systems 1

Learning objectives

The primary objective of the course is to familiarize students with distributed programming techniques and models focusing on scalable big data platforms and modern frameworks for data-intensive distributed computing.

Learning outcomes

Minimum: Successful students should be capable to develop simple distributed Java applications running on Hadoop clusters.

Desirable: At the end of the course it is expected that successful students deeply understand distributed computing programming models and are able to develop distributed Java applications for large-scale data processing in various domains.

Syllabus

Theoretical instruction

Introduction to distributed computing programming models. Overview of programming languages for distributed computing. The concept of big data and abstractions for scalable, fault-tolerant, data-intensive computing. Basics of MapReduce programming model (functional programming roots, mappers and reducers, MapReduce execution framework, combiners and partitioners). Distributed file systems. Introduction to Apache Hadoop, the Hadoop software ecosystem and the Hadoop cluster architecture. MapReduce design patterns and examples of MapReduce algorithms (counting, sorting, relational algebra operations, matrix multiplication, etc). MapReduce graph algorithms. MapReduce algorithms for large-scale information retrieval and data analysis. Processing rapid, high-speed data streams. Limitations and extensions of MapReduce and alternative programming models. Introduction to Apache Pig, PigLatin and other dataflow languages.

Practical instruction

Practical programming tasks related to the development of distributed Java applications based on the MapReduce programming model and the Hadoop framework.

Literature

Recomended

Jimmy Lin and Chris Dyer. *Data-Intensive Text Processing with MapReduce*. Morgan & Claypool Publishers, 2010.

Tom White. *Hadoop: The Definitive Guide*, 4th Edition. O'Reilly Media, 2015.

Donald Miner and Adam Shook. *MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems*, 1st Edition. O'Reilly Media, 2012.

Weekly teaching load				
Lectures: 2	Exercises:	Practical Exercises: 2	Student research:	Other:

Teaching methodology

Theoretical classes are based on the classical teaching model involving a projector. Presented algorithms, techniques and models are augmented with illustrative case studies implemented in Java. At practical exercises organized in computer labs, students have to individually solve practical programming problems related to the development of distributed Java applications running on Hadoop clusters. To approach the oral exam, students have to pass pre-exam obligations consisting of one theoretical test and two practical programming tasks.

Grading method (maximal number of points 100)

Pre-exam oblications	**points**	**Final exam**	**points**
Theoretical test	20	Oral examination	40
Practical tasks	40		