COURSE DESCRIPTION CARD - SYLLABUS

Course name

Large Scale Distributed Systems

Course

Field of study	Year/Semester
Computing	1/2
Area of study (specialization)	Profile of study
Distributed systems	general academic
Level of study	Course offered in
Second-cycle studies	Polish
Form of study	Requirements
full-time	compulsory

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
30	30	0
Tutorials	Projects/seminars	
0	0	

Number of credit points

5

Lecturers

Responsible for the course/lecturer:

dr hab. inż. Anna Kobusińska

email: anna.kobusinska@put.poznan.pl

tel. 61 6652964

Instytut Informatyki

ul. Piotrowo 2, 60-965 Poznań

Responsible for the course/lecturer:

dr inż. Tadausz Kobus

email: Tadeusz.Kobus@cs.put.poznan.pl

tel. 61 6652964

Instytut Informatyki

ul. Piotrowo 2, 60-965 Poznań

Prerequisites

Learning objectives of the first cycle studies defined in the resolution of the PUT Academic Senate that are verified in the admission process to the second cycle studies. The learning objectives are available at the website of the faculty www.fc.put.poznan.pl. In particular, students starting this course should have basic knowledge of operating systems, distributed processing, distributed algorithms, and computer networks.

Students should also be capable of continuous learning and knowledge acquisition from selected sources, understand the need to expand their competences, as well as express the readiness for collaborating as part of a team.

Course objective

The objective for this course is to give the students knowledge in the field of large scale distributed systems, presentation of theoretical and practical aspects of the design of such systems, as well as developing students' skills in solving processing problems in a large scale distributed environments.

Course-related learning outcomes

Knowledge

1. Students posesses well-grounded knowledge on key issues in the field of algorithms and their complexity, computer system architecture, operating systems, network technologies, programming languages and paradigms

2. Students have advanced and detailed knowledge related to selected issues from the field of IT, such as: architecture and classification of large scale distributed systems and communication environments, epidemic and gossiping communication protocols, big data processing

3. Students have knowledge about development trends and the most important new achievements of IT field, and other related scientific disciplines

4. Students have advanced and detailed knowledge of the life cycle processes of hardware or software information systems

Skills

1. Students can obtain information from literature, databases and other sources (both in Polish and English), integrate and interpret them, provide their critical evaluation, draw conclusions and formulate and exhaustively justify opinions

2. Students can use analytical, simulation and experimental methods to formulate and solve engineering tasks and simple research problems

3. Students can integrate knowledge from various areas of computer science (and, if necessary, knowledge from other scientific disciplines) when formulating and solving engineering tasks. They also can apply a systemic approach, taking into account also non-technical aspects

4. Students can assess the usefulness and the possibility of using new achievements (methods and tools) and new IT products

5. Students are able to assess the usefulness of methods and tools for solving an engineering task related to the construction or evaluation of an information system and its components, as well as assess the limitations of these methods and tools

6. Studnets can - using, among others conceptually new methods - solve complex IT problems, including non-standard problems and research problems

Social competences

1. Students understand the importance of using the latest knowledge from the field of computer science in solving research and practice problems

2. Students understand the importance of popularizing the latest achievements in the field of computer science

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

The knowledge acquired during lectures is verified during a problem-based written exam that consists of 4-5 open questions. The maximum number of points per question is 10. To pass the exam students must obtain at least 50% of the total points.

The skills acquired during the exercises are verified in the following way:

- assessment of the students' preparation for classes ("entrance" test),

- continuous assessment during each class (oral answers),

- assessment of knowledge and skills obtained during solving the project

It is possible to get additional points for activity during classes, especially during discussing additional aspects of the considered problems. Passing score: 50% of total points.

Programme content

Lectures cover the following topics:

1. Presentation of the challenges related to the construction and processing in large-scale distributed systems: systems' classification,processing in large-scale systems, big data definitions, big data sources, big data processing aspects

2. Introduction to NoSQL databases: classification (key value, column-oriented, document-oriented, column-oriented, graph-oriented models); construction of NoSQL systems (data partitioning, load balancing, replication, data verioning, membership management, failure handling) on the example of Amazon Dynamo; Google BigTable, HBase, Cassandra.

3. Processing big data using Resilient Distributed Datasets (RDD) and Apache Spark platform

4. Mesos and YARN resource management systems (architecture, resource allocation algorithms)

5. Distributed file systems

6. Architectures of P2P systems (structured, unstructured, hybrid); organization of nodes, topologies, scalability, load balancing; gossip-based small word networks, Kleinberg's peer sampling

7. Unstructured P2P systems: node attachment / detachment, peer-sampling, resource search strategies (flooding-based multicast, tree-based multicast, random walk, expanding ring, randezvous point, bubblecast) on the example of the Gnutella system

8. Structured P2P systems: node attachment / detachment; identifier space, Distributed Hash Table (DHT), maintaining information about the processing state, reduction of delays, resistance to failures on the example of Pastry, Chord, Tapestry, CAN, Kademlia systems

9. Bittorrent protocol and its extensions: PEX, Mulittracker, DHT and magnet links

10. Architecture of Big Data systems: big data acquisition, big data storage, batch processing / data stream processing, big data analysis; algorithms distributed in the processing of big data; examples of the use of big data tools in existing systems (e.g. Google, Facebook, Yahoo, LinkedIn, Cloudera, Microsoft)

11. Blockchain technology - basic idea, smart contracts (examples: Ethereum and IOTA platform)

The laboratory classes cover issues related to the processing of large amounts of data in the Casandra system:

1. Installation, configuration, programming interface, data types, basic operations available

2. Practical use of the system: data modeling; sorting, filtering, table options, indexes, collections, counters, lightweight transactions and other advanced features, paging results. Consistency levels, node failures, network partitions, and timestamp conflicts.

Teaching methods

1. Lectures: multimedia presentation, illustrated with examples given on the blackboard.

2. Laboratory classes: a multimedia presentation illustrated with examples given on the blackboard and project.

Bibliography

Basic

1. Distributed Systems: Principles and Paradigms, A. S. Tanenbaum, M. van Steen, Prentice-Hall, Inc, 2007

2. Peer-to-peer systems and applications, R. Steinmetz, K. Wehrle, Springer, 2005

3. NoSQL distilled, P. Sadalage, M. Flower, Addison-Wesley, 2013

Additional

1. Spark in Action, Bonaći M., Zečević P., Manning, 2015

2. Large Scale Network-Centric Distributed Systems, H.Sarbazi-Azad, A.Y.Zomaya, Wiley-IEEE Computer Society Press, 2013

3. M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, M. Van Steen: Gossip-based peer sampling. ACM Trans. Comput. Syst 25(3) 2007

4. Nitin Sawant, Himanshu Shah, Big data application architecture Q&A, Springer, 2013

5. J.Berman, Principles of Big Data: Preparing, Sharing, and Analyzing Complex Information, Morgan-Kaufman, 2013

Breakdown of average student's workload

	Hours	ECTS
Total workload	124	5,0
Classes requiring direct contact with the teacher	64	3,0
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	60	2,0

[1] delete or add other activities as appropriate