SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	55.2

2. Data about the subject

2.1 Subject name	***Parallel and Distributed Computing***				
2.2 Course responsible/lecturer	Conf.dr.ing. Anca Hangan – Anca.Hangan@cs.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	Conf.dr.ing. Anca Hangan – Anca.Hangan@cs.utcluj.ro				
2.4 Year of study	IV	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	E
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DOp

3. Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminars		Laboratory	2	Project	
3.2 Number of hours per semester	56	of which:	Course	28	Seminars		Laboratory	28	Project	
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										23
(b) Supplementary study in the library, online and in the field										14
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										24
(d) Tutoring										4
(e) Exams and tests										4
(f) Other activities:										

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	69
3.5 Total hours per semester (3.2+3.4)	125
3.6 Number of credit points	5

4. Pre-requisites (where appropriate)

4.1 Curriculum	Fundamental Algorithms, Fundamental programming techniques, Operating Systems, Structure of Computer Systems
4.2 Competence	C/C++ programming

5. Requirements (where appropriate)

5.1. For the course	Whiteboard, projector, computer
5.2. For the applications	Computers, Condor middleware, MPI library, C/C++ programming development environment

6. Specific competence

6.1 Professional competences	**C4** Improving the performances of the hardware, software and communication systems (2 credits) **C4.1** Identifying and describing the defining elements of the performances of the hardware, software and communication systems **C4.2** Explaining the interaction of the factors that determine the performances of hardware, software and communication systems

	C4.3 Applying fundamental methods and principles for increasing performance of hardware, software and communication systems **C4.4** Choosing criteria and methods for performance evaluation of hardware, software and communication systems **C4.5** Developing professional solutions for hardware, software and communication systems based on performance optimization **C5** Designing, managing the lifetime cycle, integrating and ensuring the integrity of hardware, software and communication systems (3 credits) **C5.1** Specifying the relevant criteria regarding the lifetime cycle, quality, security and computing system's interaction with the environment and human operator **C5.2** Using interdisciplinary knowledge for adapting the computing system to the specifc requirements of the application field **C5.3** Using fundamental principles and methods for security, reliability and usability assurance of computing systems **C5.4** Adequate utilization of quality, safety and security standards in information processing **C5.5** Creating a project including the problem's identification and analysis, its design and development, also proving an understanding of the basic quality requirements
6.2 Cross competences	N/A

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	1. Students become aware of differences and similarities between parallel and distributed computing so the students understand the boundaries of both domains. 2. Students become familiar with the principles of designing parallel programs. 3. Students become familiar with the main classes of distributed algorithms.
7.2 Specific objectives	Parallel algorithms performance and scalability. Parallel algorithms design. Distributed algorithms: time synchronization, distributed mutual exclusion, causal ordering, leader election.

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Introduction: goal, administrative issues, parallel vs distributed computing.	2	Interactive lectures using PPT presentations and exercises carried on on-site or on Microsoft Teams.	
Parallel computing basics: computer architectures and programming models.	2		
Parallel algorithm design: parallelization process, data dependency.	2		
Parallel algorithm design: case study - ocean simulation.	2		
Parallel algorithm design: decomposition and mapping techniques.	2		
Improving the performance of parallel programs: load balancing issues.	2		
Improving the performance of parallel programs: serialization and communication issues.	2		
Workload-driven evaluation of parallel systems.	2		
Cache coherence in symmetric multiprocessors.	2		
Parallel computing on distributed resources: Grid computing vs Hadoop.	2		
Time: physical clocks synchronization (Cristian algorithm, Berkeley algorithm, Network Time Protocol), logical clocks (Scalar time, Vector time, efficient implementation of vector clocks - Singhal-Kshemkalyani).	2		
Causal ordering: problem definition, Birman-Schiper-Stephenson,	2		

Schiper-Eggli-Sandoz.			
Leader election: problem definition, general networks (FloodMax, OptFloodMax), synchronous / asynchronous ring (LeLann, Chang-Roberts, Hirschberg-Sinclair).	2		
Leader election: synchronous / asynchronous ring (Franklin, Peterson), anonymous ring (Itai-Rodeh).	2		

Bibliography

1. *Parallel and Distributed Computing - Lecture notes* – A. Hangan ,http://users.utcluj.ro/~ancapop/pdc.html
2. *Introduction to Distributed Systems* -Concepts and design. George Coulouris, Jean Dollimore and Tim Kindberg, Prentice Hall, ISBN 0201-619-180, 2005 si editia revizuită 2008
3. *Distributed computing : principles, algorithms and systems,* M. Singhal, A Kshemkalyani,Cambridge Univesrity Press 0521876346 , 2008

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Introduction to grid computing	2	Problem based approach.	
Job execution in Condor (Part 1)	2		
Job execution in Condor (Part 2)	2		
Workflows in Condor. Assignment 1.	2		
Laboratory test 1. Introduction to MPI.	2		
Point-to-point communication in MPI	2		
Collective communication in MPI. Assignment 2.	2		
Advanced collective communication and groups in MPI.	2		
Implementing matrix multiplication using Cannon's algorithm (Part 1). Assignment 3.	2		
Implementing matrix multiplication using Cannon's algorithm (Part 2)	2		
Performance assessment of parallel programs. Shared memory model. Assigment 4.	2		
Performance assessment of parallel programs. Message passing model.	2		
Assignment 4 evaluation of individual results and group discussion.	2		
Laboratory Test 2.	2		

Bibliography

1. *Anca Hangan, Anca Rarau, Catalin Sipos, "Parallel and Distributed Computing", 2009, UTPRESS, ISBN: 978-973-662-484-1*
2. *Introduction to Parallel Computing*, V.Kumar, A. Grama, A. Gupta, G. Karypis, Benjamin-Cummings,ISBN 0-201-6486.
3. *Programming on parallel machines* - GPU, multicore and clusters,N. Mathloff, Universityof California Davis, 2016 , http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf

** Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.*

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

As Cluj software workforce market gets more sophisticated, having solid knowledge of how to develop parallel programs and mastering the distributed computing are qualities that software companies look for.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Formal assessment to test theoretical knowledge and problem solving skills. Attendance and activity.	Assignments, written exam (if activities are carried on on-site), online test and oral examination (if activities are carried on online).	60%
Seminar			
Laboratory	Formal assessment to test practical skills for designing parallel and distributed	Assignments and tests (using Moodle).	40%

	solutions and implementation . Attendance and activity.		
Project			

Minimum standard of performance:
Design and implementation of parallel/distributed solutions using the theoretical models and tools (MPI, Condor grid middleware).
Pre-requisite for written exam: 6 mandatory lecture attendances.
Grade calculus: 40% laboratory + 10%course assignments+50% final exam
Conditions for participating in the final exam: Laboratory ≥ 5
Conditions for promotion: grade ≥ 5

Date of filling in:	**Titulari**	**Titlu Prenume NUME**	**Semnătura**
	Course	Conf.dr.ing. Anca Hangan	
	Applications	Conf.dr.ing. Anca Hangan	

Date of approval in the department	Head of department Prof.dr.ing. Rodica Potolea
Date of approval in the Faculty Council	Dean Prof.dr.ing. Liviu Miclea