SYLLABUS

1. Information on the study programme

1.1. Higher education institution	West University of Timisoara
1.2. Faculty	Mathematics and Computer Science
1.3. Department	Computer Science
1.4. Study program field	Computer Science
1.5. Study cycle	postgraduate
1.6. Study programme	Artificial Intelligence and Distributed Computing

2. Information on the course

2.1. Course title	Parallel computing						
2.2. Lecture instructor	Dana Petcu						
2.3. Seminar / laboratory instructor	Dana Petcu						
2.4. Study year	1	2.5. Semester	2	2.6. Examination type	E	2.7. Course type	M

3. Estimated study time (number of hours per semester)

3.1. Attendance hours per week	3	out of which: 3.2 lecture	2	3.3. seminar / laboratory	1
3.4. Attendance hours per semester	42	out of which: 3.5 lecture	28	3.6. seminar / laboratory	14

Distribution of the allocated amount of time*	**hours**
Study of literature, course handbook and personal notes	48
Supplementary documentation at library or using electronic repositories	8
Preparing for laboratories, homework, reports etc.	48
Exams	6
Tutoring	8
Other activities…	0

3.7. Total number of hours of individual study	118
3.8. Total number of hours per semester	160
3.9. Number of credits (ECTS)	6

4. Prerequisites (if it is the case)

4.1. curriculum	Computer networks, Computer architecture
4.2. competences	C programmnibg

5. Requirements (if it is the case)

5.1. for the lecture	Online, Google Meet, check the digital materials at http://staff.fmi.uvt.ro/~dana.petcu/Calcul.htm
5.2. for the seminar / laboratory	Online, Google Meet, check the digital materials at http://staff.fmi.uvt.ro/~dana.petcu/Calcul.htm

6. Course objectives

Knowledge	To be familiar with the design, description and implementation of the applications that are using parallel computing
Abilities	• Capacity to identify, design and describe a parallel computing system • Capacity to implement an application that uses parallel computing • Capacity to use parallel computing systems
Responsability and autonomy	• Capacity to comunicate knowledge related to parallel computing used in different activities domains

7. Content

Lecture	Teaching methods	Remarks, details
Lecture 1. (2h) Introduction: Parallel computers, why parallel computing, application examples, short history, to port or not to port. Performance: overhead, performance metrics for parallel systems	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-1.pdf
Lecture 2. (2h) Performance Metrics for Parallel Programs: analytic modeling, execution time, overhead, speedup, efficiency, cost, granularity, scalability, roadblocks, asymptotic analysis	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-2.pdf
Lecture 3. (2h) Architecture: logical organization - Flynn taxonomy, SIMD, MIMD, communication; physical organization - historical context, shared memory versus distributed memory	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-3.pdf
Lecture 4. (2h) Architecture and Models: physical organization - radius-based classification, multicore, clusters, grids, trends; early models, PRAM	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-4.pdf
Lecture 5. (2h) Models: dataflow and systolic architectures, circuit model, graph model, LogP and LogGP; message-passing paradigm; levels of parallelism	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-5.pdf
Lecture 6. (2h) Implicit Parallelism - Instruction Level Parallelism. Pipeline, Vector and Superscalar Processors	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-6.pdf
Lecture 7. (2h) Cache coherence in multiprocessor systems. Interconnection Networks - classification, topologies, evaluating static and dynamic interconnection networks	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-7.pdf
Lecture 8. (2h) Communication costs, routing mechanism, mapping techniques, cost-performance tradeoffs	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-8.pdf
Lecture 9. (2h) Concurrency and Steps in Parallel Algorithm Design: concurrency in parallel programs, approaches to achieve concurrency, basic leyers of software concurrency; tasks,	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-9.pdf

processes and processors, design steps, decomposition - simple examples and classification		
Lecture 10. (2h) Decomposition and Orchestration: recursive, data, exploratory, speculative and hybrid decompositions, orchestration under the data parallel, shared-address space and message passing model	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-10.pdf
Lecture 11. (2h) Mapping Techniques for Load Balancing and Methods for Containing Interaction Overheads: mapping classification, schemes for static mapping, schemes for dynamic mapping, maximizing data locality, overlapping computations with interactions, replication, optimized collective interactions	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-11.pdf
Lecture 12. (2h) Emulations, Scheduling and Patterns: emulations among architectures, task scheduling problem, scheduling algorithms, load balancing; patterns - task decomposition, data decomposition, group tasks, order tasks, data sharing, design evaluation	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-12.pdf
Lecture 13. (2h) Models of Parallel Algorithms and Simple Parallel Algorithms: models - data parallel, task graph, work pool, master-slave, pipeline, hybrids; applying data parallel model, building-block computations; sorting networks	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-13.pdf
Lecture 14. (2h) Parallel computations in numerical analysis: linear equations, nonlinear equations, ordinary differential equations, computational fluid dynamics	Lecture, conversation, exemplify	Slides: http://staff.fmi.uvt.ro/~dana.petcu/calcul/PC-14.pdf

Recommended literature

1. Kontoghiorghes Erricos J. Handbook of Parallel Computing and Statistics, Chapman & Hall/CRC, Taylor & Francis Group, 2006
2. Mattson Timothy G., Sanders Beverly A., Massingill Berna L.Patterns for Parallel Programming, Addison-Wesley Professional, 2004
3. Dana Petcu. Parallel Numerical Algorithms. Mathematical Monographs 60 & 61, Printing House of University of Timisoara, 1996.
4. Dana Petcu. Parallelism in solving ordinary differential equations, Mathematical Monographs 64, Printing House of University of Timisoara, 1998.
5. Wittwer Tobias. An Introduction to Parallel Programming, VSSD, Netherlands, 2006
6. Zbigniew, Czech, Introduction to parallel computing, Cambridge University Press, 2016

Seminar / laboratory	**Teaching methods**	**Remarks, details**
Lab 1 (2h): OpenMP – Generalities, basic mechanisms and simple examples	Problem stating, dialogue, learn through patterns and collaboration	Textbook at http://staff.fmi.uvt.ro/~dana.petcu/calcul.htm
Lab 2 (2h): OpenMP – Matrix operations and performance studies	Problem stating, dialogue, learn through patterns and collaboration	Idem

Lab 3 (2h): OpenMP – sorting and performance studies	Problem stating, dialogue, learn through patterns and collaboration	Idem
Lab 4 (2h) OpenACC – generalities, simple examples and matrix operations	Problem stating, dialogue, learn through patterns and collaboration	Idem
Lab 5 (2h): MPI – Generalities, basic mechanisms and simple examples	Problem stating, dialogue, learn through patterns and collaboration	Idem
Lab 6 (2h): MPI – Matrix operations and performance studies	Problem stating, dialogue, learn through patterns and collaboration	Idem
Lab 7 (2h): MPI – solving linear systems and performance studies	Problem stating, dialogue, learn through patterns and collaboration	Idem

Recommended literature

[1] Karniadakis George E., Kirby Robert M. Parallel Scientific Computing in C++ and MPI, Cambridge University Press, 2003.
[2] Barbara Chapman, Gabriele Jost, Ruud van van der Pas, Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation), MIT Press, 2007
[3] John Cheng, Max Grossman, Ty MecKercher, Professional CUDA C Programming, Wiley, 2014

8. Correlations between the content of the course and the requirements of the professional field and relevant employers.

The content is consistent in structure with similar courses from other universities and covers the fundamental aspects necessary familiarity with the issue of parallel computing. Ability to identify, design, implement and analyze applications that utilize parallel calculation is essential for getting a timely response in case of scientific applications and commercial complex ones. Skills offered by this discipline are needed by an IT specialist in order to identify effective solutions for solving concrete problems, regardless of their specific activity field.

9. Evaluation

Activity	Assessment criteria	Assessment methods	Weight in the final mark
Lecture / Curs	• Knowlege about the problems associated with parallel computing and their solutions (OC)	Written exam in the exam period	50%

Seminar/ lab	• Capacity to design and programme an application that uses parallel computing (OAb)	Oral evaluation of the software project (semester homework)	50%
10.6. Minimum needed performance for passing			
• Capacity to write a simple application that uses parallel computing • Understand the basic principles of parallel computing			

Date of completion Instructor

10.09.2021

Date of approval Director of the department