SYLLABUS / FIȘA DISCIPLINEI

1. Information on the study programme / Date despre programul de studii

1.1. Institution / Instituția de învățământ superior	Universitatea de Vest din Timișoara
1.2. Faculty / Facultatea	Matematică și Informatică
1.3. Department / Departamentul	Computer Science (Informatică)
1.4. Study program field	Computer Science (Informatică)
1.5. Study cycle/ Ciclul de studii	MSc / master
1.6. Study programme / Programul de studii / calificarea*	Artificial Intelligence and Distributed Computing

2. Information on the course / Date despre disciplină

2.1. Title of the course / Denumirea disciplinei	Data Structures and Algorithms in Parallel Computing						
2.2. Teacher in charge of the course / Titularul activităților de curs	Conf. Dr. Marc Eduard Frincu						
2.3. Teacher in charge of the seminar / Titularul activităților de seminar	Conf. Dr. Marc Eduard Frincu						
2.4. Study year / Anul de studii	2	2.5. Semester / Semestrul	1	2.6. Examination type / Tipul de evaluare: C(olloquim)	C	2.7. Course type / Regimul disciplinei: E(lective)	E

3. Estimated study time (number of hours per semester) /Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Attendance hours per week / Număr de ore pe săptămână	3	out of which din care: 3.2 lecture/ curs	2	3.3. seminar/laborator	1
3.4. Attendance hours per semester / Total ore din planul de învățământ	42	out of which: 3.5 lecture / curs	28	3.6. seminar/laborator	14

Distribution of the allocated amount of time / Distribuția fondului de timp*	**hours/ ore**
Individual study /Studiu după manual, suport de curs, bibliografie și notițe	20
Supplementary documentation at library or using electronic repositories / Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate	15
Preparing for laboratories, homework, reports etc. /Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	40
Exams / Examinări	10
Tutoring / Tutorat	5

3.7. Total number of hours of individual study / Total ore studiu individual	90
3.8. Total number of hours per semester / Total ore pe semestru	132
3.9. Number of credits (ECTS)	5

/ Număr de credite	

4. Prerequisites (if it is the case) / Precondiții (acolo unde e cazul)

4.1. curriculum / de curriculum	Parallel computing, Data structures, Programming, Graph theory, Probability and statistics
4.2. skills / de competențe	C1. Knowledge of parallel programming concepts C2. Knowledge of graphs, probability and statistics theory C2. Ability to search, extract, and analyze knowledge from scientific papers

5. Requirements (if it is the case) / Condiții (acolo unde e cazul)

5.1. for the lecture / de desfășurare a cursului	Room with projector and whiteboard
5.2. for the seminar, laboratory / de desfășurare a seminarului/laboratorului	Lab with computers with access to parallel computing tools such as OpenMP, MPI, and Apache Giraph. Java and C installed as programming languages.

6. Acquired skills / Competențe specifice acumulate

Professional skills / Competențe profesionale	• Ability to design parallel algorithms based input data and problem description. Ability to optimize algorithms based on underlying hardware infrastructure.
Transversal skills / Competențe transversale	• Ability to conduct research and to prepare reports on a given topic.

7. Objectives of the course / Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1. General objective / Obiectivul general al disciplinei	To obtain knowledge on parallel algorithm design.
7.2. Specfic objectives / Obiectivele specifice	O1. To understand the link between data, algorithm, and hardware when parallel implementations are needed. O2. To understand the basic parallel algorithm design principles. O3. To understand the requirement for different parallel algorithms for different problem classes. O4. To get familiarized with various tools for building parallel algorithms.

8. Content / Conținuturi*

8.1. Lecture / Curs	**Teaching strategies / Metode de predare**	**Remarks, details / Observații**
1. Introduction. Algorithm design and programming models for parallel computing.	Discouse, conversation, teaching by example.	2 hours. Slides available at http://web.info.uvt.ro/~mfrincu/teaching.html. Bibliography as given on lecture slides.
2. Techniques for parallel	Discouse, conversation, teaching	2 hours. Slides available at

algorithm design.	by example.	http://web.info.uvt.ro/~mfrincu/teaching.html. Bibliography as given on lecture slides.
3. Parallel algorithms for graph problems.	Discouse, conversation, teaching by example.	2 hours. Slides available at http://web.info.uvt.ro/~mfrincu/teaching.html. Bibliography as given on lecture slides.
4-5. BSP model. Vertex centric model.	Discouse, conversation, teaching by example.	4 hours. Slides available at http://web.info.uvt.ro/~mfrincu/teaching.html. Bibliography as given on lecture slides.
6. BSP model. Subgraph centric model.	Discouse, conversation, teaching by example.	2 hours. Slides available at http://web.info.uvt.ro/~mfrincu/teaching.html. Bibliography as given on lecture slides.
7. Importance of graph partioning in BSP model.	Discouse, conversation, teaching by example.	2 hours. Slides available at http://web.info.uvt.ro/~mfrincu/teaching.html. Bibliography as given on lecture slides.
8. Parallel algorithms for sorting.	Discouse, conversation, teaching by example.	2 hours. Slides available at http://web.info.uvt.ro/~mfrincu/teaching.html. Bibliography as given on lecture slides.
9. Parallel algorithms for computational geometry	Discouse, conversation, teaching by example.	2 hours. Slides available at http://web.info.uvt.ro/~mfrincu/teaching.html. Bibliography as given on lecture slides.
10. Parallel algorithms for numerical computing.	Discouse, conversation, teaching by example.	2 hours. Slides available at http://web.info.uvt.ro/~mfrincu/teaching.html. Bibliography as given on lecture slides.
11-13. Discussing scientific papers on emerging topics on graph parallelelism	Conversation.	6 hours. Bibliography selected from relevant scientific papers published in the last years.
Recommended bibliography / Bibliografie 1. Algorithm Design: Parallel and Sequential, http://www.parallel-algorithms-book.com/		
8.2. Seminar, lab / Seminar,	**Teaching/learning strategies /**	**Remarks, details / Observații**

laborator	**Metode de predare/ invățare**	
1.Parallel merge-sort, parallel number sorting.	Conversation, learning through collaboration and online sources. Problem analysis.	2 hours
2. Parallel MIS, Parallel Euler tour	Conversation, learning through collaboration and online sources. Problem analysis.	2 hours
3. BSP model primer. Finding max value in a graph	Conversation, learning through collaboration and online sources. Problem analysis.	2 hours
4. BSP vertex centric examples: connected components, single source shortest path, pagerank	Conversation, learning through collaboration and online sources. Problem analysis.	2 hours
5. BSP subgraph centric examples: connected components, single source shortest path, pagerank	Conversation, learning through collaboration and online sources. Problem analysis.	2 hours
6. Sorting examples: bitonic sort, quicksort.	Conversation, learning through collaboration and online sources. Problem analysis.	2 hours
7. Parallel numerical algorithms: matrix multiplication, Fourier transform.	Conversation, learning through collaboration and online sources. Problem analysis.	2 hours

Recommended bibliography / Bibliografie

1. http://openmp.org/wp/
2. http://giraph.apache.org/
3. https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

9. Correlations between the content of the course and the requirements of the IT field / Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului

The course teaches the basics on how to consider hardware and data aspects when designing parallel algorithms. It offers students the chance to use well known as well as new tools for parallel algorithm implementation. As such it provides them with the necessary tools for building efficient parallel applications by using available technologies some in use by IT industry. Furthermore their acquired understanding will enable them to improve existing parallel code in use by various open source or proprietary software.

10. Evaluation / Evaluare*

Activity / Tip de activitate	10.1. Evaluation criteria / Criterii de evaluare**	10.2. Evaluation methods / Metode de evaluare***	10.3. Weight in the averaged mark / Pondere din nota finală
10.4. Lecture /	E1. Understanding of basic parallel	Oral evaluation.	10%

Curs	algorithm design principles.		
	E2. Ability to analyze a problem and identify key issues in parallel algorithm design.	Oral evaluation	10%
10.5. Seminar/ lab	E1. Lab assignments	Student presentation. Oral evaluation.	80%
10.6. Minimal knowledge for passing / Standard minim de performanță			
1. Students should have basic knowledge of parallel algorithm design principles 2. Students should have completed at least 2 lab asignments.			

Date/ Data completării 19.10.2016

Signature (lecture) / Semnătura titularului de curs

Signature (seminar) Semnătura titularului de seminar

Signature (director of the department)
Semnătura directorului de departament
Conf.dr. Victoria Iordan