syllabus

1. Information regarding the programme

1.1 Higher education institution	**"Babes_Bolyai" University**
1.2 Faculty	**Faculty of Mathematics and Computer science**
1.3 Department	**Department of Computer Science**
1.4 Field of study	**Informatics(Computer Science)**
1.5 Study cycle	**Master**
1.6 Study programme / Qualification	**High Performance Computing and Big Data Analytics**

2. Information regarding the discipline

2.1 Name of the discipline	**Models in Parallel Programming**						
2.2 Course coordinator	**Assoc.Prof. PhD. Niculescu Virginia**						
2.3 Seminar coordinator	**Assoc.Prof. PhD. Niculescu Virginia**						
2.4. Year of study	**1**	2.5 Semester	**2**	2.6. Type of evaluation	**E.**	2.7 Type of discipline	**Compulsory**
2.8 Discipline code	**MME8031**						

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	4	Of which: 3.2 course	2	3.3 seminar/laboratory	1 sem. +1pr.
3.4 Total hours in the curriculum	56	Of which: 3.5 course	28	3.6 seminar/laboratory	28
Time allotment:	hours				
Learning using manual, course support, bibliography, course notes	36				
Additional documentation (in libraries, on electronic platforms, field documentation)	40				
Preparation for seminars/labs, homework, papers, portfolios and essays	42				
Tutorship	14				
Evaluations	12				
Other activities:	-				
3.7 Total individual study hours	144				
3.8 Total hours per semester	200				
3.9 Number of ECTS credits	8				

4. Prerequisites (if necessary)

4.1. curriculum	Algorithms construction and evaluation, Data Structures, Object-oriented and functional programming, C++
4.2. competencies	Programming skills and basic abilities for dealing with abstractions

5. Conditions (if necessary)

5.1. for the course	Projector, blackboard
5.2. for the seminar	Projector, blackboard, computers(laptops)

6. Specific competencies acquired

Professional competencies	Knowledge, understanding of the fundamental concepts of parallel programming. Ability to apply abstraction and to reason about their efiiciency Knowledge, understanding of the theoretical foundations of parallel algorithms construction. Ability to implement parallel programs and to test them on different parallel architectures
Transversal competencies	Ability to solve problems using parallel programming. Ability to do research work in the domain of the parallel programming by studying a particular model of parallel computation. Ability to work independently or in teams for solving effiently diferent comptational problems

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• For each student has to acquire an acceptable level of knowledge and understanding of the subject, that (s)he is capable of stating the knowledge in a coherent form, that (s)he has correct habits of analysis, design, and implementation using different models of parallel computation.
7.2 Specific objective of the discipline	• To present the basic paradigms of parallel programming . • To offer different models of parallel programs development and understanding their necessity and their advantages. • To create the ability to correctly develop parallel algorithms from different domains such as: linear algebra, numerical analysis, graph theory, searching and sorting algorithms...

8. Content

8.1 **Course**	Teaching methods	Remarks
1. General Introduction to parallel programming: - reasons for using parallel programming; - problems and difficulties in parallel programming; - the necessity of using models Parallel Computer Architectures - *Taxonomies*	Exposure: description, explanation, examples, discussion of case studies	
2. Types of parallelism • Implicit parallelism	Exposure: description, explanation, examples,	

• Explicit Parallelism - Data-parallel model - Message-passing model - Shared-variable model Programming examples: Multithreading(C++ vs. Java); MPI ; OpenMP	discussion of case studies	
3. Parallel architectures- Interconnection networks	Exposure: description, explanation, examples, discussion of case studies	
4. Task Dependency Graph, Task Interaction Graph, Degree of Concurrency, Granularity, PCAM method	Exposure: description, explanation, examples, discussion of case studies	
5. Analytical Modeling of Parallel Systems Scalability	Exposure: description, explanation, examples, discussion of case studies	
6.PRAM models	Exposure: description, explanation, examples, discussion of case studies	
7. Computational networks Brent Theorem	Exposure: description, explanation, examples, discussion of case studies	
8. Bulk Synchronous Parallel programming BSP – LogP model	Exposure: description, explanation, examples, discussion of case studies	
9. Parallel programming patterns - Master-slaves - Task-Farm - Work-Pool - Divide&Conquer - Pipeline	Exposure: description, explanation, examples, discussion of case studies	
10. Algorithmic Skeletons	Exposure: description, explanation, examples, discussion of case studies	
11. Functional parallel programming *Bird-Meertens Formalism (BMF)***.** - List Homomorphisms - Categorical Data Types Map-Reduce Model	Exposure: description, explanation, examples, discussion of case studies	
12. PARES – A Model for Parallel Recursive Programs -Special data structures of parallel recursion: PowerLists, ParLists, PLists	Exposure: description, explanation, examples, discussion of case studies	
13. Scala - > multiparadigm parallel programming	Exposure: description, explanation, examples, discussion of case studies	
14. General comparison and classification of the	Exposure: description, explanation, examples,	

parallel computation models (PCM). o Requirements for PCM o Classification: - implicit parallelism - implicit decomposition - explicit decomposition - explicit mapping - explicit communication - everything explicit Main Categories of Models Classification/Comparison of the models for parallel computation.	discussion of case studies	

Bibliography

1. Michael McCool, Arch Robinson, James Reinders, Structured Parallel Programming: Patterns for Efficient Computation," Morgan Kaufmann,, 2012 .
2. A Pattern Language for Parallel Programming. Berna L. Massingill, Timothy G. Mattson, and Beverly A. Sanders, Addison Wesley Software Patterns Series, 2004.
3. Grama, A. Gupta, G. Karypis, V. Kumar. Introduction to Parallel Computing, Addison Wesley, 2003.
4. Ian Foster. Designing and Building Parallel Programs, Addison-Wesley 1995.
5. K.M. Chandy, J. Misra, Parallel Program Design: A Foundation, Addison-Wesley, 1988.
6. M J QUINN. Parallel Programming in C with MPI and OpenMP, McGraw Hill, 2004.
7. J. Misra. PowerList: A structure for parallel recursion. ACM Transactions on Programming Languages and Systems, 16(6):1737-1767, November 1994.
8. Selim Akl, Parallel Computation: Models and Methods, Prentice Hall, 1997
9. B. WILKINSON, C.M. ALLEN. Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 1999.
10. C. A. R. Hoare, Communicating Sequential Processes. June 21, Prentice Hall International, 2004.
11. V. Niculescu. Calcul Paralel. Proiectare si dezvoltare formala a programelor paralele. Presa Univ. Clujana, 2006.
12. V. Niculescu. *PARES - A Model for Parallel Recursive Programs*, Romanian Journal of Information Science and Technology (ROMJIST), Ed. Academiei Romane, Volume 14(2011), No. 2, pp. 159–182, 2011
13. A.W. Roscoe, The Theory and Practice of Concurrency. Prentice-Hall 1998.
14. D. Skillicorn. Foundations of Parallel Programming, Cambridge International Series on Parallel Computations, 1994
15. D.B. Skillicorn, D. Talia. Models and Languages for Parallel Computation. ACM Computer Surveys, 30(2) pg.123-136, June 1998.

8.2 Seminar	Teaching methods	Remarks
1. Simple examples of parallel programs.	Explanation, dialogue, case studies	The seminar is structured as 2 hours classes every second week
2. OpenMP examples	Dialogue, debate, case studies, examples, proofs	
3. MPI examples	Dialogue, debate, case studies, examples, proofs	
4. Performance analysis	Dialogue, debate, explanation, examples	
5. Student presentations	Dialogue, debate, explanation, examples	
6. Student presentations	Dialogue, debate,	

	explanation, examples	
7. Student presentations	Dialogue, debate, explanation, examples	

Bibliography

1. C. A. R. Hoare. Communicating Sequential Processes was first published in by Prentice Hall International, 2004(revised). [http://www.usingcsp.com/cspbook.pdf]
2. D. Grigoras. Calculul Paralel. De la sisteme la programarea aplicatiilor. Computer Libris Agora, 2000.
3. Rob H. Bisseling. Parallel Scientific Computation: A Structured Approach using BSP and MPI, Oxford University Press, March 2004. 324 pages.
4. Roscoe, A. W. (Revised 2005), The Theory and Practice of Concurrency, Prentice Hall, ISBN 0-13-674409-5
5. Parallel Programming Model Watch [http://view.eecs.berkeley.edu/wiki/Parallel_Programming_Model_Watch]
6. *** MPI Forum. https://www.mpi-forum.org
7. *** OpenMP home. https://www.openmp.org/

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies;
- The course exists in the studying program of all major universities in Romania and abroad;

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Knowledge level of the fundamental principles and paradigms of the domain	Written exam Complex project development and analysis	50%
10.5 Seminar	Research paper that presents a model of parallel computation	-presentation -discussion	30%
	Correctness and efficiency of the implementations of several problems using different models.	Practical works	20%
10.6 Minimum performance standards			
✶ At least grade 5 (from a scale of 1 to 10).			

Date | Signature of course coordinator: Assoc. Prof. Niculescu Virginia | Signature of seminar coordinator: Assoc. Prof. Niculescu Virginia

Date of approval | Signature of the head of department

Assoc. Prof. Adrian Sterca