SYLLABUS

Parallel and Distributed Systems – Models in Parallel Programming

University year 2025-2026

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	**Artificial Intelligence for Connected Industries**
1.7. Form of education	**Full time**

2. Information regarding the discipline

2.1. Name of the discipline	Parallel and Distributed Systems – Models in Parallel Programming					Discipline code	**MME8233**
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. Discipline regime	Optional

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/project	**2**
3.4. Total hours in the curriculum	56	of which: 3.5 course	28	3.6 seminar/laboratory/project	**28**
Time allotment for individual study (ID) and self-study activities (SA)					**hours**
Learning using manual, course support, bibliography, course notes (SA)					30
Additional documentation (in libraries, on electronic platforms, field documentation)					25
Preparation for seminars/labs, homework, papers, portfolios and essays					42
Tutorship					10
Evaluations					12
Other activities:					-
3.7. Total individual study hours	**119**				
3.8. Total hours per semester	**175**				
3.9. Number of ECTS credits	**7**				

4. Prerequisites (if necessary)

4.1. curriculum	Fundamentals of Programming Object oriented programming
4.2. competencies	Programming skills and basic abilities for dealing with abstractions Programming in C++

5. Conditions (if necessary)

5.1. for the course	Projector, blackboard
5.2. for the seminar /lab activities	Projector, internet access/cluster, computers(laptops)

6.1. Specific competencies acquired [1]

[1] One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

Professional/ essential competencies	• Capability of analysis and synthesis; • Understanding and working with basic concepts of data analysis and modelling; • Efficient modeling and solving real-life problems; • Assimilation of mathematical concepts and formal models to understand the methods and components of high performance systems; • Capability of developing of high performance programs based on parallel and distributed programming
Transversal competencies	• Etic and fair behavior, committment to professional deontology; • Team work capabilities; able to fulfill different roles; • Professional communication skills; concise and precise description, both oral and written, of professional results, negociation abilities;

6.2. Learning outcomes

Knowledge	• The student/graduate knows the basic paradigms of parallel programming • The student/graduate understands different models of parallel programs development, their necessity and their advantages • The student/graduate knows how to develop parallel algorithms using different models of parallel computation (such as algorithms from linear algebra, numerical analysis, searching and sorting algorithms)
Skills	• The student/graduate acquires the main skills and abilities to work with scalable systems that allow solving large problems by dividing them in parallel sub-problems, or by dividing the input data and process it in parallel bulks • The student/graduate acquires the fundamental knowledge that allows parallelizing and solving large and complex problems on scalable systems
Responsibility and autonomy:	• The student/graduate assumes responsibility for the product of his / her work, requests feedback and uses it constructively • The student/graduate is able to provide specialized scientific advice and develop specialized materials

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

7.1 General objective of the discipline	The students have to prove that they acquired an acceptable level of knowledge and understanding of the subject, that they are capable of stating this knowledge in a coherent form, that they have correct habits of analysis, design, and implementation using different models of parallel computation.
7.2 Specific objective of the discipline	The students have to know and understand: - the basic paradigms of parallel programming. - different models of parallel programs development and understanding their necessity and their advantages. The students have to be able to - correctly develop parallel programs using different models of parallel computation, - to apply this development skills for implementations of algorithms from linear algebra, numerical analysis, graph, 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;	Exposure: description, explanation, examples, discussion of case studies	

• the necessity of using models		
2. Types of parallelism • Implicit parallelism • Explicit Parallelism • Data-parallel model • Message-passing model • Shared-variable model	Exposure: description, explanation, examples, 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, Mapping PCAM method	Exposure: description, explanation, examples, discussion of case studies	
5. PRAM models. Computational networks Brent Theorem	Exposure: description, explanation, examples, discussion of case studies	
6. Analytical Modeling of Parallel Systems Time complexity, speed-up, cost, efficiency	Exposure: description, explanation, examples, discussion of case studies	
7. Analytical Modeling of Parallel Systems Granularity and Scalability	Exposure: description, explanation, examples, discussion of case studies	
8. Parallel programming patterns - Master-slaves/- Task-Farm/- Work-Pool - Divide &Conquer/- Pipeline	Exposure: description, explanation, examples, discussion of case studies	
9. Bulk Synchronous Parallel programming - BSP& LogP	Exposure: description, explanation, examples, discussion of case studies	
10. Functional parallel programming *Bird-Meertens Formalism (BMF)*. • List Homomorphisms/Categorical Data Types Map-Reduce Model	Exposure: description, explanation, examples, discussion of case studies	
11. CUDA	Exposure: description, explanation, examples, discussion of case studies	
12. Actor Model	Exposure: description, explanation, examples, discussion of case studies	
13. General presentation of the parallel computation models (PCM). • Requirements for PCM • Classification: - implicit parallelism - implicit decomposition - explicit decomposition - explicit mapping -explicit communication - everything explicit Main Categories of Models	Exposure: description, explanation, examples, discussion of case studies	
14.	Exposure: description, explanation,	

Scientific reports analysis	examples, discussion of case studies	

Bibliography

- Michael McCool, Arch Robinson, James Reinders, Structured Parallel Programming: Patterns for Efficient Computation," Morgan Kaufmann,, 2012 .
- A Pattern Language for Parallel Programming. Berna L. Massingill, Timothy G. Mattson, and Beverly A. Sanders,Addison Wesley Software Patterns Series, 2004.
- Grama, A. Gupta, G. Karypis, V. Kumar. Introduction to Parallel Computing, Addison Wesley, 2003.
- Ian Foster. Designing and Building Parallel Programs, Addison-Wesley 1995.
- K.M. Chandy, J. Misra, Parallel Program Design: A Foundation, Addison-Wesley, 1988.
- M J QUINN. Parallel Programming in C with MPI and OpenMP, McGraw Hill, 2004.
- B. WILKINSON, C.M. ALLEN. Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 1999.
- C. A. R. Hoare, Communicating Sequential Processes. June 21, Prentice Hall International, 2004.
- V. Niculescu. Calcul Paralel. Proiectare si dezvoltare formala a programelor paralele. Presa Univ. Clujana, 2006.
- D.B. Skillicorn, D. Talia. Models and Languages for Parallel Computation. ACM Computer Surveys, 30(2) pg.123-136, June 1998.
- Gibbons, Jeremy (2020). Troy Astarte (ed.). The School of Squiggol: A History of the Bird-Meertens Formalism (PDF). Formal Methods (Workshop on History of Formal Methods). LNCS. Vol. 12233. Springer. doi:10.1007/978-3-030-54997-8_2.
- Horacio González-Vélez and Mario Leyton. 2010. A survey of algorithmic skeleton frameworks: high-level structured parallel programming enablers. Softw. Pract. Exper. 40, 12 (November 2010), 1135–1160.
- Carl Hewitt. *Viewing Control Structures as Patterns of Passing Messages* Journal of Artificial Intelligence. June 1977.
- Hewitt, Carl; Bishop, Peter; Steiger, Richard (1973). *"A Universal Modular Actor Formalism for Artificial Intelligence"*. IJCAI'73: Proceedings of the 3rd *International Joint Conference on Artificial Intelligence*. pp. *235*–245.

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

Bibliography

***, Tutorial C++ [https://en.cppreference.com/]
***, OpenMP[http://openmp.org/wp/]
***, MPI[http://www.mpi-forum.org/]
***, CUDA, [https://docs.nvidia.com/cuda/cuda-c-programming-guide/]

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 programs oriented on High Performance Computing of all major universities abroad.
- The companies working in the domain consider that the course content is important for acquiring advanced programming skills.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	- know the basic principles and paradigms of the domain;	Teoretical exam alt. complex project	40%

10.5 Seminar/laboratory	- a research paper that presents a model of parallel computation	-presentation -discussion	30%
	-practical works	-presentation -discussion -evaluation	30%
10.6 Minimum standard of performance			
• At least grade 5 (from a scale of 1 to 10). • At least grade 5 (from a scale of 1 to 10 at teoretical exam.			

11. Labels ODD (Sustainable Development Goals)[2]

Not applicable.

Date:
15.04.2025

Signature of course coordinator

Signature of seminar coordinator

Date of approval:
...

Signature of the head of department

Assoc.prof.phd. Adrian STERCA

[2] Keep only the labels that, according to the *Procedure for applying ODD labels in the academic process*, suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable.*".