

FIȘA DISCIPLINEI

Modele in programarea paralela

Anul universitar 2025-2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babes-Bolyai Cluj-Napoca
1.2. Facultatea	Facultatea de Matematica si Informatica
1.3. Departamentul	Departamentul de Informatica
1.4. Domeniul de studii	Informatica
1.5. Ciclul de studii	Master
1.6. Programul de studii / Calificarea	Calcul de inalta performanta si analiza volumelor mai de date
1.7. Forma de învățământ	Cu frecventa

2. Date despre disciplină

2.1. Denumirea disciplinei	Modele in programarea paralela			Codul disciplinei	MME8031		
2.2. Titularul activităților de curs	Conf. univ.dr. Niculescu Virginia						
2.3. Titularul activităților de seminar	Conf. univ.dr. Niculescu Virginia						
2.4. Anul de studiu	1	2.5. Semestrul	2	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Obligativiu

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Număr de ore pe săptămână	4	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	2
3.4. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	28
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					30
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					44
Tutoriat (consiliere profesională)					20
Examinări					10
Alte activități [comunicare bidirecțională cu titularul de disciplină / tutorele]					10
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				144	
3.8. Total ore pe semestru				200	
3.9. Numărul de credite				8	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Fundamanetele programarii Programare orienta obiect
4.2. de competențe	Abilitati de programare si de abstractizare Programare C++

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sala cu proiector, tabla
5.2. de desfășurare a seminarului/ laboratorului	Sala cu proiector, acces internet/cluster

6.1. Competențele specifice acumulate¹

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

Competențe profesionale/esențiale	<ul style="list-style-type: none"> • capacitatea de analiză și sinteză; • înțelegerea și operarea cu conceptele de bază ale calculului de înaltă performanță și a tehnicilor de analiză a datelor; • modelare și rezolvarea eficientă de probleme din lumea reală. • însușirea conceptelor matematice și modelelor formale care să faciliteze înțelegerea metodelor și componentelor sistemelor de calcul de înaltă performanță; • capacitatea de realizare de programe performante folosind programarea paralelă și distribuită
Competențe transversale	<ul style="list-style-type: none"> • comportarea onorabilă, etică, respectarea deontologiei profesionale; • abilități de muncă în echipă, cu preluarea diferitelor roluri de execuție și conducere pentru realizarea unor proiecte; • abilități de comunicare profesională: descrierea clară, concisă, verbală și în scris, a rezultatelor profesionale, abilități de negociere;

6.2. Rezultatele învățării

Cunoștințe	<ul style="list-style-type: none"> • Studentul/absolventul cunoaște paradigmele de bază ale programării paralele • Studentul/absolventul înțelege diferite modele de dezvoltare a programelor paralele, necesitatea utilizării și avantajele acestora • Studentul/absolventul știe să dezvolte algoritmi paraleli folosind diferite modele de calcul paralel (precum algoritmi din algebra liniară, analiza numerică, algoritmi de căutare și sortare)
Aptitudini	<ul style="list-style-type: none"> • Studentul/absolventul dobândește principalele abilități de a lucra cu sisteme scalabile care permit rezolvarea unor probleme mari prin împărțirea lor în sub-probleme paralele, sau prin împărțirea datelor de intrare și procesarea lor în blocuri paralele • Studentul/absolventul dobândește cunoștințele fundamentale care permit paralelizarea și rezolvarea unor probleme mari și complexe pe sisteme scalabile
Responsabilități și autonomie	<ul style="list-style-type: none"> • Studentul/absolventul își asumă responsabilitatea pentru produsul muncii sale, solicită feedback și îl utilizează constructiv • Studentul/absolventul este în măsură să ofere consultanță științifică de specialitate și să elaboreze materiale de specialitate

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	Studentii trebuie să dobândească cunoștințele fundamentale din domeniul programării paralele și să fie capabili să realizeze analiza, proiectare și implementare de aplicații folosind diferite modele de calcul paralel.
7.2 Obiectivele specifice	<p>Studentii trebuie să cunoască și să înțeleagă:</p> <ul style="list-style-type: none"> - Paradigmele de bază ale programării paralele - Diferite modele folosite în dezvoltarea aplicațiilor paralele și să înțeleagă necesitatea folosirii lor și avantajele fiecăruia. <p>Studentii trebuie să fie capabili să:</p> <ul style="list-style-type: none"> - Dezvolte corect programe paralele folosind diferite modele de calcul paralel. - Să folosească aceste abilități de programare pentru implementarea de algoritmi din algebra liniară, analiza numerică, teoria grafurilor, căutare și sortare.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere în programarea paralelă: • Necesitatea folosirii programării paralele	Expuneri: concepte, exemple, studii de caz	

<ul style="list-style-type: none"> • Provocari ale programarii paralele • Neceitatea folosirii modelelor 		
2. Tipuri de paralelism <ul style="list-style-type: none"> • Implicit parallelism • Explicit Parallelism <ul style="list-style-type: none"> • Data-parallel model • Message-passing model • Shared-variable model 	Expuneri: concepte, exemple, studii de caz	
3 Arhitecturi Paralele. Retele de interconectare.	Expuneri: concepte, exemple, studii de caz	
4. Task Dependency Graph, Task Interaction Graph, Degree of Concurrency, Mapare PCAM	Expuneri: concepte, exemple, studii de caz	
5. Modele PRAM. Retele computationale Teorema lui Brent	Expuneri: concepte, exemple, studii de caz	
6. Modelarea analitica a sistemelor paralele: Time complexity, speed-up, cost, efficiency	Expuneri: concepte, exemple, studii de caz	
7. Modelarea analitica a sistemelor paralele: Granularitate si scalabilitate	Expuneri: concepte, exemple, studii de caz	
8. Sabloane de progrmare [paralela : <ul style="list-style-type: none"> - Master-slaves/- Task-Farm/- Work-Pool - Divide & Conquer/- Pipeline 	Expuneri: concepte, exemple, studii de caz	
9. Modele <ul style="list-style-type: none"> - BSP & LogP 	Expuneri: concepte, exemple, studii de caz	
10. programare paralela functionala Bird-Meertens Formalism (BMF). <ul style="list-style-type: none"> • List Homomorphisms/Categorical Data Types Map-Reduce Model	Expuneri: concepte, exemple, studii de caz	
11. CUDA	Expuneri: concepte, exemple, studii de caz	
12. Actor Model	Expuneri: concepte, exemple, studii de caz	
13. PCM -parallel computation models <ul style="list-style-type: none"> • Cerinte pentru PCM • Clasificare PCM: <ul style="list-style-type: none"> - implicit parallelism - implicit decomposition - explicit decomposition - explicit mapping -explicit communication - everything explicit Exemplificari	Expuneri: concepte, exemple, studii de caz	
14. Analiza rapoartelor stiintifice	Expuneri: concepte, exemple, studii de caz	

Bibliografie

- 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](https://doi.org/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.Exemple folosind multithreading.	Discutii, exemplificare, evaluare	Seminarul este programat la doua saptamani cate 2 ore
2. Tehniques used in parallel programs construction.	Discutii, exemplificare, evaluare	
3. Exemple OpenMP	Discutii, exemplificare, evaluare examples, proofs	
4. Exemple MPI	Discutii, exemplificare, evaluare	
5 Exemple CUDA	Discutii, exemplificare, evaluare	
6. Analiza de performanta	Discutii, exemplificare, evaluare	
7. Prezentari studenti	Discutii, exemplificare, evaluare	
Bibliografie ***, 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. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Cursul respecta recomandările curriculei ACM și IEEE pentru studiile de informatică.
- Cursul apare în planurile de învățământ a celor mai importante universități și străinătate.
- Firmele de soft consideră conținutul cursului important pentru dobândirea unor abilități avansate de programare.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	-cunosterea principiilor de baza si a paradigmelor/modelelor folosite in programarea paralela	Examen teoretic	40%
10.5 Seminar/laborator	-raport stiintific care prezinta un model de calcul paralel	-prezentare -analiza	30%
	-lucrari practice	-prezentare -analiza -testare	30%
10.6 Standard minim de performanță			
○ Media finala trebuie sa fie cel putin 5 (pe o scara de la 1 la 10).			

- Promovarea este condiționată de nota minimă 5 la examenul teoretic.

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

Data completării:

Semnătura titularului de curs

Semnătura titularului de seminar

...



Data avizării în departament:

Semnătura directorului de departament

...

Conf.dr. Adrian STERCA

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "*Nu se aplică.*".