

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematica și Informatică
1.3 Departamentul	Departamentul de matematică
1.4 Domeniul de studii	Informatică
1.5 Ciclu de studii	Licență
1.6 Programul de studiu / Calificarea	Informatică

2. Date despre disciplină

2.1 Denumirea disciplinei	Paradigme și tehnici ale programării paralele						
2.2 Titularul activităților de curs	Conf. Dr. Niculescu Virginia						
2.3 Titularul activităților de seminar	Conf. Dr. Niculescu Virginia						
2.4 Anul de studiu	3	2.5 Semestrul	6	2.6. Tipul de evaluare	colocviu	2.7 Regimul disciplinei	Optional

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

3.1 Număr de ore pe săptămână	3	Din care: 3.2 curs	2	3.3 seminar/laborator	1
3.4 Total ore din planul de învățământ	42	Din care: 3.5 curs	28	3.6 seminar/laborator	14
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					23
Tutoriat					7
Examinări					20
Alte activități:					-
3.7 Total ore studiu individual		83			
3.8 Total ore pe semestru		125			
3.9 Numărul de credite		5			

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Fundamentele programării, Structuri de date
4.2 de competențe	Abilități de implementare programe.

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	proiector
5.2 De desfășurare a seminarului/laboratorului	laborator cu stații de lucru

6. Competențele specifice acumulate

Competențe profesionale	Fiecare student trebuie să dovedească faptul că a dobândit un nivel de cunoștințe și înțelegere a subiectului și că este capabil (a) să exprime aceste cunoștințe; și, de asemenea, că poate folosi aceste cunoștințe în rezolvarea unor probleme prin implementarea soluțiilor folosind programarea paralelă.
--------------------------------	--

Competențe transversale	<ul style="list-style-type: none"> - Abilitatea de a aplica tehnici specifice programarii paralele in rezolvarea problemelor. - Abilitatea de a evalua cresterea de performanta obtinuta prin folosirea paralelizarii. - Abilitatea de a implementa programe paralele.
--------------------------------	---

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> - Cunoasterea, intelegerea conceptelor de baza ale programarii paralele. - Abilitatea de a lucra independent sau in echipa pentru a rezolva probleme intr-un context de programare paralela. - Cunoasterea, intelegerea paradigmelor si tehnicilor de baza ale programarii paralele.
7.2 Obiectivele specifice	<ul style="list-style-type: none"> - Obținerea de cunostinte de baza despre programarea paralela. - Cunoasterea si folosirea celor mai importante paradigme/tehnici de dezvoltare a programelor paralele. - Implementarea programe paralele folosind thread-uri Java(sau C#), sau Parallel Python sau OpenMP. - Realizarea unui Proiect(aplicatie) folosind pentru implementare una dintre variantele specificate anterior

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere generala in programarea paralela: - necesitatea folosirii paralelismului; - probleme si dificultati in programarea paralela; Arhitecturi paralele - <i>Taxonomii</i>	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
2. Tipuri de paralelism • Paralelism implicit • Paralelism explicit - Modelul Data-parallel - Modelul Message-passing - Modelul Shared-variable	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
3. PRAM (Parallel Random Access Machine) Rețele computationale Evaluarea performantei programelor paralele: complexitate-timp, complexitate-procesor, acceleratie, eficienta, cost; evaluare scalabilitate.	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
4. Faze in dezvoltarea programelor paralele (PCAM) - Partitionare, Comunicare, Aglomerare, Mapare Partitionare->Descompunere - functionala (task decomposition) - a domeniului de date(geometrica) distributii de date	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
5. Concurenta – concepte Race-conditions, critical section, mutual exclusion, deadlock -Sincronizare: monitoare, semafoare,	Expunere, descriere, explicatie, exemple , discutii ale unor studii de caz.	
6. Programare paralela in medii cu memorie partajata. <i>Java Threads,</i>	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
7.	Expunere, descriere,	

Programare paralela in medii cu memorie partajata. <i>OpenMP</i>	explicatie, exemple, discutii ale unor studii de caz.	
8. Programare paralela in medii cu memorie distribuita. <i>MPI</i>	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
9. Modele: - <i>Master-slaves</i> - <i>Task-Farm</i> - <i>Work-Pool</i>	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
10. Modele: - Divide &Impera Tehnici utilizate in constructia algoritmilor paraleli: - Tehnica arborelui binar - Tehnica dublarii recursive	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
11. Modele: - Pipeline - Modelul Map-Reduce	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
12. Exemple de algoritmi paraleli: - algoritmi numerici: inmultire de matrici, rezolvare sistem linear - algoritmi de sortare - algoritmi de cautare	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	

<http://www.cs.ubbcluj.ro/~vniculescu/didactic/>

Bibliografie

1. Ian Foster. Designing and Building Parallel Programs, Addison-Wesley 1995.
2. Grama, A. Gupta, G. Karypis, V. Kumar. Introduction to Parallel Computing, Addison Wesley, 2003.
3. D. Grigoras. Calculul Paralel. De la sisteme la programarea aplicatiilor. Computer Libris Agora, 2000.
4. V. Niculescu. Calcul Paralel. Proiectare si dezvoltare formala a programelor paralele. Presa Univ. Clujana, 2006.
5. D.B. Skillicorn, D. Talia. Models and Languages for Parallel Computation. ACM Computer Surveys, 30(2) pg.123-136, June 1998.
6. B. Wilkinson, M. Allen, Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 2002
7. E.F. Van de Velde. Concurrent Scientific Computing. Spring-Verlag, New-York Inc. 1994.

8.2 Seminar / laborator	Metode de predare	Observații
1. Exemple simple de algoritmi paraleli. Evaluare empirica a performantei	Dialog, dezbateri, explicatie,exemple.	
2. Tehnici folosite in constructia programelor paralele.	Dialog, dezbateri, explicatie,exemple.	
3. Exemple folosind Java Threads	Dialog, dezbateri, explicatie,exemple.	
4. Exemple OpenMP	Dialog, dezbateri, explicatie,exemple.	
5. Exemple MPI	Dialog, dezbateri, explicatie,exemple.	
6. Prezentari ale proiectelor studentilor.	Dialog, dezbateri,	

	explicatie,exemple.	
7. Prezentari ale proiectelor studentilor	Dialog, dezbateri, explicatie,exemple.	
Bibliografie		
1. Parallel Python, [http://www.parallelpython.com/]		
2. Java Programming Language [www.oracle.com/us/technologies/java/overview]		
3. OpenMP[http://openmp.org/wp/]		
4. MPI[http://www.mpi-forum.org/]		

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

Cursul respecta Recomandarile IEEE and ACM Curricula pentru studii in Computer Science; Cursuri cu tematica similara exista in programele de studii ale majoritatii universitatilor din tara si strainatate.

Cursul definește dobândirea unor abilitati care constituie avantaje evidentiate de potentialele firme angajatoare din domeniu.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	In scopul evaluarii obtinerii competentelor specifice, fiecare student va dezvolta un proiect prin care va rezolva o problema cu specificatii date, folosind programare paralela(se va alege unul dintre limbajele discutate). Proiectul va avea o documentatie care va descrie metoda de paralelizare si evaluarea performantei.	Colocviu: Prezentarea proiectului – metoda de paralelizare Evaluarea performantei obtinute- executia aplicatiei	50%
10.5 Seminar/laborator	Implementarea unor programe paralele simple.	Evaluare executie si explicarea implementarii	50%
10.6 Standard minim de performanță			
Minim 5 la media finala.			

Data completării
1.04.2015.....

Titular de curs
...Niculescu Virginia.....

Titular de laborator
..Niculescu Virginia.....

Data avizării în departament
.....

Director de departament
.....