

## 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 Ciclul de studii	Licența
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		80			
3.8 Total ore pe semestru		150			
3.9 Numărul de credite		6			

### 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Fundamentele programării, Structuri de date
4.2 de competențe	Abilitati 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

<b>Competențe profesionale</b>	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.
<b>Competențe transversale</b>	<ul style="list-style-type: none"> <li>- Abilitatea de a aplica tehnici specifice programării paralele în rezolvarea problemelor.</li> <li>- Abilitatea de a evalua creșterea de performanță obținută prin folosirea paralelizării.</li> <li>- Abilitatea de a implementa programe paralele.</li> </ul>

## 7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>- Cunoașterea, înțelegerea conceptelor de bază ale programării paralele.</li> <li>- Abilitatea de a lucra independent sau în echipă pentru a rezolva probleme într-un context de programare paralelă.</li> <li>- Cunoașterea, înțelegerea paradigmelor și tehnicilor de bază ale programării paralele.</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>- Obținerea de cunoștințe de bază despre programarea paralelă.</li> <li>- Cunoașterea și folosirea celor mai importante paradigme/tehnici de dezvoltare a programelor paralele.</li> <li>- Implementarea programe paralele folosind thread-uri Java(sau C#), sau Parallel Python sau OpenMP.</li> <li>- Realizarea unui Proiect(aplicație) folosind pentru implementare una dintre variantele specificate anterior</li> </ul>

## 8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere generală în programarea paralelă: - necesitatea folosirii paralelismului; - probleme și dificultăți în programarea paralelă; Arhitecturi paralele - <i>Taxonomii</i>	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
2. Tipuri de paralelism • Paralelism implicit • Paralelism explicit - Modelul Data-parallel - Modelul Message-passing - Modelul Shared-variable	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
3. PRAM (Parallel Random Access Machine) Rețele computaționale Evaluarea performanței programelor paralele: complexitate-timp, complexitate-procesor, accelerație, eficiență, cost; evaluare scalabilitate.	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
4. Faze în dezvoltarea programelor paralele (PCAM) - Partitionare, Comunicare, Aglomerare, Mapare Partitionare->Descompunere	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	

- functionala (task decomposition) - a domeniului de date (geometrica) distributii de date		
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, OpenMP</i>	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
7. Programare paralela in medii cu memorie distribuita. <i>MPI</i>	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
8. Programare paralela in medii cu memorie partajata si distribuita. <i>Parallel Python</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	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
13. Exemple de algoritmi paraleli - algoritmi de sortare - algoritmi de cautare	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
14. Presentare proiecte	Discutii, analize, evaluari.	

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

### 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 defineste dobandirea unor abilitati care constituie avantaje evidentiare 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 = 50%
10.5 Seminar/laborator			Implementarea proiectului = 50%
10.6 Standard minim de performanță			
Minim 5 la media finala.			

Data completării  
1.04.2014.....

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

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

Data avizării în departament  
.....

Director de departament  
.....