FIŞA DISCIPLINEI

1. Date despre program

1.1. Instituţia de învăţământ superior	Universitatea „Sapientia" din municipiul Cluj-Napoca
1.2. Facultatea/ DSPP	Facultatea de Ştiinţe Tehnice şi Umaniste din Târgu Mureş
1.3. Domeniul de studii	Calculatoare şi tehnologia informaţiei
1.4. Ciclul de studii	Licenţă
1.5. Programul de studiu	Calculatoare
1.6. Calificarea	Inginer

2. Date despre disciplină

2.0. Departamentul				Matematică-Informatică			
2.1 Denumirea disciplinei				Algoritmi paraleli şi distribuiţi			
2.2 Titularul disciplinei / activităţilor de curs				Conf.dr. Iclănzan Dávid			
2.3 Titularul (ii) activităţilor de	seminar			-			
	laborator			Conf.dr. Iclănzan Dávid			
	proiect			-			
2.4 Anul de studiu	4	2.5 Semestrul	7	2.6. Tipul de evaluare	EX	2.7 Regimul disciplinei	DI

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

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2	3.3 seminar /laborator/ proiect	0 /2/ 0
3.4 Total ore din planul de învăţământ	56	Din care: 3.5 curs	28	3.6 seminar / laborator / proiect	0 /28 /0

Distribuţia fondului de timp:	ore
a) Studiul după manual, suport de curs, bibliografie şi notiţe	21
b) Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren	24
c) Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri	13
d) Tutoriat	5
e) Examinări	6
f) Alte activităţi:	-

3.7 Total ore studiu individual	69
3.8 Total ore pe semestru	125
3.9 Numărul de credite	5

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	
4.2 de competenţe	

5. Condiţii (acolo unde este cazul)

5.1 De desfăşurare a cursului	• Sală de curs cu tablă şi videoproiector
5.2 De desfăşurare a seminarului/laboratorului/proiectului	• Laborator echipat cu calculatoare multicore, acces SSH server echipat cu 2 sau mai multe procesoare.

6. Competențele specifice acumulate

Competențe profesionale	• C4. Îmbunătățirea performantelor sistemelor hardware, software si de comunicații • C4.1 Identificarea și descrierea elementelor definitorii ale performantelor sistemelor hardware, software și de comunicații • C4.2 Explicarea interacțiunii factorilor care determina performantele sistemelor hardware, software și de comunicații • C4.3 Aplicarea metodelor si principiilor de baza pentru creșterea performantelor sistemelor hardware, software și de comunicații • C4.4 Alegerea criteriilor si metodelor de evaluare a performantelor sistemelor hardware, software și de comunicații • C4.5 Dezvoltarea de soluții profesionale pentru sisteme hardware, software și de comunicații bazate pe creșterea performantelor • C5. Proiectarea, gestionarea ciclului de viată, integrarea și integritatea sistemelor hardware, software și de comunicații • C5.1 Precizarea criteriilor relevante privind ciclul de viată, calitatea, securitatea și interacțiunea sistemului de calcul cu mediul și cu operatorul uman • C5.2 Utilizarea unor cunoștințe interdisciplinare pentru adaptarea sistemului informatic în raport cu cerințele domeniului de aplicații • C5.3 Utilizarea unor principii și metode de bază pentru asigurarea securității, siguranței și ușurinței în exploatare a sistemelor de calcul • C5.4 Utilizarea adecvată a standardelor de calitate, siguranță și securitate în prelucrarea informațiilor • C5.5 Realizarea unui proiect incluzând identificarea și analiza problemei, proiectarea, dezvoltarea și demonstrând o înțelegere a nevoii de calitate
Competențe transversale	

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	• Dezvoltarea gândirii științifice, familiarizarea viitorului specialist cu metodele paralele de prelucrare a datelor și cu sistemele de calcul pe care acestea funcționează.
7.2 Obiectivele specifice	• Cunoașterea arhitecturilor paralele și a modelelor de calcul paralel. • Înțelegerea diferențelor dintre diferitele arhitecturi și topologii de calcul paralel și a factorilor ce influențează scalabilitatea soluțiilor paralele. • Însușirea tehnologiilor și tehnicilor de programare pe sisteme cu memorie partajată și prin transfer de mesaje. • Dezvoltarea capacității de analiză și evaluare a programelor paralele.

8. Conținuturi

8.1. Curs	Metode de predare	Observații
Introducere în problematica sistemelor paralele și distribuite	expunere	2 ore
Obiectul și domeniile calculului paralel	expunere	2 ore
Taxonomia arhitecturilor paralele și modele de calcul paralel	expunere	4 ore
Programare SIMD.	expunere	2 ore
Programare paralelă pe sisteme cu memorie partajată. OpenMP.	expunere	2 ore
Primitive ale calculului paralel. Algoritm paralel de reducere.	expunere	2 ore

Metode paralele de calcul al prefixului.	expunere	2 ore
Algoritmi paraleli de sortare și selecție.	expunere	2 ore
Arhitecturi paralele cu transfer de mesaje. MPI.	expunere	2 ore
Algoritmi paraleli de multiplicare a matricelor.	expunere	2 ore
Evaluarea performanțelor calculului paralel	expunere	4 ore
Rețeaua sincronizată. Algoritmi de bază.	expunere	2 ore

Bibliografie obligatorie

1. Quinn MJ: Parallel programming in C with MPI and OpenMP, McGraw-Hill, 2004
2. Russ Miller, Laurence Boxer, Algorithms sequential and parallel: a unified approach, 2013
3. Iványi Antal: Párhuzamos Algoritmusok, ELTE Eötvös Kiadó, Budapest, 2003

Biliografie opțională

4. Dana Petcu, Procesare paralela, Ed. Eubeea Timisoara, 2001, http://www.info.uvt.ro/~petcu
5. Thomas Rauber, Gudula Rünger, Parallel programming: for multicore and cluster systems, 2010
6. Calvin Lin, Lawrence Snyder, Principles of parallel programming, 2009
7. Pacheco, Peter S., An introduction to parallel programming
8. Cormen – Leiserson - Rivest: Algoritmusok, Műszaki Könyvkiadó, Budapest, 2001

8.2 Seminar – nu este cazul	Metode de predare	Observații
8.3 laborator		
Procesarea paralelă sub Linux și Ubuntu. Aplicație cu fire de execuție	programare	2 ore
Tehnologii SIMD Intel.	programare	2 ore
Algoritmi geometrici paraleli.	programare	2 ore
Inițiere în utilizarea sistemului OpenMP	programare	2 ore
Rezolvări de probleme cu OpenMP	programare	4 ore
Inițiere în utilizarea sistemului MPI	programare	2 ore
Rezolvări de probleme cu MPI	programare	4 ore
Algoritmi paraleli de reducere și interclasare	programare	2 ore
Algoritm paraleli de sortare	programare	4 ore
Algoritm paralel de multiplicare a matricelor.	programare	2 ore
Verificare finală	programare	2 ore

Bibliografie

Quinn MJ: Parallel programming in C with MPI and OpenMP, McGraw-Hill, 2004
2. https://software.intel.com/sites/landingpage/IntrinsicsGuide/
3. www.openmp.org/mp-documents/OpenMP4.0.0.pdf
4. https://computing.llnl.gov/tutorials/mpi/
5. http://www.mcs.anl.gov/research/projects/mpi/tutorial/gropp/talk.html

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

- Conținutul disciplinei este în concordanță cu cerințele planului de învățământ, cu conținutul disciplinei predate la alte Universități din țară (Universitatea Babeș-Bolyai, Universitatea Tehnică din Cluj-Napoca) și de peste hotare (Universitatea Tehnică și de Științe Economice din Budapesta)
- Programarea, proiectarea soluțiilor paralele se realizează folosind tehnologii standard în industrie, conținutul cursului fiind adaptat pieței muncii in domeniul ITC.

10. Evaluare

A. Condiții de îndeplinit pentru prezentarea la evaluare:

Efectuarea tuturor lucrărilor de laborator și promovarea testului de evaluare la laborator

B. Criterii, metode și ponderi în evaluare:

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală

10.4 Curs		înțelegerea problemei, capacitate de a proiecta algoritmi paraleli și de a evalua criteriile de performanță și scalabilitate	examen scris cu subiect de teorie și rezolvare de probleme	40%
10.5	Seminar	-	-	-
	Laborator	înțelegerea problemei, capacitate de a proiecta și implementa algoritmi paraleli eficienți în vederea rezolvării unor probleme	Verificarea activității de laborator pe parcurs / și la sfârșitul semestrului	10% / 50%
	Proiect	-	-	-

10.6 Standard minim de performanță
1. Cunoașterea arhitecturilor de bază de calcul paralel (arbori, complet, de Bruin, latice, etc.) 2. Cunoașterea clasificării Flynn după fluxuri de date și fluxuri de instrucțiuni concurente. Exemple. 3. Abilitatea de a identifica paralelismul de date și a paralelismului funcțional în probleme exemplu. 4. Abilitatea de a evalua performanța unor algoritmi paraleli simpli. 5. Abilitatea de a paraleliza un algoritm simplu (ales de student) în pseudocod și în C++ / OpenMP sau MPI.

Semnătura titularului disciplinei:
Conf. dr. Iclănzan Dávid

Semnătura titularului /rilor de aplicații:
Conf. dr. Iclănzan Dávid

Data:
16. 09. 2019

Semnătura directorului de departament:
Conf. dr. ing. Kátai Zoltán

Semnătura coordonatorului programului de studii:
Ș.l.dr.ing. Szabó László Zsolt