

FISA DISCIPLINEI

1. DATE DE IDENTIFICARE

Titlul disciplinei:	Calcul Paralel
Codul disciplinei:	04.T.08.O.508
Titular de disciplină:	Prof. Felicia Ionescu
Tipul:	domeniu tehnic (T)
Numar ore curs:	28 ore
Numar ore aplicatii:	28 ore
Numarul de puncte de credit:	4
Semestrul:	8
Pachetul:	aria curiculara de specialitate
Preconditii:	parcursarea urmatoarelor discipline: <ul style="list-style-type: none">- Programarea calculatoarelor- Structuri de date si algoritmi- Programare obiect-orientata- Arhitectura sistemelor de calcul- Sisteme de operare

2. OBIECTIVELE DISCIPLINEI

- *Cursul* are ca obiectiv studierea modului de utilizare a procesoarelor multiple care colaboreaza pentru rezolvarea unei sarcini de calcul, precum si aspectele implicate privind arhitecturile de calcul, analiza, masurarea si predictia performantelor, limbaje, biblioteci de programare si algoritmi. Cursul este axat pe principiile generale ale calculului paralel si a stadiului actual de dezvoltare a algoritmilor pentru rezolvarea problemelor intens computationale.
- *Aplicatiile de laborator* au ca obiectiv programarea in diferite sisteme paralele. Posibilitatile includ multiprocesoare si sisteme cu transfer de mesaje.

3. COMPETENTE SPECIFICE

Dupa absolvirea acestui curs studentii vor putea evalua oportunitatea paralelizarii aplicatiilor si vor fi capabili sa dezvolte algoritmi paraleli pentru solutionarea unor probleme intens computationale, pentru diferite arhitecturi de calcul paralel (de uz general sau specializate).

Conform grilelor ACPART de competente specifice specializarii Ingineria Informatiei, studierea acestui curs va oferi studentilor competentele:

C3.1 Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice

C3.4 Evaluare comparativă, inclusiv experimentală, a alternativelor de rezolvare, pentru optimizarea performanțelor aplicațiilor software

C4.5 Dezvoltarea, implementarea și integrarea soluțiilor software

4. CONTINUTUL TEMATIC (SYLABUS)

a. Curs:

Capitol	Continut	Nr. Ore
1.	Introducere in calculul paralel 1.1. Motivarea paralelismului: aplicatii cu cerinte de calcul ridicate 1.2. Introducerea paralelismului in arhitecturile de calcul	2
2.	Arhitecturi de calcul paralel 2.1. Clasificarea arhitecturilor dupa fluxurile de control si de date 2.2. Clasificarea arhitecturilor dupa organizarea memoriei 2.3. Retele de interconectare in sistemele paralele si distribuite (retele statice si dinamice)	2
3.	Proiectarea algoritmilor paraleli 3.1. Etapele de proiectare a algoritmilor paraleli: partitionarea calculelor si a datelor, comunicarea intre task-uri, aglomerarea (gruparea) task-urilor, alocarea task-urilor 3.2. Analiza dependentelor intre task-uri 3.3. Paralelizarea buclelor de program 3.4. Paralelizarea automată a algoritmilor 3.5. Exemple de proiectare a algoritmilor paraleli	6
4.	Analiza si estimarea performantelor algoritmilor paraleli 4.1. Analiza si estimarea performantelor algoritmilor folosind complexitatea asimptotica 4.2. Accelerarea paralela – legea lui Amdahl 4.3. Algoritmi cost-optimali 4.5. Exemple de analiza a performantelor algoritmilor paraleli	4
5.	Programarea algoritmilor paraleli 5.1. Clasificarea tehnicilor de programare paralela 5.2. Crearea dinamica a proceselor si a thread-urilor 5.3. Limbaje si biblioteci de programare paralela	2
6.	Programarea paralela prin memorie partajate 6.1. Mecanisme de sincronizare intre procese si thread-uri: mutex-uri, semafoare, bariere de sincronizare 6.2. Biblioteca de thread-uri POSIX (PTHREAD) 6.3. Tehnologia OpenMP 6.4. Exemple de implementare a algoritmilor paraleli în sisteme cu memorie partajata	8
7.	Programarea paralela prin transfer de mesaje 7.1. Primitive de transfer de mesaje 7.2. Biblioteci de programare prin transfer de mesaje: MPI, PVM 7.3. Exemple de implementare a algoritmilor paraleli in sisteme cu transfer de mesaje.	4
	Total	28

b. Aplicații:

1.	Dezvoltarea programelor in sistemele de operare Unix-Linux; procese și thread-uri	4
2.	Dezvoltarea programelor paralele folosind biblioteca Pthread	4

3.	Dezvoltarea programelor paralele folosind biblioteca OpenMP	4
4.	Dezvoltarea programelor paralele folosind biblioteca MPI	4
5.	Algoritmi paraleli simpli: operații cu vectori și matrice	4
6.	Algoritmi paraleli complecși: algoritm de grupare (k-means clustering), algoritm de eliminare Gaussiană și Gauss-Jordan	4
7.	Colocviu de laborator	4
	Total	28

5. EVALUAREA

a) Activitățile evaluate și ponderea fiecăreia (conform cu Regulamentul studiilor de licență) :

- Activitate laborator 40 puncte
- Lucrare de verificare pe parcurs: 20 puncte
- Lucrare de verificare finală 40 puncte

b) Cerințele minimale pentru promovare

- obținerea a 50 % din punctajul activităților din timpul semestrului;
- obținerea a 50 % din punctajul total (50 puncte)

c) Calculul notei finale

Nota finală se calculează prin rotunjirea punctajului total obținut, dacă sunt îndeplinite cerințele minimale pentru promovare.

6. REPERE METODOLOGICE

Cursurile sunt prezentate folosind facilități multimedia.

Indrumarul de laborator este disponibil studenților în formă electronică. Lucrările se efectuează prin programare practică pe calculator (folosind limbajele și toolset-urile selectate pentru lucrarea respectivă) și interpretarea rezultatelor obținute. Se recomandă studenților studierea documentației înainte de prezentarea în sala de laborator.

7. BIBLIOGRAFIA

- Felicia Ionescu: *Principiile Calculului Paralel*, Ed. Tehnică, București, 1999.
- A. Grama, A. Gupta, G. Karypis, V. Kumar: *Introduction to Parallel Computing*, Addison Wesley, 2003.
- Jan Foster: *Designing and Building Parallel Programs*, An Online Publishing Project of Addison-Wesley Inc. and Argonne National Lab.
- *Posix Threads*, <http://www.llnl.gov/computing/tutorials/pthreads/>
- *OpenMP Library*, <http://www.openmp.org>
- *MPI Standard*, <http://www-unix.mcs.anl.gov/mpi>
- Felicia Ionescu, Gabriel Dimitriu, Cristina Stoica, Valentin Stoica: [Indrumar de laborator de Calcul Paralel](#)

SEF DE CATEDRA

Prof. Dr. Sever Pașca

TITULAR DE DISCIPLINA

Prof. Dr. Felicia Ionescu

