**FIŞA DISCIPLINEI**

**1. Date despre program**

| | |
|---|---|
| 1.1 Instituţia de învăţământ superior | **Universitatea Babes-Bolyai Cluj-Napoca** |
| 1.2 Facultatea | **Facultatea de Matematica si Informatica** |
| 1.3 Departamentul | **Departamentul de matematica** |
| 1.4 Domeniul de studii | **Informatica** |
| 1.5 Ciclul de studii | **Licenta** |
| 1.6 Programul de studiu / Calificarea | **Informatica romana** |

**2. Date despre disciplină**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2.1 Denumirea disciplinei | **Programare paralelă şi distribuită** | | | | | | |
| 2.2 Titularul activităţilor de curs | **Conf. Dr. Niculescu Virginia** | | | | | | |
| 2.3 Titularul activităţilor de seminar | **Conf. Dr. Niculescu Virginia** | | | | | | |
| 2.4 Anul de studiu | **3** | 2.5 Semestrul | **5** | 2.6. Tipul de evaluare | **E** | 2.7 Regimul disciplinei | **Obligatoriu** |

**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 | 2 |
| 3.4 Total ore din planul de învăţământ | 56 | Din care: 3.5 curs | 28 | 3.6 seminar/laborator | 28 |
| Distribuţia fondului de timp: | | | | | ore |
| Studiul după manual, suport de curs, bibliografie şi notiţe | | | | | 15 |
| Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren | | | | | 15 |
| Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri | | | | | 24 |
| Tutoriat | | | | | 5 |
| Examinări | | | | | 10 |
| Alte activităţi: ..... | | | | | - |

| | |
|---|---|
| 3.7 Total ore studiu individual | 55 |
| 3.8 Total ore pe semestru | 125 |
| 3.9 Numărul de credite | 5 |

**4. Precondiţii** (acolo unde este cazul)

| | |
|---|---|
| 4.1 de curriculum | • Fundamentele programarii, Programare orientata obiect<br>• Structuri de date si algoritmi |
| 4.2 de competenţe | • Abilitati de programare. |

**5. Condiţii** (acolo unde este cazul)

| | |
|---|---|
| 5.1 De desfăşurare a cursului | • Sala cu proiector |
| 5.2 De desfăşurare a seminarului/laboratorului | • laborator cu statii de lucru |

**6. Competenţele specifice acumulate**

| | |
|---|---|
| **Competenţe profesionale** | Fiecarea student trebuie sa dovedeasca faptul ca a dobandit un nivel de cunostinte si intelegere a domeniului si ca este capabil (a) sa exprime aceste cunostinte; si, de asemenea, ca poate folosi aceste cunostinte in rezolvarea unor probleme prin implementarea solutiilor folosind programarea paralela si distribuita.<br><br>**C6:**<br>C6.1 Identificarea conceptelor si modelelor de baza pentru sisteme de calcul si retele de calculatoare.<br>C6.4 Efectuarea de măsurători de performanţă pentru timpi de răspuns, consum de resurse.<br>C6.5 Realizarea unor proiecte de rețele de calculatoare/arhicteturi paralele si distribuite |

| | |
|---|---|
| **Competenţe transversale** | **CT1** Aplicarea regulilor de muncă organizată şi eficientă, a unor atitudini responsabile faţă de domeniul didactic-ştiinţific, pentru valorificarea creativă a propriului potenţial, cu respectarea principiilor şi a normelor de etică profesională<br>**CT3** Utilizarea unor metode şi tehnici eficiente de învăţare, informare, cercetare şi dezvoltare a capacităţilor de valorificare a cunoştinţelor, de adaptare la cerinţele unei societăţi dinamice și de comunicare în limba română și într-o limbă de circulație internațională |

**7. Obiectivele disciplinei** (reieşind din grila competenţelor acumulate)

| | |
|---|---|
| 7.1 Obiectivul general al disciplinei | - Insusirea principalelor entitati si concepte cu care se opereaza in contextul programării paralele, concurente şi distribuite.<br>- Prezentarea bazelor comunicării între procese şi threaduri, aflate pe acceeaşi maşină sau pe maşini aflate la distanţă.<br>- Insuşirea bazelor specifice ale programării paralele, concurente şi distribuite<br>- Cunoasterea, intelegerea paradigmelor si tehnicilor de baza ale programarii paralele.<br>- Intelegerea si folosirea unor sabloane de proiectare pentru dezvoltarea aplicatiilor paralele.<br>- Deprinderea folosirii unor frameworkuri pentru dezvoltarea aplicatiilor paralele si distribuite. |
| 7.2 Obiectivele specifice | • Arhitecturi paralele şi sisteme de programare paralelă<br>• Abilitatea de a aplica tehnici specifice programarii paralele in rezolvarea problem<br>• Abilitatea de evalua cresterea de performanta obtinuta prin folosirea paralelizarii.<br>• Abilitatea de a lucra independent sau in echipa pentru a rezolva probleme intr-un programare paralela si/sau distribuita.<br>• |

**8. Conţinuturi**

| 8.1 Curs | Metode de predare | Observaţii |
|---|---|---|
| C1 Introducere generala in programarea paralela si distribuita<br>- necesitatea folosirii paralelismului;<br>- programare paralela vs. progr. distribuita, vs. Progr. Concurenta<br>- niveluri de folosire a paralelismului | Expunere | |
| C2. Arhitecturi paralele – *Taxonomii*<br>□ Pipeline<br>□ Maşini vectoriale<br>□ Sisteme grid şi clustere<br>□ Supercalculatoare | Expuneri: concepte, exemple, studii de caz | |
| C3. Tipuri si modele de paralelism<br>• Paralelism implicit vs. Paralelism explicit<br>- Modelul Data-parallel<br>- Modelul Message-passing (distributed memory)<br>Modelul Shared-memory<br>Procese versus fire de excutie<br>• gestiunea proceselor | Expuneri: concepte, exemple, studii de caz | |
| C4. Mesage Passing parallel programing<br>Programare paralela in medii cu memorie distribuita.<br>*MPI* | Expuneri: concepte, exemple, studii de caz | |
| C5. Programare paralela in medii cu memorie partajata<br>- C++ Threads, Java Threads | Expuneri: concepte, exemple, studii de caz | |
| C6. Concurenta – concepte<br>Race-conditions, critical section, mutual exclusion, deadlock | Expuneri: concepte, exemple, studii de caz | |
| C7. Concurenta – concepte<br>Sincronizare: monitoare, semafoare, variabile conditionale | Expuneri: concepte, exemple, studii de caz | |
| C8. Programare paralela in medii cu memorie partajata.<br>- OpenMP. | Expuneri: concepte, exemple, studii de caz | |
| C9. Evaluarea performantei programelor paralele: complexitate-timp, complexitate-procesor, acceleratie, eficienta, cost; evaluare scalabilitate | Expuneri: concepte, exemple, studii de caz | |
| C10 *Data parallel programming*:<br>Cadrul general de dezvoltare a aplicaţiilor GPU<br>□ _Arhitectură; platforma NVIDIA<br>□ _API de programare; modelul CUDA | Expuneri: concepte, exemple, studii de caz | |

| | | |
|---|---|---|
| C11. . Faze in dezvoltarea programelor paralele (PCAM)<br>- Partitionare, Comunicare, Aglomerare, Mapare<br>Partitionare->Descompunere<br>- functionala (task decomposition)<br>- a domeniului de date(geometrica)<br>distributii de date<br>Granularitate, Grad de paralelizare(DOP), Task dependency | Expuneri: concepte, exemple, studii de caz | |
| C12- Sabloane pentru programarea paralela<br>*Master-slaves; Task-Farm/Work-Pool; Divide &Impera; Pipeline* | Expuneri: concepte, exemple, studii de caz | |
| C13 Sabloane pentru programarea distribuita | Expuneri: concepte, exemple, studii de caz | |
| C14. Analiza comparativa generala a noilor concepte/tehnici/principii/sabloane introduse | studii de caz | |

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

Bibliografie


1. Ian Foster. Designing and Building Parallel Programs, Addison-Wesley 1995.
2. Michael McCool, Arch Robinson, James Reinders, Structured Parallel Programming: Patterns for Efficient Computation, Morgan Kaufmann, 2012 .
3. F. Buschmann, K. Henney, D. C. Schmidt. Pattern-Oriented Software Architecture Volume 4: A Pattern Language for
4. Grama, A. Gupta, G. Karypis, V. Kumar. Introduction to Parallel Computing, Addison Wesley, 2003.
5. D. Grigoras. Calculul Paralel. De la sisteme la programarea aplicatiilor. Computer Libris Agora, 2000.
6. B. L. Massingill, T.G. Mattson, and B. A. Sanders, A Pattern Language for Parallel Programming. Wesley Software Patterns Series, 2004.
7. V. Niculescu. Calcul Paralel. Proiectare si dezvoltare formala a programelor paralele. Presa Univ. Clujana, 2006.
8. D.B. Skillicorn, D. Talia. Models and Languages for Parallel Computation. ACM Computer Surveys, 30(2) pg.123-136, June 1998.
9. Distributed Computing Volume 4, Wiley. 2007.
10. M. Richards. Software Architecture Patterns. Understanding Common Architecture. Patterns and When to Use Them 2015 O'Reilly Media.
11. D. Schmidt (Author), M. Stal (Author), H. Rohnert (Author), F. Buschmann. Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects Volume 2. Wiley, 2000.
12. B. Wilkinson, M. Allen, Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 2002
13. E.F. Van de Velde. Concurrent Scientific Computing. Spring-Verlag, New-York Inc. 1994.
14. Boian F.M. Ferdean C.M., Boian R.F., Dragos R.C. Programare concurenta pe platforme Unix, Windows, Java. Ed. Albastra, grupul Microinformatica, Cluj, 2002.

1. ***, Tutoriale OpenMP
2. ***, Tutoriale MPI
3. ***, Tutoriale CUDA


| | | |
|---|---|---|
| 8.2 Laborator | | |
| L1 –L2 . Threads vs. Processes | Discutii, exemplificare, evaluare | |
| L2-L4 MPI - exemple | Discutii, exemplificare, evaluare | |
| L5. Test1 practic – programare distribuita MPI | Discutii, exemplificare, evaluare | |
| L6- L8. Concurenta – conditionari, sincronizari cu exemplificari C, Java | Evaluare | |
| L9. Probleme de sincronizare | Discutii, exemplificare, evaluare | |
| L10. Test2 practic – programare concurenta- multithreading | | |
| L11. Client-Server | Discutii, exemplificare, evaluare | |
| L12 OpenMP | Discutii, exemplificare, evaluare | |
| L13-L14 CUDA | Evaluare | |

Bibliografie aditionala


1. Eckel, B., Thinking in Java, 4th Edition, New York: Prentice Hall, 2006.
2. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2004.
3. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.

4. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994.
5. J. Sanders, E. Kandrot. CUDA by Example. An Introduction to General-Purpose GPU Programming. Addison-Wesley. 2010.
6. A. WILLIAMS. C++ Concurrency in Action. PRACTICAL MULTITHREADING. MANNING, 2012.
7. ***, Tutoriale Java http://download.oracle.com/javase/tutorial/
8. ***, Tutoriale C# http://msdn.microsoft.com/en-us/library/aa288436%28v=vs.71%29.aspx
9. ***, Tutorial C++11
10. ***, OpenMP[http://openmp.org/wp/]
11. ***, MPI[http://www.mpi-forum.org/]

**9. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatori reprezentativi din domeniul aferent programului**

- Cursul respecta recomandarile curriculei ACM si IEEE pentru studiile de informatica.
- Cursul apare in planurile de invatamant a celor mai importante universitati din tara si strainatate.
- Firmele de soft considera continutul cursului important pentru dobandirea unor abilitatiavansate de programare.

**10. Evaluare**

| Tip activitate | 10.1 Criterii de evaluare | 10.2 metode de evaluare | 10.3 Pondere din nota finală |
|---|---|---|---|
| 10.1 Curs | Cunoasterea conceptelor de baza | Examen scris | 50% |
| 10.2 Laborator | Folosirea conceptelor introduse la curs pentru rezolvarea unor probleme concrete | Programe de laborator<br>- Evaluarea calitatii si completitudinii temelor primite | 30% |
| | | Teste practice | 20% |
| 10.4 Standard minim de performanţă | | | |
| o Media finala trebuie sa fie cel putin 5 (pe o scara de la 1 la 10).<br>o Nota de la examenul scris cel putin 4.5<br>o Nota de la laborator cel putin 4.5 | | | |

Data completării Titular de curs Titular de seminar

..15.04.2020... Conf. Dr. Niculescu Virginia Conf. Dr. Niculescu Virginia

Data avizării în departament Director de departament

..... ….....