DEGREE CURRICULUM

DISTRIBUTED COMPUTING AND APPLICATIONS

Coordination: CORES PRADO, FERNANDO

Academic year 2022-23

DISTRIBUTED COMPUTING AND APPLICATIONS 2022-23

Subject's general information

<table>
<tr><td>Subject name</td><td>DISTRIBUTED COMPUTING AND APPLICATIONS</td></tr>
<tr><td>Code</td><td>102027</td></tr>
<tr><td>Semester</td><td>2nd Q(SEMESTER) CONTINUED EVALUATION</td></tr>
<tr><td>Typology</td><td>
<table>
<tr><th>Degree</th><th>Course</th><th>Character</th><th>Modality</th></tr>
<tr><td>Bachelor's Degree in Computer Engineering</td><td>3</td><td>COMPULSORY</td><td>Attendance-based</td></tr>
<tr><td>Bachelor's Degree in Computer Engineering</td><td>3</td><td>OPTIONAL</td><td>Attendance-based</td></tr>
</table>
</td></tr>
<tr><td>Course number of credits (ECTS)</td><td>6</td></tr>
<tr><td>Type of activity, credits, and groups</td><td>
<table>
<tr><th>Activity type</th><td>PRALAB</td><td>TEORIA</td></tr>
<tr><th>Number of credits</th><td>3</td><td>3</td></tr>
<tr><th>Number of groups</th><td>1</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>Coordination</td><td>CORES PRADO, FERNANDO</td></tr>
<tr><td>Department</td><td>COMPUTER SCIENCE AND INDUSTRIAL ENGINEERING</td></tr>
<tr><td>Teaching load distribution between lectures and independent student work</td><td>6 ECTS = 25x6 = 150 hours
40% -> 60 classroom hours
60% -> 90 hours of autonomous student work</td></tr>
<tr><td>Important information on data processing</td><td>Consult this link for more information.</td></tr>
<tr><td>Language</td><td>Preferably in Spanish, in English if there is a foreign student.</td></tr>
<tr><td>Distribution of credits</td><td>Fernando Cores 6</td></tr>
</table>

DISTRIBUTED COMPUTING AND APPLICATIONS 2022-23

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
CORES PRADO, FERNANDO	fernando.cores@udl.cat	6	To be arranged by email

Subject's extra information

The course is eminently practical, so work practices and will have an important weight. Basically we work with two languages, C for parallel programming and Java for distributed objects (RMI). However, when we talk of distributed applications, as important as the program itself is the design of the application, which also intensively apply the concepts of software engineering.

To continue the course is essential for students to have good fundamentals and C programming Java. It is much harder, learn to develop distributed programs and parallel sequential programming if not previously mastered. In the course it is assumed that students are able to design, develop and debug sequential applications of medium difficulty without much trouble.

Learning objectives

- Introduce the basic concepts on Distributed Computing and the organisation of distributed systems.
- Provide an overview of the main architectures of Distributed Computing and his impact on the information technologies.
- Assimilate the fundamental principles and the distinct types of underlying models of Distributed Computing.
- Know the main paradigms of Distributed Computing and understand his strong points, his disadvantages and main fields of application.
- Comprise the technological challenges that represent the utilisation, the design and the implementation of the distributed systems.
- Provide an overview of the distributed systems, analysing different cases of study and applying them to solve real problems in different fields of Distributed Computing.
- Develop the skills of design and analysis of distributed systems that help to comprise, evaluate the quality the solutions proposed
- Encourage the adoption of the distributed model for the resource sharing on large scale, in a transparent form and independently of his physical location.

Competences

Strategic Competences of the UdL:

- CT2.Mastering a foreign language, especially English.
- CT3. Training Experience in the use of the new technologies and the information and communication technologies.

Cross-disciplinary competences:

- EPS11. Capacity to understand the needs of the user expressed in a no technical language.

Specific Competences:

- TI2. Capacity to choose, design, deploy, integrate, evaluate, build, manage, explode and keep the hardware, software and network technologies inside the cost and quality requirements.
- TI5. Capacity to select, deploy, integrate and manage systems of information that satisfy the needs of the organisation, within the cost and quality requirements.

- TI6. Capacity to conceive systems, applications and services based in network technologies, including Internet, web, e-commerce, multimedia, interactive services and mobile computation.

Subject contents

1. **Introduction to the Distributed Computing**
    1. Definitions and concepts
    2. Distributed Computing Challenges
    3. Types of Distributed Systems
    4. Distributed Computing systems
        1. Distributed Information systems
        2. Distributed Embedded systems.
        3. Distributed Systems Architectures
    5. Paradigms of Distributed Computing
    6. Message passing
    7. Client-server
    8. Peer-to-Peer
    9. Messages systems
        1. RPC (Remote Procedure Call)
        2. Distributed Objects
        3. Mobile agents
2. **Distributed Objects**
    1. Introduction
    2. Paradigm of distributed objects
    3. RMI: Remote method invocation
    4. RMI Advanced
    5. Callbacks
    6. Security Management in RMI
3. **Hardware and Software of Parallel Computing**
    1. Parallel architectures
    2. Networks of interconnection
    3. Clusters
    4. Job Queues
4. **Design of parallel algorithms**
    1. Introduction to parallel programming
    2. Performance of parallel applications
    3. Design of parallel applications
    4. Cases of Study
5. **Parallel programming**
    1. The message passing model
    2. The message passing interface
    3. Basic routines
    4. Point-to-point communications
    5. Collective communications
    6. Derived data-types
    7. Routines for groups and communicators administration
    8. Virtual topologies
    9. MPI- 2 and MPI -3
        1. Cases of study

Methodology

Theory Sessions (3 credits)

- Lecture: classes based on notes and transparencies where the concepts of the subject will be presented.
- Problems: The concepts of the subject will work through a series of exercises to be resolved collaboratively

and help assimilate key concepts.
- Use Cases: It will apply the techniques seen in class to real examples and their impact on application performance will be analyzed.

Laboratory Sessions (3 credits)

- Tutorials and personalized monitoring by groups of practices.
- Laboratory: technologies and APIs for distributed programming will be presented and worked through tutorials and examples.
- Problems: Making and correcting exercises related to both the theoretical and practical part of the course.

Autonomous work:

- The homework exercises and practices will be completed outside of class time.
- Forums Tool. In this space the student can raise doubts regarding the contents seen in the Theory and Laboratory sessions as well as pose all kind of doubts about the project. All students are encouraged to participate in resolving the doubts of their peers.Teachers participate to clarify or resolve those doubts that have no answer from students.

Development plan

Week	Description	Classroom Activity GG	Classroom Activity GM	Autonomous work activity
1	Presentation Introduction	Subject presentation	T1: Introduction of Distributed Computing	**Study literature and the program**
2	Introduction	T1: Introduction of Distributed Computing Problems: Distributed computing challenges	T1: Introduction of Distributed Computing	Problems: Challenges
3	Introduction	T1: Introduction of Distributed Computing	T1: Introduction of Distributed Computing	Problems: Challenges
4	Distributed Objects	Problems: Distributed systems requirements & paradigms	T2: Distributed Objects and Remote Invocation **Tutorial: RMI**	Problems: Paradigms
5	Distributed Objects	T2: Distributed Objects and Remote Invocation Practice 1: Presentation	T2: Distributed Objects and Remote Invocation **Tutorial: RMI**	RMI Practice 1
6	Distributed Objects Parallel Computing	T2: Distributed Objects and Remote Invocation **Tutorial: RMI**	T3: Parallel Computing Hardware & Software	RMI Practice 1
7	Parallel Computing	T3: Parallel Computing Hardware & Software **Tutorial: SGE**	T3: Parallel Computing Hardware & Software **Tutorial: SGE**	Practice 1 Tutorial: SGE
8	Parallel Computing	T3: Parallel Computing Hardware & Software **Tutorial: SGE**	Practice 1: Implementation	Practice 1 Tutorial: SGE
9		**1er Partial**		Study
10	Design of parallel applications	T4: Design of parallel algorithms	T4: Design of parallel algorithms Practice 2: Presentation	Practice 2
11	Design of parallel applications	T4: Design of parallel algorithms	Problems: Parallel programs design	Practice 2 Problems: Design

12	Parallel programming	T5: Parallel programming **Tutorial: MPI**	T5: Parallel programming **Tutorial: MPI** Practice 2: Design delivery	Practice 2
13	Parallel programming	T5: Parallel programming **Tutorial: MPI**	Visiting Professor: Stéphane Devismes	Practice 2 Tutorial MPI
14	Parallel programming	T5: Parallel programming **Tutorial: MPI**	T5: Parallel programming **Tutorial: MPI**	Practice 2 Tutorial MPI
15	Parallel programming	Problems: Implementation of parallel programs	Practice 2: Implementation Delivery	Practice 2
16		**2nd Partial**		Study
17		**2nd Partial**		Study
18		TUTORIAS		
19		**Recovery**		Study

Evaluation

Table. Assessment Activities

Acr.	Assessment activity	Weighting	Minimum Grade	In Group	Mandatory	Recoverable
P1	*1st Partial Exam*	20%	NO	NO	YES	YES
P2	*2nd Partial Exam*	20%	NO	NO	YES	YES
PRA	*Practices*	30%	4	NO	YES	YES
WRK	*Works/Tutorials*	25%	NO	YES/NO	NO	NO
PCL	*Class participation*	10%	NO	NO	NO	NO
It must be approved all practices individually. A practice is considered suspended if fails to reach 4.						
FinalGrade = 0.20*P1 + 0.20*P2 + 0.30*PRA + 0.25*WRK + 0.10*PCL						

The subject approves with a final makr upper or equal to 5 and having done properly the laboratory practices (all they with at least a mark of 4)

The final mark for the subject it is obtained from the pondered sum of the marks of the two examinations and the practices plus the cllass participation marks and the continued assessment.

The subject has two partial examns, each one of them with a weight of 20% in the final mark. These exams are mandatory.

The realisation and the overcoming of the laboratory practices is mandatory to approve the subject. The practices will be evaluated with a note that will represent 30% of the final note of the subject. The copy of any one of the practices of the subject will involve suspending the whole practices and the subject.

It does not exist minimum marks in the written examns to obtain the final mark of the subject. Only it exists a minimum note (4) for the laboratory practices.

Bibliography

Basic Bibliography:

- Coulouris G, Dollimore J., Kindberg T.: “Sistemas distribuidos: Conceptos y diseño”; Addison-Wesley, 2001.
- M.L. Liu, “Computación distribuida”. Edt. Addison Wesley, 2004
- Peter Pacheco, "An Introduction to Parallel Programming", Morgan Kaufmann, 2011.

Extended Bibliography:

- M. Ben-Ari, "Principles of Concurrent and Distributed Programming", Addison-Wesley, 2nd Edition, 2006
- Rajkumar Buyya: “High Performance Cluster Computing: Architectures and Systems”, Edt. Pearson Education; 1st edition 1999
- Rajkumar Buyya: “High Performance Cluster Computing: Programming and Applications”, Volume 2 , Edt. Prentice Hall, 1st edition 1999.