DEGREE CURRICULUM

DISTRIBUTED COMPUTING AND APPLICATIONS

Coordination: CORES PRADO, FERNANDO

Academic year 2023-24

DISTRIBUTED COMPUTING AND APPLICATIONS 2023-24

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 ENGINEERING AND DIGITAL DESIGN</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 2023-24

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 and operations
    6. Derived datatypes
    7. MPI-2 Extensions
        1. Parallel Input/Output
        2. Dynamic process management

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**	Tutorial: RMI Practice 1
6	Distributed Objects Parallel Computing	T2: Distributed Objects and Remote Invocation **Tutorial: RMI**	T3: Parallel Computing Hardware & Software	Tutorial: 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**	T3: Parallel Computing Hardware & Software **Tutorial: SGE**	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	T4: Design of parallel algorithms	Practice 2
12	Design of parallel applications	Problems: Parallel programs design	**Wednesday Holydays**	Practice 2 Problems: Design

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

Evaluation

ASSESSMENT

The course is passed with a final mark of 5 or more, obtaining an average mark of 4 in the exams and having completed the laboratory practices correctly.

The final grade for the course is derived from the weighted sum of the grades of the 2 exams, the practical marks and continuous assessment.

Table. Assessment Activities

Block	Acr	Assessment activity	Weighting	Minimum Grade	In Group	Mandatory	Recoverable
Theory	*P1*	*1st Partial Exam*	20%	4	NO	YES	YES
	P2	*2nd Partial Exam*	20%	4	NO	YES	YES
PRA1	*PRA1*	*Practice 1*	16%	NO	YES (<=2)	YES	YES/NO
PRA2	*PRA2*	*Practice 2*	16%	NO	YES (<=2)	YES	YES/NO
Cont. Ass.	*TR*	*Works*	7%	NO	YES (<=2)	NO	NO
	TU	*Tutorials*	21%	NO	NO	NO	NO
	PA	*Class Participation*	5%	NO	NO	NO	NO
There is a minimum mark of 4 in the average of the written exams to be able to pass the subject							
Final Grade = 0,20*(P1+P2) + 0,16*(PRA1+ PRA2) + 0,07*TR + 0,21*TU + 0,05*PA							

The subject has two partial exams, each of them with a weight of 20% in the final grade. These exams are compulsory and eliminate material. There is a minimum mark (4) for the theory grade (average of the two partial exams). If the weighting of the different marks is greater than or equal to 5, but the minimum grade of the theory exam has not been reached, then the subject will be considered failed with a 4.9.

The completion of laboratory practices is mandatory to pass the subject. The practices will be evaluated with a

grade that will represent 32% of the final grade for the subject. Of the two practices of the subject, only one can be recovered in the second call. The practices recovered will have a 20% penalty in the mark obtained.

The copy of any of the practices of the subject will imply failing the subject.

Regarding continuous assessment activities, this consists of group work (7%), plus 3 tutorials (basic programming exercises) with a weight of 7% each. Finally, class participation will also be considered with an extra 5% on the final grade.

ALTERNATIVE ASSESSMENT

In the case of requesting the alternative evaluation (by family or work conciliation), this will consist of carrying out a final exam, plus the delivery of the 2 practices of the subject. The requirements for these activities are the same as those applied in the normal assessment.

Table. Alternative Assessment Activities

Block	Acr	Assessment activity	Weighting	Minimum Grade	In Group	Mandatory	Recoverable
Theory	*EX*	*Final Exam*	50%	4	NO	YES	YES
PRA1	*PRA1*	*Practice 1*	25%	NO	NO	YES	YES/NO
PRA2	*PRA2*	*Practice 2*	25%	NO	NO	YES	YES/NO
There is a minimum mark of 4 in the average of the written tests to be able to pass the subject.							
Final Grade = 0,5*EX + 0,25*(PRA1+PRA2)							

The course is passed with a final grade greater than or equal to 5 and obtaining at least a 4 in the final exam and having completed the laboratory practices correctly.

The final grade for the subject is obtained from the weighted sum of the final exam grade and the practical marks.

There is a minimum mark (4) for the theory mark. If the weighting of the different marks is greater than or equal to 5, but the minimum grade of the theory exam has not been reached, then the subject will be considered failed with a 4.9.

The practices will be evaluated with a mark that will represent 50% of the final grade for the course. The practices will be delivered all together after taking the exam. Of the two practices of the subject, only one can be recovered in the second call. The practices recovered will have a 20% penalty in the mark obtained.

The copy of any of the practices of the subject will imply failing the subject.

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.