Degree	Type	Year	Semester
2503758 Data Engineering	OB	2	2

Contact

Name: Eduardo Cesar Galobardes

Email: eduardo.cesar@uab.cat

Teaching groups languages

You can check it through this link. To consult the language you will need to enter the CODE of the subject. Please note that this information is provisional until 30 November 2023.

Teachers

Eduardo Cesar Galobardes

Cristina Peralta Quesada

Betzabeth del Carmen León Otero

Sandra Méndez Caparrós

Prerequisites

Although there are no formally established prerequisites and the subject will provide students with the means to acquire the knowledge described in the syllabus, it is advisable: a good knowledge of programming, of the computer structure and of the operating system at the level of user programmer (Fundamentals of Computer Science, Advanced Programming, Performance Engineering).

Objectives and Contextualisation

The objective of this subject is to study high performance and parallel computing systems, multiprocessor and multicomputer systems, paradigms of parallel programming, learn to develop applications with shared memory, introduce the concept of message passing and applications in distributed memory, and analyze the performance of these applications.

The theoretical concepts about programming paradigms, shared memory and message passing, are reinforced with problem sessions and labs in which students learn to program using parallel programming languages.

All the components described in this subject should allow the student to understand the performance of high performance and parallel systems and, to a certain extent, be able to perform a simple design of a parallel application and evaluate its performance.

Competences

- Design efficient algorithmic solutions to computational problems, implement them in the form of robust software developments which are structured and easy to maintain, and verify their validity.
- Develop critical thinking and reasoning and know how to communicate it effectively in both your own language and in English.
- Make a critical evaluation of work carried out.
- Plan and manage the available time and resources.
- Search, select and manage information and knowledge responsibly.
- Students must be capable of collecting and interpreting relevant data (usually within their area of study) in order to make statements that reflect social, scientific or ethical relevant issues.

Learning Outcomes

1. Analyse the available parallel-programming models, paradigms and languages to determine which one is best suited to the needs of an application.
2. Develop critical thinking and reasoning and know how to communicate it effectively in both your own language and in English.
3. Develop parallel applications based on the most common existing paradigms.
4. Make a critical evaluation of work carried out.
5. Plan and manage the available time and resources.
6. Search, select and manage information and knowledge responsibly.
7. Students must be capable of collecting and interpreting relevant data (usually within their area of study) in order to make statements that reflect social, scientific or ethical relevant issues.
8. Use the most suitable tools and methodologies to evaluate the functionality and performance of the parallel/distributed applications developed.

Content

Topic 1: Introduction to high performance systems.

Advanced programming in C language and importing C code from Python. Concurrency: definition, race conditions, critial region, and mutual exclusion mechanisms.

Introduction to high performance systems, parallel systems, multiprocessors and multicomputers. Execution of parallel applications in high performance systems.

Topic 2: Introduction to parallel systems and algorithms

SIMD (Single Instruction, Multiple Data), MIMD (Multiple Instruction, Multiple Data). Models of parallel applications.

Definition of parallel algorithms. Examples of parallel algorithms. Development of parallel algorithms.

Topic 3: Parallel programming

Paradigms of parallel programming. Applications based on message passing. MPI standard (Message Passing Interface). Applications based on shared memory. Standards OpenMP (Open multiprocessing), OpenACC (Open Accelerators) and Cuda. Parallelism in Python. Development of parallel applications using MPI, OpenMP, OpenACC, Cuda and Python.

Topic 4: Performance Analysis

Performance analysis of parallel systems. Performance evaluation of parallel systems. Examples of performance evaluation tools.

Methodology

During the development of the subject four types of teaching activities cam be differentiated:

- Theoretical classes: explanation on the blackboard of the theoretical part of each program topic. The typical structure of a theoretical lesson will be the following: first, an introduction will be made where the objectives of the lesson and the contents to be discussed will be briefly presented. Next, the contents of the lesson will be discussed, including narrative expositions, formal developments that provide the theoretical foundations, including examples that illustrate the application of the discussed contents. Finally, the professor will present the conclusions of the lesson. Throughout the course there will be continuous assessments of groups of topics.
- Practical classes: All topics will be accompanied by a list of problems that the student should try to solve. In this sense, and as the student progresses and deepens in his/her knowledge, these problems will be more and more complex. The seminars will be the natural forum in which the development of practical work can be discussed, contributing the knowledge that the student lacks in order to carry it forward. The mission of the practical classes is to bridge the theoretical classes and the laboratory classes, which will promote the capacity for analysis and synthesis, critical reasoning, and that will train the student in solving problems. Those that the teacher considers of greater interest or in which the students find more difficulty will be developed on the blackboard. Before the beginning of each topic the professor may propose a list of exercises that the students will have to solve and deliver.
- Laboratory classes: The practical part of the theoretical topics will be completed with sessions in the laboratory, where the student will develop a series of programs and should try to solve a specific problem that will be received at the beginning of the semester. Some of these exercises must be delivered on the specified dates. The lab sessions will be developed in groups of two or three students. The subject includes a least 6 sessions in the laboratory, lasting 2 hours each, where the student will carry out the exercises.
- Written assignment: During the course, students in groups of two or three students will do develop a written assignment. Each group will write or improve an article of the Viquipèdia on a topic related to the subject. With this exercise, the student is expected to be able to present in writing topics related to the high performance computing for a generic audience.

This approach to work is oriented to promote an active learning and developing competencies of organization and planning skills, oral and written communication, teamwork and critical reasoning. The quality of the exercises carried out, their presentation and their functioning will be especially valued.

Transversal competences:

As mentioned before, the subject includes as a teaching activity the realization of a writteng assignment that consists in the development of an article to the Vikepèdia which is oriented to communicate efficiently knowledge, results and skills, both in professional environments non expert audiences, making efficient use of ICT in communication and the transmission of ideas (GT02, GT03 and GT07 competencies).

The students have material for the development of articles in the Vikepèdia (which includes the Viquiprojecte page: Còmput d'altes prestacions). During the semester a minimum of two meetings with the groups are held to discuss and evaluate the evolution of the work.

The teaching management of the subject will be done through the Virtual Campus (https://cv2008.uab.cat/), which will serve to see the materials, manage the groups of practices, make the corresponding deliveries, see the grades, communicate with professors, etc.

Annotation: Within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Labs (Development of programming exercices in the lab)	12	0.48	1, 4, 6, 3, 2, 5, 8
Problems (Solving exercices)	9	0.36	1, 4, 6, 3, 2, 5, 8
Theory	20	0.8	1, 6, 3, 2, 8
Type: Autonomous			
Autonomous study	30	1.2	1, 6, 3, 5, 7, 8
Labs preparation	30	1.2	1, 4, 6, 3, 5, 8
Problem preparation	20	0.8	1, 6, 3, 5, 7, 8
Research	15	0.6	4, 6, 2, 5, 7

Assessment

The objective of the assessment process is to verify that the student has achieved the knowledge and skills defined in the objectives of the subject, as well as the associated competences.

Four types of activities will be assessed independently, and the weighted sum of them will give the final grade. These activities are:

1. Theory (T)
2. Solution of the laboratory practices (PL)
3. Writing an article on Wikipedia (VA)
4. Completion of individualized practical exercises (PA).

The part of Theory (T) will be evaluated with two individual written exams throughout the course. The final grade of Theory will come out of the weighted sum of the two examns (0.5 * Exam 1 + 0.5 * Exam 2). There will be a second chance to recover this part on the day we have assigned in the June exams week. The parts that have not been passed in the partial theory tests can be recovered as long as the average grade of the partials is at least 3.0. Recovery exams cannot be used to raise a grade. The minimum grade for passing this part is 5.

The part of Laboratory exercises (PL) will be evaluated by group. There are four deliveries. The final grade will come from the weighted sum of the three deliveries (X0 * Delivery 1 + X1 * Delivery 2 + X2 * Delivery 3 + X3 * Delivery 4, X0 + X1 + X2 + X3 = 1). To pass the PL the minimum mark will have to be 5 i pass a validation test. There is only one opportunity to pass this part (this part can not be recovered).

The writing assigment of an article in Wikipedia (VA) will be evaluated by group. Each group will have a topic related to the contents of the subject to develop in the Wikipedia (the article may already exist or be newly created). The article will be developed during the course and will be discussed between the instructor and the members of the group during tutoring hours. The grade of each member of the group will be determined by the quality of the article and participation in the discussion. There is onlyone opportunity to pass this activity (this part can not be recovered). Consequently, in case a student does not have a grade of revision of the article,he will not be able to opt for an A in the recovery.

The practical exercises (PA) will consist of solving very specific programming problems that will be relate in some cases to those that will be found in the laboratory exercices. It is expected that each student will focus in

the resolution of a set of specific problems isolated from the most general case of the lab exercices. The value of these exercises is 30% of the final mark and given their nature and objective are not recoverable.

The final grade of the subject will be the weighted sum of the grades of each of the four activities: 30% of Theory, 10% writing an article in Wikipedia, 30% Resolution of individual practical exercises and 30% of Resolution of laboratory exercices. The result to pass the subject will have to be >= 5.

In case a student does not pass the subject due to not reaching the minimum score in any of the mandatory parts (Theory or Laboratory exercices), even though the weighted average is equal or superior to 5, the final grade of the subject will be 4.5.

In the event that the average does not reach 5, the official grade will be the average mark obtained numerically.

If the student delivers any activity, it is understood that he/she is participating in the subject and will be evaluated. If you do not deliver any activity, then it can be considered Non-evaluable.

Granting an honorific matriculation qualification is a decision of the faculty responsible for the subject. The regulations of the UAB indicate that MH can only be awarded to students who have obtained a final grade of 9.00 or more. It can be granted up to 5% of MH of the total number of students enrolled.

The dates of continuous evaluation and assignment delivery will be published on the virtual campus and may be subject to possible changes to adapt to possible incidents; the virtual campus will always inform about these changes since it is understood that the CV is the usual mechanism for exchanging information betweenprofessors and students.

For each assessment activity, a place, date and time of revision will be indicated in which the student will be able to review the activity with the professor. In this context, claims can be made about the activity grade, which will be evaluated by the professors responsible for the subject. If the student does not submit to this review, this activity will not be reviewed later.

SUMMARY

If ((T> = 5) i (P> = 5) then

NF = 0.3 * T + 0.1 * VA + 0.3 * PA + 0.3 * PL

If (NF> = 5) then PASS

else FAIL

else FAIL

Repeating students: repeating students who have passed laboratory exercices in previous editions of the subject may request the validation of this part of the subject. The rest of the assessment activities must be carried out under the same conditions as the other students.

This subject does not offer the possibility of taking a unique/single evaluation.

Use of AI tools (e.g. Chat GPT): The use of such tools will only be restricted in written tests (theory exams, problem quizzes, and practice validation tests). This means that it is imperative that you make critical use of these tools, that is, that you use them to learn, not to copy.

Note about plagiarism:

Without prejudice to other disciplinary measures deemed appropriate, and in accordance with the current academic regulations, irregularities committed by a student who maylead to a variation of the qualification in an assessable activity will be graded with zero (0). Assessment activities qualified in this way and by this procedure will not be recoverable. If it is necessaryto pass any of these assessment activities to pass the subject, this subject will be suspended directly, without opportunity to recover it in the same course. These irregularities include, among others:

- the total or partial copy of a lab exercice, report, or any other evaluation activity;
- let another student to copy;
- present a group work not done entirely by the members of the group (applied to all members and not only to those who have not worked);
- present as own materials prepared by a third party, even if they are translations or adaptations, andgenerally works with non-original and exclusive elements of the student;
- have communication devices (such as mobile phones, smart watches, pens with camera, etc.) accessible during theoretical-practical assessment tests (individual exams);
- talk with classmates during the individual theoretical-practice tests (exams);
- copy or attempt to copy from other students during the theoretical-practical assessment tests (exams);
- use or attempt to use written material related to the subject during the theoretical-practical evaluation tests (exams), when these have not been explicitly allowed.

If you do not pass the subject due to the fact that some of the evaluation activities do not reach the minimum grade required, the numerical official grade will be the lowest value between 4.5 and the weighted average of the grades. With the exceptions that the "Non-Appraising" qualification will be awarded to students who do not participate in any of the assessment activities, and that the numerical official grade will be the lowest value between 3.0 and the average Weighted grades in case the student has committed irregularities in an evaluation act (and therefore the subject cannot beapproved by compensation). In future editions of this subject, the student who has committed irregularities in an evaluation act will not be validated any of the assessment activities carried out.

In summary: copy, let copy or plagiarize (or attempt) in any of the assessment activities will lead to a FAIL, not compensable and without validations of parts of the subject in subsequent courses.

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Individual partial exam 1	15%	2	0.08	1, 3, 2, 8
Individual partial exam 2	15%	2	0.08	1, 3, 2, 8
Individually solving practical exercices	30%	6	0.24	1, 4, 6, 3, 2, 5, 8
Programming exercises	30%	3	0.12	1, 4, 6, 3, 2, 5, 8
Viquipèdia entry	10%	1	0.04	4, 6, 2, 5, 7

Bibliography

- Parallel Programming Concepts and Practice. Bertil Schmidt, Jorge González-Domínguez, Christian Hundt, Moritz Schlarb. Morgan Kaufmann, 2018
- An Introduction to Parallel Programming, 2nd edition. Peter S. Pacheco. Morgan Kaufmann. 2018
- Programming Massively Parallel Processors, A Hands-on Approach. David B. Kirk, Wen-mei W. Hwu. Morgan Kaufmann. 3th Edition. 2018
- Computer Architecture. A quantitative approach. John L. Hennessy, David A. Patterson. Morgan Kaufmann. 6a edición. 2018
- Introduction to Parallel Computing. A. Grama et alter. Addison Wesley, Second Edition, 2003.
- Parallel Program Development For Cluster Computing: Methodology, Tools and Integrated Environments. Edited by J. C. Cunha, P. Kacsuk, S. C. Winter. Nova Science Publishers, Inc., 2001.
- Parallel Programming with MPI, Peter Pacheco, Morgan Kauffman, 1996
- OpenACC Programming and Best Practices Guide (https://www.openacc.org/sites/default/files/inline-files/OpenACC_Programming_Guide_0.pdf

Software

In this course you will work on clusters of the Department of Computer Architecture and Operating Systems, consequently, you will need an ssh client to make the connections (integrated in any computer with Linux operating system) and it is highly recommended to use a development environment that allows remote connection (such as MobaXTerm for Windows [https://mobaxterm.mobatek.net/] or Visual Studio Code for all platforms [https://code.visualstudio.com/]).