

High-Performance Computation

Code: 104404
ECTS Credits: 6

Degree	Type	Year	Semester
2503740 Computational Mathematics and Data Analytics	OB	3	1

The proposed teaching and assessment methodology that appear in the guide may be subject to changes as a result of the restrictions to face-to-face class attendance imposed by the health authorities.

Contact

Name: Eduardo Cesar Galobardes
Email: Eduardo.Cesar@uab.cat

Use of Languages

Principal working language: catalan (cat)
Some groups entirely in English: No
Some groups entirely in Catalan: No
Some groups entirely in Spanish: No

Teachers

Albert Coca Abelló

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, of the operating system at the level of user programmer and how messages are sent over the network (Foundations of Computers, System Software, Introduction to Programming)

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, develop and evaluate efficient algorithmic solutions to computational problems in accordance with the established requirements.
- Implement and optimise applications based on the functions and structure of parallel, distributed and cloud systems, computer networks and internet.
- Make effective use of bibliographical resources and electronic resources to obtain information.

- Students must be capable of applying their knowledge to their work or vocation in a professional way and they should have building arguments and problem resolution skills within their area of study.
- Students must develop the necessary learning skills to undertake further training with a high degree of autonomy.
- Students must have and understand knowledge of an area of study built on the basis of general secondary education, and while it relies on some advanced textbooks it also includes some aspects coming from the forefront of its field of study.
- Using criteria of quality, critically evaluate the work carried out.

Learning Outcomes

1. Analyse and evaluate computer architectures, including parallel and distributed platforms, as well as developing and optimising software for such platforms.
2. Develop and optimise software for parallel and distributed platforms.
3. Evaluate and analyse the complexity of computational algorithmic solutions in order to develop and implement that which guarantees best performance.
4. Implement recursive solutions to programming problems.
5. Make effective use of bibliographical resources and electronic resources to obtain information.
6. Select and use the appropriate programming strategies to solve a given problem.
7. Students must be capable of applying their knowledge to their work or vocation in a professional way and they should have building arguments and problem resolution skills within their area of study.
8. Students must develop the necessary learning skills to undertake further training with a high degree of autonomy.
9. Students must have and understand knowledge of an area of study built on the basis of general secondary education, and while it relies on some advanced textbooks it also includes some aspects coming from the forefront of its field of study.
10. Understand and apply the basic principles and techniques of parallel, concurrent and distributed programming.
11. Using criteria of quality, critically evaluate the work carried out.

Content

Content

Topic 1: Introduction to high performance systems.

C advanced programming. Concurrency: definition, race conditions, critical 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: Classification of parallelism

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

Topic 3: Parallel Algorithms

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

Topic 4: 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) and OpenACC (Open Accelerators). Parallelism in Python. Development of parallel applications using MPI, OpenMP, OpenACC and Python.

Topic 5: 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 can 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 lab sessions, 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 6 sessions in the laboratory, lasting 2 hours each, where the student will carry out the exercises.

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.

Variations in the teaching methodology due to the health situation present in 2020:

During the 1st semester of the 2020-21 academic year, it is planned that this subject will be done in a partial attendance regime.

Given the exceptional nature of this fact, the specific conditions under which the course will take place are not included in this guide. At the beginning of the course you will have a plan adapted to the circumstances, which may vary according to the events.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Exercise Seminars	11	0.44	1, 11, 3, 10, 2, 4, 9, 7, 6, 5
Labs	10	0.4	1, 11, 3, 10, 2, 4, 7, 6, 5

Theory	24	0.96	1, 3, 10, 9, 8, 5
Type: Autonomous			
Autonomous preparation	30	1.2	1, 3, 10, 8, 7, 6, 5
Lab preparation	30	1.2	1, 11, 3, 10, 2, 4, 7, 6, 5
Seminar preparation	30	1.2	1, 11, 3, 10, 2, 4, 7, 6, 5

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. Completion of 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 exams ($0.5 * \text{Exam 1} + 0.5 * \text{Exam 2}$). There will be a second chance to recover this part on the day we have assigned in the February exams week. The parts that have not been passed in the partial theory exams can be recovered. The minimum grade for passing this part is ≥ 4.5 .

The part of Laboratory exercises (PL) will be evaluated by group. There number of deliveries will depend on the number of technologies introduced in the course. The final grade will come from the weighted sum of the deliveries (the number of deliveries and their weight in the evaluation will be informed at the beginning of the course). To pass the PL the minimum mark will have to be ≥ 4.5 . There is only one opportunity to pass this part (this part can not be recovered).

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 exercises. 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 exercises. Given their nature and objective these exercises are not recoverable (not delivering an assignment or delivering after its deadline implies a 0 mark).

The final grade of the subject will be the weighted sum of the grades of each of the four activities: Theory, Resolution of individual practical exercises and Resolution of laboratory exercises. The result 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 exercises), 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 between professors 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 $(T \geq 4) \wedge (P \geq 4)$ then

$$NF = 0.35 * T + 0.25 * VA + 0.2 * PA + 0.40 * PL$$

If $(NF \geq 5)$ then PASS

else FAIL

else FAIL

Repeating students: repeating students who have passed laboratory exercises 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.

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 may lead 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 necessary to 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 exercise, 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, and generally 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 be approved 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
1st Individual exam	17.5%	2	0.08	3, 4, 9, 8, 7, 6
2nd Individual exam	17.5%	2	0.08	1, 3, 10, 2, 9, 8, 7, 6
Exercises assessment	25%	6	0.24	1, 11, 3, 10, 2, 4, 6, 5
Laboratories	40%	5	0.2	1, 11, 3, 10, 2, 4, 8, 7, 6, 5

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 al. 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 Kaufman, 1996
- OpenACC Programming and Best Practices Guide (https://www.openacc.org/sites/default/files/inline-files/OpenACC_Programming_Guide_0.pdf)