Universitat Autònoma de Barcelona

2023/2024

Parallel Programming

Code: 106557
ECTS Credits: 6

Degree	Type	Year	Semester
2504392 Artificial Intelligence	OT	3	0
2504392 Artificial Intelligence	OT	4	0

Contact

Name: Anna Barbara Sikora

Email: anna.sikora@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

Christian Guzman Ruiz

Prerequisites

Although there are no formally established prerequisites and it is the responsibility of the subject itself to provide the students with a means to acquire the knowledge described in the subject's content section, it is recommended: a good knowledge of programming, of the operation of a computer and the operating system at the user programmer level (Fundamentals of Programming I and II, Fundamentals of Computing).

Objectives and Contextualisation

The objective of this subject is to know high performance and parallel computing systems, multiprocessor and multi-computing systems, paradigms of parallel programming, learn to develop applications using message passing or shared memory and analyze the performance of these applications.

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

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

Competences

Artificial Intelligence

- Analyse and solve problems effectively, generating innovative and creative proposals to achieve objectives.
- Conceptualize and model alternatives of complex solutions to problems of application of artificial intelligence in different fields and create prototypes that demonstrate the validity of the proposed system.
- Design, implement, analyse and validate efficient and robust algorithmic solutions to computational problems derived from the design of intelligent systems.
- Develop critical thinking to analyse alternatives and proposals, both one's own and those of others, in a well-founded and argued manner.
- Introduce changes to methods and processes in the field of knowledge in order to provide innovative responses to society's needs and demands.
- Work cooperatively to achieve common objectives, assuming own responsibility and respecting the role of the different members of the team.

Learning Outcomes

1. Analyse and evaluate computer architectures in parallel and distributed platforms.
2. Analyse and solve problems effectively, generating innovative and creative proposals to achieve objectives.
3. Analyse how applications perform, detect bottlenecks and apply possible optimisations.
4. Analyse the available parallel programming models, paradigms and languages to determine the one that best suits the requirements of an application.
5. Develop and optimise software for parallel and distributed platforms based on the most common existing paradigms.
6. Develop critical thinking to analyse alternatives and proposals, both one's own and those of others, in a well-founded and argued manner.
7. Implement and optimise applications based on the functionalities and structure of parallel, distributed and cloud systems.
8. Propose new methods or informed alternative solutions.
9. Understand and apply the fundamental principles and basic techniques of parallel, concurrent and distributed programming.
10. Use the most suitable tools and methodologies to evaluate the functionality and performance of the parallel / distributed applications developed.
11. Weigh up the risks and opportunities of both your own and others' proposals for improvement.
12. Work cooperatively to achieve common objectives, assuming own responsibility and respecting the role of the different members of the team.

Content

Topic 1: Introduction to high performance systems.

Advanced C programming and integration of C code with Python. Concurrency: concept, race conditions, critical section 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). Parallel application models.

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. Standard MPI (Message Passing Interface). Applications based on shared memory. Standard OpenMP (Open Multiprocessing), OpenACC (Open Accelerators) and CUDA (Compute Unified Device Architecture). Parallelism in Python. Application development using MPI, OpenMP, OpenACC, Cuda and Python.

Topic 5: Performance analysis

Analysis of performance of parallel systems. Evaluation of performance of parallel systems. Examples of performance evaluation tools.

Methodology

In the development of the subject, four types of teaching activities can be differentiated:

- Theoretical classes. Explanation on the board of the theoretical part of each theme of the program. The typical structure of a class of this type will be the following: in the first place an introduction will be made where the objectives of the class and the contents to be discussed will be briefly presented. Next, the contents studied will be uncovered, including narrative expositions, formal developments that provide theoretical foundations, and intercalating examples, which illustrate the application of the exposed contents. Finally, the professor will present the conclusions of the discussed contents. Throughout the course there will be continuous evaluations of the groups of topics.
- Classes of solving problems. All the topics will be accompanied by a set of problems that the student must try to solve. In this sense, and as the student progresses in and deepens in their knowledge, these problems will be gradually more complex. The seminars will be the natural forum in which the development of practical work can be discussed in common, contributing the knowledge that the student is lacking to carry it forward. The mission of the solving problems classes is to bridge between theoretical classes and classes in the laboratory, which will promote the ability to analyze and synthesise, the development of critical reasoning, and that will train the student in the resolution of problems. Those that the professor considers of greater interest or in which the students find greater difficulty will be corrected on the board. Before the beginning of each set of problems the professor may propose a list of exercises that students will have to solve.
- Classes of projects. The practical part of the theoretical subjects will be completed with sessions in the laboratory, where the student will develop several programs and will have to tryto solve aspecific problem that the student will receive at the beginning of the semester. These developed programs (projects) will have to be delivered to the professor at the indicated dates. The projects will be developed in groups of two students. The classes include 9 sessions in the laboratory, 2 hours long each, where the student will carry out the projects.
- Exercises. During the course there will be a set of basic, short and practical exercises related to theory and practical classes of problems. Each group (2 people) will solve these exercises and make the delivery with the corresponding date. The aim of this exercise is for the student to be able to solve theoretically the concepts worked on in theory and then be able to apply them to the practical laboratory case.

This approach to work is aimed at promoting active learning and developing competencies of organizational and planning skills, oral and written communication, teamwork and critical reasoning. The quality of the exercises carried out, of its presentation and of its operation will be valued especially.

The management of the subject will be done through the Virtual Campus (https://cv.uab.cat/), which will be used to view the materials, manage the lab groups, make the corresponding deliveries, see the notes, communicate with the 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 practical projects in the lab)	16	0.64	3, 1, 2, 5, 6, 7, 8, 12, 10
Problems (Solving exercices)	7	0.28	3, 4, 1, 2, 9, 6, 8, 12
Theory	12	0.48	3, 4, 1, 9, 5, 7
Type: Autonomous			
Autonomous study	45	1.8	3, 4, 2, 9, 5, 7, 10
Preparation of problems and basic exercises	15	0.6	3, 4, 2, 9, 5, 6, 7, 12
Projects preparation	40	1.6	3, 2, 5, 6, 7, 8, 12, 10

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.

3 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 practical projects in laboratory (PL)
3. Completion of practical exercises (EXER).

The Theory (T) part will be assessed with two individual partial exams throughout the course. The final grade of Theory will come from the weighted sum of the two exams (0.5 * Control 1 + 0.5 * Control 2). There will be a second chance to recover that part on the day at June we have assigned to. Parts that have not been passed in the partial theory exams may be recovered separately. The minimum grade to pass this part of Theory is > = 5. In the recovery only the failed exams can be recovered, the grade obtained cannot be raised (in the case of passing the Theory part). The maximum grade that can be obtained in recovery is 7.

The part of Practical projects (PL) will be evaluated by group. There are three deliveries. The final grade will come from the weighted sum of the three deliveries (X0 * Delivery 1 + X1 * Delivery 2 + X2 * Delivery 3, X0 + X1 + X2 = 1). There will be 3 short and written exams for individual validation of the practical projects in the class schedule (the third exam in the schedule established by the coordination for the second partial exams in June). A rating of 0 points on the validation exam will give a factor of 0.5, a rating of 5 points will give a factor of 0.75 and a rating of 10 will give a factor of 1. Practical projects (PL) = Lab * Validation. To pass the PL the minimum mark will have to be >= 5. There is only one opportunity to pass this part (this part can not be recovered).

The practical exercises (EXER) will be done in groups of 2 people and will consist of workingon very specific programming problems, short and related to those found in laboratory practices. We want each student to study the solution of a set of specific problems in isolation in the most general case of the practice. The value of these exercises is 30% of the final grade and given their nature and purpose are not recoverable.

The final grade of the subject will be the weighted sum of the grades of each of the activities: 40% of Theory, 30% Resolution of practical exercises and 30% of Resolution of practical projects. 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 Practical projects), 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 madeabout 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 (PL> = 5)) then

NF = 0.4 * T + 0.3 * EXER + 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.

Single Assessment Evaluation: This subject DOES NOT HAVE SINGLE ASSESSMENT EVALUATION.

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 mayleadto 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 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 aretranslationsor adaptations, and generally works with non-original and exclusive elements of the student;
- unauthorized AI use (i.e. Copilot, ChatGPT o equivalents);
- 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 theassessment 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
Partial individual exam 1	20%	2	0.08	3, 2, 9, 5
Partial individual exam 2	20%	2	0.08	3, 2, 9, 5
Practical exercices	30%	8	0.32	3, 2, 9, 5, 7, 12, 10
Practical project 1	10%	1	0.04	4, 1, 2, 6, 7, 11, 8, 12, 10
Practical project 2	10%	1	0.04	4, 1, 2, 6, 7, 11, 8, 12, 10
Practical project 3	10%	1	0.04	4, 1, 2, 6, 7, 11, 8, 12, 10

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

OpenMP

MPI

OpenACC/CUDA

queue manager (SLURM)

tools for performance evaluation (perf, TAU, nvcc)

remote connection to the laboratory (e.g. MobaTex, Eclipse, Visual Studio, Visual Studio Code, etc.)