Parallel Programming

Code: 43481
ECTS Credits: 6

2024/2025

Degree	Type	Year
4313136 Modelling for Science and Engineering	OB	0

Contact

Name: Anna Barbara Sikora

Email: anna.sikora@uab.cat

Teaching groups languages

You can view this information at the end of this document.

Prerequisites

There are no prerequisites

Objectives and Contextualisation

The objective of the module is to:

- Develop C programs
- Identify difficulties related to parallel programming
- Apply an adequate methodology for the development of parallel applications
- Understand the differences of parallel programming approaches: Shared memory, message passing
- Determine the most convinient parallel programming approach to develop a particular application
- Develop parallel applications
- Evaluate parallel application performance and collect necessary measurements to tune the application in order to improve its performance

At the end of this module, students should have enough knowledge, methods and technical skills to develop parallel applications using an adequate programming model and to evaluate the application performance.

Competences

- Analyse and evaluate parallel and distributed computer architectures, and develop and optimise advanced software for these.
- Analyse, synthesise, organise and plan projects in the field of study.
- Apply specific methodologies, techniques and resources to conduct research and produce innovative results in the area of specialisation.
- Continue the learning process, to a large extent autonomously.
- Look for new areas to open up within the field.
- Safeguard, manage, audit and certify the quality of advanced developments, processes, systems and software.
- Take part in research projects and working groups in the field of information engineering and high-performance computation.

- Use acquired knowledge as a basis for originality in the application of ideas, often in a research context.

Learning Outcomes

1. Analyse, synthesise, organise and plan projects in the field of study.
2. Apply specific methodologies, techniques and resources to conduct research and produce innovative results in the area of specialisation.
3. Continue the learning process, to a large extent autonomously.
4. Design the parallel solution to a computational problem taking the characteristics of the hardware available into account.
5. Develop the parallel solution to a computational problem by choosing the most appropriate tools.
6. Identify sources of parallelism in a computational problem.
7. Interpret information from performance-analysis and use this in actions that improve the parallel application.
8. Look for new areas to open up within the field.
9. Plan and develop research projects with content related to parallel programming.
10. Use acquired knowledge as a basis for originality in the application of ideas, often in a research context.
11. Use appropriate tools to analyse the performance of an application.

Content

1. Introduction to the course
2. C programming
3. Introduction to high performance systems, parallel systems, multiprocessors and multicomputers. Execution of parallel applications in high performance systems.
4. OpenMP programming - Applications based on shared memory. Standard OpenMP (Open Multiprocessing)
5. MPI programming - Applications based on message passing. Standard MPI (Message Passing Interface).
6. Performance Analysis - Analysis of performance of parallel systems. Evaluation of performance of parallel systems. Examples of performance evaluation tools.
7. GPU Programming

Activities and Methodology

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Attending theoretical lectures and practical exercises	14	0.56	2, 5, 4, 6, 9, 10
Lab practices	24	0.96	5, 4, 7, 9
Type: Autonomous			
Design and development of practical exercises	52	2.08	1, 5, 4, 6, 7, 9, 3, 10, 11
Study	36	1.44	1, 2, 5, 4, 6, 8, 7, 9, 3, 10, 11

The subject will be developed in theoretical classes and practical exercises.

It is recommended that students attend all classes of the subject with a laptop with a well-charged battery.

The approach applied 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.

Assessment

Continous Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
C lab exercise	15	6	0.24	3, 10
Final Exam	30	2	0.08	5, 4, 6, 3, 10
GPU Lab	20	6	0.24	5, 4, 6
MPI Lab exercise	20	6	0.24	1, 2, 5, 4, 6, 8, 7, 9, 3, 10, 11
OpenMP Lab exercise	15	4	0.16	1, 2, 4, 6, 8, 9

The aim of the
assessment process is to verify that the student has achieved the knowledge and skills defined in the objectives
Five types
of activities will be evaluated independently where the weighted sum of them will give the final grade. These five

1. Programming in C (15%)
2. Programming in OpenMP (15%)
3. Programming in MPI (20%)
4. Programming with GPUs (20%)
5. Individual final exam (30%)

The evaluations of the program will be
carried out in groups of 2 people. Final exam will be individual and written.

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.

Bibliography

Parallel Programming : Techniques and Applications using Networked Workstations and Parallel Computers. Barry Wilkinson. Prentice Hall, 1999.

Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering. Ian Foster . Addison Wesley,1995.

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.

High Performance Cluster Computing (Vols. 1 y 2), Rajkumar Buyya ed., Prentice Hall, 1999.

Parallel Programming with MPI, Peter Pacheco, Morgan Kauffman, 1996

Using MPI-2, William Gropp, Ewing Lusk and Rajeev Thakur, MIT Press, 1999.

Programming Massively Parallel Processors: A Hands-on Approach. David Kirk and Wen-mei Hwu. ISBN: 978-0-12-381472-2. Published by Elsevier Inc. 2010.

http://www.elsevierdirect.com/morgan_kaufmann/kirk/

Software

C

OpenMP

MPI

OpenACC

queue manager (SLURM)

performance evaluation tools (perf, TAU, nvcc)

remote connection with laboratory (e.g. MobaTex, Eclipse, Visual Studio, etc.)

Language list

Name	Group	Language	Semester	Turn
(PLABm) Practical laboratories (master)	1	English	first semester	afternoon
(PLABm) Practical laboratories (master)	2	English	first semester	afternoon
(TEm) Theory (master)	1	English	first semester	afternoon