Pensament Computacional Aplicat a la Programació Paral·lela 2012/2013

Code: 42251
ECTS Credits: 6

Degree	Syllabus	Type	Year	Semester
4313133 Còmput d'Altes Prestacions, Teoria de la Informació i Seguretat / High Performance Computing, Information Theory and Security	1094 Còmput d'Altes Prestacions, Teoria de la Informació i Seguretat / High Performance Computing, Information Theory and Security	P	1	0
4313136 Modelització per a la Ciència i l'Enginyeria / Modelling for Science and Engineering	1095 Modelització per a la Ciència i l'Enginyeria / Modelling for Science and Engineering	O	3	0

Contact

Name: Anna Bàrbara Sikora

Email: Anna.Morajko@uab.cat

Use of languages

Principal working language: anglès (eng)

Prerequisites

There are no requirements.

Objectives and Contextualisation

The objective of the module is to:

- Identify difficulties related to parallel programming
- Apply an adequate methodology for the development of parallel applications
- Evaluate tools and languages for parallel programming
- 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.

Skills

- Analyse and evaluate parallel and distributed computing architectures, as well as develop and optimise advanced software for them.
- Analyse, synthesise, organise and plan projects related to information theory, security and high performance computing.
- Apply the methodology of research, techniques and specific resources for investigating and producing innovative results in a certain specialised field.
- Assure, guarantee, manage, certify and investigate the quality of advanced computing developments, processes, systems and products.
- Direct innovation and research projects and work teams in the area of information theory, security and high performance computing.
- Innovate in the search for new spaces / areas in one's field of work.
- Possess and comprehend knowledge that offers the basis and opportunity to be original in the development and/or application of ideas, frequently in a research context.

- Recognise the human, economic, legal and ethical dimensions of professional exercise.
- Students must possess learning abilities to enable them to continue studying in a way that will to a large extent have to be self-managed and autonomous.

Learning outcomes

1. Analyse, synthesise, organise and plan projects related with information theory, security and high performance computing
2. Apply the methodology of specific research, techniques and resources for investigating and producing innovative results in a certain specialised field
3. Design the parallel solution to a computing problem taking into consideration the characteristics of the hardware available.
4. Develop the parallel solution for a computing problem by choosing the most adequate tools
5. Identify sources of parallelism in a computing problem
6. Innovate in the search for new spaces / areas in one's field of work
7. Interpret the information provided by performance analysis tools and transform it into actions that improve the parallel application
8. Plan and develop innovation and research projects with content related to parallel programming
9. Possess and comprehend knowledge that offers the basis and opportunity to be original in the development and/or application of ideas, frequently in a research context
10. Recognise the human, economic, legal and ethical dimensions of professional exercise
11. Students must possess learning abilities to enable them to continue studying in a way that will to a large extent have to be self-managed and autonomous
12. Use the right tools for analysing the performance of an application

Content

Introduction to the course

- Course Content, Model & Labs

MPI programming

- Introduction to MPI programming (1 session)
- C+MPI programming - practice - laboratory (3 sessions)

OpenMP programming

- Introduction to OpenMP programming (1 session)
- OpenMP programming - practice - laboratory (1 session)

Algorithms

- Parallel algorithms (1 session)

CUDA

- Introduction to CUDA (2 sessions)
- CUDA programming - practice - laboratory (1 session)

Performance analysis tools and performance models

- Overview of performance analysis tools (TAU, Scalasca, Periscope, Paraver, MATE) (1 session)
- Performance models: locality, number of resources, load balancing (1 session)

Final work

- Final practical work - laboratory (1 session)
- Final practical work - presentation (1 session)

Invited speaker (1 session)

Methodology

The methodology will combine master classes, laboratories, independent and assisted work of the students, delivery of partial practices, and the presentation of a small project where students apply the received knowledge.

Distribution of the tasks:

- Attending lectures and laboratories: 30%
- Guided learning activities (outside classroom): 10%
- Learning self-activities (outside classroom): 60% presented/displayed.

Activities

Title	Hours	ECTS	Learning outcomes
Type: Directed			
Classes	30	1.2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
Type: Autonomous			
Studying	85	3.4	1, 2, 3, 4, 5, 7, 8, 9
Studying in group	30	1.2	1, 3, 4, 7, 11, 12

Evaluation

Practices:

- Development of two MPI basic programs during the classes of C+MPI laboratory. The practice may be developed by groups of two students.
- Development of one OpenMP basic program during the classes of OpenMP laboratory. The objective is to study OpenMP and analyze differences in programming model. The practice may be developed by groups of two students.
- Development of one CUDA basic program during the classes of CUDA laboratory. The objective is to introduce CUDA and to analyze differences in programming model. The practice may be developed by groups of two students.
- Development of a final project choosing between MPI, OpenMP or CUDA. The objective is to study a problem with high performance computing requirements, its parallelization, scalability, and performance evaluation. The project is developed individually and it contains a report of the given problem and results. Each student must present the final project results during the last class (presentations of 15 minutes).

Evaluation:

- Final individual project delivery and presentation of the obtained results
- Partial practices delivered by groups of students during the semester.
- Attending to lectures (min 80%) and active participation

There will be a meeting with all responsible teachers and the coordinator of the module to decide the final evaluation.

Evaluation activities

Title	Weighting	Hours	ECTS	Learning outcomes
CUDA application	20	1	0.04	1, 3, 4, 5, 7, 12
Final project	40	1	0.04	1, 2, 3, 4, 5, 6, 7, 8, 12
MPI application 1	10	1	0.04	1, 3, 4, 5, 12
MPI application 2	10	1	0.04	1, 3, 4, 5, 7, 12
OpenMPI application	20	1	0.04	1, 3, 4, 5, 7, 12

Bibliography

1. Parallel Programming : Techniques and Applications using Networked Workstations and Parallel Computers. Barry Wilkinson. Prentice Hall, 1999.
2. Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering. Ian Foster . Addison Wesley,1995.
3. Introduction to Parallel Computing. A. Grama et alter. Addison Wesley, Second Edition, 2003.
4. 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.
5. High Performance Cluster Computing (Vols. 1 y 2), Rajkumar Buyya ed., Prentice Hall, 1999.
6. Parallel Programming with MPI, Peter Pacheco, Morgan Kauffman, 1996
7. Using MPI-2, William Gropp, Ewing Lusk and Rajeev Thakur, MIT Press, 1999.
8. 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.
9. http://www.elsevierdirect.com/morgan_kaufmann/kirk/