COURSE PROGRAM

Academic Year 2024-2025

Identification and characteristics of the course				
Code	401081	ECTS Credits	6	
Name	High Performance Computing			
Master	Master in Informatics Engineering			
Centre	School of Technology			
Semester	2	Character	Compulsory Studies	
Module	Informatics Technologies			
Material	Advanced Informatics Technologies			
Lecturers				
Name		Office	e-mail	Web page
Miguel Ángel Vega Rodríguez		ARCO	mavega@unex.es	http://arco.unex.es/mavega
Subject Area	Computer Architecture and Technology			
Department	Computer and Communication Technology			
Coordinating lecturer	Miguel Ángel Vega Rodríguez			

Competencies
BASIC COMPETENCIES
CB6: Knowledge and understanding that provide a basis or opportunity for originality in developing and/or applying ideas, often in a research context.
CB9: Skills to communicate conclusions, and the knowledge and rationale underpinning these, to specialized and non-specialized audiences in a clear and unambiguous manner.
CB10: Learning skills that enable to continue studying in a way that will be largely self-directed or autonomous.
GENERAL COMPETENCIES
CG4: Capacity for mathematical modelling, calculation and simulation in technological centres and engineering of company, particularly in research, development and innovation tasks in all fields related to Informatics Engineering.
CG8: Ability to apply acquired knowledge and solve problems in new or little-known environments within broader and multi-disciplinary contexts, being able to integrate this knowledge.
SPECIFIC COMPETENCIES
CETI07: Ability to understand and be able to apply advanced knowledge of high performance computing and numerical or computational methods to engineering problems.
CROSS-CURRICULAR COMPETENCIES
CT11: Autonomous learning capacity.

Themes and syllabus
Brief description of the syllabus
Evolution and state-of-the-art in high performance computing. High Performance Computing (HPC). High Throughput Computing (HTC). Computer architectures for HPC and HTC. Differences and similarities between HPC and HTC. Supercomputing and grid computing: Concepts, necessity, and applications. Cluster computing. Grid technologies and infrastructures.

Course syllabus
Theme 1: Supercomputing and Grid Computing 1.1. Computing: Necessities. Physical limits. Future technologies. 1.2. Classification of parallel computer architectures. 1.3. Multiprocessors: Concept of shared memory. UMA. NUMA. COMA. 1.4. Multicomputers: Concept of distributed memory. MPP. COW. Beowulf Cluster. 1.5. High performance computing: HPC vs. HTC. 1.6. Supercomputing (HPC): Some metrics of parallelism. Parallel programming. More powerful computers. Hardware accelerators. 1.7. Grid Computing (HTC): Concept. Organization by layers. Structure. Programming. Examples. Projects. Network. Use. 1.8. Spanish network of e-science. IRIS network. Spanish network of supercomputing. 1.9. Desktop Grid Computing or Institutional Computing: Concept. Alternatives. 1.10. Volunteer Computing. 1.11. BOINC: Concept. Some projects. 1.12. Parallel architectures: Last advances. 1.13. Distributed architectures: Last advances. Cloud computing. Practical activities: Parallel programming with OpenMP.
Theme 2: Supercomputing in Extremadura *(This theme will be taught by CénitS)* 2.1. Extremadura Supercomputing, Technological Innovation and Research Center (CénitS). 2.2. Architecture, technical characteristics and singularities of the supercomputers LUSITANIA, LUSITANIA II, LUSITANIA III, and other compute equipments of Cénits. *(Sections 2.1 y 2.2 will be included in a visit to CénitS, where the data center will be presented)* 2.3. Administration tools, scheduling, tuning and monitoring of a supercomputer. 2.4. Example of code parallelization for supercomputing and application of benchmarks. 2.5. Success cases and projects of supercomputing in Extremadura. Practical activities: Parallel programming with MPI (C++ and Python).
Theme 3: Computing for Problem Optimization 3.1. Algorithms and optimization. 3.2. Evolutionary computation. Application domains. 3.3. Genetic algorithms. Basics. 3.4. Developing optimization algorithms. 3.5. Application examples. Operators for specific problems. 3.6. Parallelism and problem optimization. Practical activities: Swarm intelligence – Artificial Bee Colony.
Theme 4: Project about the Contents in the Previous Themes 4.1. Selecting the evolutionary algorithm and understanding its pseudocode and operation. 4.2. Implementing the evolutionary algorithm to solve a certain problem. 4.3. Parallelizing the implemented evolutionary algorithm. 4.4. Evaluating and comparing the result quality and runtime of the implemented parallel evolutionary algorithm. Practical activities: Programming a parallel evolutionary algorithm.

Theme planning															
	Week														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	February				March				April				May		
Themes 1	■	■	■	■	■										Evaluation
Themes 2						■	■								
Themes 3								■	■	■					
Themes 4											■	■	■	■	

Educational activities

Student hours of work per theme		Classroom	Practical Activities				Monitoring activity	Non-classroom
Theme	**Total**	**LG**	**HCP**	**LAB**	**COM**	**SEM**	**PT**	**PS**
1	45	15	0	5	0	0	0	25
2	18	6	0	2	0	0	0	10
3	27	9	0	3	0	0	0	15
4	36	12	0	4	0	0	0	20
Evaluation of the whole	24	3	0	1	0	0	0	20
TOTAL	150	45	0	15	0	0	0	90

LG: Large group (100 students).
HCP: Hospital clinic practical activities (7 students).
LAB: Practical activities in laboratory or field (15 students).
COM: Practical activities in computer or language laboratory (30 students).
SEM: Problem activities or seminars or practical cases (40 students).
PT: Programmed tutorials (educational monitoring, in the form of ECTS tutorials).
PS: Personal study, individual or group tasks, and reading the literature.

Educational methodologies

Theoretical-practical classrooms

Different activities will be used in the classroom, led to the whole group or small groups. Mainly, expository classes will be performed for the development of the fundamental contents of the course. In order to get the student active participation, individual or group short activities will be performed, allowing the application of the exposed concepts and the problem solving. We will propose activities led to the knowledge application in solving typical problems in the field of High Performance Computing.

Laboratory and/or seminar sessions

Practical activities, guided laboratory sessions, seminars for problem solving, etc. will be performed in groups, under the supervision of the lecturer. Previous and subsequent activities to the laboratory and seminar sessions could be included in order to help to reach the proposed objectives. The proposed activities will be close, as much as possible, to the real activities of the Informatics Engineer during his/her professional work.

Non-classroom individual work and study

Activities, works, and study performed by the student, in an autonomous way, individually or in groups.

We will encourage non-classroom tasks similar to the ones performed by an Engineer.

Learning outcomes

- Know the state-of-the-art environments in high performance computing, both for HPC (High Performance Computing) and for HTC (High Throughput Computing).
- Comprehend the basics of computer architectures for HPC and HTC.
- Understand the fundamental concepts about supercomputing and grid computing.
- Program clusters and distributed architectures.
- Know how to speedup and make efficient the execution of compute-intensive applications, using hardware techniques in which the processors can work in a parallel or distributed way, at different levels.

Evaluation systems

The course offers **2 different evaluation itineraries**:

Itinerary A: A **continuous assessment** system is proposed, which will take into account the attendance and active participation in the theoretical classes and seminars/laboratories, the task performance, the class expositions, and the final exam. Following this evaluation itinerary, the student will obtain a grade (from 0 to 10) per every theme. The **final grade** (FG) will be the **arithmetic mean** of the **grades obtained in the different themes**, **as long as all the themes are passed** by following the continuous assessment:

FG = (Theme1_Grade + Theme2_Grade + Theme3_Grade + Theme4_Grade) / 4

Itinerary B: Students that have not passed or do not want to follow the continuous assessment will have to present to the **final exam (global final alternative exam)** in the corresponding dates of the official exams. The final exam will include questions from all the themes of the course. In this case, the **final grade** will be the **grade obtained in this final exam**.

(Basic and complementary) bibliography

- Material given by the lecturer.
- Bibliographic references:

    - Distributed and Parallel Systems: From Cluster to Grid Computing. Peter Kacsuk, Thomas Fahringer & Zsolt Nemeth. Springer.
    - The Sourcebook of Parallel Computing. Jack Dongarra, Ian Foster, Geoffrey C. Fox, William Gropp, Ken Kennedy, Linda Torczon & Andy White. Morgan Kaufmann.
    - Distributed Computing: Principles, Algorithms, and Systems. Ajay D. Kshemkalyani & Mukesh Singhal. Cambridge University Press.
    - Cluster Computing. Rajkumar Buyya & Clemens Szyperski. Nova Science Publishers.
    - The Grid 2: Blueprint for a New Computing Infrastructure. Ian Foster & Carl Kesselman. Morgan Kaufmann, 2nd edition.
    - Efficient and Accurate Parallel Genetic Algorithms. Erik Cantú-Paz. Kluwer Academic Publishers.
    - Parallel Metaheuristics: A New Class of Algorithms. Enrique Alba. Wiley.

Other resources and complementary educational materials
• Virtual classroom (download of materials, forum, news, etc.): https://campusvirtual.unex.es/zonauex/avuex/course/view.php?id=10053 • TOP500 Supercomputer List: https://www.top500.org • Green500 Energy-Efficient Supercomputer List: https://www.top500.org/lists/green500/ • Extremadura Supercomputing, Technological Innovation and Research Center (CénitS): http://www.cenits.es • Extremadura Research Center for Advanced Technologies (CETA-CIEMAT): https://www.ceta-ciemat.es • GAs website: https://www.obitko.com/tutorials/genetic-algorithms • Evolutionary Computation Bestiary: https://github.com/fcampelo/EC-Bestiary • Websites of the different books recommended in the course.