Teaching Guide					
Identifying Data					2021/22
Subject (*)	HPC Tools			Code	614473105
Study programme	Mestrado Universitario en Computación de Altas Prestacións / High Performance Computing (Mod. Presencial)				
Descriptors					
Cycle	Period		Year	Type	Credits
Official Master's Degree	1st four-month period		First	Optional	6
Language	English				
Teaching method	Hybrid				
Prerequisites					
Department	Enxeñaría de Computadores				
Coordinador	Padron Gonzalez, Emilio Jose		E-mail	emilio.padron@udc.es	
Lecturers	Andrade Canosa, Diego Padron Gonzalez, Emilio Jose		E-mail	diego.andrade@udc.es emilio.padron@udc.es	
Web	aula.cesga.es				
General description	The objective of this course is to get the students familiar with the most common types of application that are candidates to use HPC, besides being introduced to the main tools and implementations existing for them, understanding the challenges to be addressed for their parallelization and performance tuning. All this will allow the students to obtain a general knowledge about the HPC field and its different applications and use cases. Furthermore, the students will learn what tools can be used to carry out the performance characterization and benchmarking tasks in HPC environments, and how these tools can be leveraged to drive the parallelization and performance tuning of an application on a specific platform. This will allow the students to be able to analyze the expected performance on that system, identifying the different hot spots and focussing the optimization efforts on them. Finally, the students will learn different technological alternatives for a fast and efficient deployment of HPC applications. This will allow them to be able to easily and effectively deliver and execute HPC applications in different environments.				

Contingency plan	1. Modifications to the contents - None 2. Methodologies *Teaching methodologies that are maintained - The teaching methodologies used in this course are maintained, but changing the teaching method from "Blended" (hybrid face-to-face/by-distance) to "By-distance". *Teaching methodologies that are modified - None, only the teaching method is modified: blended -> by-distance 3. Mechanisms for personalized attention to students - The previously planned but limiting communication channels to e-mail and the UDC's Teams tool 4. Modifications in the evaluation - None, the evaluation is already online in this course *Evaluation observations: 5. Modifications to the bibliography or webgraphy - None

Study programme competences / results	
Code	Study programme competences / results

Learning outcomes			
Learning outcomes	Study programme competences / results		
Students will know the most common types of applications in which HPC techniques are usually applied.			
Students will learn to use tools to characterize and represent the performance of applications.			
Students will learn to use tools to compile, generate and deploy software in HPC environments.			

Contents	
Topic	Sub-topic
A survey of main application types in HPC. For each type we?ll see:	1. Problem: formal description. 2. Parallelization and performance tuning challenges. 3. Existing approaches.
Tools to measure, characterize and represent the performance of HPC applications.	1. Usage of performance characterization and benchmarking tools, such as software monitoring and hardware counters. 2. Hot spot detection to drive the optimization process. 3. Application of performance models to this process. 4. Tools for application performance representation.

Tools for the compilation, generation and deployment of HPC software.	1. Code compilation, optimization and generation in a compiler. 2. Code optimization with a compiler. 3. Automatic parallelization and vectorization. 4. Software development tools. 5. Leveraging containers for the easy deployment of HPC applications.

Planning				
Methodologies / tests	Competencies / Results	Teaching hours (in-person & virtual)	Student?s personal work hours	Total hours
Guest lecture / keynote speech	A3 B1 C4	23	0	23
Laboratory practice	A1 A2 A4 A5 C1	18	52	70
Supervised projects	B3 B4 B6 B8 B9	0	54	54
Mixed objective/subjective test	B4 B6	2	0	2
Personalized attention		1	0	1
(*)The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.				

Methodologies	
Methodologies	Description
Guest lecture / keynote speech	Lectures, discussing the different lessons of the course. Students will have available all the necessary material in advance and the teacher will promote an active attitude in the classroom, asking questions that may clarify specific aspects and leaving open issues for student reflection.
Laboratory practice	Lab sessions, allowing the students to become familiar from a practical standpoint with the issues discussed in the lectures.
Supervised projects	Guided task fulfillment: students apply the acquired knowledge to solve different problems autonomously.
Mixed objective/subjective test	Written test/exam to show that the students have acquired the Degree's competences trained in this course by answering theoretical questions and solving exercises.

Personalized attention	
Methodologies	Description
Laboratory practice Supervised projects	Personalized attention is guaranteed during the development of the laboratory practices and supervised projects, being essential to guide students in the fulfillment of their tasks. This personalized attention is also useful to validate and evaluate the work carried out throughout the different development stages, until finished. Furthermore, it is recommended for students to leverage the teacher's office hours as a complementary assistance tool.

Assessment			
Methodologies	Competencies / Results	Description	**Qualification**
Mixed objective/subjective test	B4 B6	Written test/exam to show that the students have acquired the Degree's competences trained in this course by answering theoretical questions and solving exercises.	30
Supervised projects	B3 B4 B6 B8 B9	Guided task fulfillment: students apply the acquired knowledge to solve different problems autonomously.	70

Assessment comments

Sources of information

Basic	[1] Computer Architecture: A Quantitative Approach (5th or 6th Ed.). John L. Hennessy, David A. Patterson. Morgan Kaufmann. ISBN 978-0123838728 (5th Ed. 2011) 978-0128119051 (6th Ed. 2017)[2] Performance Tuning of Scientific Applications. David H. Bailey, Robert F. Lucas, Samuel Williams. CRC Press. ISBN 978-1439815694[1] Computer Architecture: A Quantitative Approach (5th or 6th Ed.). John L. Hennessy, David A. Patterson. Morgan Kaufmann. ISBN 978-0123838728 (5th Ed. 2011) 978-0128119051 (6th Ed. 2017)[2] Performance Tuning of Scientific Applications. David H. Bailey, Robert F. Lucas, Samuel Williams. CRC Press. ISBN 978-1439815694
Complementary	[3] Intel® C++ Compiler Developer Guide and Reference https://software.intel.com/cpp-compiler-developer-guide-and-reference[4] A Guide to Vectorization with Intel® C++ Compilers https://software.intel.com/sites/default/files/m/4/8/8/2/a/31848-CompilerAutovectorizationGuide.pdf[5] Intel® VTune? Amplifier Help https://software.intel.com/en-us/vtune-amplifier-help[6] Free Software Foundation, Inc.: Using the GNU Compiler Collection (GCC). https://gcc.gnu.org/onlinedocs

Recommendations
Subjects that it is recommended to have taken before
Parallel Programming/614473102
Subjects that are recommended to be taken simultaneously
Subjects that continue the syllabus
Other comments
Because of the strong interrelation between the lectures and the lab sessions, and the progressive presentation of concepts very related each other in the lectures, it is recommended to dedicate enough time to a daily study or review.This course will leverage online communication tools in quite an intensive way: videoconference, e-mail, chat, etc.