

## COURSE DESCRIPTION – ACADEMIC YEAR 2024/2025

<b>Course title</b>	<b>Parallel Computing</b>
<b>Course code</b>	76085
<b>Scientific sector</b>	INF/01
<b>Degree</b>	Master in Software Engineering (LM-18)
<b>Semester</b>	1
<b>Year</b>	1
<b>Credits</b>	6
<b>Modular</b>	No
<b>Total lecturing hours</b>	40
<b>Total exercise hours</b>	20
<b>Attendance</b>	Attendance is not compulsory, but strongly recommended. Students who are unable to follow all lectures and labs are encouraged to interact with the lecturer.
<b>Prerequisites</b>	Good knowledge of the following subjects is expected: <ul style="list-style-type: none"> <li>• Programming</li> <li>• Computer Systems</li> <li>• Algorithms and Data Structures</li> </ul>
<b>Course page</b>	<a href="https://ole.unibz.it/">https://ole.unibz.it/</a>
<b>Specific educational objectives</b>	<p>The course belongs to the type “attività formativa affine o integrativa”</p> <p>Students will acquire a deep knowledge of how to design faster and efficient applications by exploiting modern parallel architectures, from local multi-core CPUs and GPUs, to distributed memory HPC systems. Under such a light, students will acquire professional skills and knowledge in parallel computing by understanding basic and advanced techniques that researchers have developed in the last years.</p>
<b>Lecturer</b>	<a href="#">Maurizio Tavelli</a>
<b>Contact LA</b>	mtavelli@unibz.it
<b>Scientific sector of lecturer</b>	MAT/08
<b>Teaching language</b>	English
<b>Office hours</b>	During the course (12.00-13.00), to arrange beforehand by email.
<b>Lecturing Assistant (if any)</b>	--
<b>Contact LA</b>	--
<b>Office hours LA</b>	--
<b>List of topics</b>	<ul style="list-style-type: none"> <li>• Introduction to architectures for parallel and distributed systems</li> <li>• Shared memory model and GPU Computing</li> <li>• Distributed memory model: introduction to Message Passing Interface</li> <li>• Principle and design of parallel algorithms</li> <li>• Selection of parallel algorithms</li> <li>• Performance Analysis, optimization and tuning</li> </ul>
<b>Teaching format</b>	<ul style="list-style-type: none"> <li>• Frontal lectures</li> <li>• Lab supported by the lecturer and Teaching Assistant (if any)</li> </ul>

	<p>In the lectures, new concepts and techniques are introduced by presentation on the blackboard and multimedia material (slides and videos).</p> <p>In the labs, students will:</p> <ol style="list-style-type: none"> <li>1. use the tools that will be introduced during the course and project development;</li> <li>2. solve simple exercise and discuss the solution;</li> <li>3. start to prepare the final project;</li> </ol> <p>supported by the lecturer or Teaching Assistant (if any).</p>
--	---

<p><b>Learning outcomes</b></p>	<p><b>Knowledge and understanding:</b></p> <ul style="list-style-type: none"> <li>• D1.2 To be able to analyse and solve even complex problems in the area of Software Engineering for Information Systems with particular emphasis on the use of empirical evaluation studies, methods, techniques and technologies;</li> <li>• D1.3 To know in depth the scientific method of investigation applied to complex systems and innovative technologies that support information technology and its applications;</li> <li>• D1.8 To be able to read and understand specialist scientific documentation, such as conference proceedings, articles in scientific journals, technical manuals.</li> </ul> <p><b>Applying knowledge and understanding:</b></p> <ul style="list-style-type: none"> <li>• D2.1 To know how to apply the fundamentals of empirical analysis of ICT data to the construction of mathematical models for the evaluation and prediction of characteristics of applications and software systems;</li> <li>• D2.2 To be able to design and perform experimental analyses of information systems in order to acquire measures related to their behaviour and to evaluate experimental hypotheses in different fields of application, such as business, industrial or research;</li> <li>• D2.5 To be able to extend and modify in an original way an existing technical solution or a formal model taking into account changed conditions, requirements and evolution of the technology.</li> </ul> <p><b>Making judgments:</b></p> <ul style="list-style-type: none"> <li>• D3.1 To be able to autonomously select documentation from a variety of sources, including technical books, digital libraries, technical scientific journals, web portals or open source software and hardware tools;</li> <li>• D3.4 To be able to reconcile the objectives of the project that are in conflict, to trade-off cost, resources, time, knowledge or risk;</li> <li>• D3.5 To be able to work with large autonomy, also assuming responsibility for projects and structures.</li> </ul> <p><b>Communication skills:</b></p> <ul style="list-style-type: none"> <li>• D4.1 To be able to present the contents of a scientific/technical report to an audience, including non-specialists, at a fixed time;</li> <li>• D4.2 To be able to structure and draft scientific and technical documentation describing project activities;</li> </ul>
---------------------------------	--

	<ul style="list-style-type: none"> <li>• D4.3 To be able to coordinate project teams and to identify activities to achieve project objectives;</li> <li>• D4.4 To be able to prepare and conduct technical presentations in English;</li> <li>• D4.6 To be able to carry out research and projects in collaborative manner;</li> <li>• D4.7 To be able to synthesise knowledge gained from reading and studying scientific documentation.</li> </ul> <p><b>Learning skills:</b></p> <ul style="list-style-type: none"> <li>• D5.1 To be able to independently extend the knowledge acquired during the course of study by reading and understanding scientific and technical documentation in English;</li> <li>• D5.3 In the context of a problem solving activity, to be able to extend knowledge, even if incomplete, taking into account the final objective of the project;</li> <li>• D5.4 To be able to formulate and validate theories and define new methods through empirical induction and new generation scientific investigation tools.</li> </ul>
--	---

<b>Assessment</b>	<p>The assessment is based on a final project that will be assigned during the semester. It will consist of solving a particular problem by using parallel computing techniques.</p> <p>The project will be developed by a group of two students (at most). Specifically, the team has to:</p> <ul style="list-style-type: none"> <li>• release the code by providing the instruction for result reproducibility;</li> <li>• write a short scientific document describing the solution, the methodology, the technology they adopted for solving the problem;</li> <li>• prepare a short oral presentation of the project. After the presentation will follow a Q&amp;A session to assess the knowledge of the candidate and its contribution to the project.</li> </ul> <p>The assessment is based on the <b>individual</b> contribution of each team member.</p>
<b>Assessment language</b>	English
<b>Assessment typology</b>	Monocratic
<b>Evaluation criteria and criteria for awarding marks</b>	<p>The final mark is composed by evaluating the project in terms of originality of the methods adopted/designed, results obtained and quality and clarity of the presentation which includes code, document and oral presentation. Specifically,</p> <ul style="list-style-type: none"> <li>• Code assessment (20% of the final mark). The software should follow the best practice for code writing. The experiments must be replicable.</li> </ul>

	<ul style="list-style-type: none"> <li>• Document assessment (40% of the final mark). Ability to introduce a problem. Ability to report the state of the art. Ability to describe the methodology adopted. Ability to comment and report the results obtained. The originality and the soundness of the solution will be also considered in the evaluation.</li> <li>• Quality and clarity of the oral presentation in a fixed time (40% of the final mark). Ability to answer to possible questions to the project and the related topics addressed during the course that can arise during the presentation.</li> </ul>
<p><b>Required readings</b></p>	<ul style="list-style-type: none"> <li>• V. Kumar, A. Grama, A. Gupta, G. Karypis, Introduction to Parallel Computing: design and analysis of Algorithms, Addison-Wesley, 2003;</li> <li>• Lecture notes by the teacher.</li> </ul> <p>Subject Librarian: David Gebhardi, <a href="mailto:David.Gebhardi@unibz.it">David.Gebhardi@unibz.it</a></p>
<p><b>Supplementary readings</b></p>	<ul style="list-style-type: none"> <li>• D.P. Bertsekas, J. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Trentine-Hall, 1989.</li> </ul>
<p><b>Software used</b></p>	<p>Programming languages: Fortran and Matlab (will be introduced during the course if necessary)</p> <p>Other software/frameworks: OpenMP, Intel MPI, CUDA-Fortran, oneAPI HPC Toolkit, NVIDIA HPC SDK</p>