UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

# COURSE SYLLABUS

# Parallel Computing

**2627-2-F1802Q137**

## Aims

The student understands the computational paradigms of parallel applications. To complete the analysis of these applications, performance metrics are introduced and used in the exercise activities on parallel machines in OpemMP, MPI and CUDA with the algorithms presented in class and developed by the students.
Following the Dublin indicators, students develop:

1. Knowledge and understanding - the principles on which parallel computing systems are based are illustrated and discussed;
2. Applied knowledge and understanding: students apply the principles in real cases, with practical exercises and specific projects;
3. Autonomy of judgment: the tools and metrics to develop parallel computing are provided;
4. Communication skills: the specific terms of the discipline are introduced and discussed, the final exam consists in exploring and illustrating particular topics of the course;
5. Learning skills: students explore the topics related to parallel computing with targeted projects, carried out individually or in groups.

## Contents

The course presents a comprehensive overview of parallel architectures and associated computational paradigms. It also introduces the student to the design and implementation of parallel applications in distributed environments and GPGPU with CUDA development environment emphasis. It is introduced the problem of distributed applications with access to large amounts of data, analyzing the Hadoop framework.

## Detailed program

1. Basic considerations and performance measurement.

2. The classification of parallel architectures and computational paradigms, with SIMD and MIMD architectures deepening."

3. Types of interconnection and communication patterns.

4. Examples of shared-memory platforms with references to multicore architectures.

5. Examples of distributed platform: PC clusters / networks of workstations and local and dedicated systems with graphics accelerators.

6. Parallelization techniques: types of decomposition, mapping, load balancing and optimization techniques.

7. Inhibitors of parallelization

8. Basic communication operations"

9. Analytical modelling of parallel programs

10. Overview on some relevant parallel algorithms"

11. Programming on shared memory platforms, introduction to OpenMP

12. Use of libraries Message Passing Interface (MPI) as a programming environment for the realization of parallel algorithms on multiprocessor architectures: initialization, global operations, modularity

13. The architecture of graphics accelerators and related computational paradigms

14. Introduction to the programming on graphics accelerators: CUDA environment."

15. Parallel Filesystems

16. Presentation of the Hadoop framework: the model Map-Reduce

## Prerequisites

Knowledge of computer architecture; elements of networking; programming languages, with particular regard to the C / C + +/Java languages; algorithms and models of computability and complexity.

## Teaching form

Lesson, tutorial, seminars, individual study activities supported by teaching materials in e-learning, and project activity developed in a parallel environment, especially on systems with GP-GPU accelerators from Nvidia and CUDA environment. The teaching is held in Italian, the support material is in English. The teaching is held in Italian, at the request of the students it can be delivered in English; the support material is partially in Italian and English.

## Textbook and teaching resource

1. R. Ansorge, Programming in Parallel with CUDA: A Practical Guideg, Elsevier, Cambridge University Press, 2022.

2. A. Grama, A. Gupta, G. Karypis, V. Kumar.Introduction to Parallel Computing, 2°Ed., Addison-Wesley, 2003.

3. CUDA C++ Programming Guide - Design Guide, v11.2, NVIDIA docs, 2021

4. T. White, Hadoop: The Definitive Guide, O'Reilly, 2012.

Teaching resources available on the e-learning platform of the course: lessons, articles, reference exercises.

## Semester

Second Semester 2025/2026

## Assessment method

Oral esamination. The assessment includes seminar activities by the students and the discussion of a project on a particular topic addressed during the course, with the development of code and analysis of performance on parallel architectures. The vote is determined as follows: 40% seminar activity, 60% project work.

## Office hours

By reservation, by student request via email or via the messaging on the e-learning platform.

## Sustainable Development Goals

QUALITY EDUCATION | INDUSTRY, INNOVATION AND INFRASTRUCTURE