



UNIVERSITÀ DI PISA

## PARALLEL AND DISTRIBUTED SYSTEMS: PARADIGMS AND MODELS

MARCO DANELUTTO

Anno accademico

2022/23

CdS

INFORMATICA E NETWORKING

Codice

305AA

CFU

9

Moduli	Settore/i	Tipo	Ore	Docente/i
SISTEMI DISTRIBUITI: PARADIGMI E MODELLI	INF/01	LEZIONI	72	MARCO DANELUTTO PATRIZIO DAZZI

### Obiettivi di apprendimento

#### Conoscenze

The course aims at providing a mix of foundations and advanced knowledge in the field of parallel computing specifically targeting data intensive applications. A first part of the course will provide the necessary background related to the parallel hardware, from multicore to accelerators up to distributed systems such as clusters and cloud. Then the principles of parallel computing will be addressed, including measures characterizing parallel computations, mechanisms and policies supporting parallel computing and typical data intensive patterns. Eventually a survey of existing programming frameworks will be included, aimed at preparing the students to use and exploit the more modern and advanced framework currently used in both research and production institutions. As a result, the student attending the course will be given a general perspective of the parallel computing area as well as a comprehensive survey of the currently available frameworks for data intensive computing. The whole set of arguments will be complemented with practical exercises, in class—according to the *bring your own device* principle—or as homework assignments. The correct and timely production of assignment solutions will constitute the final written essay needed to access the oral part of the exams. The student may choose to prepare a more consistent final project as substitute of the assignment exercises. The different programming frameworks used in the course will be introduced detailing the main features and usage patterns, leaving to the student the task of learning the low level syntactic details (under the supervision of the professors) as part of the homework assignments. The student successfully completing the course will be able to design and implement parallel applications/run time supports efficiently using/implementing common and useful parallel design patterns using both structured and more classical parallel programming frameworks. In particular, the student will be able to evaluate different, alternative parallelisation strategies and techniques by exploiting proper abstract and concrete parallel pattern performance models and to implement the different parallelisation strategies with state of the art parallel programming tools.

#### Modalità di verifica delle conoscenze

Proper instruction of the student will be verified through two different ways:

- continuous evaluation of short programming assignments
- final exam

During the course compact assignments will be given (with deadlines). The student successfully completing all the assignments will be able to seamlessly complete the final project assignment.

The final exam consists in a project (assigned before the end of the lessons and valid for the whole academic year) and an oral exam, including the discussion of the project.

The project that has to be implemented using one of the frameworks presented during the lessons and validated through experiments run on the parallel/distributed hardware provided.

The oral exam will start with a discussion of the project results and will conclude with questions related to the material presented during the lessons.

The student should demonstrate his/her ability in the implementation of simple parallel application or simple run time supports for structured parallel patterns through the implementation of a project assigned right before the end of the course lessons. During the oral exam, the student should demonstrate his/her ability to make limited modification to the project code and to demonstrate project code functionality and performance. During the oral exam, the student should demonstrate to have acquired suitable knowledge relatively to the course main topics and to be able to apply the techniques and methodologies covered in the course material to common parallel programming scenarios.

Methods:

- Final oral exam
- Final laboratory practical demonstration
- Laboratory practical
- Written report

Further information:



## UNIVERSITÀ DI PISA

50% lab project 50% oral exam

---

### *Capacità*

The student will eventually be able to design, develop and tune parallel and distributed applications using several state-of-the-art programming frameworks for shared memory and distributed memory architectures, mainly based on C/C++ and/or scripting languages or DSLs.

### *Modalità di verifica delle capacità*

Homework programming exercises are assigned during the course and part of the exam consist in the preparation of a small project, to be designed, developed and tested on the computing resources available for the course support.

### *Comportamenti*

Students will be educated to individual contribution to existing software projects.

### *Modalità di verifica dei comportamenti*

Part of the assignments and of the final project may be related to proper usage of and interaction with existing software tools.

### *Prerequisiti (conoscenze iniziali)*

Basic programming ability is required in C/C++ and/or Java.

### *Indicazioni metodologiche*

Delivery: face to face, with the support of computer based tools (software blackboard, audio/video of lessons available through web)

Learning activities:

- attending lectures
- individual study
- Laboratory work
- Practical

Attendance: Not mandatory

Teaching methods:

- Lectures
- laboratory (bring your own device model)
- project work

Delivery: face to face

Attendance: Not mandatory

### *Programma (contenuti dell'insegnamento)*

The main course deals with a set of arguments related to the programming models targeting parallel and/or distributed architectures. The common background of these arguments is represented by structured parallel programming models, such as those based on algorithmic skeletons or on parallel design patterns. More in detail, the course covers:

- Parallel and distributed programming principles,
- Structured parallel/distributed programming patterns (data and stream parallel),
- Parallel/distributed pattern implementation techniques for multi/many cores and cluster/networks of workstations,
- Parallel/distributed patterns refactoring,
- Autonomic management of non functional features.
- Analytical, concrete and abstract performance models for parallel/distributed patterns,
- Implementation of parallel/distributed patterns with state of the art parallel programming frameworks.
- Concise introduction to the usage of GPU accelerators
- Concise introduction to the usage of data intensive parallel programming frameworks

### *Bibliografia e materiale didattico*

Recommended reading includes:

- "SPM Course notes", by M. Danelutto, PDF available at <http://didawiki.cli.di.unipi.it/doku.php/magistraleinformaticanetworking/spm/sdpm13> (about 300 pages, covers all the lesson and lab material)
- "Structured Parallel Programming (Patterns for Efficient Computation)" by M. McCool, A. D. Robinson, J. Reinders, Morgan Kaufmann 2012 Further bibliography will be indicated, if needed, on the course web site.
- Manuals and tutorials available online relative to the programming frameworks presented and used in the different parts of the



**Indicazioni per non frequentanti**

Students that do not attend the course may use the audio/video records of the lessons available online immediately after the end of the lessons and should also try to solve the homework assignments, available on the web course platform.

**Modalità d'esame**

The exam requires the implementation of a small project and an oral exam. The latter includes a demo of the project, discussion of the project features and performances and general questions relative to the subjects of the course.

*Ultimo aggiornamento 20/09/2022 09:18*