



UNIVERSITÀ DI PISA

PROGRAMMING TOOLS FOR PARALLEL AND DISTRIBUTED SYSTEMS

MASSIMO COPPOLA

Anno accademico

2021/22

CdS

INFORMATICA E NETWORKING

Codice

535AA

CFU

6

Moduli	Settore/i	Tipo	Ore	Docente/i
PROGRAMMING TOOLS FOR PARALLEL AND DISTRIBUTED SYSTEMS	INF/01	LEZIONI	48	MASSIMO COPPOLA

Obiettivi di apprendimento

Conoscenze

The course deals with design, evaluation and utilization of programming tools and environments for parallel and distributed applications. MPI, Thread Building Blocks and OpenCL are used as examples of programming tools addressing diverse kinds of architectural parallelism. oneAPI is presented as a unifying technology that aims at expressing parallelism over several distinct architectural layers. The programming paradigms and the related cost models can be applied to achieve high performance and parallel efficiency on several types of systems, exploiting parallelism at diverse levels/scales in order to address

- high-performance stream-parallel and data-parallel computations, distributed shared memory systems
- unifying methodologies and programming environments for parallel/distributed computing
- parallel systems with hierarchical/multilevel architecture

For these paradigms, static and dynamic tools are presented, typical use-cases analyzed and their performance is evaluated in experimental and laboratory activities.

- Tools for experiment management and application scripting are also discussed, like slurm and the use of python as a scripting language to.

Several of the case studies involve the parallelization of mining/KDD/data analysis/AI algorithms.

Modalità di verifica delle conoscenze

Student knowledge is evaluated during the course thanks to

- the hands-on activities in the lab time,
- the exercises made at home
- interaction during the lessons

and after the course

- evaluation of the project code, of the project report and final oral exam.

Capacità

The student will achieve

- acquaintance with at least three different parallel/distributed programming environments, covering both shared-memory and distributed memory systems (typically MPI, Thread Building Blocks, OpenCL)
- practical experience of applying analytical behavioural models for parallel patterns and full programs with respect to performance, reliability, memory/power efficiency
- practical experience of the full cycle of : problem analysis / parallel solution modeling and design / model-driven implementation / cross evaluation of implementation and analytical models via benchmarking and test results

Modalità di verifica delle capacità

- Hands-on activities during lab-time
- "Homework" exercises
- Through the final project: the project and the written report (describing the performed work, testing activities and evaluation) are discussed with the student as part of the final examination, focusing on the "a priori" modeling and implementation choices made by the student and its ability to evaluate them "a posteriori" based on empirical results.



UNIVERSITÀ DI PISA

Comportamenti

- Critical and empirical reasoning when evaluating parallel programs against their expected behaviour to guide design and coding choices
- Ability to identify and correct algorithm / code design assumptions against the experiments, as well as the ability to devise behavioural models (e.g. performance, reliability, resource usage) and evaluate them against the experiments.

Modalità di verifica dei comportamenti

- The interaction during the lessons and lab time is part of the evaluation
- The project work time is also part of the evaluation (balance of problem solving and ability to identify blocking issues and discuss them)
- The project and the written report (describing the performed work, testing activities and evaluation) final discussion are also contributing to the evaluation.

Prerequisiti (conoscenze iniziali)

The course requires at least good proficiency in C and C++ programming in order to exploit the programming frameworks presented. The course requires some previous knowledge of parallel and distributed computing system architecture (shared and distributed memory parallel systems, multiprocessors and multi-core processors), of structured parallel programming / behavioural skeletons and the associated basic analytical models, at least with respect to performance.

Indicazioni metodologiche

Delivery: teaching with face-to-face lectures; offline recordings of the lectures or, if possible, a simultaneous online streaming in parallel to the face-to-face lectures, as indicated by the recent guidelines of the University of Pisa to face the Covid-19 pandemic.

Learning activities:

- attending lectures, active participation in the discussion
- studying reference texts/papers
- attending lab time with hands-on activities
- personal study and coding experience with the programming tools presented

Lesson Attendance: Not mandatory

Teaching methods:

- Lectures
Slide-based lessons with interactive discussion. Possibly integrated with whiteboard presentation of auxiliary and additional material.
- Lab-time with hands-on experience
Hands-on Lab time with assigned tasks and support from the teacher
Usually individual tasks, using the students' own devices, possibly in remote connection with specific parallel/distributed systems
- Course site used for the distribution of studying material
Slides, papers and book references, text of the exercises for the practical sessions are made available on the dokuwiki page maintained by the Department.
- A final project is mandatory. Access to suitable computing facilities is provided for the project work.

All teaching material is in the English language.

Programma (contenuti dell'insegnamento)

Tools and environments for parallel, high performance

1. Message-passing programming
2. Shared-memory, thread-based programming
3. Shared-memory, stream-oriented multicore programming
4. (optionally) parallel programming for FPGA devices

Applications to case studies of

1. high-performance stream- and data-parallel computations,
2. distributed shared memory algorithms,
3. adaptive and context-aware programming,
4. high-performance event-based programming,
5. run-time supports of languages/frameworks
6. AI relevant algorithms

Bibliografia e materiale didattico



UNIVERSITÀ DI PISA

- B. Wilkinson, M. Allen – Parallel Programming, 2nd edition. 2005, Prentice-Hall.
- Michael Mc Cool, Arch D. Robinson and James Reinders – Structured Parallel Programming (patterns for Efficient Computation) 2012, Morgan Kaufmann.
- Lesson slides, papers, exercises -- made available via the Department's dokuwiki course official page.
- The MPI official standard, version 3.0 (as reference)

Additional material

- James Reinders – Intel Threading Building Blocks 2007, O'Reilly Media.
- M. Voss, R. Asejo, J. Reinders – Pro TBB Book code samples ported to oneAPI (Open access book on Springer)
- J. Reinders et al. - Data Parallel C++ (Open access book on Springer)
- Slides/notes from the teacher (via the course web page).

Indicazioni per non frequentanti

Offline recordings of the lectures are made available or, if possible, a simultaneous online streaming in parallel to the face-to-face lectures is provided, as indicated by the recent guidelines of the University of Pisa to face the Covid-19 pandemic.

The course web page lists slides and additional sources in the "course journal" sub-page.

Please contact the teacher at least by email, or via the MS teams channel in order to

- receive announces and provided additional material
- obtain login credential on the systems that are made available for homework and project work
- define the goal and tools for your personal homework

It is also advised to contact the teacher to book a question-time meeting and discuss any issue that the student may experience.

Modalità d'esame

- Project work comprising algorithm design, coding, testing and evaluating the code
- Final written examination (written report on project work and evaluation)
- Final oral exam (including project discussion)

Pagina web del corso

<http://didawiki.cli.di.unipi.it/doku.php/magistraleinformaticanetworking/spd/start>

Altri riferimenti web

MS teams channel for the course

[MS TEAMS CHANNEL URL](#)

Ultimo aggiornamento 04/03/2022 12:45