095946 ADVANCED ALGORITHMS AND PARALLEL PROGRAMMING

Fabrizio Ferrandi

a.a. 2023-2024

COURSE OBJECTIVES

- This course deals with advanced topics in algorithm design and parallel programming.
- The course is structured in two parts. The first part focuses on general methods and algorithms that are not usually covered by the course “Algoritmi e Principi dell’Informatica”, such as, randomization, amortization, approximation algorithms, string searching/matching, etc.
- The second part deals with parallel programming: Automatic vs. Manual Parallelization, Parallelizing Compilers, Parallel Patterns, Partitioning (domain vs functional decomposition), Communication (cost, latency, bandwidth, visibility, synchronization, etc.), Data Dependencies and Tools/languages such as OpenMP and MPI (no more on CUDA).

COURSE LEARNING OUTCOME: LIFELONG LEARNING SKILLS

- Students will understand how a complex algorithm, possibly parallel, have to be analyzed, designed and assessed. They will play with real problems understanding where pitfall may come when you move from a theoretic formulation down to a real implementation taking into account existing tools and real architectures.

PART I - GENERAL METHODS AND ALGORITHMS

Course Objectives and Introduction

Randomized algorithms: Las Vegas and Monte Carlo algorithms. Analyzing Randomized algorithms.

Hiring Problem and Generating Random Permutations

Randomized Quicksort. Worst-case analysis. Average-case analysis

Order Statistics, Randomized divide-and-conquer algorithm

Primality test. Fast exponentiation. Secret key and cryptosystems

PART I - GENERAL METHODS AND ALGORITHMS

Karger's Min-Cut Algorithm. Faster version by Karger and Stein

Randomized data structures: Skip Lists, Treaps

Dynamic Programming: Memoization. Examples of Dynamic Programming: String Matching, BDDs, etc.

Amortized Analysis: Dynamic tables, Aggregate method, Accounting method and Potential method.

Approximate programming

Competitive Analysis Self-organizing lists Move-to-front heuristic

PART II - ON PARALLELIZATION

- Design of Parallel Algorithms - Parallel Algorithms and Parallel Programming.
- Parallel Machine Model
- Introducing parallel patterns: Reduce, Split, Compact / Expand and Parallel Prefix Sum.
- More on parallel patterns: Segmented Scan, Sort, Map-reduce, Kernel Fusion.
- Tools and languages for parallel programming: Posix Threads, OpenMP, Message Passing Interface.
- Comparison of Parallel Programing Technologies.
- Optimizing parallel performance.
- A library for parallel programming: SHAD.
- Halide: high-performance image and array processing code.
- An introduction to parallel graph computation.

PREVIOUS KNOWLEDGE

Operations Research

Basic data structures

Computational Complexity theory

C/C++ knowledge could help

AAPP TEACHING MATERIAL

- Introduction to Algorithms, T. Cormen, C. Leiserson, R. Rivest and C. Stein,
  MIT Press, Cambridge; 3rd edition (20 Aug 2009) ISBN-10: 0262533057, ISBN-13: 978-0262533058
- Randomized Algorithms, Rajeev Motwani, Prabhakar Raghavan.
  Cambridge University Press (August 25, 1995) ISBN-10: 0521474655, ISBN-13: 978-0521474658
- Michael McCool, James Reinders, Arch Robison, Structured Parallel Programming: Patterns for Efficient Computation, Editore: Morgan Kaufmann, Anno edizione: 2012, ISBN: 0124159931 http://parallelbook.com/
- An Introduction to Parallel Programming, Peter Pacheco,
  Publisher: Morgan Kaufmann; 1 edition (January 21, 2011), ISBN-10: 0123742609, ISBN-13: 978-0123742605
-
- Parallel programming online material link
- Additional material is available on the WeBeep platform of Politecnico di Milano (access restricted to course participants): https://webeep.polimi.it

EXAM HOW

- Evaluation is based on a written exam
- Exam content
  - The solution of some problems based on the practical application of the course concepts and techniques
  - Open answers to some questions on the course concepts and techniques
- After each written test, the teacher can complement the assessment procedure with an oral examination

CONTINUOUS ASSESSMENT

- Continuous assessment will be implemented through two intermediate tests: one as a mid-term test and one at the end of the semester: Nov. 9 – Dec 21.
- All students are admitted to the second test, regardless of the outcome of the first one
- The achieved results will be valid till the end of the academic year or till a student ask to repeat the given part
- Each intermediate test contributed to the final grade with 16 points
- The exam is considered passed if, in both parts, the students get a grade not less than 7, and the sum of the two grades is greater or equal to 18
- 30 cum laude is assigned if students get a sum of grades greater than 30
- If either the first or the second test has a grade less than 7 or the total is less than 18, the student has to take the written test on one of the following dates according to the schedule provided by the School's Academic Calendar
- The student may use one of the valid partial results on the next exam dates. In this case, a customized written exam version will be provided to the student for the parts not yet valid

INNOVATIVE-LEARNING CLASSROOM ACTIVITIES

- Participation in innovative-learning classroom activities will be assessed and will contribute to the final evaluation grade. In particular, in case of valid grades, some questions of the written exam could be skipped.

LECTURERS

- Fabrizio Ferrandi
  - Dipartimento di Elettronica, Informazione e Bioingegneria, first floor,
  - phone. 02 2399 3479,
  - e-mail fabrizio.ferrandi@polimi.it,
  - Home page Ferrandi: http://ferrandi.faculty.polimi.it/
  - Send an e-mail to get in touch with me
- Serena Curzel
  - Dipartimento di Elettronica, Informazione e Bioingegneria,
  - e-mail serena.curzel@polimi.it,
  - Send an e-mail to get in touch with her
- Davide Gadioli
  - Dipartimento di Elettronica, Informazione e Bioingegneria,
  - e-mail davide.gladioli@polimi.it,
  - Send an e-mail to get in touch with him