Parallelism (PAR)

Course presentation

Mario Acosta, Eduard Ayguadé,
Rosa M. Badia (Q1), Josep Ramon Herrero (Q1),
Daniel Jiménez-González[1], Pedro Martínez-Ferrer,
Adrian Munera, Jordi Tubella and Gladys Utrera

Computer Architecture Department
Universitat Politècnica de Catalunya

Course 2022/23 (Spring semester)

[1] Course coordinator (daniel.jimenez-gonzalez@upc.edu)

Outline

Objectives

Syllabus

Methodology

Course materials

Evaluation

General objectives I

Create simple performance models, analyse performance and understand parallel architecture support

- Specific objectives
  - Create simple models to understand the behaviour of parallelisation strategies
  - Know the factors that contribute to performance degradation and how to detect and correct them: granularity, load balance, task interaction overheads, etc.
  - Analyse the performance of a parallel program using instrumentation and analysis tools
  - Understand the required support from the architecture to the parallel programming model

General objectives II

Design, implement, compile and execute parallel programs

- Specific objectives
  - Create a task or data decomposition strategy to parallelise a serial application
  - Implement the parallelisation strategy using the extensions provided by a given parallel programming model
    - Use of task creation and work distribution mechanisms that appropriately balance work and exploit data locality
    - Use of synchronisation techniques to avoid race conditions while minimising overheads
    - Task vs. data decomposition

Outline

Objectives

Syllabus

Methodology

Course materials

Evaluation

Chronological syllabus (2T/P and 2L per week)

1. Why parallel computing?
   - Parallelism and concurrency
2. Understanding parallelism
   - Amdahl's law, speedup, scalability, overheads, performance models, ...
3. Introduction to parallel architectures
4. Parallel programming strategies I: Task decomposition
5. Parallel programming strategies II: Data-aware task decomposition

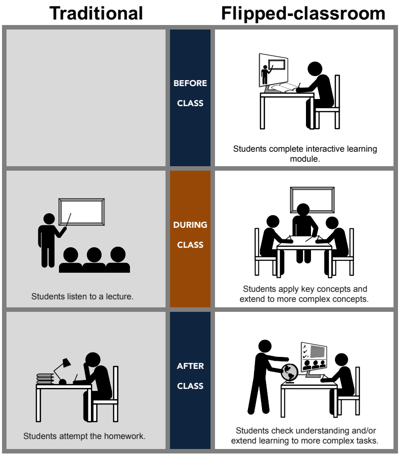
Outline

- Objectives
- Syllabus
- Methodology
- Course materials
- Evaluation

Course methodology: Theory

- Theory/Problems (T/P): 2 hours/week
  - 3 theory/problems sessions per Unit, except Unit 1 (today)
  - Each session as a combination of traditional and flipped-classroom approaches (see next slide)
    - Introductory video lessons and short quizzes to support your interactive at-home learning
    - In class: going deeper and problem resolution
    - Short quizzes and/or problems to assess learning for the session
  - Exams (**In-classroom**, closed book exams: not allowed to use textbooks, notes, collection of problems/exams, ...)
    - Mid-term exam (April 26^{th}, 8:00–10:00)
    - Final Theory exam (June 21^{st}, 15:00–18:00)
    - Final Laboratory exam (June 21^{st}, 15:00–18:00): written in paper

Course methodology: (pseudo-)flipped classroom



Traditional		Flipped-classroom
	BEFORE CLASS	Students complete interactive learning module.
Students listen to a lecture.	DURING CLASS	Students apply key concepts and extend to more complex concepts.
Students attempt the homework.	AFTER CLASS	Students check understanding and/or extend learning to more complex tasks.

Course methodology: Laboratory I

- Laboratory (L): 2 hours/week
  - Participative Laboratories with discussions
  - Only 3 laboratory assignments, done in groups of 2 students
    - One deliverable per group for each assignment
    - Deadline: just before starting the next laboratory assignment
  - Attendance to and performance during laboratory sessions is important to keep up to day subject, prepare assignments and question of the final laboratory exam
- Development context
  - Remote access to a multiprocessor server machine at the Computer Architecture Department
  - Programming language: C using OpenMP extensions

Course methodology: Laboratory II

- Guided laboratory sessions
  - Lab 1: Compilation and execution of OpenMP programs, performance prediction and analysis tools
  - Lab 2: OpenMP tutorial
  - Labs 3, 4 and 5: Parallelisation of applications using OpenMP
  - 2 or 3 sessions devoted to each laboratory assignment (to be detailed by your laboratory professor)
- Students repeating PAR may opt to do a different itinerary for intermediate assignments (not final laboratory exam) in the laboratory if they got a mark ≥ 7.0 in the last course

Outline

Objectives

Syllabus

Methodology

Course materials

Evaluation

Course materials

- All documentation published through **Atenea**
  - Slides, collection of exercises and collection of solved in-term/final exams for the T/P sessions
  - Introductory videos and quizzes to support your interactive at-home learning (pseudo–flipped classroom methodology)
  - Description of L assignments
  - Links to manuals and quick reference guides for the programming models and tools used in L sessions
- All the documentation is in English (third-language transversal competence)

Outline

Objectives

Syllabus

Methodology

Course materials

Evaluation

Evaluation I

$$N = 0.65 \times max(FT, 0.35 \times P + 0.65 \times FT) + 0.25 \times FL + 0.10 \times SL$$

- Based on the following components (all over 10):
  - P: mark of the mid-term exam (includes Units 1 to 3)
  - Final exam (two parts)
    - FT: mark in the final theory exam (includes all Units)
    - FL: mark in the final laboratory exam (includes all lab sessions)
  - SL: laboratory follow-up reports $(1/3(Lab3 + Lab4 + Lab5))$
    - Grading of laboratory deliverables modulated by performance during sessions and attendance, and the result of a possible interview at the end of the course. By active participation, we refer to the reliable demonstration of being working on the laboratory assignment, advancing as far as possible to achieve each session's objectives.
    - Individual student interview, if necessary

Evaluation II

The previous grade will be positively affected by your online activities via Atenea:

- AA: mark of the flipped–classroom (**carried out before the corresponding T session**) and some laboratory activities, over 10
- Final mark:
  If $N \geq 5.0$ then $NF = min(10, N \times (1 + AA \div 100))$;
  otherwise $NF = N$

Third-language transversal competence

- Reading/comprehension: implicit evaluation, no explicit contribution to Final mark (NF)
  - All course material in English
  - Mid–term control and final exam **statement** in English
    - Answer in catalan, spanish or english
- The *third-language generic competence* will be evaluated through
  - Reports for Labs 3, 4 and 5 fully written in English
  - Grading: A, B, C, D or NA

Parallelism (PAR)

Course presentation

Mario Acosta, Eduard Ayguadé,
Rosa M. Badia (Q1), Josep Ramon Herrero (Q1),
Daniel Jiménez-González[2], Pedro Martínez-Ferrer,
Adrian Munera, Jordi Tubella and Gladys Utrera

Computer Architecture Department
Universitat Politècnica de Catalunya

Course 2022/23 (Spring semester)

[2] Course coordinator (daniel.jimenez-gonzalez@upc.edu)