Parallelism (PAR)

Course presentation

Lluc Álvarez, Eduard Ayguadé[1], José Ramón Herrero,
Julián Morillo, Jordi Tubella and Gladys Utrera

Computer Architecture Department
Universitat Politècnica de Catalunya

Course 2018/19 (Fall semester)

[1] Course coordinator (eduard@ac.upc.edu)

General objectives I

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

General objectives II

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

- Specific objectives
  - Create simple models based on the decomposition strategy
  - Analyse the performance of a parallel program using instrumentation and analysis tools
  - Detect performance degradation factors: granularity, load balance, task interaction, etc.
  - Understand the required support from the architecture to the parallel programming model

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. Parallel programming strategies I: Task decomposition
4. Introduction to (shared–memory) parallel architectures
5. Parallel programming strategies II: Data decomposition

Course methodology: Theory

- Theory/Problems (T/P): 2 hours/week (in class)
  - Material available (through **Atenea**):
    - Slides presenting concepts and examples
    - Collection of exercises and exams from previous courses
    - Video lessons and short quizzes (additional material)
  - Two in-term exams:
    - First in-term exam: November 7^{th} (12:30–14:30)
    - Second in-term exam: December 12^{nd} (12:30–14:30)
    - Closed book exams: not allowed to use textbooks, notes, collection of problems/exams, ...

Course methodology: Laboratory I

- Laboratory (L): 2 hours/week (in lab)
  - 5 laboratory assignments (Lab1–5)
  - Documentation for each assignment and additional material available through **Atenea**
  - One deliverable per assignment and group, submitted via **FIB Raco**
    - `pdf` documents, C codes, scripts, etc.
    - Deadline: Just before starting the next laboratory assignment
- Attendance to and performance during laboratory sessions contribute to laboratory grading

Course methodology: Laboratory II

- Guided laboratory sessions
  - Lab 1: Compilation, execution, performance prediction and analysis tools
  - Labs 2, 3 and 5: Parallelisation of applications using OpenMP
  - Lab 4: understanding parallelisation/data–sharing overheads
  - 3 sessions devoted to each laboratory assignment, except for Lab4 (2 sessions)
- Development context:
  - Groups of two students
  - Remote access to a multiprocessor server machine at the Computer Architecture Department
  - Programming language: C using OpenMP extensions

Course materials

- All documentation published through **Atenea**
  - Slides, collection of exercises and collection of solved in-term/final exams for the T/P sessions
  - Videos and quizzes to support your study
  - 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)

Evaluation I

- Based on the following components:
  - AC $= (1^{st}$ in-term exam $\times$ 0.5) $+ (2^{nd}$ in-term exam $\times$ 0.5)
  - Final Exam (EF): **mandatory if** $AC < 5.0$, optional otherwise
  - Lab $=$ Grading of laboratory deliverables modulated by performance during sessions (including attendance to classroom)
    - Lab1 (1 point, 3 sessions), Lab2 (2.5 points, 3 sessions), Lab3 (2.5 points, 3 sessions), Lab4 (1.5 points, 2 sessions) y Lab5 (2.5 points, 3 sessions)
    - Attendance to lab sessions: each not attended session subtracts 10% from the maximum grading of the specific assignment
    - Individual student interview, if necessary

Evaluation II

- Final mark (F)
  - If $AC \geq 5.0$ then
    $F = 0.7 \times max(EF, AC) + 0.3 \times Lab$
    being EF optional in this case (to improve AC)
  - If $AC < 5.0$ then
    $F = 0.7 \times max(EF, 0.25 \times AC + 0.75 \times EF) + 0.3 \times Lab$
- Date for final exam: January 16^{th} (11.30–14:30)

Third-language transversal competence

- Reading/comprehension: implicit evaluation, no explicit contribution to Final mark (F)
  - All course material in English
  - Mid–term controls and final exam **statement** in English
    - Answer in catalan, spanish or english ...
    - ... except for students in group 10 (English group)

- The *third-language generic competence* will be evaluated (optional) through
  - Reports for Labs 2, 3 and 5 fully written in English
  - Rubrics and evaluation criteria known in advance
  - Grading: A, B, C, D or NA

Parallelism (PAR)

Course presentation

Lluc Álvarez, Eduard Ayguadé[2], José Ramón Herrero,
Julián Morillo, Jordi Tubella and Gladys Utrera

Computer Architecture Department
Universitat Politècnica de Catalunya

Course 2018/19 (Fall semester)

[2] Course coordinator (eduard@ac.upc.edu)