270020 - PAR - Parallelism

Coordinating unit: 270 - FIB - Barcelona School of Informatics
Teaching unit: 701 - AC - Department of Computer Architecture
Academic year: 2019
Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Compulsory) BACHELOR'S DEGREE IN ENGINEERING PHYSICS (Syllabus 2011). (Teaching unit Optional)
ECTS credits: 6

Teaching languages: Catalan, Spanish, English

Teaching staff

Coordinator: - Eduard Ayguadé Parra (eduard@ac.upc.edu)
Others: - Daniel Jimenez Gonzalez (djmenez@ac.upc.edu)
- Gladys Miriam Utrera Iglesias (gutrera@ac.upc.edu)
- Jordi Tubella Murgadas (jordit@ac.upc.edu)
- Josep Ramon Herrero Zaragoza (josepr@ac.upc.edu)
- Julian David Morillo Pozo (jmorillo@ac.upc.edu)
- Lluc Álvarez Martí (lluca@ac.upc.edu)
- Rosa Maria Badia Sala (rosab@ac.upc.edu)

Prior skills

The capabilities are defined by the prior pre-requisites for the course.

Requirements

- Prerequisite AC

Degree competences to which the subject contributes

Specific:
CT1.1B. To demonstrate knowledge and comprehension about the fundamentals of computer usage and programming. Knowledge about the structure, operation and interconnection of computer systems, and about the fundamentals of its programming.
CT5.1. To choose, combine and exploit different programming paradigms, at the moment of building software, taking into account criteria like ease of development, efficiency, portability and maintainability.
CT5.3. To design, write, test, refine, document and maintain code in an high level programming language to solve programming problems applying algorithmic schemas and using data structures.
CT5.6. To demonstrate knowledge and capacity to apply the fundamental principles and basic techniques of parallel, concurrent, distributed and real-time programming.
CT6.2. To demonstrate knowledge, comprehension and capacity to evaluate the structure and architecture of computers, and the basic components that compound them.
CT7.2. To evaluate hardware/software systems in function of a determined criteria of quality.
CT8.1. To identify current and emerging technologies and evaluate if they are applicable, to satisfy the users needs.

General:
G3. THIRD LANGUAGE: to know the English language in a correct oral and written level, and accordingly to the needs of the graduates in Informatics Engineering. Capacity to work in a multidisciplinary group and in a multi-language environment and to communicate, orally and in a written way, knowledge, procedures, results and ideas related to the technical informatics engineer profession.
270020 - PAR - Parallelism

Teaching methodology

The theory classes introduce all the knowledge, techniques, concepts needed to be put into practice problems in class and lab as well as personal work using a collection of problems.

Two hours of theory/problems are done per week. The two hours of laboratory classes are also done every week.

The course uses the C programming language and mainly the OpenMP parallel programming model.

Learning objectives of the subject

1. The student should be able to formulate simple performance models given a parallelization strategy for an application, that allow to estimate the influence of major architectural aspects: number of processing elements, data access cost, cost of interaction between processing elements, among others.
2. The student should be able to measure, using instrumentation, visualization and analysis tools, the performance achieved with the implementation of a parallel application and to detect factors that limit this performance: granularity of tasks, equitable load, interaction between tasks, among others.
3. The student should be able to compile and execute a parallel program, using the basic command line tools to measure the execution time.
4. The student should be able to apply simple optimizations in parallel kernels to improve their performance for parallel architectures, attacking the factors that limit performance.
5. The student should be able to choose the most appropriate decomposition strategy to express parallelism in an application (tasks, data).
6. The student should be able to apply the basic techniques to synchronize parallel execution, avoiding race conditions and deadlock, and enabling the overlap between computation and interaction, among others.
7. Students must be able to program in OpenMP the parallel version of a sequential application
8. The student should be able to identify the different types of parallelism that can be exploited in a computer architecture (ILP, TLP, and DLP within a processor, multiprocessor and multicomputer) and describe its principles of operation.
9. Students must be able to understand the basics of coherence and data sharing in shared-memory parallel architectures, both with uniform and non-uniform access to memory.
10. The student should be able to follow the course using the materials provided in English (slides, laboratory and practical sessions), as well as to do the mid-terms and final exams with the statement written in English.
11. If the foreign language competence is chosen, the student should be able to write the deliverables associated to laboratory assignments (partially or fully) in English.

Study load

<table>
<thead>
<tr>
<th>Study load</th>
<th>Hours large group:</th>
<th>Hours medium group:</th>
<th>Hours small group:</th>
<th>Guided activities:</th>
<th>Self study:</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total learning time: 150h</td>
<td>30h</td>
<td>0h</td>
<td>30h</td>
<td>6h</td>
<td>84h</td>
<td>20.00%</td>
</tr>
<tr>
<td></td>
<td>20.00%</td>
<td>0.00%</td>
<td>20.00%</td>
<td>4.00%</td>
<td>56.00%</td>
<td></td>
</tr>
</tbody>
</table>
## Introduction and motivation

**Degree competences to which the content contributes:**

**Description:**
Necessitat del paral·lelisme, paral·lelisme vs. concurrència, possibles problemes en l'ús de concurrència: deadlock, lifelock, starvation, fairness, data races

## Analysis of parallel applications

**Degree competences to which the content contributes:**

**Description:**
Mètriques bàsiques: paral·lelisme, temps d'execució, speedup i escalabilitat. Anàlisi de l'impacte dels overheads associats a la creació de tasques i la seva sincronització i la compartició de dades. Eines per la predicció i l'anàlisi de paral·lelisme i visualització de comportament: Paraver i Tareador

## Parallel programming principles: task decomposition

**Degree competences to which the content contributes:**

**Description:**
Task decomposition vs. data decomposition. Descomposicio en tasques, granularitat i anàlisi de dependències. Identificació de patrons de paral·lelisme: iterative vs. divide and conquer task decompositions. Mecanismes per implementar la descomposició en tasques: creació de regions paral·leles i tasques; mecanismes per garantir task ordering i data sharing.

## Introduction to parallel architectures

**Degree competences to which the content contributes:**

**Description:**
Paral·lelisme dins d'un processador (ILP, DLP i TLP) i entre els processadors que formen els multiprocessadors de memòria compartida SMP i ccNUMA (coherència de cache, consistència de memòria, sincronització).

## Parallel programming principles: data decomposition

**Degree competences to which the content contributes:**

**Description:**
Descomposició de dades (descomposició geomètrica vs. estructures recursives) per arquitectures amb memòria compartida. Localitat en l'accés a les dades en arquitectures paral·leles de memòria compartida. Generació de codi en funció de la descomposició de dades. Breu introducció a les arquitectures de memòria distribuïda i la seva programació (cas concret: MPI).
Shared-memory programming: OpenMP

<table>
<thead>
<tr>
<th>Degree competences to which the content contributes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
</tr>
</tbody>
</table>

Midterm problems review

<table>
<thead>
<tr>
<th>Degree competences to which the content contributes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
</tr>
<tr>
<td>En aquestes sessions es resoldran dubtes que els estudiants puguin tenir en els problemes dels controls.</td>
</tr>
</tbody>
</table>
## Planning of activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Hours</th>
<th>Theory classes:</th>
<th>Practical classes:</th>
<th>Laboratory classes:</th>
<th>Guided activities:</th>
<th>Self study:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assimilation of fundamental concepts and tools for modeling and analyzing the behavior of parallel applications</td>
<td>20h</td>
<td>6h</td>
<td>0h</td>
<td>8h</td>
<td>0h</td>
<td>6h</td>
</tr>
<tr>
<td>Description: Actively participate in sessions of theory/problems. Study the contents of topics 1 and 2 and perform the proposed exercises. Resolution of the exercises in the laboratory sessions and understand the results.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific objectives: 1, 2, 3, 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Using OpenMP to express of parallelism in shared memory</td>
<td>44h</td>
<td>0h</td>
<td>0h</td>
<td>22h</td>
<td>0h</td>
<td>22h</td>
</tr>
<tr>
<td>Description: Actively participate in sessions of theory/problems. Study the contents of topic 6 and prepare the implementation of exercises for the laboratory sessions. Resolution of the exercises in the laboratory sessions and extraction of conclusions.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific objectives: 4, 7, 10, 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assimilation of the fundamentals for task decomposition</td>
<td>18h</td>
<td>8h</td>
<td>0h</td>
<td>0h</td>
<td>0h</td>
<td>10h</td>
</tr>
<tr>
<td>Description: Actively participate in sessions of theory/problems. Study the contents of topic 4 and perform the proposed exercises. Apply new knowledge when solving the laboratory exercises for topic 6.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific objectives: 5, 6, 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control for topics 1, 2 and 3</td>
<td>10h</td>
<td></td>
<td></td>
<td></td>
<td>2h</td>
<td>8h</td>
</tr>
<tr>
<td>Description:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Specific objectives:
1, 5, 6, 7, 9, 10

### Assimilation of the fundamental aspects in parallel architectures

**Hours**: 12h  
Theory classes: 6h  
Practical classes: 0h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 6h

**Description:**  
Actively participate in sessions of theory/problems. Study the contents of topic 5 and perform the proposed exercises.

**Specific objectives:**  
8, 10

### Assimilation of the fundamentals for data decomposition

**Hours**: 14h  
Theory classes: 6h  
Practical classes: 0h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 8h

**Description:**  
Actively participate in sessions of theory/problems. Study the contents of topic 5 and perform the proposed exercises. Use OpenMP to express data decompositions for shared-memory architectures.

**Specific objectives:**  
5, 6, 10

### Control for topics 4 and 5

**Hours**: 10h  
Guided activities: 2h  
Self study: 8h

**Specific objectives:**  
4, 5, 6, 7, 8, 10

### Midterm problems review

**Hours**: 7h  
Theory classes: 0h  
Practical classes: 0h  
Laboratory classes: 0h  
Guided activities: 3h  
Self study: 4h
270020 - PAR - Parallelism

**Description:**
Actively participate in sessions of problems.

**Specific objectives:*
9, 10

---

**Final exam**

**Hours:** 15h
  - Guided activities: 3h
  - Self study: 12h

**Specific objectives:**
4, 5, 6, 7, 8, 9, 10

---

**Qualification system**

The grade for the course is computed from 2 notes:
- Theory contents (weight 70%).
- Laboratory evaluation (weight 30%).

The laboratory grade (Lab) is mainly obtained from the marks obtained in the deliverables at the end of each assignment, modulated with the performance during the laboratory sessions and a possible interview at the end of the course by the laboratory professor.

During the course, 2 mid-term exams are done (C1 and C2). The continuous assessment mark (AC) is computed as the mean of the marks obtained in the 2 mid-term exams:

\[ AC = 0.5 \times C1 + 0.5 \times C2 \]

If \( AC \geq 5 \) then the student's final grade (NF) will be:

\[ NF = 0.3 \times Lab + 0.7 \times AC \]

Students with \( AC < 5 \) will have to do the final exam (EF) that determines their grade for the theory part. In this case, the new final grade will be:

\[ NF = 0.3 \times Lab + 0.7 \times \max(EF, 0.25 \times AC + 0.75 \times EF) \]

Students with \( AC \geq 5 \) that want to do the final exam in order to improve their mark will have to send an e-mail to the coordinator at least one week before the exam date. In this case, the new final grade will be calculated as follows:

\[ NF = 0.3 \times Lab + 0.7 \times \max(EF, AC) \]

The foreign language competence will be evaluated from the reports delivered for the laboratory assignments. These reports should be written (partially or fully) in English and they will require reading the laboratory assignment description (also in English) as well as the OpenMP specifications. Both the structure of the written document and the ability to transmit the results and conclusions of the work will be used to evaluate the competence (following a rubrics document). The grade for the competence will be A (excellent), B (good), C (satisfactory) , D (fail) or NA (Not evaluated).
Bibliography

Basic:


Complementary:
