270068 - PAP - Parallel Programming and Architectures

Coordinating unit: 270 - F1B - Barcelona School of Informatics
Teaching unit: 701 - AC - Department of Computer Architecture
Academic year: 2016
Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Optional)
ECTS credits: 6  Teaching languages: Catalan

Teaching staff

Coordinator: - Eduard Ayguadé Parra (eduard@ac.upc.edu)
 Others: - Lluc Álvarez Martí (lluca@ac.upc.edu)

Prior skills

Defined by the pre-requisites for the course

Requirements

- Prerequisite PAR

Degree competences to which the subject contributes

Specific:
CEC2.1. To analyse, evaluate, select and configure hardware platforms for the development and execution of computer applications and services.
CEC2.2. To program taking into account the hardware architecture, using assembly language as well as high-level programming languages.
CT8.7. To control project versions and configurations.

General:
G8. APPROPRIATE ATTITUDE TOWARDS WORK: to have motivation to be professional and to face new challenges, have a width vision of the possibilities of the career in the field of informatics engineering. To feel motivated for the quality and the continuous improvement, and behave rigorously in the professional development. Capacity to adapt oneself to organizational or technological changes. Capacity to work in situations with information shortage and/or time and/or resources restrictions.

Teaching methodology

The theory lessons introduce the knowledge, techniques, and concepts using examples of real code or pseudo-code. These lessons will be complemented with the realization of problems in the practical lessons. The laboratory sessions put into practice the theoretical contents, and evaluate the behavior and performance of the solutions proposed.

The course assumes that part of the theoretical contents, or laboratory statements, will have to be developed by the student independently.

The course is mainly focused on cluster architectures, using the C programming language, the Pthreads library and the OpenMP and MPI programming models.

Learning objectives of the subject
1. Students should be able to implement the basic functionalities in a library supporting the execution of parallel applications on a shared-memory architecture.
2. Students must be able to write and understand parallel programs that make use of the low-level Pthreads interface.
3. Students must be able to understand the main components used to build a multiprocessor architecture, and design on paper a system that fulfill certain design restrictions.
4. Students must be able to write simple applications using the MPI programming model, evaluate their performance and identify the critical parts that limit scalability.
5. Students must be able to perform a specific job even though the statement is incomplete or information relevant to the implementation is missing.
6. Students must be able to identify the possibilities of professional use of the knowledge acquired in the course.
7. Students must be able to assess the quality of a proposed solution to a specific problem.
8. Students must be capable of completing or expanding knowledge independently.

<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>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total learning time:</strong></td>
<td>15h</td>
<td>15h</td>
<td>30h</td>
<td>6h</td>
<td>84h</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.00%</td>
<td>10.00%</td>
<td>20.00%</td>
<td>4.00%</td>
<td>56.00%</td>
</tr>
</tbody>
</table>
### Implementation of a shared-memory programming model: threads and synchronization

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

**Description:**
This topic presents how to design and implement a library supporting the execution of parallel programs in OpenMP, in particular the aspects related with thread management and synchronization.

### Implementation of a shared-memory programming model: work sharing

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

**Description:**
This topic presents how to design and implement a library supporting the execution of parallel programs in OpenMP, in particular the aspects related with work sharing in the OpenMP worksharing constructs.

### Implementation of a shared-memory programming model: tasking model

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

**Description:**
This topic will present how to design and implement a library supporting the execution of parallel programs in OpenMP, in particular the aspects related with the tasking model.

### Components and design of a cluster architecture

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

**Description:**
This topic will introduce the main components in a cluster architecture we the objective of doing a design with certain performance/power trade-offs and budget.

### MPI: parallel programming for distributed-memory architectures

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

**Description:**
This topic will introduce how to program parallel applications using MPI, a programming model based on message passing for distributed-memory cluster architectures.
## Planning of activities

| **Implementation of a shared-memory programming model** | **Hours:** 49h  
Theory classes: 4h  
Practical classes: 4h  
Laboratory classes: 18h  
Guided activities: 0h  
Self study: 23h |
|--------------------------------------------------------|--------------------------------------------------|
| **Specific objectives:**  
1, 2, 5, 7, 8 | |

| **Components and design of a cluster architecture** | **Hours:** 26h  
Theory classes: 4h  
Practical classes: 4h  
Laboratory classes: 0h  
Guided activities: 0h  
Self study: 18h |
|------------------------------------------------------|--------------------------------------------------|
| **Specific objectives:**  
3, 8 | |

| **Control 1: Implementation of a shared-memory programming model** | **Hours:** 10h  
Guided activities: 2h  
Self study: 8h |
|------------------------------------------------------------------|--------------------------------------------------|
| **Specific objectives:**  
1, 2, 3 | |

| **Other parallel programming models: MPI** | **Hours:** 36h  
Theory classes: 4h  
Practical classes: 4h  
Laboratory classes: 10h  
Guided activities: 0h  
Self study: 18h |
|-----------------------------------------|--------------------------------------------------|
| **Specific objectives:**  
4, 5, 7, 8 | |

| **Control 2: Other parallel programming models** | **Hours:** 10h  
Guided activities: 2h  
Self study: 8h |
|-------------------------------------------------|--------------------------------------------------|
| **Specific objectives:**  
4 | |
Possibilities of professional use for parallel application programming

<table>
<thead>
<tr>
<th>Description:</th>
<th>Hours: 5h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visit the MareNostrum supercomputer, description of some of the parallel applications that are running on it, and relationship to the contents of the course</td>
<td>Theory classes: 0h</td>
</tr>
<tr>
<td></td>
<td>Practical classes: 0h</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 0h</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 4h</td>
</tr>
<tr>
<td></td>
<td>Self study: 1h</td>
</tr>
</tbody>
</table>

**Specific objectives:**

6

Final Exam

<table>
<thead>
<tr>
<th>Hours: 10h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes: 0h</td>
</tr>
<tr>
<td>Practical classes: 0h</td>
</tr>
<tr>
<td>Laboratory classes: 0h</td>
</tr>
<tr>
<td>Guided activities: 2h</td>
</tr>
<tr>
<td>Self study: 8h</td>
</tr>
</tbody>
</table>

**Specific objectives:**

1, 2, 3, 4

Qualification system

The grade for the course is calculated from 3 Notes:
- Theory (50%)
- Lab (30%)
- Autonomy and motivation (20%)

The theoretical grade (T) is obtained from two tests C1 and C2, contributing with a weight of 60% and 40%, respectively.

The laboratory grade (H) is obtained from the grades of the delivery and monitoring of the practice sessions by the teacher.

The grade of autonomy and motivation (A) evaluates the ability of students to face situations of lack of information and their motivation to explore additional topics or go beyond what is initially assigned. It is obtained from the results of these laboratory experiments that lack information or where the theory has not been developed in class, requiring the exploration of additional material. Laboratory practices and theoretical questions that contribute to this grade will be indicated in the corresponding statements.

The final grade is calculated thus:

\[
T = C1 \times 60\% + C2 \times 40\%
\]

\[
F = T \times 50\% + L \times 30\% + A \times 20\%
\]

If a student fails the theoretical part (T < 5), or wants to improve his grade, he will take a final exam that will determine its new theoretical grade T.
Bibliography

Basic:


Complementary:


Others resources:

Hyperlink

https://computing.llnl.gov/tutorials/pthreads/

https://computing.llnl.gov/tutorials/mpi/

http://www.bsc.es/computer-sciences/performance-tools