COMPUTAÇÃO EM LARGA ESCALA

Ano lectivo de 2023/2024

1. GENERAL INFO

Semester: 2
Weekly load: 3H (lectures + labs)
Credit units: 6 *ECTS*
Audience: Mestrado em Engenharia Informática (*MEI*)
Scientific area: Informática
Coordination: António Rui Borges (*ARB*) – ruib@det.ua.pt

2. TEACHING STAFF

António Rui Borges (*ARB*) – ruib@det.ua.pt
T 2.ª feira: 10h – 11h (Anf IV)
P1 2.ª feira: 11h – 13h (Room 101)
P2 2.ª feira: 15h – 17h (Room 101)
P3 2.ª feira: 17h – 19h (Room 101)
Tutorial 2.ª feira: 14h – 15h (Room 101)
Student assistance is available by appointment (IEETA)

3. OBJECTIVES

- to acquaint the students to the principles of high performance computing
- to introduce the most important paradigms of algorithmic design, communication and synchronization in parallel programming.

4. LEARNING OUTCOMES

- to gain a good understanding on the main issues related to programming at different levels of parallelism (coarse, medium and fine)
- to develop skills for the design and the implementation of simple parallel applications on multi processing architectures
- to acquaint the students with the functionality of C-based parallel programming environments using the *pthread* library, MPI and CUDA.

5. PRE-REQUISITES

- basic knowledge of computer architecture
- basic knowledge about operating systems and multiprogrammed environments
- fair to advanced working knowledge of C language and some knowledge of the principles of concurrent programming.

6. SYLLABUS

- *High performance computing*
  Architectural basics of a parallel machine
  Decomposition techniques
  Law of Amdahl
  Tools used in programming parallel applications
- *Medium-grain parallelism*
  Revision of basic notions about computer architecture
  General principles of concurrency
  Synchronization devices
  Decomposition model
  Library pthead
- *Coarse-grain parallelism*
  Concept of message exchange
  Decomposition model
  Synchronization devices
  Computational model provided by MPI
- *Fine-grain parallelism*
  Heterogeneous computing using GPUs (Graphical Processing Units)
  Architecture of a GPU
  Language CUDA C
  Computational model
  Decomposition techniques.

7. BIBLIOGRAPHY

Introduction to HPC with MPI for Data Science, F. Nielsen, Springer, 2016

Parallel Programming in C with MPI and OPenMP, M. Quinn, McGraw-Hill, 2003

Professional Cuda C Programming, J. Cheng, M.Grossman, T. McKercher, John Wiley & Sons, 2014

Programming Massively Parallel Processors, D. Kirk, W. Hwu, Morgan Kaufmann, 2017

8. TEACHING / LEARNING MODEL

The course is organized in lectures, lab classes and tutorials.

Lectures present specific topics of the syllabus. The adopted approach tries to entice the students to participate actively in the discussion and to help them to develop skills of critical reasoning and to learn general techniques of problem solving.

Labs follow the motto "*you learn by doing*" and are organized in two parts. In the first, aimed to acquaint the students with specific topics C language, a set of problems is proposed whose solution as single-threaded applications is discussed. In the second, the remaining classes, another set of problems is considered and different implementations based on different communication constructs are proposed. These will constitute the assignments the students have to carry out.

***Work assignment* 1**
Purely concurrent (multithreaded) solution of the problems.

***Work assignment* 2**
Parallel solution of the problems based on message passing using MPI.

***Work assignment* 3**
Parallel solution of the problems based on shared variables using CUDA.

Students are organized in working groups of two elements. Each group must present and defend its approach to the solution and its implementation during a query session.

Tutorials have for the most part an exposition character and aim to help the students to overcome deficiencies on the background knowledge some of them may have, as well as to provide a space for the discussion of specific aspects of the course.

9. GRADING

1. Course grade is determined by the formula

$$\textit{course grade} = \frac{5 \text{ x } \textit{theoretical mark} + 5 \text{ x } \textit{lab mark}}{10},$$

   rounding is always carried out *half up* to unities, except when the lab mark is higher than the theoretical mark by more than three units; in this case, rounding is carried out *half down*.

2. Theoretical mark is obtained by seating to a written examination which takes place at *época normal* or *época de recurso*. A *minimum mark* of 8,5 units is required.
3. Lab mark is obtained by the evaluation of the three work assignments. Each has an equal weight. A *minimum mark* of 8,5 units is required.
4. The student *passes* the course if the course grade is higher or equal to 10 units and simultaneously both the theoretical and lab marks are at least equal to the *minimum mark*.
5. The student *fails* the course if the course grade is lower than 10 units, or either the theoretical or the lab mark is lower than the *minimum mark*.
6. When the lab mark is lower than the *minimum mark*, the student *fails by minimum mark*.
7. A *regular* student may also *fail* by missing more than three lab classes. In this case, the student *fails by absence*.

10. SPECIAL DATES

deadline for delivering work assignment 1: 31 de Março de 2024

deadline for delivering work assignment 2: 4 de Maio de 2024

deadline for delivering work assignment 3: 5 de Junho de 2024