# Course guide
# 270647 - SCA - Supercomputing for Challenging Applications

**Last modified:** 13/07/2022

**Unit in charge:** Barcelona School of Informatics
**Teaching unit:** 701 - DAC - Department of Computer Architecture.

**Degree:** MASTER'S DEGREE IN INNOVATION AND RESEARCH IN INFORMATICS (Syllabus 2012). (Optional subject).

**Academic year:** 2022 **ECTS Credits:** 6.0 **Languages:** English

## LECTURER

**Coordinating lecturer:** CARLOS ALVAREZ MARTINEZ

**Others:** Primer quadrimestre:
CARLOS ALVAREZ MARTINEZ - 10
JOSE RAMON HERRERO ZARAGOZA - 10
JOSEP LARRIBA PEY - 10

## PRIOR SKILLS

Basic understanding of parallel architectures, including shared- and distributed-memory multiprocessor systems.

Useful programming skills of some parallel programming model.

## DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

**Specific:**
CEC3. Ability to apply innovative solutions and make progress in the knowledge that exploit the new paradigms of Informatics, particularly in distributed environments.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.
CEE4.2. Capability to analyze, evaluate, design and optimize software considering the architecture and to propose new optimization techniques.

**Generical:**
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG5. Capability to apply innovative solutions and make progress in the knowledge to exploit the new paradigms of computing, particularly in distributed environments.

## TEACHING METHODOLOGY

During the course there will be two types of activities:

a) Activities focused on the acquisition of theoretical knowledge.
b) Activities focused on the acquisition of knowledge through experimentation by implementing and evaluating empirically in the laboratory the mechanisms explained at a theoretical level.

The theoretical activities include participatory lecture classes, which explain the basic contents of the course. The practical activities include seminar laboratories using the student's laptop in class, where students implement the mechanisms described in the lectures. The seminars require a preparation by reading the statement and supporting documentation, and a further elaboration of the conclusions in a report.

## LEARNING OBJECTIVES OF THE SUBJECT

1.The student should be able to understand the complexity of different algorithms, identify the computationally intensive parts of a simulation or data processing, and decide which parts need to be optimized and parallelized.
2.The student must be able to design and implement efficient parallel simulation and data processing algorithms using a parallel programing model.
3.The student must be able to evaluate the different tradeoffs (robustness, computational cost, scalability) in order to select a specific algorithm for a simulation or data processing problem

## STUDY LOAD

| Type | Hours | Percentage |
|---|---|---|
| Self study | 96,0 | 64.00 |
| Hours large group | 54,0 | 36.00 |

**Total learning time:** 150 h

## CONTENTS

### Introduction

**Description:**
- Introduction: overview
- The modern scientific method
- Simulation and optimization
- HPC vs. HTC
- Numerical simulations
- Limits of parallelization
- Evolution and limits of HPC systems

### Introduction to Numerical Simulations

**Description:**
- From Models to Algorithms
- Discretization and PDEs
- Types of PDEs: Elliptic, Parabolic, Hyperbolic
- From problem to math and solution
- Numerical schemes: Explicit vs. Implicit
- Finite differences and Finite Elements

### Direct Solving of Large-Scale Linear Systems of Equations

**Description:**
- Triangular systems and parallelization
- Gauss elimination
- LU factorization
- Partitioning methods
- HPL

### Iterative Solving of Large-Scale Linear Systems of Equations

**Description:**
- Direct vs iterative methods
- Jacobi
- Parallelization of iterative methods
- Gauss-Seidel and SOR
- Krylov methods and preconditioning: HPCG
- Software for numerical methods: BLAS, LAPACK, etc.

### Numerical systems practical cases

**Description:**
- Parallel programming models
- Parallelism and granularity
- Block parallelism
- Sparse systems

### ML problems introduction

**Description:**
- Introduction to the common operations of DNNs
- Most common DNN applications
- Architectures for DNNs
- Optimizations for DNNs

### Sequence Alignment

**Description:**
This is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences.
Search, scoring and parallel strategies are important to overcome this challenge.

### Practical Cases

**Description:**
- DNNs and precision
- Parallelism in DNNs
- Sequence Alignment algorithms

### Introduction to the kernel of DBMSs and the execution of queries in such systems

**Description:**
This topic has the objective to understand the different software layers of a DBMS and how they interact, the different complexities that they incarnate and how their interaction determines the performance of such DBMSs.

### Big data kernels and graph databases

**Description:**
This session has the objective to introduce the students to the design of such Big-data and graph database systems, how they are built and how their performance is influenced by the structure of their different software layers.

### Benchmarking for Databases

**Description:**
Benchmarking is one of the most important issues in Database design and evolution. This part of the course will be designed to understand the different efforts being carried in benchmarking from the USA and Europe for relational and graph databases.

### Big Data practical cases

**Description:**
Big Data practical cases

## ACTIVITIES

### Introduction

**Description:**
Follow the lectures, study the materials and practices.

**Specific objectives:**
1

**Related competencies :**
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.

**Full-or-part-time:** 2h
Theory classes: 2h

**Part I: Numerical Applications**

**Description:**
Follow the lectures, study the materials and practices.

**Specific objectives:**
1, 2, 3

**Related competencies :**
CG5. Capability to apply innovative solutions and make progress in the knowledge to exploit the new paradigms of computing, particularly in distributed environments.
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE4.2. Capability to analyze, evaluate, design and optimize software considering the architecture and to propose new optimization techniques.
CEC3. Ability to apply innovative solutions and make progress in the knowledge that exploit the new paradigms of Informatics, particularly in distributed environments.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.

**Full-or-part-time:** 18h
Theory classes: 8h
Self study: 10h

**Numerical Laboratory**

**Full-or-part-time:** 22h
Laboratory classes: 8h
Guided activities: 2h
Self study: 12h

**Deliverable: Assignment on Numerical Applications**

**Description:**
Assignment for the Numerical Applications part. To be delivered at Racó.

**Specific objectives:**
1, 2, 3

**Related competencies :**
CG5. Capability to apply innovative solutions and make progress in the knowledge to exploit the new paradigms of computing, particularly in distributed environments.
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE4.2. Capability to analyze, evaluate, design and optimize software considering the architecture and to propose new optimization techniques.
CEC3. Ability to apply innovative solutions and make progress in the knowledge that exploit the new paradigms of Informatics, particularly in distributed environments.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.

**Full-or-part-time:** 12h
Self study: 12h

**Part II: Non-Numerical Applications: ML & Bio-Informatics**

**Description:**
Follow the lectures, study the materials and practices.

**Specific objectives:**
1, 2, 3

**Related competencies :**
CG5. Capability to apply innovative solutions and make progress in the knowledge to exploit the new paradigms of computing, particularly in distributed environments.
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE4.2. Capability to analyze, evaluate, design and optimize software considering the architecture and to propose new optimization techniques.
CEC3. Ability to apply innovative solutions and make progress in the knowledge that exploit the new paradigms of Informatics, particularly in distributed environments.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.

**Full-or-part-time:** 20h
Theory classes: 10h
Self study: 10h

**Non numerical applications laboratory**

**Full-or-part-time:** 16h
Laboratory classes: 6h
Self study: 10h

**Deliverable: Non-numerical assignment**

**Specific objectives:**
1, 2, 3

**Related competencies :**
CG5. Capability to apply innovative solutions and make progress in the knowledge to exploit the new paradigms of computing, particularly in distributed environments.
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE4.2. Capability to analyze, evaluate, design and optimize software considering the architecture and to propose new optimization techniques.
CEC3. Ability to apply innovative solutions and make progress in the knowledge that exploit the new paradigms of Informatics, particularly in distributed environments.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.

**Full-or-part-time:** 12h
Self study: 12h

**Part III: Big Data problems**

**Description:**
Follow the lectures, study the materials and practices.

**Full-or-part-time:** 20h
Theory classes: 10h
Self study: 10h

**Big Data Laboratory**

**Full-or-part-time:** 16h
Laboratory classes: 6h
Self study: 10h

**Big Data assignment**

**Specific objectives:**
1, 2, 3

**Related competencies :**
CG5. Capability to apply innovative solutions and make progress in the knowledge to exploit the new paradigms of computing, particularly in distributed environments.
CG3. Capacity for mathematical modeling, calculation and experimental designing in technology and companies engineering centers, particularly in research and innovation in all areas of Computer Science.
CG1. Capability to apply the scientific method to study and analyse of phenomena and systems in any area of Computer Science, and in the conception, design and implementation of innovative and original solutions.
CEE4.2. Capability to analyze, evaluate, design and optimize software considering the architecture and to propose new optimization techniques.
CEC3. Ability to apply innovative solutions and make progress in the knowledge that exploit the new paradigms of Informatics, particularly in distributed environments.
CEE3.3. Capability to understand the computational requirements of problems from non-informatics disciplines and to make significant contributions in multidisciplinary teams that use computing.

**Full-or-part-time:** 12h
Self study: 12h

## GRADING SYSTEM

The course will be evaluated with a partial grade for each content part (Numerics, No numerics and Big Data). Each part has the same weight in the final grade:

Grade = A1/3 + A2/3 + A3/3

Where

Ai := Grade from part i (i from 1 to 3)

and

- Each part of the course has few short and partial deliverables (PD) and a final project deliverable (FPD).

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

## BIBLIOGRAPHY

**Basic:**
- Burnett, D.S. Finite element analysis: from concepts to applications. Addison-Wesley, 1987. ISBN 0201108062.
- Chapra, S.C.; Canale, R.P. Numerical methods for engineers. 8th ed. New York: McGraw-Hill, 2021. ISBN 9781260571387.
- PrÅ¾ulj, Nataša. Analyzing network data in biology and medicine. Cambridge University Press, 2019. ISBN 9781108432238.
- Saad, Y. Iterative methods for sparse linear systems. 2nd ed. SIAM, 2003. ISBN 0898715342.
- Quinn, M.J. Parallel programming in C with MPI and OpenMP. International edition. McGraw-Hill, 2003. ISBN 0071232656.
- Briggs, W.L.; Henson, V.E.; McCormick, S.F. A multigrid tutorial. 2nd ed. Society for Industrial and Applied Mathematics, 2000. ISBN 0898714621.
- Tramontano, A. The ten most wanted solutions in protein bioinformatics. Chapman and Hall/CRC, 2005. ISBN 1584884916.
- Gray, J.; Reuter, A. Transaction processing: concepts and techniques. Morgan Kaufmann, 1993. ISBN 1558601902.

**Complementary:**
- Cohen, G.C. Higher-order numerical methods for transient wave equations. Przulj, Natasa. Springer, 2001. ISBN 354041598X.
- Fichtner, A. Full seismic waveform modelling and inversion. Springer, 2011. ISBN 9783642158063.
- Wesseling, P. An introduction to multigrid methods. Edwards, 2004. ISBN 1930217080.

## RESOURCES

**Hyperlink:**
- http://glaros.dtc.umn.edu/gkhome/views/metis/- http://openmp.org/- http://www.mgnet.org/- http://www.netlib.org/utk/people/JackDongarra/la-sw.html- http://www.open-mpi.org/