2SC7610 – Parallel Computing Methods and Algorithms, and Optimization Methods

Instructors: Frédéric Magoules
Department: DÉPARTEMENT INFORMATIQUE
Language of instruction: ANGLAIS
Campus: CAMPUS DE PARIS - SACLAY
Workload (HEE): 60
On-site hours (HPE): 34,50

Description

Simulation is today at the heart of many design and optimization approaches, to reduce the impact of the products created: reducing the carbon footprint, the sound footprint ... Such problems are often complex systems, whose simulation requires specific skills in high performance and large scale simulations.

In this course, students will learn to develop models and simulations without limit on the size of the problem, without sacrificing the accuracy of computations. For this purpose:

- They will design models based on blocks of operations which can be run in parallel.
- They will design distributed algorithms that can be deployed on a growing number of processors as the size of the problem increases, without sacrificing accuracy of results.
- They will identify mathematical optimization methods adapted to the problem.
- They will experiment parallel programming for optimization.

Quarter number

ST7

Prerequisites (in terms of CS courses)

- SG1 common course « Systèmes d'Information et Programmation » (1CC1000)
- ST2 common course « Algorithmique et complexité » (1CC2000)
- Basic knowledge in linear algebra

Syllabus

Architectures and Programming Models
- Introduction to computer architectures (types of architecture and parallelism)
- Sequential and parallel programming (Python) calling an external program (FORTRAN/C/C++) of high performance computing on multi-cores and on clusters
- Introduction to parallel computers

Parallel and distributed algorithms
- Introduction to parallel models and environments with message passing (MPI)
- Introduction to gradient algorithms
- Methodology of parallelization for the gradient algorithm
- Implementation on parallel computers

Domain decomposition methods
- Introduction to minimization algorithms in the context of optimization
- Partitioning techniques and methodology of parallelization
- Parallel Domain decomposition methods (Primal Schur method, Dual Schur method, Schwarz method, FETI method, optimized interface conditions)
- Minimization of communications

Genetic algorithm and meta-heuristics
- Introduction to optimization with meta-heuristics calling parallel kernels
- Parallelization of meta-heuristics based on local research (simulated annealing, tabu search, variables neighborhood search)
- Parallelization of meta-heuristics based on population estimation (genetic algorithms and colony optimization algorithms)
- Optimal allocation of ressources with meta-heuristics calling parallel kernels

Performance criteria
- Efficiency, strong and weak scalability, Amdhal’s law, Gustafson’s law, load balancing, granularity
- Illustration of performance losses and of code optimization

Class components (lecture, labs, etc.)
Lectures (25,5 hours) and tutorials (7,5 hours) with written final exam (1,5 hours).

Grading

Evaluation 100% with a written final exam split between : AA.1 and AA.3 evaluated by the first part of the exam, and AA.2 and AA.3 evaluated by the second part of the exam

Course support, bibliography

- Frédéric Magoulès, François-Xavier Roux, Guillaume Houzeaux. Parallel Scientific Computing. Wiley & Sons, Inc., 2015. Hardcover 354 pages (in English). This course support is also available in other languages: in French (Dunod, 2017), in Spanish (CIMNE, 2014), in Japanese (Morikita Publishing Co Ltd, 2015), in Hungarian (Pollack Press, 2018).
- Frédéric Magoulès, Stéphane Vialle. Parallel and Distributed Computing, Numerical Methods: Slides of the lectures

Resources

- Teachers : Filippo GATTI et Frédéric MAGOULES et Stéphane VIALLE
- Lectures and tutorials composed on group of 25 students working on computers.
- Access to various clusters (Data Center for Education of CentraleSupelec, and Mésocentre de CentraleSupelec-ENS Paris Saclay).
- Validation with standards languages: C/C++/Python, message passing interface library (MPI).

Learning outcomes covered on the course

AA.1 At the end of the lectures, students will be able to parallelize computation kernels, through domain decomposition methods, involved in optimization techniques (core skills C2.1 and C3.6).

AA.2 At the end of the lectures, students will be able to parallelize optimization methods based on genetic algorithms, and heuristics methods (core skills C1.3 and C3.6).

AA.3 At the end of the lectures, students will be able to implement parallelization techniques allowing to solve a problem in a limited time and where the sequential solution is not possible in a limited time (core skills C3.6).

Description of the skills acquired at the end of the course

C1.3 Apply problem-solving through approximation, simulation and experimentation. / Solve problems using approximation, simulation and experimentation

C2.1 Thoroughly master a domain or discipline based on the fundamental sciences or the engineering sciences.

C3.6 Evaluate the efficiency, feasibility and strength of the solutions offered. / proposed solutions