

# EC Algorithmique parallèle et répartie

Composante  
**Sciences Fondamentales et Appliquées**

Période de l'année  
**Semestre 8**

---

## Présentation

### Description

Cette UE présente les principaux patrons pour la programmation sur des environnements multiprocesseurs modernes, incluant les processeurs multicœurs, les grilles de calculs et les GPU. Cette présentation des patrons porte autant sur leurs principes de fonctionnement, que leur implantation. Plus en détails et dans le désordre, cette UE aborde les points suivants.

- 1 – Machine vectorielle ; machine à mémoire partagée vs mémoire séparée ; modèles théoriques de calcul parallèle : PRAM – Anneau / Grille / Tore de processeurs ; complexité d'un algorithme parallèle ; facteur d'accélération.
- 2 – Rappels sur les systèmes d'exploitation (processus, thread, instruction atomique, sémaphore, moniteur) ; programmation multithreads : modèle fork/join, pool de threads.
- 3 – Patrons de conception sur machine à mémoire partagée : Map, Gather, Scatter, Reduce, Scan, Compact, Segmented-Reduce, Segmented-Scan, Sort ... Implantation CPU via des threads et AVX.
- 4 – Programmation GPU : modèle de calcul CUDA, implantation des patrons de base (Map, Gather, Scatter, Reduce, Scan, Compact, Segmented-Reduce, Segmented-Scan).
- 5 – Programmation sur machine à mémoire séparée : patrons spécifiques liés à la communication entre nœuds ; implantations d'un anneau, d'une grille/d'un tore en MPI ; algèbre linéaire en MPI.
- 6 – Hybridation des différents niveaux de parallélisme possibles (multithread, GPU et distribution).

### Objectifs

Cette unité d'enseignement présente les outils permettant de résoudre n'importe quel problème exhibant un potentiel de parallélisme non nul. Le parallélisme vu ici est soit un moyen d'accélérer un algorithme donné, soit un moyen de permettre l'exécution de cet algorithme sur une machine parallèle construite via plusieurs ordinateurs reliés en réseau, et donc disposant de plus de mémoire que chacun des ordinateurs qu'elle relie.

L'objectif principal est la compréhension des patrons de conception pour ces machines parallèles, et la capacité à choisir le ou les patrons les plus adaptés pour résoudre un problème donné.

Par effet de bord, l'objectif secondaire est de savoir reconnaître si un algorithme donné peut être accéléré via du parallélisme (vectorisation, multithreading, GPU, grille de calculs), et si un problème trop complexe (en général en termes d'occupation mémoire) pour un seul processeur peut être résolu sur une grille.

Enfin, un troisième et dernier objectif est la compréhension fine des patrons sur différents environnements parallèles (simple ou bi CPU, GPU, HPC, hybrides), afin d'identifier en amont la pertinence de la parallélisation d'un problème, et de comprendre les performances obtenues.

## Heures d'enseignement

CM	CM	20h
TP	TP	30h

## Pré-requis obligatoires

Connaissances en programmation C++, algorithmique, programmation Système.