

3MD4110 - Programmation GPU

Responsables : **Stephane VIALLE**

Langues d'enseignement : **FRANCAIS**

Campus où le cours est proposé : **CAMPUS DE METZ**

Nombre d'heures d'études élèves (HEE) : **30**

Nombre d'heures présentielles d'enseignement (HPE) : **20**

Année académique : **2024-2025**

Niveau avancé : **non**

Présentation, objectifs généraux du cours :

Ce cours a pour objectif de présenter l'algorithmique et la programmation à haute performance sur GPU, avec des mises en oeuvre sur des algorithmes de Machine Learning, et sur des plates-formes de calculs équipées de GPU.

Période(s) du cours (n° de séquence ou hors séquence) :

SM11

Prérequis :

- Cours commun de 1A : Systèmes d'Information et Programmation (1CC1000)
- Cours commun de 1A : Algorithmique & Complexité (1CC2000)
- Cours de Programmation Avancée en C++ (3MD1020) de la mention SDI à Metz
- Cours d'Apprentissage Automatique (3MD1040) de la mention SDI à Metz

Plan détaillé du cours (contenu) :

- Architecture des GPU
- Principes algorithmiques du parallélisme de données à grain fin sur GPU (modèles SIMD et SIMT)
- Programmation en CUDA
- Utilisation de la bibliothèque CUBLAS
- Optimisations de codes GPU et CPU-GPU, en CUDA
- Conception et mise en oeuvre d'un algorithme de K-means sur GPU

Déroulement, organisation du cours :

Cours structuré en 3 parties :

- 2 parties de développement en CUDA (bases et concepts avancés), contenant chacune du cours et des mises en oeuvre sur machines
- une partie sur le développement en CUDA d'un algorithme de K-means, contenant un TD et des mises en oeuvre sur machines

Répartition globale : CM : 7h30, TD : 1h30, TP : 11h00 (total 20 HPE)

Plan du cours :

- Partie 1 : Architecture des GPU, et principes d'algorithmique et de programmation CUDA
 - CM : 3h00, TD : 1h30, TP : 3h00
- Partie 2 : Optimisation de codes CUDA, et utilisation de bibliothèques
 - CM : 4h30, TP : 3h00
- Partie 3 : K-means sur GPU
 - TP Part-1 : 3h00, TP Part-2 : 2h00

Organisation de l'évaluation :

Evaluation à partir des TP des parties 2 et 3

- **Comptes rendus des TP des parties 2 et 3** (le contenu et le nombre de pages des comptes rendus sont contraints, afin de forcer les étudiants à un effort de synthèse et de clarté)
- En cas d'absence non justifié à un TP la note de 0 sera appliquée, en cas d'absence justifiée la moyenne des autres TPs sera appliquée.
- L'examen de rattrapage sera un examen oral qui constituera 100% de la note de rattrapage.

Moyens :

- Equipe enseignante : **Stéphane Vialle**.
- Plateforme de développement et d'exécution : **serveurs de GPU du Data Center d'Enseignement du campus de Metz de CentraleSupélec**.
- Environnements de développement **CUDA** de NVIDIA.

Acquis d'apprentissage visés dans le cours :

- A la fin de ce cours les élèves seront en mesure :
- **AA1** : d'analyser l'adéquation d'une solution mathématique avec une implantation et une exécution sur GPU,
- **AA2** : de concevoir un algorithme sur GPU, ou d'adapter un algorithme pour qu'il soit plus efficace sur GPU,
- **AA3** : de concevoir des algorithmes hybrides CPU-GPU avec des recouvrement des transferts de données et des calculs
- **AA4** : d'implanter des algorithmes et de mettre au point des codes sur GPU
- **AA5** : d'analyser et de décrire synthétiquement des solutions sur GPU

Description des compétences acquises à l'issue du cours

:

Design of GPU algorithms and codes remains original, and needs:

- **to improve the following skills:**
 - **C6.3:** Conceive, design, implement and authenticate complex software
 - **Marker 3:** parallel software development (massively parallel software running on GPU)
 - Related to Learning Outcomes **AA2, AA3, AA4**
 - **C6.4:** Solve problems through mastery of computational thinking skills
 - **Marker 2:** exploitation of parallel architectures (fine grained and massively parallel GPU architecture)
 - Related to Learning Outcomes **AA1**
- **to know how to clearly introduce his/her original software approach to his/her colleagues:**

- **C7.1:** Persuade at core value level; to be clear about objectives and expected results. To apply rigour when it comes to assumptions and structured undertakings, and in doing so structure and problematise the ideas themselves. Highlight the added value
 - **Marker 1:** evaluated from a report or a solution introduction during classical learning
 - Related to Learning Outcomes **AA5**

Bibliographie :

- Slides de l'enseignant
- J. Sanders and E. Kandrot. "CUDA by Example: An Introduction to General-Purpose GPU Programming". NVIDIA. 2010.