

Module I2, Option Calcul Hautes Performances, Master Modélisation et Simulation (M2)

Responsable : Patrick Ciarlet

Important : Les cours de cette option sont au nombre de six. Il est **impératif** de suivre les trois premiers (nomenclature I2-1, I2-2 & I2-3) pour pouvoir suivre les trois suivants (nomenclature I2-a, I2-b, I2-c).

Par ailleurs, cette série de cours inclut des cours dispensés dans les modules A1 "Parallélisme et Calcul Réparti" et B13 "High Performance Computing" de troisième année de l'ENSTA.

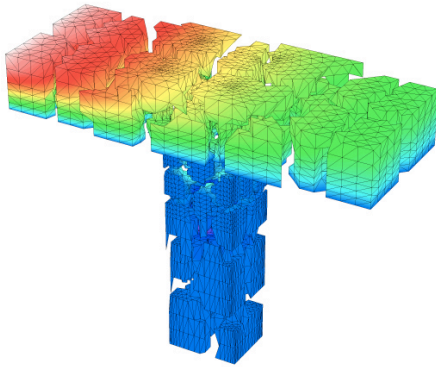
I. Intitulés et résumés des cours :

I2-1 : Aspects théoriques du calcul réparti.

Dans ce cours, on se propose de présenter quelques problèmes théoriques reliés à des problèmes classiques de calcul réparti, puis de les résoudre sur des applications pratiques.

Nous étudierons les classes de problèmes ci-dessous :

+ Placement de tâches : supposons que l'on dispose d'une architecture répartie, composée de processeurs et de liens de communication entre les processeurs, ainsi que d'un ensemble de tâches (programmes) à exécuter sur cette architecture. Le placement consiste à déterminer une affectation des tâches sur les processeurs, de façon à ce que celles-ci soient efficacement exécutées.



Partitionnement de maillage (vue éclatée)

+ Parallélisation automatique : dans ce cas, un programme informatique a été écrit, et l'on désire l'exécuter sur une architecture parallèle donnée. Plutôt que de le paralléliser 'à la main', c'est le compilateur qui est chargé de reconnaître les parties parallèles du programme, ce qui doit permettre de le paralléliser automatiquement. Il s'agit donc de comprendre les règles mises en œuvre dans le compilateur.

+ Partitionnement et équilibrage : on s'intéresse enfin au problème de la répartition des données, sur une architecture parallèle et/ou distribuée. Pour opérer une affectation convenable, il convient d'étudier a priori les interactions entre les données, ainsi que leur poids respectif. Ensuite, on peut les répartir, en tenant compte de ces deux paramètres. Voir la Figure de gauche.

Les applications abordées en TD ou TP auront pour thèmes principaux l'algèbre linéaire et le calcul scientifique. On considérera en particulier la résolution itérative de systèmes linéaires, à l'aide des méthodes de Jacobi et Gauss-Seidel. D'autres méthodes seront proposées au cours I2-a.

I2-2 : Initiation au calcul hautes performances.

L'objectif de ce cours est de fournir les bases pratiques du calcul parallèle permettant de développer des applications sur des machines parallèles et massivement parallèles. Le but recherché est d'exploiter au mieux la puissance des calculateurs à haute performance. L'enseignement comprendra quelques exposés théoriques et de nombreux travaux pratiques avec la progression suivante :

- + Initiation à la bibliothèque d'échange de messages MPI ;
- + Exercices sur un cluster de quelques dizaines de processeurs ;
- + Initiation au calcul haute performance (High Performance Computing) et optimisation.

I2-3 : Application de la programmation parallèle au calcul distribué.

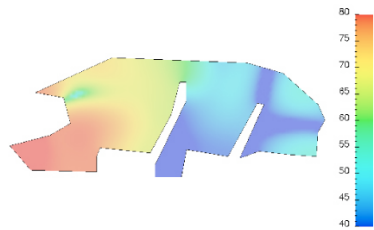
Ce cours se compose de six séances de travaux dirigés sur la machine parallèle du Laboratoire de Mathématiques Appliquées de l'ENSTA. Cette machine ainsi que son utilisation seront présentées en début de cours.

Les premières séances seront consacrées à la parallélisation de l'algorithme de décomposition de domaine par la méthode de Schwartz présenté lors du cours I2-a. La mise en œuvre de cette parallélisation reposera sur les notions introduites lors du cours I2-2 et sur de nouvelles fonctionnalités MPI enseignées dans ce cours.

Les séances suivantes consisteront en une introduction à la programmation à mémoire partagée avec OpenMP, et aux fonctions MPI-2 d'accès à la mémoire distante.

12-a : Méthodes de décomposition de domaine pour l'analyse numérique des EDP.

Pour la résolution des systèmes linéaires sur des architectures parallèles trois approches sont possibles :



Acoustique dans une voiture par DDM

- + Parallélisation des méthodes directes ;
- + Parallélisation des méthodes de type gradient conjugué via la parallélisation des produits matrices-vecteur ;
- + Mise en œuvre de méthodes de décomposition de domaine (DDM).

Ces dernières méthodes sont hybrides dans le sens où les méthodes directes sont utilisées dans les sous-domaines et le raccord des solutions entre les sous-domaines est réalisé de manière itérative par exemple à l'aide d'une méthode de gradient conjugué. On présentera les principes de base de ces méthodes et on fera le lien avec le cours I2-1. On programmera la méthode de Schwarz avec recouvrement. Dans le cours I2-3, on parallélisera le programme à l'aide la bibliothèque MPI.

12-b : Calcul parallèle avancé et "High Performance Computing".

La première partie du cours comprendra une introduction générale au calcul haute performance (High Performance Computing), incluant une présentation du panorama mondial, européen et français, et présentant les nombreux débouchés qui s'offrent aux experts de cette filière.

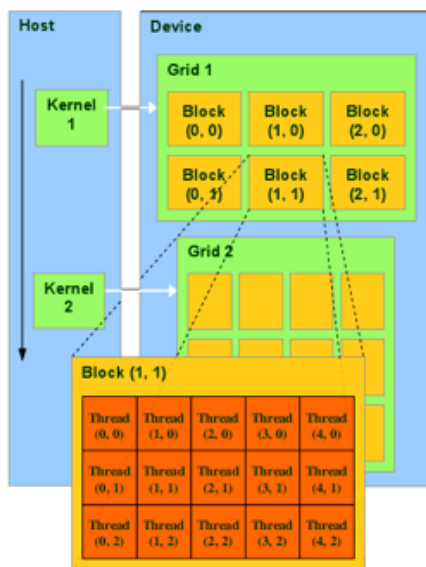
La deuxième partie du cours présentera la programmation OpenMP (Open Multi-Processing), interface de programmation pour le calcul parallèle sur architecture à mémoire partagée, devenant avec l'apparition des processeurs multicœurs une option de plus en plus utilisée afin de programmer les cœurs d'un même processeur. Cette interface de programmation applicative (API) est supportée sur de nombreuses plateformes, incluant Unix et Windows, pour les langages de programmation tels que C/C++ et Fortran.

La dernière partie du cours est consacrée à des scientifiques, qui viendront présenter leur domaine de recherche (astrophysique, climat, nanomatériaux,...), ainsi que leurs codes de simulations parallèles, en cours de développement ou en production, sur les plus grandes architectures de calcul du moment. Ils donneront un aperçu de la façon dont ils ont appréhendé la programmation parallèle en en présentant les principes ainsi que des exemples concrets de mise en œuvre.

12-c : Programmation hybride et multi-cœurs.

La course à la puissance des ordinateurs est désormais couplée avec une maîtrise de la consommation de l'énergie et l'impact environnemental. Ainsi, les machines sont désormais des supercalculateurs à plusieurs dizaines, voire centaines, de milliers de processeurs multi cœurs mais également couplés avec des accélérateurs (le meilleur exemple étant la machine la plus puissante au monde qui est une machine hybride). Ces nouvelles architectures amènent à repenser la façon dont les programmes de simulation sont écrits. Ainsi, on peut imaginer que l'utilisation de la seule bibliothèque MPI pourra être limitée par un trop grand nombre de tâches à gérer simultanément, et qu'il faut alors utiliser plusieurs niveaux de parallélisme.

Après une introduction rapide à ces évolutions d'architectures et de modèles de programmation, on présentera dans ce cadre la programmation "hybride" MPI + OpenMP dont le principe est d'utiliser des échanges explicites de messages entre processeurs (via MPI) et un modèle multi-thread à mémoire partagée (via OpenMP) entre les cœurs des processeurs qui partagent la même mémoire.



Modèle d'architecture CUDA

Dans une deuxième partie du cours, on présentera une autre façon de programmer des machines hybrides, qui utilisent des accélérateurs de calcul à base de processeurs de cartes graphiques (GPU). On rappellera comment en quelques années les GPU se sont transformés de puces dédiés aux fonctions graphiques en processeurs massivement parallèles (Single Instruction Multiple Threads) que l'on programme en langage haut-niveau (C/C++). A travers l'exemple de l'architecture CUDA (Compute Unified Device Architecture) de Nvidia, on présentera le modèle d'exécution et de programmation de ces processeurs et leurs limitations. On étudiera enfin quelques cas concrets d'accélération d'algorithmes sur GPU montrant comment exploiter au mieux leur puissance de calcul (jusqu'à 1TFlops par GPU de dernière génération).

Ces calculateurs hybrides sont de plus en plus utilisés, comme en témoigne l'installation au CEA/CCRT d'un supercalculateur d'une puissance de 100 Tflop/s en CPU auxquels viennent s'ajouter 200 Tflop/s en GPU.

II. Répartition des cours et affiliations des enseignants responsables :

I2-1 :

Localisation :

ENSTA Paris

Créneau horaire :

mardi matin, en septembre - octobre

Enseignant responsable :

Patrick Ciarlet

Enseignant-Chercheur

Laboratoire de Mathématiques Appliquées

ENSTA

I2-2 :

Localisation :

CEA Saclay, Orme des Merisiers

Créneau horaire :

vendredi matin, en septembre - octobre

Enseignant responsable :

Edouard Audit

Ingénieur-Chercheur

CEA/Direction des Sciences de la Matière

CEA Saclay

I2-3 :

Localisation :

ENSTA Paris

Créneau horaire :

mardi matin, en janvier - février

Enseignant responsable :

Fabrice Roy

Ingénieur de recherche CNRS

Laboratoire Univers et Théories, CNRS UMR 8102

Observatoire de Meudon

I2-a :

Localisation :

ENSTA Paris

Créneau horaire :

mardi matin, en octobre - novembre - décembre

Enseignant responsable :

Frédéric Nataf

Directeur de Recherche CNRS

Laboratoire Jacques Louis Lions, CNRS UMR 7598

Université Pierre et Marie Curie

I2-b :

Localisation :

ENSTA Paris

Créneau horaire :

mercredi matin, en novembre - décembre - janvier

Enseignant responsable :

Laurent Crouzet

Ingénieur

CEA/Direction des Sciences de la Matière

CEA Saclay

I2-c :

Localisation :

ENSTA Paris

Créneau horaire :

mercredi matin, en janvier - février - mars

Enseignants responsables :

Christophe Calvin

Chef de Laboratoire

CEA/Direction de l'Energie Nucléaire

CEA Saclay
Pierre Kestener
Ingénieur-Chercheur
CEA/Institut de Recherches sur les lois Fondamentales de l'Univers
CEA Saclay

NB. Chaque cours est composé de six séances, de 3h ou 3h30 chacune.