

Programmation Parallèle

Le calcul intensif en fonction des architectures

Sophie Robert

UFR ST Pôle info

Le déroulement du module

Le contenu

- 15h CM Intro, Mémoire distribuée (MPI), Mémoire partagée (OpenMP), Vectorisation
- 14h de TP (utilisation de EasyPAP pour aider à comprendre)
- 6h de TD (algorithmique)
 - Intervenants : Allan Blanchard (sous réserve) Sébastien Rivault et Sophie Robert

La modalité d'évaluation

- Sous forme de contrôle continu avec projet ou TP noté.

Programmation Parallèle ?

Calcul Intensif

De nombreuses applications nécessitent du calcul intensif avec une contrainte de temps pour obtenir les résultats.

Une puissance de calcul sans cesse accrue

04/10/2021

Début 2021, Météo-France a déployé son nouveau système de calcul intensif. La solution retenue repose sur la plateforme Sequana XH2000, développée par Bull (filiale du groupe Atos) et fabriquée à Angers. Le facteur de gain de puissance de calcul est de 5,5, permettant des prévisions météorologiques plus précises géographiquement et dans le temps.

Des progrès à différents niveaux

La mise en œuvre de ce nouveau supercalculateur est un enjeu majeur pour Météo-France en tant que centre météorologique national et international de référence. Cette acquisition va notamment permettre progressivement :

- d'améliorer la prévision des phénomènes dangereux avec un gain de 1 à 2 heures d'échéance sur les prévisions ;
- d'améliorer la précision géographique et donc mieux déterminer les risques, en descendant à une échelle infra-départementale (résolution horizontale amenée à 1,3 km) ;
- de prendre en compte un nombre croissant d'observations et de nouveaux types d'observations tels que les objets connectés ;

Programmation Parallèle ?

Calcul Intensif

De nombreuses applications nécessitent du calcul intensif avec une contrainte de temps pour obtenir les résultats.



Programmation Parallèle ?

Effectuer beaucoup de calculs en utilisant plusieurs cœurs ou processeurs

- Le principe du parallélisme est simple
 - * Exécuter en même temps des instructions indépendantes
- La mise en œuvre nécessite de connaître
 - * Les architectures des machines parallèles
 - * Les techniques de parallélisation
 - * Les techniques de programmation

Introduction à l'algorithmique et la programmation parallèles

- Savoir **paralléliser** un problème en fonction de la cible
- Savoir mettre en œuvre en fonction du paradigme de parallélisation

Applications d'aide à la décision

Data-mining

- Gestion d'énormes bases de données
- Exploration efficace de gros volumes de données Exemples : marketing, bio-informatique ...

⇒ **nécessitent de répartir les données**

Simulations

- phénomènes physiques
- phénomènes biochimiques
Exemple : dynamique moléculaire pour la conception de molécules d'intérêt thérapeutique
- phénomènes environnementaux
Exemples : météo, inondations, diffusion des pesticides

⇒ **nécessitent une grande puissance de calcul**

Introduction intuitive du parallélisme

Corrections de copies

- des copies : **les données**
- des exercices constitués de x questions à corriger : **les tâches**
- des enseignants capables tous de corriger **1 question par minute**

Introduction intuitive du parallélisme

Les principes associés

- le processus
- l'équilibrage de charge, l'accélération et l'efficacité
- le parallélisme de tâches ou de données ?
- la mémoire et les coûts d'accès
- la mémoire partagée ou distribuée

Les modèles du parallélisme

Qu'est ce qui est indépendant ?

- Le parallélisme de contrôle
 - * Faire plusieurs choses en même temps
- Le parallélisme de données
 - * Répéter une action sur des données similaires
- Le parallélisme de flux
 - * Travailler à la chaîne

La classification de Flynn

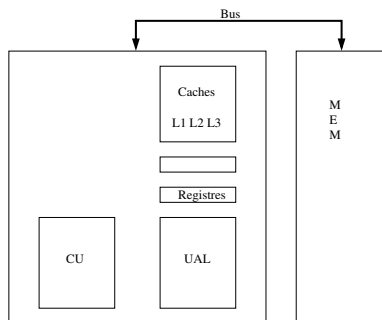
En fonction du flot d'instructions et de données

- Single Instruction Single Data (SISD)
- Single Instruction Multiple Data (SIMD)
- Multiple Instruction Multiple Data (MIMD)

La classification de Flynn

En fonction du flot d'instructions et de données

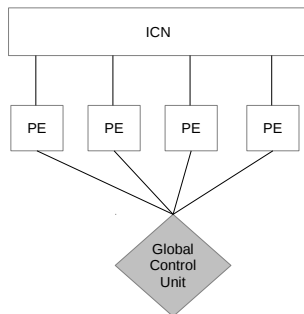
- Single Instruction Single Data (SISD)
- Single Instruction Multiple Data (SIMD)
- Multiple Instruction Multiple Data (MIMD)



La classification de Flynn

En fonction du flot d'instructions et de données

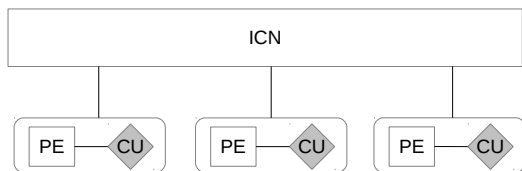
- Single Instruction Single Data (SISD)
- Single Instruction Multiple Data (SIMD)
- Multiple Instruction Multiple Data (MIMD)



La classification de Flynn

En fonction du flot d'instructions et de données

- Single Instruction Single Data (SISD)
- Single Instruction Multiple Data (SIMD)
- Multiple Instruction Multiple Data (MIMD)



L'Inter Connection Network

Accès à la mémoire

L'ensemble des mécanismes pour le transfert de données entre processeurs ou entre les processeurs et les modules de mémoire.

Espace d'adressage partagé

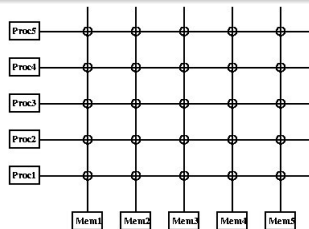
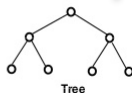
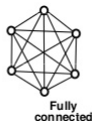
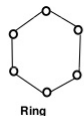
- Uniform Memory Access (UMA)
- Non Uniform Memory Access (NUMA)
 - * Cache Coherent NUMA (ccNUMA)
 - * noCache Coherent (ncNUMA)

Espace d'adressage distribué (Message Passing Platform)

- Chaque processeur a son propre espace d'adressage
- Les échanges sont explicites par envoi/réception.

L'Inter Connection Network

- static pour du point-à-point, tous les éléments sont reliés entre eux
 - * Completed Connected Networks (CCN)
 - * Limited Connected Networks (LCN)
- dynamic en utilisant des switches
 - * exemple du crossbar



Classification simplifiée

- Architecture à mémoire partagée
- Architecture à mémoire distribuée
- Architecture hybride

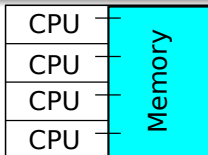
Classification simplifiée

- Architecture à mémoire partagée

- * Tous les processus partagent le même espace mémoire
- * Cela semble facile à programmer car on n'a pas à se soucier de ce que chaque processus peut voir
- * Cependant il faut gérer dans le code les accès concurrents en mémoire ce qui est complexe (synchronisations)
- * En pratique on a de grandes pertes de performances si les données ne sont pas bien gérées, c'est en fait très compliqué

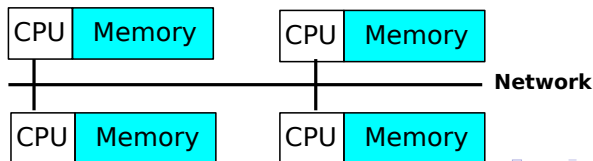
- Architecture à mémoire distribuée

- Architecture hybride



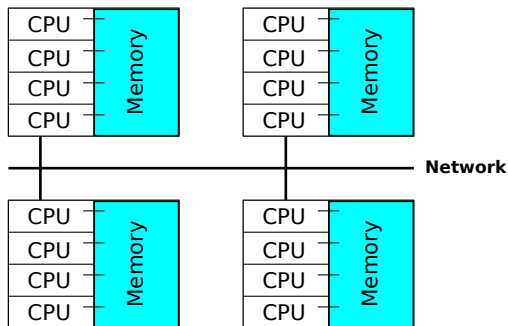
Classification simplifiée

- Architecture à mémoire partagée
- Architecture à mémoire distribuée
 - * Chaque processus est indépendant en mémoire
 - * Il faut gérer la distribution des données aux différents processus
 - * Il faut éventuellement introduire des communications pour échanger des informations utiles entre processus
 - * Le travail semble plus important, mais il est en réalité moins complexe dans des programmes de taille importante
- Architecture hybride



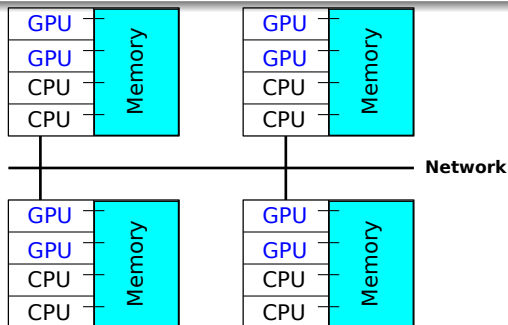
Classification simplifiée

- Architecture à mémoire partagée
- Architecture à mémoire distribuée
- Architecture hybride
 - * C'est encore plus compliqué !



Classification simplifiée

- Architecture à mémoire partagée
- Architecture à mémoire distribuée
- Architecture hybride
 - * On mélange la programmation sur architecture à mémoire partagée et distribuée !



Le présent et l'avenir c'est le parallélisme !

Comment obtenir plus de puissance ?

- **Machines personnelles** avec plus de processeurs et de cœurs
- **Cluster** : ensemble de machines homogènes et localisées
- **Massively Parallel Processing(MPP)** : Machine spécialisée à mémoire distribuée
- **Grid** : ensemble de ressources hétérogènes et dé-localisées (peut contenir des clusters)
- **Cloud** : un parc de machines, d'équipements de réseau et de logiciels maintenu par un fournisseur, que les consommateurs peuvent utiliser en libre service via un réseau informatique
- ...

L'algorithme des k -moyennes

Les entrées

- le nombre de classes k
- un ensemble de n objets

Les sorties

- Un ensemble de k classes d'objets qui minimise le critère des moindres carrés

$$E = \sum_{i=1}^k \sum_{p \in C_i} d(p, m_i)^2$$

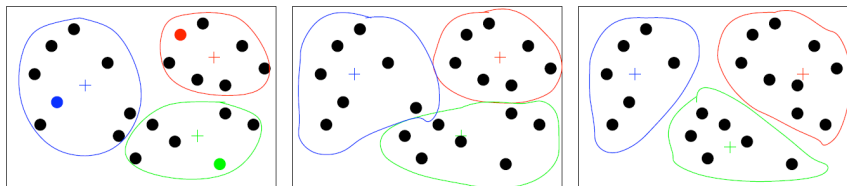
où $d(p, m_i)$ est la distance entre les points p et m_i le centre de la classe C_i .

Une application marketing

Algorithme des k -moyennes

- choisir arbitrairement k objets comme centres des classes
- répéter
 - * affecter chaque objet à la classe dont il est le plus similaire au sens de la distance choisie.
 - * calculer la valeur moyenne des objets pour chaque classe qui devient le nouveau centre

jusqu'à convergence



Données numériques

Suppositions sur les données

- 6 millions de clients dans la base de données
- un client caractérisé par un tableau de 500 nombres flottants

Objectif

Partitionner la base de données en 60 classes distinctes à partir desquelles les profils types seront construits.

Suppositions sur l'algorithme des k -moyennes

- le temps de calcul de la distance entre un profil type et un client nécessite 1000 opérations flottantes
- l'algorithme converge en 200 étapes.

Suppositions sur la machine “séquentielle”

Caractéristiques

- un processeur 1 cœur
 - * 1 milliard d'opérations sur des nombres flottants par seconde (1 GFlop/s)
- un disque dur
 - * 3,33 millions de nombres flottants par seconde entre la mémoire vive et le disque dur en lecture et écriture
- la mémoire vive
 - * 1 milliard de nombres flottants (64bits) en plus du système d'exploitation et de l'application (environ 8 Go)

Suppositions sur la machine “séquentielle”

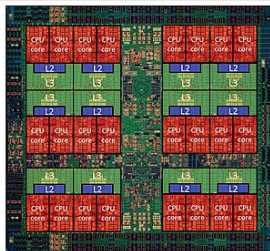
Caractéristiques

- FLOPS (Flop/s) : Nombre d'opérations à virgule flottante par seconde
 - FPU Floating-Point Unit
 - LINPACK comme référence
 - nombre de cœurs \times fréquence \times nombre de cycles pour une opération flottante

Processeurs multi-cœurs

Architecture à mémoire partagée

- Les processeurs actuels ont de 2 à 4 cœurs avec la possibilité d'avoir deux processeurs sur la même carte mère
- Ils partagent la mémoire vive
- Un programme classique (en Java, C, ADA, Ocaml) utilise un seul cœur, il faut une programmation spécifique pour qu'un programme utilise plusieurs cœurs



HPC / IA

Grappe de calculs

Machine à mémoire distribuée

- Plutôt que d'avoir une machine avec plusieurs cœurs, on considère maintenant plusieurs machines connectées par un réseau
- Chaque processeur a sa propre mémoire qui ne peut être accédée directement par un autre processeur : on parle de machine à mémoire distribuée
- Un exemple : les grappes de PCs

Grappe : Configuration

Hypothèses

- La grappe est constituée de 8 machines séquentielles (mêmes caractéristiques)
- Le réseau est de type Gigabit Ethernet avec
 - * une latence pour l'établissement d'une communication de $4 \times 10^{-5} \text{s}$
 - * un débit pour une communication point-à-point de 10^7 float/s.
- Les données initiales se trouvent sur un seul disque.

Parallélisation

Questions à se poser

- Machine à mémoire partagée :
 - * comment répartir les calculs ?
 - * comment limiter ou gérer les accès concurrents aux données
- Machine à mémoire répartie :
 - * comment répartir les données ?
 - * comment répartir les calculs ?