



Introduction au Calcul Parallèle

Master I RID

Introduction

- Le parallélisme est une stratégie pour accélérer les temps d'exécution de tâches de grande taille. Les tâches sont initialement décomposées en sous-tâches de taille plus petite et celles-ci sont ensuite assignées à plusieurs processeurs qui les exécutent simultanément. Devant les limites physiques de la taille de la mémoire et de la vitesse de calcul des systèmes monoprocesseurs, le parallélisme apparaît comme une solution compétitive pouvant résoudre Les problèmes nécessitant de grandes capacités de calcul et la manipulation de grandes quantités de données

Taxonomie des architectures parallèles et distribuées

- La multiplication des architectures comportant plusieurs processeurs entraîne une certaine confusion dans les terminologies utilisées dans la littérature particulièrement lorsqu'il s'agit de caractériser les systèmes parallèles et distribués

Systemes distribués vs parallèles

- **Un système parallèle** est un système composé de plusieurs processeurs fortement couplés, localisés sur la même machine, connectés par des liens de communication point à point à base de switch ,et qui travaillent à la résolution d'un même problème. Les multiprocesseurs UMA (Uniform Memory Access) et les multiprocesseurs NUMA (Non Uniform Memory Access) appartiennent à cette catégorie.

Systemes distribués vs parallèles

- **Un système distribué** est un système composé de plusieurs processeurs physiquement distribués et faiblement couplés, reliés entre eux par un réseau conventionnel de communication (ATM, FDDI, Token Ring, Ethernet, etc.) et qui collaborent à la résolution d'un ou plusieurs problèmes. Internet, les intranets, les systèmes embarqués, les systèmes mobiles et les systèmes de téléphonie appartiennent à cette catégorie.

La Classification de Flynn

- Il existe de nombreuses méthodes de classification des architectures parallèles et distribuées dont celle de Flynn
- Les programmes et les architectures sont classés selon le type d'organisation du flux de données et du flux d'instructions.

La Classification de Flynn

SISD Single Instruction, Single Data Ordinateur séquentiel	SIMD Single Instruction, Multiple Data Ordinateur vectoriel en pipeline
MISD Multiple Instruction, Single Data Tableau systolique	MIMD Multiple Instruction, Multiple Data Multiprocesseur et Multi ordinateurs

SISD (Single instruction, Single Data)

- Les machines les plus simples traitent une donnée à la fois : ces systèmes sont dits « séquentiels ». Ce type de fonctionnement était prédominant pour les ordinateurs personnels jusqu'à la fin des années 1990

SIMD (Single Instruction, Multiple Data)

- Les systèmes traitant de grandes quantités de données d'une manière uniforme ont intérêt à être des **SIMD** (Single Instruction, Multiple Data) ; c'est typiquement le cas des processeurs vectoriels où des unités de calcul gèrent le traitement du signal comme la vidéo ou le son.

MISD (Multiple Instruction, Single Data)

- Le type MISD a été beaucoup plus rarement utilisé, il semble néanmoins adapté à certains problèmes comme les réseaux neuronaux et aux problèmes temps réel liés. L'architecture appelée Systolic array est un type d'architecture **MISD**.

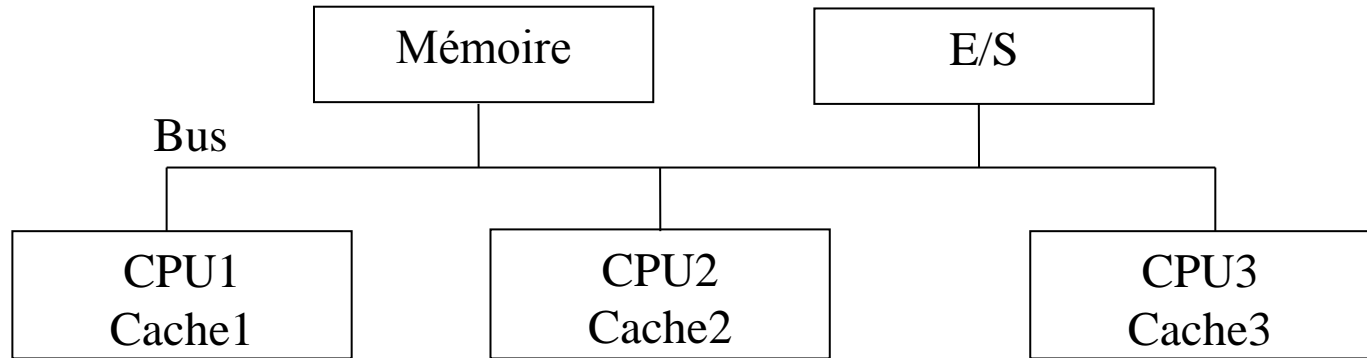
MIMD (Multiple Instruction, Multiple Data)

- Les systèmes utilisant plusieurs processeurs ou un processeur multi-cœur sont plus polyvalents et pleinement parallèles,
- Aujourd'hui les machines SIMD sont abandonnées au profit des machines MIMD qui offrent la souplesse nécessaire à la résolution de certains problèmes. Une classification plus fine des machines MIMD essentiellement basée sur l'organisation de la mémoire et les types d'interconnexion permet de distinguer deux grandes familles : **les multiprocesseurs et les multi-ordinateurs**

Les multiprocesseurs

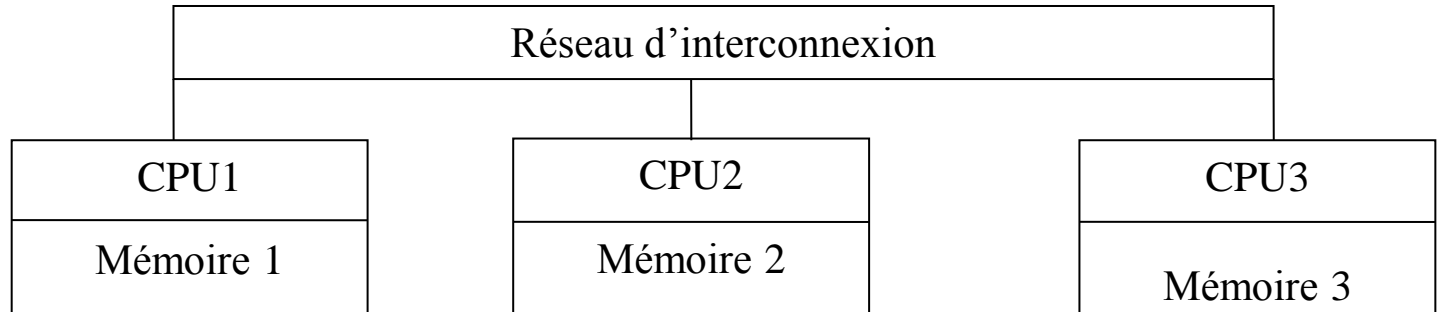
- Un multiprocesseur est un système informatique dans lequel deux ou plusieurs processeurs partagent l'accès total à une mémoire commune. Les processeurs, la mémoire et les périphériques d'entrée et sortie sont reliés par un réseau d'interconnexion pouvant prendre la forme d'un bus commun, ou d'un réseau multi-étages. Il n'y a qu'une seule copie du système d'exploitation qui tourne sur ce type d'architecture. Les multiprocesseurs sont de deux types : **Les machines UMA** et **les machines NUMA**

Les machines UMA



- Toute opération effectuée sur la mémoire est immédiatement visible par l'ensemble des autres processeurs.
- Les machines UMA ont l'avantage d'être faciles à programmer
- Cependant ces systèmes sont peu scalables, car au delà de quelques dizaines de processeurs il devient très difficile de maintenir la cohérence des caches et un temps d'accès uniforme à la mémoire.

Les Machines NUMA

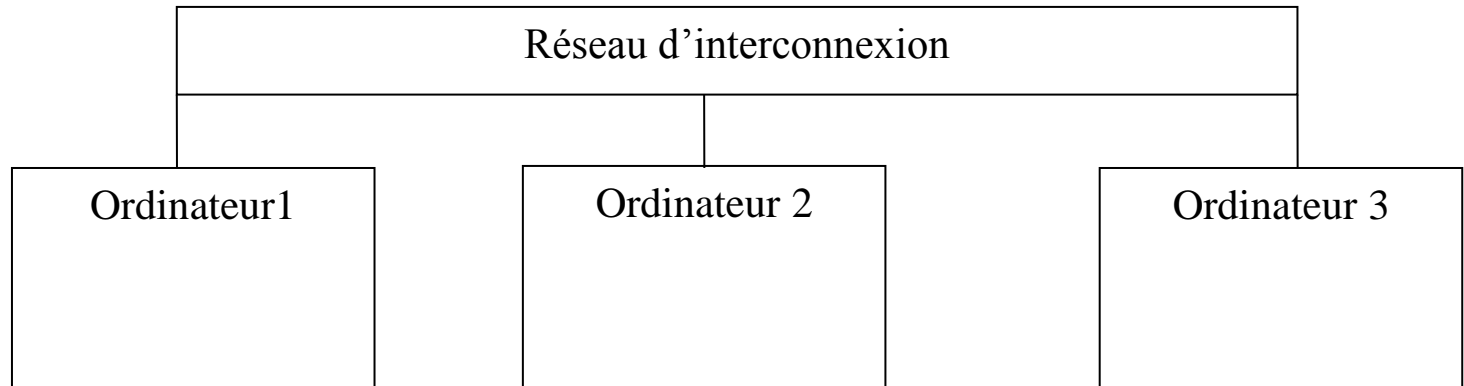


- Les machines NUMA sont des systèmes à mémoire partagée. Cependant ils ont la particularité d'avoir une mémoire physiquement distribuée, ce qui fait varier le temps d'accès en fonction de l'emplacement mémoire.
- Chaque processeur dispose d'une mémoire locale. L'ensemble de toutes ces mémoires forme un espace d'adressage global et unique, accessible à tous les processeurs.

Les Multi-ordinateurs

- Un système multi-ordinateur est constitué par un ensemble de machines indépendantes reliées entre elles par un réseau d'interconnexion. Chaque machine possède une mémoire privée à laquelle elle accède directement. Les mémoires distantes ne sont pas directement adressables et les données sont partagées en envoyant explicitement ou en recevant des messages.

Les Multi-ordinateurs



- Les multi-ordinateurs se répartissent en deux catégories : **les multi-ordinateurs homogènes et les multi-ordinateurs hétérogènes.**

les multi-ordinateurs homogènes

- Parfois appelés MPPs (Massively Parallel Processors), ils sont constitués de processeurs identiques reliés par un réseau d'interconnexion unique qui utilise la même technologie partout. Ces systèmes sont particulièrement scalables puisqu'ils ne demandent pas un dispositif logiciel ou matériel pour maintenir la cohérence des données. Cependant, ils sont relativement difficiles à programmer à cause de la charge supplémentaire due au placement et au partitionnement des données entre les différents processeurs, ainsi que les latences élevées dans certains types de réseaux.

Les multi-ordinateurs hétérogènes

- Il s'agit de plusieurs ordinateurs aux caractéristiques matérielles et logicielles différentes, connectés à travers des réseaux utilisant des technologies différentes. Il n'y a pas une vision globale du système, les performances et les services offerts ne sont pas les mêmes partout pour les applications (à cause de l'hétérogénéité). Un tel système est relativement facile à construire et à étendre. De plus, la puissance sans cesse croissante des ordinateurs, leur faible coût ainsi que les progrès technologiques réalisés dans les réseaux haut débit font que ce type de machine ait un excellent rapport coût/performance même si le partage des ressources pose des problèmes de sécurité et nécessite un dispositif logiciel qui masque l'hétérogénéité de l'environnement.

Comparaison

<u>Architecture à mémoire distribuée</u>	<u>Architecture à mémoire partagée</u>
Temps d'accès à la mémoire court	Temps d'accès à la mémoire dépendant de la charge
Grand nombre des processeurs	Petit nombre des processeurs
Communication spécifique	Communication par boîte à message
Architecture flexible et scalable	Architecture peu modifiable
Architecture d'interconnexion spécifique	Interconnexion par bus central
Petites mémoires distribuées	Grande mémoire centralisée

La programmation Parallèle

- La programmation parallèle consiste à développer des programmes qui s'exécutent simultanément sur plusieurs processeurs à la fois. Un modèle de programmation parallèle est une abstraction du matériel que le programmeur utilise pour coder et exécuter des programmes parallèles.
- Les modèles de programmation parallèle les plus populaires sont le modèle à mémoire partagée, et le modèle à échange de messages.

Le modèle à mémoire partagée

- Le modèle à mémoire partagée permet d'exprimer de manière simple et puissante le parallélisme dans une application. Dans ce modèle, plusieurs processus partagent un espace mémoire commun où ils peuvent lire et écrire des données de manière asynchrone. Pour éviter les conflits d'accès aux données partagées, des mécanismes de synchronisation sont nécessaires.

Le modèle à mémoire partagée

- Le modèle à mémoire partagée est implémenté dans une large variété de multiprocesseurs (SMP (Symmetrical Multiprocessor) et machines NUMA). Le regain d'intérêt pour les multiprocesseurs a poussé plusieurs constructeurs à définir un standard qui est OpenMP.

Le modèle à mémoire partagée

- OpenMP (Open Multi-Processing) est une interface de programmation pour le calcul parallèle sur architecture à mémoire partagée, cette API est supportée sur de nombreuses plateformes, incluant Unix et Windows, pour les langages de programmation C/C++ et Fortran. Il se présente sous la forme d'un ensemble de directives, d'une bibliothèque logicielle et de variables d'environnement.

Le modèle à échange de messages

- Dans ce modèle, un programme est constitué par un ensemble de processus opérant chacun dans un espace d'adressage privé. Les processus communiquent par échange explicite de messages. Un système d'échange de messages demande un schéma d'adressage qui repère les processus les uns par rapport aux autres et deux primitives de base (Send/Receive) permettant le transfert de données entre processus.

Le modèle à échange de messages

- Le modèle à échange de messages est largement utilisé dans les machines à mémoire distribuée (multi-ordinateurs) mais il est aussi implémenté sur des machines à mémoire physiquement partagée (machines NUMA) avec de bonnes performances. L'ensemble des opérations de communication autorisées par une implémentation du modèle à échange de messages est disponible à travers des bibliothèques et des langages de programmation classiques.

Le modèle à échange de messages

- MPI (Message Passing Interface) et PVM (Parallel Virtual Machine) sont les bibliothèques les plus utilisées dans le modèle de programmation à échange de messages. Elles supportent les langages C, C++ et Fortran. Elles permettent d'écrire de manière simple et efficace des programmes parallèles s'exécutant sur n'importe quelle plateforme.
- Elle fournit des routines pour la communication et la gestion des processus.

Le modèle à échange de messages

- Il existe des implémentations libres respectant les spécifications de MPI dont MPICH et LAM. MPICH est une implémentation complète de MPI pour une large variété de systèmes parallèles et distribués.
- MPI a été écrite pour obtenir de bonnes performances aussi bien sur des machines massivement parallèles à mémoire partagée que sur des clusters d'ordinateurs hétérogènes à mémoire distribuée.

Le modèle du parallélisme de données

- Cet autre modèle exploite le parallélisme dérivant de l'application de la même opération à tous les éléments d'un ensemble de données. Les processus exécutent le même code mais ils opèrent sur des données différentes.
- Le parallélisme de données est utilisé aussi bien dans les systèmes à mémoire partagée que dans les systèmes à mémoire distribuée. Les implémentations de ce modèle sont souvent des extensions des langages de programmation séquentielle existants (Java, C++, Fortran, etc.).

Le modèle du parallélisme de données

- HPF (High Performance Fortran) est une extension du langage Fortran. Il est le langage le plus utilisé pour écrire des programmes en utilisant le parallélisme de données.
- HPF fournit aussi aux programmeurs une bibliothèque de fonctions particulièrement précieuses pour le calcul haute performance

Conclusion

- Ce Cours présente un certain nombre des divers aspects de calcul parallèle. C'est ainsi qu'après avoir donné les motivations pour ce type de calcul, nous avons présenté une taxonomie des architectures parallèles et distribuées, et les principaux modèles de programmation parallèle.
- le parallélisme offre de multiples avantages (puissance de calcul, disponibilité, tolérance aux pannes, flexibilité, etc.)