

AU: 2024-2025

Programmation concurrente et parallèle

Imene LAHYANI

Introduction

- Définition du parallélisme
- Définition de la programmation parallèle
- Intérêt de la programmation parallèle
- Machines parallèles
- challenge

-
- **Définition du parallélisme**
 - Définition de la programmation parallèle
 - Intérêt de la programmation parallèle
 - Machines parallèles
 - challenge

Définition du parallélisme (1/2)

- Le parallélisme consiste à l'occurrence **simultanée** (parallèle) **d'événements** dans **un système informatique**.
- Un événement signifie généralement l'un des suivants:
 - Une opération arithmétique
 - Une opération logique
 - Un accès mémoire
 - Exécution Entrée ou sortie (E/S)

Définition du parallélisme (2/2)

- Le parallélisme peut être examiné à **plusieurs niveaux**:
 - **Niveau Job**: plusieurs jobs indépendants fonctionnent simultanément sur le même système informatique.



Niveau job

Niveau programme

Niveau Instruction

Niveau bit

Définition du parallélisme (2/2)

- Le parallélisme peut être examiné à plusieurs niveaux:
 - **Niveau Job**: plusieurs jobs indépendants fonctionnent simultanément sur le même système informatique.
 - **Niveau programme**: plusieurs tâches sont exécutées simultanément pour résoudre un problème commun.

Niveau programme

- Prog sequentiel

Tache 1;

Tache 2;

Tache 3;

.....

- Prog sequentiel

x=x+1;

y=y+2;

z=z-5;

.....



- Prog parallèle

Tache 1// tache2// tache 3.....

Définition du parallélisme (2/2)

- Le parallélisme peut être examiné à plusieurs niveaux:
 - **Niveau Job**: plusieurs jobs indépendants fonctionnent simultanément sur le même système informatique.
 - **Niveau programme**: plusieurs tâches sont exécutées simultanément pour résoudre un problème commun.
 - **Niveau instruction**: le traitement d'une instruction, comme l'ajout de deux nombres, peut être divisée en plusieurs sous-instructions. Si plusieurs instructions similaires doivent être exécutées leurs sous-instructions peuvent être superposées en utilisant une technique appelée pipeline.

Niveau Instruction

- Inst 1 //

- Inst 2



sou_inst_1; sou_inst_'1 //

sou_inst_2; sou_inst_'2



sou_inst_1; sou_inst_2; sou_inst_'1; sou_inst_'2

Pipeline

Définition du parallélisme (2/2)

- Le parallélisme peut être examiné à plusieurs niveaux:
 - **Niveau Job**: plusieurs jobs indépendants fonctionnent simultanément sur le même système informatique.
 - **Niveau programme**: plusieurs tâches sont exécutées simultanément pour résoudre un problème commun.
 - **Niveau instruction**: le traitement d'une instruction, comme l'ajout de deux nombres, peut être divisée en plusieurs sous-instructions. Si plusieurs instructions similaires doivent être exécutées leurs sous-instructions peuvent être superposées en utilisant une technique appelée pipeline.
 - **Niveau bit**: lorsque les bits d'un mot sont traités l'un après l'autre, il s'agit d'une opération de bits en série. Si les bits sont sollicités en parallèle, il s'agit donc d'une opération de bits en parallèle.

-
- Définition du parallélisme
 - **Définition de la programmation parallèle**
 - Intérêt de la programmation parallèle
 - Machines parallèles
 - challenge

Définition de la programmation parallèle (1/2)

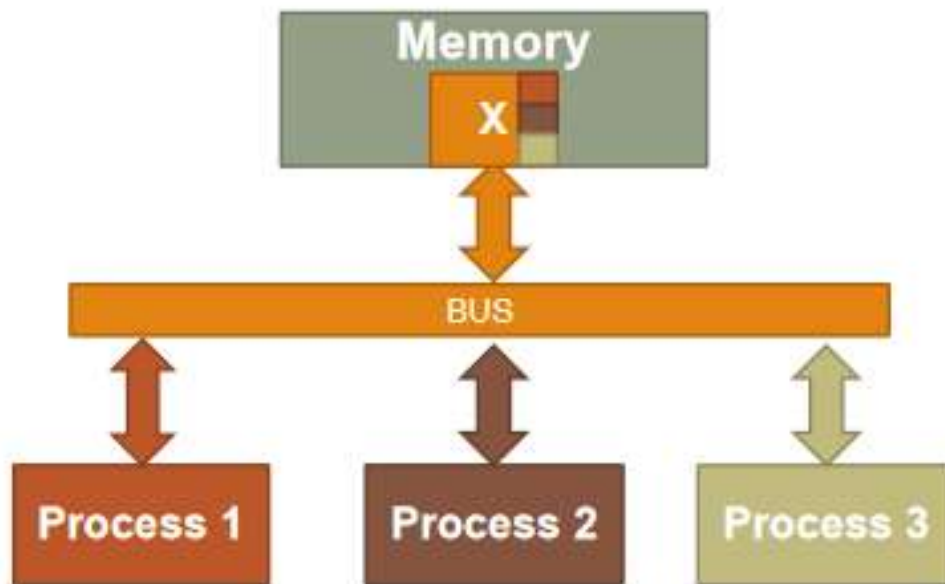
- La **programmation parallèle** est l'activité de construire **un programme parallèle**
 - Un programme parallèle est composé de **plusieurs processus** qui s'exécutent « **simultanément** »
 - Un programme séquentiel est composé d'une suite d'instructions = **un seul processus**

Définition de la programmation parallèle (1/2)

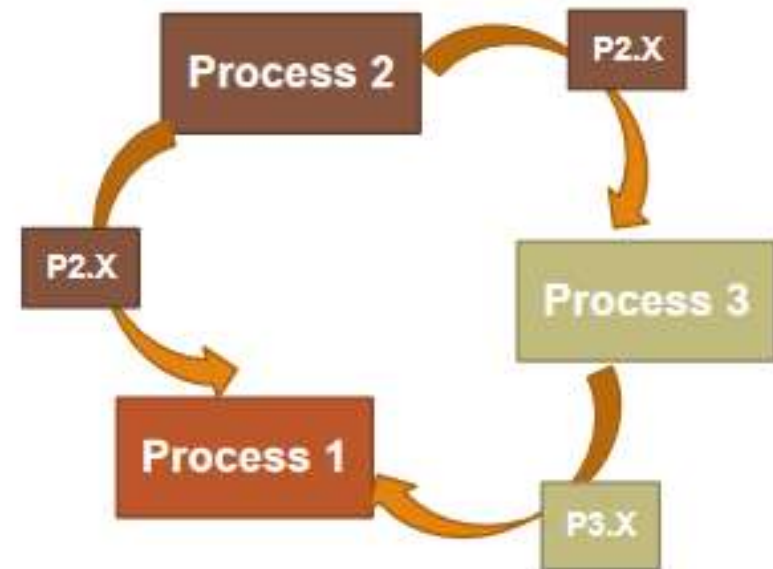
- Les processus
 - **coopèrent** pour effectuer une tâche bien déterminée
 - Sont en **compétition** pour la possession de ressources (le processeur)
- La coopération entre les processus se fait par la communication des données en utilisant
 - **soit des mémoires partagées**
 - en écrivant et lisant des variables partagées
 - **soit des canaux.**
 - en émettant et recevant des messages via des canaux

Définition de la programmation parallèle (2/2)

Partage de la mémoire



Passage de messages



-
- Définition du parallélisme
 - Définition de la programmation parallèle
 - **Intérêt de la programmation parallèle**
 - Machines parallèles
 - challenge

Intérêt de la programmation parallèle (1/2)

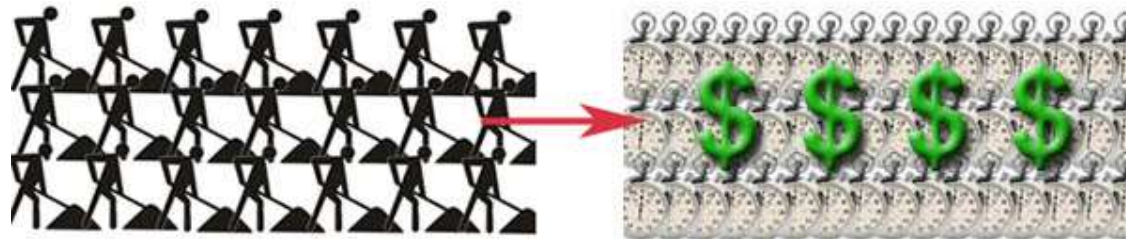
- La vitesse des processeurs ne peut pas augmenter infiniment car limitation physique (taille des transistors)
- les applications sont de plus en plus demandeuses en puissance de calcul



- **Solution:**
 - Des algorithmes plus intelligent
 - Exploiter au mieux les ressources de la machine

Intérêt de la programmation parallèle (2/2)

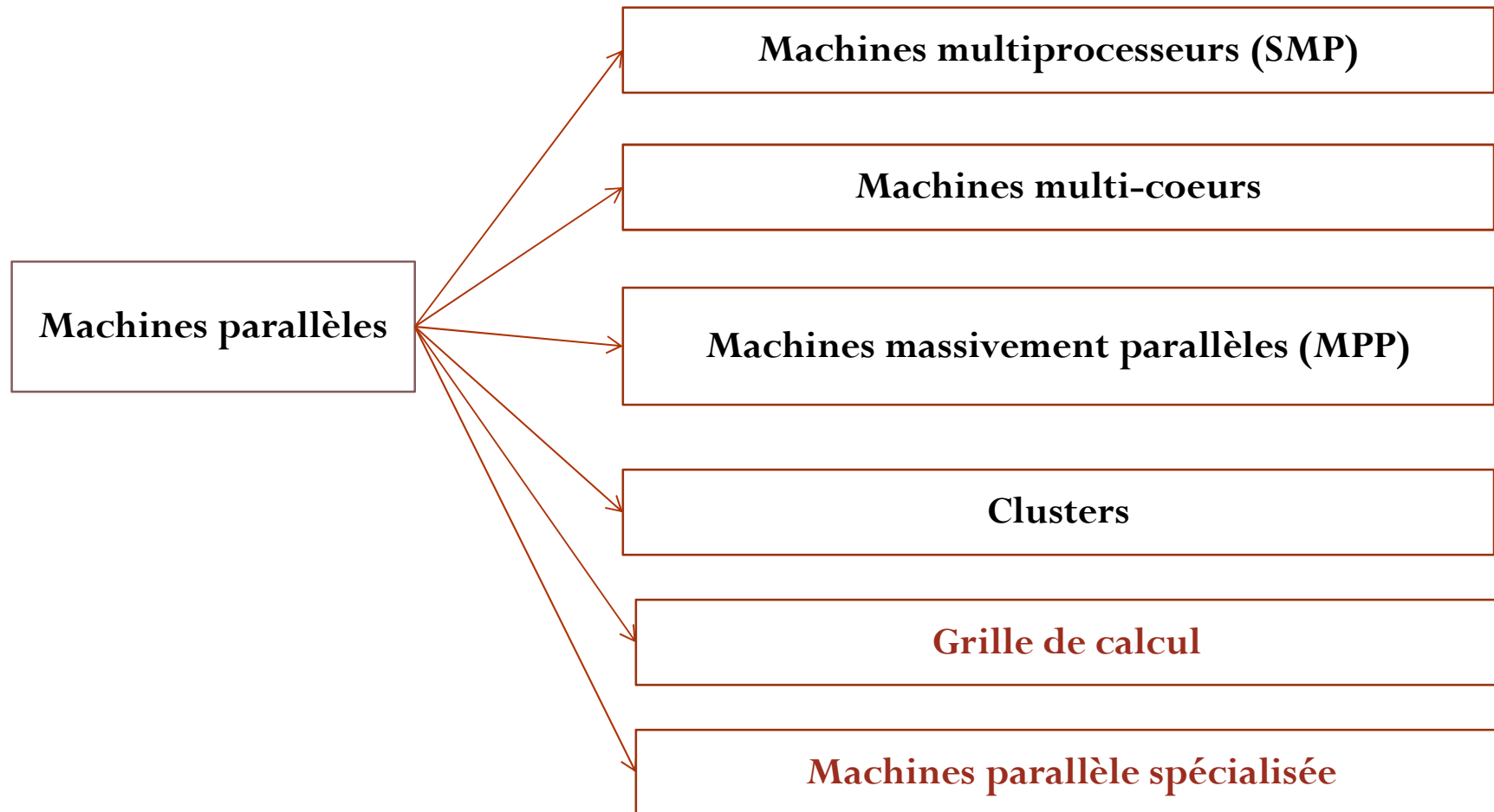
- La programmation parallèle:
 - 1 Permet d'exécuter plusieurs opérations en même temps en dupliquant les éléments de calcul (processeurs)
 - Gagner du **temps** et/ou de **l'argent**: faire coopérer de **nombreux** éléments de calculs de « **faible** » **puissance** et de **moindre coût**



- 2 Permet aussi une meilleure utilisation du processeur grâce à la multiprogrammation

-
- Définition du parallélisme
 - Définition de la programmation parallèle
 - Intérêt de la programmation parallèle
 - **Machines parallèles**
 - Propriétés des processus concurrents

Classification des machines parallèles



-
- Définition du parallélisme
 - Définition de la programmation parallèle
 - Intérêt de la programmation parallèle
 - Machines parallèles
 - **challenge**

Challenge

- La programmation parallèle est plus difficile que la programmation séquentielle.
- Cause:
 - Un nombre énorme d'exécutions possibles
 - non déterministe
 - Les programmes sont plus difficiles à écrire (!)
 - Les programmes sont plus difficiles à déboguer(!)
 - Les erreurs ne sont pas toujours reproductibles (!)
 - De nouveaux types d'erreurs possibles (!):
 - interblocage, famine, inversion de priorité, interférence ...

Exemple d'erreurs (1/2)

- Un exemple simple:

(x: variable partagée, initialisée à 1)

♦ process 1: x := x + 1

♦ process 2: x := x - 1

- Quel est le résultat final?
- Généralement, une instruction n'est pas atomique.
- Traduction des instructions en actions atomiques:
 - ♦ Process 1: reads x (R1), writes to x (W1) .
 - ♦ Process 2: reads x (R2), writes to x (W2) .
- 6 exécutions possibles et 3 valeurs possibles de x : **2, 0, et 1**.
- Le programme n'est pas déterministe car le résultat de son exécution varie d'une exécution à une autre.

Exemple d'erreurs

- Un exemple reel:
 - Therac – 25: une machine de radiothérapie.
 - Entre 1985 et 1987, le Therac-25 fut impliqué dans au moins six accidents durant lesquels des patients reçurent des doses massives de radiation, qui a engendrer leurs décès.



- La cause directe du dysfonctionnement de la machine était d'ordre informatique: erreur de programmation parallèle