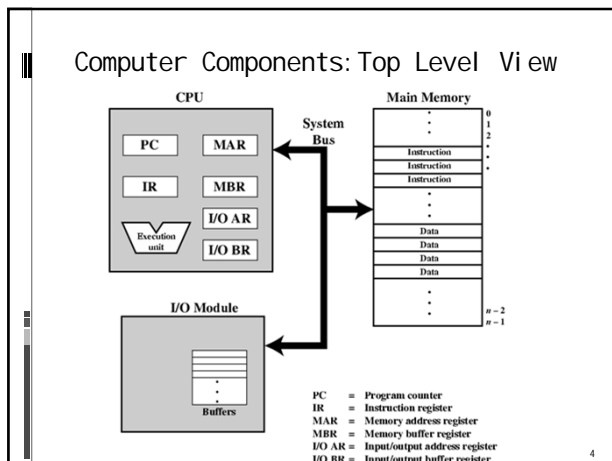


Jean-luc.dekeyser@lifl.fr
Version 2013

MODÈLES D'EXÉCUTION



Supports de cours

- **Organisation et architecture de l'ordinateur**
William Stallings - Sixième Edition
Pearson Education
ISBN : 2-7440-7007-6
- Programme de licence et Master
- <http://www.hpcresearch.nl/euroben/reports/web11/overview.php>

Type de fonctionnement

- Le processeur exécute (interprète) des instructions élémentaires
- Une séquence d'opérations peut décrire tous les problèmes

notion de Programme

PARALLÉLISME INTRINSÈQUE

Extension du modèle Von Neumann

Déroulement du programme

- Le déroulement du programme est contrôlé par le Compteur Ordinal qui pointe vers la prochaine instruction à exécuter.
- Les instructions sont exécutées en séquence sauf en cas de saut. (Jump)

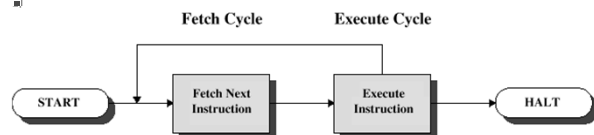
The given bus widths are examples only!

Les données

- La mémoire conserve les données et les instructions, on parle de mémoire banalisée.
- Les instructions sont amenées une à une vers le processeur
- Les échanges entre mémoire / processeur se font via le bus

Cycle du processeur

- L'exécution d'une instruction peut être découpée en plusieurs phases successives.
- Deux phases sont définies: la phase de chargement 'Fetch' et la phase d'exécution 'Execute'



Séquence des instructions

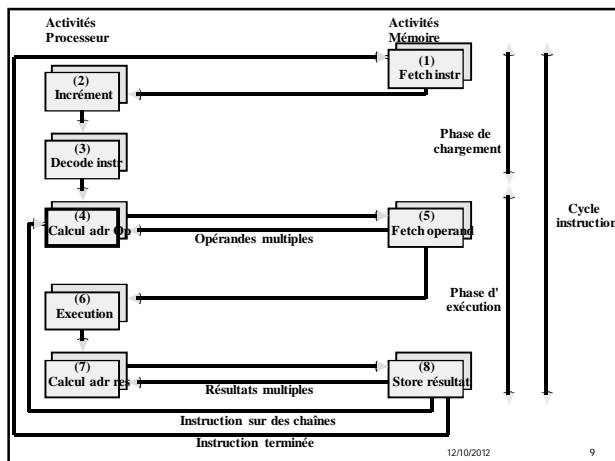
- L'exécution d'une instruction passe par plusieurs étapes successives, chacune étant considérée comme une micro-opération.
- Certaines de ces actions correspondent à une activité mémoire, d'autres à une activité processeur.
- Pour effectuer une instruction, il faut toujours effectuer les actions suivantes

Parallèle ou séquentiel

- Parallélisme est inhibé par le modèle d'exécution séquentiel

Forcer une représentation séquentielle de problèmes présentant un certain degré de parallélisme entraîne nécessairement une perte d'efficacité dans l'amélioration des performances

Vers une explicitation parallèle des algorithmes donc des programmes



TYPES DE PARALLÉLISME

parallélisme

Modèles de programmation vs Modèles de fonctionnement

- **Modèle de fonctionnement (d'exécution) est lié à l'architecture de la machine**
 - Il caractérise la façon dont sont exécutées les instructions élémentaires.
 - Diverses classifications différencient ces modes de fonctionnement.

12/10/2012

13

Mise en oeuvre du parallélisme

- **1) Le fonctionnement du processeur**
 - Le déroulement du cycle de l'instruction prend en compte l'anticipation des instructions suivantes: principe du pipeline

12/10/2012

16

Niveaux du parallélisme

- **Modèle de programmation est lié à la traduction de l'algorithme**
 - Il caractérise la méthode de parallélisation d'algorithmes utilisée

12/10/2012

14

Mise en oeuvre du parallélisme(suite)

- **2) La hiérarchie du système mémoire**
 - Des mémoires rapides intermédiaires qui mémorisent dynamiquement des données et instructions qui seront réutilisées par le programme

12/10/2012

17

Développements architecturaux

- **Solutions architecturales pour la mise en oeuvre du parallélisme intrinsèque du modèle séquentiel**
- **Mise en oeuvre du parallélisme à différents niveaux d'exécution**

12/10/2012

15

Mise en oeuvre du parallélisme(suite)

- **3) L'interface Processeurs / Mémoires**
 - Elle doit permettre d'assurer un transfert des données efficace entre un ou plusieurs processeurs et une ou plusieurs mémoires.

12/10/2012

18

Mise en oeuvre du parallélisme (suite)

- 4) Les systèmes Multi-Processeurs
 - Par l'association de plusieurs unités de calcul indépendantes, on augmente le potentiel en terme de puissance de calcul.

12/10/2012

19

Classification de Flynn

		Flux d'instructions	
		Un	Plusieurs
Flux de données	Un	SISD	MISD
	Plusieurs	SIMD	MIMD

12/10/2012

22

Classifications

- Flynn 72
 - Flux data et Instruction
- Flux d'instruction : séquence d'instruction exécutée par la machine.
- Flux de données : séquence des données appelées par le flux d'instructions

12/10/2012

20

Les classes de Flynn

- **SISD**: La plupart des ordinateurs actuels (Von Neumann)
- **SIMD**: Array processeur - même instruction sur des données différentes (Data parallélisme)

12/10/2012

23

Les caractéristiques de Flynn

- **S Single** **I Instruction**
- **M Multiple** **D Data**

12/10/2012

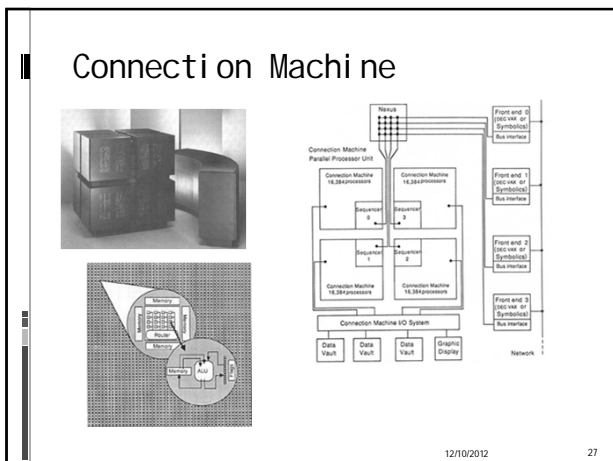
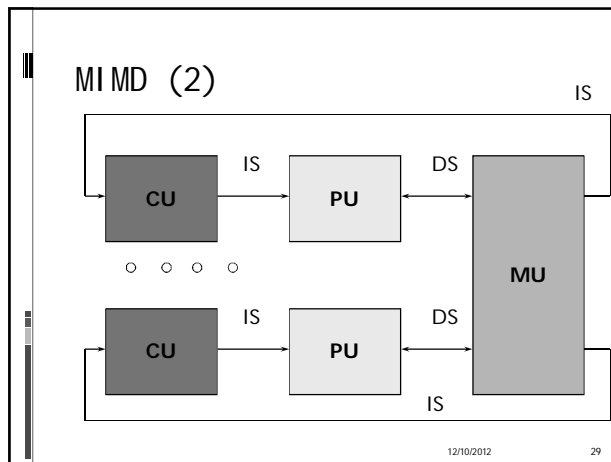
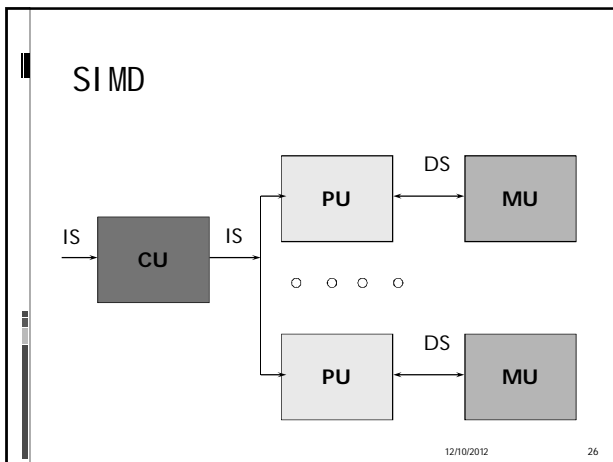
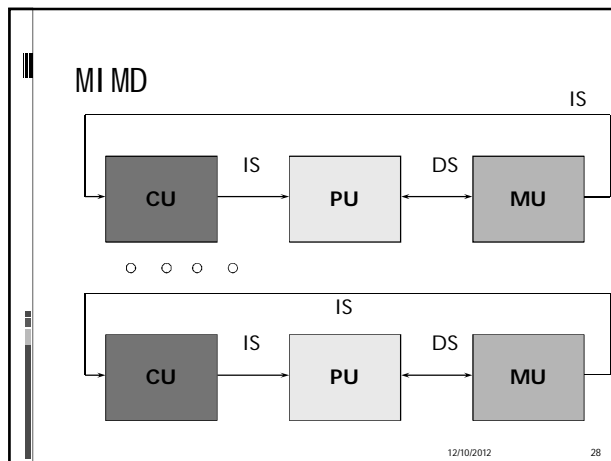
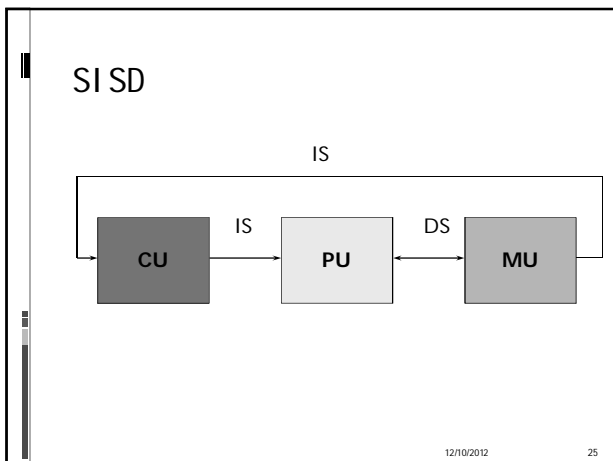
21

Les classes de Flynn

- **MISD**: n PU recevant des instructions différentes mais avec le même Flux de données (ou un Flux dérivé). Notion de macro-pipe-line, Machine systolique. Pas de machine commerciale de ce type !
- **MIMD**: n PU se partagent l'accès à une mémoire unique ou multiple.
- **SPMD**: Même programme sur des données différentes.
 - SIMD < SPMD < MIMD

12/10/2012

24



Mémoire partagée / mémoire distribuée

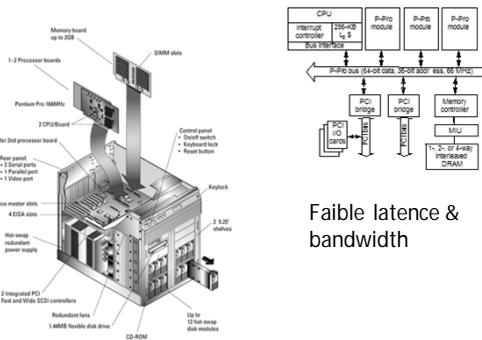
- La mémoire peut être physiquement construite par un ensemble de bancs mémoires visibles par l'ensemble des processeurs.

12/10/2012 30

Constructi on

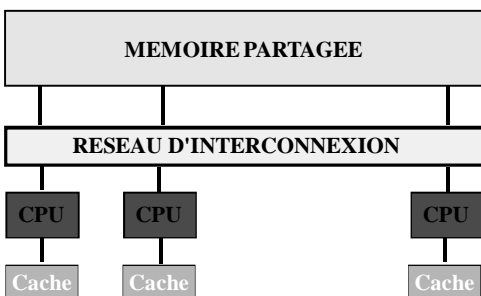
- Dans ce cas un réseau d'alignement permet d'associer au même instant un certain nombre de bancs et de processeurs.
 - La mémoire physiquement partagée est nécessairement logiquement partagée.

Intel Pentium Pro Quad



Faible latence & bandwidth

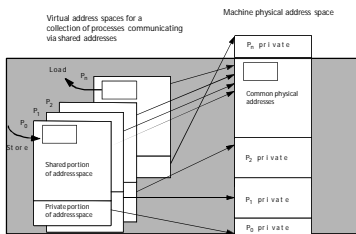
Mémoire partagée



Mémoire partagée / mémoire distribuée

- Elle peut être construite par association de l'ensemble des mémoires de chaque processeur.
- Dans ce cas un réseau d'interconnexion relie l'ensemble des processeurs.

Espace d'adressage partagé

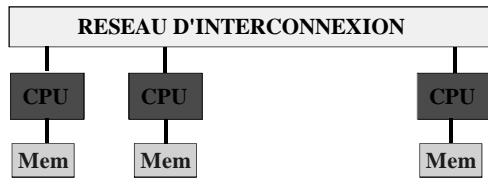


- Code système parallèle add hoc
- Nombreuses applications parallèles

Constructi on

- Les données externes au processeur sont véhiculées par ce réseau.
 - La mémoire physiquement distribuée peut être logiquement distribuée ou logiquement partagée.

Mémoire distribuée



12/10/2012

37

UMA

- Partage uniforme de l'accès mémoire par tous les processeurs
- Temps d'accès égal pour chaque mot de la mémoire
- On parle de système fortement couplé: haut degré de partage de ressources

12/10/2012

40

Classification / machines parallèles

- **SIMD:**
 - mémoire partagée
 - ?
 - mémoire distribuée
 - DAP
 - Connection machine
 - MasPar (USTL)

12/10/2012

38

UMA(suite)

- Utilisation d'un bus commun, d'un cross-bar, d'un réseau multi-étage
- Communication et synchronisation se font via des variables partagées en mémoire

12/10/2012

41

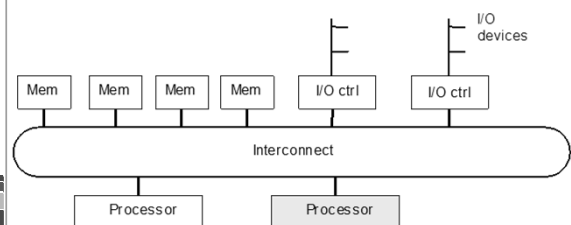
Multiprocesseur à mémoire partagée

- **Trois modèles existent**
 - **UMA : Uniform-Memory-Access**
 - **NUMA : Non-Uniform-Memory-Access**
 - **COMA : Cache-Only-Memory-Architecture**

12/10/2012

39

UMA



Sequent Symmetry S-81

12/10/2012

42

NUMA

- Le temps d'accès dépend de la location de l'information en mémoire.
- 2 exemples

12/10/2012 43

NUMA Exemple 2

- On peut utiliser un système hiérarchisé de mémoires.
 - Chaque processeur possède sa mémoire locale
 - Les processeurs sont regroupés en cluster.
 - Une mémoire est globalement partagée entre tous les clusters.

12/10/2012 46

NUMA:exemple 1

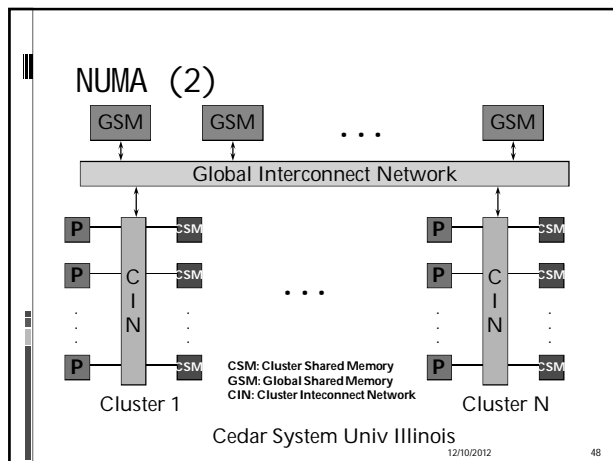
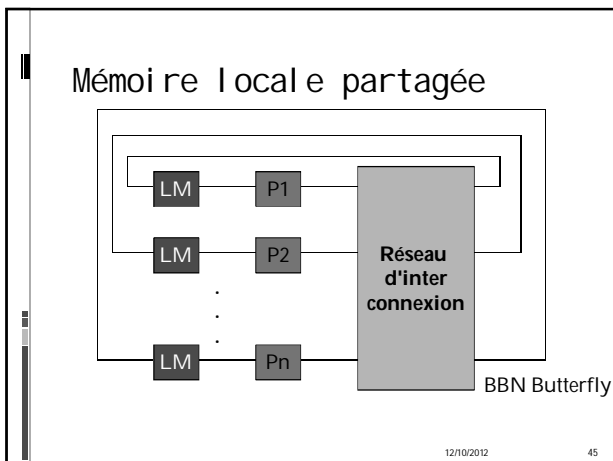
- Une mémoire partagée peut être physiquement distribuée.
- L'adressage de la mémoire est global mais chaque processeur possède sa mémoire locale.
- Les accès à des adresses en mémoire locale sont plus rapides que ceux aux mémoires d'autres processeurs.

12/10/2012 44

NUMA 2

- L'accès le plus rapide est local,
- Puis l'accès à la mémoire globale
- Enfin l'accès à une mémoire locale d'un autre processeur.

12/10/2012 47



COMA

- Le modèle COMA est un cas particulier du modèle NUMA.
- Les mémoires locales sont remplacées par des mémoires caches.
- L'ensemble des caches forment l'espace d'adressage.

12/10/2012

49

Autres modèles

- D'autres modèles ont été définis pour telle ou telle machine
- CC-NUMA : Cache Coherent Non Uniform Memory Access utilise une mémoire distribuée et des caches sur chaque processeur. (Ex Dash Stanford Whang Chap 9)

12/10/2012

52

COMA (suite)

- L'accès aux caches externes est assuré par un catalogue (directory) distribué sur l'ensemble des processeurs. (Voir le cours sur les mémoires caches)
- Les données vont migrer lors du calcul en fonction du processeur qui les utilise.

12/10/2012

50

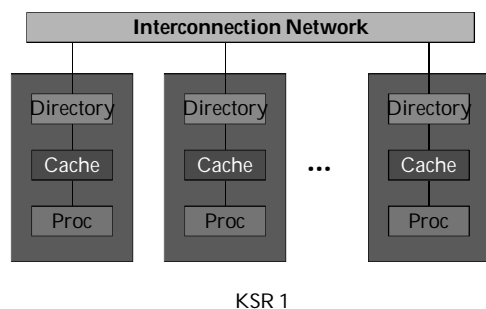
Multicomputer à mémoire distribuée

- Un multicomputer est composé d'un certain nombre de noeuds interconnectés par un réseau à passage de messages.
 - Il permet d'établir des liaisons point à point entre les noeuds de la machine.
 - Les communications et synchronisations se font par l'envoi de messages entre les noeuds.

12/10/2012

53

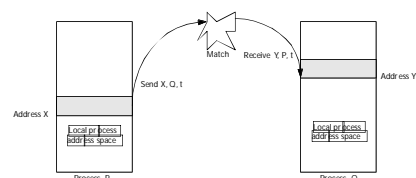
COMA



12/10/2012

51

Message-Passing Abstraction



- Send spécifie un buffer à envoyer
- Recv spécifie une zone de rangement pour réception
- Mémoire à mémoire, identifier les processus
- Send/recv demande une synchronisation
- Beaucoup d'overhead: copie, buffer, protection

12/10/2012

54

Mémoire des multi-computers

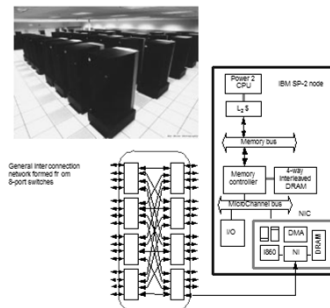
- Il n'y a pas d'accès à une mémoire autre que celle du noeud.
- Les mémoires locales sont dites **mémoire privée**. On parle alors de modèle **NORMA: No Remote Memory Access**.
 - De nouvelles machines proposent actuellement des architectures équivalentes mais qui propose des mémoires partagées distribuées. (Cray T3D)

12/10/2012

55

IBM SP-2

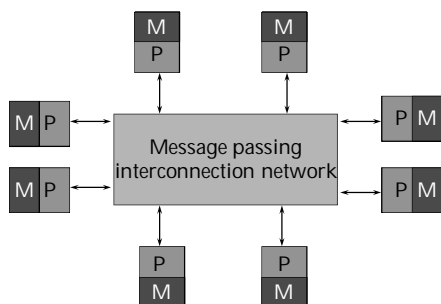
- Construction à partir de WS RS6000
- Network interface intégrée au I/O bus (bw limité par le I/O bus)



12/10/2012

58

Multi-computer message passing



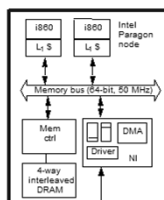
12/10/2012

56

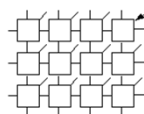
Intel Paragon



Sandia's Intel Paragon XP/S-based Supercomputer



2D grid network with processing node attached to every switch



12/10/2012

57