

LES ARCHITECTURES PARALLÈLES

Plan

2

1. Avènement du parallélisme
2. Objectifs du parallélisme
3. Classifications
4. Evaluation des performances
5. Réseaux d'interconnexion

Avènement du parallélisme

3

Historiquement, un logiciel était écrit pour un **traitement séquentiel** :

- ❖ Pour être exécuté sur une machine avec une seule unité de calcul
- ❖ Un calcul était découpé en une série discrète d'instructions
- ❖ Les instructions étaient exécutées les unes après les autres
- ❖ Une seule instruction pouvait s'exécuter à un instant donné

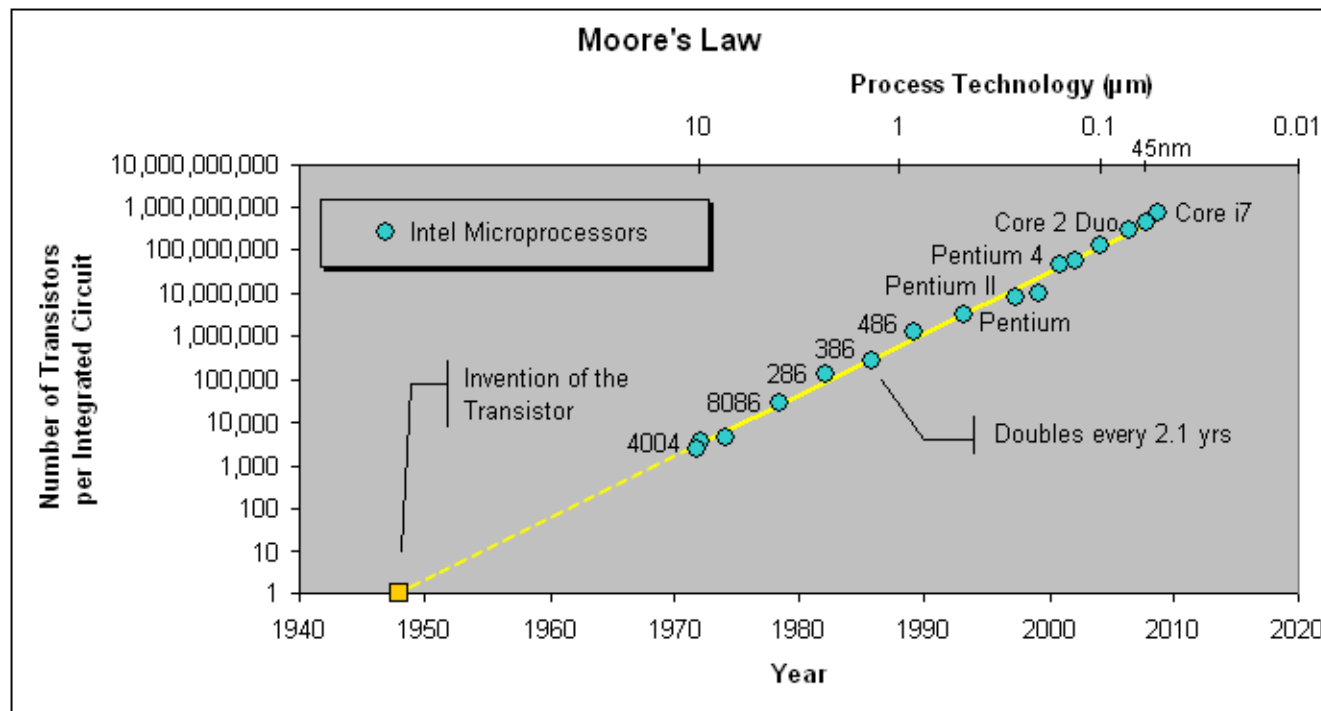
Avènement du parallélisme

4

Loi de Moore :

Le nb de transistors par circuit intégré se double tous les 2 ans

Exprimée en 1965 par Gordon Moore, co-fondateur d'Intel, la "Loi de Moore" fête ses 50 ans et reste d'actualité.



Avènement du parallélisme

5

Loi de Moore :

Le nb de transistors par circuit intégré se double tous les 2 ans

- ❖ Limitation due à la mémoire
- ❖ Limites physiques : consommation, chaleur
- ❖ Limitations des technologies de gravures

Avènement du parallélisme

6

Loi de Moore :

nb de transistors par circuit intégré se double tous les 2 ans

- ❖ Limitation due à la mémoire
- ❖ Limites physiques : consommation, chaleur
- ❖ Limitations des technologies de gravures

Solution :

- ❖ Faire plus avec moins
- ❖ Augmenter le nombre d'élément de calcul
- ❖ Exploiter le concept de **parallélisme**

Avènement du parallélisme

7

En effet, l'apparition du parallélisme est liée au :

❖ **Besoin en puissance de calcul :**

- **Demande continue pour plus de vitesse d'exécution des programmes**
- **Pb : il existe toujours une limite technologique (**loi de Moore**)**

❖ **Besoin de réduction des coûts :**

- **L'accroissement de la puissance d'un élément de calcul entraîne une augmentation des coût**
- **Pb : diminuer les coûts**

Avènement du parallélisme

8

En effet, l'apparition du parallélisme est liée au :

❖ **Besoin en puissance de calcul :**

- **Demande continue pour plus de vitesse d'exécution des programmes**
- **Pb : il existe toujours une limite technologique (**loi de Moore**)**
- **Sol : Parallélisme**

❖ **Besoin de réduction des coûts :**

- **L'accroissement de la puissance d'un élément de calcul entraîne une augmentation des coût**
- **Pb : diminuer les coûts**
- **Sol : coopérer des éléments de calcul à faible coût (Parallélisme)**

Avènement du parallélisme

9

Top 500

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	Titan - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945
6	Swiss National Supercomputing Centre (CSCS) Switzerland	Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115,984	6,271.0	7,788.9	2,325
7	King Abdullah University of Science and Technology Saudi Arabia	Shaheen II - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Aries interconnect Cray Inc.	196,608	5,537.0	7,235.2	2,834
8	Texas Advanced Computing Center/Univ. of Texas United States	Stampede - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell	462,462	5,168.1	8,520.1	4,510
9	Forschungszentrum Juelich (FZJ) Germany	JUQUEEN - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	458,752	5,008.9	5,872.0	2,301
10	DOE/NNSA/LLNL United States	Vulcan - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	393,216	4,293.3	5,033.2	1,972

Avènement du parallélisme

10

La calcul parallèle

- ❖ Un calcul parallèle est exécuté sur plusieurs unités de calcul
- ❖ Le calcul est découpé en plusieurs parties pouvant s'exécuter simultanément
- ❖ Chaque partie est découpée en séries d'instructions
- ❖ Des instructions de chaque partie s'exécutent simultanément sur différents CPU

Avènement du parallélisme

11

La calcul parallèle

Les ressources de calcul peuvent être :

- ❖ **Une seule machine avec plusieurs processeurs**
- ❖ **Plusieurs machines avec un ou plusieurs processeurs interconnectées par un réseau rapide**
- ❖ **Une grille de calcul : plusieurs ensembles de plusieurs machines**

On peut donc améliorer le temps d'exécution en utilisant plusieurs processeurs à la fois

Avènement du parallélisme

12

Domaines

Quelques exemples qui nécessitent la calcul parallèle :

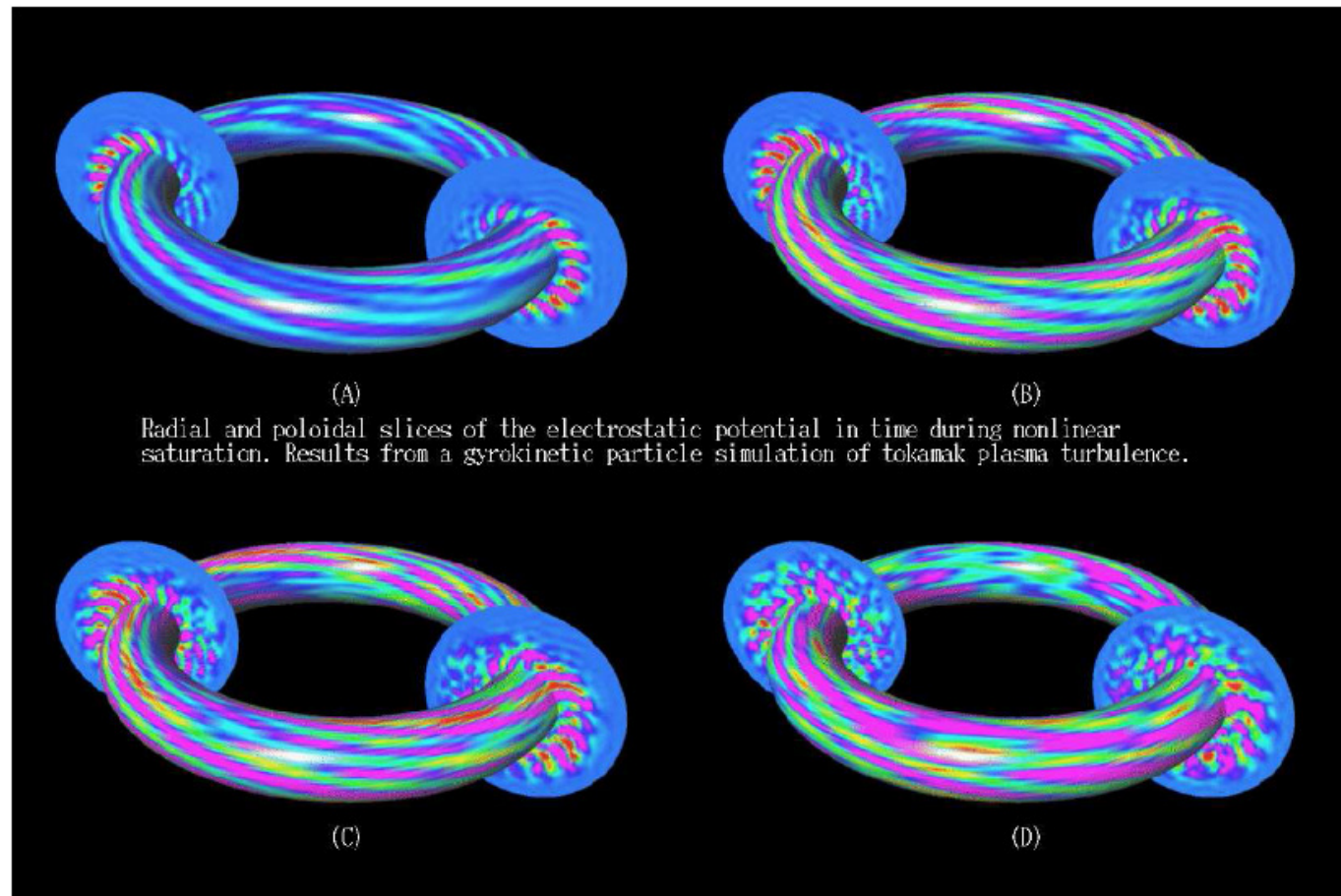
- ❖ **Les modèles de météo et océans**
- ❖ **Les recherches scientifiques**
- ❖ **La biologie moléculaire**
- ❖ **La géophysique (activité sismique)**
- ❖ **Les chaînes d'assemblage automobile**
- ❖ **Les réactions chimiques et nucléaires : 1 ns de simulation demande plus de 300 milliards d'opérations flottantes par seconde (GFLOPS)**
- ❖ **Base de données parallèles de grande taille; big data**
- ❖ **La recherche sur le Web**
- ❖ **La réalité virtuelle ou augmentée**

Avènement du parallélisme

13

Domaines

Physique Nucléaire



Avènement du parallélisme

14

Domaines

Météo



Objectifs du parallélisme

15

Plusieurs objectifs :

- ❖ Economiser du temps : exécution plus rapide
- ❖ Résoudre des problèmes de grande taille
- ❖ Pouvoir traiter plusieurs problèmes en même temps
- ❖ Avoir accès à plusieurs mémoires et plusieurs unité de calcul en cumulant les ressources de plusieurs machines
- ❖ Gagner en terme de puissance et de performance

Quatre décennies de calcul

16

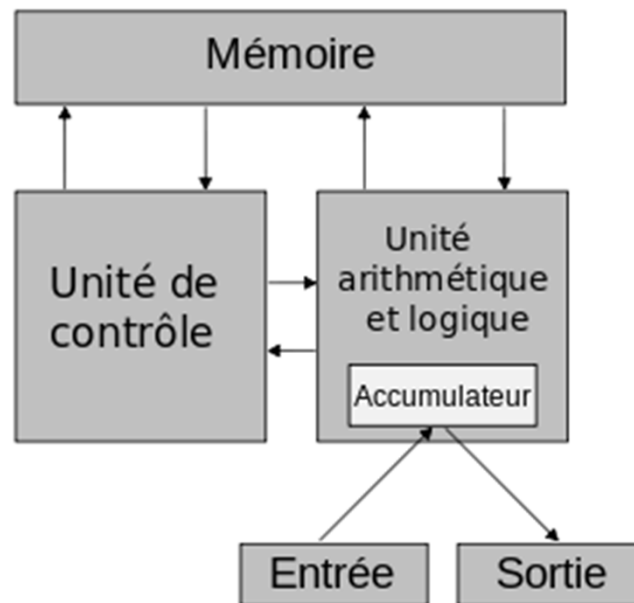
- La plus part des informaticiens admettent qu'il existe 4 domaines de calcul distincts : **batch**, **time-sharing**, **desktop** et **network**.

Caractéristique	Batch	Time-sharing	Desktop	Network
Décennie	1960s	1970s	1980s	1990s/2000s
Emplacement	Computer room (ordinateur)	Terminal room (mini-ordi)	Desktop (micro-ordi)	Mobile
Utilisateurs	Experts	spécialistes	Individus	Groupes
Connectivité	Aucune	Câble périphérique	LAN	Internet/Wireless
Propriétaires	Centre de calcul	Divisional IS shops	Utilisateurs	N'importe qui

Classification

17

Architecture de base de John Von Neumann (Machine séquentielle)



Calcul parallèle

18

- Le domaine de la programmation parallèle est visiblement en continue évolution.
- Depuis 1985, le domaine du parallélisme s'est considérablement développé à cause des demandes toujours croissantes en puissance de calcul.
- Historiquement, il s'est développé dans le cadre de la construction de systèmes d'exploitation.
- Puis, des machines parallèles composées de multiples processeurs sont apparues.

Calcul parallèle

19

- Elles peuvent généralement être classées en deux principales catégories :
 - Machines à mémoire partagée (incluant les machines vectorielles).
 - Machines à mémoire distribuée.
 - Autre type de machines : Machines synchrones (grand nombre d'éléments de calcul (jusqu'à 65536) de faible puissance).
 - Le nombre de processeurs composant une machine peut varier de :
 - Quelques dizaines dans une machine à mémoire partagée.
 - Centaines de milliers dans un système massivement parallèle (MPP).
 - Actuellement : dominance des clusters (grappes).
-

Principales architectures parallèles

20

De nombreuses classifications des machines parallèles existent. La plus répandue est celle de Flynn.

1. Classification de Flynn (1966)

- C'est une classification simpliste, basée sur le nombre (seul ou multiple) de :
 - Flux d'instructions : séquence d'instructions exécutées par la machine.
 - Flux de données : séquence des données appelées par le flux d'instructions.
- Elle divise les machines parallèles en 4 catégories : SISD, SIMD, MISD (n'est pas concrétisée par une machine réelle!) et MIMD.
- Insuffisante pour classer finement les architectures parallèles.

Classification

21

1. Classification de Flynn (1966)

Se base sur deux critères : flux de données et flux d'instructions

		Flux de Données	
		Unique	Multiple
Flux d'Instructions	Unique	SISD	SIMD
	Multiple	MISD	MIMD

	Single Data Stream	Multiple Data Stream
Single Instruction Stream	SISD Machines séquentielles	SIMD machines vectorielles
Multiple Instruction Stream	MISD	MIMD machines multiprocesseur

SPMD (Single Program Multiple Data) : variante de MIMD. Même programme sur des données différentes.

$SIMD < SPMD < MIMD$

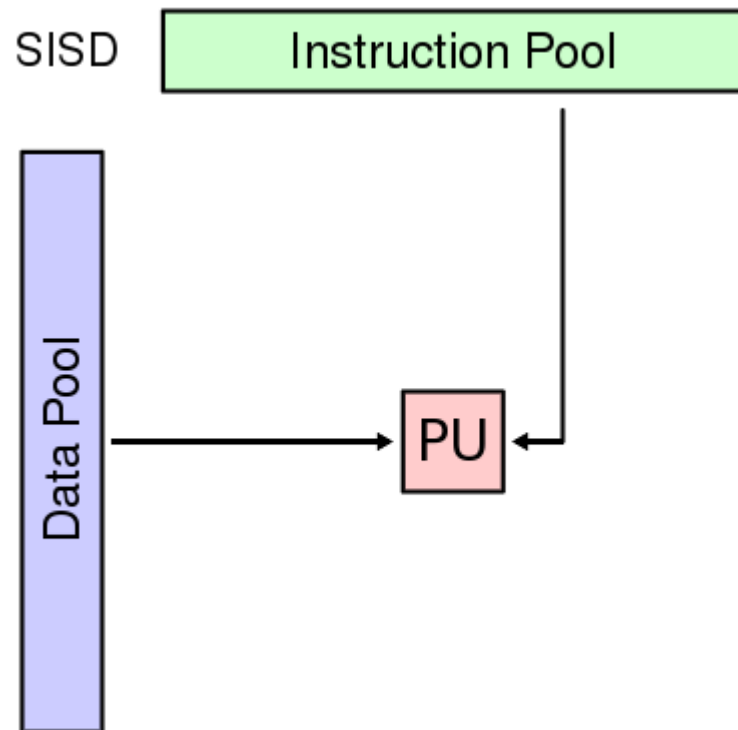
Classification

23

1. Classification de Flynn (1966)

SISD : Single Instruction Single Data

machine séquentielle (J. Neumann)

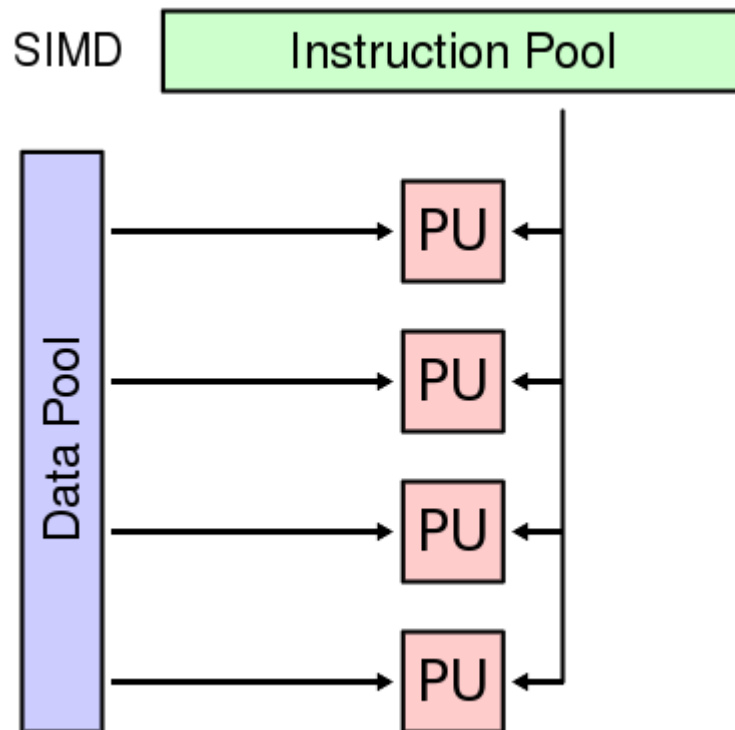


Classification

24

1. Classification de Flynn (1966)

SIMD : Single Instruction Multiple Data



Les unités de calcul exécutent

**la même instruction sur
des données différentes**

Classification

25

1. Classification de Flynn (1966)

SIMD : Single Instruction Multiple Data

- Le modèle SIMD consiste en 2 parties : un frontal et un ensemble de processeurs identiques synchronisés.
- Chaque processeur exécute la même opération que les autres mais sur des données différentes.
- Chaque processeur possède une petite mémoire locale.
- Les processeurs sont liés en utilisant un réseau d'interconnexion.
- Ce modèle convient principalement pour des calculs numériques réguliers (comme le traitement d'images).
- En voie de disparition : il demande des processeurs spécifiques: PAS ceux du marché -> dépassés par l'évolution constante des processeurs du marché et par les machines parallèles MIMD qui les utilisent.

Exemples :

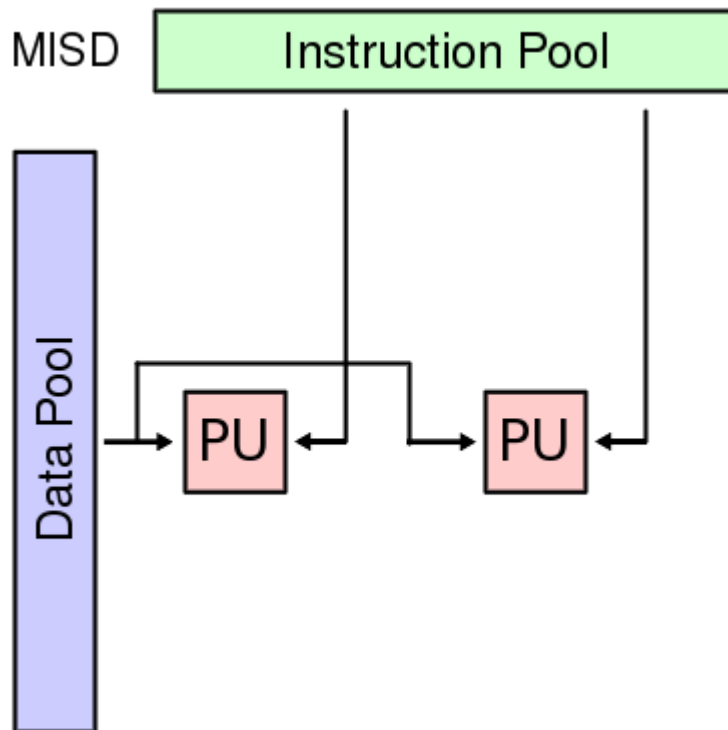
Cray-1 (1976), NEC SX-5, Tera/Cray SV1, Connection Machine CM-1 (1985),...

Classification

26

1. Classification de Flynn (1966)

MISD : Multiple Instruction Single Data



Les unités de calcul exécutent

des instructions différentes sur

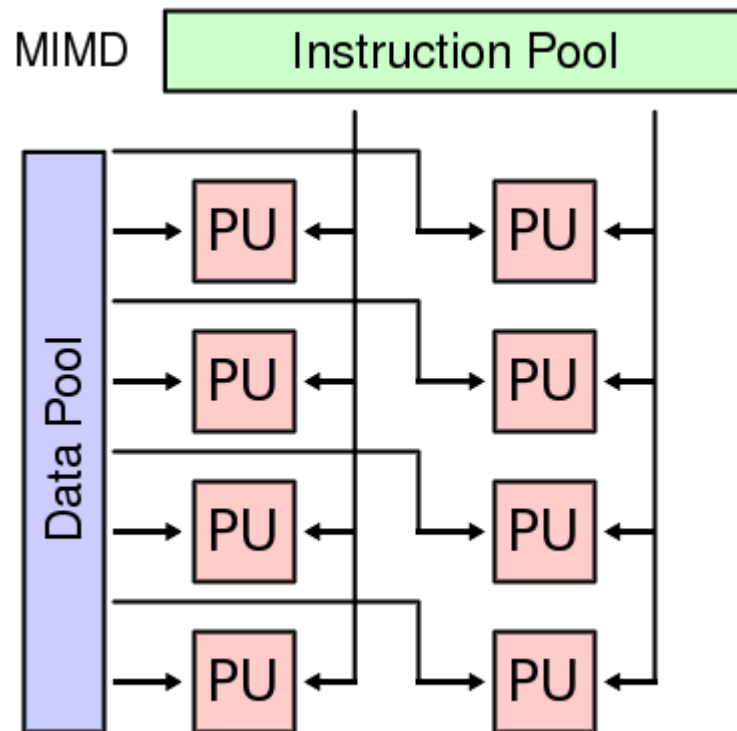
des mêmes données

Classification

27

1. Classification de Flynn (1966)

MIMD : Multiple Instruction Multiple Data



Les unités de calcul exécutent

des instructions différentes sur

des données différentes

Classification

28

1. Classification de Flynn (1966)

		Flux de Données	
		Unique	Multiple
Flux d'Instructions	Unique	SISD	SIMD
	Multiple	MISD	MIMD

Trois classes de machines parallèles

Classification

29

1. Classification de Flynn (1966)

		Flux de Données	
		Unique	Multiple
Flux d'Instructions	Unique	SISD	SIMD
	Multiple	MISD	MIMD



Aucune indication n'est faite sur l'organisation de la mémoire

Classification

30

2. Organisation de la mémoire

❖ SM : Shared Memory

❖ DM : Distributed Memory

Classification

31

2. Organisation de la mémoire

❖ SM : Shared Memory

Tous les processeurs peuvent accéder à l'intégralité de la mémoire

- Espace d'adressage global
- Communication à travers la mémoire partagée
- Pb : conflit d'accès

❖ DM : Distributed Memory

Classification

32

2. Organisation de la mémoire

❖ SM : Shared Memory

Tous les processeurs peuvent accéder à l'intégralité de la mémoire

- Espace d'adressage global
- Communication à travers la mémoire partagée
- Pb : conflit d'accès

❖ DM : Distributed Memory

Chaque processeur dispose de sa propre mémoire

- Espace d'adressage local
- Évite le conflit
- Communication par échange de messages

Classification

33

2. Organisation de la mémoire

❖ **SM : Shared Memory**

❖ **DM : Distributed Memory**



Dans les deux cas, le réseau d'interconnexion est indispensable

Classification

34

2. Organisation de la mémoire

❖ SM : Shared Memory

Le réseau relie l'ensemble de processeurs à la mémoire partagée

❖ DM : Distributed Memory

Classification

35

2. Organisation de la mémoire

❖ SM : Shared Memory

Le réseau relie l'ensemble de processeurs à la mémoire partagée

❖ DM : Distributed Memory

Le réseau relie l'ensemble de processeurs entre eux

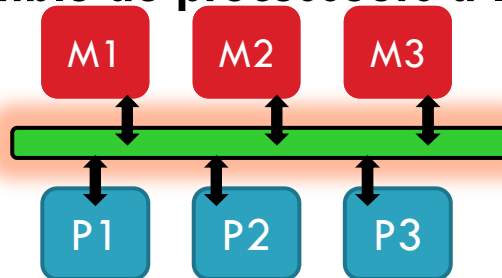
Classification

36

2. Organisation de la mémoire

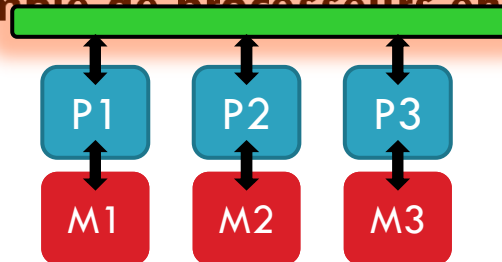
❖ SM : Shared Memory

Le réseau relie l'ensemble de processeurs à la mémoire partagée



❖ DM : Distributed Memory

Le réseau relie l'ensemble de processeurs entre eux



Classification

37

3. Classification unifiée

Cette classification combine les deux précédentes

- ❖ SIMD-SM
- ❖ SIMD-DM
- ❖ MIMD-SM
- ❖ MIMD-DM
- ❖ MISD-SM
- ❖ MISD-DM

Classification

38

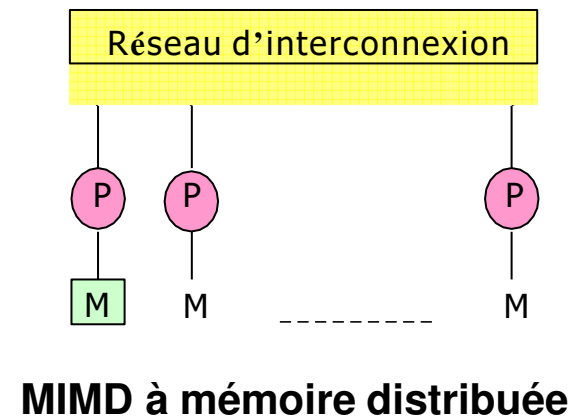
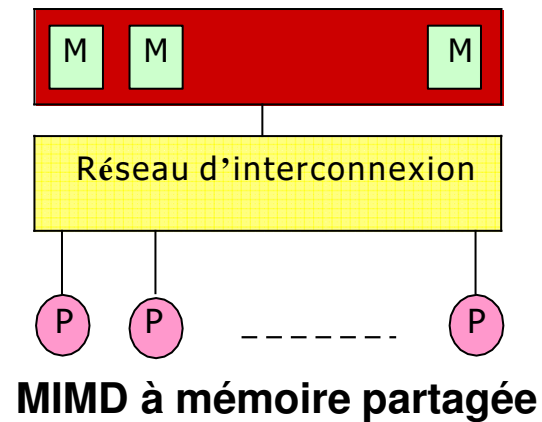
Architecture MIMD

- Classe de machines la plus répandue.
- Chaque processeur exécute son propre code sur ces propres données de manière asynchrone et indépendante.
- On distingue habituellement 2 sous-classes :
 - **MIMD à mémoire partagée** : les processeurs de la machine ont accès à une mémoire commune.
 - **MIMD à mémoire distribuée** : les processeurs disposent chacun d'une mémoire locale. Dans ce cas, un réseau d'interconnexion est nécessaire pour échanger les informations entre processeurs.

Classification

39

Architecture MIMD



Classification

40

Système à mémoire partagée

- ❑ La coordination inter-processeurs est accomplie à travers une mémoire globale partagée par tous les processeurs.
- ❑ Ces systèmes sont aussi appelés SMP (Symmetric Multiprocessors). En général entre 2 et 64 processeurs.
- ❑ Leur programmation est relativement facile puisqu'il n'est plus nécessaire de mettre en œuvre des communications explicites.
- ❑ Extensibilité? En raison des coûts très élevés et de la complexité de réalisation (il est très difficile de gérer une mémoire partagée à laquelle pourraient accéder plusieurs dizaines de processeurs), **les concepteurs de machines parallèles ont opté pour le « distribué »**.
- ❑ **Exemples** : PC bi-processeurs, Cray SV1, SGI Origin3000, un nœud de l'IBM SP3, ...

Classification

41

Organisation de la mémoire : ces systèmes peuvent être classés en :

- UMA (Uniform Memory Access) :
 - Tous les processeurs ont le même temps d'accès à n'importe quelle case mémoire.
 - Aussi nommé SMP.

 - NUMA (Non UMA) : le temps d'accès à un module mémoire dépend de la distance le séparant au processeur
 - Appelé aussi cc-NUMA.
 - Exp : SGI ORIGIN3000.
 - Souvent construit comme reliant plusieurs SMP.
 - Chacun des SMP peut accéder à la mémoire des autres SMP.
 - Temps d'accès à la mémoire n'est pas égal pour tous.
 - Extensible.
-

Classification

42

Système à mémoire distribuée

- La notion de mémoire globale n'existe pas.
- **Les machines à mémoire distribuée sont les seules susceptibles de fournir de très hautes performances : Extensibilité.**
- **Exemples de systèmes dans les années 90** : Systèmes à base de Transputers, Tera/Cray T3E, IBM SP2, IBM SP2 , ...

MIMD CLUMPS : Clusters of SMP

- Construits à partir de SMP.
- Mémoire partagée au sein de chacun des SMP.
- Mémoire distribuée entre SMP.
- **Exemple** : IBM SP2 : des nœuds SMP de 4 processeurs chacun reliés par un réseau dédié.

Classification

43

Système à mémoire distribuée

MIMD CLUMPS : Clusters of SMP

- Construits à partir de SMP.
- Mémoire partagée au sein de chacun des SMP.
- Mémoire distribuée entre SMP.
- **Exemple** : IBM SP2 : des nœuds SMP de 4 processeurs chacun reliés par un réseau dédié.



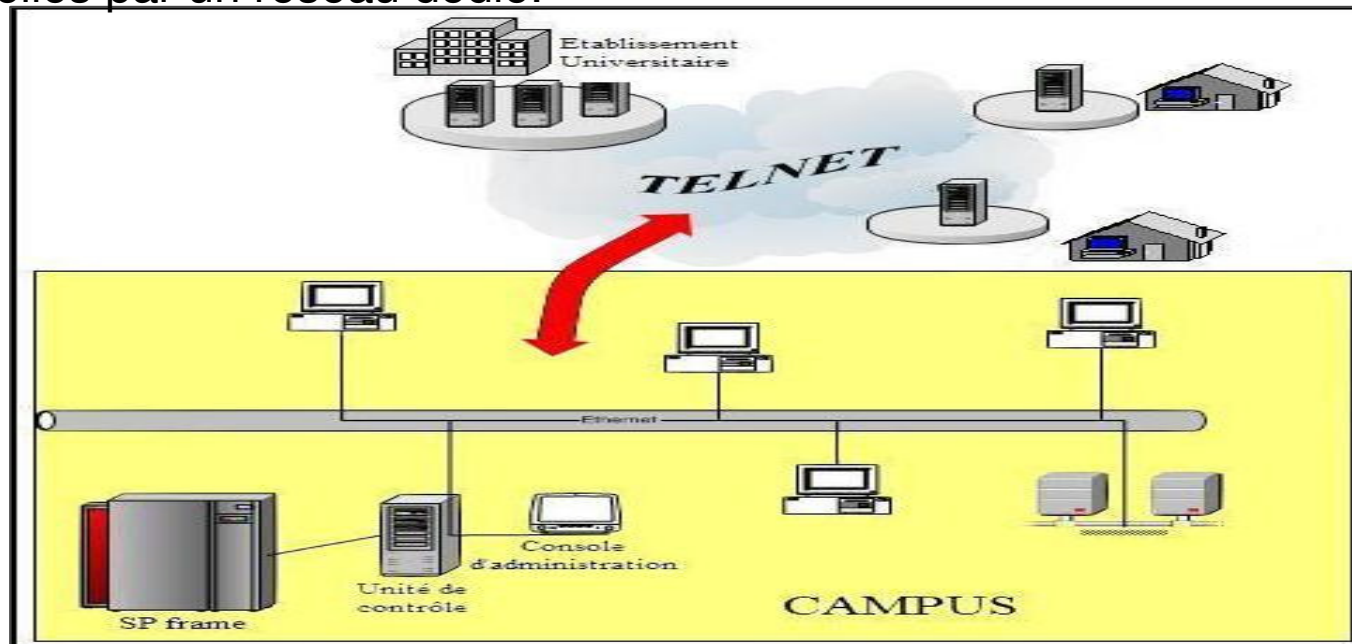
Classification

44

Système à mémoire distribuée

MIMD CLUMPS : Clusters of SMP

- Construits à partir de SMP.
- Mémoire partagée au sein de chacun des SMP.
- Mémoire distribuée entre SMP.
- **Exemple** : IBM SP2 : des nœuds SMP de 4 processeurs chacun reliés par un réseau dédié.



Classification

45

Système à mémoire distribuée

Remarque :

Malheureusement, pour exécuter un programme **p fois plus rapide**, il ne suffit pas de faire exécuter ce programme **sur p processeurs**.

En effet, les processeurs devront échanger des données. Or, **les coûts de communication** sont très importants. Les échanges de données s'effectueront ainsi au détriment de la performance globale du calcul.

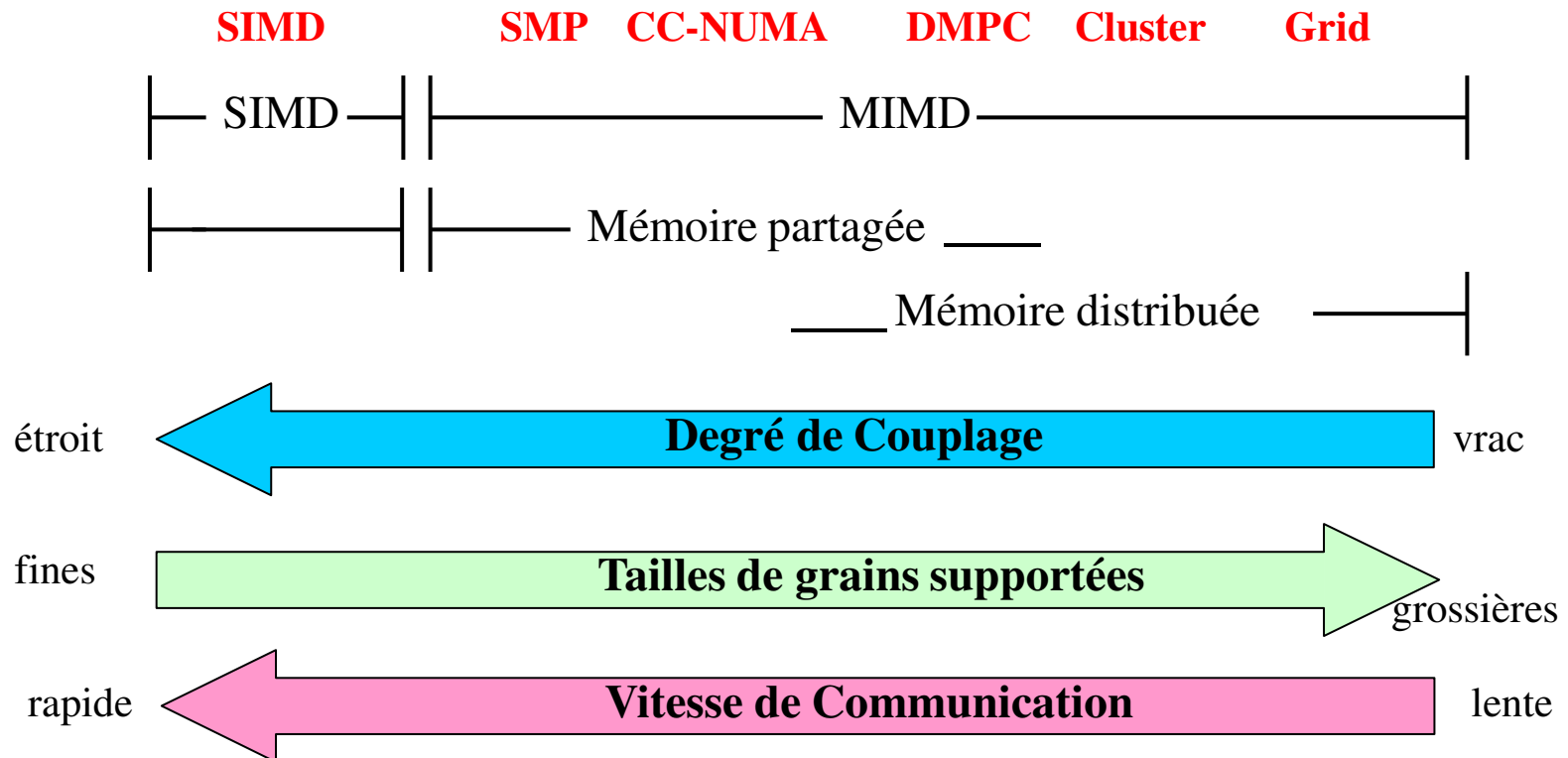


Optimiser les temps des mouvements de données au sein des machines parallèles à mémoire distribuée est ainsi une clé pour l'obtention de performances significatives sur ces machines.

Classification

46

Architectures parallèles et distribuées (Leopold, 2001)



Classification

47

Architecture hybride

- ❑ La situation réelle est plus compliquée. On peut avoir des architectures qui présentent des aspects couvrant plus d'une catégorie. En effet, plusieurs machines sont de conception hybride.
- ❑ Combiner les avantages des architectures à mémoire partagée et à mémoire distribuée.
- ❑ Donner naissance à une architecture à ***mémoire distribuée-partagée***.
- ❑ La mémoire est physiquement distribuée : l'architecture matérielle peut suivre le modèle de passage de message, et la façon de programmation suit le modèle à mémoire partagée.

Exemple de système : SGI Origin2000.

Classification

48

Systèmes parallèles et systèmes distribués

- Systèmes parallèles :
 - ✓ Ensemble d'éléments de calcul.
 - ✓ Qui peuvent communiquer et coopérer.
 - ✓ Pour résoudre rapidement de grands problèmes.
- Systèmes distribués :
 - ✓ Ensemble de processeurs autonomes.
 - ✓ Qui ne se partagent pas de mémoire primaire.
 - ✓ Mais qui coopèrent par envoi de messages à travers un réseau de communication.

Classification

49

Systèmes parallèles / systèmes distribués

- **Pas** parallèles, **pas** distribués
 - PCs ou stations de travail (qui ne peuvent pas communiquer)
- Parallèles, **pas** distribués
 - Machines vectorielles
 - Machines multiprocesseurs à mémoire partagée (SMP)
- Distribués, **pas** parallèles
 - Réseau de stations large distance (communications trop lentes)
- Parallèles et distribués
 - Réseau de stations de travail connectées à un réseau local/spécialisé CLUMPS

Classification

50

Une autre taxinomie

La difficulté de transposer la classification de Flynn dans des architectures physiques réelles justifie l'investigation d'une deuxième taxinomie. Cette deuxième approche va donner une vue complémentaire à celle de Flynn et classifie les architectures selon les interconnexions des processeurs et des mémoires. Cinq classes sont souvent identifiées.

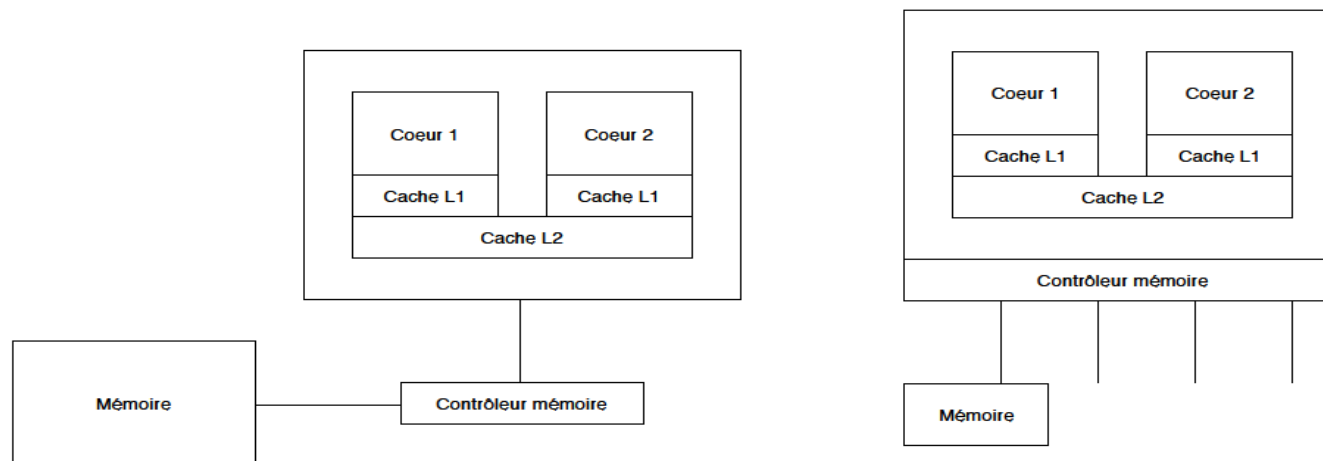
Principaux systèmes matériels actuels

- **Processeurs multi-cœurs**
- **Processeurs vectoriels**
- **Processeurs graphiques**
- **Grappes (Clusters)**
- **Grilles (Grid)**
- **Clouds**
- **Ordinateurs quantiques, neuronaux, réseaux de neurones**

Principaux systèmes matériels actuels

Processeurs multi-cœurs

Un processeur multi-cœurs est composé d'au moins deux unités de calcul (cœurs) gravées au sein d'une même puce. Les cœurs des processeurs dans la plupart des cas sont homogènes (identiques). Mais IBM, Sony et Toshiba ont exploité le cas de cœurs hétérogènes (différents) et spécialisés dans des domaines bien précis (audio, affichage, calcul pur). Depuis 2005, les processeurs multi-cœurs sont sur le marché.



Deux approches différentes d'architecture multi-cœurs

Processeurs vectoriels

Les processeurs vectoriels sont des processeurs pouvant traiter un vecteur entier (tableau de dimension 1) de données en une instruction (dite vectorielle e.g. addition vectorielle). Ces processeurs ont été la clé de performance pour les premiers super-ordinateurs du type Cray dans les années 80 et même pour des plates-formes assez récentes comme Earth-simulator qui a marqué ce début de siècle en détenant la tête de la liste du Top500 pendant plus de deux ans successifs.

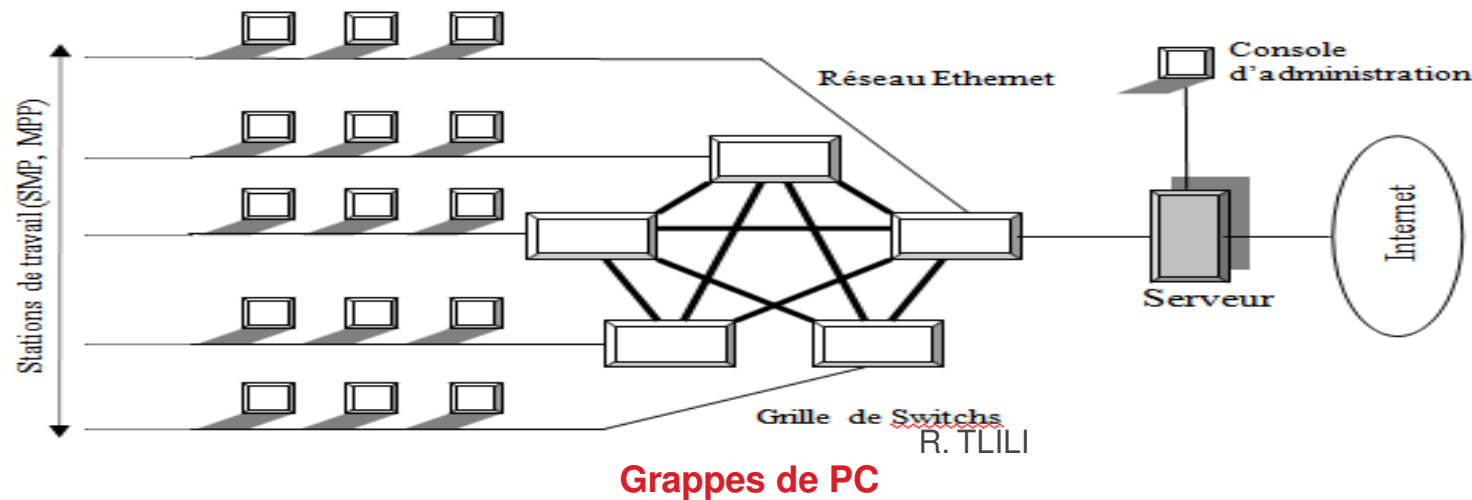
Processeurs graphiques

Depuis quelques années, les processeurs graphiques ont été envisagés comme accélérateurs de traitements parallèles. Un processeur graphique communément connu sous le nom GPU (Graphical Processing Unit) est un processeur conçu pour décharger les processeurs (CPU : Central Processing Unit) des traitements graphiques. Cependant, vu que les opérations traitées par ces processeurs ne sont qu'une forme d'arithmétique, les GPUs se sont progressivement développés vers un modèle utilisable aussi pour les calculs non graphiques.

Principaux systèmes matériels actuels

Grappes (Clusters)

Une grappe (cluster) n'est pas une machine parallèle unique au sens classique mais est constitué d'un ensemble de nœuds (machines monoprocesseurs, processeurs multi-cœurs ou SMP) interconnectés par un réseau local et (souvent) rapide. C'est donc un réseau d'ordinateurs en général de caractère homogène. Un des objectifs principaux d'un grappe est de fournir un environnement à image d'un système unique pour des applications destinées au cluster.



Grilles (Grid)

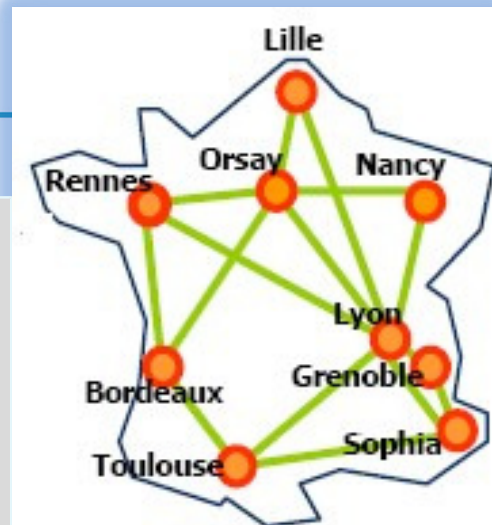
- Grid signifie littéralement 'grille'. Une grille peut être vue comme un environnement informatique et de collaboration sans frontière. Les grilles de calcul permettent d'utiliser les cycles CPU durant lesquels les machines sont inactives pour exécuter une application. Les capacités de stockage des différents utilisateurs sont les ressources agrégées au sein de la grille.



constituée de 5000 CPU
réparties sur 9 sites

Grappe de Lille
Ressources : 500 cœurs
de calcul

R. TLILI



Principaux systèmes matériels actuels

Grilles (Grid)

- Une caractéristique importante des grilles de calcul est de fournir une importante capacité pour le calcul parallèle.
- Les domaines académique et industriel utilisent énormément de telles capacités. Les ressources partagées rendent possible l'accès à des ressources spéciales et des logiciels dont le prix de licence est élevé.
- Il s'agit d'une mise en commun des ressources logicielles et matérielles. En raison de la disponibilité d'importantes quantités de ressources, si certaines ressources deviennent inaccessibles localement, la continuité du service reste assurée.

Clouds

- Le terme Cloud fait référence habituellement à un modèle basé sur Internet et dans lequel les données ne sont pas maintenues par l'utilisateur.
- Plusieurs définitions ont été attribuées à ce terme selon son utilisation:
 - ✓ **Gartner** : c'est un style de calcul offrant à plusieurs clients à travers Internet des capacités massives et scalables relatives aux technologies d'information (la scalabilité fait référence à la capacité d'un système à accroître sa capacité de calcul sous une charge accrue quand des ressources généralement matérielles sont ajoutées).

Clouds

Plusieurs définitions ont été attribuées à ce terme selon son utilisation:

✓ **IBM** : c'est un paradigme de calcul émergeant, dans lequel les données et services résident dans des centres de calcul massivement scalables qui peuvent être accédés à partir de n'importe quel périphérique connecté à Internet.

- Le premier cloud qui a été rendu accessible est l'**EC2 (Elastic Compute Cloud)** lancé en 2006.

Ordinateurs quantiques, neuronaux, réseaux de neurones

L'origine des ordinateurs quantiques

Le grand physicien Richard Feynman a été un des premiers à comprendre au début des années 1980 que l'on pouvait mettre à profit les lois de la mécanique quantique pour simuler et mieux comprendre des systèmes quantiques à l'aide d'autres systèmes quantiques. Il se trouve en effet, notamment dans le domaine de la chimie quantique et de la physique du solide que l'on soit **assez rapidement limité par le volume de calculs nécessaires pour les simuler à l'aide d'ordinateurs classiques.**

Mais comme l'explique Feynman dans son célèbre ouvrage « Leçons sur l'informatique » , il est possible au minimum de faire des calculateurs ou des simulateurs quantiques qui permettent de contourner l'obstacle. On doit pour cela utiliser des généralisations des bits d'information classique que l'on appelle des qubits et construire des portes logiques quantiques qui opèrent sur ces qubits. **La superposition quantique et le phénomène d'intrication quantique permettent alors, en quelque sorte, d'effectuer un grand nombre de calculs en parallèle.**

Ordinateurs quantiques, neuronaux, réseaux de neurones

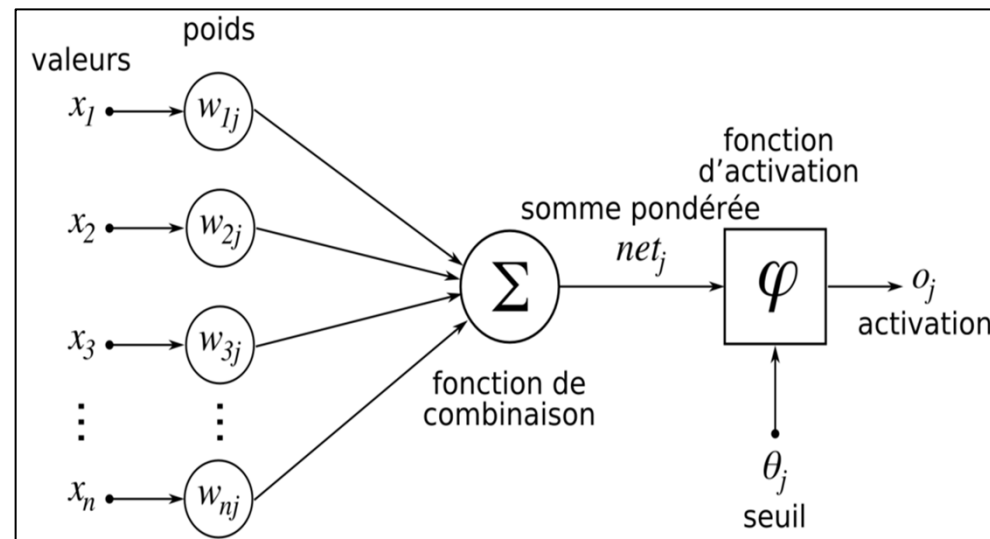
Un réseau de neurones est en général composé d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. Chaque couche (i) est composée de N_i neurones, prenant leurs entrées sur les N_{i-1} neurones de la couche précédente. À chaque synapse est associé un poids synaptique, de sorte que les N_{i-1} sont multipliés par ce poids, puis additionnés par les neurones de niveau i, ce qui est équivalent à multiplier le vecteur d'entrée par une matrice de transformation. Mettre l'une derrière l'autre les différentes couches d'un réseau de neurones reviendrait à mettre en cascade plusieurs matrices de transformation et pourrait se ramener à une seule matrice, produit des autres, s'il n'y avait à chaque couche, la fonction de sortie qui introduit une non linéarité à chaque étape. Ceci montre l'importance du choix judicieux d'une bonne fonction de sortie : **un**

réseau de neurones dont les sorties seraient linéaires n'aurait aucun intérêt.

Au-delà de cette structure simple, le réseau de neurones peut également contenir des boucles qui en changent radicalement les possibilités mais aussi la complexité. De la même façon que des boucles peuvent transformer une logique combinatoire en logique séquentielle, **les boucles dans un réseau de neurones transforment un simple dispositif de reconnaissance d'entrées en une machine complexe capable de toutes sortes de comportements.**

Ordinateurs quantiques, neuronaux, réseaux de neurones

Le neurone propage son nouvel état interne sur son axone. Dans un modèle simple, la fonction neuronale est simplement une fonction de seuillage : elle vaut 1 si la somme pondérée dépasse un certain seuil ; 0 sinon. Dans un modèle plus riche, le neurone fonctionne avec des nombres réels (souvent compris dans l'intervalle $[0,1]$ ou $[-1,1]$). **On dit que le réseau de neurones passe d'un état à un autre lorsque tous ses neurones recalculent en parallèle leur état interne, en fonction de leurs entrées.**



Réseaux d'interconnexion

62

- Les réseaux d'interconnexion des multiprocesseurs peuvent être classés en se basant sur un nombre de critères, comme:
 - Mode d'opération (synchrone vs asynchrone).
 - Stratégie de contrôle (centralisée vs décentralisée).
 - Techniques de switching (circuit vs paquet).
 - Topologie (statique vs dynamique).

Réseaux d'interconnexion

63

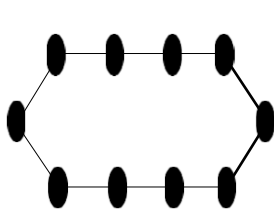
- **Topologie**

- Généralement, un réseau de processeurs est représenté par un graphe, dont les arêtes sont les liens de communication et les sommets (nœuds), les processeurs.
- La **plus part** des architectures se fondent sur des **topologies régulières et statiques**.
- Un **réseau dynamique** établit une connexion entre deux nœuds ou plus à la volée lorsque des messages arrivent sur les liens.

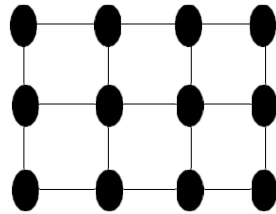
Réseaux d'interconnexion

64

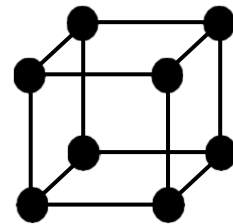
- Les **topologies statiques** les plus populaires sont les suivantes :



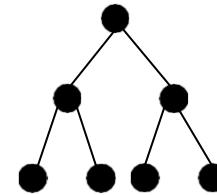
Anneau



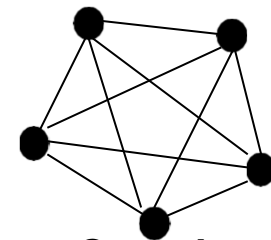
Grille 2D



Hypercube



Arbre



Complet

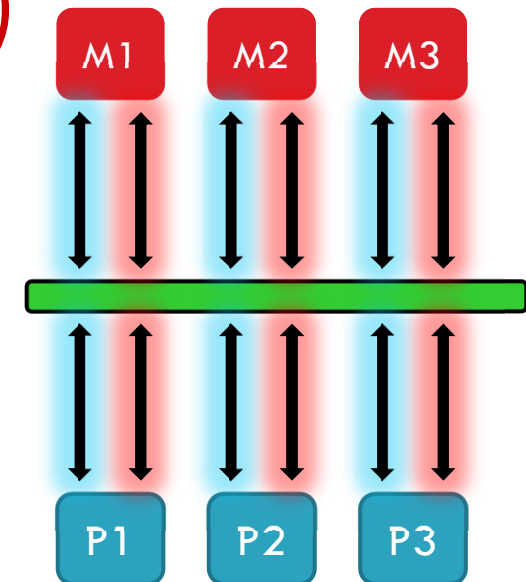
- Topologies dynamiques** : crossbar, multi-étages,...
-

Réseaux d'interconnexion

65

1. Cas de SM (Shared Memory)

Le réseau dans ce cas permet de relier
les **bus des données** et les **bus d'adresses**
à n'importe quel module de mémoire



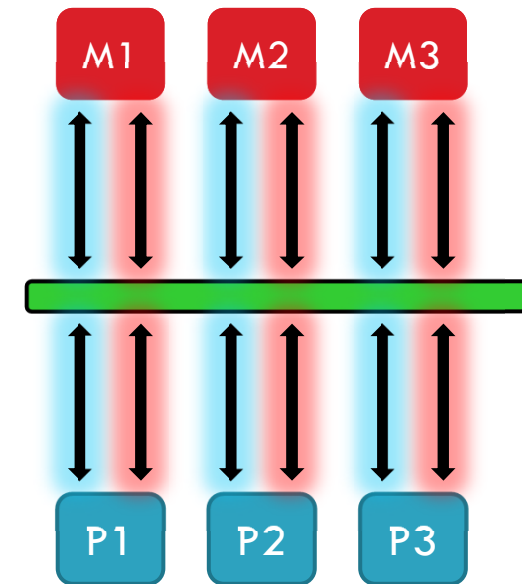
Réseaux d'interconnexion

66

1. Cas de SM

Le réseau dans ce cas permet de relier
les **bus des données** et les **bus d'adresses**
à n'importe quel module de mémoire

- ❖ Les largeur des bus des données sont les mêmes
- ❖ La largeur des bus d'adresses est déterminée
par la taille de module mémoire



L'adresse = (adresse de module (banc) mémoire, adresse de la case mémoire dans ce banc)

(b, a)

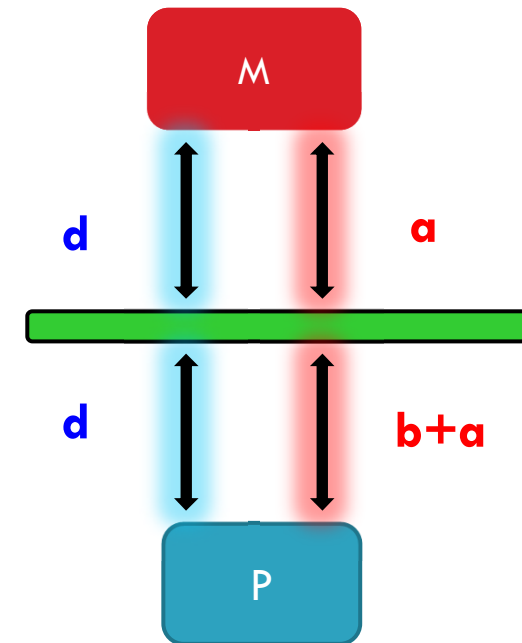
Réseaux d'interconnexion

67

1. Cas de SM

Le réseau dans ce cas permet de relier
les **bus des données** et les **bus d'adresses**
à n'importe quel module de mémoire

- ❖ Les largeur des bus des données sont les mêmes
- ❖ La largeur des bus d'adresses est déterminée
par la taille de module mémoire



Réseaux d'interconnexion

68

Types des réseaux d'interconnexion dans une machine // **SM**

- ❖ **Type1: Zéro étage**
- ❖ **Type2: Un étage ou crossbar**
- ❖ **Type3: Multi-étages**

Réseaux d'interconnexion

69

Types des réseaux d'interconnexion dans une machine // SM

❖ Zéro étage

- Réseau connecté à bus
- La mémoire dans ce cas est constituée d'un seul banc mémoire
- Pb : conflit sur la mémoire
- Sol : accès en exclusion mutuelle au banc mémoire

Réseaux d'interconnexion

70

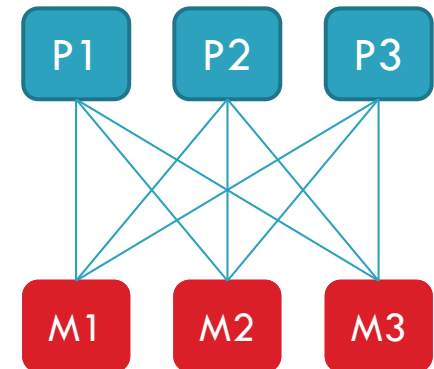
Types des réseaux d'interconnexion dans une machine // SM

❖ Un étage ou crossbar

- La mémoire est composée de plusieurs bancs mémoire
- Tous les processeurs sont connectés à tous les bancs

➤ Graphe complet

- Les processeurs peuvent accéder en // à différents bancs
- Exclusion mutuelle sur le même banc mémoire



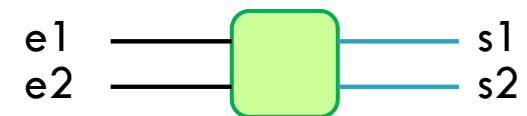
Réseaux d'interconnexion

71

Types des réseaux d'interconnexion dans une machine // SM

❖ Multi-étages

- Ces réseaux se basent sur des circuits dits **boîtes d'échange** (exchange box)
- Les boîtes d'échange permettent d'établir des liens entre deux entrées et deux sorties :



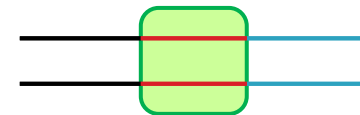
Réseaux d'interconnexion

72

Types des réseaux d'interconnexion dans une machine // SM

❖ Multi-étages

- Ces réseaux se basent sur des circuits dits **boîtes d'échange** (exchange box)
- Les boîtes d'échange permettent d'établir des liens entre deux entrées et deux sorties :
- Faire transiter les entrées tout droit (straight)



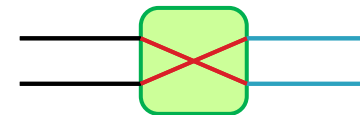
Réseaux d'interconnexion

73

Types des réseaux d'interconnexion dans une machine // SM

❖ Multi-étages

- Ces réseaux se basent sur des circuits dits **boîtes d'échange** (exchange box)
- Les boîtes d'échange permettent d'établir des liens entre deux entrées et deux sorties :
- les croiser (exchange)



Réseaux d'interconnexion

74

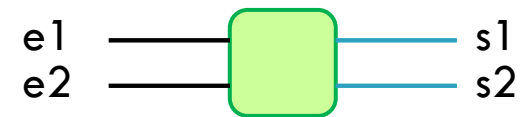
Types des réseaux d'interconnexion dans une machine // SM

❖ Multi-étages

- Ces réseaux se basent sur des circuits dits **boîtes d'échange** (exchange box)
- Les boîtes d'échange permettent d'établir des liens entre deux entrées et deux sorties

Si deux processeurs transitent par la même boîte et demandent straight ou exchange

- Pas de conflit (accès simultané)
- Sinon : nécessité d'arbitrage



Réseaux d'interconnexion

75

3. Types des réseaux d'interconnexion dans une machine // SM

❖ Multi-étages

■ Schéma général



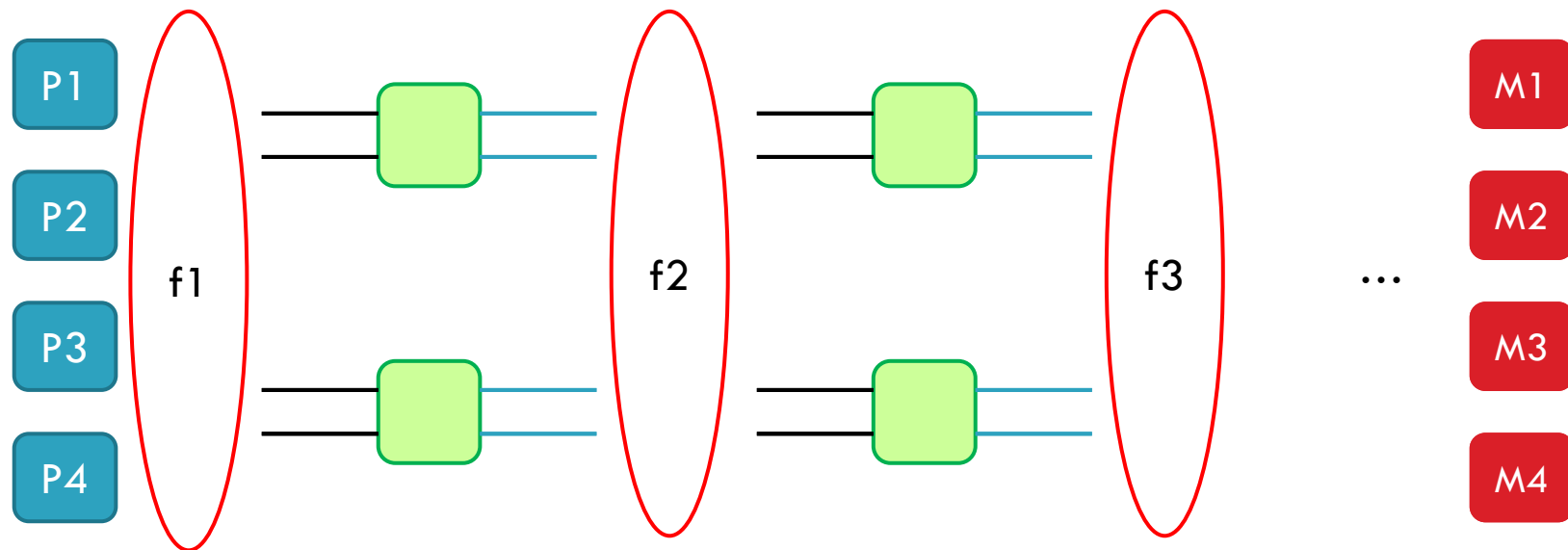
Réseaux d'interconnexion

76

Types des réseaux d'interconnexion dans une machine // SM

❖ Multi-étages

■ Schéma général



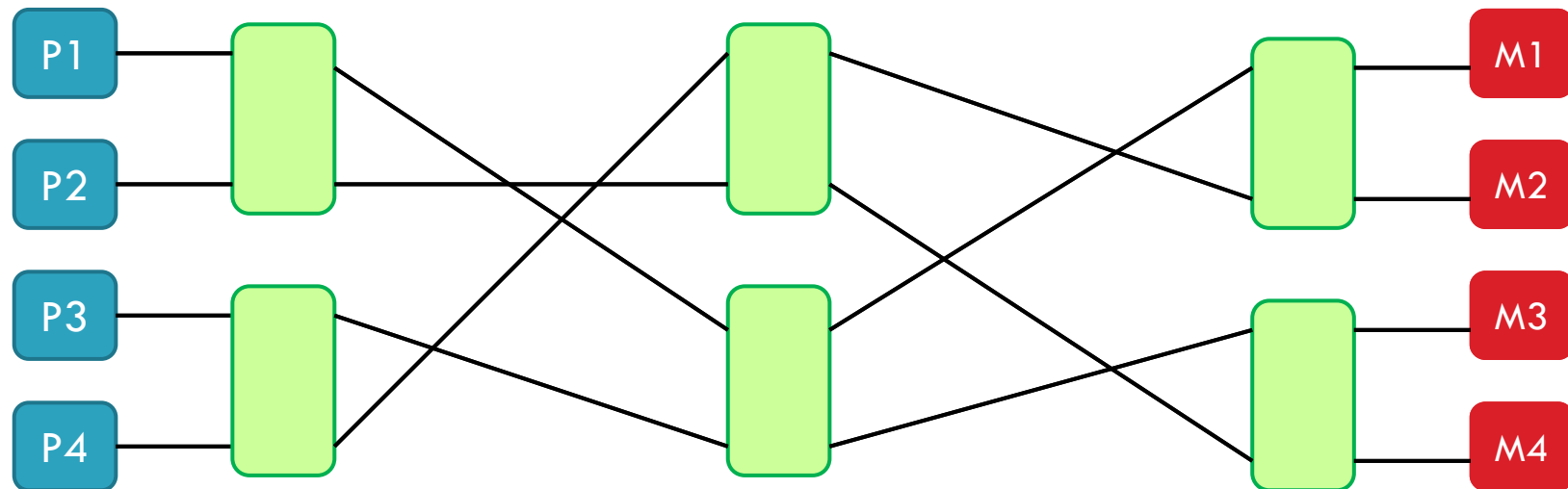
Réseaux d'interconnexion

77

Types des réseaux d'interconnexion dans une machine // SM

❖ Multi-étages

■ Exemple



Réseaux d'interconnexion

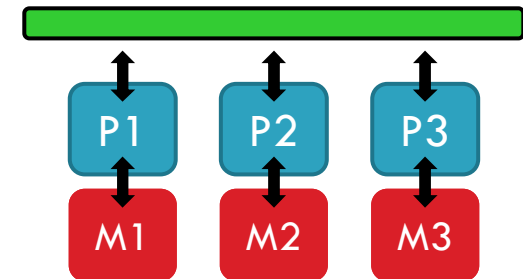
78

2. Cas de DM (Distributed Memory)

**Le réseau dans ce cas achemine les messages
entre les différents processeurs
(machine à passage de messages)**

Les message = (entête, corps)

Entête = (destinateur, expéditeur, informations)



Réseaux d'interconnexion

79

Types des réseaux d'interconnexion dans une machine // DM

❖ Rappel sur la théorie des graphes

$G(X, U)$ un graphe

X : ensemble des nœuds (sommets)

U : ensemble des arcs (arrêtes/liens)

Réseaux d'interconnexion

80

Types des réseaux d'interconnexion dans une machine // DM

❖ Rappel sur la théorie des graphes

$G(X, U)$ un graphe

X : ensemble des nœuds (sommets)

U : ensemble des arcs (arrêtes/liens)

$|X| = n$ ordre de graphe G

$|U| = m$ nombre d'arcs dans G

Degré d'un nœud d° = nombre d'arcs qui passent par ce nœud

Distance entre deux nœuds = longueur de plus court chemin

Diamètre d'un graphe D = distance maximale dans le graphe

Connectivité K = nombre de chemins différents possibles entre deux nœuds

Réseaux d'interconnexion

81

Types des réseaux d'interconnexion dans une machine // DM

❖ Grille à une dimension



Réseaux d'interconnexion

82

4. Types des réseaux d'interconnexion dans une machine // DM

❖ Grille à une dimension



$$D(G) = p - 1$$

$$d^{\circ}(\text{nœud}) = 1 \text{ extrémité}$$

$$d^{\circ}(\text{nœud}) = 2 \text{ intérieur}$$

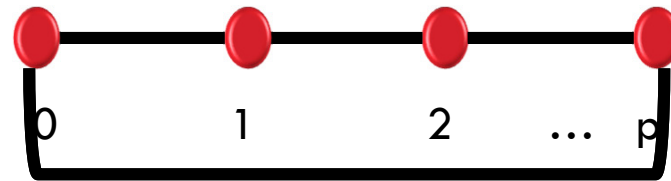
$$K(G) = 1 \quad \text{pas de parallélisme possible}$$

Réseaux d'interconnexion

83

Types des réseaux d'interconnexion dans une machine // DM

❖ Réseau torique = anneau

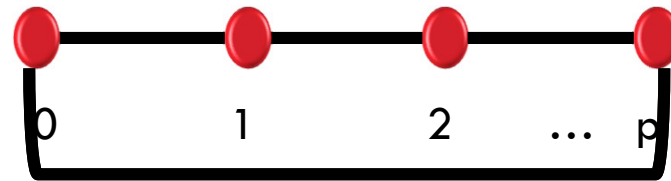


Réseaux d'interconnexion

84

Types des réseaux d'interconnexion dans une machine // DM

❖ Réseau torique = anneau



$$D(G) = E(p / 2)$$

$$d^{\circ}(\text{nœud}) = 2$$

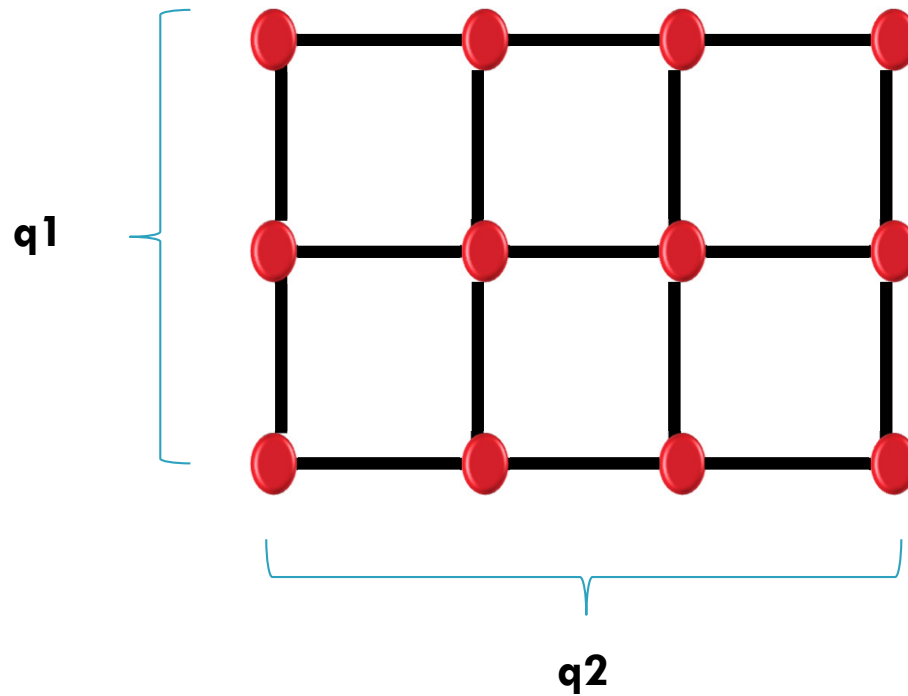
$$K(G) = 2 \quad \text{possibilité de parallélisme}$$

Réseaux d'interconnexion

85

Types des réseaux d'interconnexion dans une machine // DM

❖ Grille à deux dimensions

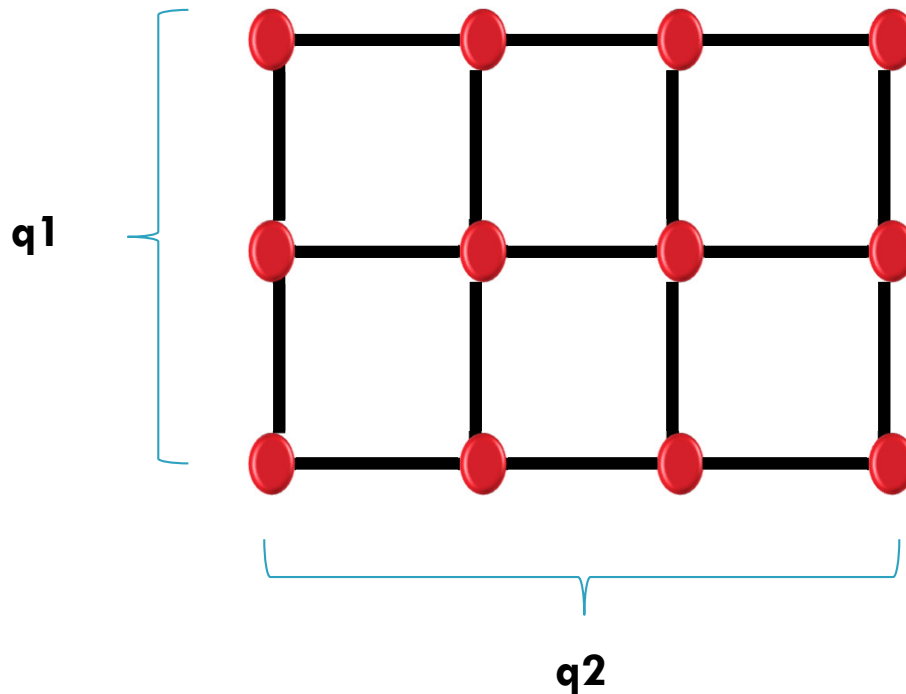


Réseaux d'interconnexion

86

4. Types des réseaux d'interconnexion dans une machine // DM

❖ Grille à deux dimensions



$$D(G) = q1 + q2 - 2$$

$$\begin{aligned} d^{\circ}(\text{nœud}) &= 2 \text{ extrémité} \\ &= 3 \text{ bord} \\ &= 4 \text{ intérieur} \end{aligned}$$

$$K(G) \geq 2$$

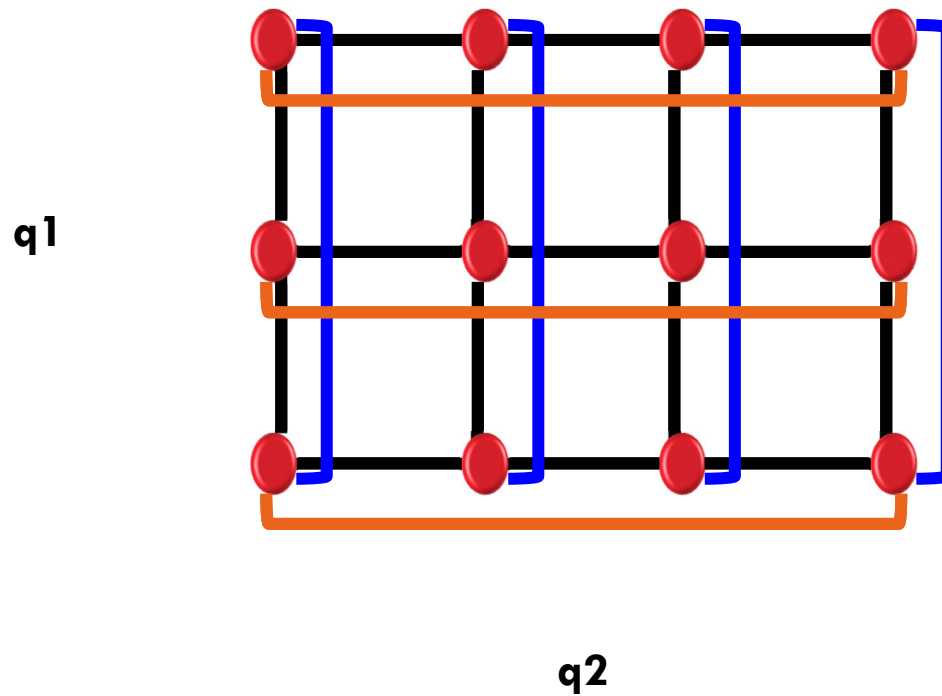
parallélisme possible

Réseaux d'interconnexion

87

Types des réseaux d'interconnexion dans une machine // DM

❖ Grille à deux dimensions torique

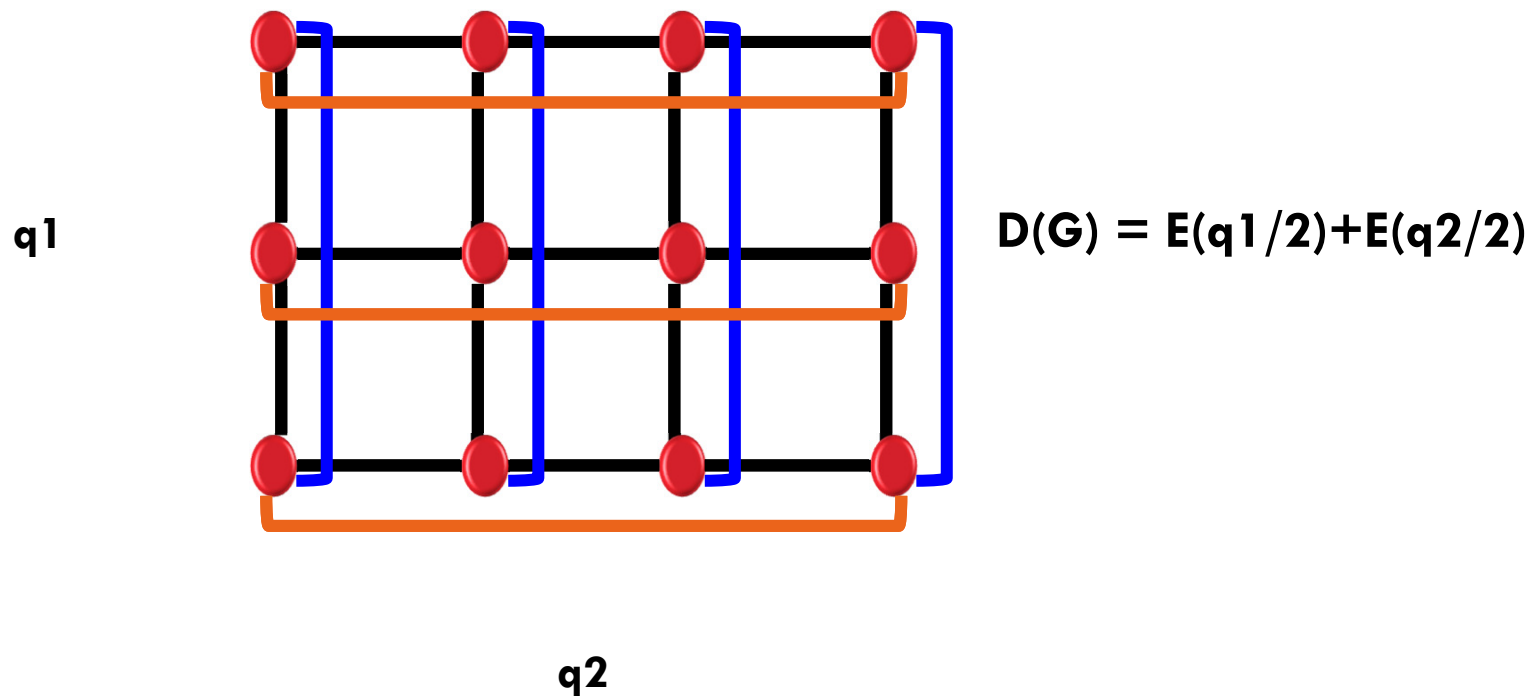


Réseaux d'interconnexion

88

Types des réseaux d'interconnexion dans une machine // DM

❖ Grille à deux dimensions torique

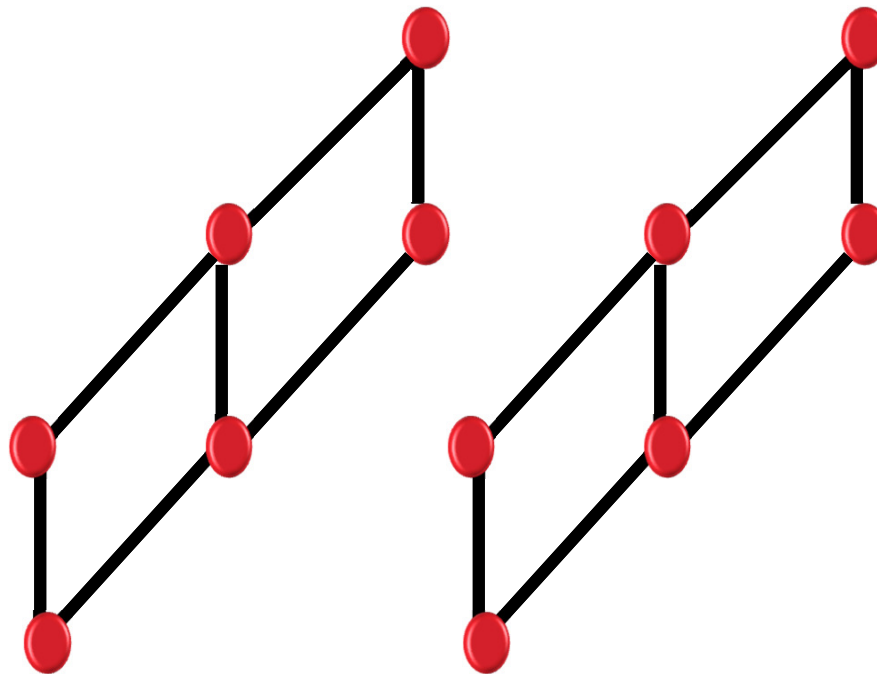


Réseaux d'interconnexion

89

Types des réseaux d'interconnexion dans une machine // DM

❖ Grille à trois dimensions (Maille)

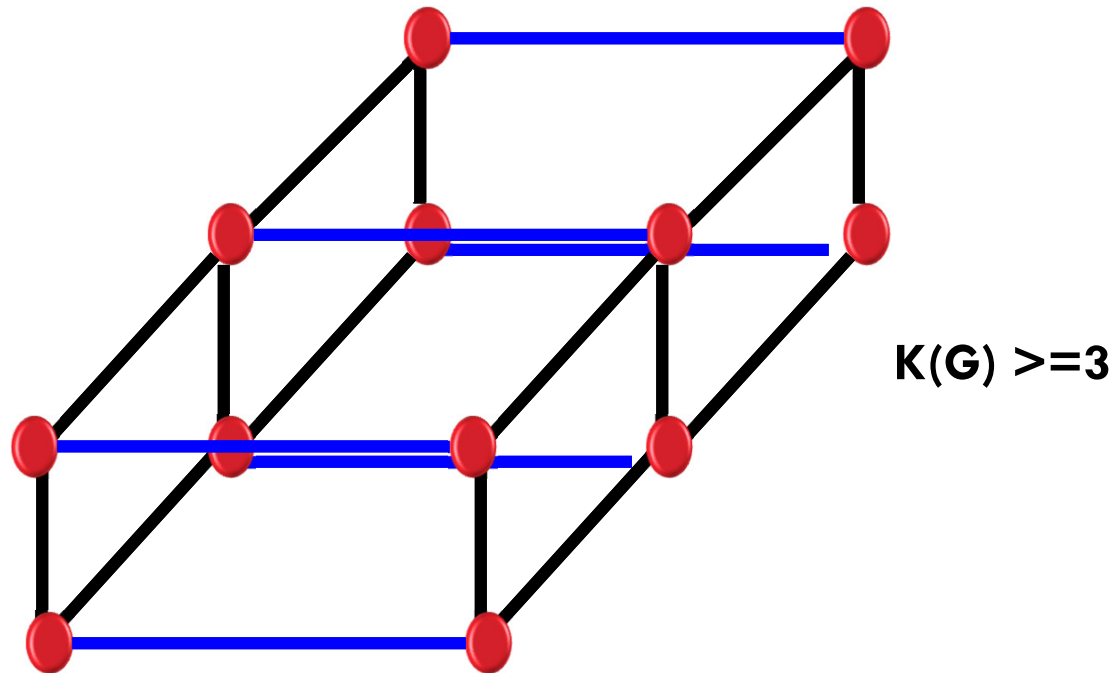


Réseaux d'interconnexion

90

4. Types des réseaux d'interconnexion dans une machine // DM

❖ Grille à trois dimensions (Maille)

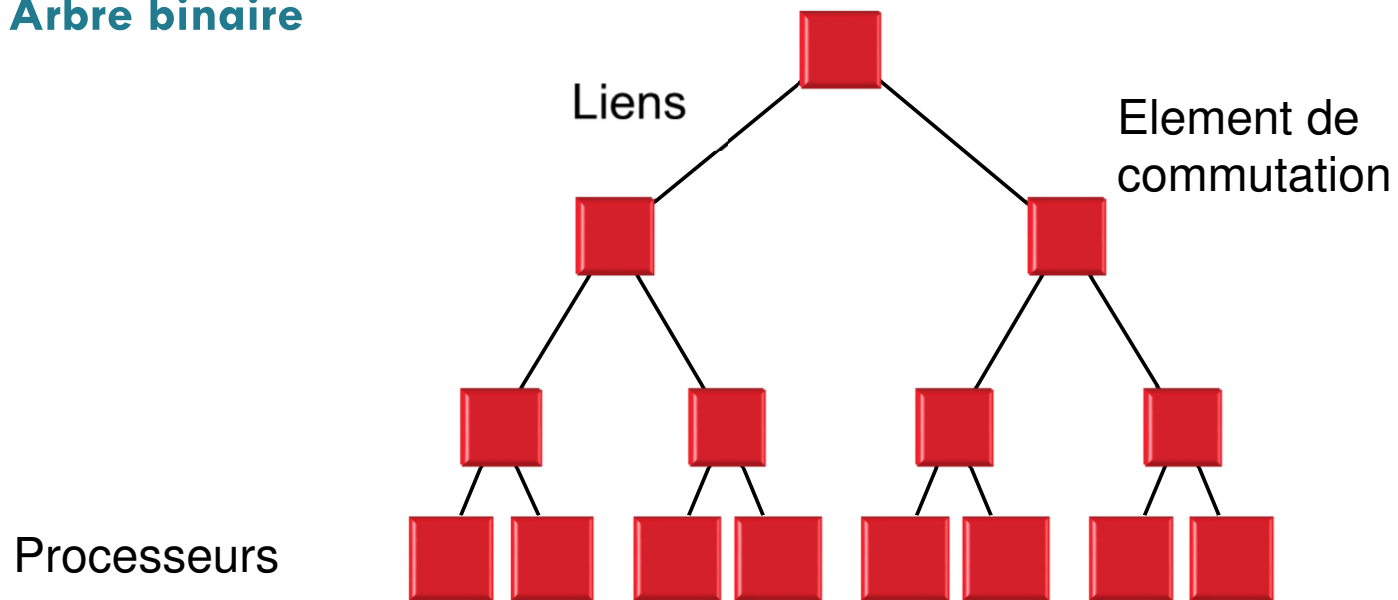


Réseaux d'interconnexion

91

Types des réseaux d'interconnexion dans une machine // DM

❖ Arbre binaire

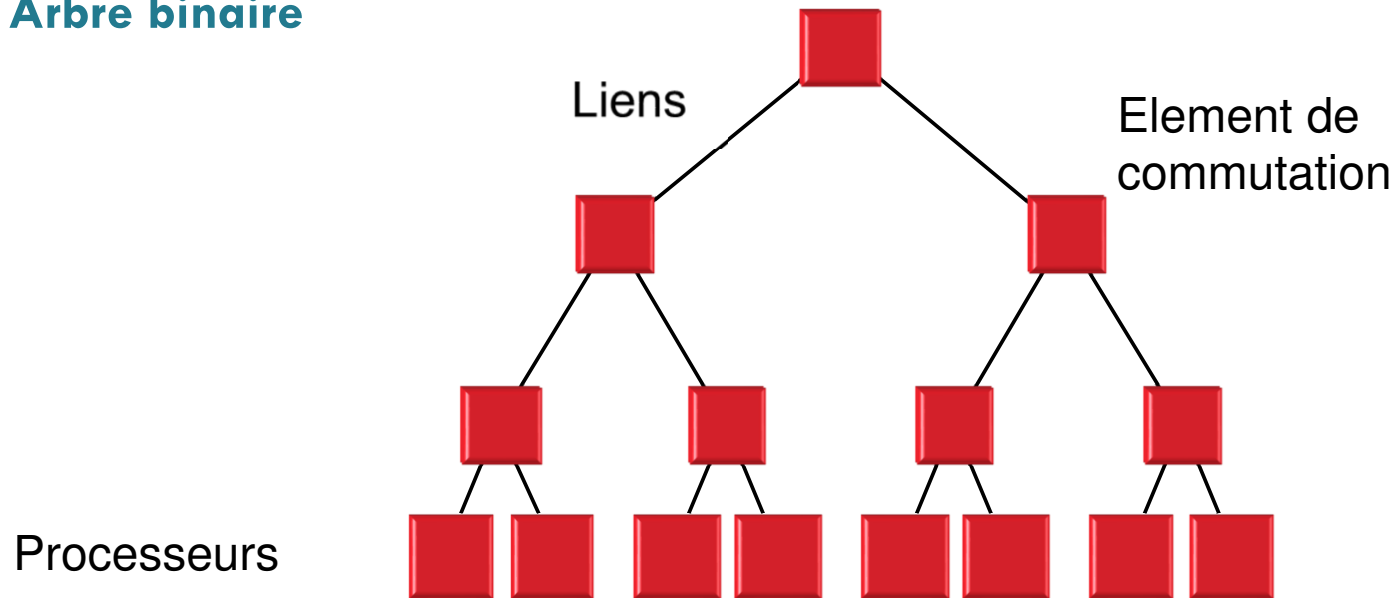


Réseaux d'interconnexion

92

Types des réseaux d'interconnexion dans une machine // DM

❖ Arbre binaire



Arbre binaire de niveau 4

$p = 2^n - 1$ nœuds (n : niveaux)

$K(G) = 1$

Réseaux d'interconnexion

93

Types des réseaux d'interconnexion dans une machine // DM

❖ Hypercube

d dimension d'un hypercube = d° de chaque nœud

- **2^d nœuds**
- **$d \cdot 2^{d-1}$ liens**
- **la numérotation des nœuds d'un d-cube est sur d bits (code de Hamming)**
- **Deux voisins : différence de numérotation d'un seul bit**
- **Un d-cube est la liaison des deux (d-1)cube**

Réseaux d'interconnexion

94

Types des réseaux d'interconnexion dans une machine // DM

❖ Hypercube

d dimension d'un hypercube = d^0 de chaque nœud

- 2^d nœuds
- $d \cdot 2^{d-1}$ liens
- la numérotation des nœuds d'un d-cube est sur d bits (code de Hamming)
- Deux voisins : différence de numérotation d'un seul bit
- Un d-cube est la liaison des deux (d-1)cube



0-cube

Réseaux d'interconnexion

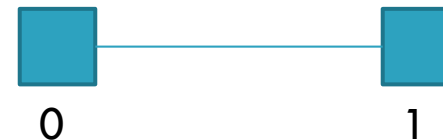
95

Types des réseaux d'interconnexion dans une machine // DM

❖ Hypercube

d dimension d'un hypercube = d° de chaque nœud

- 2^d nœuds
- $d \cdot 2^{d-1}$ liens
- la numérotation des nœuds d'un d-cube est sur d bits (code de Hamming)
- Deux voisins : différence de numérotation d'un seul bit
- Un d-cube est la liaison des deux (d-1)cube



1-cube

Réseaux d'interconnexion

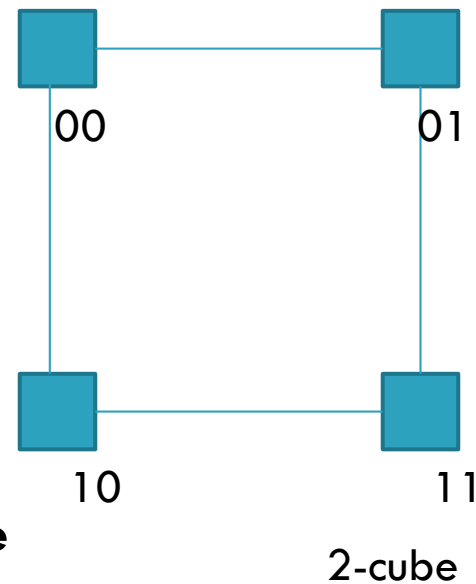
96

Types des réseaux d'interconnexion dans une machine // DM

❖ Hypercube

d dimension d'un hypercube = d° de chaque nœud

- 2^d nœuds
- $d \cdot 2^{d-1}$ liens
- la numérotation des nœuds d'un d-cube est sur d bits (code de Hamming)
- Deux voisins : différence de numérotation d'un seul bit
- Un d-cube est la liaison des deux (d-1)cube



Réseaux d'interconnexion

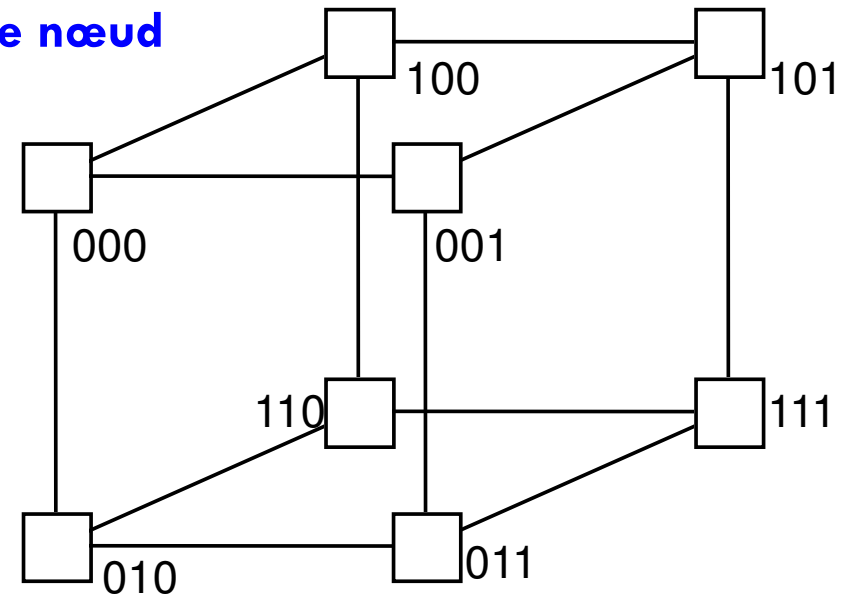
97

Types des réseaux d'interconnexion dans une machine // DM

❖ Hypercube

d dimension d'un hypercube = d° de chaque nœud

- 2^d nœuds
- $d \cdot 2^{d-1}$ liens
- la numérotation des nœuds d'un d-cube est sur d bits (code de Hamming)
- Deux voisins : différence de numérotation d'un seul bit
- Un d-cube est la liaison des deux (d-1)cube



3-cube

Réseaux d'interconnexion

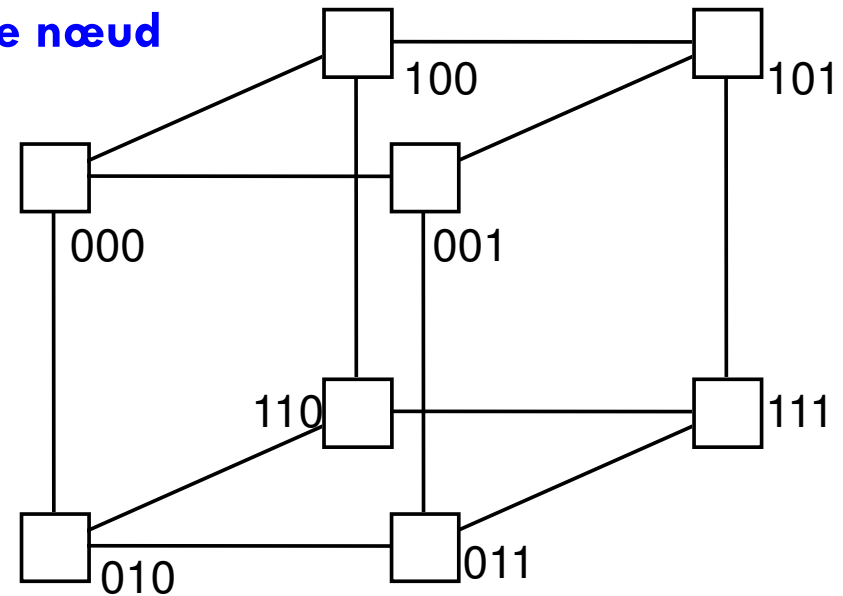
98

4. Types des réseaux d'interconnexion dans une machine // DM

❖ Hypercube

d dimension d'un hypercube = d° de chaque nœud

- 2^d nœuds
- $d \cdot 2^{d-1}$ liens
- Distance entre deux nœuds = nombre de bits différents



3-cube

Réseaux d'interconnexion

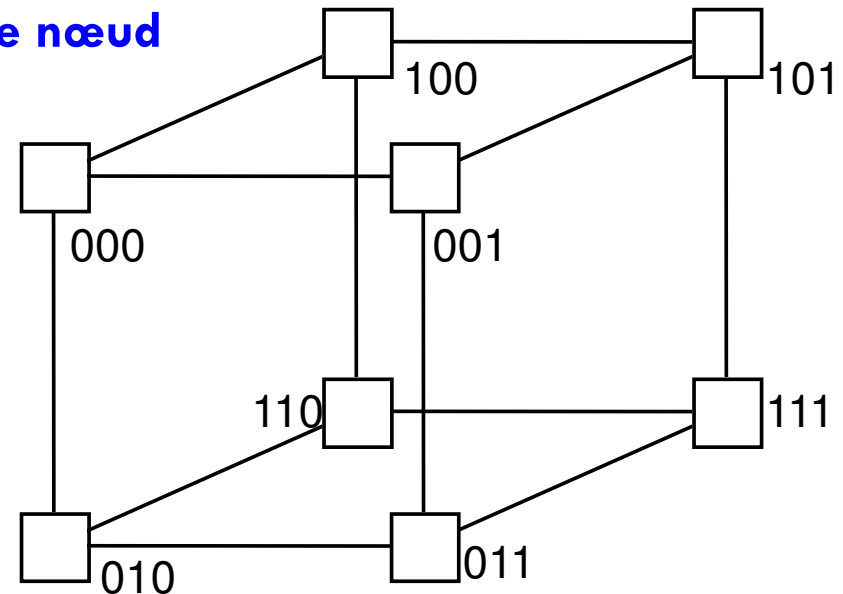
99

Types des réseaux d'interconnexion dans une machine // DM

❖ Hypercube

d dimension d'un hypercube = d° de chaque nœud

- 2^d nœuds
- $d \cdot 2^{d-1}$ liens
- Distance entre deux nœuds = nombre de bits différents
- $D(G) = d$ ➤ plusieurs chemins //



3-cube

Réseaux d'interconnexion

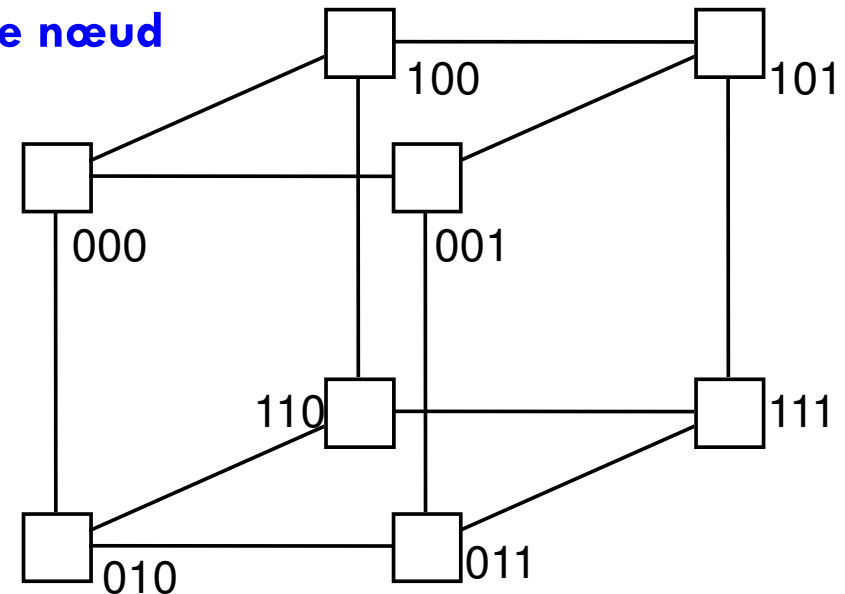
100

Types des réseaux d'interconnexion dans une machine // DM

❖ Hypercube

d dimension d'un hypercube = d° de chaque nœud

- 2^d nœuds
- $d \cdot 2^{d-1}$ liens
- Distance entre deux nœuds = nombre de bits différents
- $D(G) = d \rightarrow$ plusieurs chemins //



Pour les déterminer, il faut modifier à chaque fois un bits

3-cube

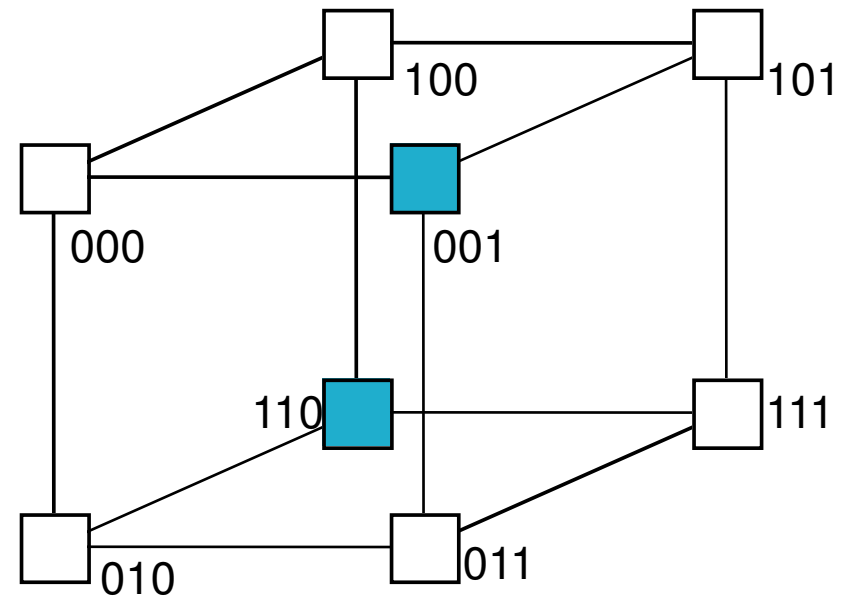
Réseaux d'interconnexion

101

Types des réseaux d'interconnexion dans une machine // DM

❖ Hypercube

Entre 001 et 110 combien de chemins // ?



3-cube

Réseaux d'interconnexion

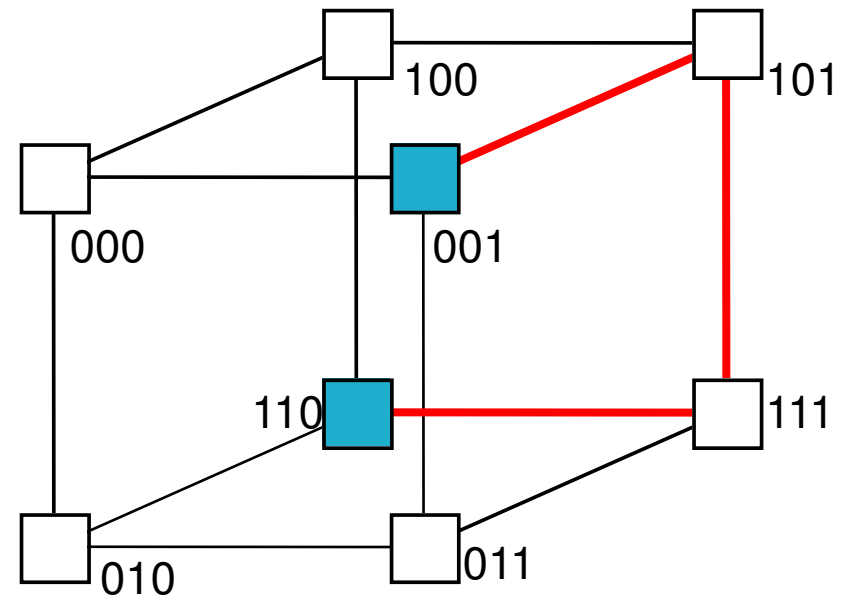
102

Types des réseaux d'interconnexion dans une machine // DM

❖ Hypercube

Entre 001 et 110 combien de chemins // ?

001 – 101 – 111 – 110



3-cube

Réseaux d'interconnexion

103

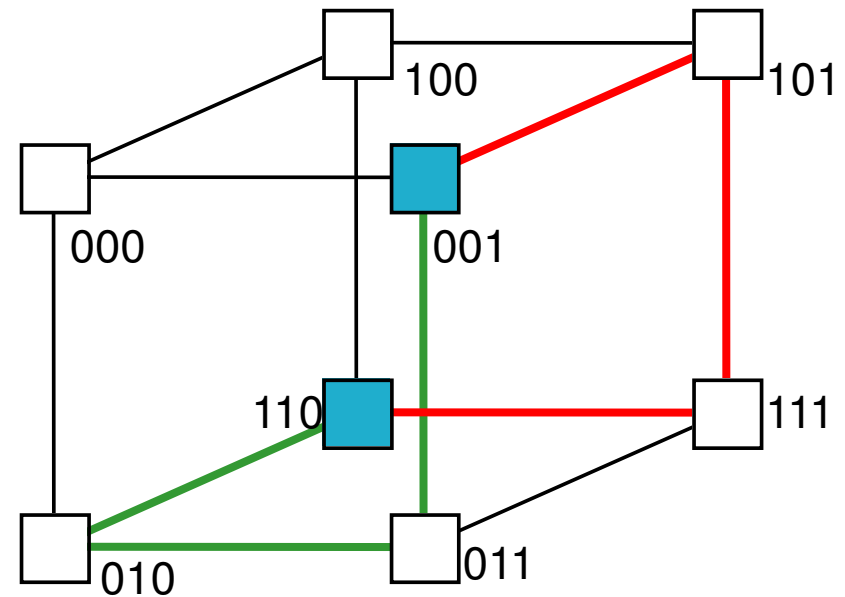
Types des réseaux d'interconnexion dans une machine // DM

❖ Hypercube

Entre 001 et 110 combien de chemins // ?

001 – 101 – 111 – 110

001 – 011 – 010 – 110



Réseaux d'interconnexion

104

Types des réseaux d'interconnexion dans une machine // DM

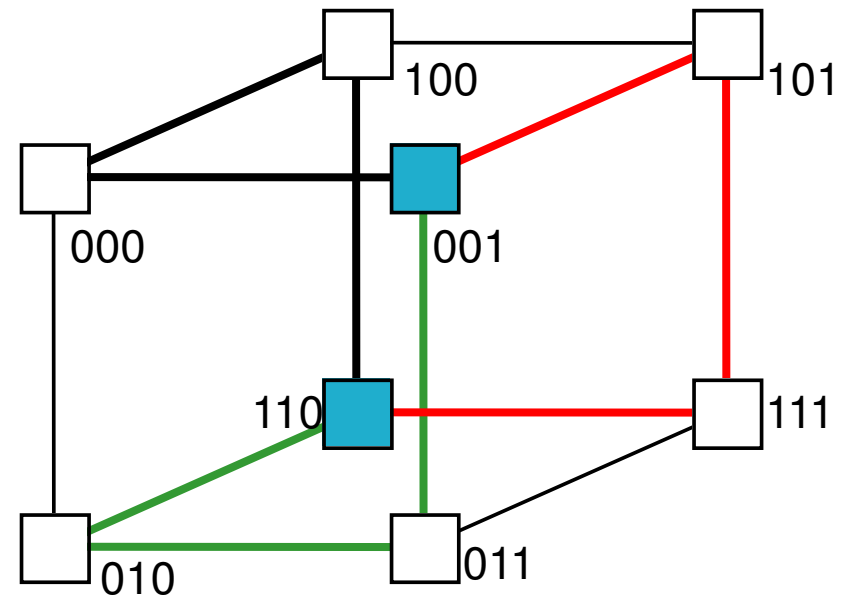
❖ Hypercube

Entre 001 et 110 combien de chemins // ?

001 – 101 – 111 – 110

001 – 011 – 010 – 110

001 – 000 – 100 – 110



3-cube

