Architecture des calculateurs

Violaine Louvet [1]

[1] Institut Camille jordan - CNRS

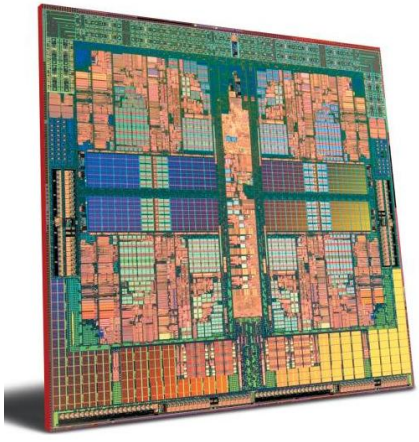
Ecole Doctorale 2012-2013

Sommaire

Importance des aspects matériels

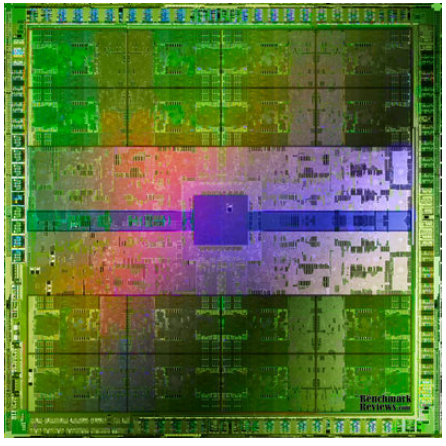
La connaissance des architectures de calcul permet :

- de choisir et d'adapter son matériel en fonction de ses besoins
- de le dimensionner correctement en retenant le meilleur compromis : architecture équilibrée
- de comprendre le comportement d'un programme
- d'adapter les méthodes numériques, les algorithmes, la programmation



Multi-cœurs

Blue-gene P



GPU

Sommaire

Bits, octets & co

- bit (b) : binary digit, plus petite unité d'information d'un composant en informatique. Vaut 0 ou 1
  - 1 bit permet donc d'avoir 2 états (0 ou 1), 2 bits permettent d'avoir 4 (2^2) états (00,01,10,11),... , n bits permettent ainsi d'avoir 2^n états
- octet (o) ou byte (B) = composé de 8 bits

Normalisation 1998

Depuis 1998 les préfixes binaires associés aux octets ou bytes ont été normalisés (kibi = kilo binaire, mébi = méga binaire ...) :

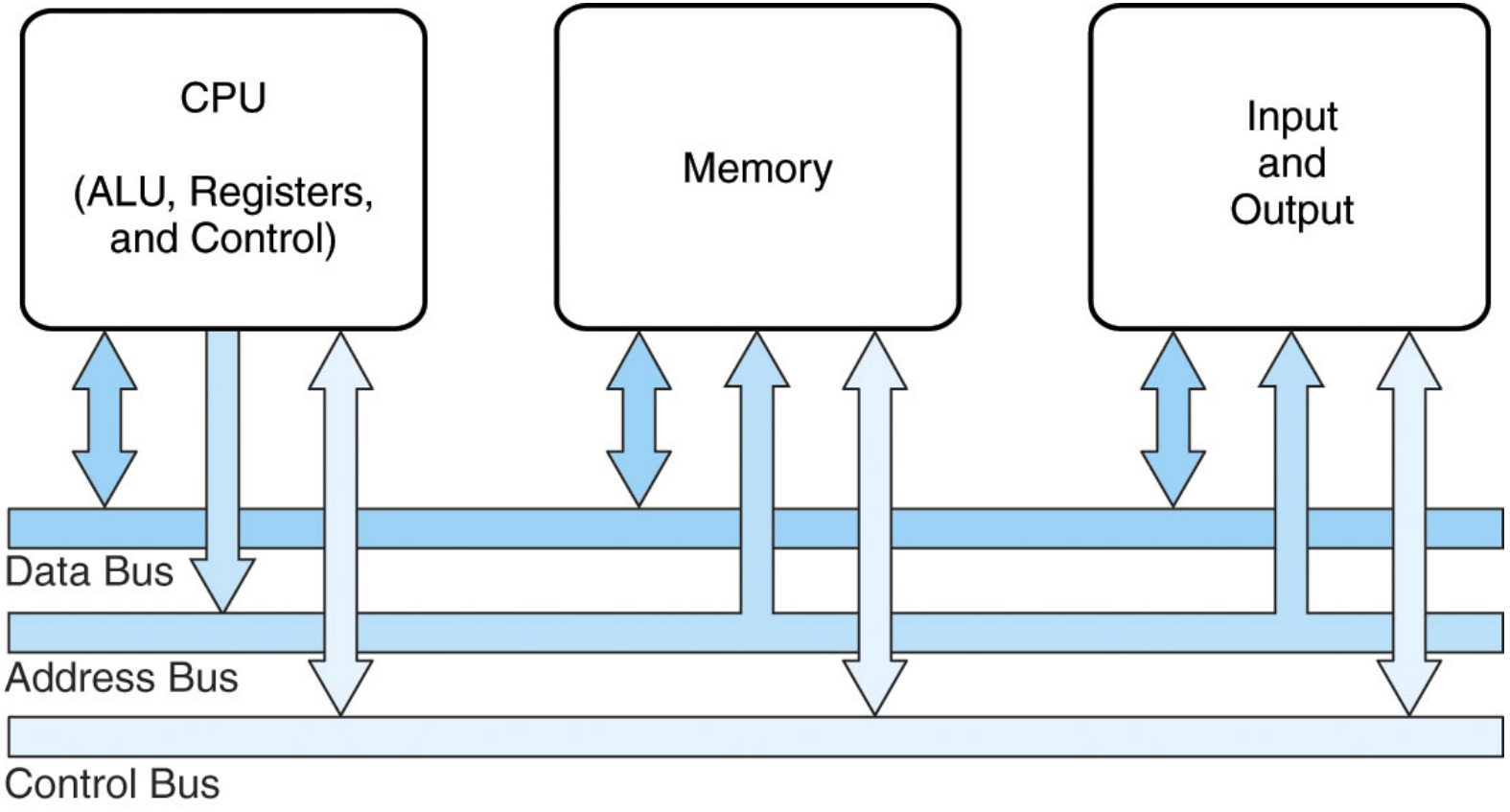
- $1 kibio = 2^{10}$ octets $= 1024$ octets $= 8192$ bits
- $1 ko = 1000$ octets

Mais attention, de façon courante : 1 kilo-octet $= 1 Ko = 1024$ octets

Usage

- Processeurs et mémoires opèrent sur des octets
- Débits des réseaux et des bus exprimés en bits par seconde

Architecture générale



Sommaire

Caractéristiques d'un processeur

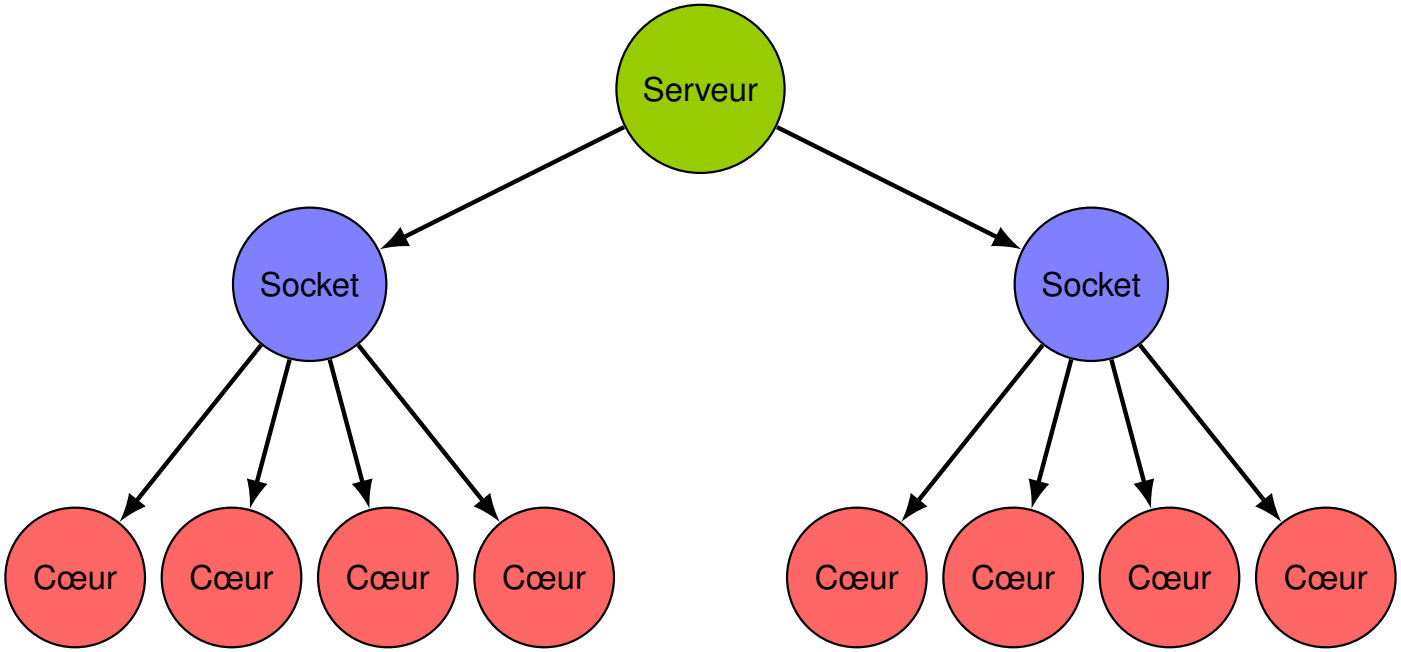
- Fréquence d'horloge (MHz) : vitesse de fonctionnement du processeur = nombre de millions de cycles que le processeur est capable d'effectuer par seconde
  - Cycle = plus petite unité de temps au niveau du processeur. Chaque opération/instruction nécessite au minimum un cycle, et plus souvent plusieurs
  - $1GHz = 10^9 Hz = 10^9 \text{cycle/s}$
- Largeur (32 ou 64 bits) : notamment du bus de données et des registres internes. Bon indicateur de la quantité d'information que celui-ci peut gérer en un temps donné
- Jeu d'instructions : ensemble des opérations qu'un processeur peut exécuter, plus ou moins complexes

Adressage mémoire

Les processeurs 32 bits ne peuvent pas adresser plus de 4 gibioctets (2^{32}) de mémoire centrale, tandis que les processeurs 64 bits peuvent adresser 16 exbioctets (2^{64}) de mémoire.

Cœurs, sockets

On évite de parler de CPUs : il faut distinguer le support (socket) de l'unité de calcul elle-même (cœur).
Le socket ou slot est le connecteur qui interfère entre la carte mère d'un ordinateur et le processeur lui-même.

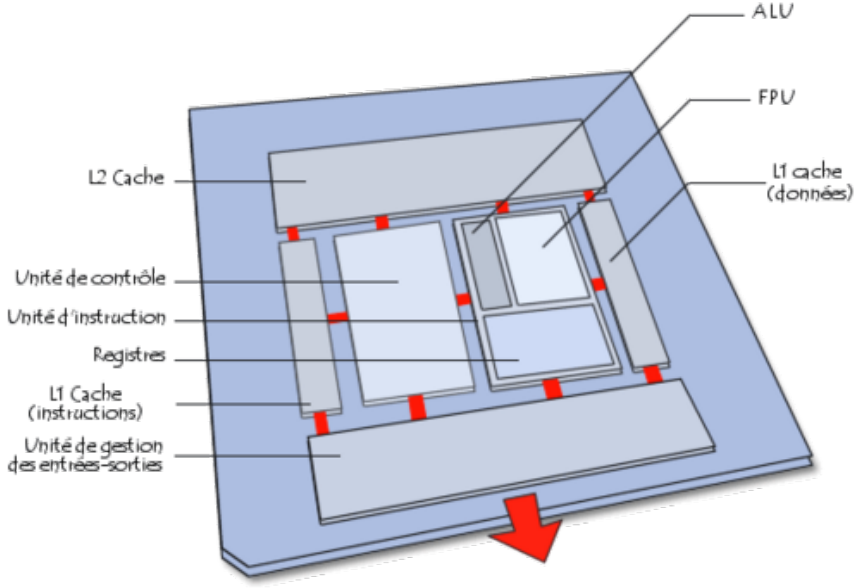


Serveur bi-sockets quadri-cœurs

Composition d'un processeur

Eléments importants du processeur :

- Plusieurs Unité Arithmétique et Logique (UAL), qui prennent en charge notamment les calculs arithmétiques élémentaires ce qui permet de traiter plusieurs instructions en même temps
- Registres : mémoires de petite taille (quelques octets) rapides
- Unité de calcul en virgule flottante (en anglais Floating Point Unit - FPU), qui permet d'accélérer les calculs sur des nombres flottants
- Mémoires caches, qui diminuent les temps d'accès à la mémoire



Fonctionnement

- Le processeur est rythmé par une horloge défini par sa fréquence (GHz)
- Pour schématiser, on peut dire que le traitement d'une instruction passe par 5 étapes fondamentales dont la durée est d'au minimum un cycle d'horloge :
  1. lecture de l'instruction (IF, Instruction Fetch)
  2. décodage de l'instruction (ID, Instruction Decode)
  3. exécution de l'instruction (EX, Execute)
  4. écriture ou chargement depuis la mémoire en fonction du type de l'instruction (MEM, Memory)
  5. stockage du résultat dans un registre (WB, Write Back)
- Plus la fréquence est élevée et plus le processeur peut traiter les instructions rapidement

5 cycles au minimum sont nécessaires pour accomplir une instruction

Pipeline

IF	ID	EX	MEM	WB					
	IF	ID	EX	MEM	WB				
		IF	ID	EX	MEM	WB			
			IF	ID	EX	MEM	WB		
				IF	ID	EX	MEM	WB	

- Chaque « travailleur » effectue le même travail sur des objets différents
- Si il y a 5 étapes dans la chaîne, 5 objets peuvent être traités simultanément
- Et si ces étapes prennent le même temps, la chaîne est continuellement occupée

Jeux d'instructions (ISA, Instructions Set Architecture)

- Ensemble des opérations élémentaires qu'un processeur peut accomplir
- Plusieurs familles de processeurs possédant chacune un jeu d'instructions propre :
  - 80x86 : le « x »représente la famille. On parle ainsi de 386, 486, 586, 686, etc.
  - ARM
  - IA-64
  - PowerPC ...

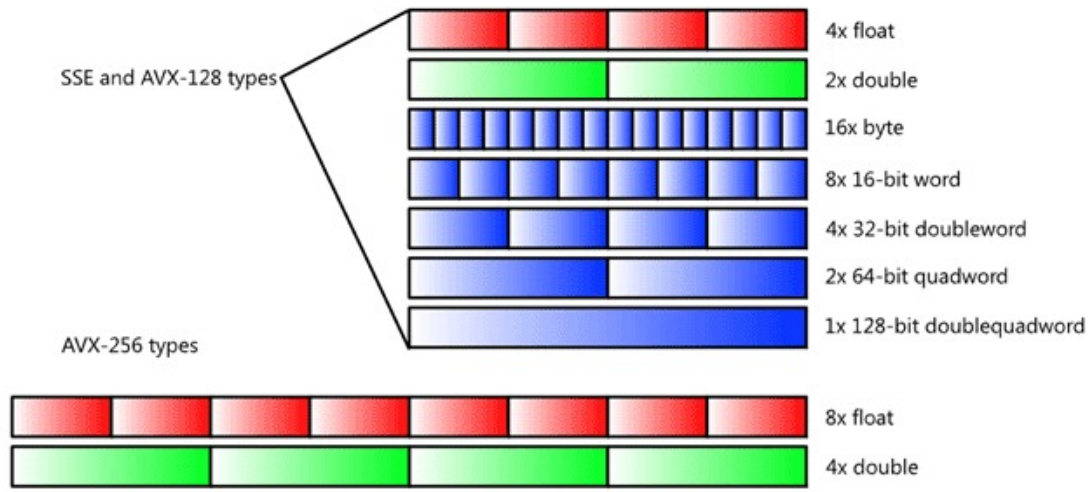
Attention

Un programme réalisé pour un type de processeur ne peut pas fonctionner directement sur un système possédant un autre type de processeur.

Parallélisation de données

SIMD (Single Instruction Multiple Data) : appliquer la même instruction simultanément à plusieurs données

- Jeux d'instructions :
  - SSE (Streaming SIMD Extensions) chez Intel
  - AVX (Advanced Vector Extensions) chez Intel et AMD (prise en charge de registres vectoriels plus grands)
- Capacités vectorielles des processeurs : possibilité d'exécuter une opération arithmétique sur un registre vectoriel (contenant deux DP ou quatre SP en général).



Parallélisation de données

Exemple SSE

Algorithme :

```
for i = 1,n
    x[i] = sqrt(x[i])
```

Travail du processeur :

```
for i = 1,n
    load x[i] to the floating-point register
    calculate the square root
    write the result from the register to memory
```

Avec SSE :

```
for {i1,i2,i3,i4} in {1:n}
    load x[i1],x[i2],x[i3],x[i4] to the SSE register
    calculate 4 square roots in one operation
    write the result from the register to memory
```

Hyperthreading

- SMT = Simultaneous MultiThreading, plus connu sous le nom d'Hyperthreading chez Intel.
- Principe : créer deux processeurs logiques sur une seule puce, chacun doté de ses propres registres. Ces deux unités partagent les éléments du cœur physique comme le cache et le bus système.
- Dégradation des performances individuelles des threads mais amélioration des performances de l'ensemble.

Attention

Ce fonctionnement est généralement activé par défaut.

En pratique

Sur un core I7, commande en ligne *cpuinfo* :

```
Intel(R) Core(TM) i7  Processor (Intel64 Q 820  )
=====  Processor composition  =====
Processors(CPUs)  : 8
Packages(sockets) : 1
Cores per package : 4
Threads per core  : 2
```

Flops et performance du processeur

- Opérations (additions ou multiplications) à virgule flottante par seconde = FLoating point Operations Per Second

- Puissance crête (point de vue théorique) : mesure les performance des unités de calcul en virgule flottante (FPU) contenues dans le cœur.
  - Processeur Intel Sandy Bridge six core, 3.2 GHz : 4 opérations flottantes possibles par cycle en DP soit 8 en SP, 6 cœurs par nœud : $8 \times 6 \times 3.2 = 153.6 GFlops$.
  - Processeur PowerPC A2 (Blue Gene/Q), 1.6 GHz : 4 opérations flottantes possibles par cycle en DP, soit 8 en SP, 16 cœurs par nœud : $8 \times 16 \times 1.6 = 204.8 GFlops$.

- D'un point de vue pratique, la puissance d'une machine dépend de l'ensemble de ses composants : fréquence du processeur, accès mémoire, vitesse des bus, complexité de l'architecture ... mais aussi charge de la machine, système d'exploitation ... On est souvent loin de la puissance théorique ...

Sommaire

Quelques définitions

- Bande passante = débit d'informations ; d'un périphérique : mémoire, disque dur ... ou d'un médium de communication : réseau, bus ....
  - Mesurée généralement en octets (byte) par seconde (o/s, ou B/s) ou en bits par seconde (bit/s ou bps)
  - Exemple : USB 2.0 $\longrightarrow$ 480 Mbits par seconde. Le transfert d'une vidéo de 900 Mo mettra donc 15 secondes ($900 Mo = 8 \times 900 Mbits = 7200 Mbits$; $7200/480 = 15 s$).

- Latence = temps minimum d'établissement de la connexion : indépendant de la quantité de données à transporter
  - Exemple : latence de l'ADSL : 30 ms, latence de l'infiniband QDR : 100 ns

Problématique

« Memory wall »

- L'accès aux données est un des principaux facteurs limitant la performance
- Les processeurs sont déséquilibrés en ce qui concerne le rapport performance crête / bande passante mémoire

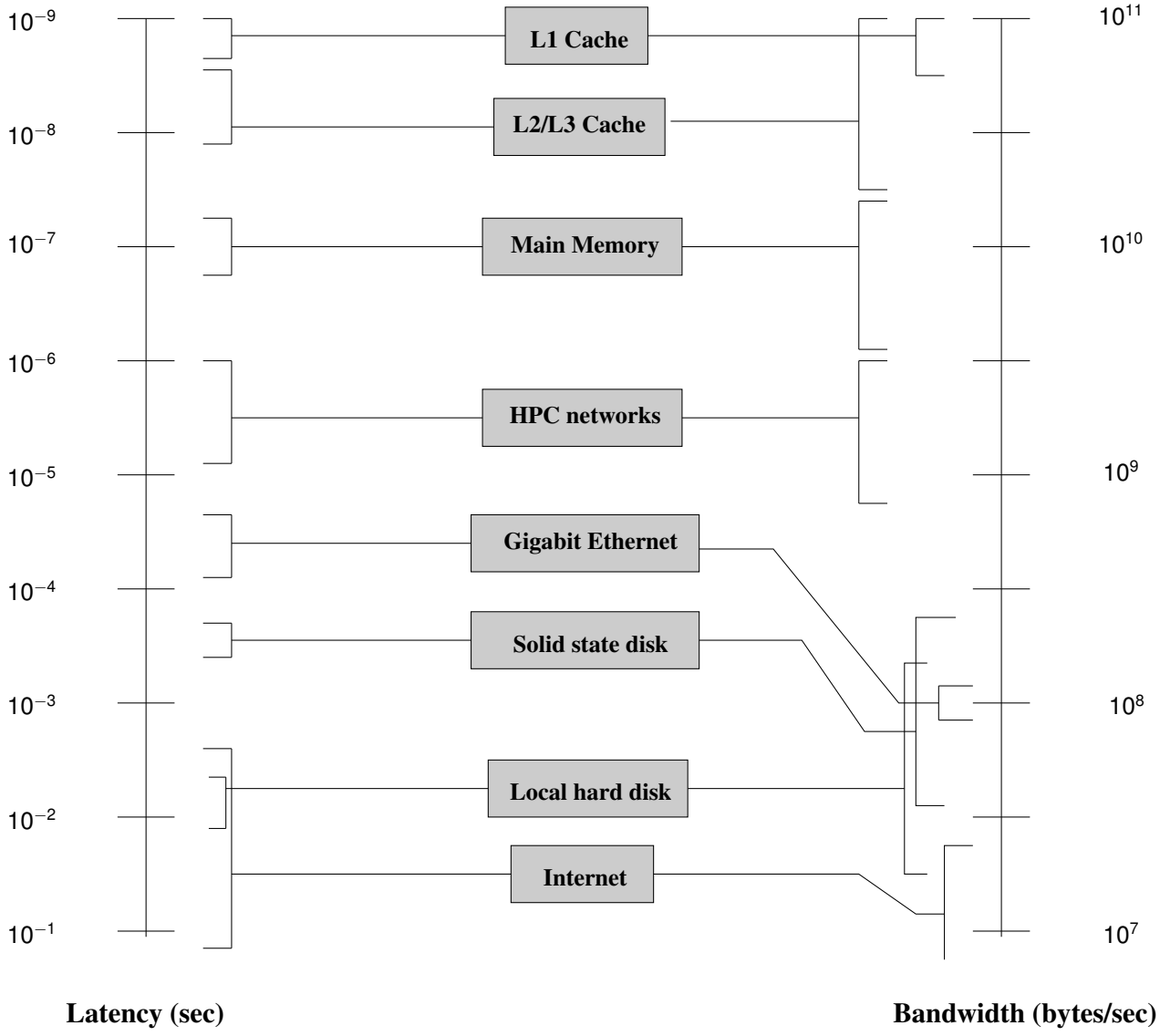
Equilibre d'une machine

- Ratio entre la bande passante mémoire et la performance crête

$$B_m = \frac{\text{memory bandwith (GWords/sec)}}{\text{peak performance (GFlops/sec)}}$$

data path	**balance** (W/F)
cache	0.5 - 1.0
machine (RAM)	0.03 - 0.5
réseau rapide	0.001 - 0.02
GBit ethernet	0.0001 - 0.0007
disque	0.0001 - 0.01

Hiérarchie Mémoire

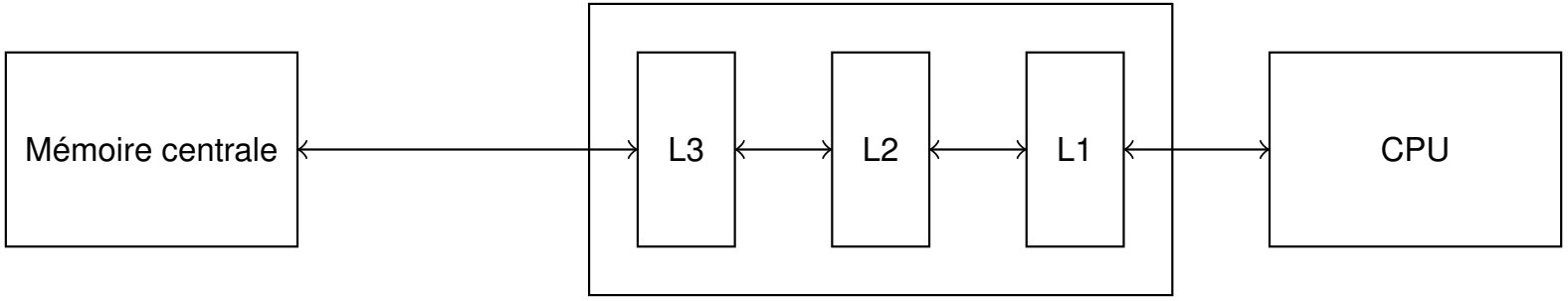


Accès mémoire : problématique

Quels facteurs influencent les performances des accès mémoire ?

- Localisation : cache, RAM, disque ?
- Manière dont elles sont accédées :
  - au travers d'un chipset (jeu de composants électroniques qui permet le contrôle des échanges d'information : Northbridge par exemple)
  - directement par le processeur
  - via le processeur voisin ...

Plus une donnée est proche du processeur, plus elle est accédée rapidement.



Principes de localité

Localité spatiale

Lorsqu'un programme accède à une donnée ou à une instruction, il est probable qu'il accédera ensuite aux données ou instructions voisines

Localité temporelle

Lorsqu'un programme accède à une donnée ou à une instruction, il est probable qu'il y accédera à nouveau dans un futur proche

Exemple

```
subroutine sumVec(vec,n)
    integer :: n
    integer :: vec(n)
    integer :: i,sum=0
    do i = 1,n
        sum = sum + vec(i)
    end do
end subroutine
```

- Bonne localité spatiale des données du tableau `vec` : accès en séquence
- Bonne localité temporelle de la donnée `sum` : accès fréquent
- Bonne localité spatiale et temporelle des instructions : accès en séquence et fréquemment

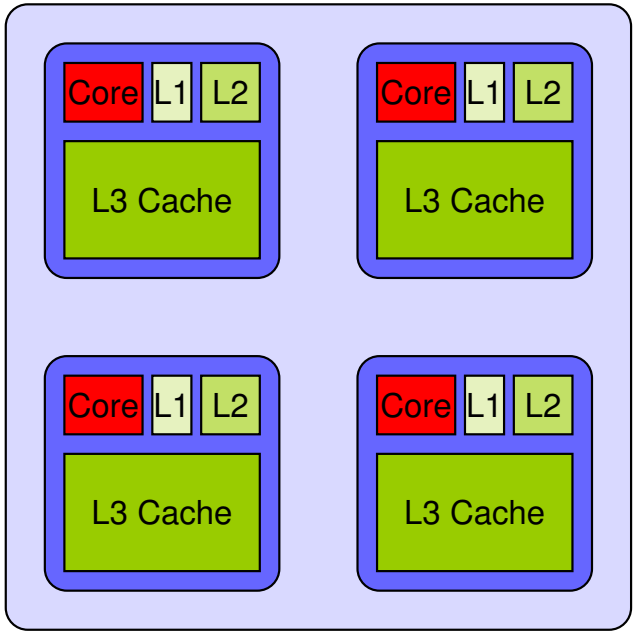
Fonctionnement des caches

- Le cache est divisé en lignes (ou blocs) de mots
- 2 niveaux de granularité :
  - le CPU travaille sur des mots (par ex 32 ou 64 bits)
  - les transferts mémoire se font par ligne (ou bloc, par ex 256 octets)
- Les lignes de caches sont organisées en ensembles à l'intérieur du cache, la taille de ces ensembles est constante et appelée le degré d'associativité.
- Exploitation de la localité spatiale : le cache contient des copies des mots par lignes de cache
- Exploitation de la localité temporelle : choix judicieux des lignes de cache à retirer lorsqu'il faut rajouter une ligne à un cache déjà plein

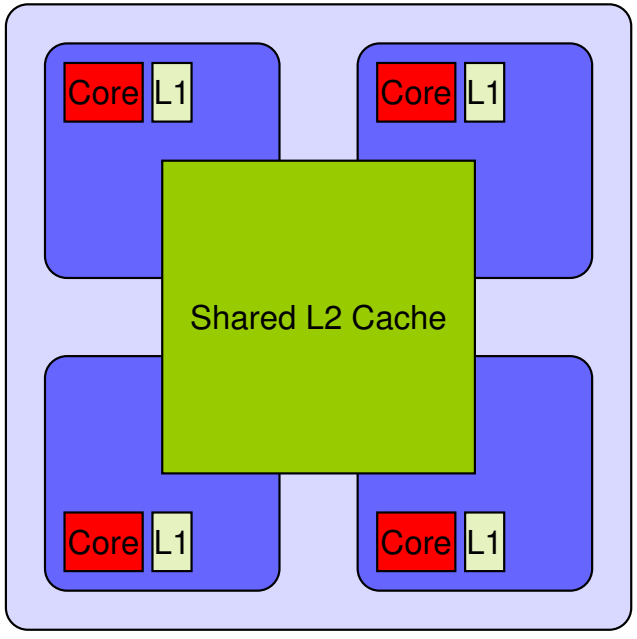
Lorsque le processeur tente d'accéder à une information (instruction ou donnée)

Si l'information se trouve dans le cache (hit), le processeur y accède sans état d'attente, sinon (miss) le cache est chargé avec un bloc d'informations de la mémoire

Organisation des caches



Caches indépendants



Caches partagés

- Problème de cohérence
- Problème d'accès concurrent

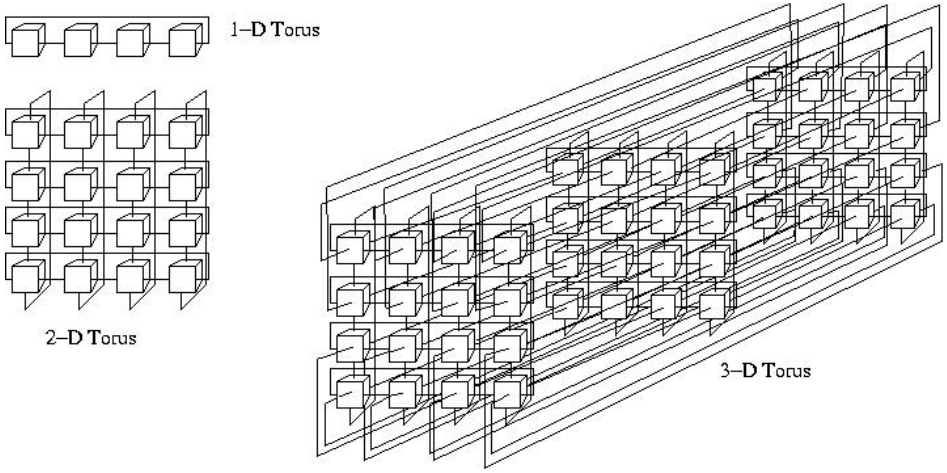
Page faults / Caches miss

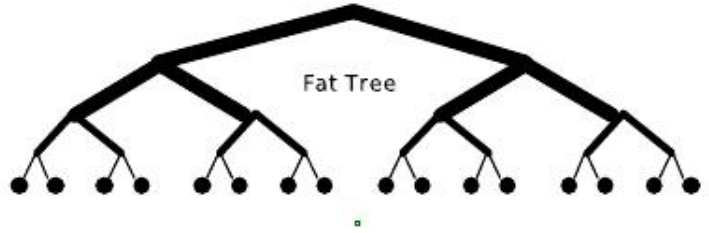
- Défaut de cache (cache miss) : lorsque le processeur essaie d'accéder une donnée qui n'est pas dans le cache
  - Raté obligatoire (compulsory miss) : premier accès au bloc pendant l'exécution du programme. Inévitable.
  - Raté par capacité insuffisante (capacity miss) : accès à un bloc qui a été remplacé par un autre dans le cache car la capacité du cache est insuffisante.
- Défaut de page (page fault) : lorsqu'un processus tente d'accéder à une adresse logique correspondante à une page non résidente en mémoire centrale $\longrightarrow$ interruption du processus le temps de charger la page.
  - Temps de traitement d'un défaut de page : $\sim 25ms$

Sommaire

Caractéristiques des réseaux

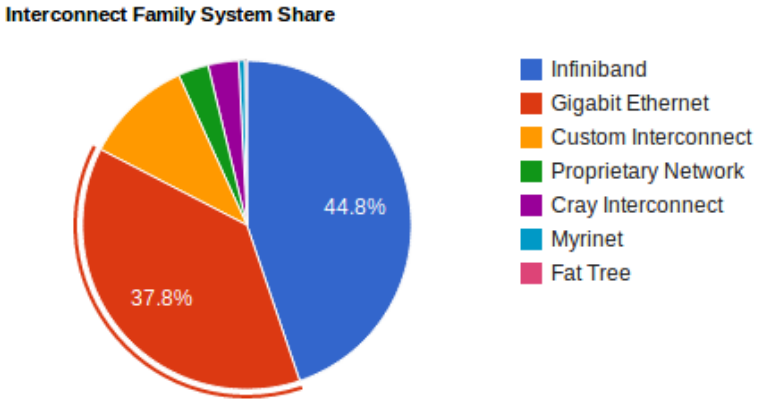
- La bande passante (bandwidth) est le débit maximal.
  - La bande passante effective est généralement inférieure à la bande passante physique (souvent à cause de l'overhead du protocole)
  - Ce qui intéresse l'utilisateur est la bande passante MPI
  - Les performances bas niveaux sont intéressantes pour le stockage
- La latence : temps d'initialisation de la communication.
- La distance maximum entre deux nœuds.
- La topologie : tore, fat-tree ...





Technologies réseaux

Technologie	Latence	Bande passante
1 Gb Ethernet	-	1 Gb/sec
10 Gb Ethernet	$\sim$10 μs	10 Gb/sec
Infiniband	< 2 μs	10, 20 & 40 Gb/sec
Myrinet	2.5 - 5.5 μs	10 Gb/sec



Répartition des familles d'interconnexion, top 500, nov 2012

Sommaire

Goulot d'étranglement

Le débit global est dépendant de toute la chaîne d'I/Os (baie, serveur, interconnexion ...).

Bottleneck

Loi d'Amdahl sur l'équilibre des machines de calcul : une instruction par seconde requiert un octet de mémoire et un bit/seconde de capacité d'entrée-sortie.

- La capacité de stockage augmente beaucoup plus vite que la vitesse d'accès
- Le débit vers les disques augmente moins vite que la puissance de calcul
- La quantité de données générées augmente avec la puissance de calcul
- L'approche classique un fichier/processus génère de plus en plus de fichiers
- Trop de fichiers = saturation des systèmes de fichiers (nombre maximum de fichiers et espace gaspillé à cause des tailles de bloc fixes)

Sommaire

Sommaire

Kilo, Méga, Giga, Téra, Péta, Exa ...

1er Top 500 Bench Linpack	Puissance soutenue en Gflops	Puissance crête en Gflops	Nombre de processeurs ou de cœurs
Juin 1993	59.7	131	1024
Nov 2012	17 590 000	27 112 500	560 640

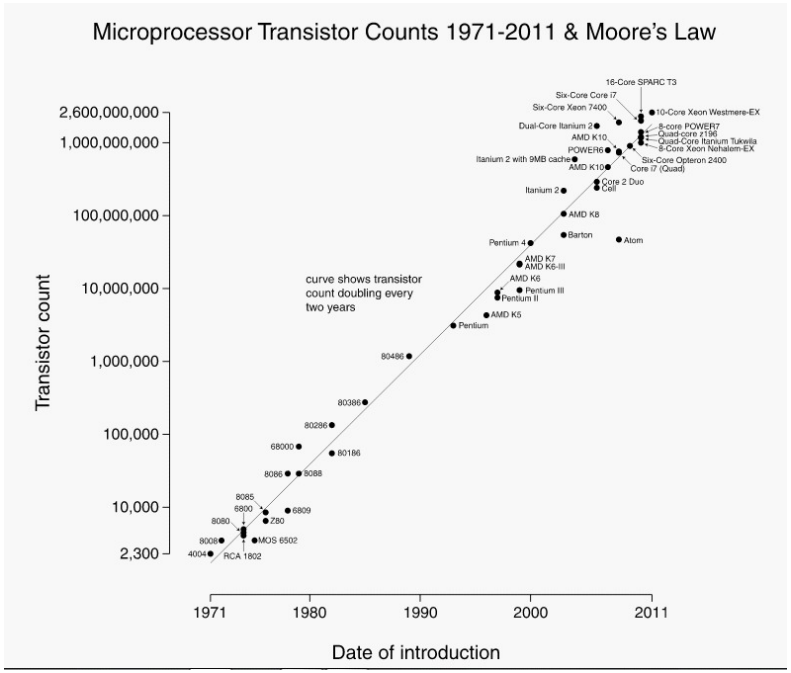
1997 : année du Teraflops (10^{12} Flops)

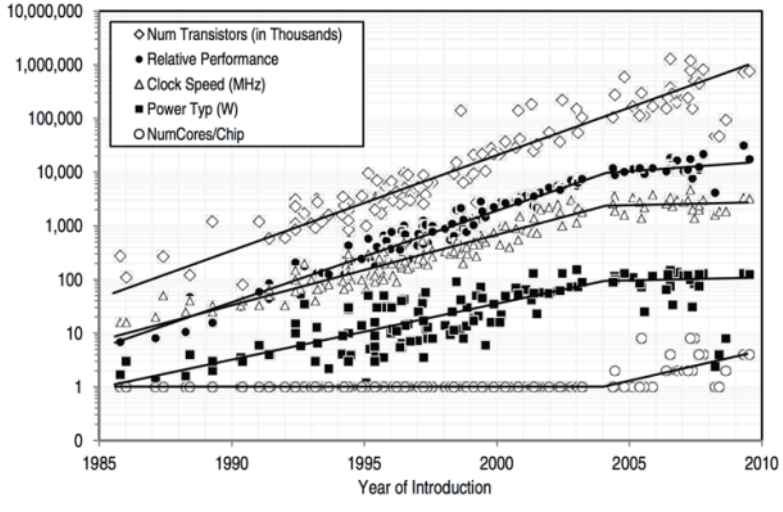
2008 : année du Petaflops (10^{15} Flops)

2017 : année de l'Exaflops (10^{18} Flops) ? ? ?

Comment faire des calculateurs plus rapides ?

1. Améliorer la vitesse du processeur : augmenter la fréquence d'horloge.
2. Augmenter le nombre de transistors, donc augmenter le nombre de cœurs.
3. Utiliser des processeurs spécialisés ou des cartes accélératrices.





Les principales limitations

Consommation électrique

- Augmente de façon exponentielle en fonction de la fréquence d'horloge :

$$P \propto f^3$$

  - P : puissance (Watt)
  - f : fréquence
- Mais pas que ! ! tous les éléments sont consommateurs : mémoires, carte mère, alimentation inefficace ...
- *Pour fonctionner à une vitesse d'un exaflops avec ses composants actuels, le super-calculateur Tianhe nécessiterait une puissance de 1,6 milliard de watts, soit un peu plus que la puissance de la centrale nucléaire de Belleville dans le Cher !*

Les principales limitations

Dissipation thermique

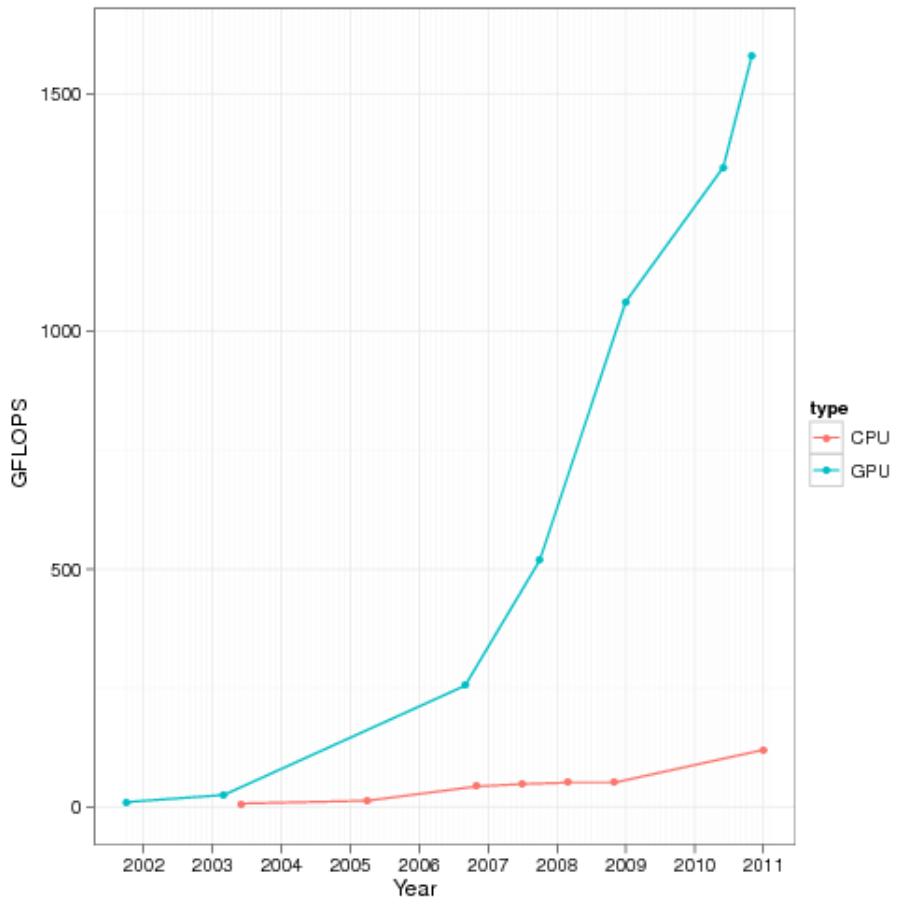
- Directement liée à la puissance consommée : plus on augmente la fréquence, plus la dissipation thermique est importante
- Problème du refroidissement au niveau des chips, et au niveau de l'infrastructure

Finesse de gravure

- Actuellement 22 nm. Feuille de route Intel : 4 nm en 2022.
- Plus de transistors donc plus de cœurs, et moins de dissipation thermique (plus c'est petit moins ça dissipe)
- Mais des défis technologiques pour les fondeurs et des technologies de plus en plus coûteuses !
- Jusqu'à quand ? ?

Sommaire

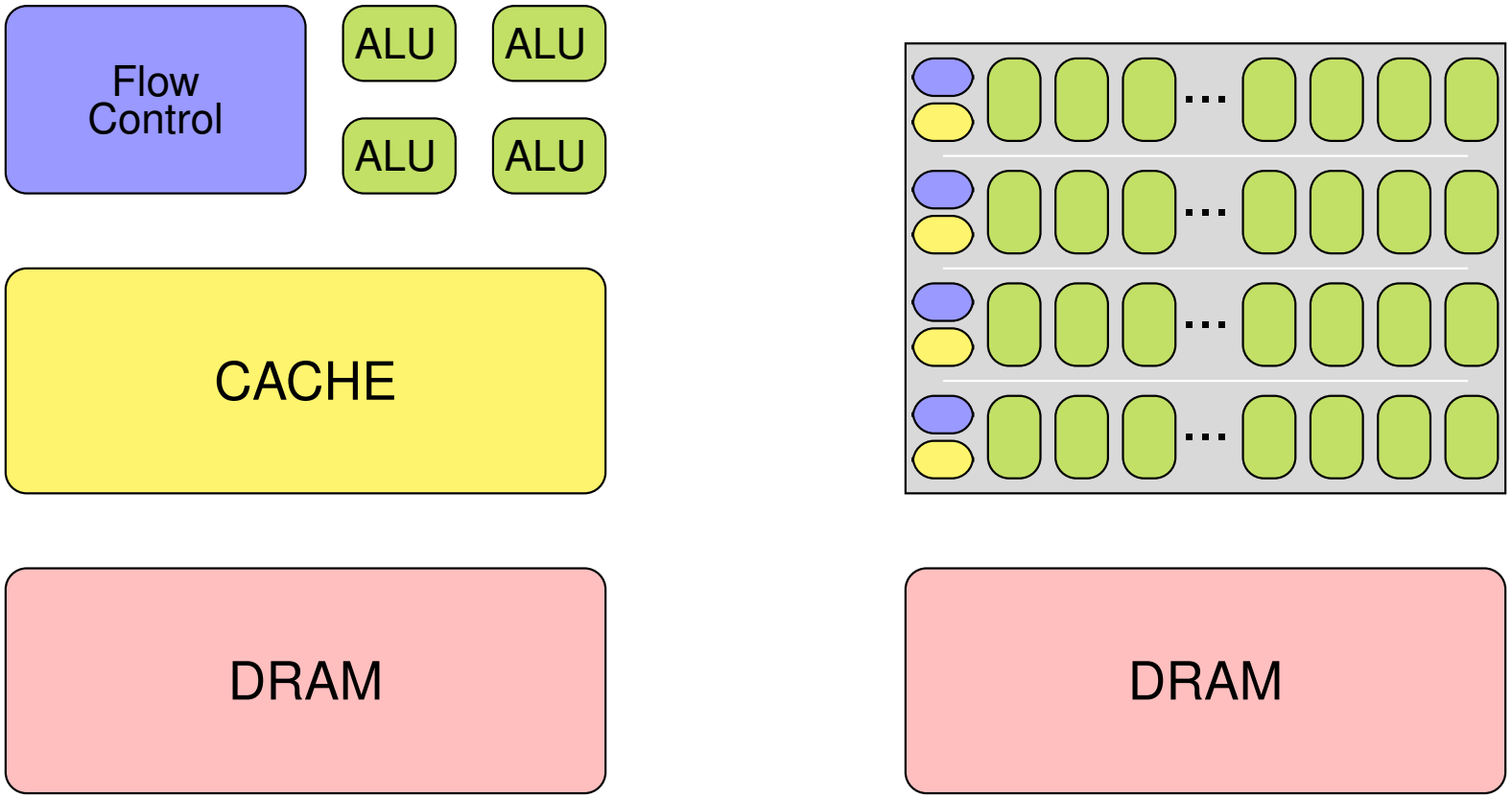
Pourquoi les GPU ?



A performance théorique égale, les plateformes à base de GPU :

- occupent moins de place
- sont moins chères
- sont moins consommatrices d'électricité

Architecture comparée d'un CPU et d'un GPU



Caractéristiques d'un GPU

- Beaucoup de cœurs : 2496 dans le Nvidia Tesla K20 (Kepler) avec une fréquence de 705 MHz.
- Support IEEE double précision en virgule flottante
- Mémoire : jusqu'à 6 GB

Exploitation d'un GPU

- Programmation de type SIMD, gestion de milliers de thread
- Transfert CPU-GPU très couteux
- API de programmation (CUDA, OpenCL)

→ Complexe à exploiter, toutes les applications ne s'y prêtent pas

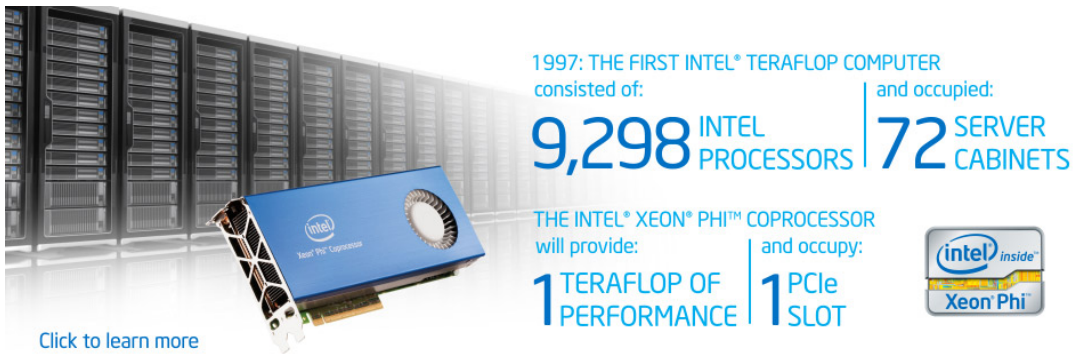
Principaux fournisseurs

- Nvidia
- Depuis peu, AMD avec sa carte graphique serveur FirePro S10000

Sommaire

Evolution actuelle

- Apparition sur le marché de cartes Many Integrated Core : les Xeon Phi©
- Intel Xeon Phi 5110P : 60 coeurs à 1053 MHz, 8 Go de mémoire



Principal avantage

- Les Xeon Phi supportent un environnement de programmation complètement standard.

Sommaire

Sommaire

Micro éclairage sur le HPC mondial

Top 500

- Projet de classification des 500 premiers supercalculateurs connus au monde.
- Depuis juin 1993, liste mise à jour tous les six mois
- LinPack : créé par Jack Dongarra, il mesure le temps mis par un ordinateur pour résoudre un système de n équations à n inconnues dense, la solution étant obtenue par une utilisation partielle du pivot de Gauss, par $2/3.n^3 + n^2$ opérations à virgule flottantes. La performance est ensuite calculée en divisant le nombre d'opérations par le temps mis, donc en FLOPS.

Top 500, novembre 2012

Petit aperçu du classement

Site	Const.	Computer	Country	RMax (TFlops/s)
Oak Ridge	Cray	Cray XK7 / Nvidia K20x	USA	17590.0
LNLL	IBM	BlueGene/Q	USA	16324.8
Riken AICS	Fujitsu	K computer (Sparc64)	Japon	10510.0
Argonne	IBM	BlueGene/Q	USA	8162.4
FZJ (Juelich)	IBM	BlueGene/Q	Allemagne	4141.2

Quelques statistiques

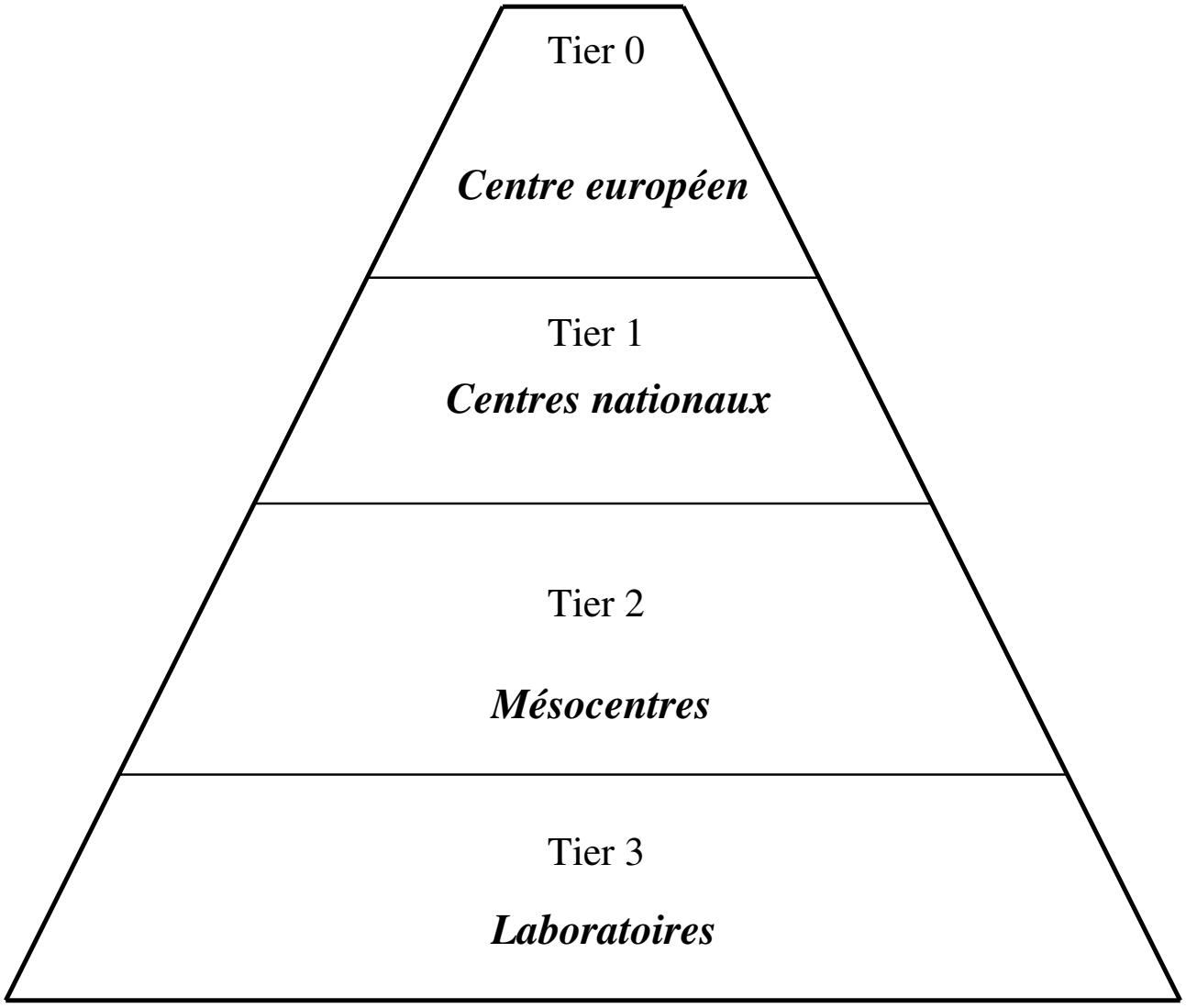
- Cartes accélératrices : 87% sans, 10% Nvidia, 1.4% Xéon Phi.
- Pays : 50.2% USA, 14.4% Chine, 6.4% Japon, 4.8% UK, 4.2% France, 3.8% Allemagne

IBM BlueGene

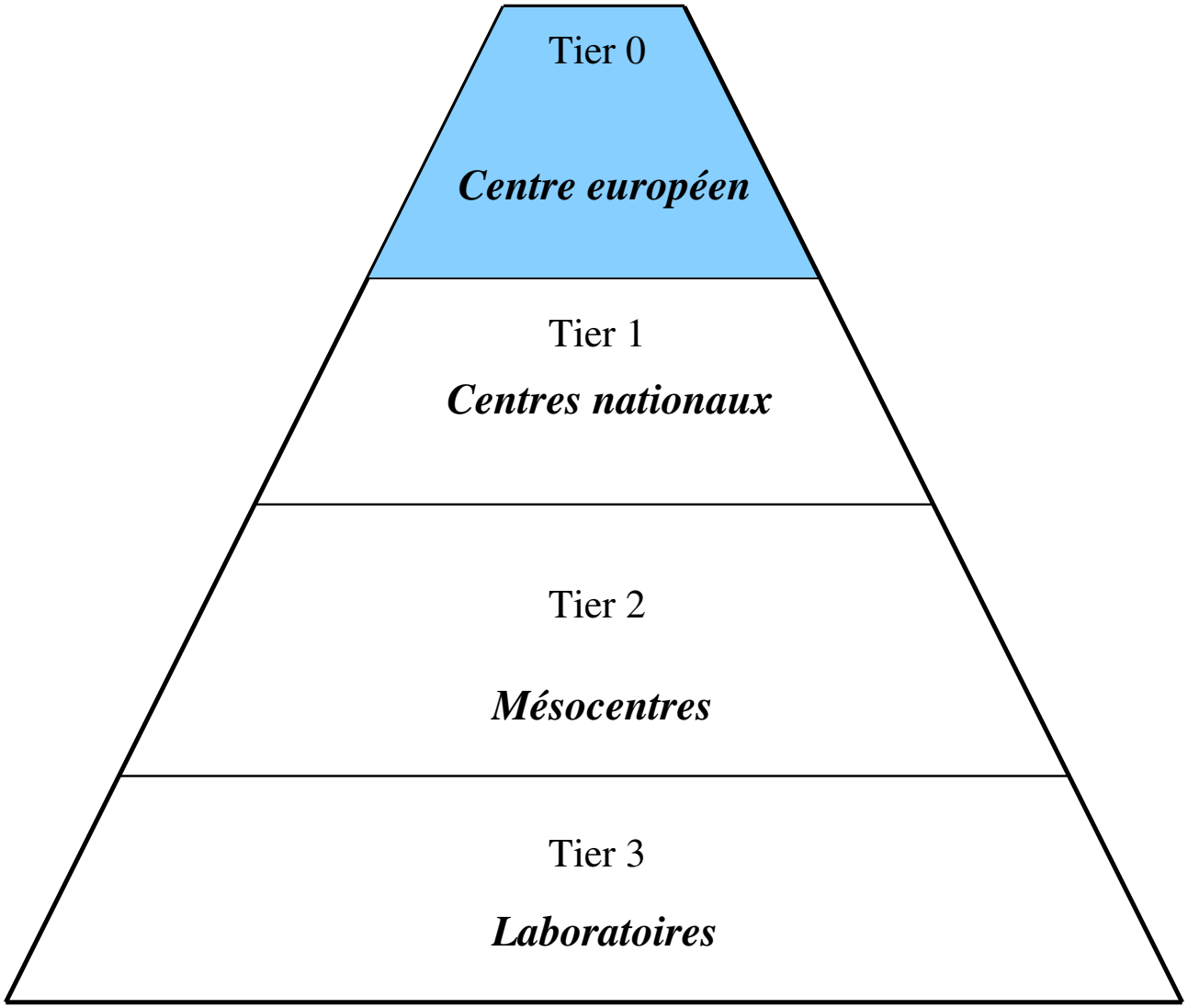
- Augmenter le nombre de coeurs en réduisant leur fréquence.
- Diminution de l'écart entre les débits et les latences mémoires, diminution de l'écart entre les débits et les latences réseaux $\longrightarrow$ machine plus équilibrée, une puissance de calcul élevée à consommation électrique équivalente
- Mais chaque coeur est lent, un très grand nombre de coeurs, peu de mémoire par coeur

Sommaire

Paysages européen et français



Paysages européen et français



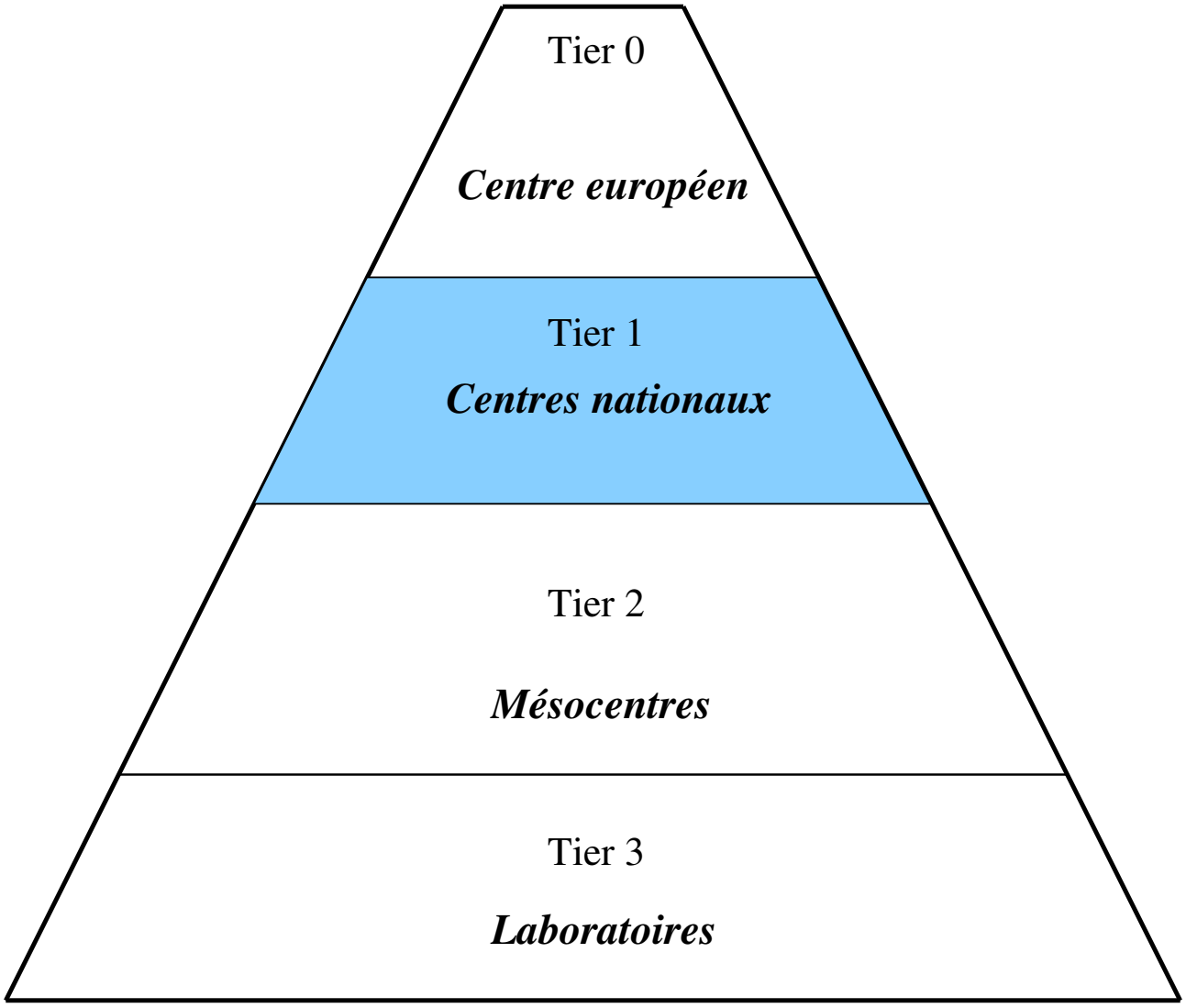
PRACE et l'échelle européenne

PRACE : Partnership for Advanced Computing in Europe

- Regroupement de 25 pays (dont Allemagne, Espagne, Grande-Bretagne, France ...)
- Accès à 6 supercalculateurs aux architectures complémentaires, d'une puissance crête globale de près de 15 petaflops, localisés en Allemagne, en Espagne, en France (CURIE) et en Italie.

`http://www.prace-ri.eu/`

Paysages européen et français



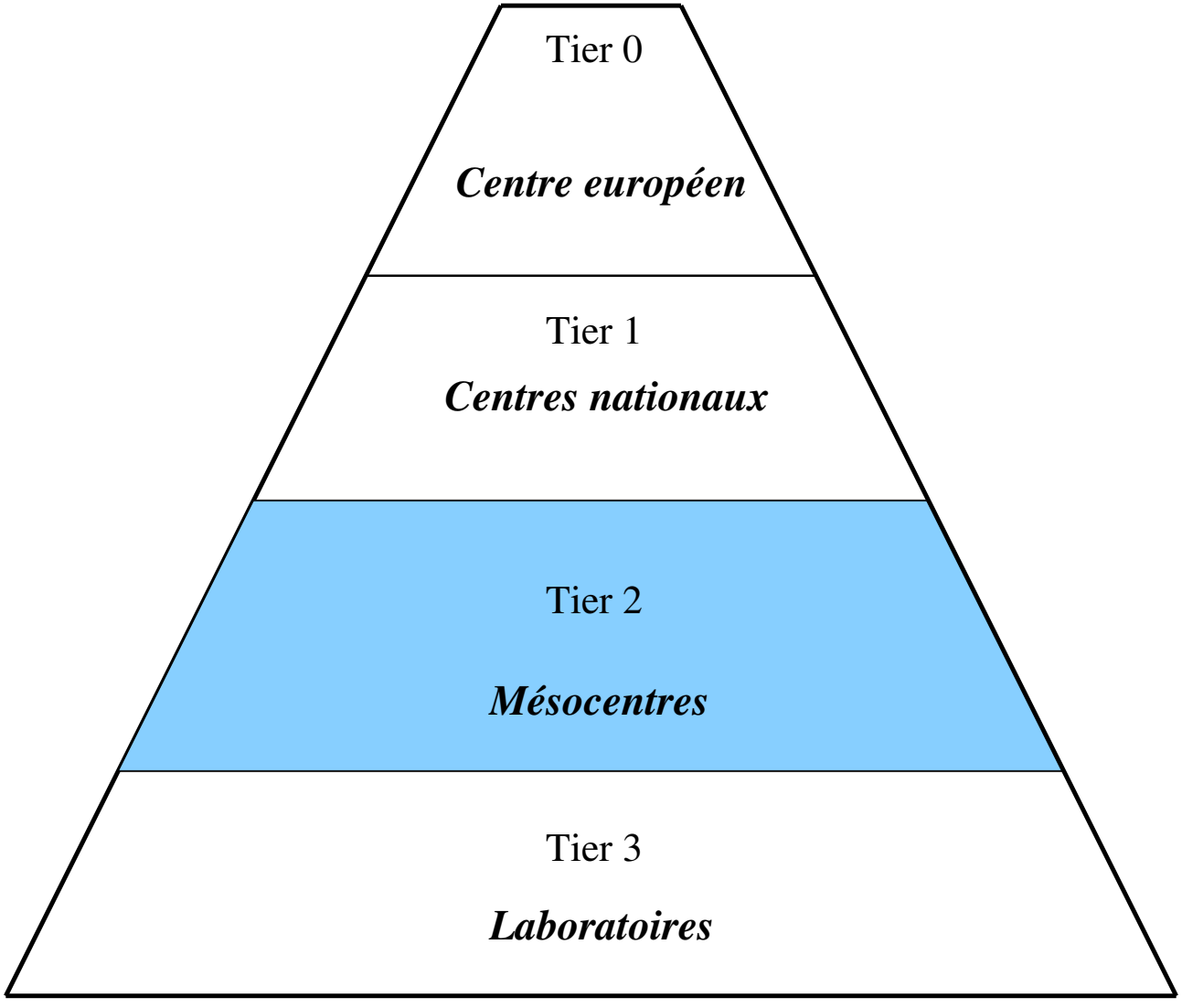
GENCI et l'échelle nationale

GENCI, Grand Equipement National de Calcul Intensif

Société de droit civil détenue à 49% par l'Etat représenté par le Ministère de la Recherche et l'Enseignement Supérieur, 20% par le CEA, 20% par le CNRS , 10% par les Universités et 1% par l'INRIA.

- TGCC/CCRT au CEA
- CINES
- IDRIS

`http://www.genci.fr`


Tier 0
Centre européen
Tier 1
Centres nationaux
Tier 2
Mésocentres
Tier 3
Laboratoires

Les mésocentres et l'échelle régionale

Définition d'un mésocentre

- Un ensemble de moyens humains, de ressources matérielles et logicielles à destination d'une ou plusieurs communautés scientifiques, issus de plusieurs entités (EPST, Universités, Industriels) en général d'une même région, doté de sources de financement propres, destiné à fournir un environnement scientifique et technique propice au calcul haute performance.

FLMSN

- Fédération Lyonnaise de Modélisation et des Sciences Numériques
- Regroupe le P2CHPD (Lyon 1), le PSMN (ENS), le PMS2I (ECL), le Centre Blaise Pascal et l'IXXI.

Sommaire

Les grilles de production et de recherche

France Grille

- Infrastructure de grille de production française
- Contribue à l'infrastructure de grille européenne EGI
- Favorise les échanges entre grilles de production et grilles de recherche

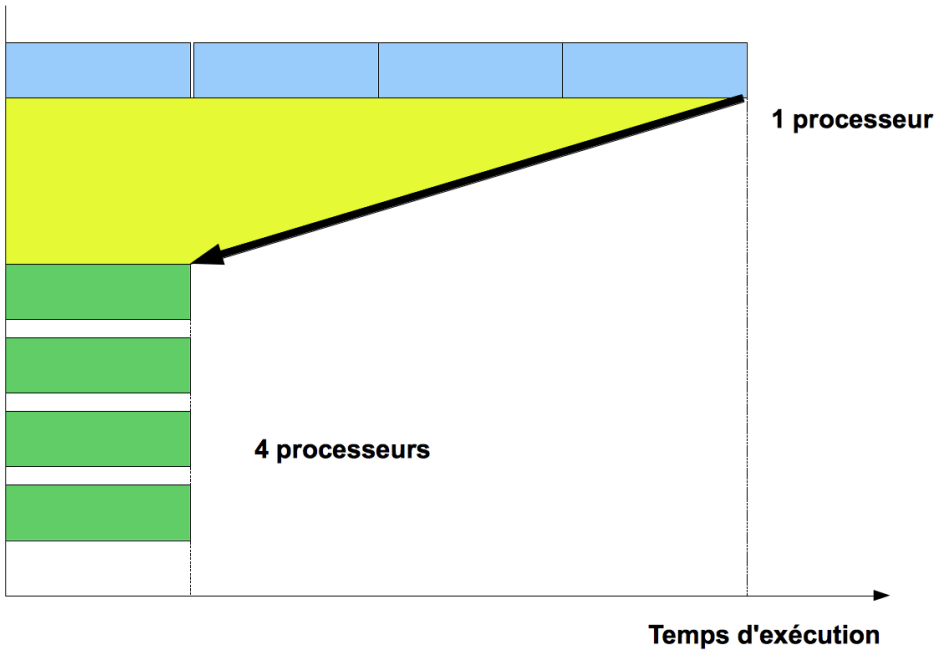
http://www.france-grilles.fr

Sommaire

Sommaire

Introduction

- Exécutions séquentielles des programmes : le processeur n'exécute les instructions que l'une après l'autre, en passant éventuellement d'un processus à un autre via des interruptions en simulant le multitâche.
- Parallélisation : Ensemble de techniques logicielles et matérielles permettant l'exécution simultanée de séquences d'instructions indépendantes, sur des processeurs différents.



Pourquoi paralléliser ?

- Traitement de problèmes scientifiques constituant de grands challenges, tels que : météo et climat, biologie (génome), géophysique (activité sismique), réactions chimique et réactions nucléaires, ...
- Mais aujourd'hui également pour des applications commerciales : Bases de données parallèles, exploration pétrolière, moteurs de recherche web, réalité virtuelle, ...

✓ Traiter des problèmes plus grands et/ou plus complexes.
✓ Mais aussi exploiter les architectures actuelles !

La multiplication du nombre d'unités de calcul ne divise pas spontanément le temps d'exécution des programmes !

Sommaire

Classification de Flynn

Les programmes et les architectures sont classés selon le type d'organisation du flux de données (« data ») et du flux d'instructions.

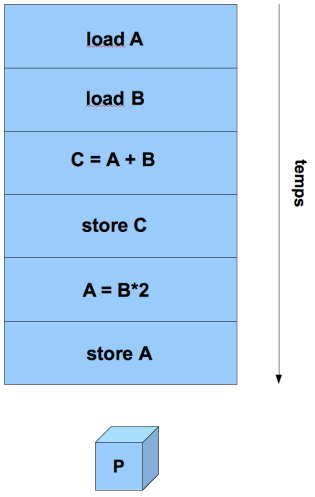
- 2 états possibles pour chacune : « single » ou « multiple »

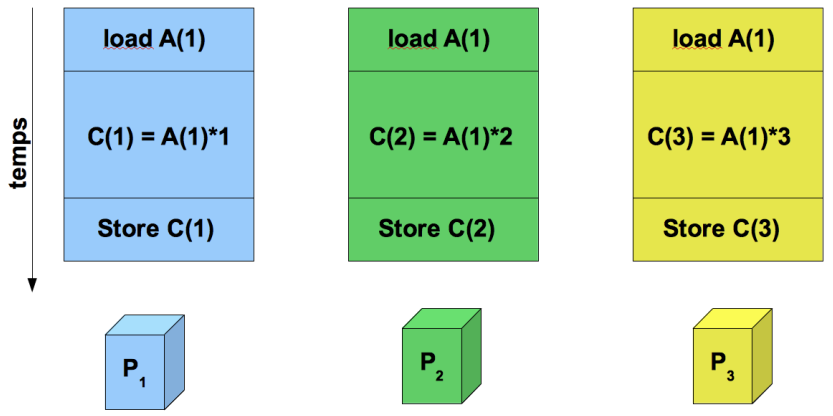
SISD Single Instruction, Single Data $\Longrightarrow$ machines séquentielles	SIMD Single Instruction, Multiple Data $\Longrightarrow$ processeurs vectoriels, GPU
MISD Multiple Instruction, Single Data $\Longrightarrow$ rare	MIMD Multiple Instruction, Multiple Data $\Longrightarrow$ multiproc, multicores

SISD, MISD

SISD architecture séquentielle avec un seul flot d'instructions, un seul flot de données, exécution déterministe.

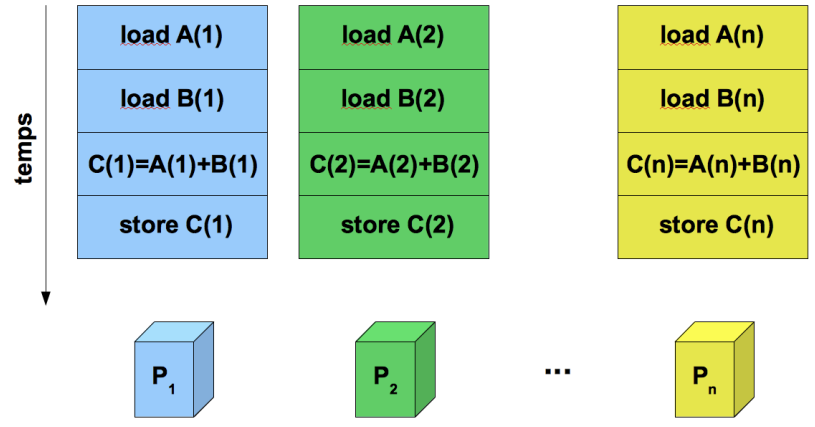
MISD un seul flot de données alimente plusieurs unités de traitement, chaque unité de traitement opère indépendamment des autres, sur des flots d'instructions indépendants.
Peu implémenté.
Ex d'utilisation : exécution en // de plusieurs algos de cryptographie pour le décodage d'1 même message.



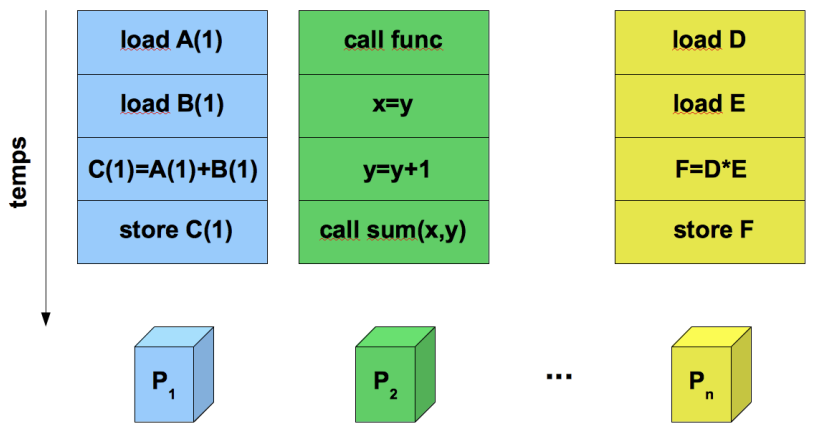


SIMD, MIMD

SIMD architecture // , toutes les unités de traitement exécutent la même instruction à un cycle d'horloge donnée, chaque unité peut opérer sur des données différentes, exécution déterministe.



MIMD architecture la plus courante. Chaque unité de traitement peut gérer un flot d'instructions différent. Chaque unité peut opérer sur un flot de données différent. L'exécution peut être synchrone ou asynchrone, déterministe ou non-déterministe.



Terminologie

- Tâche : une portion de travail à exécuter sur un ordinateur, du type un ensemble d'instructions d'un programme qui est exécuté sur un proc.
- Tâche parallèle : une tâche qui peut s'exécuter sur plusieurs processeurs, sans risque sur la validité des résultats.
- Exécution séquentielle : exécution d'un programme séquentiel, une étape à la fois.
- Exécution parallèle : exécution d'un programme par plusieurs tâches, chaque tâche pouvant exécuter la même instruction ou une instruction différente, à un instant donné.
- Mémoire partagée : D'un point de vue hard, réfère à une machine dont tous les proc. ont un accès direct à une mémoire commune (généralement via un bus).
  D'un point de vue modèle de programmation : toutes les tâches ont la même image mémoire et peuvent directement adresser et accéder au même emplacement mémoire logique, peu importe où il se trouve en mémoire physique.

Terminologie

- Mémoire distribuée : d'un point de vue physique, basée sur un accès mémoire réseau pour une mémoire physique non commune.
  D'un point de vue modèle de programmation, les tâches ne peuvent voir que la mémoire de la machine locale et doivent effectuer des communications pour accéder à la mémoire dune machine distante, sur laquelle d'autres tâches s'exécutent.
- Communications : les tâches parallèles échangent des données, par différents moyens physiques : via un bus à mémoire partagée, via un réseau. Quelque soit la méthode employée, on parle de « communication ».
- Synchronisation : la coordination des tâches en temps réel est souvent associée aux communications, elle est souvent implémentée en introduisant un point de synchronisation au-delà duquel la tâche ne peut continuer tant que une ou plusieurs autres tâches ne l'ont pas atteint.

Terminologie

- Granularité : mesure qualitative du rapport calcul / communications
  - Grain grossier (coarse) : relativement bcp de calculs entre différentes communications.
  - Grain fin (fine) : relativement peu de calcul entre différentes communications.
- speedup : mesure de l'amélioration des performances due au parallélisme.
  (wall clock time of serial execution) / (wall clock time of parallel execution)
- Scalabilité : réfère à la capacité d'un système parallèle à fournir une augmentation de l'accélération proportionelle à l'augmentation du nombre de processeurs.

Sommaire

Parallélisation efficace

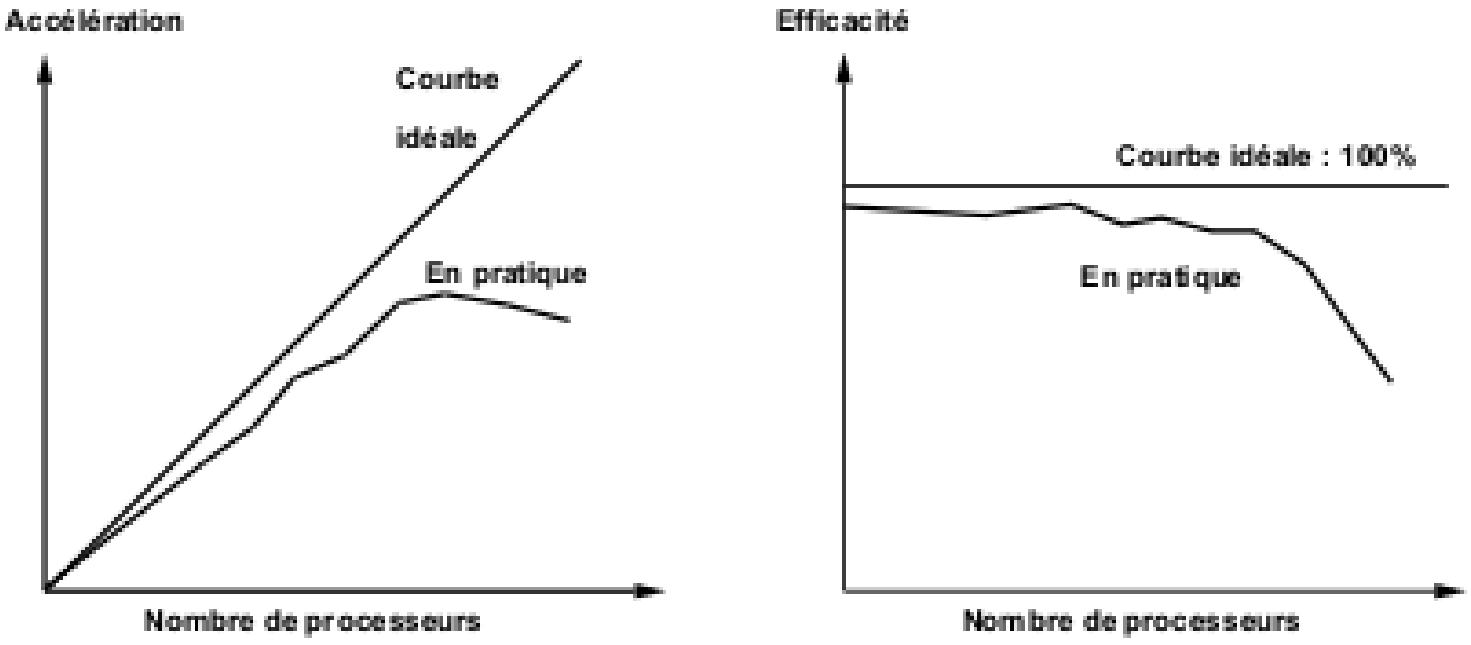
- Accélération et efficacité sont une mesure de la qualité de la parallélisation.
- Soit $T(p)$ le temps d'exécution sur p processeurs.
- L'accélération $A(p)$ et l'efficacité $E(p)$ sont définies comme étant :

$$\begin{aligned} A(p) &= T(1)/T(p) \ \ (p = 1, 2, 3...) \\ E(p) &= A(p)/p \end{aligned}$$

- Pour une accélération parallèle parfaite, on obtient :

$$\begin{aligned} T(p) &= T(1)/p \\ A(p) &= T(1)/T(p) = T(1)/(T(1)/p) = p \\ E(p) &= A(p)/p = p/p = 100\% \end{aligned}$$

Accélération et efficacité



- Les programmes scalables demeurent efficaces pour un grand nombre de processeurs (scalables = passages à l'échelle).

Scalabilité, loi d'Amdahl

$\Longrightarrow$ Propriété d'une application à être exécutée efficacement sur un très grand nombre de processeurs.

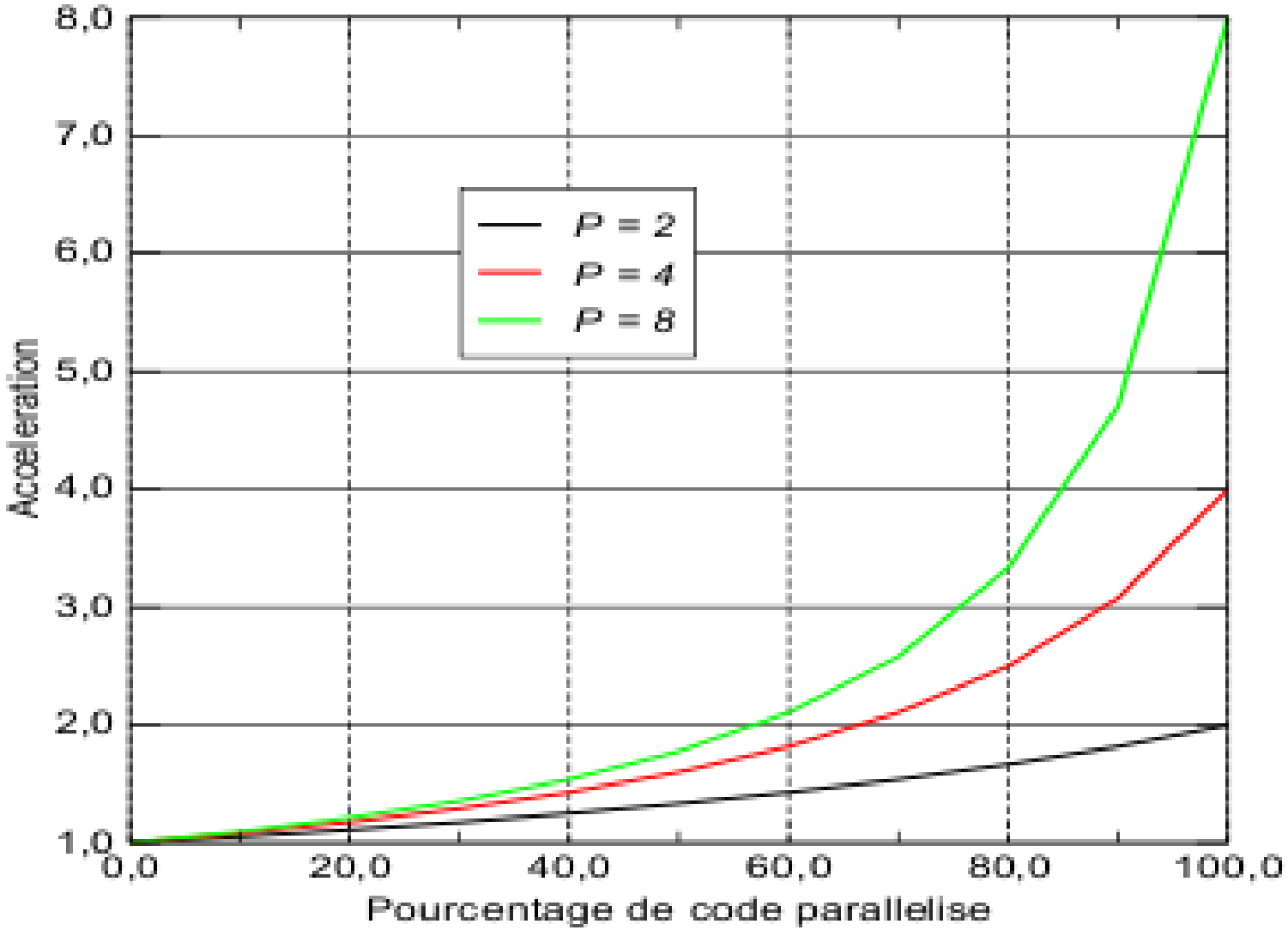
$\Longrightarrow$ Raison possibles d'une faible scalabilité :

- La machine parallèle employée : architecture inadaptée, charge ...
- Le programme parallélisé : analyser le comportement, améliorer les performances.
- un peu des deux ...

Loi d'Amdahl

- Plus une règle qu'un loi.
- Principe : **la partie non parallèle d'un programme limite les performances et fixe une borne supérieure à la scalabilité.**

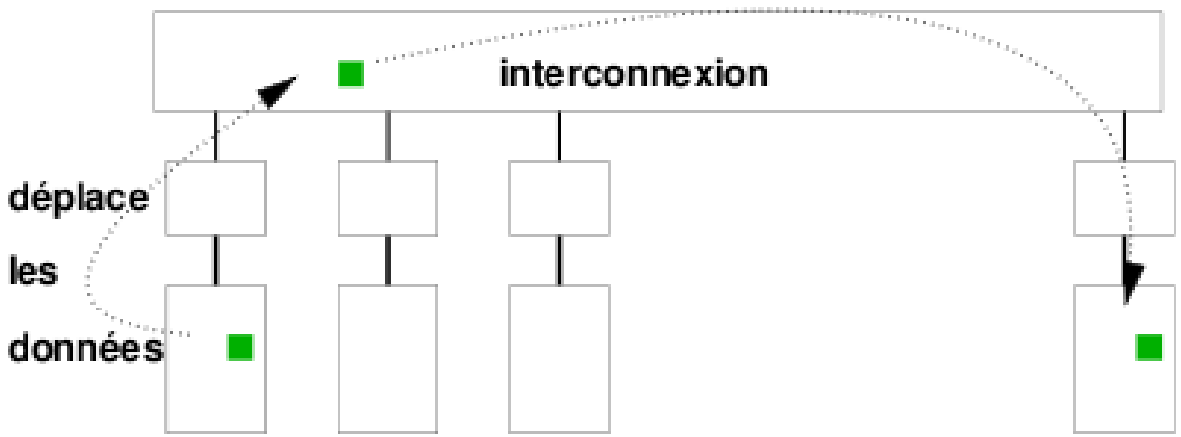
Loi d'Amdahl


8,0
7,0
6,0
5,0
4,0
3,0
2,0
1,0
Accelération
P = 2
P = 4
P = 8
0,0
20,0
40,0
60,0
80,0
100,0
Pourcentage de code parallelise

Sommaire

Mémoire Partagée

- Tous les processeurs accèdent à toute la mémoire globale avec un même espace d'adressage global.
- Chaque processeur travaille indépendamment des autres, mais les modifications effectuées par un processeur à un emplacement mémoire (en mémoire globale) sont visibles par tous les autres.
- Les processeurs ont leur propre mémoire locale (cache).



Mémoire Partagée

- UMA, Uniform Memory Access :
  - Des processeurs identiques, ayant tous le même temps d'accès à la mémoire. Plus couramment appelées SMP (Symmetric MultiProcessor).
  - Gestion de la cohérence des caches. Certains systèmes sont CC-NUMA : si un processeur modifie un emplacement mémoire partagé, la modification est visible par tous les autres processeurs.
- NUMA (Non Uniform Memory Access) :
  - Conçue pour pallier aux problèmes d'accès mémoire concurrents via un unique bus.
  - En général fabriquées à base de plusieurs blocs SMP interconnectés.
  - Des temps d'accès à la mémoire diffèrent suivant la zone accédée. Le temps d'accès via le lien d'interconnexion des blocs est plus lent.
  - Gestion de la cohérence des caches. CC-NUMA : Cache Coherent NUMA.

Mémoire Partagée

- Espace d'adresse globale $\longrightarrow$ facilite le travail du programmeur.
- Mémoire proche des CPUs $\longrightarrow$ le partage des données entre tâches est rapide.

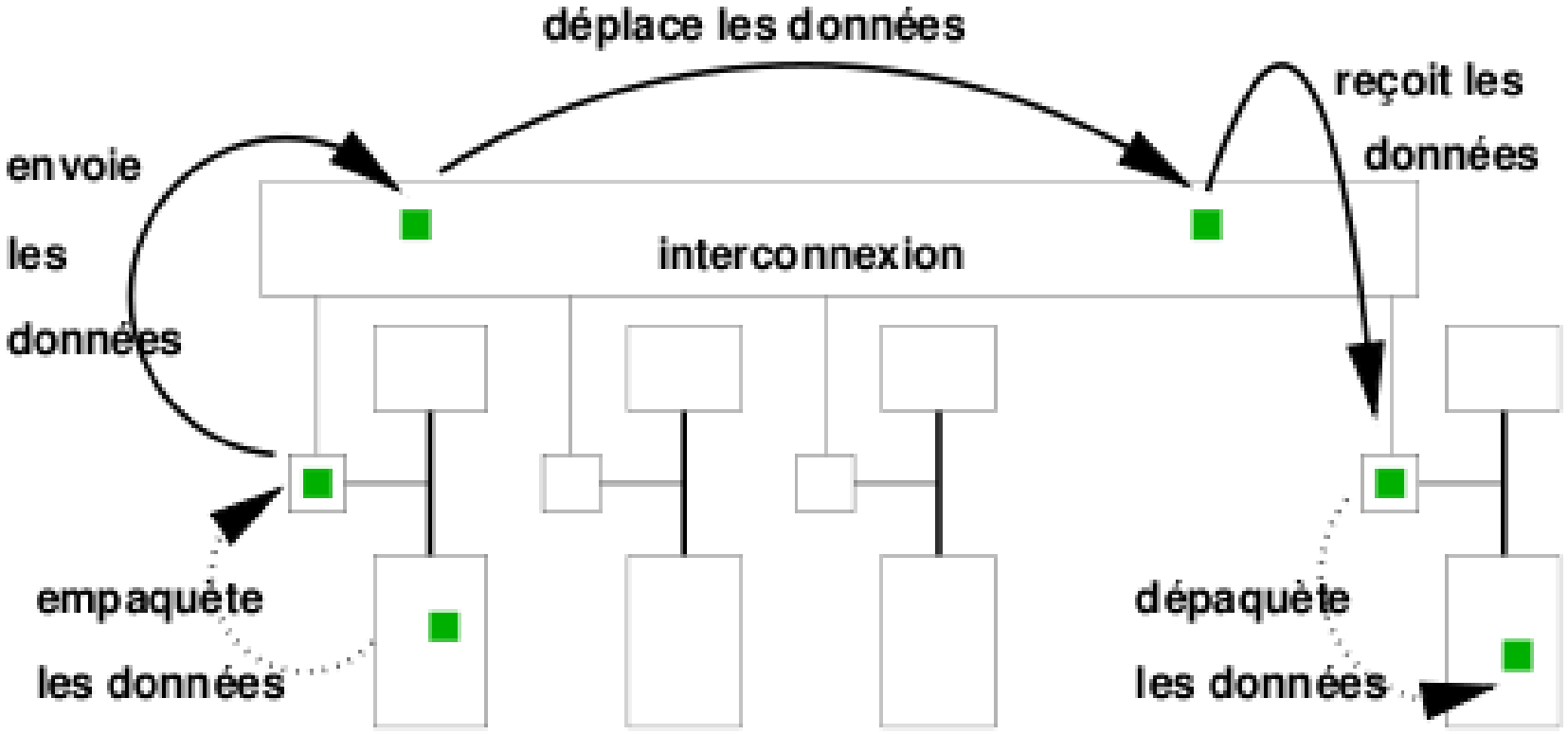
Mais :

- Manque de scalabilité : augmenter le nombre de CPUs accroît le trafic sur le chemin d'accès à la mémoire partagée.
- Le programmeur doit gérer la synchronisation pour un accès correct à la mémoire globale.
- Très coûteux de construire des architectures à mémoire partagée avec un grand nombre de CPUs.

Mémoire Distribuée

- Le point commun de ces architectures : elles possèdent un réseau d'interconnexion pour communiquer entre les mémoires des différents processeurs.
- Chaque processeur a sa propre mémoire locale, il n'y a pas de notion d'espace d'adressage global entre tous les procs.
- Les procs opèrent indépendamment les uns des autres, une modification en mémoire locale n'a pas d'effet sur la mémoire des autres procs.
- Si un processeur a besoin d'une donnée dans la mémoire d'un autre processeur, le programmeur doit définir explicitement la communication.
- Les réseaux d'interconnexion sont divers, avec des niveaux de performance très variables.

Mémoire Distribuée



Mémoire Distribuée

- On peut augmenter le nombre de processeurs et la mémoire proportionnellement.
- Accès rapide à la mémoire locale sur chaque proc, sans surcoût de gestion de cohérence de cache.
- Coût raisonnable, ex : PCs en réseau.

Mais :

- Le programmeur doit gérer toutes les communications entre processeurs.
- Peut être difficile de faire coïncider une structure de données basée sur une mémoire globale, à cette organisation physique de la mémoire.
- Temps d'accès mémoire non locaux élevés.

Architecture mémoire hybride

- La plupart des calculateurs actuels sont aujourd'hui un mixte de mémoire partagée et mémoire distribuée.
- La brique de base (nœud) est un multiprocesseur à mémoire partagée.
- Ces briques sont interconnectées par un réseau (type ethernet, Infiniband, ...).
- A noter qu'on trouve aussi de l'hybride entre CPU et cartes accélératrices.

Processus / Thread

Processus

- Un processus a son propre espace mémoire, sa propre pile qu'il ne partage pas avec les autres
- les processus sont plus lourds que les threads à créer : en général, 100 fois plus rapides à créer qu'un processus
- les processus sont réellement indépendants

Processus léger / Thread

- Les threads partagent le même espace mémoire
- les threads peuvent se partager des informations facilement
- les threads doivent eux-mêmes faire attention à ne pas se marcher sur les pieds : il n'y a pas de protection entre les threads d'un même processus
- Les threads ont peu d'informations propres

Sommaire

Conclusions

Fréquence d'horloge, nombre de cœurs, nombre d'unités fonctionnelles, fréquence et capacité mémoire, caches, stockage, interconnexions internes et réseaux ...
Tous les éléments ont leur importance ; l'architecture doit être équilibrée pour que l'ensemble soit performant

Au niveau technologique

- Beaucoup de cœurs
- Des cœurs hybrides CPU / cartes accélératrices
- Une différence entre la vitesse de calcul et les transferts mémoire et I/O

Conclusions

Au niveau applicatif

- Nécessité d'une bonne connaissance des architectures, notamment au niveau de la mémoire.
- Plus de croissance des performances au niveau du cœur $\Longrightarrow$ parallélisation obligatoire pour un gain de performance.
- Nécessité d'élaborer des algorithmes et des programmes capables d'exploiter un grand nombre de processeurs.
- Nécessité d'exploiter le parallélisme aux différents niveaux du hardware.
- Programmation sur des architectures hybrides avec différents types de processeurs.