

Introduction au parallélisme

Cécile Germain-Renaud
cecile.germain@lri.fr

Organisation

- 12 séances cours et TD, 4h chacune
- 3 séances par Joël Falcou
- Les slides ne suffisent vraiment pas
 - Aide mémoire + illustrations
 - Le cours oral est supposé connu pour l'examen
- Calendrier et toutes informations sur le site du cours
 - vous devez le consulter.

www.lri.fr/~cecile/ENSEIGNEMENT/IPAR

Introduction au parallélisme

2

Contrôle des connaissances

- Contrôle continu
 - Deux devoirs - largement commencés en TD. A rendre à la date prévue. 1 jour de retard = -1 point. Devoirs individuels.
 - Un exposé. Choix dans 6 semaines, exposés + fiche résumé à la dernière séance. Travail en binôme, l'un est noté sur la fiche de lecture, l'autre sur l'exposé.
- Examen
 - Tous documents autorisés. Vous devez amener les documents, énoncés,... utilisés en cours/TD.

www.lri.fr/~cecile/ENSEIGNEMENT/IPAR

Introduction au parallélisme

3

Plan

1. Motivation et concepts
2. Algorithmique parallèle
3. Programmation parallèle
 - Passage de messages
 - OpenMP
4. Modèles formels
 - Parallélisation automatique
 - Modèles avancés

Introduction au parallélisme 4

1. Motivation et concepts

Introduction au parallélisme 5

Plan

- Motivation
- Architectures parallèles
 - Les infrastructures parallèles en 2009
 - Typologie
 - Exemples
- Performance
 - Accélération
 - Autres indicateurs

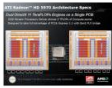
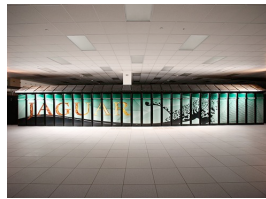
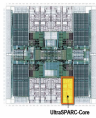
Introduction au parallélisme 6

Motivation I : Les applications

- *Simulation numérique : expériences in silico - Troisième composante de la science*
 - Trop grandes : météorologie, matériaux,...
 - Trop dangereuses : maintenance nucléaire
 - Trop coûteuses : crash-tests, conception aéronautique
 - Impossibles : climat, astrophysique
- *Information deluge*
 - Puces à ADN
 - Traitement personnalisé
- *Réalité virtuelle ou augmentée* : médias, médical

Besoin en puissance de calcul (computing) virtuellement illimités

Motivation II : Les ordinateurs



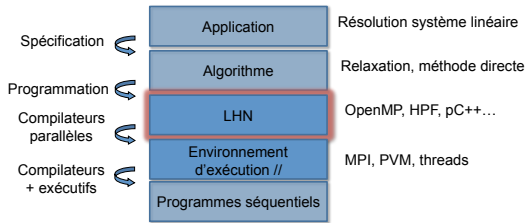
Toutes les infrastructures de calcul sont parallèles

Motivation III : Les sources du parallélisme

- Parallélisme de données
 - Dès qu'on a des tableaux (à préciser)
- Parallélisme de contrôle
 - High Throughput applications
 - Procédures récursives
 - RPC
 - ...

Les applications peuvent exploiter les infrastructures parallèles

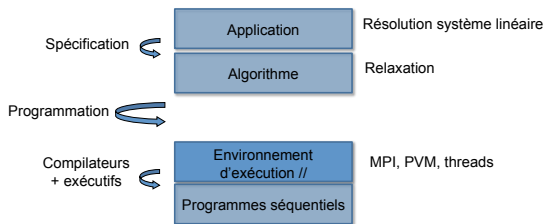
Un environnement logiciel stabilisé mais non unifié



Introduction au parallélisme

10

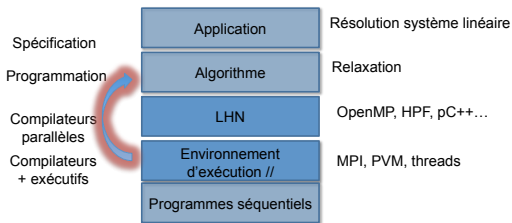
Un environnement logiciel stabilisé mais non unifié



Introduction au parallélisme

11

Un environnement logiciel stabilisé mais non unifié



Introduction au parallélisme

12

Le Top 500

- Les 500 ordinateurs les plus puissants
 - Volontariat
 - 2 fois par an depuis 1993.
- Evalué sur Linpack, un benchmark facile et non trivial
- Performance R_{Max} mesurée en FLOPS
- A comparer avec R_{Peak} , la puissance indépassable.



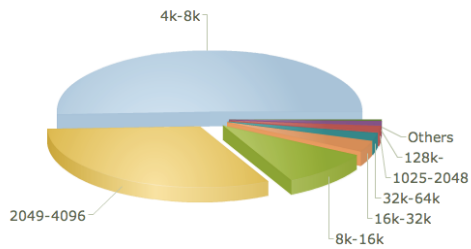
www.top500.org

La barrière du PetaFlops a été franchie en 2008

Introduction au parallélisme

13

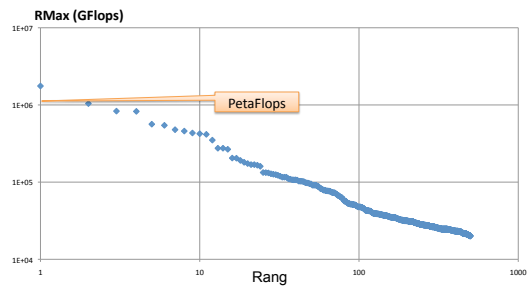
TOP 500 : Nombre de processeurs / système Novembre 2009



Introduction au parallélisme

14

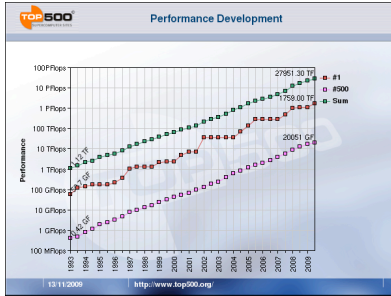
Le Top 500 en Novembre 2009



Introduction au parallélisme

15

Evolution



Introduction au parallélisme

16

Linpack est-il un benchmark pertinent ?

(...) the LINPACK Benchmark is to solve a dense system of linear equations (...)

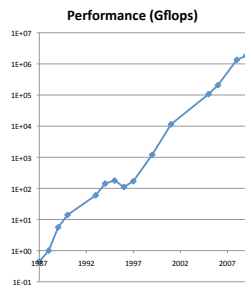
In an attempt to obtain uniformity across all computers in performance reporting, the algorithm used in solving the system of equations in the benchmark procedure **must conform to LU factorization with partial pivoting**. In particular, the operation count for the algorithm must be $\frac{2}{3}n^3 + O(n^2)$ double precision floating point operations. This excludes the use of a **fast matrix multiply algorithm** like "Strassen's Method" or algorithms which compute a solution in a precision lower than full precision (64 bit floating point arithmetic) and refine the solution using an iterative approach.

Introduction au parallélisme

17

Gordon Bell Prize: "peak performance"

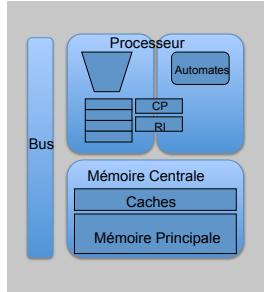
Année	Performance
1987	450 Mflops
1988	1 Gflops
1990	14 Gflops
1996	111 Gflops
1999	1.2 Tflops
2001	11.4 Tflops
2005	107 Tflops
2006	207 Tflops
2008	1.35 Pflops
2009	1.84 Pflops



18

Processeur séquentiel

- Partie opérative
- Partie contrôle
- Hiérarchie mémoire
- Bus (au pluriel)



Introduction au parallélisme

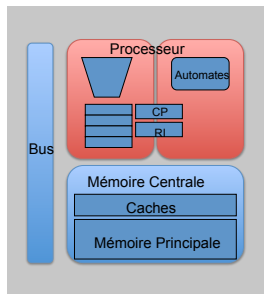
19

La classification de Flynn

Degré de réplication de la partie opérative et de la partie contrôle

Instructions

Données	SISD	MISD
	SIMD	MIMD



Introduction au parallélisme

20

MIMD non vectoriel

Exploite les acquis des architectures de microprocesseurs: Hiérarchie mémoire, OS



Introduction au parallélisme

21

Modèles de programmation : Dijkstra/Flynn

Parallélisme de contrôle

Composition parallèle de processus séquentiels

PAR(SEQ) – MIMD

```
parfor i = 1, n
  a(i) = b(i) + c(i)
  x(i) = y(i) + z(i)
end do
```

Parallélisme de données

Composition séquentielle de processus parallèles

SEQ(PAR) - SIMD

```
forall i = 1, n
  a(i) = b(i) + c(i)
forall i = 1, n
  x(i) = y(i) + z(i)
```

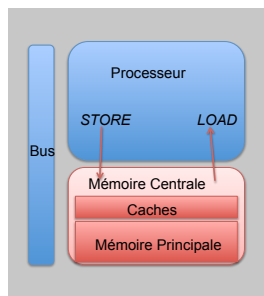
Et beaucoup d'autres modèles

Introduction au parallélisme

22

Organisation et accès mémoire

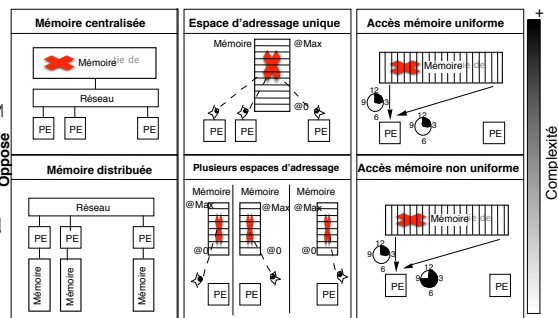
- Situation de la mémoire/ des caches par rapport aux processeurs : Centralisée ou Distribuée
- Organisation de l'espace d'adressage: Unique ou Plusieurs
- Hiérarchie des temps d'accès mémoire: Uniforme ou non uniforme



Introduction au parallélisme

23

Organisation mémoire



24

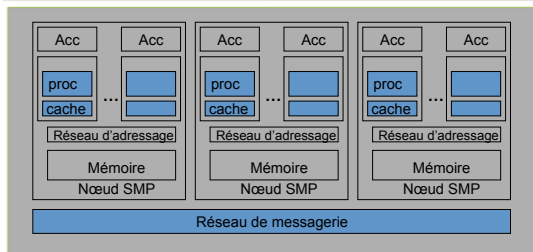
Modèles de programmation/exécution

- Espaces d'adressage multiples
 - p processus
 - Communiquent par messagerie
 - Le programmeur ou le compilateur ou l'exécutif doit définir le placement des données et l'ordonnancement des calculs
 - Le placement des données peut définir celui des calculs : ex. OCR
- Espace d'adressage unique
 - p threads
 - Communiquent à travers des variables partagées
 - Le programmeur ou le compilateur ou l'exécutif doit définir le placement et l'ordonnancement des calculs

Introduction au parallélisme

25

Architectures actuelles



Introduction au parallélisme

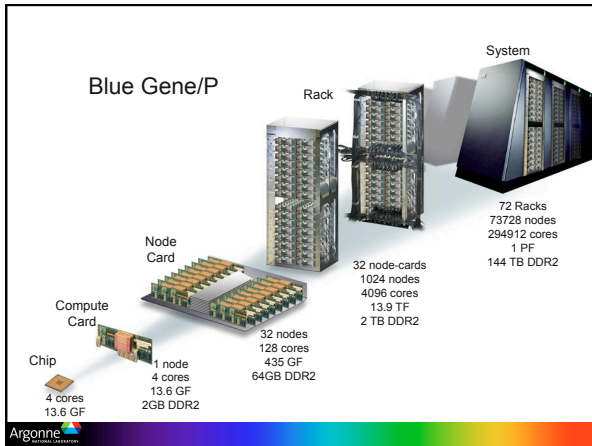
26

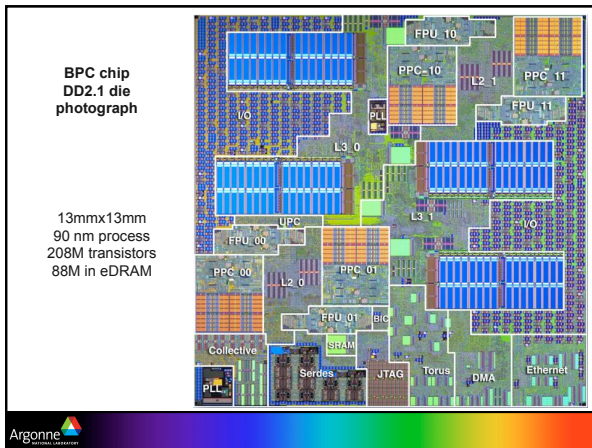
Terminologie

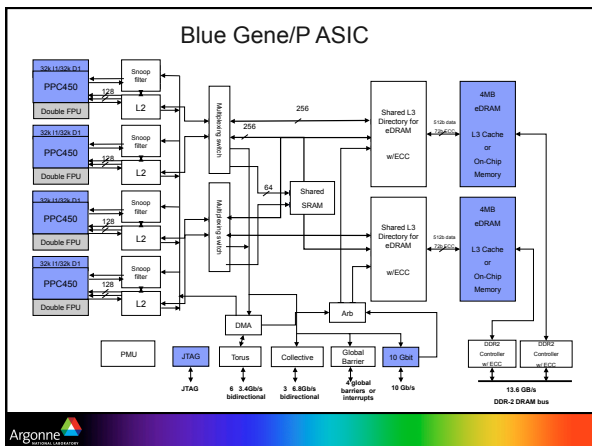
- MPP : Massively Parallel Multiprocessor
- Clusters
- La principale différence est dans le degré réseau (au sens large) non propriétaire pour les clusters

Introduction au parallélisme

27








Blue Gene/P Interconnection Networks


- 3 Dimensional Torus**
 - Interconnects all compute nodes
 - Virtual cut-through hardware routing
 - 3.4 Gb/s on all 12 node links (5.1 GB/s per node)
 - 0.5 μ s latency between nearest neighbors, 5 μ s to the farthest
 - MPI: 3 μ s latency for one hop, 10 μ s to the farthest
 - Communications backbone for point-to-point
- Collective Network**
 - One-to-all broadcast functionality
 - Reduction operations for integers and doubles
 - 6.8 Gb/s of bandwidth per link per direction
 - Latency of one way tree traversal 1.3 μ s, MPI 5 μ s
 - Interconnects all compute nodes and I/O nodes
- Low Latency Global Barrier and Interrupt**
 - Latency of one way to reach 72K nodes 0.65 μ s, MPI 1.6 μ s

Combin
d'opérations
flottantes
par
seconde ?



Blue Gene/P Architectural Highlights

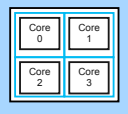
- Scaled performance relative to BG/L through density and frequency
 - 1.2x from frequency bump 700 MHz => 850 MHz
 - 2x performance through doubling the processors/node
- Enhanced function
 - 4 way SMP, cache coherent, supports threads, OpenMP
 - Improved memory subsystem supporting higher bandwidths
 - DMA for torus, remote put-get, user programmable memory prefetch
 - Memory chip kill implemented.
 - Enhanced performance counters (including 450 core)
 - Architectures of double Hummer FPU, torus, collective network, barrier, and JTAG networks were left intact.
- Higher signaling rate
 - 2.4x higher bandwidth, lower latency for Torus and Tree networks
 - 10x higher bandwidth for Ethernet IO



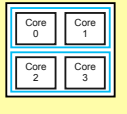
Execution Modes in BG/P

Hardware Elements Black
Software Abstractions Blue

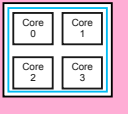
Quad Mode (VNM)
4 Processes
1 Thread/Process




Dual Mode
2 Processes
1-2 Threads/Process



SMP Mode
1 Process
1-4 Threads/Process





RoadRunner

Introduction au parallélisme 34

Accélération

$$Acceleration = \frac{Performance(native)}{Performance(améliorée)}$$

Introduction au parallélisme 35

Accélération et efficacité parallèles

$T(p)$: temps d'exécution à p processeurs

- $S(p) = p$
Application *parfaitement parallélisable*
Parallélisme trivial
- $S(p) = kp$
 $k < 1$ en général,
mais $k > 1$ est possible

$$S(p) = \frac{T(1)}{T(p)}$$

$$E(p) = \frac{S(p)}{p}$$

Introduction au parallélisme 36

La loi d'Amdhal

L'application comporte

- Une fraction séquentielle f : contrôle, calculs scalaires répliqués, ...
- Une fraction parfaitement parallélisable $1-f$

$$T(p) = f + \frac{1-f}{p} \quad T(1) = f + (1-f) = 1$$

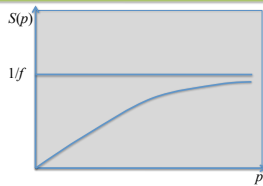
$$S(p) < \frac{1}{f}$$

Inutile d'ajouter des processeurs au delà

Introduction au parallélisme

37

La loi d'Amdhal



$$S(p) < \frac{1}{f}$$

Inutile d'ajouter des processeurs au delà

Introduction au parallélisme

38

Optimiser pour d'autres critères

- L'hypothèse implicite est que le problème est de taille fixe
- Les problèmes de taille fixe existent : traitement d'images
- « Améliorer le traitement des problèmes ordinaires »
- Mais d'autres contraintes peuvent être plus pertinentes par rapport à la demande de l'utilisateur : résoudre un problème, effectuer une simulation etc.
 - En temps constant
 - En encombrement mémoire constant

Introduction au parallélisme

39

La loi de Gustafson

L'application comporte

- Une partie séquentielle a : contrôle, calculs scalaires répliqués, ...
- Une partie parfaitement parallélisable b
- L'application est extensible

$$T(p) = a + \frac{b}{p} = 1 \qquad T(1) = a + b$$

$$S(p) = a + (1 - a)p$$

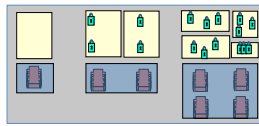
Accélération affine

Introduction au parallélisme

40

Autres facteurs pénalisants pour l'accélération

- Le parallélisme introduit des surcoûts
 - Communications (EAM) ou synchronisations (EAU)
- Matériel
 - Le coût des communications ou des synchronisations augmente avec p
- Algorithme
 - Déséquilibre de charge
 - Augmentation du nombre de communications ou de synchronisations par unité de travail



Introduction au parallélisme

41

Autres facteurs pénalisants pour l'accélération

- Le parallélisme introduit des surcoûts
 - Communications (EAM) ou synchronisations (EAU)
- Matériel
 - Le coût des communications ou des synchronisations augmente avec p
- Algorithme
 - Déséquilibre de charge
 - Augmentation du nombre de communications ou de synchronisations par unité de travail

La parallélisation triviale est presque toujours la meilleure solution lorsqu'elle est possible
Exception : Changements de niveau dans la hiérarchie mémoire. Out-of-Core -> MP, ou MP -> cache

Introduction au parallélisme

42

Critères économiques

- Iso-efficacité : Taille nécessaire pour atteindre une efficacité donnée
 - Peut-on atteindre une certaine efficacité sur un système donné ?
 - De combien faut-il augmenter la taille du problème ?
 - Existe-t-il une taille maximale du système telle qu'on ne puisse plus atteindre l'efficacité requise ?
 - Comparaison des applications : la meilleure iso-efficacité est la plus petite
- Exemple canonique : $N_{1/2}$

Introduction au parallélisme

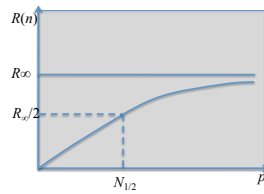
43

Débit

$$R(n,p) = \frac{\text{Nombre_Operations}(n)}{T(n,p)}$$

$$R_\infty(p) = \lim_{n \rightarrow \infty} R(n,p)$$

$$R(n_{1/2}, p) = \frac{R_\infty(p)}{2}$$



Introduction au parallélisme

44
