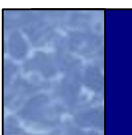
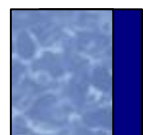


Systemes, algorithmes et applications: Efficacité et utilité des systemes paralleles.

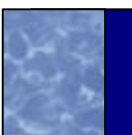
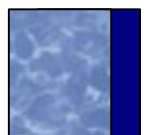
Gaétan Hains
LACL, Université de Paris-Est

Séminaire au LRDE, EPITA, Sept 2008.



Plan de l'exposé

- La mesure des systèmes parallèles:
pas que des Mega-Flops ...
- Le choix des algorithmes: densité et
régularité.
- Des applications:
problèmes faciles / difficiles à paralléliser.



Le modèle BSP:

une vue unifiée archi. + appli.

Systeme parallele $\cong$ *réseau de p ordinateurs*

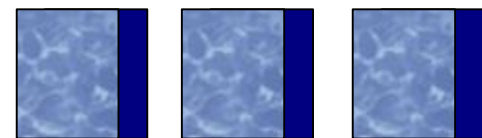
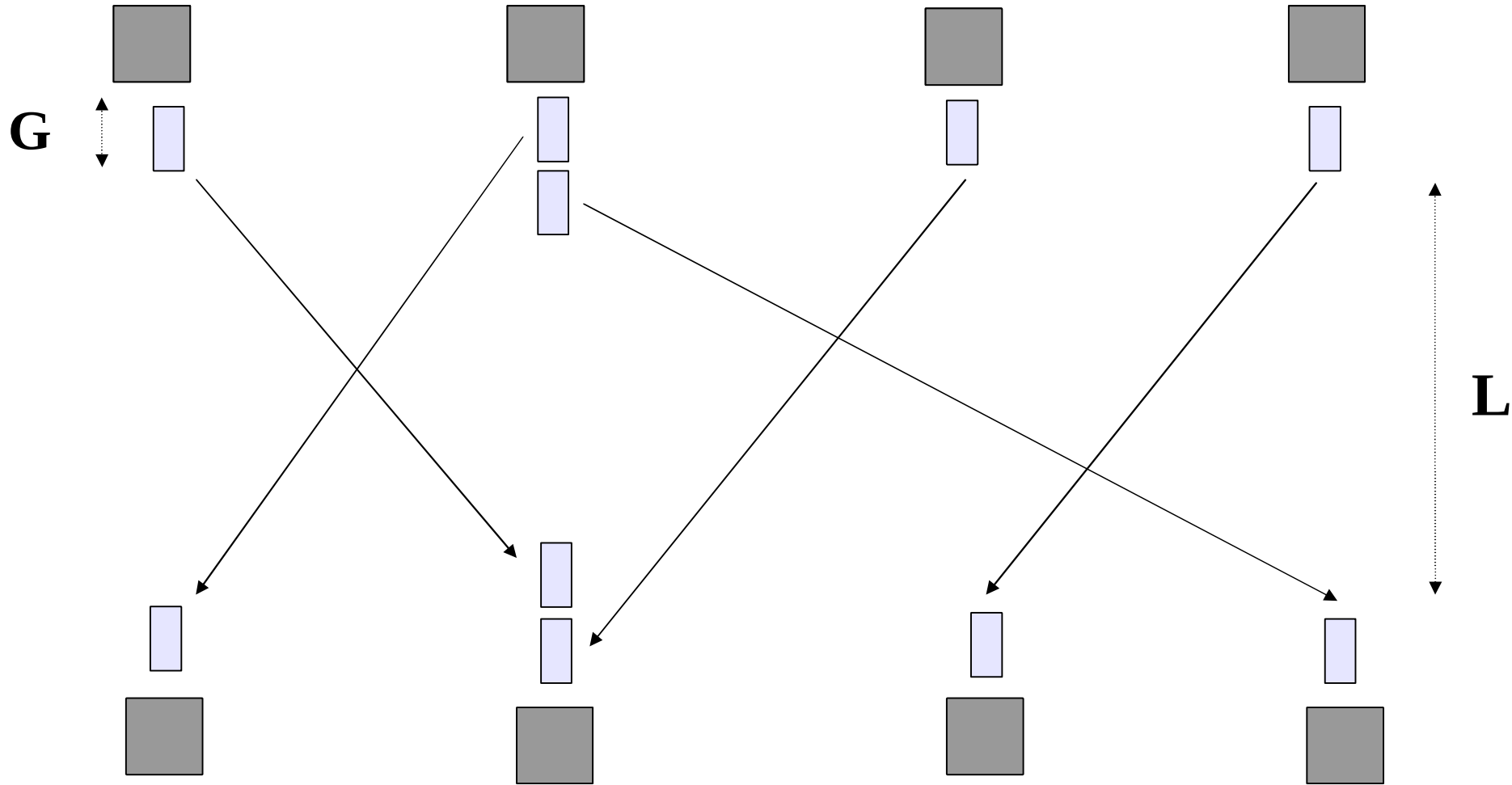
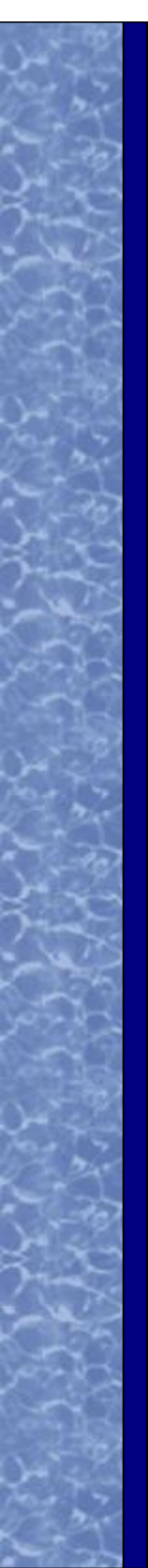
*Calcul asynchrone parallele. (s : MFlops)
 $\times p$*

Echange « permutation ». g : Mmots/s

Echange « tous vers tous »: $p \times$ plus lent

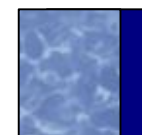
Synchronisation globale. L : MSynch./s





G et L relatifs (en Flops)

Machine	P	S	L	G
T3E	16	512	1745	4
Origin2000	4	90	2900	12
Origin2000	8	90	4700	13
Origin2000	16	90	12500	26
Sun bi-proc.	2	29	610	3
PentiumIII bi-pro.	2	41	961	2
Cluster PentiumIII	5	88	57000	50



G et L absolus en μs *et* $\mu s/mot$

Machine	P	S	L	G
T3E	16	512	3	0,01
Origin2000	4	90	30	0,12
Origin2000	8	90	54	0,14
Origin2000	16	90	132	0,27
Sun bi-proc.	2	29	21	0,10
PentiumIII bi-pro.	2	41	23	0,06
Cluster PentiumII	5	88	644	0,57



Les algorithmes

- **Algèbre linéaire matrices pleines:**
efficacité OK, équilibrage OK.
- **Matrices creuses:**
efficacité OK, équilibrage difficile.
- **« SGBD »: tri, sélection, jointure:**
efficacité OK, équilibrage difficile.
- **Séquentiel pur + mémoire parallèle:**
1 processeur + $(p-1)$ RAM via réseau.



Des applications

- **SGBD**: jointure en parallèle.
- **Langue naturelle**: DyALog.
- **Résolveurs numériques**: PIN.
- **Data-mining**: programmation génétique.



Jointure en parallèle (SGBD)

- Problème $O(N^2)$, N = taille des relations.
- Efficacité parallèle élevée.
- Problème: déséquilibre « imprévisible ».
- Solution: **algorithme préemptif**,
 - Surcoût $\ll$ déséquilibre.
 - Fujitsu AP3000 $p=12$, *speedup* = 8.

Bamha, Hains (LIFO)



Traitement de la langue naturelle

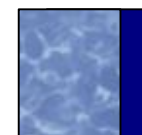
- DyALog = analyseur syntaxique généralisé
- Grammaires peu ambiguës: *speedup* ≤ 5 .
- ***Grammaires très ambiguës:***
 - ***Cray T3E, $p \geq 64$: speedup = 25***
 - ***Cluster PC, $p \geq 2$: ralentissement.***

Debourges, Guilloiré, DeLaClergerie, Hains (INRIA-LIFO)



Résolveur parallèle pour domaines numériques

- INPUT: variables, intervalles, contraintes algébriques.
- ***OUTPUT: intervalles maximalement réduits.***
- ***Algorithme centralisé:***
 - ***Cray T3E, $p \geq 16$: speedup = 5.***
 - ***Cray T3E, système creux: speedup < 2.***
 - ***Cluster PC, $p \geq 2$: ralentissement.***



Data-mining par programmation génétique.

- Evolution de requêtes relationnelles. Très coûteux.
- Algorithme parallèle asynchrone: mauvaise convergence.
- ***Algorithme parallèle « cosmopolite »:***
 - ***Bonne convergence.***
 - ***Bi-processeur: speedup = 1,8***



Conclusions

- Problèmes denses: relativement faciles.
- Problèmes peu denses ou irréguliers:
 - Bons algorithmes + bon réseau
- NE PAS: mesurer Mflops / \$
- MAIS: **(MFlops,g,L) / \$.**
- Cluster de PC: tout dépend du réseau !

