

Systemes, algorithmes et applications:  
Efficacité et utilité des systèmes parallèles.

Gaétan Hains  
LIFO, Université d'Orléans

Séminaire au LESI, ESPEO, Mai 2000.



# 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.

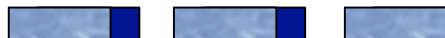
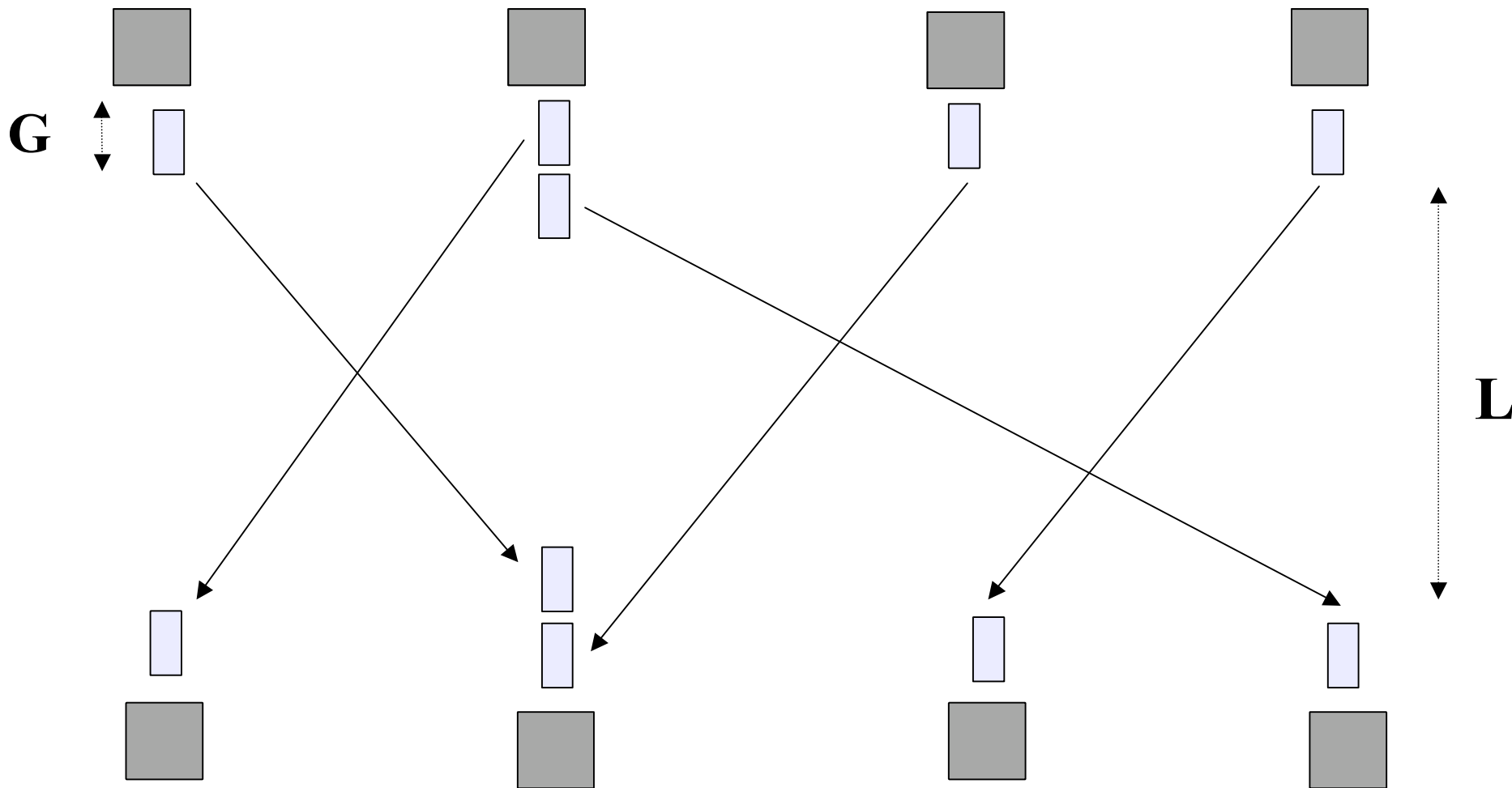
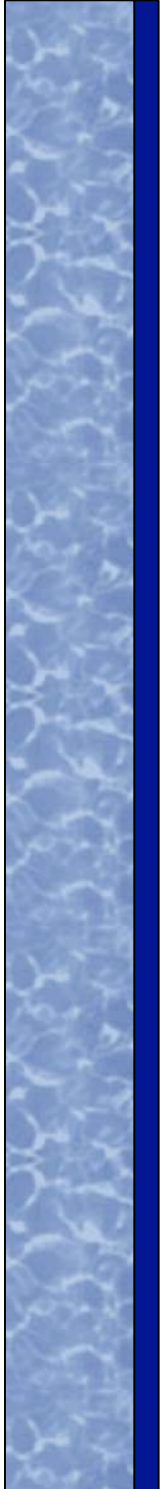
Systeme parallele  $\cong$  *réseau de  $p$  ordinateurs*

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

*Echange « permutation ».  $g$ : Mmots/s*

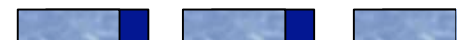
*Echange « tous vers tous »:  $p \times$  plus lent*

*Synchronisation globale.  $L$ : MSynch./s*



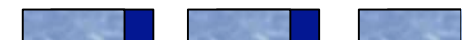
# G et L relatifs (en Flops)

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



# G et L absolus en $\mu s$ 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	<b>132</b>	<b>0,27</b>
Sun bi-proc.	2	29	21	0,10
PentiumIII bi-pro.	2	41	23	0,06
Cluster PentiumII	5	88	<b>644</b>	<b>0,57</b>

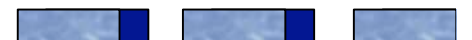


# Les algorithmes

- ' **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 pr`cesseur +  $(p-1)$  RAM via réseau.

# Des applications

- **SGBD:** j`inture en parallèle.
- **Langue naturelle:** DyAL`g.
- **Résolveurs numériques:** PIN.
- **Data-mining:** pr`grammati`n  
génétiq.

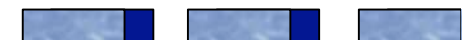




# Jointure en parallèle (SGBD)

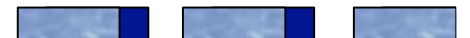
- Pr`blème  $O(N^2)$ ,  $N$ = taille des relations.
- Efficacité parallèle élevée.
- Pr`blème: déséquilibre « imprévisible ».
- S`luti`n: **algorithme préemptif**,
  - Surc`ût << déséquilibre.

Bamha, Hains (LIFO) Fujitsu AP3000  $p=12$ , *speedup* = 8.



# Traitement de la langue naturelle

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



# Résolveur parallèle pour domaines numériques

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

# *Data-mining* par programmation génétique.

- Ev`luti`n de requêtes relatives. Très coûteux.
- Alg`rithme parallèle asynchr`ne: mauvaise c`nvergence.
- ***Algorithme parallèle « cosmopolite »:***
  - ***Bonne convergence.***
  - ***Bi-processeur: speedup = 1,8***



# Conclusions

- Pr`blèmes denses: relativement faciles.
- Pr`blèmes peu denses `u irréguliers:
  - B`ns alg`rithmes + b`n réseau
- NE PAS: mesurer Mfl`ps / \$
- MAIS: (MFlops,g,L) / \$.**
- Cluster de PC: t`ut dépend du réseau !