

Évaluation d'un algorithme de redistribution optimal pour applications itératives : la propagation de la chaleur



HÉLÈNE RENARD
ABDOU GUERMOUCHE

Séminaire High Performance Computing
30 juin 2009

Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- Le contexte
- Algorithmes de redistribution

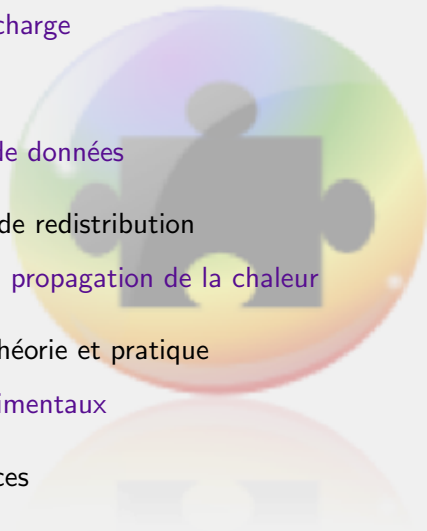
3. Un exemple : la propagation de la chaleur

- Description
- Liens entre théorie et pratique

4. Résultats expérimentaux

- Le contexte
- Les expériences

5. Conclusion



Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- Le contexte
- Algorithmes de redistribution

3. Un exemple : la propagation de la chaleur

- Description
- Liens entre théorie et pratique

4. Résultats expérimentaux

- Le contexte
- Les expériences

5. Conclusion



Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- Le contexte
- Algorithmes de redistribution

3. Un exemple : la propagation de la chaleur

- Description
- Liens entre théorie et pratique

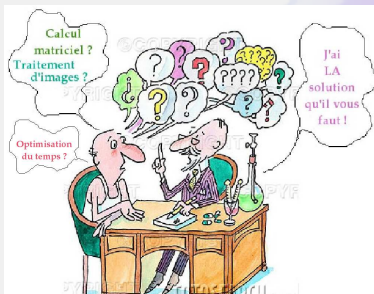
4. Résultats expérimentaux

- Le contexte
- Les expériences

5. Conclusion



Le contexte



Plates-formes ciblées : plates-formes distribuées hétérogènes (réseau de stations de travail, clusters de clusters, grilles, etc.).

Équilibrage de charge :

- ★ Statique.

Redistribution de données :

1. Différentes variations : ressources ou besoins de l'application.
2. Les données doivent être redistribuées afin d'obtenir un meilleur équilibrage de charge.

Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- **Un exemple**

2. Redistribution de données

- Le contexte
- Algorithmes de redistribution

3. Un exemple : la propagation de la chaleur

- Description
- Liens entre théorie et pratique

4. Résultats expérimentaux

- Le contexte
- Les expériences

5. Conclusion



La théorie en images

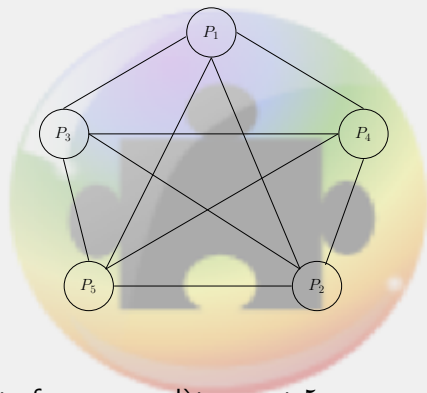
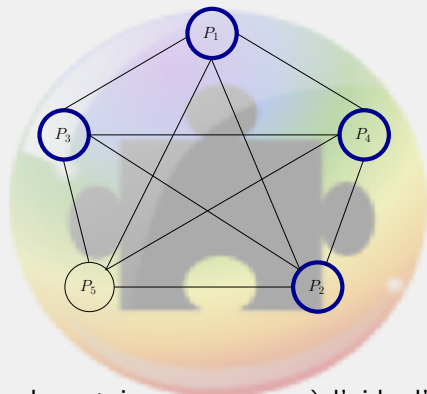


Plate-forme complète ayant 5 processeurs

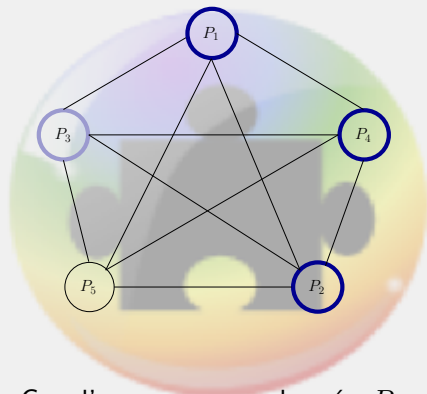
La théorie en images



Sélection de certains processeurs à l'aide d'un algo

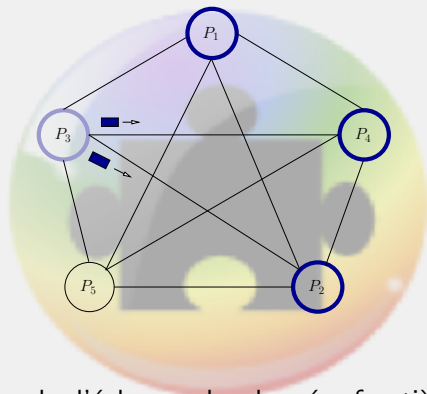
Prise en compte la contention des liens de communications
⇒ Apparition des problèmes NP-complets

La théorie en images



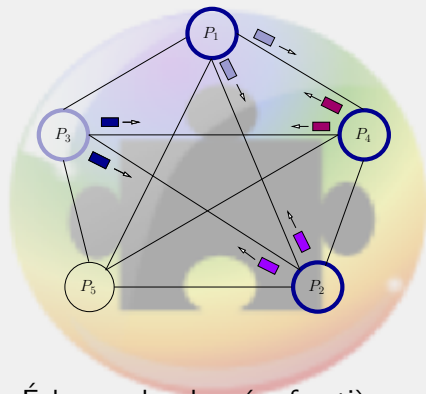
Cas d'un processeur donné : P_3

La théorie en images



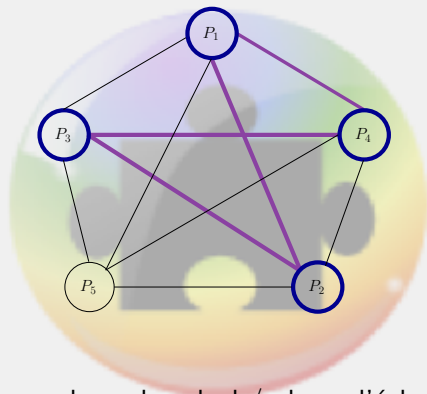
Exemple d'échange des données frontières

La théorie en images



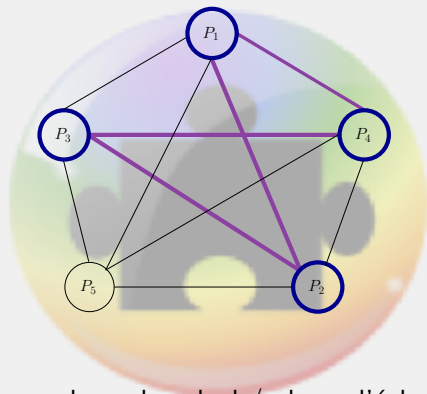
Échange des données frontières

La théorie en images



Répéter « phase de calcul / phase d'échange »

La théorie en images



Répéter « phase de calcul / phase d'échange »

Une distribution uni-dimensionnelle des données induit un arrangement uni-dimensionnel des processeurs.

Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- Le contexte
- Algorithmes de redistribution

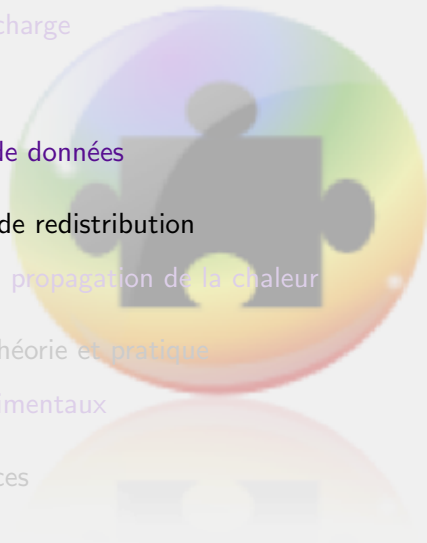
3. Un exemple : la propagation de la chaleur

- Description
- Liens entre théorie et pratique

4. Résultats expérimentaux

- Le contexte
- Les expériences

5. Conclusion



Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- **Le contexte**
- Algorithmes de redistribution

3. Un exemple : la propagation de la chaleur

- Description
- Liens entre théorie et pratique

4. Résultats expérimentaux

- Le contexte
- Les expériences

5. Conclusion



Le contexte



Plates-formes ciblées : plates-formes distribuées hétérogènes (réseau de stations de travail, clusters de clusters, grilles, etc.)

1. Différentes variations : ressources ou besoins de l'application.
2. Les données doivent être redistribuées afin d'obtenir un meilleur équilibrage de charge.
3. Pas de discussion du mécanisme d'équilibrage de charge : nous le considérons comme extérieur.

Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- Le contexte
- **Algorithmes de redistribution**

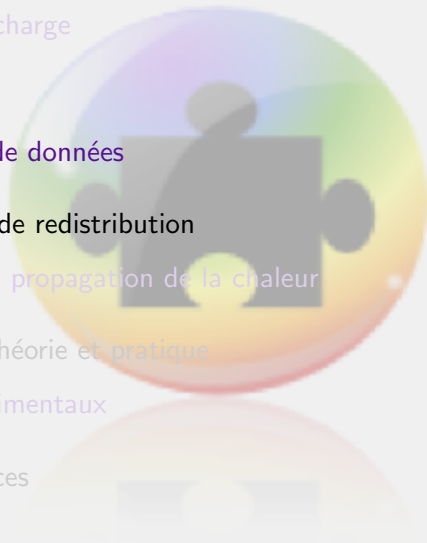
3. Un exemple : la propagation de la chaleur

- Description
- Liens entre théorie et pratique

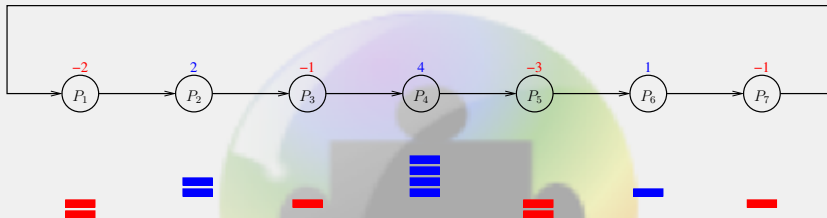
4. Résultats expérimentaux

- Le contexte
- Les expériences

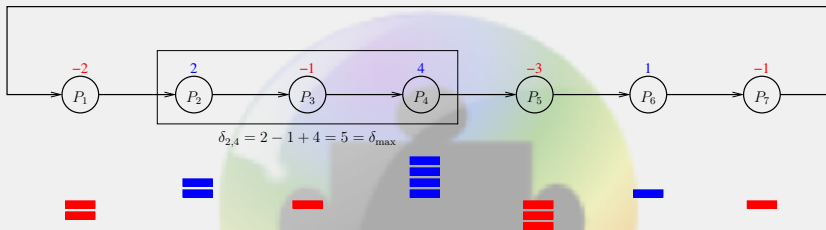
5. Conclusion



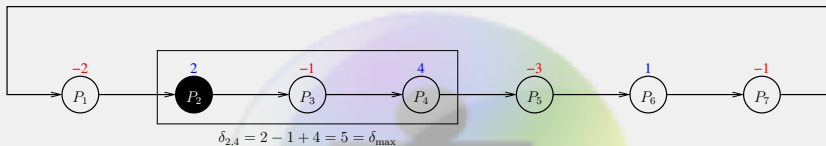
Anneau homogène unidirectionnel : en images !



Anneau homogène unidirectionnel : en images !



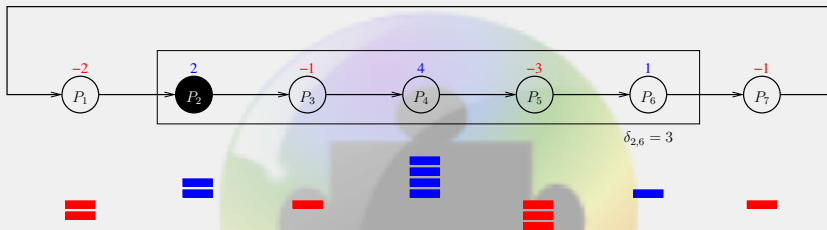
Anneau homogène unidirectionnel : en images !



$$\delta_{\max} = 5$$

L'algorithme complet de redistribution est défini par le premier processeur de déséquilibre maximal.

Anneau homogène unidirectionnel : en images !

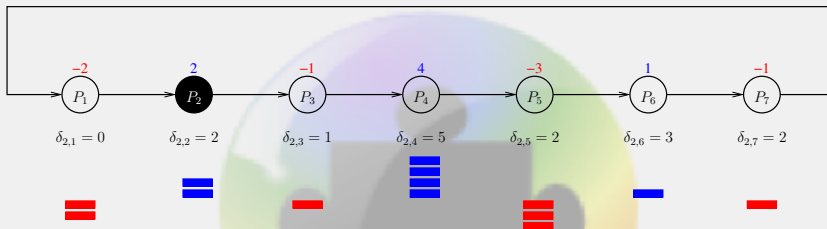


$$\delta_{\max} = 5$$

L'algorithme complet de redistribution est défini par le premier processeur de déséquilibre maximal.

Pendant l'exécution de l'algorithme, le processeur P_i envoie $\delta_{2,i}$ données.

Anneau homogène unidirectionnel : en images !

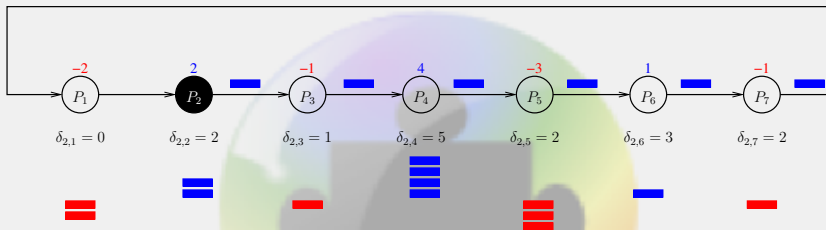


$$\delta_{\max} = 5$$

L'algorithme complet de redistribution est défini par le premier processeur de déséquilibre maximal.

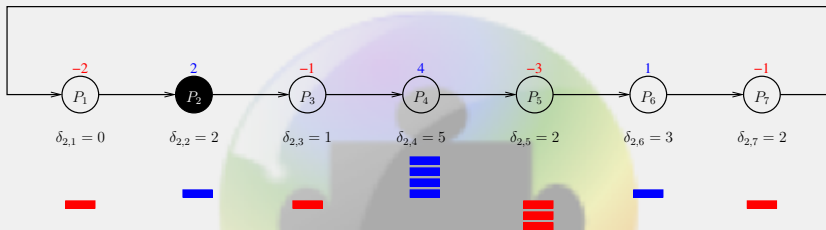
Pendant l'exécution de l'algorithme, le processeur P_i envoie $\delta_{2,i}$ données.

Anneau homogène unidirectionnel : en images !



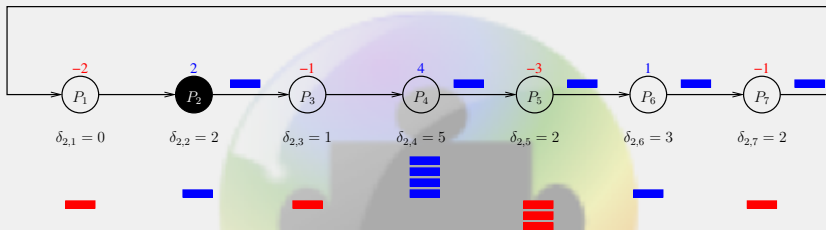
À l'étape 1, P_i envoie une donnée si et seulement si $\delta_{2,i} \geq 1$

Anneau homogène unidirectionnel : en images !



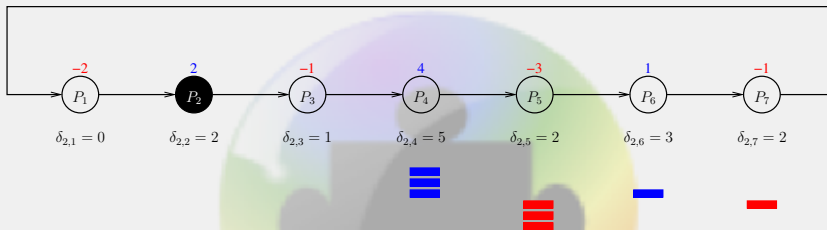
À l'étape 1, P_i envoie une donnée si et seulement si $\delta_{2,i} \geq 1$

Anneau homogène unidirectionnel : en images !



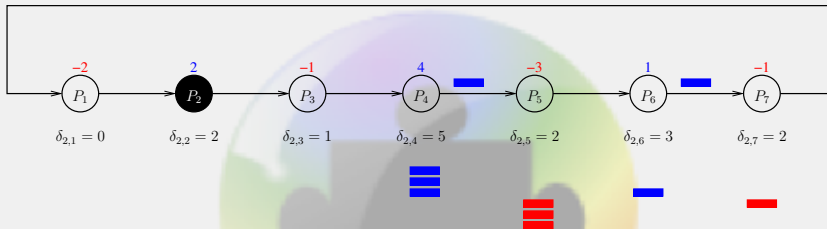
À l'étape 2, P_i envoie une donnée si et seulement si $\delta_{2,i} \geq 2$

Anneau homogène unidirectionnel : en images !



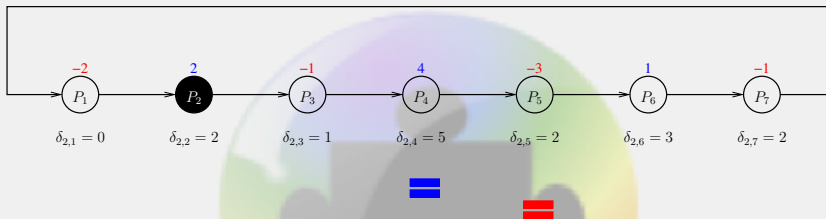
À l'étape 2, P_i envoie une donnée si et seulement si $\delta_{2,i} \geq 2$

Anneau homogène unidirectionnel : en images !



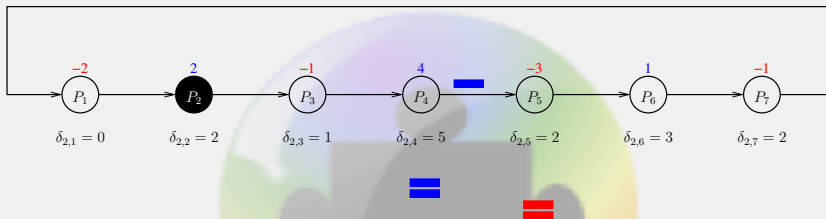
À l'étape 3, P_i envoie une donnée si et seulement si $\delta_{2,i} \geq 3$

Anneau homogène unidirectionnel : en images !



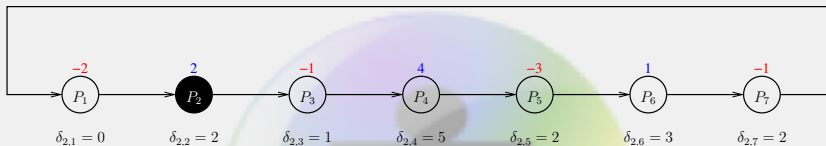
À l'étape 3, P_i envoie une donnée si et seulement si $\delta_{2,i} \geq 3$

Anneau homogène unidirectionnel : en images !



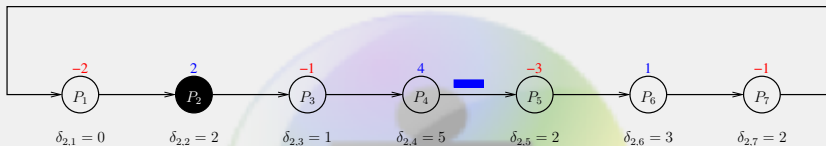
À l'étape 4, P_i envoie une donnée si et seulement si $\delta_{2,i} \geq 4$

Anneau homogène unidirectionnel : en images !



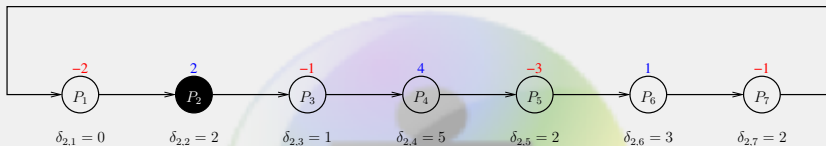
À l'étape 4, P_i envoie une donnée si et seulement si $\delta_{2,i} \geq 4$

Anneau homogène unidirectionnel : en images !



À l'étape 5, P_i envoie une donnée si et seulement si $\delta_{2,i} \geq 5$

Anneau homogène unidirectionnel : en images !



À l'étape 5, P_i envoie une donnée si et seulement si $\delta_{2,i} \geq 5$

De manière générale

- ★ Algorithme de redistribution optimal pour les anneaux :
 - ▷ homogènes unidirectionnels,
 - ▷ hétérogènes unidirectionnels,
 - ▷ homogènes bidirectionnels,
 - ▷ hétérogènes bidirectionnels avec l'hypothèse de redistribution légère.

Définition

Une redistribution est légère si chaque processeur possède initialement toutes les données qu'il devra envoyer pendant l'exécution de l'algorithme.

Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- Le contexte
- Algorithmes de redistribution

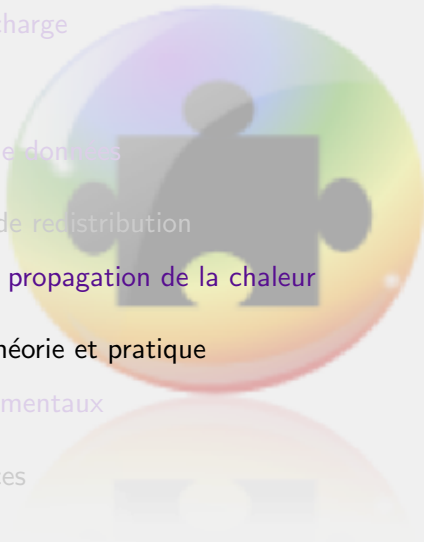
3. Un exemple : la propagation de la chaleur

- Description
- Liens entre théorie et pratique

4. Résultats expérimentaux

- Le contexte
- Les expériences

5. Conclusion



Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- Le contexte
- Algorithmes de redistribution

3. Un exemple : la propagation de la chaleur

- **Description**
- Liens entre théorie et pratique

4. Résultats expérimentaux

- Le contexte
- Les expériences

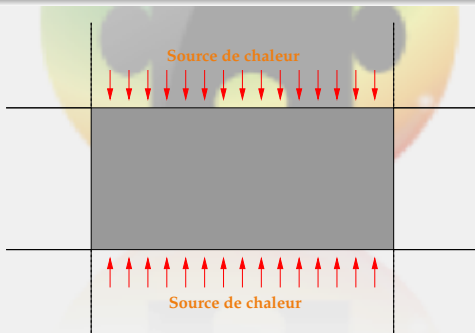
5. Conclusion



Équation de Laplace

Hypothèse

Imaginons une plaque métallique à laquelle on applique une source de chaleur aux bords. La chaleur va se répandre à l'intérieur de la plaque. La température aux bords est maintenue constante, la distribution de chaleur dans la plaque tend vers un état stationnaire.

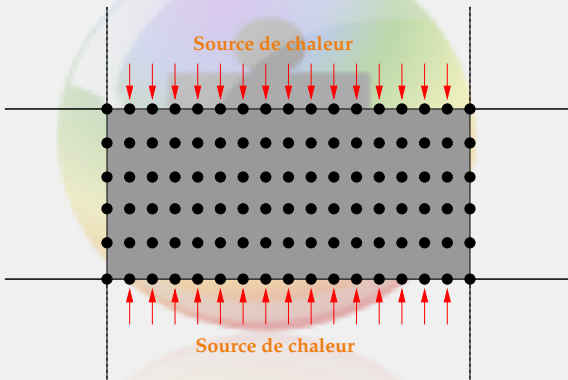


$$\text{Équation de Laplace : } \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$$

Équation de Laplace : $\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$

Résolution :

1. Approximation de la solution \Rightarrow on discrétise \Rightarrow grille à n^2 points



2. En utilisant les différences finies sur l'équation de Laplace, cela revient à résoudre itérativement l'équation suivante :

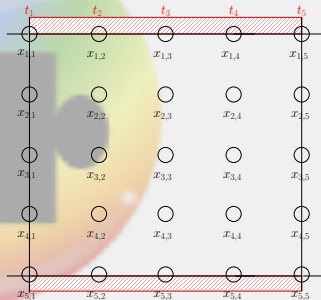
$$4x_{i,j} - (x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}) = 0$$

Équation de Laplace : exemple

4. Enrichissement de la matrice : le vecteur b est nul sauf pour

- ▷ les points sur les bords inférieur et supérieur (source de chaleur)
- ▷ les points voisins des points source de chaleur

- $x_{1,1} = t_1, x_{1,2} = t_2, x_{1,3} = t_3,$
- $x_{1,4} = t_4, x_{1,5} = t_5$
- $x_{5,1} = t_6, x_{5,2} = t_7, x_{5,3} = t_8,$
- $x_{5,4} = t_9, x_{5,5} = t_{10}$



$$\begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & & \ddots & & & & \vdots \\ 0 & \cdots & -1 & -1 & 4 & -1 & -1 & \cdots \\ \vdots & & & & \ddots & & & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ \vdots \\ x_{5,4} \\ x_{5,5} \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ 0 \\ \vdots \\ 0 \\ t_9 \\ t_{10} \end{pmatrix}$$

Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- Le contexte
- Algorithmes de redistribution

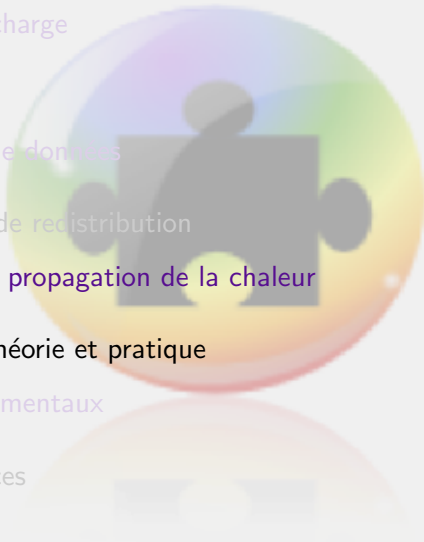
3. Un exemple : la propagation de la chaleur

- Description
- **Liens entre théorie et pratique**

4. Résultats expérimentaux

- Le contexte
- Les expériences

5. Conclusion



Ordonnancement & chaleur ?

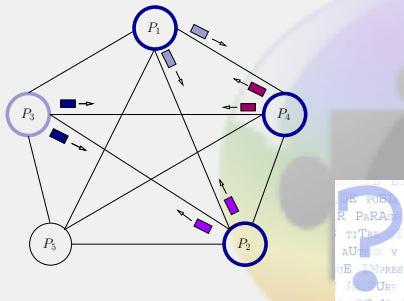


FIG.: Équilibrage de charge
Redistribution de données

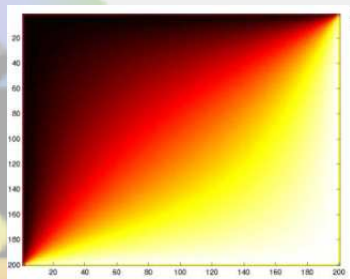


FIG.: Propagation de la chaleur :
 - température nulle sur les bords supérieur et gauche
 - température constante positive sur les deux autres bords

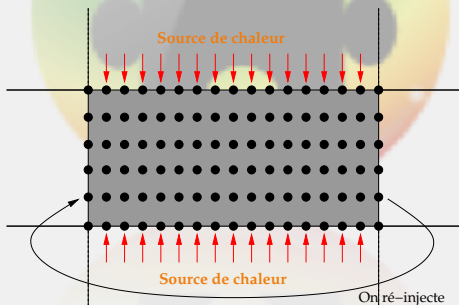
Histoire d'anneau

Hypothèse

Supposons que nous voulions étudier :

- ★ l'écoulement de la chaleur, mais sur un plateau de taille $1m \times 1km$

Trop volumineux ! \Rightarrow Nous « découpons » le plateau :



Nous obtenons alors des communications sous forme d'anneau

Nous répètons cette opération pour aboutir au total de $1km$

Convergence

Question : Comment est-ce que nous savons que nous convergions ?

Connaissance de tous les $x_{i,j}$ et calcul d'un résidu pour savoir quand nous nous arrêtons.

Calcul du résidu : $\|Ax^{t+1} - b\| < \epsilon$

Exemple : (sur 100 itérations)

- ★ Test de la convergence toutes les 20 itérations (trop coûteux à chaque étape au niveau communication)
- ★ Redistribution de données toutes les 20 itérations, en même temps que le test de convergence

Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- Le contexte
- Algorithmes de redistribution

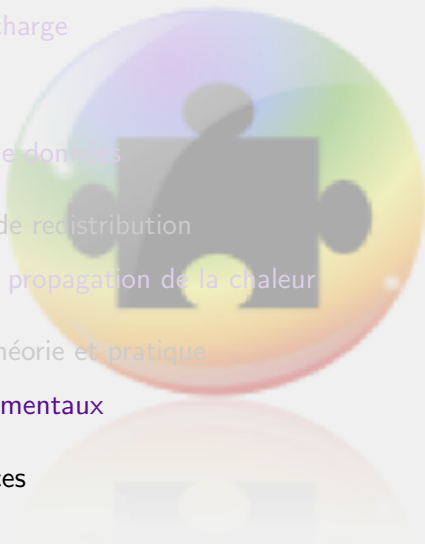
3. Un exemple : la propagation de la chaleur

- Description
- Liens entre théorie et pratique

4. Résultats expérimentaux

- Le contexte
- Les expériences

5. Conclusion



Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- Le contexte
- Algorithmes de redistribution

3. Un exemple : la propagation de la chaleur

- Description
- Liens entre théorie et pratique

4. Résultats expérimentaux

- **Le contexte**
- Les expériences

5. Conclusion



Grid'5000 vs SimGrid

But : Comparer le comportement des algorithmes d'équilibrage de charge et redistribution de données sur deux « plate-formes » différentes

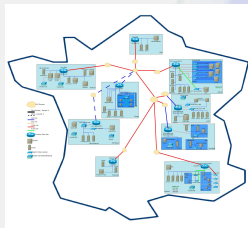


FIG.: Grid 5000



FIG.: SimGrid

Grid'5000 : 9 sites distincts de 5000 unités de calcul \Rightarrow restreint à 22 nœuds distribués sur 4 sites

SimGrid : autant de nœuds que nous voulons

Grid'5000 vs SimGrid

1. Nous essayons d'avoir les mêmes plates-formes expérimentales,
2. Nous avons utilisé l'outil `wreakavoc` pour contrôler l'hétérogénéité de la plate-forme cible,
⇒ Dégrader à distance les performances du processeur et des liens réseau de chacun des hôtes de la plate-forme
⇒ À chaque variation de plate-forme, une plate-forme simulée ayant les mêmes caractéristiques est générée.

L'objectif est de comparer le comportement des algorithmes dans le cas d'une exécution réelle et dans le cas d'une exécution simulée.

Grid'5000 vs SimGrid

1. Exécution réelle

- ▷ récupère les informations :
 - * vitesse de processeur
 - * latence réseau
 - * bande passante entrante
 - * ...

⇒ Différences :

- ▷ exécution réelle : l'application découvre les variations (*en faisant des mesures*),
- ▷ exécution simulée : un schéma « simple » (*les caractéristiques de la plate-forme sont fournies à chaque modification*).

Exécution simulée

- ▷ nous fournissons l'état de la plate-forme au simulateur

Plan de l'exposé

1. Équilibrage de charge

- Le contexte
- Un exemple

2. Redistribution de données

- Le contexte
- Algorithmes de redistribution

3. Un exemple : la propagation de la chaleur

- Description
- Liens entre théorie et pratique

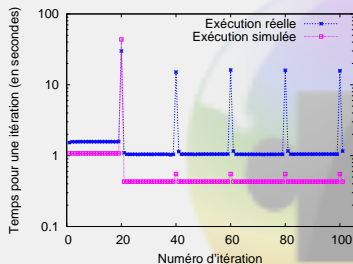
4. Résultats expérimentaux

- Le contexte
- Les expériences

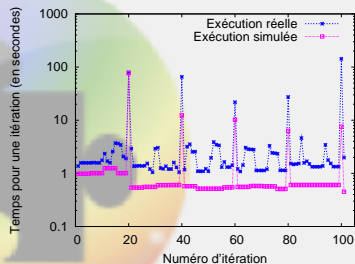
5. Conclusion



Grid'5000 vs SimGrid



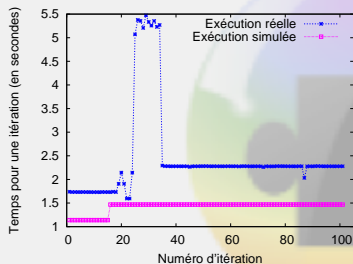
Sans modification
des caractéristiques
de la plate-forme



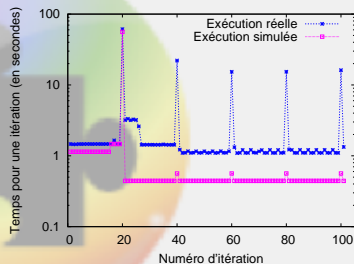
Avec modification
des caractéristiques
de la plate-forme

FIG.: Temps nécessaire (en secondes) pour chaque itération pour une exécution réelle et une exécution simulée : Impact des modifications de la plate-forme.

Grid'5000 vs SimGrid




Sans redistributions



Avec redistributions

FIG.: Temps nécessaire (en secondes) pour chaque itération pour une exécution réelle et une exécution simulée : Impact des redistributions. Une seule modification de l'état de la plate-forme est effectuée à l'itération 15.

Plan de l'exposé

1. Équilibrage de charge
 - Le contexte
 - Un exemple
 2. Redistribution de données
 - Le contexte
 - Algorithmes de redistribution
 3. Un exemple : la propagation de la chaleur
 - Description
 - Liens entre théorie et pratique
 4. Résultats expérimentaux
 - Le contexte
 - Les expériences
 5. Conclusion
- 

Conclusion

1. Deux versions d'une même application : la propagation de la chaleur
 - SimGrid
 - Grid'5000
 - ⇒ wreka voc : contrôler de manière externe et dynamique les caractéristiques de la plate-forme
2. Le comportement observé pendant une exécution simulée est très proche de celui d'une exécution réelle,
3. Ce travail représente une première étape pour la validation à travers la comparaison avec des cas réels de SimGrid dans le contexte d'applications complexes.