



Résolution de systèmes linéaires à très large échelle : méthodes classiques versus méthodes à large échelle

Raphaël Couturier

raphael.couturier@univ-fcomte.fr

FEMTO-ST, Université de Franche-Comté

3 juin 2014



- Utiliser dans de très nombreux domaines scientifiques
- $Ax = b$
- 2 grandes classes : A pleine ou creuse



- Méthodes directes :
 - Robustes et prévisibles
 - Résolution de systèmes linéaires
 - Difficiles à paralléliser et supportent mal le passage à l'échelle
- Méthodes itératives :
 - Solutions par approximation
 - Résolution de systèmes linéaires et non linéaires
 - Faciles à paralléliser et supportent mieux le passage à l'échelle

Algorithmes itératifs synchrones

- Un processeur peut commencer le calcul d'une itération que lorsqu'il a reçu les données de tous ses voisins
- Période d'inactivité entre les itérations

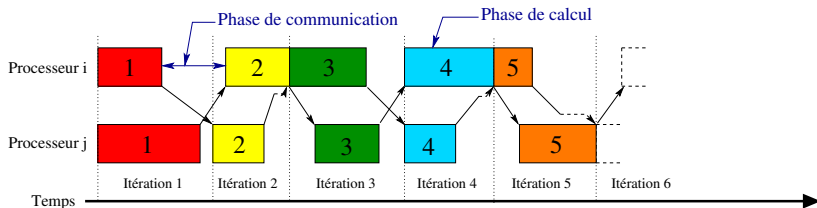


FIGURE: Schéma d'exécution d'un algorithme itératif synchrone

Algorithmes itératifs asynchrones

- Chaque processeur exécute ses propres itérations sans tenir compte de la progression des autres processeurs
- Pas d'attente des données venant des autres processeurs
- Pas de période d'inactivité entre les itérations
- Étude des conditions de convergence

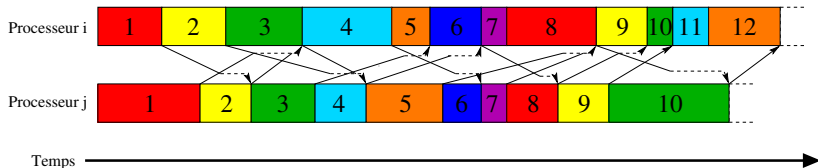


FIGURE: Schéma d'exécution d'un algorithme itératif asynchrone

Méthodes itératives très connues



- Méthode de Jacobi : $X_i^k = TX^{k-1}$
- Méthode de Gauss Seidel : $X_i^k = T X_{(l < i)}^k X_{(l \geq i)}^{k-1}$
- Méthode de Krylov : Gradient Conjugé, GMRES,

Principe : Minimiser une erreur sur un espace de krylov
 $\text{span} \{b, Ab, A^2b, \dots, A^{r-1}b\}$

- Nombreuses autres variantes

Parallélisation



- Produit matrice vecteur creux : communication locale ou globale
- Produit scalaire : communication globale
- Norme pour le résidu, test d'arrêt : communication globale
- Partitionnement de la matrice : équitable ou non
- Volume de calcul souvent faible par rapport au volume de communication

Schéma parallèle simple



- Initialisation
- Faire
 - Echange des parties de vecteurs entre processeurs
 - Calcul produit matrice vecteur
 - Souvent : calcul d'un ou plusieurs produit scalaire
 - Calcul du résidu local puis global avec échange entre processeurs
- Tant que résidu $>$ tolérance

Passage à l'échelle des méthodes



- Communication globale pénalisante
- Difficulté de faire un bon partitionnement des données
- Préconditionneur accélère la convergence (bons preconditionneurs basés sur méthodes directes)
- Utilisation d'une matrice stockée dans un fichier fastidieuse

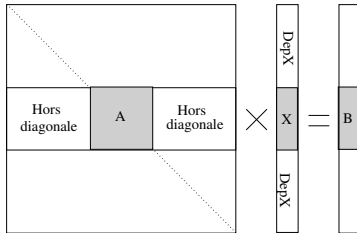
Principe de la multi-décomposition

- Décomposition (sans recouvrement) du système linéaire creux $Ax = b$ en α parties :

$$\begin{cases} A & = & [A_1, \dots, A_\alpha] \\ x & = & [X_1, \dots, X_\alpha] \\ b & = & [B_1, \dots, B_\alpha] \end{cases}$$

- Chaque sous-système linéaire $i \in \{1, \dots, \alpha\}$ est résolu par un processeur :

$$\begin{cases} A_{ij}X_j = Y_i, \text{ tel que} \\ Y_i = B_i - \sum_{j=1, j \neq i}^{\alpha} A_{ij}X_j, \end{cases}$$



Algorithme à deux niveaux avec GMRES



- Méthode à deux niveaux :
 - Itérations internes : résolution d'un gros sous-système local avec **GMRES parallèle**
 - Itérations externes : résolution du système linéaire global
- Atouts :
 - Recouvrement possible
 - Seules les itérations externes sont globales

Expérimentation Multisplitting synchrone 1/2

- Objectif : résoudre un problème de Poisson en 3D
- Matrices : 7 éléments non nuls par ligne : 6 sur diag, -1 off diag
- Hector Edinbourg Cray XE6 supercomputer (16-core AMD Opteron 2.3 Ghz 32 GB de ram réseau tore 3D)



Expérimentation Multisplitting synchrone 2/2

- GMRES standard
- Multisplitting avec minimisation : toutes les 20 itérations une étape de minimisation pour accélérer la convergence

Pb size	Nb. cores	GMRES			Multisplitting avec minimisation			Ratio
		Time (s)	nb Iter.	Δ	Time (s)	nb Iter.	Δ	
468 ³	2,048 (2x1,024)	299.7	41,028	5.02e-8	48.4	691(6,146)	8.24e-08	6.19
590 ³	4,096 (2x2,048)	433.1	55,494	4.92e-7	74.1	1,101(8,211)	6.62e-08	5.84
743 ³	8,192 (2x4,096)	704.4	87,822	4.80e-07	151.2	3,061(14,914)	5.87e-08	4.65
743 ³	8,192 (4x2,048)	704.4	87,822	4.80e-07	110.3	1,531(12,721)	1.47e-07	6.39



- Autres problèmes numériques
- Itérations asynchrones ?
- Recouvrement pour accélérer la convergence
- Expérimentations à plus large échelle => Exascale
- SimGrid ?