Méthodologie pour la Parallélisation d'Algorithmes de Reconstruction 3D Application à la Tomographie X

C. LAURENT*, F. PEYRIN*+ et J.-M. CHASSERY*

*TIMC-IMAG, Institut Albert Bonniot, Domaine de la Merci, 38706 La Tronche
+CREATIS, bat 502,INSA, 69621 Villeurbanne
GDR 134 TDSI CNRS

RÉSUMÉ

Depuis quelques années, on constate une nette évolution de l'imagerie médicale vers l'imagerie 3D, qui permet d'obtenir et de traiter non plus des coupes isolées mais des volumes entiers. Toutefois la reconstruction et la manipulation d'images 3D soulèvent un certain nombre de problèmes, tant au niveau des concepts, que des ressources informatiques nécessaires. Pour reconstruire des images médicales de taille réaliste dans des temps acceptables, les approches parallèles semblent bien adaptées. Nous proposons une méthodologie pour implémenter des méthodes de reconstruction 3D sur une machine parallèle virtuelle, pour ensuite les optimiser sur nos machines cibles. Nos résultats montrent les performances obtenues et comparent la qualité des reconstructions des méthodes utilisées.

ABSTRACT

Since a few years, we note an evolution of medical imaging towards 3D imaging, which allows to obtain and to treat a fully 3D data instead of just a set of human sections. However the reconstruction and the manipulation of 3D images raise a number of problems, on concept level as well as on the necessary computer ressources. The parallel approach seems to be a good solution for the reconstruction of a medical image with a realistic size in acceptable time. We propose a methodology to implement the 3D reconstruction methods on a parallel virtual machine and then to optimize this implementation on the target machines. The results show the obtained performances and compare the quality of the used reconstruction methods.

1 Introduction

En tomographie par rayons X réellement tridimensionnelle, les images 3D sont obtenues par résolution d'un problème de reconstruction d'images à partir de projections. Récemment de nouveaux systèmes d'acquisition utilisant des sources coniques de rayons X ont été développés [6]. Ils permettent d'acquérir un ensemble de radiographies 2D sous différents angles de vue, à partir duquel l'image 3D doit être reconstruite. Les algorithmes séquentiels classiquement utilisés, doivent alors gérer des volumes importants de données (536 MBytes pour une image 512^3) et conduisent à des temps rédhibitoires (10^3 secondes pour reconstruire une image 128^3 à partir de 128 projections 128^2). Pour reconstruire des images médicales de taille réaliste dans des temps acceptables, les approches parallèles ont déjà été envisagées [1, 3, 4, 11, 12]. Nous présenterons différentes approches pour paralléliser ce problème de reconstruction 3D sur différentes architectures parallèles. Notre objectif est de proposer une méthodologie pour implémenter ces méthodes sur une machine parallèle virtuelle, pour ensuite les optimiser sur nos machines cibles. Nos résultats montrent les performances obtenues et comparent la qualité des reconstructions des méthodes utilisées.

2 La tomographie par rayons X

2.1 Son concept

Le problème de la tomographie 3D consiste à reconstruire une image 3D à partir d'un ensemble de données 2D correspondant à des "projections" 2D de l'objet 3D. Globalement ces données correspondent à des intégrales de l'objet sur des droites de l'espace. Leurs organisations dépendent de la géométrie du système d'acquisition. Le problème de la reconstruction consiste à déterminer une fonction $f(x, y, z)$ à partir d'un ensemble de projections $P_{\vec{\tau}}(r')$, $\vec{\tau}$ parcourant un sous-ensemble de S^2, la sphère unité.

2.2 Les méthodes de reconstruction

Les principales méthodes de reconstruction 3D peuvent être regroupées en plusieurs classes [7]. Ces méthodes peuvent être mises en oeuvre en utilisant des concepts similaires, c'est à dire en exploitant la dualité des opérateurs de base de type "projection" et "rétroprojection". On présente ici quelques méthodes de reconstruction.

Méthode de Feldkamp

Cette méthode analytique est la généralisation de l'algorithme par rétroprojection filtrée en 3D, elle fournit une méthode approchée de reconstruction. Chaque projection 2D est d'abord pondérée puis filtrée, ce qui nécessite une transformée de Fourier 2D, puis le résultat est rétroprojeté.

Méthode Art

C'est une méthode itérative qui résulte d'une formulation discrète. Le problème de la reconstruction revient alors à la résolution d'un système linéaire. Chaque ligne du système linéaire correspond à l'équation d'un rayon de projection.

Méthode Art par blocs

C'est une méthode dérivée de la méthode ART. Au lieu de résoudre l'équation de chaque rayon, on traite un bloc de données correspondant à une projection cônique.

Méthode SIRT

Cette méthode, similaire à ART, traite les projections 2D pixel par pixel, au lieu de traiter des rayons de projection à l'intérieur du volume à reconstruire.

Méthodologie pour la parallélisation des méthodes

La parallélisation des méthodes de reconstruction 3D peut être abordée par deux concepts différents. L'analyse ascendante consiste à définir les méthodes de reconstruction 3D parallèles comme un problème classique d'algèbre linéaire. La parallélisation est alors basée sur l'utilisation de routines d'algèbre linéaire parallèles. L'analyse descendante consiste à conserver la structure des méthodes de reconstruction basée sur les opérateurs de projection et de rétroprojection. L'effort de la parallélisation se porte alors sur ces opérateurs.

3.1 Analyse ascendante

Comme les méthodes de reconstruction 3D sont assimilables à la résolution de systèmes linéaires creux, la parallélisation de ces méthodes va consister à identifier les noyaux de calculs parallélisables et à utiliser au mieux la version parallèle de ces noyaux de calcul. De nombreuses librairies de calculs scientifiques ont été développées dans ce sens [5]. Généralement pour ce type d'approche, l'efficacité de l'implémentation est priviligiée au détriment de la portabilité de l'implémentation.

3.2 Analyse descendante

Notre analyse consiste à paralléliser les méthodes de reconstruction 3D sur une machine parallèle virtuelle. Pour définir le modèle de programmation de notre machine parallèle, on doit s'intéresser au parallélisme intrinsèque des méthodes de reconstruction qui est définie suivant deux modèles: le parallélisme de contrôle et parallélisme de données. Dans notre cas, les données sont réparties en deux ensembles distincts: les acquisitions 2D et l'image 3D à reconstruire. Les voxels sont reliés aux pixels par des relations de type projection. Cependant les voxels comme les pixels sont indépendants entre eux. Ainsi, la reconstruction 3D peut être abordée par un modèle de parallélisme de données.

Nous déterminons dans un premier temps l'architecture de cette machine virtuelle et quelques critères pour évaluer ses performances. Puis, nous explicitons la répartition des données sur les mémoires des processeurs et leurs communications à travers le réseau d'interconnexion des processeurs. La parallélisation des méthodes consiste en un découpage de celles-ci en terme d'opérateurs de base. Nous parallélisons ces opérateurs de bases et nous les intégrons aux différentes méthodes. Nous obtenons ainsi des méthodes parallèles basées sur des opérateurs similaires. Ces opérateurs sont implémentés soit par une approche locale, soit par une approche globale.

3.2.1 Architecture et évaluations

Notre machine parallèle virtuelle est une machine composée de PE processeurs à mémoire distribuée. La topologie de cette machine virtuelle n'est pas fixe, ce qui nous permet de la mapper sur des machines éxistantes. Pour évaluer les performances de notre machine parallèle virtuelle, nous utilisons trois critères:

- l'accélération qui mesure le gain obtenu entre la version parallèle et la version séquentielle d'un algorithme. $Acc = \frac{Temps\ séquentiel}{Temps\ parallèle}$ où le temps séquentiel représente le temps d'exécution sur un processeur de la machine parallèle virtuelle.
- l'efficacité qui mesure si les processeurs sont utilisés de manière optimale. $Eff = \frac{Accélération}{PE}$
- l'accélération relative qui mesure le gain obtenu entre la version parallèle d'un algorithme et sa version séquentielle sur une machine séquentielle de référence. $AccRel = \frac{Temps\ séquentiel\ de\ référence}{Temps\ parallèle}$.

3.2.2 Répartition des données et schéma de communication

L'image 3D (N^3) comme les m acquisitions 2D (M^2) sont réparties sur la mémoire des processeurs. Nous proposons deux répartitions en fonction de la taille de la mémoire des processeurs.

- Quand la taille des données est plus grande que la taille des mémoires des processeurs, l'image 3D est décomposée en T tranches de taille respective $N_T.N^2$ avec $T.N_T = N$. Nous considérons que nous avons à chaque instant, une sous-image 3D de taille $N_T.N^2$ et une acquisition 2D réparties sur les PE mémoires.

- Quand la taille des données est moins grande que la taille des mémoires des processeurs, nous supposons que l'image 3D est décomposée en NPE sous-images de taille $N_{PE}.N^2$ avec $PE.N_{PE} = N$, et que les m acquisitions 2D sont distribuées sur PE processeurs, avec m_{PE} images sur chaque processeur ($m = m_{PE}.PE$).

Les données étant réparties sur les processeurs, le problème est de choisir un schéma de communication pour le transfert des données. Chaque voxel se projette sur toutes les images 2D en fonction de l'angle de vue. Cela signifie que chaque sous-image 3D est en relation avec les m images 2D. Les données étant réparties sur les processeurs, on doit choisir de communiquer soit les images 2D, soit les sous-images 3D en fonction de la taille des données à transférer. Nous supposons que généralement $N^3 > m.M^2$, donc nous communiquons les images 2D. Le schéma de communication dépend de l'approche locale ou globale. Son principal critère est qu'il puisse se plonger sur les topologies des machines existantes.

3.3 Parallélisation des opérateurs de base

Les opérateurs de nombreuses méthodes de reconstruction sont la projection et la rétroprojection. Ils utilisent des opérations de projection et de rétroprojection avec une interpolation bilinéaire. Nous présentons leurs algorithmes séquentiels avec les notations suivantes.

V: image 3D à reconstruire composée de N^3 voxels v_i et répartie en sous images V_i.

Im_j: image 2D pour les acquisitions ou les projections de l'image 3D. Il y a m Im_j composées de M^2 pixels im_{ja}.

$\xrightarrow{\mathcal{P}}$: Opération de projection : v_i contribue à la valeur des pixels $im_{ja}, im_{jb}, im_{jc}, im_{jd}$.

$\xleftarrow{\mathcal{R}}$: Opération de rétroprojection : les pixels $im_{ja}, im_{jb}, im_{jc}, im_{jd}$ contribuent à la valeur du voxel v_i.

Algorithme de Projection	Algorithme de Rétroprojection
lire(V)	créer(V)
Pour tous les Im_j	Pour tous les Im_j
$\quad$ créer(Im_j)	$\quad$ lire(Im_j)
$\quad$ Pour tous les v_i	$\quad$ Pour tous les v_i
$\quad\quad v_i \xrightarrow{\mathcal{P}} im_{ja}, im_{jb}, im_{jc}, im_{jd}$	$\quad\quad v_i \xleftarrow{\mathcal{R}} im_{ja}, im_{jb}, im_{jc}, im_{jd}$
$\quad$ écrire(Im_j)	écrire(V)
Complexité:$4.m.N^3$	Complexité:$4.m.N^3$

3.3.1 Approche locale

L'approche locale consiste à utiliser les opérateurs sur les données se trouvant sur un même processeur. Par exemple, pour la projection d'une image 3D, le processeur PE_i effectue la projection de sa sous-image V_i sur les images 2D présentes dans sa mémoire. Puis il envoie ses images 2D aux processeurs n'ayant pas encore traité ces données, et il attend un nouveau jeu de données venant d'un autre processeur. Le schéma de communication que nous utilisons pour cette approche est l'anneau, où chaque processeur communique avec son voisin.

3.3.2 Approche globale

Au contraire de l'approche locale, les opérateurs de l'approche globale utilisent des données se trouvant sur l'ensemble des processeurs. Typiquement pour la projection sur une image Im_j, les processeurs calculent des projections partielles Im'_j, puis pour avoir la projection complète de l'image 3D sur Im_j, les processeurs effectuent une réduction-sommation sur le processeur possédant l'image Im_j. Le schéma de communication dépend de l'implémentation de la réduction.

3.3.3 Machines et librairies de communication

Les implémentations des approches locale et globale sont réalisées sur des architectures parallèles SIMD[1] et MIMD[2] ayant respectivement un mode de contrôle synchrone et asynchrone. Le tableau 1 présente les machines cibles. Nous utilisons comme librairies parallèles, soit PVM[2] qui modélise bien notre approche de machine virtuelle, soit celles propres aux architectures cibles.

Machine	Type	Processeur(Nb)	Topologie
SUN4	Séquentiel	Sparc2(1)	
MasPar	SIMD	Elémentaire (1024)	Grille
Réseau de Stations	MIMD	Sparc2 (2) Sparc1(3)	Anneau
Paragon (Intel)	MIMD	i860(32)	Grille
T3D (Cray)	MIMD	AXP(128)	Tore 3D
SP1 (IBM)	MIMD	RS600(32)	Cross-bar
Ferme de Processeurs	MIMD	AXP(16)	Giga-Switch (fibre optique)

TAB. 1 - *Machines Parallèles*

3.3.4 Optimisation du parallélisme

Pour améliorer l'efficacité des implémentations, nous utilisons deux approches complémentaires. La première permet de minimiser les coûts de communication, la seconde permet de réguler au mieux la charge de chaque processeur. Nous les avons développés dans une précédente étude [8].

Recouvrement des communications par les calculs
Il existe différentes méthodes pour minimiser le coût des communications. Une idée simple est le recouvrement des communications par les calculs. Pour cela, les routines de communications utilisées sont des routines non-bloquantes. Notre approche consiste à calculer localement les données en attendant que de nouvelles données arrivent. Si à la fin du traitement des données et de leur envoi, le processeur n'a toujours pas reçu de nouvelles données, il ira les chercher sur l'unité de stockage.

1. Single Instruction Multi Data
2. Multi Instruction Multi Data

Partitionnement adaptatif
Nous développons ici une méthode simple pour équilibrer la charge des processeurs. Son principe est de calculer, pour chaque processeur PE_i, sa vitesse relative :
$Vr_i^k = \frac{V_i^k}{V_{max}^k}$ où $V_i^k = \frac{D_i^k}{Texe_i^k}$ avec $Texe_i^k$ son temps d'exécution et D_i^k la taille de sa sous-image 3D.
Le nouveau partitionnement est $D_i^{k+1} = \frac{N}{\sum_{i=1}^{PE} Vr_i^k} . Vr_i^k$
Cette méthode est applicable dans deux cas de figure, si l'architecture de la machine cible est composée de processeurs hétérogènes, ou si la taille de la zone est réduite par un seuillage sur l'image.

3.4 Les implémentations parallèles des méthodes

Les méthodes sont implémentées à partir des opérateurs parallélisés. Nous avons déterminé pour chacune la meilleure approche possible pour leur parallélisation [10].

la méthode de Feldkamp: Cette méthode ne définit pas un ordre pour le traitement des acquisitions 2D. On peut donc calculer leurs contributions par une approche locale. Nous utilisons un opérateur de rétroprojection parallèle auquel se rajoute une opération de pondération et de filtrage.

la méthode ART par blocs: Cette méthode nécessite de reconstruire l'image 3D à partir de l'image de différence entre chaque acquisition 2D et la projection de l'image 3D pour chaque angle de vue. Ce traitement ne peut être fait que de manière séquentielle. Nous utilisons un opérateur de projection globale pour calculer la projection de l'image 3D, et après avoir diffusé l'image de différence, on effectue en local une rétroprojection de celle-ci.

la méthode SIRT: Cette méthode a l'avantage comme la méthode de Feldkamp de pouvoir calculer les contributions des acquisitions 2D sans ordre précis. Elle est basée sur des opérateurs parallèles de projection et de rétroprojection implémentés par une approche locale.

Le problème de ces deux dernières méthodes itératives est le nombre d'itérations nécessaires pour obtenir une bonne reconstruction.

4 Résultats

Nous avons comparé les différentes implémentations des méthodes de reconstructions 3D. A titre d'exemple, la figure 1 présente les accélérations relatives par rapport au temps sur une station SUN4, obtenues sur nos machines cibles. Nous reconstruisons une image de taille 128^3 à partir de 128 acquisitions 128^2. Cette comparaison montre bien que les machines SIMD comme la Maspar ne sont pas adaptées à ce type de problème à cause de la faible capacité de la mémoire de leurs processeurs. Par contre, les machimes MIMD obtiennent des bonnes performances de manière générale [9]. A titre indicatif notre machine séquentielle de référence, un SUN4, met 10000 secondes pour reconstruire l'image 3D, tandis que le T3D met seulement 22 secondes. Les techniques de recouvrement et de partitionnement adaptatif ont permis d'augmenter les efficacités des implémentations.

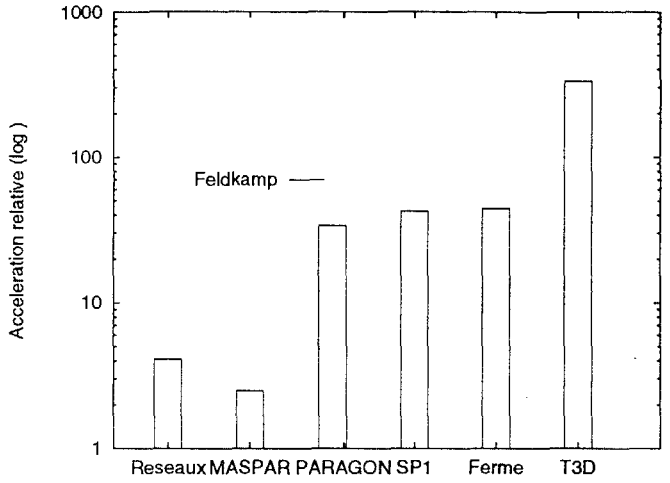

FIG. 1 - *Accélération relative de l'implémentation de Felkamp sur différentes machines parallèles.*

Le tableau 2 compare les temps d'exécution des méthodes de reconstructions sur le Cray T3D pour les données précédentes. Il montre que les approches locales sont plus efficaces en moyenne. Mais le problème est le prix à payer en termes de temps de calcul pour obtenir une image reconstruite de bonne résolution.

Algorithme	Feldkamp	ART par blocs	SIRT
Approche	locale	globale	locale
Nombre d'itérations	1	5	30
Temps Total (sec)	22	432	1021
Temps moyen (sec)	22	86	33

TAB. 2 - *Implémentations des méthodes sur le Cray T3D*

En effet, la figure 2 permet de comparer la qualité de la reconstruction par rapport à une image simulée. La méthode SIRT obtient la meilleure reconstruction mais pour 400 itérations, tandis que la méthode de Felkamp obtient une bonne approximation en une opération d'inversion. En faisant varier le facteur de relaxation de la méthode ART (ici 1.75), la méthode converge plus vite et on diminue ainsi le nombre d'itérations, donc le temps d'exécution.

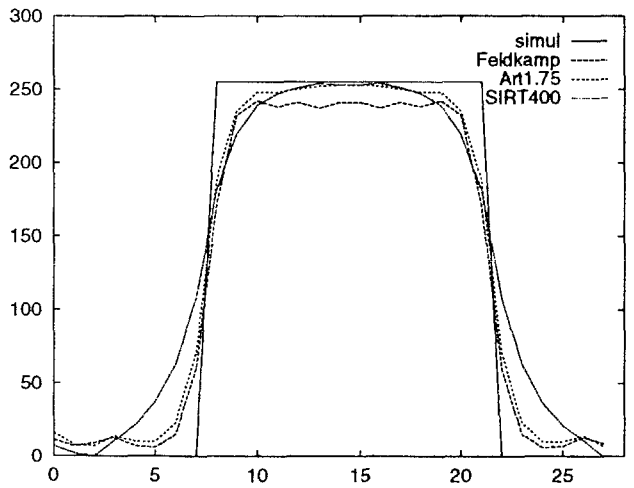

FIG. 2 - *Comparaison des profils d'une sphère reconstruite avec le profil d'une sphère simulée.*

5 Conclusion

Nous avons proposé une méthodologie générale pour paralléliser les méthodes de reconstruction 3D. Notre démarche s'est appuyée sur une analyse descendante du problème en développant nos méthodes sur une machine parallèle virtuelle. Cela nous a permis de comparer les différentes implémentations sur nos machines cibles sans contraintes d'architecture.

Les résultats montrent que les algorithmes de reconstruction 3D sont des problèmes très parallélisables. Pour augmenter l'efficacité des méthodes itératives, il est nécessaire de choisir de bons paramètres pour accélérer la convergence. Pour valider nos reconstructions, Les images sont reconstruites à partir de données simulées.

Nous travaillons actuellement sur des images expérimentales provenant du Morphomètre [6]. D'un point de vue pratique, les temps de reconstruction obtenus (5mn pour une image 256^3 à partir de 256 projections 256^2) doivent permettre d'améliorer la résolution maximale actuelle des images 3D. D'autre part, l'utilisation de réseaux de stations et de fermes de processeur fournit une solution à moindre coût pour des temps de reconstruction intéressants.

Références

[1] M. S. Atkins, D. Murray, and R. Harrop. Use of transputers in a 3-D Positron Emission Tomograph. *IEEE Transactions on Medical Imaging*, 12(2):173–181, 1993.

[2] A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. PVM3 User's Guide and Reference Manual. Technical report, Oak Ridge National Laboratory, 1994.

[3] H.-P. Charles, J.-J. Li, and S. Miguet. 3D image processing on distributed memory parallel computers. In *SPIE*, volume 1905, pages 379–390, 1993.

[4] C. M. Chen and S.-Y. Lee. On Parallelizing the EM Algorithm for 3-D PET Image Reconstruction. *IEEE Transactions on Parallel and Distributed Sytems*, 5(8):860–873, 1994.

[5] J. Choi, J. Dongarra, R. Pozo, and D. Walker. ScaLAPACK: A Scalable Linera Algebra Library for Distributed Memory Concurrent Computers. Technical Report LAPACK WN 55, University of Tennessee, 1992.

[6] D. Saint-Félix et al. A New system for 3D computerized X-RAY angiography: first in vivo result. In *Annual International Conference of the IEEE Engineering in Medecine and Biology Society*, pages 2051–2052, 1992.

[7] L. Garnero and F. Peyrin. Methodes de Reconstruction 3D en Tomographie X. Technical report, GDR TDSI CNRS, France, May 1993. Rapport de synthèse 93-01.

[8] C. Laurent, C. Calvin, F. Peyrin, and J.-M. Chassery. Efficient Implementation of Parallel Image Reconstruction Algorithms for 3D X-RAY Tomography. In *PARCO 95*, Gent(Belgique), September 1995.

[9] C. Laurent, F. Peyrin, and J.-M. Chassery. Comparaison of Parallel Reconstruction Algorithms for 3D X-RAY Tomography on MIMD computers. In *High Performance Computing Symposium 95-HPCS 95*, Montréal(Canada), July 1995.

[10] C. Laurent, F. Peyrin, and J.-M. Chassery. Parallélisation de Méthodes de Reconstruction 3D par des Approches Locales et Globales. In *Actes des 7ièmes rencontres sur le parallélisme-RenPar7*, Mons(Belgique), Mai 1995.

[11] M. I. Miller and C. S. Butler. 3-D maximun *a posteriori* Estimation for Single Photon Emission Computed Tomography on Massively-Parallel Computers. *IEEE Transactions on Medical Imaging*, 12(3):560–565, 1993.

[12] E. L. Zapata, I. Benavides, F. F. Rivera, J. D. Bruguera, and J. M. Carazo. Image Reconstrcution on Hypercube Computers. In *The 3rd Symposium on the Frontiers of Massively Parallel Computation*, pages 127–133, 1990.