

DEFINITION DE MESURES OBJECTIVES DE PERFORMANCES  
POUR LA MISE EN OEUVRE PARALLELE D'ALGORITHMES DE  
TRAITEMENT D'IMAGE.

E. MARTIN, H. DUBOIS, O. SENTIEYS & J.L. PHILIPPE

L.A.S.T.I. - E.N.S.S.A.T et GDR 134, GRECO TDSI.  
Laboratoire d'Analyse des Systèmes de Traitement de l'Information  
E. N. S. de Sciences Appliquées et de Technologie, 6 rue de Kerampont - 22305 LANNION Cedex

RÉSUMÉ

*Le problème de la mise en oeuvre parallèle des algorithmes de traitement d'images, lié au choix d'une architecture cible, est crucial dans toutes les applications sous contraintes de temps réel. Il est souvent bien trop coûteux de concevoir entièrement un circuit spécifique, et de nombreux concepteurs s'orientent vers la mise en oeuvre de systèmes multiprocesseurs. Nous présentons dans cet article, une méthodologie de conception des réseaux multiprocesseurs, reposant sur une mesure objective des performances.*

**1 Méthode de conception:** modèles et mesures.

La mise en oeuvre parallèle d'un algorithme passe par les deux étapes de parallélisation et de conception de l'architecture.

Nous avons retenu pour la première, le cadre de la parallélisation par le partage des données. Cette orientation est bien adaptée aux algorithmes de traitement des images, avec cependant quelques réticences concernant les algorithmes irréguliers.

La conception d'une architecture multiprocesseurs repose sur le choix du nombre de processeurs à mettre en oeuvre, et celui du réseau d'interconnexion leur permettant d'échanger les données à traiter. Ces échanges dépendent étroitement de la nature de l'algorithme (échanges locaux, de voisinages ou globaux) et du partage des données effectué. Nous montrons qu'il est possible d'évaluer le comportement de l'algorithme parallélisé à partir d'une émulation du code séquentiel, et de modèles simples sur le processeur, l'algorithme et le réseau d'interconnexion considérés. Cette approche offre l'avantage de pouvoir explorer une grande diversité de solutions, sans générer à chaque fois le code parallèle.

1.1 Modélisation.

Les algorithmes sont caractérisés par:

- \*  $p\%$ : pourcentage de tâches effectuées en parallèle sur N processeurs.
- \*  $nb(T\hat{a})$ : nombre de tâches correspondant au traitement de l'ensemble de l'image.
- \*  $Dp$ : volume de données échangées pour tout processeur 'i' communiquant avec tout processeur 'j'.

ABSTRACT

*The parallel implementation of image processing algorithms, and the choice of target architectures is a very crucial problem in all applications under the constraint of real time. The design of application specific circuits is often too costly and many designers use multi processors systems. We present in this article a methodology depending on objective performance measures in order to design multi processors networks.*

\*  $C$ : taux de communication, nombre de communications par tâche élémentaire.

$C$  représente l'importance des communications à effectuer, vis à vis des calculs, ces deux grandeurs étant ramenées à une même unité de temps. Nous verrons que  $C$  est un paramètre important pour juger de la possibilité de saturation des réseaux de communications entre processeurs.

La valeur de  $p\%$  dépend de la nature de l'algorithme, par contre  $Dp$  varie suivant les propositions de découpages de l'image à traiter sur les  $N$  processeurs à mettre en oeuvre.

Les processeurs sont modélisés par:

\*  $T_{com}$ : durée d'une communication entre deux processeurs voisins.

\*  $t\%T_{com}$ : 'pollution' apportée par les communications sur le calcul, encore appelé taux de recouvrement.

$T_{com}$  et  $t\%$  dépendent uniquement du processeur considéré. Dans le cas du transputer,  $t\%$  a été obtenu par une mesure expérimentale sur le comportement des processus gérant les échanges de données.

Nous nous placerons dans une hypothèse d'équirépartition temporelle des échanges: L'échéance dans le temps des transferts ne bloque en aucune manière le déroulement des calculs en cours; il sera possible de concevoir que les temps de calculs élémentaires ne diffèrent pas quel que soit le degré de parallélisme exploité. Ne s'appliquant qu'à une classe limitée d'algorithmes, nous montrerons sur le second exemple (2.3) comment considérer cette hypothèse dans le cadre d'une chaîne de traitements.

Nous émulerons les valeurs de temps:

\*  $temps(T\hat{a})$ : temps de traitement d'une de ces tâches.



\*  $T_{seq}$ : temps nécessaire à l'exécution de l'algorithme sur l'ensemble des données, à savoir l'image dans sa totalité sur un seul processeur:  $T_{seq} = nb(T\hat{a}) \times temps(T\hat{a})$ .

Le réseau d'interconnexion reliant les différents processeurs élémentaires est caractérisé par les deux paramètres suivants:

\* degré: nombre de liaisons utilisables simultanément pour dialoguer entre deux processeurs..

\* diamètre: distance moyenne d'un processeur à un autre.

Nous trouverons les caractéristiques de quelques réseaux d'interconnexion dans le tableau qui suit.

### 1.2. Mesures prévisionnelles.

Nous avons défini trois mesures devant guider le choix d'un réseau: l'accélération, l'efficacité et la saturation. Si les deux premières sont bien connues (cas des architectures vectorielles), la troisième par contre apportera une connaissance importante: tout réseau hors saturation présente une capacité d'écoulement des communications aussi performante que le plus complexe d'entre-eux. Cette mesure importante doit conduire au choix objectif du réseau répondant au problème d'écoulement des données, sous contrainte de coût minimum.

Hors saturation du réseau de communication, le temps d'exécution de notre algorithme sur le multiprocesseur est donné par:

$$(1) T_{cal \text{ hors sat}} = p \frac{T_{seq}}{N} + (1-p) T_{seq}$$

L'efficacité de la mise en oeuvre parallèle est identique pour tout réseau:

$$(1') \text{Eff}_{\text{hors sat}} = \frac{1}{(1-p).N + p}$$

En supposant les communications équiréparties, la saturation du réseau est obtenue lorsque le temps des échanges est prépondérant devant le temps nécessaire aux calculs. Le temps d'exécution de l'algorithme devient alors:

$$(2) T_{cal \text{ sat}} = T_{ech} = N D_p \frac{T_{com \text{ diamètre}}}{\text{degré}}$$

Le tableau suivant présente les valeurs de limite de saturation et l'efficacité après saturation, calculées pour différents réseaux.

Reseau	Crossbar	Multibus (B bus)	Anneau	Carré maillé
Diamètre	1	1	$(N + 1) / 4$	$\sqrt{N} / 2$
Degré	N	B	2 N	4 N
Limite de saturation	$\frac{(C-1)p}{1-p} > N$	$\frac{Bp}{(Cp-(1-p)B)} < N$	$\frac{8p}{Cp-8(1-p)} < N$	$[\frac{8p}{Cp-8(1-p)}]^2 < N$
Efficacité après saturation	$\frac{1}{C.p}$	$\frac{B}{N.C.p}$	$\frac{8}{C.p.(N+1)}$	$\frac{8}{C.p.\sqrt{N}}$

Nous ferons trois commentaires:

- Un réseau cross-bar ne peut se saturer que si  $C > 1$
- La limite de saturation croit lorsque p diminue,
- Lorsque l'efficacité décroît en N, l'accélération est constante, et il devient inutile d'ajouter des processeurs.

Les valeurs ci-dessus sont considérées dans le cas idéal où les communications ne provoquent aucun recouvrement sur les calculs. En tenant compte du paramètre t%, on ajoutera au temps de calcul  $T_{cal}$  le terme  $\frac{t\% \cdot T_{com} D_p \text{ diamètre}}{\text{taille}}$  (3) où 'taille' représente la taille des données élémentaires circulant sur le lien.

Les mesures sont émulées à l'aide de notre outil ESPION, à partir du comportement de l'algorithme décrit séquentiellement. Nous retrouverons ci-dessous (2.2 et 2.3) les valeurs de saturation et d'efficacité. La première est obligatoirement liée à chaque fraction de traitements où calculs et transferts se font en parallèle; la seconde mesure sera évaluée sur l'ensemble des traitements.

## 2 Résultats.

Nous avons utilisé et vérifié notre méthodologie de conception dans le cadre de la mise en oeuvre d'algorithmes de traitement d'image sur réseau de transputers. Les algorithmes

retenus, la transformée de Hadamard 2D et un filtre de Kalman, sont séparables. La valeur de p% est égale à 1 dans les deux cas.

### 2.1 Modélisation du processeur.

Le transputer est un processeur bien adapté à la configuration de machines MIMD à mémoire distribuée. Les communications entre processeurs se font point à point à l'aide de liens séries. Cependant la complexité de sa structure interne (mémoires rapides et lentes, gestionnaire de processus, unité de calculs flottants) alliée à l'environnement de programmation OCCAM, ne permettent pas une prévision aisée du temps de calculs d'un algorithme. Ces temps seront donc émulés à partir du code séquentiel.

Nous avons mesuré la valeur  $T_{com}$ , temps de communication d'une donnée élémentaire sur un lien.  $T_{com}$  dépend du format de la donnée et de la rapidité du lien série: à 20 Mbit/s sur le T800,  $T_{com}$  vaut 0,85  $\mu$ s pour un octet et 6,8  $\mu$ s pour une donnée complexe.

t% défini le taux de recouvrement des transferts sur les calculs, et dépend dans le cas du transputer de la taille des messages communiqués. Difficile à modéliser, il a été mesuré pour un contexte bien défini de processus gérant les communications. Pour une donnée complexe, on relèvera  $t\% \cdot T_{com} = (2,9 \cdot \text{taille} + 100) \mu$ s.

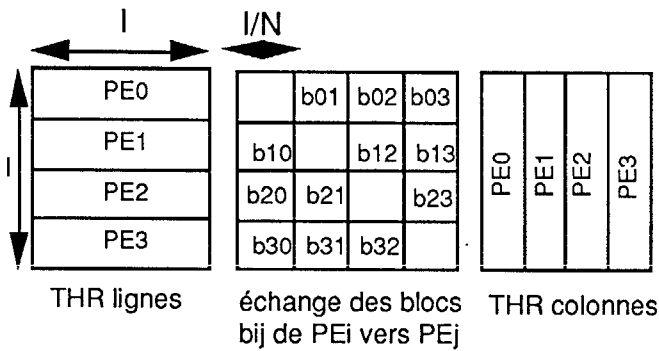
Le temps nécessaire aux échanges est la somme des temps de communication transparents pour les calculs et des temps recouvrant les calculs:

$$Tech = N \cdot Dp \frac{Tcom \cdot \text{diamètre}}{\text{degré}} + t\% \cdot Tcom \frac{Dp}{\text{taille}} \text{diamètre.}$$

Ces résultats montrent qu'il serait loisible de discuter la taille optimale des blocs de données à transférer (taille), en retenant que plus celle-ci augmente, plus les échanges sont efficaces, mais plus elle diminue et moins le retard amené par les dernières communications sera important.

2.2 Transformée de Hadamard 2D.

Nous utiliserons la séparabilité de l'algorithme pour effectuer en parallèle les transformées 1D rapides (FHT) sur les lignes, puis les transformées sur les colonnes. Pour diminuer les coûts de communications, une rotation de l'image ligne-colonne est effectuée immédiatement à la suite de chaque transformée ligne.



La taille des blocs transférés est égale à I/N, longueur de chaque fraction d'une ligne à communiquer à chaque autre processeur; Dp vaut:  $I^2 \cdot (N-1) / N^2$  données. Le taux de communication C = N.Dp.Tcom/Tseq où Tseq est le temps séquentiel d'exécution de l'algorithme:  $Tseq = I^2 \cdot \log_2 I$ . Tpap, où Tpap, temps de calcul d'un papillon, est le temps nécessaire à effectuer 4 additions (ou soustractions) réelles; On émule une valeur de Tpap = 4μs (qui diffère notablement d'une prévision à partir du code).

Lorsque N est grand devant 1, C est quasiment indépendant de N, et vaut pour une image 128x128: C = 0,22 (les données sont réelles, soit 4 octets).

Les limites de saturation du réseau d'interconnexion en anneau sont relevées dans le tableau suivant. Il est intéressant de noter que celui-ci reste performant (hors saturation) jusqu'à un nombre de processeurs important, bien que sa topologie ne soit pas adaptée a priori aux communications des données.

Taille de l'image I	32	64	128	256	512
Nsat pour FHT	23	28	32	37	41

Nsat représente la valeur de N, nombre de processeurs, à partir duquel le réseau est en saturation.

Hors saturation, on montre ((1) + (3)) que l'efficacité décroît lorsque le nombre de processeurs augmente. La chute de performance est due principalement à l'accroissement des communications, et leur recouvrement sur les calculs.

Nous avons relevé les performances prévisionnelles pour une image de taille 128x128.

N processeurs	1	2	3	4	7	8	9	10
Temps idéal (s)	0,4469	0,2234	0,1490	0,1117	0,0638	0,0559	0,0497	0,0447
Temps prévu (s)	0,4469	0,232	0,162	0,128	0,0906	0,086	0,827	0,0808
Efficacité en %	100	96,1	91,7	86,8	70,43	65,1	60	55,3

La finesse des prévisions nous permet d'obtenir une efficacité qui diffère peu de celle qui est mesurée.

Lorsque la taille de l'image augmente, l'efficacité augmente de même. Nous le constatons sur l'exemple mettant en œuvre 10 processeurs.

Taille I de l'image	32	64	128	256	512
Temps idéal (s)	0,002	0,01	0,045	0,204	0,92
Temps prévu (s)	0,01	0,025	0,08	0,29	1,13
Efficacité en %	20	36,5	55,3	70,8	81

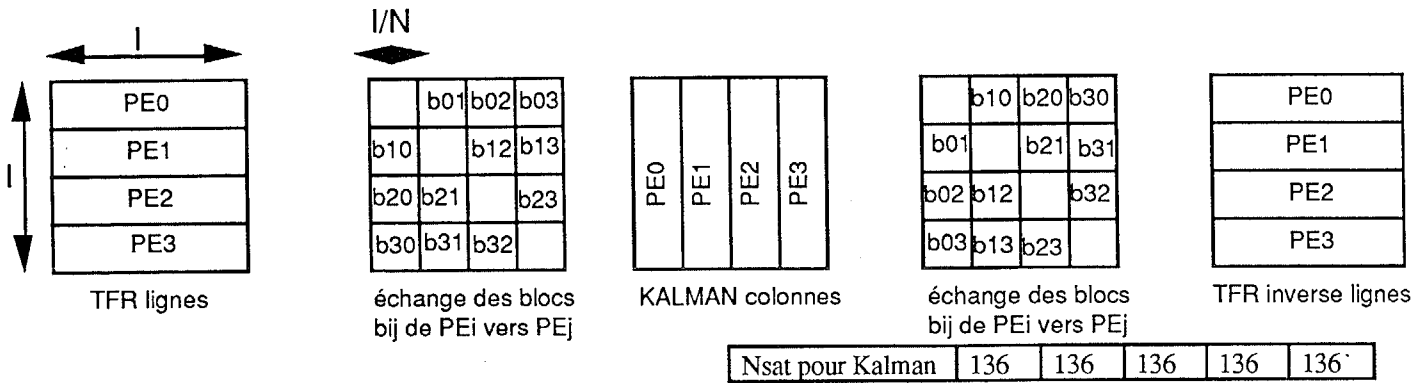
Ces valeurs prévisionnelles doivent permettre au concepteur de rejeter une solution parallèle dont l'efficacité s'avère trop faible. Il est alors nécessaire de s'orienter soit vers une autre découpe des images traitées, soit vers un autre parallélisme (dans le temps, parallélisme de tâches, ...).

2.3 Filtrage de Kalman.

Le filtrage de Kalman, dans le cadre du traitement d'image, a pour application la restauration d'images floues. Il a été montré dans [1] que suivant certaines hypothèses de bruit, on peut obtenir la séparabilité des calculs du filtre de Kalman 2D dans l'espace de Fourier. Nous exploiterons cette séparabilité de manière à organiser les calculs en trois étapes: Transformées de Fourier sur les lignes et rotation de l'image; filtrage de Kalman 1D sur les colonnes et rotation de l'image; transformée de Fourier inverse sur les lignes.

Les paramètres Dp et C sont identiques aux précédents. Seul Tseq varie et vaut pour le filtre de Kalman  $K^3 \cdot I^2 \cdot Tkal$ , où K, égal à 3, représente la taille du modèle du flou de l'image et Tkal vaut 4,3μs. Pour la transformée de Fourier rapide  $Tseq = I^2 \cdot \log_2 I$  Tpap, avec Tpap = 18,6μs. Pour des données complexes et une image de taille 128x128, C devient quasiment constant lorsque N est grand et vaut:

- C = 0,058 pour le filtre de Kalman
- C = 0,104 pour la FFT



Suivant les tailles d'images à traiter, la saturation du réseau d'interconnexion apparaîtra en premier soit durant les calculs de la transformée de Fourier, soit durant le filtrage de Kalman. On peut déterminer le nombre de processeurs au delà duquel apparaît la saturation du réseau en anneau bidirectionnel reliant les transputers:

Taille de l'image I	32	64	128	256	512
Nsat pour FFT	54	65	76	87	98

Nsat représente la valeur de N, nombre de processeurs, à partir duquel le réseau est en saturation. Le réseau approprié doit être sélectionné en fonction du critère de saturation le plus sévère.

Les résultats prévisionnels de l'efficacité hors saturation pour une image 128x128 sur réseau de transputers interconnectés par le réseau en anneau sont:

=====

N processeurs	1	2	3	4	7	8	9	10
Temps idéal (s)	4,03	2,016	1,345	1,008	0,576	0,504	0,45	0,403
Temps prévu (s)	4,03	2,042	1,38	1,054	0,64	0,58	0,53	0,49
Efficacité en %	100	98,7	97,2	95,6	89,6	87,3	84,8	82,3

On constatera que les efficacités sont supérieures lors des calculs du filtrage de Kalman que lors des calculs de la transformée de Hadamard. Ceci était naturellement prévisible, puisque le taux de communication C de ce dernier algorithme est supérieur aux taux de communication pour le filtrage de Kalman.

**3 conclusion.**

Les exemples retenus pour nos démonstrations, bien que simples à paralléliser, illustrent bien l'importance de la mesure prévisionnelle du comportement des calculs parallèles de l'algorithme. L'ensemble des modèles et des émulations retenus sont relativement simples, et permettent une approche aisée de l'évaluation; le concepteur pourra ainsi élargir l'espace de recherche des solutions parallèles, à moindre coût.

Nous avons montré que les mesures d'efficacité bien connues dans le cas des machines SIMD sont adaptées aux machines MIMD; de plus la notion novatrice de saturation, et l'utilisation qui en est faite sur des algorithmes complexes est un guide déterminant vers le choix optimal d'un réseau d'interconnexion.

Les travaux menés actuellement visent à élargir les hypothèses sur les algorithmes, et à considérer d'autres modèles de machines MIMD.

**BIBLIOGRAPHIE:**

- [1] J. BIEMOND & all "A fast Kalman filter for images degraded by both blur and noise" IEEE Transaction on ASSP Vol31, 1983.
- [2] H. DUBOIS "Analyse des systèmes multiprocesseurs: application a la mise en oeuvre sous contraintes d'algorithmes de traitement d'images" Thèse de doctorat Université de Rennes I, 21-01-1991.
- [3] J.L. PHILIPPE & all "Evaluation of parallel structure dedicated to undersea image processing: application to the TOS projet" International workshop on algorithms, Pont à Mousson 10-16 juin 1990.
- [4] H. DUBOIS & all "Configurable architecture: a methodologic approach applied to a transputer based machine for image processing" Seventh Workshop on multidimensionnal signal processing, September 23-25 Lake Placid, New York.