



Parallélisation d'un algorithme de relaxation sur OPSILA

Laurent Kwiatkowski, Jean-Paul Stromboni

Université de Nice Sophia-Antipolis
Laboratoire de Signaux et Systèmes, Equipe de l'URA 1376 du CNRS
41, Boulevard Napoléon III, 06041 Nice Cedex

RÉSUMÉ

On analyse la parallélisation d'une méthode de relaxation sur le calculateur parallèle OPSILA à 16 processeurs, un réseau reconfigurable et deux modes parallèles SIMD et SPMD. Le traitement vectoriel est peu efficace car la matrice d'interconnexion est creuse. Deux schémas parallèles sont donc envisagés. L'un utilise le mode SPMD, une allocation de tâche devrait en améliorer la performance jusqu'à une limite idéale. L'autre, qui tire parti de la reconfigurabilité du réseau est supérieur dès la troisième itération.

ABSTRACT

The parallelization of a relaxation method is conducted on the parallel computer OPSILA, with 16 processors, a configurable network, and the two parallel modes SIMD and SPMD. Vector processing is not efficient due to sparse interconnection matrix. Two parallel scheme are then introduced, one using SPMD mode should be improved from proper task allocation. The other makes intensive use of network reconfiguration and denotes better efficiency after the third iteration.

Introduction

Les algorithmes connexionnistes constituent une catégorie dont la formulation suggère une parallélisation importante du traitement. Ces algorithmes sont décrits par l'interaction d'un ensemble de cellules généralement identiques, i.e. exécutant un traitement identique: les données à traiter sont contenues dans l'état initial des cellules, les résultats dans l'état final, et le programme de traitement est inscrit dans le réseau reliant les cellules et les influences de voisinage pondérées entre cellules voisines.

Le traitement d'un problème connexionniste peut être décrit par un graphe spécifiant l'ensemble des variables d'état du problème et leurs interactions. Ce graphe est régulier dans le cas des réseaux neuromimétiques. Les cellules sont entièrement connectées dans le réseau de Hopfield, ou connectées par blocs dans le réseau multicouche, ou encore selon un stencil, schéma local invariant, dans le réseau auto-organisateur de Kohonen. Le traitement cellulaire est une accumulation des influences du voisinage suivie de l'application d'une non linéarité décisionnelle dure ou douce.

Pour certaines méthodes d'optimisation telles que la classification par relaxation, le problème est décrit par un ensemble d'hypothèses à confirmer ou infirmer par application d'un jeu de contraintes de dépendance. Les sommets du graphe représentatif sont alors associés aux hypothèses et les arcs aux contraintes. La relaxation, application répétée des contraintes aux hypothèses initiales déterminées localement aboutit à un ensemble d'hypothèses améliorées tenant compte des dépendances.

L'algorithme de mise en correspondance d'images de cette étude appliqué par ses auteurs /1, 2/ au traitement de paires ou triplets stéréoscopiques en Vision Artificielle s'inscrit dans cette catégorie.

Présentation de l'algorithme

On trouve la notion de relaxation introduite dans la modélisation de phénomènes physiques, où un ensemble de processus élémentaires interagissent localement au cours de la recherche d'un état d'équilibre. Des méthodes de traitement de l'information s'en inspirent, dont le parallélisme potentiel réside dans le parallélisme du traitement des sommets, et dans le parallélisme des communications associées aux arcs. On peut ainsi espérer un traitement parallèle efficace des algorithmes correspondants, bien que la recherche d'un état d'équilibre, principe de la méthode, nécessite un nombre d'itérations de relaxation que l'utilisateur ne maîtrise pas et qui peut détériorer le temps d'exécution.

Tirer le meilleur parti du parallélisme des sommets et des arcs sur un calculateur parallèle n'est pas a priori un problème simple. Le schéma de parallélisation associant un noeud de traitement de la machine à un sommet du graphe et une voie de communication à un arc se heurte aux contraintes pratiques sur les dimensions de la machine et les caractéristiques de son réseau. Le nombre de noeuds n'est pas égal au nombre n de sommets, le nombre de voies de communication offertes par le réseau est fixé indépendamment du graphe irrégulier à traiter.



1 . Classification par relaxation

La méthode de classification par relaxation (formalisée cf. /3, 4/) considère n objets et m étiquettes ou classes. Le traitement se ramène à l'attribution d'une étiquette à chaque objet.

Pour cela, un vecteur étiquette p_i est associé à l'objet i , la composante $p_i(k)$ $k=1 \dots m$, mesurant la plausibilité de l'appartenance de i à la classe k . Un voisinage est attaché à l'objet i , ensemble des objets dont les classements ont une influence sur celui de i . Des relations de compatibilité des classements dans un voisinage sont alors exprimées au moyen de coefficients c_{ijk} dans l'intervalle $0, \dots, 1$ quantifiant (de 0 à 1) l'influence du classement de l'objet i dans la classe k sur le classement de j dans l .

A partir de la détermination initiale des n objets, de l'étiquetage p^0 de chacun et du réseau des compatibilités c_{ijkl} fondées sur le problème posé, la relaxation améliore l'étiquetage en faisant intervenir les relations de compatibilité. Pour cela, un second vecteur q_i est associé à l'objet i , qui contient le classement de i le plus cohérent avec l'état des voisins, calculé par accumulation des étiquettes pondérées par les coefficients de compatibilité. La recherche de l'étiquetage optimal est basée sur la comparaison des p_i et q_i . Ainsi pour la loi récurrente $p_i^{k+1} = q_i^k / |q_i^k|$ qui recherche le maximum de cohérence de l'étiquetage, produit scalaire $p_i \cdot q_i$. D'autres lois considèrent l'ambiguïté de l'étiquetage à l'aide d'une fonction critère entropie (cf. /1, 4/). La non-linéarité de la loi, contenue dans la normalisation, semble essentielle pour aboutir à un résultat pertinent.

La classification par relaxation apparaît analogue à plusieurs titres aux réseaux neuromimétiques, avec la loi cellulaire d'accumulation des influences, et la non-linéarité du traitement. Le réseau est cependant irrégulier et les cellules multivariées.

2 . Algorithme considéré (Berthod, Long /1, 2/)

Il s'agit de reconstruire l'information de profondeur à partir de vues (ici deux images D et G) réalisées depuis des points différents. Des primitives (ici segments) sont d'abord extraites de D et G .

La phase d'initialisation retient alors un ensemble d'objets, associations hypothétiques entre un segment de D et un segment de G , établit les deux réseaux des relations renforçatrices et inhibitrices entre ces associations, et détermine un étiquetage initial pour chaque objet, c'est à dire deux composantes $P(i)$ et $A(i)$ liées dans $0 \dots 1$ par une contrainte de normalisation et quantifiant respectivement la possibilité de présence et d'absence de l'association i .

La phase suivante de relaxation recherche un équilibre de l'étiquetage du graphe. On modifie séparément les $P(i)$ et $A(i)$ selon l'état des voisins renforçateurs et inhibiteurs, puis on normalise dans chaque objet le vecteur étiquette.

La phase finale, ou tri, décide selon l'équilibre atteint quelles seront les associations retenues. On en déduira la profondeur dans le champ.

La parallélisation de la seconde phase, qui est étudiée ici, n'est pas la plus coûteuse en temps. Sur l'ex-

emple considéré, la phase d'initialisation est de loin la plus longue.

Algorithme

On note P et A les vecteurs des $P(i)$ et $A(i)$ avec: n , nombre d'objets retenus, $nrenf(i)$, nombre de voisins renforçateurs de i , $ninh(i)$, nombre d'inhibiteurs, et $CPrij = (P^0(i) + P^0(j)) R_{ij}$, $CARij = (A^0(i) + A^0(j)) R_{ij}$, $CPIij = (P^0(i) + A^0(j)) I_{ij}$ et $CAIij = (A^0(i) + P^0(j)) I_{ij}$, R_{ij} , I_{ij} renforcement et inhibition entre i et j .
 $P := P^0$; $A := A^0$; /* initialisation de l'étiquetage */
 faire /* relaxation */

$P' := \alpha P$; $A' := \alpha A$; /* $\alpha = 10$ impose solution réelle */

de $i := 1$ jusqu'à n

de $j := 1$ jusqu'à $nrenf(i)$

$P'(i) := P(i) + CPrij A(j)$; $A'(i) := A(i) + CARij A(j)$;

de $j := 1$ jusqu'à $ninh(i)$

$P'(i) := P(i) + CPIij A(j)$; $A'(i) := A(i) + CAIij P(j)$;

de $i := 1$ jusqu'à n

$P''(i) := P'(i) / \sqrt{P'(i)^2 + A'(i)^2}$

$A''(i) := A'(i) / \sqrt{P'(i)^2 + A'(i)^2}$

si $MAX_i (|X''(i) - X(i)|) < \text{seuil}$ /* test arrêt ($X = P$ et A) */

alors sortir_faire /* fin de la relaxation */

sinon $P := P''$; $A := A''$;

fin_faire.

Parallélisation de l'algorithme

1 . Le calculateur parallèle

OPSILA a été conçu et expérimenté au LASSY à Nice (cf. /6/). Il comporte 16 processeurs élémentaires ou PE et connaît deux modes de calcul parallèle, le mode vectoriel SIMD et le mode MIMD contraint Single Program Multiple Data (SPMD) où les PEs exécutent simultanément un programme unique de façon indépendante et désynchronisée, les échanges entre les processeurs n'étant possibles qu'après l'exécution de ce programme par tous.

Le réseau d'interconnexion d'OPSILA relie les 16 bancs de la mémoire vectorielle aux 16 PE. Il est reconfigurable dynamiquement et reconstitue, en deux passages au besoin, toutes les permutations du réseau de Benes. Le processeur d'instruction PI distribue les instructions aux PE en SIMD et commande le réseau d'interconnexion, mais chaque PE dispose d'un séquenceur lui permettant d'exécuter un programme local en mode SPMD. Le processeur scalaire PS analyse les instructions du programme en cours. Il exécute les instructions scalaires, aiguille les instructions parallèles vers le PI et les PE. Pour programmer OPSILA, on utilise le langage évolué HELLENA, qui permet de piloter les modes parallèles et de contrôler le réseau d'interconnexion. La parallélisation de l'algorithme de relaxation utilise ainsi les primitives d'adressage indirect de la machine, Scatter et Gather cf. /6, 7/, qui diffusent les composantes d'un vecteur V_s dans un vecteur V_d comme spécifié dans un vecteur index I_x selon le schéma:

$V_d(I_x(i)) := V_s(i)$ ou $V_d(i) := V_s(I_x(i))$

2 . Analyse de la parallélisation

Lors d'une itération de relaxation, la mise à jour de l'étiquetage sous l'influence du voisinage peut être décrite par le produit d'une matrice d'influence par un vecteur regroupant toutes les étiquettes du problème (cf. /2/). Avec les notations précédentes, et en introduisant $V = (P(1) P(2) \dots P(n) A(1) A(2) \dots A(n))^T$, c'est $V_{\text{suivant}} := M \cdot V$ où la matrice M est composée de 4 blocs carrés $n \times n$ symétriques dont deux correspondent aux relations de renforcement, deux aux relations d'inhibition. Les termes généraux sont les C_{PRij} , C_{ARij} , C_{PIij} et C_{AIij} pour les blocs I, II, III et IV avec $C_{PRii} = C_{ARii} = \alpha$ et $C_{PIii} = C_{AIii} = 0$.

La forme résultante considère donc un graphe à $2n$ sommets où chaque objet initial donne deux sommets. Il faut cependant réunir les composantes $P(i)$ et $A(i)$ après le produit $M \cdot V$ pour la normalisation. A partir de cette formulation, différents schémas de parallélisation ont été envisagés selon la répartition des données dans les PE d'OPSILA.

3 . Implémentation sur OPSILA

Une première alternative est le traitement séquentiel de l'algorithme par le PS, implémenté pour évaluer l'accélération du traitement parallèle. M et le V sont rangés dans la mémoire scalaire, l'itération n'inclut pas d'échange de données, mais se traduit pour le PS par l'adressage de la mémoire scalaire pour accéder aux données.

Une seconde alternative est le traitement vectoriel SIMD du produit $M \cdot V$, où l'on utilise la faculté de la machine de découper automatiquement le vecteur en tranches de 16 composantes pour les traiter successivement. Cette solution est d'autant moins efficace que la matrice M est creuse. Dans l'exemple considéré, il y a 91 objets et 1034 liaisons répertoriées parmi les 8281 possibles. Un calcul du remplissage de M donne 7 %. On exécute alors 93% d'opérations inutiles, i.e. multiplications par zéro et accumulation.

Une autre possibilité proposée ici consiste à répartir les objets, un PE se chargeant de la mise à jour des étiquettes d'un ou de plusieurs objets selon le rapport du nombre de processeurs au nombre d'objets. Cette parallélisation "par noeud" correspond pour la matrice à une contraction selon les lignes pour éliminer les termes nuls, et impose la mémorisation du numéro de la colonne avant contraction, i.e. des numéros des objets du voisinage. Cela revient à ajouter une matrice M_I d'index à la matrice M des valeurs contractées. Cette méthode a déjà été utilisée sur OPSILA (cf. /7/) pour les problèmes d'éléments finis. Pour utiliser le vecteur M_I , le principe de rangement des vecteurs dans la mémoire impose de compléter les lignes contractées jusqu'au nombre maximum de liaisons pour un sommet. L'étude /7/ conclut que compléter avec le numéro du PE équilibre les communications.

Dans l'exemple traité, le nombre de liaisons varie de 1 à 32 selon le sommet. Cependant, selon le rapport entre la dimension de M et le nombre de PE, 11 à 12 ici, on pourra tirer parti du mode SPMD lors du calcul après l'échange des données, chaque PE traitant

successivement et de façon autonome les liaisons des 11 ou 12 objets qui lui sont confiés.

L'échange utilisera les primitives d'adressage indirect et la configurabilité dynamique du réseau. Au contraire du cas suivant, le réseau n'est configuré qu'une fois durant les itérations, et selon un calcul effectué par OPSILA à partir de M_I .

La dernière alternative mise en oeuvre consiste à répartir les liaisons entre les PE. Cette parallélisation "par liens" aboutit à une machine dimensionnée en nombre de liaisons traitées, plutôt qu'en nombre de sommets du problème traité. Cela revient à contracter M aussi selon les colonnes et donc à rajouter un vecteur d'index M_Z pour les numéros de ligne. La diffusion est réalisée comme précédemment par adressage indirect sur le réseau à partir du vecteur M_I . Le calcul des influences est un produit de vecteurs SIMD. L'accumulation des influences par zones correspondant aux lignes de M peut être parallélisée, compte tenu de l'associativité de l'addition, en $\log_2 d$ additions binaires, si la zone la plus étendue comporte d liaisons. Cette accumulation par zones se traduit durant chaque itération par $\log_2 d$ configurations successives du réseau d'OPSILA, dont le calcul à partir du vecteur M_Z prend du temps et détériore la performance de la première itération, mais n'est pas à refaire pour les itérations suivantes.

Les composantes sont d'abord accumulées vers l'origine des zones, puis réunies dans le vecteur étiquette V à travers le réseau. Dans l'exemple, le maximum de 32 liaisons impose 5 configurations successives et additions, bien qu'un sommet au moins, pourvu d'un voisin unique, ne nécessite pas d'accumulation.

4 . Mesure des performances

Pour évaluer l'accélération due au traitement parallèle de la relaxation, on utilise l'horloge programmable d'OPSILA. Sur l'exemple précédent, on chronomètre une itération comprenant l'échange de données, l'accumulation des influences et la normalisation des étiquettes, mais pas le test d'arrêt. La comparaison de la version séquentielle (utilisant une matrice M contractée) et des versions parallèles aboutit aux résultats suivants:

Version	Temps	Accélération	Efficacité
séquentielle	1451	1	6,25 %
SIMD	275	5,28	32 %
SPMD par noeud	129	11,25	70 %
par liens itération 1	157	9,24	57,7 %
itération > 1	107	13,5	85 %

Discussion des résultats et conclusion

- Le schéma parallèle par noeuds est un exemple d'utilisation du mode SPMD. Le schéma par liens n'utilise que le SIMD mais repose plus encore sur la reconfigurabilité du réseau et l'adressage indirect d'OPSI-



LA pour effectuer l'accumulation finale. La performance du schéma par noeud est supérieure pour un nombre d'itérations restreint du fait de la charge supplémentaire de traitement lors de la première itération avec le schéma par liens. Les résultats précédents montrent qu'au delà de la 3ème itération le schéma par liens dont l'efficacité tend vers 85 % avec le nombre d'itérations devient le plus efficace:

$129N - (157 + 107(N-1)) = 22N - 50 > 0$ pour $N > 2$.
D'autre part, l'efficacité de la parallélisation de ces deux schémas est deux et trois fois plus grande que le produit SIMD de la matrice creuse M.

• L'amélioration de performance repose principalement sur un allègement du coût des communications (préparation et exécution). C'est d'abord l'optimisation de l'architecture du réseau, de sa bande passante, du contrôle par le PI et de la compilation et du séquençement des échanges.

Des schémas parallèles intermédiaires sont envisageables pour équilibrer les échanges en homogénéisant le graphe représentatif du traitement, qui consistent à éclater les sommets reliés à un trop grand nombre de voisins. On augmente la dimension et le nombre de termes non nuls de la matrice d'interconnexion pour améliorer l'homogénéité. Un compromis apparaît entre l'équilibrage de l'échange et le nombre d'échanges à effectuer.

Enfin, l'allocation des objets aux PE influe sur l'équilibrage des communications: ici, la répartition a été opérée séquentiellement en allouant l'objet suivant au PE suivant (modulo le nombre de processeurs 16). L'évaluation du nombre de liens alloués à chaque processeur après répartition suivant cette règle aboutit à un maximum de 109 liaisons pour le PE n° 1 et un minimum de 37 pour le PE n° 13 alors qu'une répartition égale des 1034 liaisons du problème en donne 64 par PE. On peut donc espérer à l'aide d'une règle de répartition judicieuse améliorer au maximum la performance actuelle d'un facteur limite $109/64 = 1,7$.

Remerciements à Marc Berthod, Gérard Giraudon et Pascale Limozin-Long, à l'équipe PASTIS de l'INRIA à Sophia-Antipolis pour leur disponibilité, l'accès à l'algorithme, aux stations, et aux données stéréo.

Bibliographie

/1/ Scene Labeling: an optimization approach, O. Faugeras, M. Berthod dans Pattern Recognition, vol. 12, pp 339- 347, Pergamon, 1980
/2/ Vision Stéréoscopique appliquée à la Robotique, Thèse de l'Université de Nice, Octobre 1986, Pascale Limozin-Long
/3/ Scene Labeling by Relaxation Operations, IEEE Trans. SMC-6, n°6, June 1976, pp 420-433, by Rosenfeld, Hummel, Zuckert
/4/ Utilisation de la relaxation pour la mise en correspondance des segments d'une carte et d'une image aérienne, Salmeron E. et Milgram M., 2ème colloque image, GRETSI, Nice, Avril 1986,32-38

/5/ Squelettisation à partir de contours: application à la mise en correspondance de régions,

F. Pepin, M. Milgram, COGNITIVA 87, pp 23-28

/6/ Experiments on a parallel SIMD/SPMD architecture and its programming, F. Boeri, M. Auguin France Japan Artificial Intelligence and Computer Science Symposium 87, 9-11 Nov. 1987, 385-411

/7/ Vectorisation de la méthode des éléments finis sur un ordinateur SIMD/SPMD, Karl Schubert, Thèse Docteur Ingénieur de l'Université de Nice, Oct. 1987.

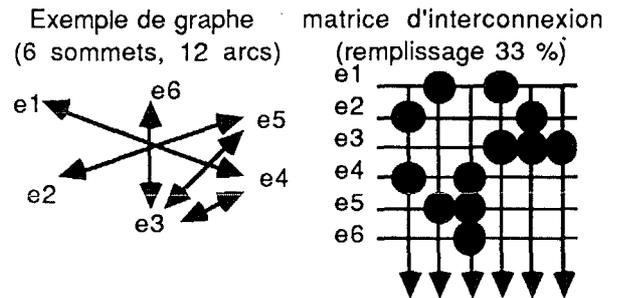


Schéma parallèle "par noeuds"

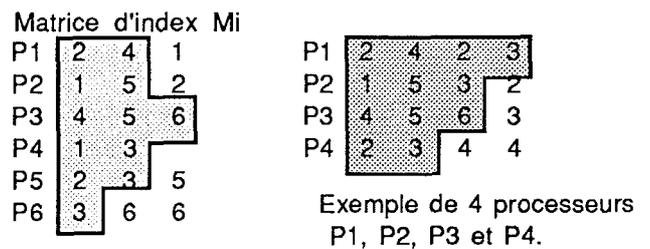


Schéma parallèle "par liens"

