



Traitement, Synthèse, Technologie et Applications

BIARRITZ - Mai 1984 -

STRUCTURES PARALLELES EN TRAITEMENT D'IMAGES

PARALLEL STRUCTURES IN IMAGE PROCESSING

J.L. BASILLE, S. CASTAN, J.Y. LATIL

Laboratoire CERFIA - Université P. Sabatier - TOULOUSE

RESUME

Le Traitement d'Images est sans aucun doute un domaine privilégié pour l'étude et la mise en oeuvre du parallélisme. Mais, comme toute discipline, le calcul parallèle implique des contraintes et présente des limites. De façon à fournir des éléments de réponse aux questions quand, comment et que paralléliser, nous présentons une approche théorique des limites de structures M.I.M.D. en Traitement d'Images. Nous donnons également des résultats de simulation du projet SY.MP.A.T.I. (SYstème MultiProcesseur Adapté au Traitement d'Images), un système à deux niveaux de parallélisme. Le premier niveau est de type S.I.M.D. permettant d'effectuer des traitements locaux. Le second niveau est de type M.I.M.D., plus particulièrement orienté vers les traitements de régions.

Ces résultats théoriques et pratiques concordants, permettent de mieux aborder les problèmes de choix d'une structure parallèle pour un algorithme donné, en particulier dans le cas MIMD/SIMD, ainsi que les problèmes de finesse de décomposition des algorithmes sur une structure MIMD.

Le système d'exploitation mis en oeuvre dans la simulation comporte une stratégie d'allocation des processeurs basée sur l'arbre dynamique d'enchaînement des processus de façon à prendre en compte le comportement de ceux-ci. Les processus sont répartis en deux classes : les processus de calcul et les processus d'enchaînement. Les priorités sont établies par un algorithme fondé à la fois sur le principe de l'algorithme horizontal et sur le principe de l'algorithme vertical, c'est-à-dire qu'il conserve un taux de parallélisme des processeurs satisfaisant tout en favorisant les échanges entre les processus et sans risque de blocage.

SUMMARY

Image Processing is undoubtedly a particularly well fitted field for studying and practising parallelism. But, as every discipline, parallel computing has constraints and limits. In order to give elements for an answer to the question how to parallelise, we present here a theoretical approach of M.I.M.D. structure limits in Image Processing. We also give simulation results of the SY.MP.A.T.I. project, a two level parallel system. The first level is of multiple S.I.M.D. type, and it makes it possible to process plain treatments at memory level. The second level is of M.I.M.D. type, which is much more adapted to region level processing. These concordant theoretical and practical results provide elements for an answer to the problems of choosing a parallel structure for a given algorithm, especially in the MIMD/SIMD case, as well as an answer to the problems of accurate splitting of algorithms on a MIMD structure.

The operating system involved in the simulation makes use of a resource allocation strategy based on the dynamic tree of the requests, in order to take into account the behaviour of the processes. The processes are split into two classes : the scheduling class and the computing class. The priority philosophy is based on an algorithm which keeps both the advantage of the horizontal algorithm and the advantage of the vertical algorithm, i.e. keeps a good parallel rate of the processors while making the exchanges between the processes easy, and without possible deadlock.



STRUCTURES PARALLELES EN TRAITEMENT D'IMAGES

PARALLEL STRUCTURES IN IMAGE PROCESSING

J.L. BASILLE, S. CASTAN, J.Y. LATIL

1. INTRODUCTION

Domaine privilégié pour l'étude et la mise en oeuvre du parallélisme, le Traitement d'Images met en évidence deux niveaux de calculs parallélisables : les traitements concernant chaque point et son voisinage, et les traitements concernant les différentes régions constituant l'image.

Pour aider à prendre en compte ce constat dans le choix entre les multiples structures parallèles envisageables, nous avons effectué une approche théorique des limites des structures de type M.I.M.D. en Traitement d'Images. Les résultats de cette étude sont confirmés par les résultats de simulation du niveau M.I.M.D. du projet SY.MP.A.T.I. (SYstème MultiProcesseur Adapté au Traitement d'Images).

D'autre part, nous avons développé et mis en oeuvre dans cette simulation, un système d'exploitation dans lequel la politique d'allocation des processus est basée sur l'arbre dynamique d'enchaînement des processus.

2. LES LIMITES DES STRUCTURES MULTI-PROCESSEURS

2.1. APPROCHE THEORIQUE

Les structures M.I.M.D. sont, par définition, les structures les plus générales et les plus souples. Mais, qu'en est-il de leur efficacité ? L'une des caractéristiques principales d'un tel système est le mode d'interconnexion des processeurs et des mémoires, et un système d'interconnexion s'évalue essentiellement en termes de débit d'information. Nous allons donc étudier, dans ce paragraphe l'effet sur les performances, du temps de transfert des données comparé au temps d'exécution, ainsi que l'effet du nombre de processeurs mis en oeuvre.

Parmi ces structures, nous nous intéressons ici aux architectures M.I.M.D. composées de processeurs banalisés. De tels systèmes sont particulièrement bien adaptés aux traitements par régions. Les différentes régions constituant l'image doivent donc être envoyées vers les processeurs afin d'être traitées. Le système d'interconnexion entre les processeurs et les bancs de mémoire revêt donc une importance certaine du double fait de la quantité d'information transmise et de la banalisation des processeurs. Néanmoins, il n'est pas essentiel d'étudier toutes les possibilités d'interconnexion ou de configuration générales. En effet, il est possible, sans restreindre la généralité de l'étude, de n'étudier que le cas du bus unique, toute autre configuration pouvant être considérée, du point de vue des caractéristiques de transfert, comme un bus unique plus ou moins performant.

La segmentation de l'image en régions peut être effectuée d'une façon ou d'une autre, la seule chose qui importe ici est la capacité mémoire de chaque processeur. Il est donc possible, là encore sans restreindre la généralité du problème, de considérer

que les régions sont toutes de même dimension.

Pour étudier la variation du temps total de traitement de l'image en fonction du temps d'exécution d'une région et du nombre de processeurs, nous devons considérer les paramètres suivants :

NR : nombre de régions de segmentation de l'image,
NP : nombre de processeurs,
TP : temps de traitement pour une région,
TD : temps de transfert d'une région,
TT : temps total de traitement de l'image,

les hypothèses de travail étant les suivantes :

1. les NR régions sont identiques en taille,
2. Les régions subissent des traitements semblables,
3. Le système d'interconnexion reliant les processeurs et les mémoires est réduit à un bus unique,
4. Un processeur travaille sur une seule région à la fois,
5. NR NP.

Le temps total TT est alors donné par :

Si $TP/TD \geq NP-1$

$TT = [(NP+1).TD+TP].NR/NP$ si $NR=K.NP$
 $TT = (K+1).TP + [K.(NP+1)+L+1].TD$ si $NR=K.NP+L$
avec $0 < L < NP$

Si $TP/TD < NP-1$

$TT = 2.NR.TD$ si $NR=K.NP$
 $TT = (2.K.NP+L+1).TD+TP$ si $NR=K.NP+L$
avec $0 < L < NP$
et si $TP/TD \geq L-1$
 $TT = 2.NR.TD$ si $NR=K.NP+L$
avec $0 < L < NP$
et si $TP/TD < L-1$

La variation du temps total de traitement de l'image en fonction du rapport TP/TD a alors l'allure présentée sur la figure 1. Le temps total TT est exprimé par rapport à TP. Le rapport TP/TD du temps de traitement d'une région sur le temps de transfert permet d'exprimer une certaine notion de la complexité du traitement appliqué à chaque région. Ce rapport doit donc être suffisamment grand, c'est-à-dire le traitement doit être suffisamment complexe pour chaque région, pour que le gain obtenu soit significatif.

Si nous considérons maintenant la variation du temps total TT, toujours ramené à TP, en fonction du nombre de processeurs, l'algorithme étant donné donc TP/TD fixé pour un certain système d'interconnexion, nous obtenons des courbes qui ont l'allure présentée sur la figure 2.

Le nombre de processeurs mis en oeuvre doit donc être assez grand mais est de toute façon limité par $(TP/TD)+1$. Et quand cette limite $(TP/TD)+1$ est grande, conformément à la conclusion précédente, il n'est pas intéressant d'augmenter considérablement le nombre de processeurs car le gain obtenu n'est plus significatif.

STRUCTURES PARALLELES EN TRAITEMENT D'IMAGES

PARALLEL STRUCTURES IN IMAGE PROCESSING
 J.L. BASILLE, S. CASTAN, J.Y. LATIL

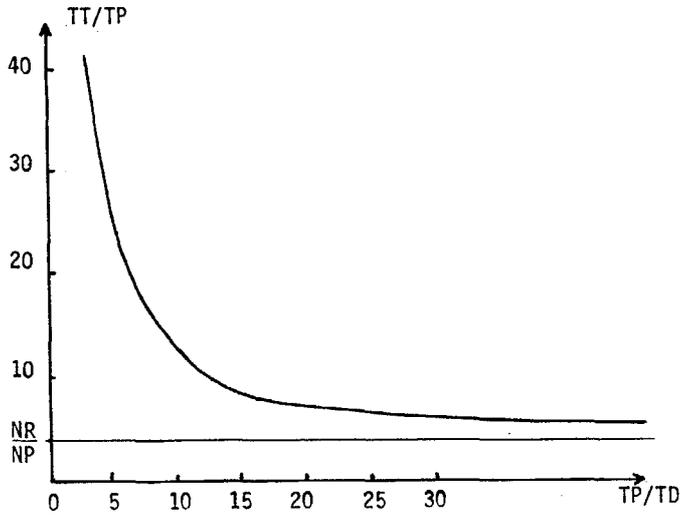


Figure 1

Variation du temps total de traitement en fonction du rapport TP/TD

2.2. RESULTATS DE SIMULATION

La simulation du niveau M.I.M.D. du système SY.MP.A.T.I. que nous développons actuellement nous a permis de confirmer les résultats obtenus dans l'approche théorique précédente.

La figure 3 donne deux exemples de résultats de simulation montrant la variation du temps total de traitement, ramené à TP, en fonction du nombre de processeurs. Nous retrouvons bien la même allure qu'avec les courbes de la figure 2. Nous retrouvons également en 3.b les ruptures dues aux restes de la division de NR par NP et qui étaient déjà visibles sur la figure 2.

Notons que la politique d'allocation du bus est, bien sûr, plus élaborée dans la simulation que l'hypothèse de périodicité admise, dans l'approche ci-dessus, pour des raisons de simplification. De plus, le temps d'exécution du système d'exploitation lui-même est pris en compte dans la simulation alors qu'il est négligé dans la présente étude.

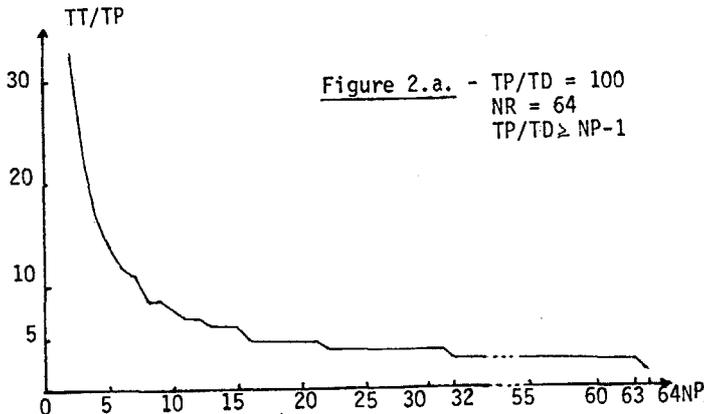


Figure 2.a. - TP/TD = 100
 NR = 64
 TP/TD ≥ NP-1

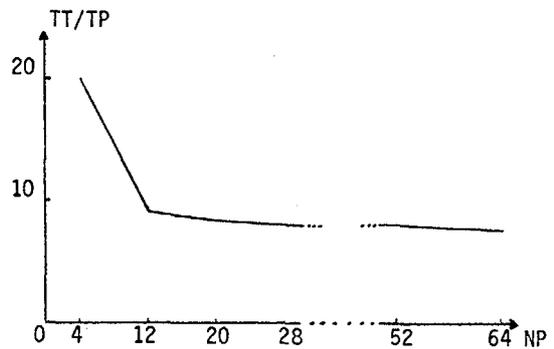


Figure 3.a - TP/TD = 12
 NR = 64

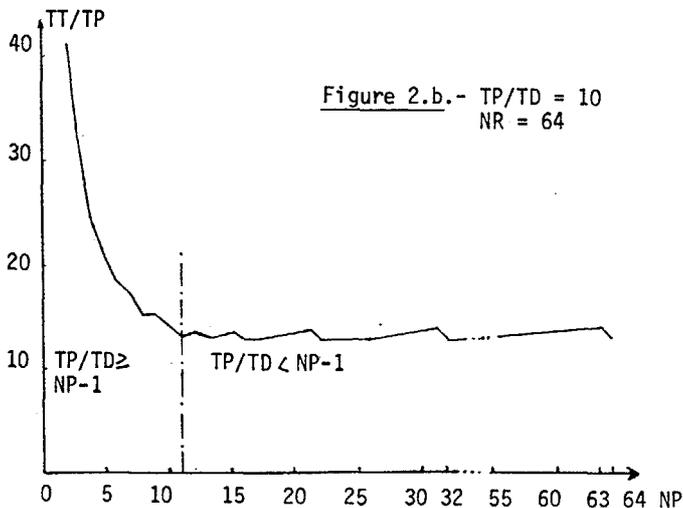


Figure 2.b.- TP/TD = 10
 NR = 64

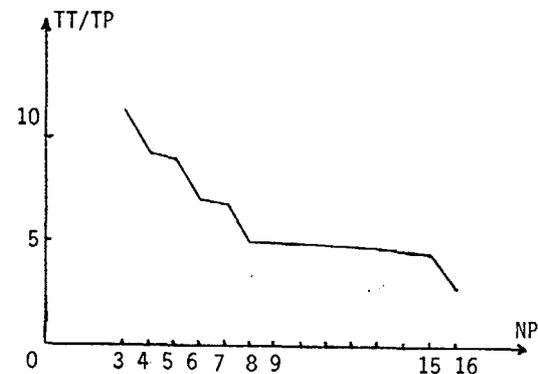


Figure 3.b - TP/TD = 50
 NR = 16

Figure 2

Variation du temps total de traitement en fonction du nombre de processeurs.

Figure 3 - Résultats de simulation



STRUCTURES PARALLELES EN TRAITEMENT D'IMAGES

PARALLEL STRUCTURES IN IMAGE PROCESSING

J.L. BASILLE, S. CASTAN, J.Y. LATIL

Néanmoins, les résultats concordants de la simulation et de l'approche théorique permettent de dégager deux conclusions selon que l'algorithme ou le nombre de processeurs est fixé.

Si l'algorithme est fixé, le nombre de processeurs peut être augmenté jusqu'à une valeur qui doit, de toute façon, rester inférieure au rapport TP/TD et qui, en pratique, même si ce rapport est grand, se situera, dans le cadre de nos hypothèses, autour d'une vingtaine de processeurs.

Si le nombre NP de processeurs est fixé, alors les algorithmes profiteront au maximum du parallélisme offert par cette structure s'ils sont tels que leur rapport TP/TD est grand et en particulier supérieur à NP-1.

3. LE PROJET SY.MP.A.T.I.

3.1. PRESENTATION GENERALE

La structure du système SY.MP.A.T.I. a été conçue de façon à permettre plusieurs approches du parallélisme. Nous avons pour cela bâti cette structure autour de deux bus (figure 4) :

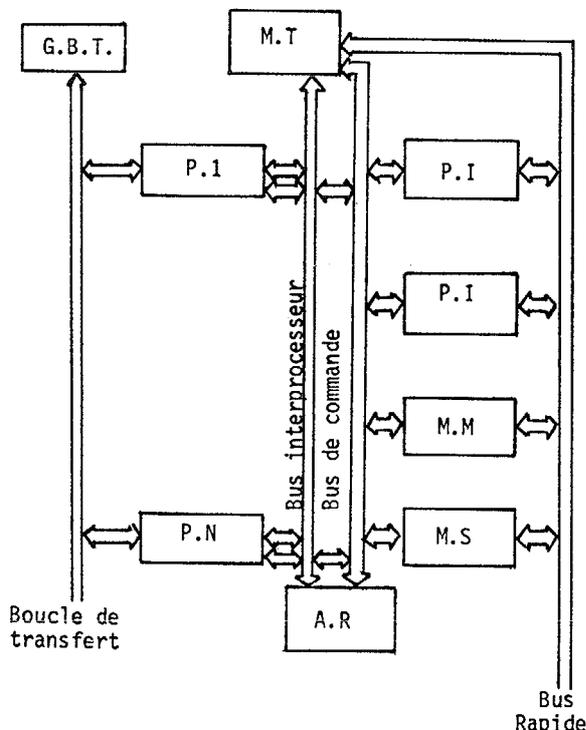


Figure 4

Structure générale du projet SY.MP.A.T.I.

G.B.T.: gestionnaire de la boucle de transfert,
 P.I : processeur i,
 M.T : module de transfert,
 A.R. : allocateur de ressources,
 P.1 : processeur image de type S.I.M.D.,
 M.M. : module masque,
 M.S. : modules spécialisés.

- des processeurs banalisés sont reliés par l'intermédiaire d'un bus interprocesseur, ce qui permet d'effectuer des traitements parallèles de façon M.I.M.D.,

- le deuxième bus est un bus rapide qui permet les transferts à la cadence vidéo. On peut relier sur ce bus, outre des modules spécialisés, un ou plusieurs processeurs d'image dans lesquels l'information image est stockée et peut être traitée de façon S.I.M.D.

Ces deux bus sont reliés l'un à l'autre par un module de transfert. L'ensemble du système est géré par un allocateur de ressources.

3.2. LE SYSTEME D'EXPLOITATION

Parmi les tâches de l'allocateur de ressources, l'attribution d'un processeur à un processus est particulièrement importante et délicate. Quel processus choisir ? Quels sont les risques de blocage ? Le taux de parallélisme obtenu est-il satisfaisant ? Autant de questions qui nous ont guidés dans notre choix.

3.2.1. Algorithme de "sous-arbre final"

3.2.1.1. Principe

Si nous considérons l'arbre dynamique d'enchaînement des processus dans lequel chaque noeud i représente un processus P_i et chaque arc ij la relation de création du processus-fils P_j par le processus-père P_i , plusieurs stratégies d'ordonnement des processus sont possibles. Deux méthodes se dégagent a priori :

- l'algorithme "horizontal" qui consiste à attribuer une haute priorité aux processus feuilles. Cet algorithme est basé sur le fait que ces processus, à l'instant considéré, ne sont pas encore des processus-pères. Or les paramètres nécessaires à l'exécution de ces processus-feuilles sont transmis au moment de leur création. Donc ces processus sont activables et un parallélisme élevé est possible avec cet algorithme. Mais les envois de résultats d'un fils vers son père sont défavorisés, le père n'étant pas nécessairement actif.

- l'algorithme "vertical" favorise au contraire ces échanges fils-père mais, de ce fait, laisse de nombreux processeurs occupés par des processus en attente des résultats de leurs fils. Il conduit donc à un taux de parallélisme beaucoup plus faible.

Nous avons donc conçu un algorithme [AL ROZZ] cherchant à satisfaire chacun des critères de base des algorithmes précédents. Cet algorithme s'appuie sur la notion de sous-arbre final. Un tel sous-arbre a pour racine un noeud dont tous les fils sont des feuilles. En attribuant une forte priorité aux processus appartenant à un sous-arbre final, cet algorithme permet, d'une part, d'obtenir un bon taux de parallélisme comme avec l'algorithme "horizontal" puisque la plupart des processus des sous-arbres finaux, sont des

STRUCTURES PARALLELES EN TRAITEMENT D'IMAGES

PARALLEL STRUCTURES IN IMAGE PROCESSING

J.L. BASILLE, S. CASTAN, J.Y. LATIL

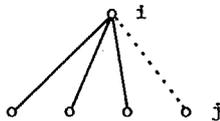
processus-feuilles, et permet, d'autre part, une exploitation rapide des résultats puisque les pères de ces processus ont également une forte priorité.

3.2.1.2. Algorithme

Le changement des priorités des processus se fera à chaque création de processus. Si *i* est le processus-père et *j* le processus créé, nous avons alors à considérer les 4 cas exhaustifs suivants :

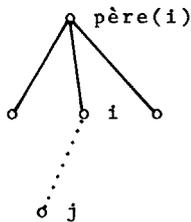
1. *i* est racine d'un sous-arbre final.

j est alors un processus-feuille supplémentaire dans le sous-arbre final de racine *i* et il sera le moins prioritaire parmi ses frères, le rapport des autres priorités ne changeant pas :



2. *i* est processus-feuille d'un sous-arbre final.

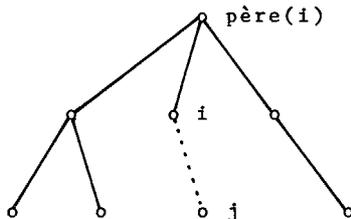
la création de *j* détruit donc le sous-arbre final de racine père (*i*) :



donc père(*i*) doit devenir moins prioritaire que certains processus-feuilles qui étaient moins prioritaires que lui.

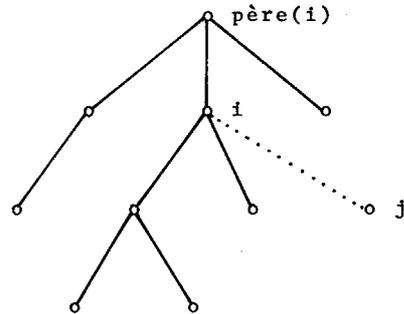
3. *i* est processus-feuille mais n'appartient pas à un sous-arbre final.

Le sous-arbre final de racine *i* est donc le plus récent sous-arbre final et il doit être moins prioritaire que tous les autres sous-arbres finaux et que tous les processus-feuilles n'appartenant pas à des sous-arbres finaux :



4. *i* est ni racine d'un sous-arbre final ni processus-feuille.

j est donc le dernier processus-feuille créé, n'appartenant pas à un sous-arbre final. Il doit donc être moins prioritaire que tous les sous-arbres finaux et que tous les processus-feuilles n'appartenant pas à des sous-arbres finaux :



3.2.2. Processus de calcul et processus d'enchaînement

Afin d'éviter les risques de blocage, nous avons été amenés à distinguer deux types de processus :

- les processus de calcul qui travaillent sur des données-image et ne lancent pas d'autres processus. Ces processus seront nécessairement des feuilles dans l'arbre dynamique d'enchaînement, donc seront dotés d'une priorité permettant de les privilégier plus ou moins.

- les processus d'enchaînement dont le rôle est de demander l'exécution des processus de calcul et d'explicitier le parallélisme et la synchronisation entre ceux-ci. Ces processus ne travaillent pas sur des données-image et ne sont donc pas privilégiés.

Il est aisé de voir que tout traitement peut s'écrire en n'utilisant que des processus de l'un ou l'autre de ces deux types. Une telle écriture permet de privilégier les traitements sur des données-image c'est-à-dire les traitements les plus coûteux et donc de fournir un taux de parallélisme le plus satisfaisant possible sans risquer de tomber sur un cas de blocage. En effet, un processus de calcul ne peut être en attente que sur une requête d'information-image. Une telle requête sera nécessairement satisfaite donc un processus de calcul s'achèvera dans tous les cas. Quant aux processus d'enchaînement, ils ne peuvent être en attente que sur des résultats de processus-fils ; or ceux-ci s'achèvent nécessairement, donc tout risque de blocage est levé.

4. CONCLUSION

A la question du choix du parallélisme, la réponse ne peut être simple sauf si on se limite à un type d'application très précis. Dans le cadre du traitement d'images, une réponse multiple, tenant compte des différents niveaux de parallélisme rencontrés, doit s'imposer, respectant les limites des diverses structures parallèles proposées.

Dans le cas des structures M.I.M.D. nous venons de voir l'importance du rapport TP/TD qui, pour un système donné, fournit un outil simple de mesure de la complexité d'un algorithme. Un tel système sera utilisé avec d'autant plus d'efficacité que les algorithmes auront un rapport TP/TD élevé. Ceci confirme l'idée intuitive que



STRUCTURES PARALLELES EN TRAITEMENT D'IMAGES

PARALLEL STRUCTURES IN IMAGE PROCESSING

J.L. BASILLE, S. CASTAN, J.Y. LATIL

de telles structures sont plus particulièrement orientées vers les traitements de régions. D'autre part, c'est un élément qui peut permettre de mieux aborder le problème de la finesse de décomposition.

Enfin, une écriture des traitements basée sur l'utilisation de processus de calcul et de processus d'enchaînement, permet, grâce à un système d'exploitation fondé sur l'arbre dynamique d'enchaînement des processus, d'obtenir un taux de parallélisme satisfaisant sans risque de blocage. Les résultats de simulation du niveau M.I.M.D. du projet SY.MP.A.T.I. [AL ROZZ] viennent corroborer ces conclusions.

5. BIBLIOGRAPHIE

- AL ROZZ, M. (1983). Système d'exploitation d'une structure multi-processeur adaptée au traitement d'images. Thèse de docteur-ingénieur. Toulouse.
- BARNES, G.H. et al. (1968). The ILLIAC IV computer. IEEE Trans. Comp. C-17, 746.
- BASILLE, J.L., CASTAN S. and LATIL J.Y. (1981). Système multi-processeur adapté au traitement d'images. In Languages and Architectures for Image Processing (M.J.B. Duff and S. Levialdi, eds). Academic Press, London.
- BASILLE, J.L., CASTAN, S. and LATIL, J.Y. (1981). A two-level parallel structure. SY.MP.A.T.I. Application to chromosome analysis. IVth European Chromosome Analysis Workshop Edinburgh, Scotland.
- BASILLE, J.L., CASTAN, S., DELRES, B. and LATIL, J.Y. (1982). A typical propagation algorithm on the line processor SY.MP.A.T.I. The region labelling. In Multi-computers and Image Processing (K. Preston, Jr and L. Uhr, eds). Academic Press, New York.
- COMTE, D. and DURRIEU, G. (1975). Techniques et exploitations de l'assignation unique. Contrat SESORI 75.167.
- CORDELLA, L., DUFF, M.J.B. and LEVIALDI, S. (1967). Comparing sequential and parallel processing of pictures. Proc. 3rd IJCPR, Coronado, Ca., pp. 707.
- DUFF, M.J.B., WATSON, D.M. and DEUTSH, E.S. (1974). A parallel computer for array processing. Proc. IFIP Congress, Stockholm, Sweden, pp. 94-97.
- DUFF, M.J.B. (1978). Parallel processing techniques. In Pattern Recognition. Ideas in Practise B.G. Batchelor, ED. (Plenum Press, New York).
- FISHBURN, J.P. (1981). Analysis of speedup in distributed algorithms. Computer Sci. Dept. Tech. Rep.431. University of Wisconsin.
- FLYNN, M.J. (1972). Some Computer Organizations and their Effectiveness. IEEE Trans. on Comp., Vol. C-21, n° 9, pp. 11-23.
- HAYNES, L.S., LAU, R.L., SIEWIOREK, D.P. and MIZELL, D.W. (1982) A survey of highly parallel computing. IEEE Trans. Comput. January.
- LENFANT, J. (1982). Mémoires parallèles et réseaux d'inter-connexion. Technique et Science Informatiques., Vol. 1 n°2, RAIRO.
- LEVIALDI, S., MAGGIOLLO-SCHETTINI, A., NAPOLI, M. and UCCELLA, G (1979). Considerations on parallel machines and their languages. In Multicomputers and Image Processing (K.Preston Jr. and L. Uhr, eds). Academic Press. New York.
- LISTER, A.M. (1977). Principes fondamentaux de systèmes d'exploitation. Eyrolles, Paris.
- MADNICK, S.E. (1968). Multi-processor software lockout, in Proc. 1968 Ass. Comput. Mach. Nat. Conf., pp 19-24.
- SHAPIRO, D. (1978). Theoretical limitations on the efficient use of parallel memories. IEEE Trans. Comput. C-27, 421-428.
- TIMSIT, C. and BOUDAREL, R. (1977). PROPAL II : une nouvelle architecture de calculateur adaptée au traitement du signal. Colloque National sur le traitement du signal et ses applications, Nice.