Prédiction des performances des machines parallèles spécialisées. Application à la simulation en temps réel des Perceptrons Multi-Couches à Champs d'Activation Locaux

Bertrand GRANADO et Patrick GARDA

Laboratoire des Instruments et Systèmes
Université Pierre et Marie Curie
Case 252 4, place Jussieu - 75252 Paris Cedex 05
Téléphone: 01.44.27.75.07 - Télécopie: 01.44.27.75.09
Courrier Electronique:Bertrand.Granado@lis.jussieu.fr,Patrick.Garda@lis.jussieu.fr

RÉSUMÉ

Dans cet article, nous proposons une méthode originale de prédiction des performances des machines parallèles spécialisées, que nous appliquons à la simulation en temps réel des Perceptrons Multi-Couches à Champs d'Activation Locaux. Ce type de Perceptrons est très largement utilisé dans le domaine de la reconnaissance de caractères. Nous démontrons que pour ces réseaux les performances obtenues par une machine parallèle spécialisée, la machine CNAPS, sont très inférieures à ses performances crêtes, mais qu'elles sont très proches des performances obtenues avec des ASICs. Nous montrons aussi que notre méthode de prédiction permet de prévoir le gain obtenu grâce à des modifications de l'architecture de la machine évaluée.

ABSTRACT

In this paper we propose a new methodology to predict the time simulation of parallel specilized computers. We use this methodology for the simulation of the MLPs with receptive fields used for handwritten character recognition. Our measures show that the effective performances obtain by the CNAPS neurocomputer are far below the peak performances or the performances achieved for fully connected MLPs, but they are close to those of direct hardware implementations and that they can even be improved.

Introduction La simulation en temps réel de grands réseaux connexionnistes, tel que les Perceptrons Multi-Couches à Champs d'Activation Locaux (PMC-CAL), nécessite le développement d'ordinateurs paralléles spécialisés ou d'ASICs. Le but de cet article est d'introduire une méthode d'évaluation et de prédiction des performances des machines parallèles spécialisées pour la simulation de PMC-CAL. Cette méthode sert à élaborer des éléments de comparaison entre différentes architectures dédiées ou généralistes, et permet une mesure du gain en performance obtenu par des modifications architecturales sur des machines spécialisées.

Simulation de Réseaux Nous avons choisi les PMC-CAL, qui sont des PMC combinant des couches 1-D et des couches 2-D, parce que ce type de réseaux permet une réduction considérable du nombre de connexions. Ils obtiennent de bons résultats en combinant des informations locales dans les couches cachées, et en regroupant ces informations à l'aide de connexions complètes dans les couches de sortie. Ces réseaux ont été utilisés avec succès pour la reconnaissance de caractères.

La machine CNAPS La méthode a été validé sur la machine CNAPS [1, 7, 6, 10] qui est un machine programmable en langage C-parallèle et en assembleur. Elle est constituée d'un tableau mono-dimensionnel (figure 1) de processeurs appelés Processeurs Noeuds (PN). L'ensemble des PN est contrôlé par un séquenceur externe.

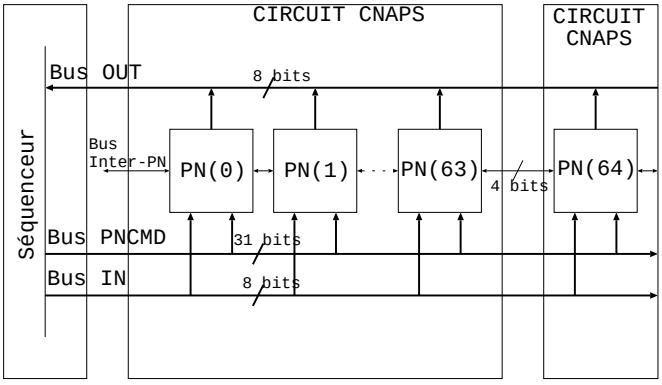

FIG. 1 — Architecture de CNAPS

Les processeurs sont reliés entre eux par deux bus, un bus à diffusion et un bus en anneau bidirectionnel. Le bus à diffusion est composé de deux entités IN et OUT et le bus en anneau est composé d'une entité Inter-PN. IN et OUT ont huit bits de large et servent respectivement pour la diffusion des données du séquenceur vers les PNs, et pour le transfert de données d'un PN vers le séquenceur. Inter-PN a quatre bits de

large, deux bits dans chaque sens.
Les PN ne travaillent qu'en arithmétique entière ou en virgule-fixe. Chaque processeur possède 4 kilo-Octets de mémoire propre. La machine qui a été utilisée pour la simulation possède 128 processeurs, elle est cadencée à 20 Mhz, et elle a une performance crête de 2,56 GCPS en vitesse de simulation.

Nom du Réseau	Temps de Simulation (ms)	Vitesse (MCPS)
LeNet	2,57	38,3
Nettalk	0,021	684

TAB. 1 — Résultats obtenus lors de la simulation de LeNet et de Nettalk

Dans un premier temps, nous avons simulé LeNet [9] qui est un PMC-CAL dédié à la reconnaissance de caractères manuscrits possédant 4365 neurones, 1920 connexions complètes et 96522 connexions locales. Nous avons aussi simulé Nettalk, pour effectuer une comparaison avec un PMC à connexions entièrement complètes, ces réseaux étant ceux pour lesquels CNAPS a été optimisée. Les résultats de ces simulations, rassemblés dans le tableau 1, montrent une grande différence entre les performances obtenues pour ces deux réseaux, puisqu'il existe un rapport 17 entre les vitesses de simulation mesurées. Les performances obtenues pour la simulation des PMC-CAL sont donc très inférieures à celles obtenues pour la simulation des PMC à connexions complètes et a fortiori à la performance crête à la machine.

Architecture	Temps de Simulation (ms)	Vitesse (MCPS)
Sparc 10	12,30	8,0
CNAPS 128 20 Mhz	2,57	38,3
ANNA chip	1,20	82,0

TAB. 2 — Simulation de LeNet sur Différents architectures

Comparaison Après ces constatations, il devient utile de comparer les résultats obtenus avec ceux d'autres plate-formes architecturales comme ANNA [11, 2, 3] qui est un ASIC analogique dédié aux PMC-CAL. Les mesures effectuées pour comparer différentes architecture sont résumées dans le tableau 2. Notons que CNAPS est seulement 2 fois plus lente qu'ANNA. Il est alors intéressant de déterminer quelles modifications de l'architecture de CNAPS permettraient d'augmenter la vitesse de simulation. Pour cela nous avons élaboré un modèle analytique du temps de simulation de la machine.

Modèle Analytique L'architecture d'un PMC-CAL est représentée par un graphe de couches orienté. Chaque couche est représentée par un sommet dans le graphe, chaque connexion entre deux couches est représentée par un arc entre deux sommets dans le graphe.
L'algorithme de simulation consiste en un parcours en largeur du graphe de couches en partant des couches d'entrée. Pour chaque sommet et chaque arc rencontré, l'une des trois primitives suivantes est exécutée :

- Le calcul des potentiels post-synaptiques dans le cas de connexions locales,
- Le calcul des potentiels post-synaptiques dans le cas de connexions complètes,
- La mise à jour de l'état des neurones.

La machine CNAPS possédant 2 bus de communications, cela nous donne en tout 5 différentes primitives. Le modèle analytique de CNAPS, pour ces 5 primitives, est décrit dans le tableau 3. Dans ce tableau les variables NPPA et NPPD indiquent le nombre de neurones par couches pour, respectivement, la couche de départ de la connexion et celle d'arrivée, PD indique le nombre de processeurs utilisés par la couche de départ de la connexion, C_1 indique la taille du voisinage de la connexion pour la première dimension et C_2 pour la seconde dimension, DG et DD indiquent respectivement la distance de communication à gauche et à droite, ces valeurs dépendant de C_1, de C_2 et de la taille de la couche de départ. Les paramètres α, γ, ε, θ, ψ, ω, β, ont été quant à eux estimés grâce à une analyse du code source et des mesures expérimentales [5].
Ce modèle nous permet de prédire la durée de simulation d'un PMC-CAL, en effectuant un parcours de son graphe de couches et en accumulant les temps d'exécution de chaque primitive.

réseau d'interconnexion	temps en nombre de cycles
Potentiels Post-Synaptiques pour les connexions complètes	
Bus IN et OUT	$\alpha + \gamma$*NPPD + ε*NPPD*NPPA + θ*PD*NPPA*NPPD
Bus Inter-PN	$\alpha + \varepsilon$*NPPD + θ*NPPD*P + ψ*NPPD*NPPA + ϕ*NPPD*NPPA*PD
Potentiels Post-Synaptiques pour les connexions locales	
Bus IN et OUT	$\alpha + \gamma$*NPPD + ε*NPPD*PD + θ*NPPA + ψ*NPPA*C_1*C_2
Bus Inter-PN	$\alpha + \gamma$*NPPD + ε*DD + ω*DG + β*NPPA + θ*NPPA*C_2 + ψ*NPPA*C_1*C_2
Mise à jour de l'état des neurones	
	$\alpha + \theta$*NPPA

TAB. 3 — Fonctions temps pour les primitives de base

Pour vérifier la validité du modèle, nous l'avons utilisé pour prédire la durée de simulation de LeNet. En la comparant au temps réellement mesuré (tableau 4) nous constatons que l'erreur commise n'est que de 5% ce qui est tout-à-fait acceptable.

Architecture	Temps de Simulation (ms)	Erreur
CNAPS 128 20 Mhz (mesuré)	2,57	
CNAPS 128 20 Mhz (prédit)	2,44	5%

TAB. 4 — Temps de simulation de LeNet - Prédit et Mesuré

Améliorations Grâce à notre modèle analytique nous avons effectué des prédictions de temps de simulation sur plusieurs

Architecture	Temps de Simulation (ms)	Vitesse (MCPS)
CNAPS 128 25 Mhz	1,95	50,4
CNAPS 512 20 Mhz	1,93	51,0
E-CNAPS 128 20 Mhz	1,92	51,2
E-CNAPS 512 25 Mhz	1,28	76,6

TAB. 5 — Prédiction de temps de simulation de LeNet avec différents changements dans l'architecture de CNAPS

architectures virtuelles, obtenues par des modifications de l'architecture initiale de CNAPS :

- CNAPS avec 128 PNs à la fréquence de 25 Mhz (augmentation de la féquence à nombre de PNs constant),
- CNAPS avec 512 PNs à la fréquence de 20 Mhz (augmentation du nombre de PNs à fréquence constante),
- E-CNAPS avec 128 PNs à la fréquence de 20 Mhz. (E-CNAPS diffère de CNAPS par un bus Inter-PN de 8 bits et par un registre de base d'adresse auto-décrémental),
- E-CNAPS avec 512 PNs à la fréquence de 25 Mhz (augmentation de la fréquence et du nombre de PNs de E-CNAPS).

Les prédictions effectuées, tableau 5, montrent un gain sensible de performance de l'ordre de 30% pour chacune de trois premières modifications. La quatrième modification, qui est une combinaison des trois premières, montre un gain de l'ordre 100%, ce qui donne un temps de simulation de 1,28 ms sensiblement égal à celui du circuit ANNA.

Conclusion Nous avons montré dans cet article, qu'une architecture SIMD tel que CNAPS, même si elle n'obtenait pas des vitesses de simulation proche de sa performance crête dans le cas de PMC-CAL, était seulement 2 fois plus lente qu'un ASIC comme ANNA, alors que CNAPS est programmable. De plus à l'aide du modèle analytique que nous avons élaboré, il nous a été possible de démontrer qu'une accélération de sa vitesse de simulation de 30% à 100% était envisageable grâce à de simples modifications de l'architecture. La méthode de prédiction proposée peut être généralisée, dans un premier temps aux machines parallèles spécialisées de type SIMD comme SYMPATI2 [8] et SYMPHONIE [4], pour lesquelles il suffit de mesurer les paramètres du tableau 3, dans un second temps à toutes les machines parallèles spécialisées en élaborant un modèle pour chaque classe d'architecture.

Références

[1] Jim Baley and Dan Hammerstrom. Why vlsi implementation of associative vlcns require connection multiplexing. In *Proceedings of the IEEE 2nd Annual International Conference on Neural Network*, pages 112–119, 1988.

[2] B. Boser, E. Sackinger, J. Bromley, Y. LeCun, and L. J ackel. An analog neural network processor with programmable topology. *IEEE Journal of Solid-State Circuits*, 26(12) :2017–2025, December 1991.

[3] B. Boser, E. Sackinger, J. Bromley, Y. LeCun, and L. J ackel. Hardware requirements for neural network pattern classifiers. *IEEE Micro*, 12(1) :32–40, February 1992.

[4] T. Colette, C. Gamrat, D. Juvin, J.F Larue, L. Letellier, and R.Schmit M. Viala. Symphonie, calculateur massivement parallèle : modélisation et réalisation. *Journée adéquation algotirhme architecture en traitement du signal et images*, pages 279–286, January 1996.

[5] Bertrand Granado and Patrick Garda. Evaluation of the two differents interconnection networks of the cnaps neurocomputer. In *Proceedings of ICANN'96*, Juillet 1996.

[6] Matthew Griffin, Gary Tahara, Kurt Knorpp, Ray Pinkham, Bob Riley, Dan Hamerstrom, and Eric Means. An 11 million transistor digital neural network execution engine. In *Proceedings of IEEE International Solid-State Circuits Conference*, 1991.

[7] Dan Hammerstrom. A vlsi architecture for high-performance, low-cost, on-chip learning. In *Proceedings of Internationnal Join Conference on Neural Network*, pages 537 – 544, 1990.

[8] D. Juvin, J.F Basille, H. essafi, and J.Y Latil. Sympati2, a 1.5d processor array for image application. *Signal Processing IV : Theories and applications*, pages 311–314, 1988.

[9] Y. LeCun, B. Boser, J.S. Denker, D.henderson, R.E. Howard, W. hubbard, and L.J. Jackel. Handwritten digit recognition with a back-propagation network. In *Neural Information Process and System*, pages 396–404, 1990.

[10] Dean Mueller and Dan Hammerstrom. A neural network systems component. In *Proceedings od International Conference on Neural Network*, pages 1258–1264, 1993.

[11] Eduard Säckinger, Bernhard Boser, Jane Bromley, Yann LeCun, and Lawrence D. Jackel. Application of the ANNA neural network chip to high-speed character recognition. *IEEE Transaction on Neural Networks*, 3(2), March 1992.