

TAO is SILENT ou les Machines LISP chez NTT

Introduction

Information Science Research Laboratory est un des laboratoires de recherche du Basic Research Laboratories de NTT, situé dans le complexe de Musashino, légèrement à l'ouest de Tokyo intra-muros. Ikuo Takeuchi dirige une équipe dans ce laboratoire depuis environ une vingtaine d'années. Il conçoit des ordinateurs spécialisés adaptés au calcul symbolique, appelés Machines Lisp. Signalons tout d'abord que Ikuo Takeuchi a déjà laissé son nom (ou plutôt le début de son nom) à la postérité dans le domaine de la programmation fonctionnelle (et Lisp en particulier). La fonction récursive "tak" qu'il a définie est en effet une des fonctions classiques d'évaluation de performances (benchmarks) de systèmes Lisp.

Calcul Symbolique

Rappelons que par opposition au calcul numérique mis en oeuvre dans les applications scientifiques, le calcul symbolique manipule (comme son nom l'indique!) principalement des symboles. Le domaine d'application traditionnel et privilégié est l'Intelligence Artificielle (résolution de problèmes, représentation de connaissances, traduction...) mais les applications informatiques devenant de plus en plus sophistiquées, cette approche tend à se généraliser. Les langages de programmation par objets sont d'ailleurs les représentants les plus récents du calcul symbolique. Les langages de programmation fonctionnelle (le plus ancien et le plus connu étant Lisp) et les langages de programmation logique (Prolog) en sont les deux autres grands représentants.

Machines dédiées

Pour pouvoir obtenir d'une part une efficacité optimale ainsi que d'autre part un environnement de développement adapté au mieux à un langage de programmation, les chercheurs ont commencé à s'intéresser au début des années 70 à des ordinateurs dédiés à un langage. Le premier prototype en aura été la machine Smalltalk, également premier environnement de développement à objets. L'idée est que l'ensemble de l'environnement de la machine, incluant donc le système d'exploitation, la gestion des fichiers, l'affichage... soit programmé dans un même langage, de surcroît de haut niveau (et non pas un assembleur classique). Ce type de machine permet ainsi d'avoir un environnement de programmation très sophistiqué, une efficacité optimisée, et surtout tous les aspects de la machine (jusqu'au système d'exploitation) accessibles via un unique langage de programmation (et donc plus aucune distinction stricte entre langage machine et langage utilisateur). Cette machine Smalltalk a entraîné le développement de machines Lisp au MIT (Massachusetts Institute of Technology, Boston, USA). La société Symbolics, issue du MIT, en est la plus illustre représentante. Des machines Prolog ont ensuite également été développées, notamment dans le cadre du projet Japonais des ordinateurs de 5ème génération (ICOT, 1982-1992). Signalons qu'une machine Lisp a également été développée en France au début des années 80 par un projet CGE+CNET. Ce projet hélas n'a pas donné lieu à une commercialisation. La tendance du marché des machines dédiées s'est en effet inversée à partir du milieu des années 80, comme nous allons le voir. Les constructeurs apparus tardivement sur ce marché (également Texas Instruments, ainsi que NTT, le sujet de cet article) auront ainsi été les premiers à faire les frais d'un marché en réduction.

Tendances

Il faut signaler que la période faste pour les machines dédiées est terminée depuis la fin des années 80. En effet l'avènement de stations de travail très rapides (notamment avec les nouveaux processeurs RISC, c'est-à-dire, à jeu d'instructions réduites et donc optimisées) dotées d'environnements de programmation très sophistiqués bien que génériques (par exemple le standard de multi-fenêtrage X-Windows, retombée du travail de pionnier des

machines Lisp) et enfin de prix très attractifs (production de masse aidant) a progressivement rendu moins intéressant le concept de machine dédiée. Au sujet de X-Windows, signalons une retombée paradoxale (après avoir contribué à l'éliminer) du concept de machine dédiée: avec l'apparition de terminaux spécialisés (appelés terminaux-X), petits processeurs/terminaux dédiés à l'affichage.

Symbolics, le principal constructeur américain de machines Lisp et le dernier à rester en piste, a alors abandonné le développement de machines autonomes et s'est spécialisé dans des processeurs spécialisés à adjoindre à une station de travail conventionnelle (le processeur/puce "Ivory"). C'est d'ailleurs, comme nous allons le voir, la nouvelle approche également suivie chez NTT.

Signalons enfin, et ce n'est pas l'information la moins importante, le rachat récent de Symbolics par NTT. NTT reste donc maintenant de fait le seul organisme à contrôler le marché (maintenant quasi-inexistant, mais de technologie à retombées potentielles) des machines Lisp.

SILENT

Nous allons maintenant présenter le développement de machine(s) Lisp chez NTT. La machine prototype actuelle, appelée SILENT et en cours de fabrication. Elle est en fait la deuxième conçue par cette équipe. Son prédécesseur se nommait ELIS.

ELIS

Ikuo Takeuchi a dirigé le développement de machines Lisp aux laboratoires NTT depuis un certain temps. Une première machine a été développée au début des années 80. Elle se nommait ELIS, son langage de programmation se nommant TAO. Le langage TAO était présenté comme une unification (présentée comme une somme harmonique!) de Lisp, Prolog et Smalltalk. Pour être plus précis TAO était un langage fonctionnel de type Lisp qui intégrait les principes de la programmation logique et de la programmation par objets. Cette machine a été commercialisée par NTT au milieu des années 1985, mais sans succès. Malgré son échec commercial, NTT n'a pas abandonné ce projet et au contraire soutient une deuxième mouture.

Silent

La machine successeur de ELIS se nomme SILENT. Ikuo Takeuchi propose lui-même deux explications au nom choisi. C'est une référence/hommage au livre "TAO is Silent" de Raymond Smullyan, introduction "lumineuse" au TAO d'après Takeuchi. De plus une fois inversé SILENT devient un acronyme de The New ELIS.

Cette nouvelle machine est destinée à ne pas être autonome, mais sera une unité de calcul symbolique spécialisée, associée (via une interface SCSI) à une station de travail classique (par exemple une station Sparc de Sun). Ainsi il n'est plus question comme pour ELIS et les premières machines Lisp de reconstruire un environnement complet (multi-fenêtrage, gestion de fichiers...), il sera donc délégué à la station de travail.

Caractéristiques du Processeur

Nous donnons quelques caractéristiques du processeur telles que précisées par Takeuchi. Le processeur spécialisé de SILENT est de type CISC et utilise la technologie CMOS 0.7 micron gate array (100k gates). SILENT est micro-programmable. La taille mémoire est de 160 Mo. Un mot mémoire fait 40 bits, dont 8 bits de tag. Le cycle est de 30 nano-secondes. Le processeur est en cours de fabrication et sera disponible au mois d'Avril. La carte mère est déjà terminée et donc prête à accueillir la puce.

Applications

Les applications envisagées pour SILENT sont les applications temps réel de haut niveau: contrôle (routeur) de réseau de communications intelligent, robotique, infographie. SILENT peut traiter une requête sous 100 micro-secondes.

Au sujet de l'infographie, un constat de Takeuchi est que la programmation d'applications de synthèse/traitement d'images devient de plus en plus sophistiquée et donc bénéficiera pleinement d'un processeur symbolique spécialisé. SILENT est destinée à être

connectée notamment à une autre machine dédiée au traitement d'images et développée à NTT. Il s'agit de la machine TARAI, composée de 3 processeurs (3 étant un nombre "idéel" relatif aux 3 dimensions graphiques et aux 6 degrés de liberté utilisés en général en robotique, dicit Takeuchi). TARAI est prévue pour complétion à mi-1994.

TAO

TAO est le langage (machine) de SILENT. Chaque primitive possède un temps d'exécution déclaré dans le manuel. Ainsi par exemple, l'opération "car" (accès au premier élément d'un doublet) s'effectue en 3 cycles (et donc 90 nano-secondes), ceci incluant un contrôle de type. Le temps d'exécution tombe à 500 nano-secondes si la donnée n'est pas dans la mémoire cache.

TAO est un dialecte de Lisp, très proche de Common Lisp (le standard de fait de Lisp). Comparé au premier TAO de la machine ELIS, le TAO nouveau est présenté par Takeuchi comme améliorant l'intégration des 3 types de programmation (fonctionnel, logique, et à objets).

TAO n'est pas défini comme un langage de programmation directement à l'usage de l'utilisateur. Les primitives de gestion des objets et de la concurrence en particulier sont d'assez bas-niveau. L'objectif est d'offrir au concepteur de langages les primitives de base pour pouvoir construire des langages de niveau plus élevés.

Objets

Ainsi en particulier, TAO offre des objets très primitifs, en fait constitués d'environnements lexicaux (au sens de Lisp lexical, tel que Scheme). Les primitives d'auto-référence, d'écriture dans les environnements, de définition d'opérations privées à l'objet, et enfin de gestion d'erreur en cas d'invocation d'une opération absente sont proposées, mais pas plus. Il n'y a donc pas de classes ni d'héritage. Mais on peut construire de tels mécanismes de plus haut niveau à l'aide de ces objets primitifs et des mécanismes d'invocation et de délégation primitifs. Signalons que Takeuchi n'envisage pas de développer lui-même une telle extension, et notamment pas d'implémentation du standard de fait en ce qui concerne les couches objets en Lisp, c'est-à-dire CLOS (Common Lisp Object System).

Logique

TAO offre également le nécessaire pour une programmation de type Prolog. Le préfixe "_" (souligné) permet de retourner la position (mémoire) d'une expression, en plus de sa valeur. Ceci permet d'implémenter les variables logiques. Un mécanisme de retour en arrière (backtrack) est également fourni. La syntaxe des crochets permet de distinguer entre appel fonctionnel "...", démonstration de but logique "{...}", ou transmission de message à objet "[...]".

Concurrence

SILENT est une machine mono-utilisateur mais multi-tâche. TAO intègre donc des mécanismes de concurrence: processus (activités), sémaphores (primitives de synchronisation), boîtes aux lettres (primitives de communication), mémoires tampon (comme les boîtes aux lettres mais de taille bornée). Ces primitives sont de facture très classique, et relativement de bas niveau. Signalons par contre qu'un processus possède une queue de tâches. En ce sens un processus TAO est relativement évolué (proche d'un objet ou/et d'un séquenceur) puisqu'il exécute successivement plusieurs tâches communiquées. Signalons que chaque processus possède également une queue d'événements. Ceci permet ainsi le traitement d'interruptions.

Le temps de basculement (context switching) d'un processus à un autre est de 5 micro-secondes. La pile d'exécution contient l'ensemble des piles propres à chaque processus. Cela limite en conséquence le nombre de processus simultanés à une cinquantaine. Cette limitation indique que le grain de concurrence visé reste relativement important, et donc ne peut couvrir les programmes concurrents à grain très fin.

Il existe un pseudo-processus ayant un statut particulier. Il s'agit du récupérateur de mémoire (garbage collector) nécessaire dans les systèmes à gestion implicite de la

mémoire (notamment Lisp qui en a d'ailleurs été l'inventeur). L'algorithme choisi reste de facture classique (mark and sweep).

Conclusion

La machine SILENT et son langage de programmation TAO représentent un travail de qualité produit par une équipe réduite. La machine annonce des performances intéressantes sinon époustouflantes. La facture reste très classique, ce qui souligne le côté artisanal et autarcique de ce développement. Ainsi on peut remarquer que des résultats récents sur des algorithmes (ainsi sur des récupérateurs de mémoire optimisés pour le temps-réel) ou des standards (ainsi le standard à objets CLOS) n'ont pas été envisagés. L'aspect parfois quelque peu idiosyncrasique de la syntaxe (notamment les notations "condensées" pour affectations dans les objets, à base de "!") confirme cette impression de relative autarcie.

Il est intéressant de constater que NTT continue à soutenir l'équipe de Takeuchi, à notre avis sans optique de commercialisation (cela a déjà été tenté une première fois au milieu des années 80 avec la première machine ELIS, mais cela a abouti à un échec, le marché se rétrécissant fortement à ce moment). Les objectifs annoncés d'application (réseaux, robotique, infographie) ne sont pas sans raison, mais à moyen terme au minimum. Il s'agirait donc plutôt d'une politique de positionnement à long terme (une politique de recherche de base/attente en quelque sorte...). La technologie et l'équipe qui l'entretient et la développe restent ainsi activées et soutenues dans l'optique de retombées quasi-certaines mais d'échéances inconnues. De plus cette technologie n'est maintenant perpétrée que par NTT (qui vient de racheter Symbolics, le premier et dernier fabricant spécialisé, il est vrai beaucoup plus petit que le "monstre" NTT). Ceci illustre clairement la stratégie à long terme Japonaise, que l'on peut opposer notamment à l'optique de rentabilité à court terme des Etats-Unis.

Nous tenons à remercier Nicolas RAGUIDEAU, du Switching Communication Labs, qui a organisé notre visite. Nous remercions tout spécialement Ikuo TAKEUCHI qui nous a accueilli très chaleureusement et extensivement.

Jean-Pierre Briot(*) et FXTV

(*) Department of Information Science, Faculty of Science,

The University of Tokyo

Hongo, Bunkyo-ku, Tokyo 113

tel: (03) 38 12 21 11 ext. 4120

ligne directe: (03) 58 00 69 13

fax: (03) 56 89 43 65

e-mail: briot@is.s.u-tokyo.ac.jp