Actes des $6^{es}$ Rencontres
francophones du parallélisme,
RenPar'6, L. Bougé (réd.),
ENS Lyon, France, 6–10 juin 1994. 

# Modèle de mémoire distribuée-partagée pour machine massivement parallèle

Lionel Brunie   Laurent Lefèvre   e-mail:(lbrunie,llefevre)@lip.ens-lyon.fr
L.I.P., ENS Lyon, 46, allée d'Italie, 69364 LYON Cedex 07

Deux classes d'architectures se confrontent dans le monde des machines parallèles MIMD. La première est celle des machines a mémoire partagée dans laquelle tous les processeurs ont accès à une seule mémoire physique. Beaucoup de ces machines sont limitées en nombre de processeurs mais leur programmation est facile (pas de phases de communication). A l'inverse, les machines à mémoire distribuée peuvent intégrer un grand nombre de processeurs, elles permettent ainsi d'atteindre de bonnes performances. Mais la programmation d'applications sur ces machines est difficile; toutes les communications étant à la charge de l'utilisateur. Un compromis logiciel entre ces deux types de machines est un système à base de mémoire partagée-distribuée. Par l'ajout d'une couche logicielle, ce modèle offre, ainsi, la facilité de programmation des machines à mémoire partagée tout en utilisant l'efficacité des machines à mémoire distribuée.

Néanmoins la plupart des systèmes de mémoire distribuée-partagée (à base de pages comme à base d'objets) ne supportent qu'un nombre réduit de processeurs pour être encore efficaces.

Cet article présente un modèle de système de mémoire partagée-distribuée à base d'objets partagés adapté à l'implémentation sur une machine massivement parallèle. Ce modèle comporte deux classes de processus, ceux qui éxécutent l'application et les processus dédiés à la gestion des objets partagés. Le système doit éviter au maximum les communications globales qui bloquent l'ensemble des processus. On peut ainsi envisager deux types de partitionnements (domaines d'objets ou groupes de processus) pour réduire volume et distance des communications nécéssaires à la gestion de nos objets partagés. On définit le domaine d'un objet comme l'ensemble des processus qui partagent cet objet. Ce découpage fonctionnel assure le fait que les processus d'un domaine n'auront pas besoin, au cours de l'application, d'accéder à d'autres domaines d'objets. La difficulté réside dans le placement et la distribution des objets dont les domaines sont fortement intersectés. On préfère donc utiliser une organisation de groupes hiérarchiques dans lesquels les processus partagent des objets communs. Par contre les processus peuvent exceptionnellement utiliser les objets d'autres groupes. Cette organisation n'est pas fonctionnelle. Si un processus se met à accéder fréquemment à un objet d'un autre groupe; le système de gestion de la mémoire sera surchargé.

Pour éviter qu'un processus ne dévie ainsi en cours d'application, nous proposons un modèle hybride qui permet l'évolution de la répartition des processus dans les groupes. Si un processus effectue souvent de nombreuses demandes sur un objet hors de son groupe, il faudra peut-être envisager de l'inclure dans le groupe accédant souvent à cet objet. Bien entendu, les communications intra-groupe pour maintenir la cohérence des données (processus chefs) et communications inter-groupes (grâce à des processus passerelles) sont prises en compte par le système.

Afin de démontrer la faisabilité de notre modèle sur une machine parallèle, un développement logiciel de notre système (appelé DOSMOS[1]) est en cours de réalisation. Nous espérons ainsi pouvoir présenter au cours de ces journées des résultats d'expérimentations d'applications tournant sous le système DOSMOS.

1. DOSMOS: Distributed Objects Shared MemOry System - Système de mémoire partagée d'objets distribués