
Editorial

Ce numéro spécial de la revue *Calculateurs Parallèles, Réseaux et Systèmes Répartis* fait un tour d'horizon, tant au niveau national qu'international, des récents travaux en matière de conception de noyaux d'exécution ou d'environnements utilisant les techniques de "multithreading"; l'apport de ces techniques à la conception d'applications temps réel est aussi présenté dans ce numéro.

Les fils ou "threads" que l'on retrouve parfois dans la littérature sous la dénomination de processus légers ou "lightweight process" partagent un espace d'adressage ainsi que d'autres ressources et présentent des temps de commutation de contexte plus brefs que ceux des processus lourds supportés par le noyau du système; ils peuvent notamment partager un nœud de calcul de manière parallèle comme dans le cas d'un multiprocesseur symétrique (SMP) ou un monoprocesseur en temps partagé. Les processus légers ont été introduits en particulier dans les systèmes d'exploitation afin de permettre à un serveur d'un système réparti de répondre efficacement à des requêtes simultanées. De manière plus précise, lorsqu'une requête est bloquée sur une entrée ou une sortie, une autre requête peut être exécutée en maintenant ainsi une activité soutenue du serveur.

Les quatre premiers articles de ce numéro portent sur la conception de supports d'exécution permettant l'expression explicite d'un réseau dynamique de fils sur une architecture à mémoire distribuée. Les applications visées sont essentiellement à grain fin et la portabilité est un des objectifs majeurs. La première approche fait usage d'opérateurs de découpage d'applications de type appel de procédure à distance.

R. Namyst, Y. Denneulin, J. M. Geib et J. F. Méhaut exposent l'environnement PM2 d'utilisation de processus légers conçu essentiellement pour des applications de calcul scientifique et plus particulièrement des applications irrégulières à grain fin. Ils décrivent le modèle de programmation résultant de PM2 et ses objectifs en mettant l'accent sur la virtualisation des architectures distribuées grâce à des opérateurs de découpage efficaces. Les auteurs mettent en valeur la mobilité des activités au travers de mécanismes de migration et plus précisément la facilité de construction d'un régulateur de charge au dessus de PM2. Les performances de l'approche PM2 sont présentées dans le cadre d'une application particulière: le problème du voyageur de commerce.

Les trois articles suivants sont axés sur le mariage entre des fonctionnalités de manipulation de fils et l'envoi de message.

L'article de I. Foster, C. Kesselman et S. Tuecke présente la bibliothèque Nexus qui est un outil original permettant d'associer les communications aux processus légers. L'objectif de Nexus est l'utilisation souple de machines à haute performance reliées par un réseau, cadre où prédomine l'hétérogénéité des machines et des liens entre les processeurs. Ceci conduit au concept général de "multimethod communication" qui se rapporte à l'utilisation conjointe de diverses méthodes de communication. Les applications visées sont de type modèles couplés pour lesquels il est nécessaire de posséder des mémoires de très grande taille et d'utiliser toutes les possibilités de parallélisme. Nexus propose un cadre abstrait naturel pour les concepts de lien de communication et de demande de service à distance.

J. Briat, I. Ginzburg et M. Pasin décrivent le noyau exécutif Athapascan-0 à base de processus légers qui est au cœur de l'environnement Athapascan, conçu pour la programmation d'applications irrégulières de grande taille par un paradigme de parallélisme de tâches. L'architecture parallèle est vue dans cet article comme un réseau de multiprocesseurs. Athapascan-0 permet le placement et le lancement de calculs à distance (exécutés par des fils), qui communiquent par messages et accèdent aux mémoires distantes. L'équilibre de la charge est réalisé via l'exploitation du parallélisme interne des nœuds de calcul et le placement dynamique des calculs sur des nœuds distants. L'asynchronisme de tous les opérateurs est renforcé afin de recouvrir au mieux les attentes et les latences par du travail utile.

L'article de J. N. Colin et D. Pavisic présente l'environnement ArchipeL qui propose une bibliothèque de processus légers servant à mettre en œuvre un modèle de structuration d'applications distribuées basé sur la notion de grappes de tâche; ce modèle de type SCPMD (Single Cluster of Programs Multiple Data) permet de représenter un ensemble de programme groupés, chaque groupe travaillant sur son propre ensemble de données. Cette approche est adaptée à des applications irrégulières à grain fin, les méthodes employées pouvant être associées par exemple à l'utilisation d'algorithmes génétiques ou de réseaux de neurones. La présentation privilégie les aspects dynamiques du modèle ainsi que sa mise en œuvre. Les différents niveaux de validation sont aussi abordés.

Dans le dernier article, Ph. Owezarski et M. Diaz présentent une utilisation des techniques de "multithreading" qui est orientée vers les applications temps réel. Les auteurs donnent une méthodologie originale pour la conception d'applications multimédia. Ils exposent les contraintes imposées par l'application et le modèle retenu : TSPN (Time Stream Petri Net) qui offre un bon pouvoir d'expression et qui est très bien adapté à la modélisation des contraintes de synchronisation de flux multimedia en environnement asynchrone. L'architecture "multithreads" de Solaris 2 est détaillée et l'accent est mis sur les problèmes d'ordonnancement. Puis les problèmes rencontrés avec Solaris 2 sont analysés. L'article s'achève par la présentation de l'outil de visioconférence synchronisé PNSVS.

Didier EL BAZ
LAAS du CNRS

Brigitte PLATEAU
LGI/IMAG