Ecole Doctorale STIC

Sciences & Technologies de l'Information et de la Communication

ED STIC - Proposition de Sujets de Thèse

pour la campagne d'Allocation de thèses 2011

Titre du sujet :	Modèles de Programmation pour l'Analyse de Donnés sur Machines Multicoeurs , Grilles et Cloud.
Mention de thèse :	Informatique
HDR Directeur de thèse inscrit à l'ED STIC :	Baude Françoise

Co-encadrant de thèse éventuel :

Nom :	Henrio
Prénom :	Ludovic
Email :	ludovic.henrio@inria.fr
Téléphone :	04 92 38 71 64

Email de contact pour ce sujet :	ludovic.henrio@inria.fr
Laboratoire d'accueil :	I3S/INRIA

Description du sujet :

Cette thèse se situe à l'intersection de deux contextes actuellement très dynamiques, les architectures parallèles et réparties, et le traitement des données sémantiques. Elle est orientée par un cas d'utilisation fort : le traitement de données RDF.

Les architectures multicoeurs sont maintenant incontournables. Par exemple un processeur récent associe huit cœurs avec une quantité importante de mémoire cache. Pour écrire des programmes pour ces architectures, il est possible d'avoir recours à des environnements de bas niveau, tel les pthreads [1], ou de beaucoup plus haut niveau comme les modèles à acteur [2]. Cependant, pour écrire des programmes performants et fiables, il faut tenir compte de l'organisation de ces machines en cluster ou en Cloud. Alors que dans un cluster la topologie est connue et maitrisée, dans un Cloud IaaS (Infrastructure as a Service), un programme peut s'exécuter sur un ensemble de machines, géographiquement distribuées, dans des

environnements virtualisés. Il faut donc pouvoir exprimer la distribution des calculs ainsi que le placement des données de manière logique, indépendamment de l'architecture physique sous-jacente sur laquelle le programme va
s'exécuter. L'équipe Oasis a, dans ce domaine, une expertise reconnue. Nous avons travaillé sur un modèle, ASP [3] qui a été implémenté dans le middleware ProActive [4]. Ce modèle implémente le concept d'objet actif. Il s'agit de regrouper, au sein d'une même entité les données ainsi que le fil d'exécution. Les objets actifs sont ainsi des objets mono-threadés avec communications asynchrones. Les données entre objets actifs ne sont pas partagées ce qui facilite la programmation et la vérification de programme. Cependant, ce modèle ne permet pas actuellement de tirer efficacement parti des architectures multicoeurs.

Avec le développement du Web Sémantique, une énorme quantité de données sémantiques est aujourd'hui accessible. Celles-ci peuvent être stockées en utilisant le format RDF [5] qui permet de les structurer sous forme de triple ou de n-tuple. Par exemple, la version structurée de Wikipedia, DBPedia, compte actuellement 672 millions de triples. Analyser ces données peut se faire à travers des langages de requêtes de haut niveau. Ainsi, le langage SPARQL [6] permet d'effectuer des opérations de filtrage ou de jointure. De manière générale, stocker et traiter ces données de manière efficace requiert des algorithmes, des programmes, et des architectures distribués. Dans le cadre de deux projets Européens, SOA4All (IP-FP7) et Play (STERPs-FP7), nous développons un stockage réparti de données RDF pouvant être interrogé grâce à des requêtes SPARQL. L'architecture actuellement choisie repose sur des technologies pair-à-pair utilisant les objets actifs.

Le but de ce travail de thèse est d'étudier les modèles de programmation permettant de manipuler et d'analyser de grandes quantités de données distribuées sur des architectures hiérarchiques, des processeurs multicoeurs au Cloud. Notre but est de promouvoir des modèles de programmation haut niveau, et fournissant au programmeur une vue abstraite des aspects de concurrence et de distribution. En effet, un modèle de programmation plus abstrait est tout d'abord plus facile à programmer, mais aussi à de meilleures propriétés, il est donc plus facile de garantir l'exécution sûre des programmes écrits dans de tels langages. Durant ce travail, il sera particulièrement intéressant d'étudier des extensions du modèle ASP.

L'idée est de fournir un modèle de programmation automatisant l'exécution concurrente locale à chaque machine pour mieux profiter des architectures multicoeurs : l'utilisateur fournit des informations de haut niveau nous permettant de diriger une exécution plus concurrente que ce que permet le modèle ASP. À plus gros grain, des modèles de programmation comme Map/Reduce [7] pourront être abordés. En effet de tels modèles de programmations, encore plus haut niveau que le modèle à objets actifs et plus spécifiques, fourniront une approche complémentaire à la notion d'objets actifs développés ci-dessus, nous pensons ainsi obtenir a la fin de cette thèse un modèle de programmation adapté aux architectures de type Cloud et fournissant des propriétés de sûreté.

Ce travail de thèse peut être décomposé en plusieurs étapes. En premier lieu, nous souhaitons nous concentrer sur des extensions du modèle ASP afin de mieux tirer partie des architectures

modernes. Ces extensions devront être suffisamment expressives pour permettre à des non spécialistes de développer des applications parallèles et distribuées fiables. Afin de les valider, elles seront implémentées dans le middleware ProActive et des expériences à large échelle seront conduites sur des plateformes nationales (Grid'5000) et internationales (Amazon EC2).

Dans une deuxième étape nous étudierons les modèles de plus haut niveau qui sont plus expressifs lorsque l'on considère des architectures à gros grain comme les Clouds. Ci-dessus nous avons cité un tel modèle de plus haut niveautrès expressif et prometteur, Map/Reduce, mais d'autres modèles devraient être étudiés afin de fournir une diversité dans les abstractions fournies au programmeur.

Références :

[1] Bradford Nichols, Dick Buttlar, and Jacqueline P. Farrell. Pthreads Programming. O'Reilly, 101 Morris Street, Sebastopol, CA 95472, 1998

[2] Gul Agha, Actors: A Model of Concurrent Computation in Distributed Systems. Doctoral Dissertation. MIT Press, 1986.

[3] Denis Caromel ad Ludovic Henrio, A Theory of Distributed Objects, Springer-Verlag,2004

[4] ProActive, http://proactive.activeeon.com

[5] Dave Beckett, RDF/XML Syntax Specification (Revised), W3C Recommandation, 2004

[6] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. Technical report, W3C, 2006.

[7] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. Commun. ACM 51, 1, January 2008.

English version: