

Exécution d'applications à base de composants logiciels sur des architectures à large échelle

Eddy Caron et Christian Pérez

14 mai 2008

Mots-clés : modèle de composants, découverte de services, ordonnancement, plates-formes large échelle

Encadrants : Eddy Caron et Christian Pérez.

Laboratoire : LIP, ENS Lyon, UMR CNRS-INRIA 5668, 46 allée d'Italie, 69364 Lyon Cedex 07.

Téléphone : +33(0)437287646

Télécopie : +33(0)472728080

Adresse électronique : Eddy.Caron@ens-lyon.fr et Christian.Perez@inria.fr

1 Contexte

Le sujet de cette thèse concerne l'étude de l'appariement des éléments d'une application à base de composants logiciels à des ressources et/ou services qu'il aura fallu découvrir.

L'évolution continue des moyens de calculs permet d'envisager de réaliser des simulations numériques toujours plus réalistes. Si dans un premier temps, le parallélisme a surtout permis de simuler plus finement des phénomènes physiques, le très grand nombre d'unités de calcul disponibles permet désormais de simuler simultanément *plusieurs* phénomènes physiques. Les architectures de types grilles de calcul, ou les machines parallèles "péta-flopiques", sont de nos jours capables d'offrir des centaines de milliers d'unités de calcul. On peut imaginer que dans peu de temps, l'ordre de grandeur du nombre d'unités de calcul sera le million.

L'exécution de telles applications numériques sur des architectures aussi puissantes mais complexes demande en particulier de disposer d'un modèle de programmation adapté ainsi que de mécanismes pour efficacement associer les calculs aux ressources disponibles.

Concernant les modèles de programmation, les modèles à base de composants logiciels apparaissent particulièrement bien adaptés. Ils visent la simplification de la programmation en se focalisant en particulier sur la ré-utilisabilité des codes. Ainsi, un composant apparaît comme une boîte noire qui explicite les services fournis ainsi que ceux requis à son bon fonctionnement. Parmi les nombreuses initiatives internationales, deux modèles de composants logiciels sont particulièrement intéressants. D'une part, le Grid Component Model est un modèle de composant issue du réseau d'excellence européen CoreGRID qui a spécialisé le modèle de composant Fractal pour les grilles de calcul. D'autre part, le Common Component Architecture est à l'origine une initiative du département américain de l'énergie (DoE) qui vise en priorité l'obtention de hautes performances sur des machines parallèles. Cependant, ces modèles ne prennent en compte que des compositions spatiales. Récemment, nous avons entrepris des travaux unifiant les modèles de composants avec des modèles de flots de travail afin de disposer d'un modèle cohérent vis à vis des compositions spatiales et temporelles.

Les architectures visées sont des architectures à large échelle; elles sont sujet à une certaine volatilité causée par le partage de ressources, des défaillances matérielles ou logicielles, etc. Ainsi, la construction d'une connaissance globale et précise de ce type d'architectures n'est pas envisageable. Or, une exécution efficace des applications demandent un placement et un ordonnancement pertinent des éléments de l'application, opérations fortement dépendantes de la connaissance des ressources.

2 Travail à effectuer

L'objectif de la thèse est d'étudier comment exécuter efficacement des applications basées sur des compositions spatiales et temporelles sur des architectures à large échelle. Les difficultés à appréhender sont la dynamique des applications, la large échelle et la volatilité de l'infrastructure.

2.1 Modèle de composants spatio-temporelles

Ce n'est que récemment que des modèles de composants permettant de construire des applications via des compositions spatiales et temporelles ont été proposées. La dimension temporelle (coordination) introduite se base sur les modèles de flot de travail. Si les modèles de flot de travail sont très populaires, ils reposent souvent sur une gestion centralisée de l'état de l'application (orchestration) qui peut présenter un goulot d'étranglement. Certains modèles plus académiques visent à permettre une gestion plus décentralisée (chorégraphie). Ainsi, il pourrait être nécessaire de revisiter les modèles de coordination pour limiter les goulots d'étranglement à l'exécution.

2.2 Ordonnancement spatio-temporel

Traditionnellement, les modèles de composants permettent de décrire l'architecture (statique) à l'exécution d'une application. Ainsi, il n'était pas besoin d'ordonner les applications à base de composants logiciel ; l'ordonnancement d'applications s'exprimant sous forme de graphe de tâches (dimension temporelle) a par contre été très étudié. Ainsi, la prise en compte de la dimension spatiale rajoute des communications entre les nœuds d'un graphe de tâches. Il s'agit donc d'étudier s'il est possible de ramener l'ordonnancement d'une application à base de composition spatio-temporelles à un graphe de tâches.

2.3 Découverte de services

L'ordonnancement d'une application sur des ressources demandent de connaître l'infrastructure. Vu la taille et la volatilité des architectures visées, il ne semble pas réaliste de supposer que l'ordonnanceur peut disposer d'une vision globale de toutes les ressources. Ainsi, au lieu des traditionnels services d'information plus ou moins centralisés, des modèles de gestion de ressources/services basés sur des technologies pair-à-pair ont été proposés. Dans le cadre de la thèse de Cédric Tedeschi qui se termine en 2008, nous avons mis en place des solutions algorithmiques.

L'efficacité de ce système est un point crucial à la viabilité de l'architecture visée, dynamique et large échelle. Grâce à ses caractéristiques, ce type d'architecture présente des intérêts communs avec les systèmes P2P. Cette convergence a donné lieu à d'importants travaux initiés récemment. L'un des inconvénients majeurs des premières solutions proposées pour la recherche de ressources dans un milieu P2P, comme les tables de hachage distribuées est leur rigidité, rendant difficile leur utilisation au sein des grilles de calcul. Partant de ce constat, un challenge a été d'ajouter flexibilité et complexité à la fonctionnalité de recherche dans les réseaux pair-à-pair.

Une approche prometteuse consiste à maintenir un arbre lexicographique (ou de préfixes) sur les services déclarés par les serveurs. Parmi cette dernière série de travaux, l'équipe GRAAL a proposé une architecture appelée DLPT (Distributed Lexical Placement Table) destinée à la gestion totalement décentralisée de la découverte de services dans les grilles de calcul. Cette architecture étend par exemple les fonctionnalités de recherche aux requêtes sur des intervalles de valeurs, permet l'autocomplétion des chaînes partielles de recherche et s'étend facilement aux requêtes multi-critères. Cependant cette approche a ouvert de nombreux points qu'il nous faut encore explorer tels que l'équilibrage de charge, des approches *on-demand*, la localité, la tolérance aux pannes ou encore l'extensibilité.

2.4 Mise en œuvre

Dans le cadre de nos collaborations avec divers partenaires (EDF R&D, CERFACS, Ames Lab, l'équipe-projet INRIA RESO), nous disposons de plusieurs applications aux caractéristiques différents. Il est attendu que les travaux réalisés au cours de cette thèse soit validés expérimentalement en particulier sur la plate-forme expérimentale Grid'5000.