Dossier de candidature à la qualification aux fonctions de maître de conférences

Loris Marchal

Laboratoire de l'Informatique du Parallélisme
UMR CNRS-ENS Lyon-UCB Lyon-INRIA 5668
École Normale Supérieure de Lyon

Contenu du dossier :

État civil

Loris Marchal

- Né le 22 février 1980 à Lyon 6eme (69).
- Nationalité française.
- Célibataire.
- Adresse professionnelle : LIP, ENS-Lyon, 46 allée d'Italie, 69364 Lyon Cedex 07.
- Téléphone professionnel : 04 72 72 85 04.
- Adresse personnelle : 23 rue René Leynaud, 69006 Lyon.
- Téléphone personnel : 04 64 51 58 95.
- Adresse électronique : Loris.Marchal@ens-lyon.fr.
- Page personnelle : http://perso.ens-lyon.fr/loris.marchal/

Langues

- Anglais : lu, écrit et parlé couramment.
- Allemand : notions.

Cursus Universitaire

sept. 2006 – jan. 2007 :
Visitor Assistant au laboratoire ACIS, Université de Floride (États-Unis).

oct. 2006 : Thèse de l'École Normale Supérieure de Lyon (mention *Très honorable*).
Rapporteurs :
Pierre FRAIGNIAUD (directeur de recherche CNRS, LRI)
Alix MUNIER KORDON (professeur, Université de Paris 12, LIP6).
Membres du jury :
Olivier BEAUMONT (maître de conférences, Univ. de Bordeaux 1, LABRI)
Franck CAPPELLO (directeur de Recherche INRIA, LRI)
Yves ROBERT (professeur ENS-Lyon, LIP)
Laurent VIENNOT (chargé de recherche INRIA, LRI)

2003-2006 : Thèse sous la direction d'Olivier BEAUMONT et d'Yves ROBERT au Laboratoire de l'Informatique du Parallélisme.

2002-2003 : Troisième année de Magistère d'Informatique et de Modélisation de l'École Normale Supérieure de Lyon (ENS-Lyon), mention bien.
Diplôme d'études approfondies (DEA) d'informatique fondamentale de l'ENS-Lyon, mention bien.

été 2002 : Stage de deux mois à l'Université de Californie, San Diego (États-Unis), sous la direction d'Henri CASANOVA : «Un modèle de réseau pour la simulation d'applications sur les grilles de calcul».

2001-2002 : Deuxième année de Magistère d'Informatique et de Modélisation de l'ENS-Lyon, mention bien.
Maîtrise d'informatique, université Claude Bernard-Lyon I, mention bien.

été 2001 : Stage de six semaines au Laboratoire d'Informatique de Marseille, sous la direction de Bruno DURAND et Enrico FORMENTI : «Automates cellulaires number-conserving : dynamique et décidabilité».

2000-2001 : Première année de Magistère d'Informatique et de Modélisation de l'ENS-Lyon, mention bien.
Licence d'informatique, université Claude Bernard-Lyon I, mention bien.

sept. 2000 : Intégration de l'École Normale Supérieure de Lyon.

1998-2000 : Classes préparatoires au lycée Aux Lazaristes (Lyon 5eme).

Activités pédagogiques

Enseignements dispensés

Depuis septembre 2004, je suis moniteur (titulaire d'une «allocation couplée»). J'ai donc effectué annuellement pendant ces trois dernière années exactement 64 heures (équivalent TD). La première année, j'ai effectué la moitié de ce service en classes préparatoires de l'Institut National des Sciences Appliquées (INSA) de Lyon et l'autre moitié au département d'Informatique de l'École Normale Supérieure (ENS) de Lyon. Les deux années suivantes, j'ai effectué la totalité de ces enseignements au département d'Informatique de l'ENS Lyon. Ces enseignements se répartissent comme suit :

- TD et TP du cours de *Algorithmique et architectures parallèles* en maîtrise (puis master 1) à l'ENS Lyon en 2004 et 2005. Le plan du cours et une grande partie des sujets des TD et TP avaient été élaborés par l'enseignant chargé du cours (Yves ROBERT) et les précédents chargés de TD (Arnaud LEGRAND et Olivier BEAUMONT). J'ai réalisé des sujets de travaux pratiques ainsi que des sujets de partiels et de devoirs et leurs corrigés.
- Pendant ma première année de monitorat, j'ai effectué les TD/TP du cours «Base de données et Algorithmique» à l'INSA de Lyon. Il s'agissait d'une introduction, d'une part au bases de données par l'utilisation de SQL, d'autre part à l'algorithmique par la programmation en Pascal, destinée à des étudiants de première année des classes préparatoires de l'INSA (niveau Licence 1). Le but était de faire découvrir les notions de bases de données et d'algorithmique au étudiants, et de les familiariser aux outils qu'ils auront à utiliser dans la suite de leur parcours. Le cours et les sujets de TD étaient préparés par les enseignants en charge du cours, de sorte que tous les groupes de TD aient les mêmes sujets.
- Pendant l'année 2005-2006, j'ai effectué les TD du cours «Algorithmique des Réseaux et des Télécoms», que je vais également effectuer pendant l'année 2006-2007. Ce cours était complètement nouveau. Le plan du cours a été conçu en collaboration étroite avec l'enseignant chargé du cours (Anne BENOIT). J'ai réalisé les sujets des TD et des TP, ainsi qu'un sujet de devoir. Comme le sujet de ce cours est assez peu classique, il a souvent fallu s'appuyer sur des articles de recherches pour concevoir les TD, comme par exemple pour l'étude des réseaux pair-à-pair ou du graphe du Web, ce qui demande plus de travail mais donne un attrait supplémentaire à ces enseignements.
- Durant le second semestre de l'année 2006-2007, je vais effectuer les TD du cours «Architecture, Réseaux et Système 2». Ce nouveau cours rassemble les précédents cours d'«Architecture des ordinateurs» de «Réseaux» et de «Systèmes d'exploitation» en deux semestres. Il faudra, en collaboration avec les deux autres chargés de TD et l'enseignant chargé du cours, adapter les sujets de TD déjà existants ou en concevoir de nouveaux et concevoir les évaluations.

Matières pouvant être enseignées

Je peux bien évidemment enseigner dans les matières que j'ai déjà eu l'occasion d'enseigner :

- algorithmique
- bases de donnée,

– algorithmique parallèle,
– algorithmique des réseaux,
– systèmes d'exploitation,
– réseaux,

et également dans des matières que je connais bien, sans les avoir déjà enseignées directement

– programmation (impérative, fonctionnelle ou orientée-objets)
– théorie des graphes

J'aimerais également enseigner dans des matières qui m'intéressent et sont reliées à mon programme de recherche, comme les réseaux pair-à-pair, l'algorithmique distribuée et l'optimisation combinatoire. À plus long terme, j'aimerai également étudier et enseigner des matières comme l'étude des algorithmes probabilistes ou randomisés.

D'autre part, la majorité de mes enseignements concerne le second cycle (c'est le cas de tous les moniteurs d'informatique à l'ENS Lyon) : dans ce contexte, l'essentiel de mon travail a été d'acquérir et de transmettre une connaissance approfondie du domaine enseigné, à des groupes d'étudiants souvent en groupe restreint et intéressés. Cependant, l'enseignement d'un module en premier cycle m'a permis de découvrir un aspect différent de l'enseignement, face à des étudiants n'ayant pas choisi l'informatique comme matière principale. J'ai également pu enrichir ces compétences à l'aide des stages proposés par le CIES.

Le tableau suivant récapitule les différents enseignements que j'ai dispensés au cours de ces trois dernières années.

Année	Intitulé	Public	Lieu	Élaboration des sujets	Heures (en équivalent TD)
2004-2005 2005-2006	Algorithmique et architectures parallèles	Master 1	ENS Lyon	20%	32 32
2004-2005	Bases de données et Algorithmique	Licence 1	INSA Lyon	0%	32
2005-2006 2006-2007	Algorithmique des Réseaux et des Télécoms	Master 1	ENS Lyon	100%	32 32
2006-2007	Architecture, Réseaux et Système	Licence 3	ENS Lyon	33%	32

Résumé de la thèse

Contexte de la thèse

L'objet de ce travail est d'étudier diverses techniques d'ordonnancement pour les plates-formes distribuées à grande échelle. Ce type de plates-formes, rassemblant des ressources de calculs distribuées à l'échelle d'un pays, ou d'un continent, voit son intérêt grandir, en particulier parce qu'il offre une alternative relativement peu coûteuse aux super-calculateurs monolithiques comme les récents «BlueGene» ou «Earth Simulator». De nombreux projets tentent de rassembler des ressources distribuées afin de créer des «grilles de calcul», comme le projet français Grid5000. La contrepartie du coût limité de ces plates-formes par rapport à des super-calculateurs est leur irrégularité et leur hétérogénéité, présente à tous les niveaux :

- au niveau matériel, les processeurs sont différents, et le réseaux d'interconnexion très complexe,
- au niveau logiciel, ces machines utilisent des systèmes d'exploitation, des librairies de calcul, et des intergiciels différents,
- et même au niveau administratif l'accès aux différentes ressources n'est souvent pas centralisé.

Notre objectif est d'étudier comment utiliser efficacement de telles plates-formes. Une première constatation s'impose : l'utilisation d'une plate-forme distribuée, puissante mais complexe, n'est justifiée que pour l'exécution d'une application nécessitant beaucoup de calculs. De plus, vu l'hétérogénéité de la plate-forme, on ne peut espérer y exécuter un code fortement couplé. Notre étude s'oriente donc vers des applications nécessitant beaucoup de calculs, mais relativement simples. On peut par exemple penser au modèle de tâches indépendantes, pour lequel l'application consiste en un grand nombre de tâches similaires et indépendantes. Ce modèle convient par exemple au projets de grilles participatives, utilisant les ordinateurs de personnes volontaires pendant leur absence ; on peut par exemple citer le projet BOINC, ou encore les très classiques SETI@home ou Einstein@home.

Nous voulons également étudier des applications qui ne sont pas aussi simples que des tâches indépendantes, quoique présentant une certaine régularité. Pour ceci, on s'intéresse aux communications impliquées par l'exécution d'une application distribuée. Ces communications peuvent souvent être rassemblées sous la forme de primitives de communications collectives, comme par exemple la diffusion de données : une des machines transmet à toutes les autres une copie d'une donnée qu'elle possède. Pour qu'une application mérite d'être exécutée sur une plate-forme à grande échelle, il est probable que le volume de données à communiquer soit important, on peut par exemple imaginer qu'une base de données de grande taille soit nécessaire à l'application, et doive ainsi être diffusée à toutes les machines avant le début du calcul. Pour effectuer ces transferts d'importants volumes de données, nous les découpons en une série d'un grand nombre de communications de taille plus modeste : la diffusion d'une donnée de grande taille sera alors transformée en une série d'un grand nombre de diffusion de messages de taille plus modeste, que l'on cherche à effectuer de façon pipelinée.

Régime permanent

Nous allons profiter de la régularité des applications pour optimiser leur temps d'exécution. Nous supposons que ces applications se composent d'un grand nombre d'actions répé-

titives : soit une série de tâches indépendantes, soit une série de communications collectives, soit même une série de graphes de tâches similaires. Plutôt que de chercher à minimiser le temps total d'exécution de l'application, nous tentons de maximiser le débit d'opérations effectuées pendant la phase de régime permanent. Ceci a un double avantage. D'abord, cette approche *simplifie* le problème d'optimisation : en négligeant les phases d'initiation et de terminaison des calculs, nous nous occupons uniquement des quantités moyennes de calcul et de communication allouées à chaque machine. Ensuite, cette approche est *efficace* : nous construisons des solutions optimales pour le régime permanent sous la forme d'ordonnancements périodiques, qui sont décrits de façon compacte et peuvent ainsi être implantés facilement.

Contributions

Communications collectives

Nous avons tout d'abord étudié l'optimisation du régime permanent pour les communications collectives. Nous avons proposé un cadre théorique d'étude de différentes primitives de communications collectives, sous l'angle de la maximisation du débit. Pour certaines d'entre elles, la diffusion, la distribution et la réduction de données, nous avons proposé des algorithmes efficaces qui construisent des ordonnancements périodiques de débit optimal. D'autres primitives de communications se sont révèlées plus difficiles : nous avons montré en particulier que pour la diffusion partielle et le calcul des préfixes, la recherche du débit optimal est un problème NP-complet. D'un point de vue expérimental, nous avons validé notre approche en la comparant aux méthodes existantes par simulation, et nous avons également proposé des heuristiques pour résoudre les problèmes d'optimisation difficiles. Nous avons également fourni, pour le problème de la diffusion, une implantation réelle, testée et validée sur la grille de recherche française Grid5000.

Applications multiples

Dans un deuxième temps, nous avons étendu notre étude du régime permanent à l'ordonnancement d'applications sur les grilles de calcul. Nous avons pris en compte le fait que plusieurs applications concurrentes se partagent les ressources disponibles. Nous avons étudié en particulier comment ordonnancer des applications consistant chacune en un grand nombre de tâches indépendantes, sur une plate-forme hiérarchique, en forme d'arbre. Nous avons montré que des stratégies non coopératives peuvent conduire à des situations de famine (une application est complètement défavorisée par rapport aux autres), et nous avons évalué des stratégies décentralisées, en les comparant à une stratégie optimale centralisée.

D'autre part, nous nous sommes également intéressés à la modélisation fine du réseau d'interconnexion des plates-formes distribuées. Dans ce modèle, nous avons montré que l'exécution équitable de débit optimal d'un ensemble d'applications divisibles est un problème NP-complet, et nous avons proposé des algorithmes heuristiques pour résoudre ce problème. En collaboration avec l'équipe RESO du LIP, nous nous sommes intéressés également à la réservation de ressources dans les réseaux d'interconnexion des grilles de calcul, en particulier pour les problèmes liés à la contention des connexions au niveau de points d'entrée et de sortie du cœur du réseau. Nous avons montré que le problème d'optimisation associé est NP-complet, et nous avons proposé des stratégies heuristiques pour le résoudre.

Autres problèmes d'ordonnancement

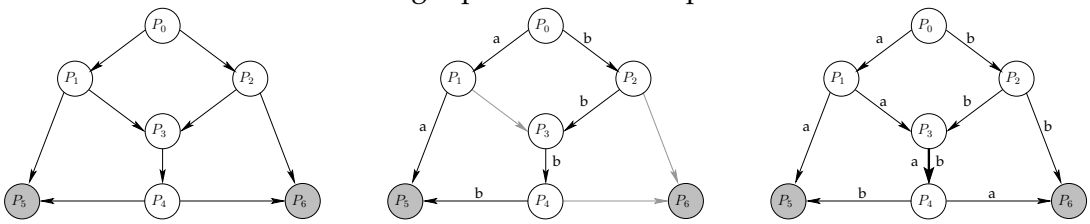
Au cours de cette thèse, nous avons également été amenés à nous intéresser à d'autres problèmes d'ordonnancement en marge du thème principal de l'optimisation du régime permanent. Nous nous sommes intéressés à l'ordonnancement de tâches indépendantes en présence de mémoire limitée : dans les autres travaux, nous supposons avoir à notre disposition sur chaque processeur une mémoire pouvant contenir une nombre illimité de tâches à traiter. Nous avons montré que si cette hypothèse n'est pas vérifiée, alors un grand nombre de problèmes d'ordonnancement que l'on savait résoudre, en particulier l'optimisation du débit, deviennent NP-complets. Par contre, nous présentons des résultats de simulation montrant que lorsqu'un seuil dans la taille de la mémoire est franchi, alors on peut espérer atteindre le débit optimal calculé sans contrainte de mémoire. Nous avons également apporté une contribution à la théorie des tâches divisibles, en étudiant l'ordonnancement de telles tâches avec messages de retour. En effet, on ne considère souvent que les données en entrée des calculs, tout en négligeant les résultats. Nous avons montré que prendre en compte les messages de retour contenant les résultats rend les problèmes d'ordonnancement significativement plus difficiles. En particulier, nous avons exhibé le premier cas (à notre connaissance) d'ordonnancement optimal sous un modèle sans latence où tous les processeurs ne prennent pas part au calcul.

Projet de recherche

Les travaux entrepris pendant ma thèse offrent de nombreuses perspectives de recherches, d'une part pour compléter les résultats obtenus, et d'autres part afin d'étudier des problèmes d'ordonnancement sur des topologies à grande échelle émergentes. En effet, si nous avons jusqu'à présent étudié les problèmes d'ordonnancement d'un point de vue statique et centralisé, les plates-formes d'exécution parallèles modernes se caractérisent par leur hétérogénéité, leur dispersion géographique et leur instabilité.

Diffusion restreinte (multicast)

Malgré la relaxation en régime permanent, le calcul du débit d'une diffusion restreinte reste un problème NP-complet, et c'est une des rares opérations dans ce cas. On peut donc penser que la relaxation appliquée n'est pas suffisante. En particulier, si on autorise les combinaisons entre blocs diffusés, le problème semble plus simple. Considérons par exemple le réseau illustré sur la figure de gauche ci-dessous, où chaque lien peut transporter un message par unité de temps, sans contention au niveau des nœuds. La source P_0 peut envoyer un débit de deux messages simultanément à la cible P_5 : a et b représentent ces deux messages sur la figure centrale. Il en est de même pour la cible P_6, puisque le réseau est symétrique. L'approche par programmation linéaire développée dans ma thèse pourrait être étendue naïvement ici, et prédirait un débit de diffusion restreinte égal lui aussi à deux messages par unités de temps. Cependant, comme on le remarque sur la figure de droite, cela nécessiterait que deux messages soient envoyés sur le lien central (P_3, P_4), ce qui n'est pas possible dans le modèle de communication choisi. Par contre, si P_3 calcule et envoie à P_4 le ou exclusif des deux messages (a XOR b), alors P_5 et P_6 peuvent reconstruire les deux messages a et b, on obtient bien un débit de deux messages par unités de temps.



La technique illustrée sur ce petit exemple pourrait se généraliser en utilisant les résultats de la théorie du Network Coding. D'après les travaux de Ahlswede[1] et de Koetter[2], il existe un codage qui permet d'atteindre le flot maximal pour tout couple source/destination dans un graphe quelconque. Il serait intéressant d'adapter ces résultats pour essayer de montrer que la diffusion restreinte est un problème polynomial en autorisant les combinaisons. Cependant, ceci ne nous fournira pas nécessairement de méthode utilisable en pratique. On pourrait alors s'inspirer des travaux sur l'utilisation de combinaisons aléatoires[3] : ceux-ci montrent que si chaque nœud choisit alétoirement les combinaisons de blocs qu'il va diffuser, alors on peut reconstruire le message total en chaque nœud cible avec forte probabilité. Si cette méthode peut être utilisée ici, elle permettrait d'éviter un contrôle centralisé et coûteux des combinaisons. Si technique paraît indispensable pour la diffusion restreinte, elle

[1] Ahlswede et al., *Network information flow* IEEE Transactions on Information Theory, 2000.
[2] Ho et al. *An information theoretic view of network management*, INFOCOM, 2003.
[3] Gkantsidis et al. *Network coding for large scale content distribution*, In INFOCOM, 2005.

peut également utile pour la diffusion totale, en évitant la construction centralisée d'arbres de diffusion concurrents.

Un des problèmes qui peut limiter l'utilisation de combinaisons de blocs est la puissance de calcul nécessaire pour (i) créer de nouvelles combinaisons (en particulier, même les nœuds qui ne sont pas des destinations doivent créer des combinaisons, et on voudrait leur éviter une grosse charge de calcul pour une opération dont il ne font pas partie) et (ii) reconstituer le message initial à partir de combinaisons, ce qui demande une inversion de matrice de grande taille, à coefficients dans des corps finis de grande taille.

Réplication de tâches

Un autre problème qui demeure NP-complet avec la relaxation en régime permanent est la détermination du débit optimal du calcul des préfixes parallèles. Dans cette variante de l'opération de réduction, on ne cherche plus simplement à calculer le résultat de l'opération $v_1 \oplus v_1 \oplus \cdots \oplus v_N$ (où $\oplus$ est un opérateur associatif, et les valeurs v_i sont réparties sur la plate-forme), mais également le résultat intermédiaires $v_1 \oplus v_1 \oplus \cdots \oplus v_i$ sur le processeur P_i, pour toute valeur de i. Alors que pour la réduction, il n'y avait pas lieu de dupliquer les calculs, il peut maintenant être intéressant de ne pas attendre le résultat du processeur P_i pour calculer celui de P_{i+1}. Cette liberté supplémentaire rend le problème plus difficile, mais permet d'optimiser les performances. Il en est de même pour les graphes de tâches : pour un graphe de tâches de type *fork*, où un tâche initiale T_0 crée de nombreux fichiers devant être traités par des tâches subalternes $T_1, \ldots, T_K$. Si la taille de ces fichiers est importante, il peut être judicieux de répliquer la tâches T_0 sur autant les machines où sont exécutés les tâches $T_1, \ldots, T_K$; on évite ainsi le transfert de nombreux fichiers de grande taille.

Décentralisation et gestion de la dynamicité

Une autre limitation de notre étude des communications collectives en vue de son utilisation sur des plates-formes à grande échelle est son aspect centralisé : il nous faut en effet rassembler les caractéristiques de la plate-forme sur un seul nœud qui prend les décisions d'ordonnancement en conséquence. Sur une plate-forme à grande échelle, ceci nécessite de nombreuses communications longue-distance préalables à la construction de l'ordonnancement. Un autre argument en faveur de la décentralisation est la nécessité de réagir à des petites variations de l'environnement : si une congestion se produit sur un lien de communication et que sa capacité diminue, il faut attendre que le nœud en charge de l'ordonnancement soit prévenu, qu'il modifie l'ordonnancement, puis que cette décision atteigne enfin les paquets attendant d'être transmis sur ce lien. Au contraire, avec une prise de décision locale, on peut espérer qu'un problème de congestion puisse être résolu plus rapidement, en utilisant d'autres routes pour contourner le lien défectueux.

Nous avons entrepris une première étude de l'étude d'un ordonnancement décentralisé et dynamique en adaptant un algorithme d'Awerbuch et Leighton pour ordonnancer plusieurs applications consistant en des tâches indépendantes. Une fois calculé le débit de chaque application, cet algorithme permet de trouver de façon décentralisée une solution réalisant ce débit. De plus cette solution est stable : si les capacités des liens varient, l'algorithmique converge vers une solution, pourvu que le débit initial soit encore réalisable.

Ceci ne concerne que la phase d'ordonnancement, après que les débits des applications aient été calculés. Il serait également intéressant de pouvoir distribuer les deux étapes de

l'ordonnancement. Comme dans le cas de la diffusion restreinte, les algorithmes probabilistes ou randomisés sont peut-être la bonne solution dans ce contexte.

Topologies

Dans la plupart des travaux précédents, nous supposons connaitre parfaitement le graphe de communication, et de pouvoir contrôler tous les nœuds de la plate-forme : même dans la diffusion restreinte (multicast), on suppose que les nœuds qui ne sont pas des cibles de la diffusion participent en routant les messages de la façon souhaitée par l'ordonnancement. Cette hypothèse est valable pour des réseaux locaux et/ou privés, mais devient de moins en moins pertinente lorsqu'on s'intéresse à des réseaux à grande échelle, surtout lorsque les liens de communication sont partagés entre de nombreux utilisateurs. Nous avons d'ailleurs proposé une modélisation simple du partage de bande-passante dans les liens longue-distance, pour ordonnancer des applications divisibles sur une plate-forme utilisant plusieurs sites de calcul reliés par de tels liens. Cette modélisation convient pour une plate-forme simple constituée de sites (grappes de calcul) reliés par des liens longue-distance, mais elle ne permet pas d'utiliser les techniques d'ordonnancement de communications collectives que nous avons développé (le problème d'ordonnancement simple étudié sur cette plate-forme est déjà NP-complet).

On pourrait également essayer de reconstruire le graphe de plate-forme que nous utilisons. Cependant, aquérir une information complète de la topologie est une tâche longue et ardue. Pour connaître précisément le graphe de la plate-forme, il est a priori nécessaire de vérifier pour toute paire de routes ($P_i \rightarrow P_j$ et $P_k \rightarrow P_l$) si des transferts concurrents sur ces deux routes interfèrent. De plus, il faut être capable de mesurer ou d'estimer la bande-passante de chaque lien de communication dans le graphe ainsi construit. Plutôt que d'utiliser un tel graphe «exhaustif» de la plate-forme, il serait intéressant de pouvoir obtenir une modélisation approchée mais plus «utilisable» de la plate-forme, en ne cherchant pas à modéliser précisément les parties du réseau qui nous sont inaccessibles : le réseau interne (longue-distance) est souvent surdimensionné ; on peut dans ce cas se contenter d'une analyse locale.

Il serait enfin très intéressant de se tourner vers d'autres topologies naturellement adaptées à des environnement à grande-échelle, comme les topologies pair-à-pair. Celles-ci ont fait leurs preuves pour le partage de données de grande taille, sur des environnements distribués à grande échelle. Un réseau pair-à-pair serait particulièrement adapté pour gérer des environnements de calcul participatif (comme les projets seti@home, «Berkeley Open Infrastructure for Network Computing» ou «World Community Grid»). En général, les réseaux pair-à-pair ont de bonnes propriétés de tolérance aux pannes, de stabilité, et de passage à l'échelle, grâce à l'utilisation d'un réseau virtuel (overlay) dont la topologie est bien connue. Les opérations possibles sur ces réseaux se limitent souvent à la recherche de donnée ou la diffusion de données. De même la description des pairs est très simple : ils sont le plus souvent tous équivalents, et quelque fois muni de bande-passante d'entrée et de sortie. Il faudrait donc adapter ces techniques afin de pouvoir concevoir des ordonnancements sur ces plates-formes, en particulier pour des tâches indépendantes, qui constituent une application naturelle pour ce type de plates-formes distribuées à grande échelle.

Ordonnancement de machines virtuelles

Lors de mon séjour post-doctoral au laboratoire ACIS de l'université de Floride, je suis amené à considérer des problèmes d'ordonnancement pour machines virtuelles. On considère en général un ensemble de machines virtuelles exécutées sur un ensemble de processeurs, chaque processeur pouvant héberger une ou plusieurs machines virtuelles, à l'aide de logiciels de virtualisation tels que WMWare ou Xen. On cherche alors à équilibrer la charge des processeurs physiques en élaborant le placement des machines ou en migrant des machines lorsque la charge varie, ce que permet par exemple Xen. On rencontre deux types d'applications, qui nécessitent aussi bien un placement initial que du rééquilibrage de charge :

- Une application des machines virtuelles est dans les services Web : une machine virtuelle est affecté au traitement d'un service (recherche dans une base de données, système de réservation de places, génération de pages Web dynamiques,...). Des serveurs hébergent ces différents machines virtuelles, dont l'activité varient en fonction du trafic qu'elles rencontrent. On peut par exemple citer l'exemple d'Amazon, qui met ses serveurs à disposition pour héberger des machines virtuelles dans son «Amazon Elastic Compute Cloud».
- Nous nous intéressons plus particulièrement à l'utilisation de machines virtuelles pour du traitement de signal avec des contraintes de temps réel. Différents traitements, représentés par différentes machines virtuelles, sont appliqués au signaux reçus. Le résultat des ces traitements conditionnent les traitements futurs et leur contrainte de temps réel : par exemple, un filtre qui se révèle calculer une bonne prédiction du signal peut se voir attribuer un rôle prépondérant, on voudra alors l'exécuter plus rapidement que les autres, soit en isolant la machine virtuelle qui le calcule sur un nœud physique dédié, soit en lui attribuant une partie importante des ressources d'un nœud physique partagé.

Pour ces deux applications, il faut être capable de configurer dynamiquement l'allocation des différentes machines virtuelles. Les travaux précédents se sont surtout intéressés à offrir la possibilité technique de migrer les machines virtuelles, en réduisant le coût de cette migration (temps d'inactivité et charge sur le réseau d'interconnexion).

Autres activités scientifiques

Exposés et séminaires

Au cours de ces trois années de thèse, j'ai été amené à intervenir dans divers groupes de travail, en plus des conférences dans lesquelles j'ai présentés mes travaux.

- En avril 2004, j'ai présenté à l'université du Colorado (Colorado State University) un exposé intitulé «The validation problem on distributed heterogeneous platforms : simulation, modeling, observation... ?»
- En juin, j'ai présenté un exposé intitulé «Comparaison des stratégies centralisées et distribuées pour l'ordonnancement d'applications concurrentes sur plate-forme maître-esclave hétérogène » à la réunion du projet ALPAGE de l'ANR Masses de données.
- En septembre 2006, j'ai présenté deux exposé d'introduction aux techniques d'ordonnancement pour les membres du laboratoire ACIS de l'université de Floride : «Traditionnal and Divisible Load Scheduling slides» et «Steady-State Scheduling and Simulation for Grid Computing slides».
- J'ai également présenté de nombreux séminaires au sein de mon équipe de recherche Graal (environ 6 en trois ans).

Collaborations

J'ai effectué un stage de deux mois, avant de commencer ma thèse, à l'Université de Californie, San Diego (UCSD). J'y ai travaillé, sous la direction d'Henri CASANOVA, à la conception d'une modélisation réaliste du réseau dans un simulateur d'applications distribuées sur la grille nommée SimGrid.

Cette collaboration avec San Diego s'est poursuivie dans le cadre l'équipe associée Inria I-ARTHUR. J'ai tout d'abord été amené à collaborer de nouveau avec Henri CASANOVA ainsi qu'avec son étudiant Yang YANG sur un projet d'ordonnancement de tâches divisibles sur les grilles de calcul. J'ai également collaboré avec deux autres chercheurs de l'université de San Diego, Larry CARTER et Jeanne FERRANTE, dans le cadre d'une étude des ordonnanceurs centralisés et décentralisés sur les plates-formes distribuées organisées en arbres. Ces diverses collaborations se sont traduites par les publications [19, 12, 2, 8].

En collaboration avec Pascale VICAT-BLANC PRIMET et Jingdi ZENG de l'équipe RESO de mon laboratoire, nous avons étudié le partage de bande passante entre requêtes pour le cœur des réseaux des grilles de calcul, ce qui a donné lieu aux publications [11, 7]

J'ai également collaboré avec Étienne RIVIÈRE et Anne-Marie KERMARREC de l'IRISA (Rennes) sur un projet de réseau pair-à-pair s'appuyant sur une topologie utilisant un diagrame de Voronoï des points correspondant aux caractéristiques des objets stockés. Cette collaboration a débouché sur la publication [21].

Encadrement

Durant ma thèse, j'ai participé à l'encadrement de Véronika REHN, lors de deux stages :

- stage de master 1 en 2005 : deux mois, 50% d'encadrement, co-encadré avec Yves ROBERT
- stage de master 2 (DEA) en 2006 : six mois, 30% d'encadrement, co-encadré avec Frédéric Vivien et Yves Robert.

Cette étudiante commence cette année une thèse sous la direction d'Anne BENOIT et Yves ROBERT.

Relectures

J'ai effectué des relectures scientifiques pour différentes conférences et revues internationales :

- conférences :
  - Workshop on Practical Aspects of high-level Parallel Programming (PAPP) 2006
  - International Parallel & Distributed Processing Symposium (IPDPS) 2006
  - Grid 2005 (conférence satellite de SuperComputing 2005)
  - International Conference on Parallel and Distributed Systems (ICPADS 2006)
- revues et journaux :
  - International Journal of High Performance Computing and Applications
  - Journal of Parallel Computing
  - Transactions on Parallel and Distributed Systems
  - Journal on Parallel and Distributed Computing

Liste des publications

Revues internationales avec comité de lecture

[1] O. Beaumont, L. Marchal et Y. Robert. «Complexity results for collective communications on heterogeneous platforms». *Int. Journal of High Performance Computing Applications* (2006, à paraître).

[2] L. Marchal, Y. Yang, H. Casanova et Y. Robert. «Steady-state scheduling of multiple divisible load applications on wide-area distributed computing platforms». *Int. Journal of High Performance Computing Applications* (2006, à paraître).

[3] A. Legrand, L. Marchal et Y. Robert. «Optimizing the steady-state throughput of scatter and reduce operations on heterogeneous platforms». *Journal of Parallel and Distributed Computing* **65**, numéro 12 (2005), 1497–1514.

[4] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Steady-state scheduling on heterogeneous clusters». *Int. J. of Foundations of Computer Science* **16**, numéro 2 (april 2005).

[5] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Pipelining broadcasts on heterogeneous platforms». *IEEE Trans. Parallel Distributed Systems* **16**, numéro 4 (2005).

[6] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Scheduling strategies for mixed data and task parallelism on heterogeneous clusters». *Parallel Processing Letters* **13**, numéro 2 (2003).

Conférences internationales avec comité de lecture

[7] L. Marchal, P. V.-B. Primet, Y. Robert et J. Zeng. «Optimal Bandwidth Sharing in Grid Environment». Dans *15th International Symposium on High Performance Distributed Computing (HPDC 2006* (2006), IEEE Computer Society Press.

[8] O. Beaumont, L. Carter, J. Ferrante, A. Legrand, L. Marchal et Y. Robert. «Centralized versus distributed schedulers for multiple bag-of-task applications». Dans *International Parallel and Distributed Processing Symposium IPDPS'2006* (2006), IEEE Computer Society Press.

[9] O. Beaumont, L. Marchal, V. Rehn et Y. Robert. «FIFO scheduling of divisible loads with return messages under the one-port model». Dans *HCW'2006, the 15th Heterogeneous Computing Workshop* (2006), IEEE Computer Society Press.

[10] O. Beaumont, L. Marchal et Y. Robert. «Scheduling divisible loads with return messages on heterogeneous master-worker platforms». Dans *International Conference on High Performance Computing HiPC'2005* (2005), volume 3769 des *LNCS*, Springer Verlag, pp. 498–507.

[11] L. Marchal, P. V.-B. Primet, Y. Robert et J. Zeng. «Optimizing Network Resource Sharing in Grids». Dans *IEEE Global Telecommunications Conference (Gloebcom'2005)* (2005, to appear), IEEE Computer Society Press.

[12] L. Marchal, Y. Yang, H. Casanova et Y. Robert. «A realistic network/application model for scheduling divisible loads on large-scale platforms». Dans *International Parallel and Distributed Processing Symposium IPDPS'2005* (2005), IEEE Computer Society Press.

[13] O. Beaumont, L. Marchal et Y. Robert. «Broadcast Trees for Heterogeneous Platforms». Dans *International Parallel and Distributed Processing Symposium IPDPS'2005* (2005), IEEE Computer Society Press.

[14] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Independent and Divisible Tasks Scheduling on Heterogeneous Star-schaped Platforms with Limited Memory». Dans *13th Euromicro Conference on Parallel, Distributed and Network-based Processing PDP'2005* (2005), IEEE Computer Society Press, pp. 179–186.

[15] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Pipelining broadcasts on heterogeneous platforms». Dans *International Parallel and Distributed Processing Symposium IPDPS'2004* (2004), IEEE Computer Society Press.

[16] A. Legrand, L. Marchal et Y. Robert. «Optimizing the steady-state throughput of scatter and reduce operations on heterogeneous platforms». Dans *APDCM'2004, 6th Workshop on Advances in Parallel and Distributed Computational Models* (2004), IEEE Computer Society Press.

[17] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Complexity results and heuristics for pipelined multicast operations on heterogeneous platforms». Dans *Proceedings of the 33rd International Conference on Parallel Processing (ICPP'04)* (2004), IEEE Computer Society Press.

[18] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Steady-state scheduling on heterogeneous clusters : why and how ?». Dans *6th Workshop on Advances in Parallel and Distributed Computational Models APDCM 2004* (2004), IEEE Computer Society Press.

[19] H. Casanova, A. Legrand et L. Marchal. «Scheduling Distributed Applications : the SimGrid Simulation Framework». Dans *Proceedings of the third IEEE International Symposium on Cluster Computing and the Grid (CCGrid'03)* (may 2003).

[20] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Assessing the impact and limits of steady-state scheduling for mixed task and data parallelism on heterogeneous platforms». Dans *HeteroPar'2004 : International Conference on Heterogeneous Computing, jointly published with ISPDC'2004 : International Symposium on Parallel and Distributed Computing* (2004), IEEE Computer Society Press.

Conférences nationales avec comité de lecture

[21] O. Beaumont, A.-M. Kermarrec, L. Marchal et Étienne Rivière. «Voronet, un réseau objet-à-objet sur le modèle petit-monde». Dans *CFSE'5 : Conférence Française sur les Systèmes d'Exploitation* (2006).

Rapports de recherche

[22] L. Marchal, V. Rehn et F. Vivien. «Scheduling and data redistribution strategies on star platforms». Research report, LIP, ENS Lyon, France, juin 2006.

[23] O. Beaumont, A.-M. Kermarrec, L. Marchal et E. Rivière. «VoroNet : A scalable object network based on Voronoi tessellations». Research report, LIP, ENS Lyon, France, février 2006.

[24] O. Beaumont, L. Marchal, V. Rehn et Y. Robert. «FIFO scheduling of divisible loads with return messages under the one-port model». Research report, LIP, ENS Lyon, France, octobre 2005.

[25] O. Beaumont, L. Carter, J. Ferrante, A. Legrand, L. Marchal et Y. Robert. «Scheduling multiple bags of tasks on heterogeneous master-worker platforms : centralized versus distributed solutions». Research report, LIP, ENS Lyon, France, septembre 2005.

[26] L. Marchal, P. V.-B. Primet, Y. Robert et J. Zeng. «Scheduling network requests with transmission window». Research report, LIP, ENS Lyon, France, juillet 2005.

[27] O. Beaumont, L. Marchal et Y. Robert. «Scheduling divisible loads with return messages on heterogeneous master-worker platforms». Research report, LIP, ENS Lyon, France, mai 2005.

[28] L. Marchal, P. V.-B. Primet, Y. Robert et J. Zeng. «Optimizing Network Resource Sharing in Grids». Research report, LIP, ENS Lyon, France, mars 2005. Also available as INRIA Research Report RR-5523.

[29] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Complexity results and heuristics for pipelined multicast operations on heterogeneous platforms». Research report, LIP, ENS Lyon, France, février 2004. Also available as INRIA Research Report RR-5123.

[30] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Steady-State Scheduling on Heterogeneous Clusters : Why and How ?». Research report, LIP, ENS Lyon, France, mars 2004.

[31] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Assessing the impact and limits of steady-state scheduling for mixed task and data parallelism on heterogeneous platforms». Research report, LIP, ENS Lyon, France, avril 2004. Also available as INRIA Research Report RR-5198.

[32] L. Marchal, Y. Yang, H. Casanova et Y. Robert. «A realistic network/application model for scheduling divisible loads on large-scale platforms». Research report, LIP, ENS Lyon, France, avril 2004. Also available as INRIA Research Report RR-5197.

[33] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Independent and Divisible Task Scheduling on Heterogeneous Star-shaped Platforms with Limited Memory». Research report, LIP, ENS Lyon, France, avril 2004. Also available as INRIA Research Report RR-5196.

[34] O. Beaumont et L. Marchal. «Pipelining broadcasts on heterogeneous platforms under the one-port model». Research Report RR-2004-32, LIP, ENS Lyon, France, juillet 2004.

[35] O. Beaumont, L. Marchal et Y. Robert. «Broadcast Trees for Heterogeneous Platforms». Research Report RR-2004-46, LIP, ENS Lyon, France, novembre 2004.

[36] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Optimal algorithms for the pipelined scheduling of task graphs on heterogeneous systems». Research Report RR-2003-29, LIP, ENS Lyon, France, avril 2003. Also available as INRIA Research Report RR-4870.

[37] A. Legrand, L. Marchal et Y. Robert. «Optimizing the steady-state throughput of scatter and reduce operations on heterogeneous platforms». Research Report RR-2003-33, LIP, ENS Lyon, France, juin 2003. Also available as INRIA Research Report RR-4872.

[38] O. Beaumont, A. Legrand, L. Marchal et Y. Robert. «Optimizing the steady-state throughput of broadcasts on heterogeneous platforms heterogeneous platforms». Research report, LIP, ENS Lyon, France, juin 2003. Also available as INRIA Research Report RR-4871.

[39] H. Casanova et L. Marchal. «A Network Model for Simulation of Grid Application». Research Report RR-2002-40, LIP, ENS Lyon, France, octobre 2002. Also available as INRIA Research Report RR-4596.