Dossier de candidature
à un poste de CR2
concours 07/03

Loris Marchal

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

Contenu du dossier :

Curiculum Vitæ

Loris Marchal

– Né le 22 février 1980 à Lyon 6ème (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/

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*).

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).

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 [21, 14, 3, 10].

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 [13, 9]

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 [23].

Enseignement

Depuis septembre 2004, je suis moniteur (titulaire d'une « allocation couplée »). J'ai donc effectué annuellement pendant ces trois dernière années 64 heures d'enseignement (équivalent TD). Ces enseignements ont été effectué en partie en classes préparatoires de l'Institut National des Sciences Appliquées (INSA) de Lyon et en partie au département d'Informatique de l'École Normale Supérieure (ENS) de Lyon. J'ai enseigné dans plusieurs disciplines :

- Base de données,
- Algorithmique,

- Algorithmique et architectures parallèles,
- Algorithmique des réseaux et de télécommunications,
- Architecture, réseaux et systèmes.

Pour la plupart de ces enseignements, j'ai participé à l'élaboration des sujets de TD ainsi qu'aux évaluations.

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

Titre de la thèse, résumé et composition du jury

Titre et résumé

Communications collectives et ordonnancement en régime permanent
sur plates-formes hétérogènes.

Résumé

Les travaux présentés dans cette thèse concernent l'ordonnancement pour les plates-formes hétérogènes à grande échelle. Nous nous intéressons principalement aux opérations de communications collectives comme la diffusion de données, la distribution de données ou la réduction. Nous étudions ces problèmes dans le cadre de leur régime permanent, en optimisant le débit d'une série d'opérations de communications, en vue d'obtenir un ordonnancement asymptotiquement optimal du point de vue du temps d'exécution total. Après avoir présenté un cadre général d'étude qui nous permet de connaître la complexité du problème pour chaque primitive, nous développons, pour le modèle de communication un-port bidirectionnel, une méthode de résolution pratique fondée sur la résolution d'un programme linéaire en rationnels. Cette étude du régime permanent est illustrée par des expérimentations sur Grid5000 et se prolonge vers l'ordonnancement d'applications multiples sur des grilles de calcul.

Mots-clés : Algorithmique parallèle, ordonnancement, hétérogénéité, régime permanent, communications collectives, optimisation combinatoire.

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, Université de Bordeaux 1, LaBRI)
- Franck CAPPELLO (directeur de Recherche INRIA, LRI)
- Pierre FRAIGNIAUD (directeur de recherche CNRS, LRI) ;
- Alix MUNIER KORDON (professeur, Université de Paris 12, LIP6).
- Yves ROBERT (professeur ENS-Lyon, LIP)
- Laurent VIENNOT (chargé de recherche INRIA, LRI)

Rapport sur les travaux effectués

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.

Séjour post-doctoral

Pendant le permier semestre de l'année scolaire 2006-2007, j'ai effectué un court séjour post-doctoral au laboratoire ACIS de l'université de Floride, sous la direction de Jose Fortes.

Le but des recherches menées lors de séjour est d'ordonnancer une application particulière, ayant des contraintes de temps-réel, sur un environnement distribué organisé comme un ensemble de machines virtuelles. Pour ceci, nous reposons sur l'architecture In-Vigo, développée au sein du laboratoire ACIS, qui permet de rassembler dynamiquement un ensemble de ressources virtuelles pour une application. L'application que nous visons est une interface cerveau-machine : elle reçoit périodiquement des signaux provenant du cerveau d'un animal et doit produire une commande moteur envoyée à un robot. Le traitement des données consiste en un la collaboration d'un grand nombre de « processus experts » : chacun de ces experts calcule une réponse (une commande moteur) et ces réponses sont rassemblées en tenant compte de l'importance et de la crédibilité de chaque expert à un instant donné. Il existe également un processus d'apprentissage qui permet à chaque expert de raffiner ses réponses en mettant à jour ses paramètres.

Avant de considérer les problèmes d'ordonnancement liés à cette application, il nous a fallu optimiser le traitement nécessaire à un expert. En particulier, nous avons cherché à améliorer la phase d'apprentissage. Plutôt que d'attendre que toutes les données nécessaires à cet apprentissage soient rassemblées (soit une période d'environ 10 secondes), nous avons proposé que la mise à jour des paramètres d'un expert se fasse « à la volée ». Pour ceci, nous nous sommes appuyés sur une littérature abondante en traitement du signal, particulièrement sur les filtres adaptatifs et le calcul des moindres carrés. L'adaptation de ces algorithmes nous permet d'effectuer l'apprentissage dans le temps imparti et de garantir la stabilité numérique du résultat. Ce travail était un préliminaire nécessaire à l'étude de l'exécution parallèle de l'application que nous étudions maintenant. Comme l'étude des ordonnancements de machines virtuelles est un travail en cours qui soulève de nombreuses

problèmatiques que je compte étudier, ces questions sont développées dans la partie « projet de recherche » de ce dossier.

Projet de recherche

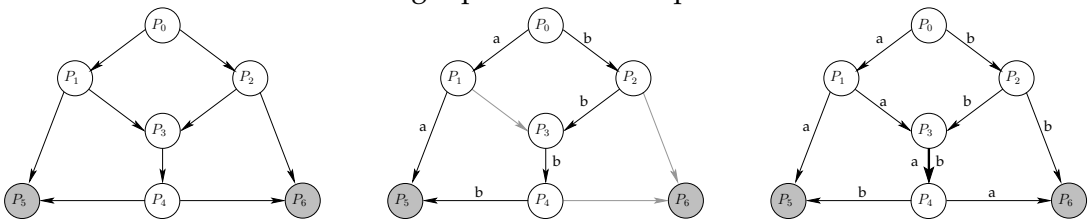
Les travaux entrepris pendant ma thèse offrent de nombreuses perspectives de recherches pour compléter et élargir les résultats obtenus. Ils soulèvent également des problématiques de recherches plus vastes que j'aimerais étudier à plus long terme. Mon projet de recherche est donc partagé entre les perspectives à court terme et à moyen terme.

Court Terme

Parmi les extensions que j'aimerais développer dans le court terme, on peut distinguer trois axes principaux.

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 unité 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 unité 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 polynômial en autorisant les combinaisons (en tenant compte des coûts de calcul associés aux 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 de façon aléatoire les combinaisons de blocs qu'il va diffuser, alors on peut

[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.

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 cette technique paraît indispensable pour la diffusion restreinte, elle peut également être 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 dimension, à coefficients dans des corps finis de grande taille. Il s'agit alors d'étudier le compromis entre le bénéfice de cette approche et la surcharge de calcul nécessaire.

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 intermédiaire $v_1 \oplus v_1 \oplus \cdots \oplus v_i$ 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 les machines où sont exécutées les tâches $T_1, \ldots, T_K$; on évite ainsi le transfert de nombreux fichiers de grande taille. Nous voulons donc nous intéresser à l'adaptation à des stratégies autorisant la réplication des heuristiques d'ordonnancement pour les graphes de tâches en régime permanent.

Topologies pour le calcul distribué

Dans la plupart des travaux précédents, nous supposons connaître 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 développées (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, acquérir 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 sur-dimensionné ; 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 ou la diffusion de données. De même la description des pairs est très simple : ils sont le plus souvent tous équivalents, et quelque fois munis de bande-passante d'entrée et de sortie. Il faudrait donc adapter ces topologies virtuelles afin de pouvoir concevoir des ordonnancements sur ces plates-formes, en particulier pour l'exécution de tâches indépendantes, qui constituent une application naturelle pour ce type de plates-formes distribuées à grande échelle.

Moyen Terme

À plus long terme, mon projet de recherche s'articule autour de trois thèmes, chacun en coopération soit avec une équipe du Laboratoire de l'Informatique du Parallélisme (ENS Lyon), soit avec un laboratoire extérieur.

Gestion de la dynamicité et algorithmes décentralisés

Nous avons jusqu'à présent étudié les plates-formes hétérogènes et conçu des algorithmes statiques pour ces plates-formes. Nous savons relativement bien modéliser et utiliser de façon performante ces plates-formes à l'aide de divers outils théoriques, comme la modélisation des applications en tâches indépendantes, en tâches divisibles ou encore l'ordonnancement en régime permanent.

Cependant ces algorithmes sont valides sur des plates-formes dont les caractéristiques ne changent pas au cours du temps. Or les plates-formes réelles que nous souhaitons utiliser ne vérifient pas cette hypothèse : si les plates-formes de taille moyenne (de type grappes de calcul) sont à peu près statiques, les plates-formes à grande échelle (grappes de grappes, grilles de calcul) sont fondamentalement dynamiques. Les performances des liens de communication ne sont pas constantes car il existe sur ces liens un trafic extérieur que l'on ne

maîtrise pas et qui est difficile à modéliser. La vitesse des unités de calcul est également dynamique car affectée par une charge extérieure. Sur de telles plates-formes, il est également réaliste de considérer que certaines unités de calcul vont subir de défaillances pendant leur utilisation. Il nous faut donc concevoir des algorithmes robustes aux fluctuations et tolérants aux pannes, qui puissent s'adapter à des ressources dynamiques, voire volatiles.

Nous avons parfois considéré que les algorithmes périodiques élaborés par notre approche en régime permanent étaient dynamiques : on peut enregistrer les performances des ressources (réseaux et processeurs) pendant l'exécution d'une période, puis utiliser cette connaissance pour l'élaboration de la période suivante. Cette approche souffre de plusieurs inconvénients :

- Il est délicat de changer le motif d'ordonnancement d'une période à l'autre : pour les tâches qui sont en cours d'exécution dans la plate-forme, il faut choisir soit de conserver le motif précédent (mais prendre l'existence de ces tâches en compte lors du calcul du motif suivant), soit de changer le devenir d'une tâche (la fin de son exécution et le routage de ses fichiers de sortie). Ceci est difficile pour des tâches indépendantes et encore plus pour des graphes de tâches.
- Cette approche consiste à rassembler en un nœud responsable de l'ordonnancement les informations relatives aux ressources de la plate-forme. Il est donc possible qu'au moment où l'on utilise les mises à jour des caractéristiques des liens et des unités de calculs pour calculer le motif de la nouvelle période, celles-ci soient déjà obsolètes.
- Pour avoir des performances les plus proches du résultat optimal théorique, il faut utiliser une période de grande taille, ce qui diminue d'autant la réactivité de l'algorithmique aux changements dans la plate-forme.

Avant de concevoir des algorithmes pour les plates-formes dynamiques, un premier travail de modélisation est nécessaire. Il faut modéliser la variation de charge, le comportement des utilisateurs sur des machines partagées, le trafic du réseau, etc. La plupart des outils disponibles pour modéliser des ressources dynamiques ne sont pas adaptés à notre approche. D'autre part, les lois de distribution de probabilités généralement utilisées sont des lois de Poisson ou plus souvent de distribution exponentielle qui ont l'avantage d'être manipulables. Cependant, les récents travaux de Feitelson (http://www.cs.huji.ac.il/~feit/wlmod/) sur la modélisation de la charge des plates-formes et du comportement des utilisateurs montrent que d'autres distributions sont plus appropriées.

Outre la modélisation de l'instabilité de la plate-forme, il nous faut également mettre au point une métrique adaptée pour comparer les performances de différents ordonnancements. On peut imaginer qu'un algorithme robuste est un algorithme performant en moyenne, étant données les distributions de probabilités des performances des ressources ; au contraire, ce peut être un algorithme garanti sur tout un domaine de conditions possibles.

De là, nous pouvons imaginer concevoir deux types d'algorithmes. On peut choisir d'élaborer des algorithmes résistants, qui ne sont pas sensibles aux variations de charge et les absorbent naturellement. Dans ce cadre là, un algorithme statique (ne modifiant pas ses décisions d'ordonnancement au cours du temps) peut être adapté à une plate-forme dynamique, probablement si les variations de performances sont limitées. Pour des variations plus importantes ou pour faire face à des pannes, il est nécessaire de concevoir des algorithmes réactifs, qui construisent l'ordonnancement à la volée ou remettent en cause l'ordonnancement initial.

Une approche pour aboutir à des algorithmes s'adaptant aux plates-formes dynamiques est de concevoir des algorithmes résistant localement aux variations de performances, afin

d'obtenir une garantie globale. On peut par exemple penser aux algorithmes utilisant une stratégie d'équilibrage local qui permet d'obtenir un équilibre global. C'est ce que nous avons commencé à faire en utilisant l'algorithme d'Awerbuch et Leighton qui, par des équilibrages locaux sur des files d'attente, offre une garantie de convergence sous des conditions dynamiques.

On peut ensuite imaginer ajouter à l'ordonnanceur distribué une propagation des informations de variations de performances, afin d'améliorer la vitesse de convergence, en s'appuyant éventuellement sur une vision hiérarchique (ou au moins organisée) de la plate-forme : une information détectée au sein d'une grappe de calcul sera propagée au reste de la plate-forme pour que les ordonnanceurs locaux s'y adaptent. Si on peut espérer concevoir un schéma expérimental de validation de tels algorithmes, il est sans doute beaucoup plus difficile de prédire et de garantir le comportement des ordonnancements, même si on peut espérer obtenir des résultats théoriques dans des cas simples.

L'intérêt des algorithmes adaptatifs décentralisés est double : ils permettent de garantir des performances acceptables malgré la dynamicité des plates-formes et ils offrent des propriétés de passage à l'échelle, qui les autorisent à être mis en œuvre sur de larges plates-formes.

L'étude des topologies pair-à-pair pour le calcul distribué proposée dans la partie « Court terme » trouve naturellement son prolongement dans l'étude des algorithmes décentralisés et robustes : nous espérons que les connaissances aquises sur ces topologies et les structures de données distribuées nous permettrons de développer des algorithmes dynamiques et performants.

Ce projet d'étude des algorithmes dynamiques et décentralisés s'inscrit naturellement dans le cadre de l'équipe GRAAL du LIP.

Ordonnancement de machines virtuelles

Lors de mon séjour post-doctoral au laboratoire ACIS de l'université de Floride, j'ai été amené à considérer des problèmes d'ordonnancement pour les machines virtuelles. L'application à laquelle nous nous intéressons est une interface cerveau-machine : elle reçoit périodiquement des signaux provenant du cerveau d'un animal et doit produire une commande moteur envoyée à un robot. Le but ultime est de permettre à des personnes handicapées de déplacer des membres artificiels uniquement par la pensée. Le traitement des données provenant du cerveau consiste en un mélange d'avis de « processus experts » : un grand nombre de modèles indépendants (les experts) produisent chacun une réponse. Ces résultats sont agrégés en une réponse finale en tenant compte de l'importance des différents modèles, les importances étant mises à jour périodiquement. Il existe également un processus d'apprentissage qui permet à chaque modèle de raffiner ses réponses en mettant à jour ses paramètres. Durant chaque période, les modèles désignés comme importants doivent calculer leur réponse en un temps très limité, alors que autres calculs (les autres modèles et l'apprentissage) peuvent s'exécuter plus lentement.

Pour exécuter ces calculs, nous comptons utiliser l'architecture In-Vigo développée à l'université de Floride. Celle-ci permet de créer de façon dynamique un ensemble de ressources virtuelles pour une application. Les tâches correspondants aux différents « experts » peuvent ainsi être exécutées comme des processus communicants au sein d'une même machine virtuelle, ou encore comme différentes machines virtuelles. Certaines de ces tâches doivent être exécutées rapidement, le temps de traitement ne devant pas excéder quelques

centaines de millisecondes, alors que d'autres tâches ont des contraintes de temps d'exécution plus souples.

Plusieurs problèmes d'ordonnancement doivent être résolus. Il faut savoir ordonnancer précisément les différentes tâches présentes sur une machine physique afin de garantir le temps d'exécution des tâches importantes. Il faut également savoir répartir les différentes tâches entre les machines disponibles, pour en même temps minimiser le nombre de machines utilisées (et le temps de communications entre elles) et garantir le temps d'exécution de certaines tâches. Pour ceci, les modifications apportées à l'ordonnanceur en charge des machines virtuelles cohabitant sur un même hôte physique doivent être prises en compte pour modéliser le comportement d'une machine. Enfin, il faut aussi savoir redistribuer ces différentes tâches : comme le système évolue, l'importance des différents experts change, certaines tâches deviennent donc plus prioritaires, d'autres moins prioritaires, et le nombre d'experts (et donc de tâches) peut aussi varier. Les algorithmes existants de rééquilibrage de charge doivent donc être adaptés afin de prendre en compte les contraintes de temps-réel spécifiques à cette application.

Si ce travail concerne principalement une application donnée de traitement du signal pour les neurosciences, l'ordonnancement pour machines virtuelles est également un sujet d'actualité pour d'autres applications. Les machines virtuelles sont en particulier utilisées pour des applications de services Web : on peut citer l'exemple d'Amazon qui met ses serveurs à disposition d'utilisateurs extérieurs pour héberger des machines virtuelles exécutant des services Web dans son « Amazon Elastic Compute Cloud ». Là encore, il faut réfléchir aux stratégies de placement des différentes machines virtuelles sur les machines physiques, et au problème de la redistribution : la charge des différents servives hébergés évolue, et avec elle les ressources nécessaires aux différentes machines virtuelles. Un service qui voit son trafic augmenter peut être amené à se dupliquer. On peut alors imaginer concevoir une collaboration entre l'ordonnanceur du service (qui connaît la charge et peut décider de la duplication) et l'ordonnanceur des machines virtuelles qui gère l'ensemble de serveurs disponibles.

Ce travail se poursuivra naturellement en collaboration avec le laboratoire ACIS de l'université de Floride.

Ordonnancement pour les réseaux

Le troisième axe de mon projet de recherche est en collaboration avec l'équipe RESO du Laboratoire de l'Informatique du Parallélisme. Il consiste en une poursuite des travaux déjà entrepris avec cette équipe (et publiés dans les conférences GlobeCom 2005 et HPDC 2006).

Nous partons de la constatation qu'actuellement, dans l'utilisation des réseaux à grande échelle, aucune anticipation n'est faite pour ordonnancer et gérer les différents flux se partageant les ressources disponibles. Des informations sur la charge et la nature du trafic sont bien utilisées par les opérateurs pour « provisionner » le réseau, c'est-à-dire pour connaître le nombre et la localisation des équipements nécessaires lors de la construction ou de l'évolution d'un réseau, mais aucune information explicite de charge n'est utilisée lors du routage et de l'acheminement des différents flux.

Or aujourd'hui, il devient possible de prédire l'utilisation du réseau par certaines applications, en particulier pour des applications de calcul scientifique distribuées. Lors du lancement d'une telle application, on peut connaître assez précisément ses besoins en communications. Ces besoins sont d'une nature différente des autres flux utilisant le réseaux : il

s'agit généralement de gros volumes de données à transférer sans taux de transmission imposé, mais qui doivent être effectués à l'intérieur d'une fenêtre temporelle prédéfinie (afin de respecter les dépendances de données et les contraintes de synchronisation de l'application).

En collaboration avec l'équipe RESO, nous avons déjà commencé à étudier ce problème pour le cas de topologies simples, en supposant que le cœur du réseau était sur-dimensionné et que la seule contention possible avait lieu au niveau des points d'accès (entrants ou sortants). Nous souhaiterions maintenant nous intéresser à des topologies de réseaux plus générales. Un grand nombre d'applications de calcul scientifique ne sont en effet plus destinées à être exécutées sur des environnements dédiés, mais doivent s'adapter à des plates-formes ou la ressource réseau est partagée et peut constituer un goulet d'étranglement.

De nombreux travaux existent dans le domaine des réseaux qui visent à la réservation de ressources et/ou à la qualité de service. On peut par exemple citer le protocole de réservation RSVP qui, immédiatement avant l'exécution d'un transfert, réserve la bande passante souhaitée le long de la route empruntée. Cependant, aucune information sur les besoins des applications n'est signalée à l'avance, alors que cette information est disponible : nous voulons donc la prendre en compte afin d'optimiser l'utilisation des ressources. D'autres approches comme RON (*Resilient Overlay Network*) créent un réseau logique (*overlay*) par dessus le réseau physique (Internet) pour assurer que les routes reliant les sites utilisés ont les propriétés requises. L'inconvénient d'une telle approche est que pour assurer un « routage utilisateur », les transferts doivent « remonter » vers chacun des sites présents sur leur route. Or la congestion se trouve le plus souvent au niveau de la connexion des sites au réseau longue-distance, qui est une partie critique pour les performances de cette approche. On peut également citer d'autres travaux plus spécifiques comme la réservation de longueur d'ondes dans les réseaux optiques pour des plates-formes dédiées, ou encore l'utilisation de plusieurs supports de communication (optique, IP, sans-fil, etc.) en fonction du type de trafic en vue de garantir une qualité de service (QoS-routing).

En vue de réserver des ressources pour des transferts anticipés, nous pouvons donc agir sur la dimension spatiale : en réservant des ressources différentes, comme des routes avec une qualité de service garantie, pour un trafic spécial. En anticipant les requêtes, nous pouvons également agir au niveau temporel, en modifiant le taux de transfert d'une application au cours du temps. Nous voulons également concevoir des algorithmes qui offrent une certaine forme de robustesse et peuvent faire face à des variations du trafic tout en étant adaptés à des plates-formes de grande taille. Il s'agit là encore d'utiliser des algorithmes décentralisés pour résoudre ces problèmes. Le protocole de calcul des plus courts chemins OSPF est un bon exemple d'algorithme décentralisé dont nous voudrions nous inspirer pour concevoir des stratégies de réservation et d'ordonnancement de flux qui passe à l'échelle.

Liste des publications

Revues internationales avec comité de lecture

[1] L. Marchal, V. Rehn, Y. Robert et F. Vivien. «Scheduling algorithms for data redistribution and load-balancing on master-slave platforms». *Parallel Processing Letters* (2007, à paraître).

[2] 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).

[3] 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).

[4] 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.

[5] 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).

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

[7] 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

[8] V. Rehn, Y. Robert et F. Vivien. «Scheduling and data redistribution strategies on star platforms». Dans *PDP'2007, 15th Euromicro Workshop on Parallel, Distributed and Network-based Processing* (2007), IEEE Computer Society Press.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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).

[22] 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

[23] 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

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

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

[38] 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.

[39] 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.

[40] 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.

[41] 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.