INRIA

n

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Projet ReMaP

Régularité et parallélisme massif

Rhône-Alpes

THÈME 1A

Table des matières

1 Composition de l'équipe **3**

2 Présentation et objectifs généraux **4**

3 Fondements scientifiques **5**
3.1 Algorithmique, bibliothèques et compilateurs 6
3.2 Support exécutif pour le métacomputing 8

4 Domaines d'applications **10**

5 Logiciels **12**
5.1 Outils pour le calcul parallèle scientifique 12
5.2 Environnement d'exécution PM2 13
5.3 Services de communications BIP 14
5.4 Système de stockage vidéo 15
5.5 Cache Web parallèle 15

6 Résultats nouveaux **15**
6.1 Algorithmique, bibliothèques et compilateurs 15
6.1.1 Algorithmique sur réseau hétérogène de processeurs 16
6.1.2 Serveurs de calculs et optimisation d'algorithmes numériques parallèles . 17
6.1.3 Algorithmique des télécommunications 19
6.1.4 Transformations de programmes 21
6.2 Environnements d'exécution parallèles et distribués 23
6.2.1 Extensions de l'ordonnanceur Marcel/PM2 24
6.2.2 Communications hétérogènes en Madeleine/PM2 25
6.2.3 Support générique des fonctionnalités DSM en PM2 26
6.2.4 MPI au-dessus de PM2 27
6.2.5 Évolutions de l'interface de communication BIP/Myrinet 27
6.2.6 Support exécutif pour les entrées-sorties multimédia 27

7 Contrats industriels (nationaux, européens et internationaux) **28**
7.1 *LHPC* 28
7.2 Contrat INRIA/Microsoft 29
7.3 Accord-cadre INRIA avec la société Alcatel 29
7.4 Contrat Myricom 29

8 Actions régionales, nationales et internationales **30**
8.1 Actions nationales 30
8.2 Actions financées par la commission européenne 31
8.3 Relations bilatérales internationales 31

9 Diffusion de résultats **33**
9.1 Animation de la communauté scientifique 33
9.2 Enseignement universitaire 34
9.3 Autres enseignements 35
9.4 Participation à des colloques, séminaires, invitations 35

10 Bibliographie **36**

Le projet ReMaP *est un projet commun CNRS/ENS Lyon/INRIA du Laboratoire de l'Informatique du Parallélisme (LIP), UMR CNRS/ENS Lyon/INRIA 5668. Ce projet est localisé à Lyon dans les locaux de l'ENS Lyon.*

1 Composition de l'équipe

Responsable scientifique

Frédéric Desprez [CR INRIA]

Assistantes de projet

Sylvie Boyer [Adjointe INRIA, 30% sur le projet]

Anne-Pascale Botonnet [Adjointe administrative ENS Lyon, 30% sur le projet]

Personnel Inria

Frédéric Desprez [CR]

Isabelle Guérin-Lassous [CR]

Jean-Yves L'Excellent [CR, arrivée le 01/09/01]

Tanguy Risset [CR, arrivée le 01/09/01]

Gil Utard [CR (détachement), arrivée le 01/10/01]

Personnel CNRS

Alain Darte [CR]

Jean-Christophe Mignot [IR]

Loïc Prylli [CR]

Nicolas Schabanel [CR]

Personnel ENS Lyon

Olivier Beaumont [Maître de conférences, départ le 01/09/01 (MdC ENSEIRB)]

Luc Bougé [Professeur, départ le 01/09/01 (Prof. antenne ENS Cachan Ker Lann)]

Raymond Namyst [Maître de conférences]

Yves Robert [Professeur]

Ingénieurs experts

Eddy Caron [Ingénieur contractuel INRIA, arrivée le 01/04/01]

Philippe Combes [Ingénieur contractuel INRIA, arrivée le 01/09/01]

Patrick Dargenton [Ingénieur contractuel ENS Lyon, 3 mois]

Chercheur extérieur

Jean-François Méhaut [Prof. Univ. Antille-Guyanne]

Chercheurs invités

Larry Carter [Professeur UCSD, Mars-Juin 2001]

Jeanne Ferrante [Professeur UCSD, Mars-Juin 2001]

Chercheurs doctorants

Gabriel Antoniu [Allocataire MENRT]

Olivier Aumage [Allocataire MENRT]

Alice Bonhomme [Boursière Cifre *SAM*]

Vincent Boudet [Allocataire moniteur normalien]

Claude Chaudet [Allocataire MENRT, arrivée le 01/10/01]

Frédérique Chaussumier [Boursière Cifre *SAM*]

Vincent Danjean [Allocataire moniteur normalien]
Dominique Douthaut [Allocataire MENRT]
Guillaume Huard [Allocataire MENRT]
Arnaud Legrand [Allocataire moniteur normalien]
Guillaume Mercier [Allocataire MENRT]
Martin Quinson [Allocataire MENRT]
Frédéric Suter [Allocataire MENRT]

2 Présentation et objectifs généraux

Mots clés : data-parallélisme, parallélisation automatique, compilation, environnement de programmation, bibliothèques, réseaux à haut débit, application répartie, algorithmique hétérogène, métacomputing, télécommunications.

À sa création, l'objectif du projet *ReMaP* était de contribuer à l'élaboration des connaissances dans le domaine du calcul massivement parallèle régulier. La compilation data-parallèle reste au cœur du projet, mais nous nous intéressons aussi aux applications sur réseaux hétérogènes (bibliothèques, processus légers, protocoles de communication et leurs impacts) et aux télécommunications sur réseaux sans fil (protocoles).

Les axes de recherche de *ReMaP* sont les suivants :
- techniques de parallélisation automatique et outils de parallélisation ;
- outils pour l'algèbre linéaire parallèle : algorithmique parallèle en algèbre linéaire dense et creuse, mise en œuvre de bibliothèques numériques sur réseaux hétérogènes de processeurs ;
- protocoles pour les réseaux sans fil ;
- compilation data-parallèle pour processus légers migrables : HPF (High Performance Fortran), C* et Java, support d'exécution multi-threads ;
- protocoles orientés applications pour les réseaux à haut débit.

Un point fort du projet est son ancrage industriel et ses activités de transfert. Citons principalement :
- laboratoire *LHPC* commun avec Sycomore Aérospatiale Matra (*SAM*, anciennement MATRA Systèmes et Information) ;
- collaboration avec *SAM*, le centre multimédia *Érasme*, *Rhône Vision Câble* et *Tonna Informatique* dans le cadre du projet *CHARM* ;
- collaboration avec *Myricom* autour des développements de *BIP* ;
- coopération avec l'université du Tennessee, Knoxville, dans le cadre de la bibliothèque d'algèbre linéaire parallèle *ScaLAPACK* qui est mise à la disposition de la communauté internationale.

Une vision schématique mais synthétique de nos axes de recherches est proposée à la Table 1. Les mots-clés de *ReMaP* sont :

Compilation + Bibliothèques + Réseaux + Applications

Niveau	**Logiciels**	**Collaborations**	
		France	*Internat.*
Architectures	Piles de PC, serveur multimédia	*SAM*	
Protocoles	*BIP*, SCI	RHDM, VASY, ARC ResCapA	Myricom, UC Berkeley
Bibliothèques	MPI, PVM, SSCRAP *ScaLAPACK*, *Scilab*	Métalau, LORIA LIFC, LaRIA	U Tennessee, Argonne
Supports d'exécution	PM2, Marcel, Madeleine	LIFL, Apache, Sirac, Caps	Argonne, Rice
Compilateurs	HPF, C* *Nestor*	ex Paradigme/PRS, iHPerf	GMD, U New Hamp. Rice, HP Labs (CA)
Applications	Numérique dense + creux Cache Web	*LaBRI* *SAM*	UCSD

TAB. 1 – Vision synthétique de *ReMaP*

3 Fondements scientifiques

Depuis quelques années, le paysage des architectures parallèles s'est considérablement modifié et ce, aux deux extrémités du spectre : d'une part le parallélisme est descendu au niveau interne des processeurs (superscalaire, VLIW), et d'autre part on assiste à l'avènement des grappes de PC (*cluster computing*), avec l'interconnexion à plus large échelle de ressources de calcul distribuées (*metacomputing* ou *Grid computing*). Un des nouveaux défis qui se posent pour la mise en œuvre des applications parallèles est la maîtrise de l'hétérogénéité. Pour les utilisateurs et les programmeurs, le caractère hétérogène des architectures pose de nouveaux problèmes dans le développement des applications. Deux approches pour la conception des applications parallèles sont possibles :

- L'application est conçue de manière séquentielle et le programmeur doit pouvoir disposer d'outils de parallélisation automatique et d'interfaces conviviales. Cette problématique est rendue particulièrement complexe avec les nouvelles technologies de processeurs qui permettent par exemple l'exécution simultanée de plusieurs instructions. Les compilateurs de langages, même séquentiels, doivent maintenant réussir à extraire du parallélisme des programmes.
- L'application est conçue de manière parallèle et de nombreuses difficultés surgissent : difficultés algorithmiques bien sûr, mais aussi difficultés de mise au point qui dépendent de la complexité d'une plate-forme parallèle hétérogène. Les principaux problèmes concernent les communications (différentes technologies de réseaux et de protocoles), la répartition des données qui tienne compte de l'hétérogénéité (puissance de calcul, mémoire, bande passante) et répartition des calculs.

L'objectif de *ReMaP* est de (tenter de) relever ces nouveaux défis, en apportant des contributions au niveau des algorithmes, des bibliothèques, de la compilation, des environnements de programmation et des protocoles pour les réseaux à haut débit. La pyramide des problèmes auxquels nous nous intéressons a été décrite succinctement au paragraphe précédent ; ceux-ci seront abordés de manière plus précise ci-dessous.

Notre credo est triple :
- participer aux projets de collaboration internationaux (comme *ScaLAPACK*) plutôt que de développer en interne des logiciels propriétaires ;
- travailler en forte collaboration avec des partenaires industriels pour valider notre approche et nos résultats ;
- mener de front des travaux de recherche fondamentale et appliquée.

Ces objectifs, bien que parfois difficilement conciliables, guident nos travaux. Ceux-ci sont structurés en deux grands axes :

1. Algorithmique, bibliothèques et compilateurs ;
2. Environnement d'exécution multi-threads et réseaux à haut débit.

3.1 Algorithmique, bibliothèques et compilateurs

Participants : Olivier Beaumont, Vincent Boudet, Eddy Caron, Larry Carter, Claude Chaudet, Frédérique Chaussumier, Alain Darte, Frédéric Desprez, Dominique Douthaut, Jeanne Ferrante, Isabelle Guérin-Lassous, Guillaume Huard, Arnaud Legrand, Jean-Yves L'Excellent, Martin Quinson, Yves Robert, Tanguy Risset, Frédéric Suter, Nicolas Schabanel, Gil Utard.

L'objectif de ce premier thème du projet est de rendre le parallélisme *transparent* pour l'utilisateur, ou du moins de faciliter sa mise en œuvre.

Parallélisation automatique et transformation de programmes Il s'agit de développer et d'intégrer de nouvelles stratégies permettant de transformer (semi-)automatiquement des portions de code séquentiel (principalement des boucles Fortran) en codes annotés par des directives de type HPF (High Performance Fortran) ou OpenMP. Le but est d'aider le programmeur à identifier le parallélisme potentiel de son code au niveau des boucles et d'effectuer automatiquement les transformations nécessaires à sa place (ordonnancement, placement, partitionnement, etc.).

Si les technologies de processeurs évoluent rapidement, elles exigent des compilateurs de plus en plus performants pour les exploiter. Les techniques de parallélisation automatiques trouvent à présent leur application dans la compilation pour microprocesseurs pour l'optimisation des unités parallèles et pipelinées ainsi que dans la compilation de circuits spécialisés. Mais les objectifs et techniques doivent être repensés et adaptés aux besoins de ces domaines.

Nos travaux présentent deux facettes, fortement couplées :
- développements théoriques relatifs aux problèmes de transformations de code, de leur classification théorique à la mise au point d'algorithmes les résolvant et à l'étude de la faisabilité de leur intégration au sein d'un compilateur-paralléliseur ;
- développements logiciels dont le but est de mettre en œuvre et de valider nos techniques dans le cadre d'une plate-forme de parallélisation (semi-) automatique de programmes Fortran vers HPF (*TransTool* et *Nestor*) ou d'évaluation d'heuristiques de pipeline logiciel (*Pastaga*).

Bibliothèques et algorithmique parallèle hétérogène

Algorithmique numérique parallèle La parallélisation d'applications numériques parallèles performantes nécessite l'utilisation de bibliothèques fortement optimisées pour divers types d'architectures. Les optimisations concernent le choix des algorithmes, le placement des données sur les processeurs et la réduction des communications (voire leur recouvrement). Nous travaillons sur plusieurs aspects de l'algorithmique numérique parallèle et notamment autour de la bibliothèque d'algèbre linéaire *ScaLAPACK* qui vise aussi bien la classe des super-calculateurs à mémoire partagée que celle des réseaux de stations de travail à mémoire distribuée :

- utilisation des bibliothèques dans des environnements de type *Matlab* (*Scilab*) ;
- pipelines de calculs et de communications avec recouvrements ;
- modélisation de ces routines pour les machines à mémoire distribuée ;
- recherche de nouveaux algorithmes de redistribution de tableaux et amélioration des procédures existantes ;
- utilisation de parallélisme mixte (utilisant à la fois le parallélisme de tâches et le parallélisme de données) ;
- propositions algorithmiques d'extension des noyaux de calcul de *ScaLAPACK* aux grappes hétérogènes et aux collections de telles grappes.

***Scilab* parallèle** *Scilab* est un logiciel de calcul scientifique de type *Matlab* développé par le projet Métalau à l'INRIA. Une intégration de PVM dans *Scilab* permet déjà l'accès à la bibliothèque *ScaLAPACK*. Nous développons une version parallèle de *Scilab* en collaboration avec les projets Métalau et Résédas de l'INRIA et les laboratoires LaBRI, LIFC et LARIA. Ainsi l'utilisateur pourra-t-il appeler une procédure de la bibliothèque qui s'exécutera de manière relativement transparente sur les ressources parallèles auxquelles il a accès. Les problématiques de recherche dans le développement de tels outils sont l'évaluation des performances statiques et dynamiques, l'équilibrage des charges, la recherche de distributions optimales, le traitement d'architectures hétérogènes de machines et l'algorithmique parallèle sur des plates-formes hétérogènes en général.

Algorithmique des télécommunications Cette activité a démarré l'année dernière dans le cadre du recrutement d'Isabelle Guérin Lassous et de Nicolas Schabanel.

Il s'agit de développer des algorithmes pour résoudre différentes problématiques dans différents types de réseaux de télécommunications. Les problématiques abordées couvrent de nombreux aspects rencontrés sur ce type de réseaux puisqu'elles vont des problèmes de planification (placement, allocation de fréquences) jusqu'aux problèmes de qualité de service (réservation de bande passante, délai moyen, etc.) en passant par les problèmes de routage (point-à-point ou/et multipoints) et de dissémination de données. Les réseaux de télécommunications étudiés concernent les réseaux filaires hétérogènes et les réseaux sans fil (réseaux satellitaires, réseaux cellulaires, réseaux locaux et réseaux ad hoc).

Nous essayons d'apporter nos connaissances en algorithmique séquentielle et distribuée afin de proposer des protocoles efficaces pour les problèmes énoncés ci-dessus. Nous pensons qu'aborder les différents aspects des réseaux de télécommunications (planification, routage, qualité de service) en même temps nous permet d'avoir une très bonne vue d'ensemble de

l'algorithmique des télécommunications d'une part et que ces aspects sont très fortement liés d'autre part (par exemple, suivant la planification choisie, les protocoles de routage ou de mise en œuvre d'une certaine qualité de service peuvent être très différents).

Si, dans un premier temps, les résultats obtenus sont théoriques, il nous semble néanmoins très important de les valider par une approche expérimentale à l'aide de simulateurs.

3.2 Support exécutif pour le métacomputing

Participants : Gabriel Antoniu, Olivier Aumage, Alice Bonhomme, Luc Bougé, Philippe Combes, Vincent Danjean, Jean-Christophe Mignot, Raymond Namyst, Loïc Prylli.

Notre objectif : le métacomputing départemental Depuis quelques années, la notion de processus léger (*thread*) est apparue comme un outil très utile pour le calcul haute performance comme l'ont montré de nombreux projets de recherche de par le monde, notamment *Nexus* (Ian Foster, ANL) et *Chant* (Matthew Haines, ICASE). En France, on peut citer le système Athapascan 0 (projet Apache, Grenoble), contemporain de PM2.

D'autre part, l'apparition de réseaux à haut débit dédiés à la construction de plates-formes parallèles (les grappes de stations) a eu un impact déterminant sur le monde du parallélisme durant la dernière décennie. En effet, même si les concepts de base restent identiques aux générations précédentes de machines parallèles, ces réseaux ont rendu indépendants les choix des trois aspects principaux d'une plate-forme parallèle : les nœuds de calculs, le réseau de communication et le logiciel de base (système d'exploitation et services de communications). Parmi les travaux dans ce domaine de recherche, on peut citer Score/PM (RWCP, Japon), LFC (Université libre d'Amsterdam), NOW (Berkeley), Trapeze (Duke University), GAMMA (Université de Gênes, Italie).

Notre objectif de recherche est de conjuguer ces deux développements pour permettre l'utilisation de grappes de PC standards pour le calcul haute performance. Il s'agit d'une forme de *métacomputing* spécialisée pour le niveau *départemental*, par exemple à l'échelle d'un bâtiment ou d'un petit groupe de bâtiments physiquement proches les uns des autres. Cette proximité géographique permet de réduire considérablement les problèmes d'administration et de sécurité liés au métacomputing, sans toutefois les annuler totalement comme ce serait le cas pour une grappe privée. D'autre part, cette proximité permet de considérer une interconnexion à très haut débit, typiquement de l'ordre du gigabit par seconde, entre les noeuds. Dans ce cadre, les temps de calcul locaux et les temps de communication sont du même ordre de grandeur, et les interfaces entre les différents modules ont un impact direct sur les performances globales (copies superflues, en particulier). Le modèle d'architecture de référence est donc un réseau hiérarchique de grappes de PC (une *grappe de grappes*), chaque grappe étant interconnectée par un réseau rapide (1 Gb/s). L'hétérogénéité des grappes est une donnée fondamentale du problème : hétérogénéité des processeurs (Intel, Alpha, etc., avec des cadencements variables), hétérogénéité des réseaux (Myrinet, SCI, Giganet, etc. pour les grappes, Fast-Ethernet pour l'interconnexion globale).

Notre approche se fonde sur deux développements logiciels étroitement coordonnés : PM2 et BIP/Myrinet.

L'environnement d'exécution multithread PM2 L'environnement multithread distribué PM2 (*Parallel Multithreaded Machine*) a été initialement développé au sein du projet Espace du LIFL, avant d'être transféré dans le projet ReMaP. PM2 est en fait constitué d'une *suite* de modules. Le module de gestion de thread, appelé *Marcel*, fournit une implémentation très efficace des primitives POSIX en espace utilisateur. Le module de gestion des communications, appelé *Madeleine*, fournit une interface portable particulièrement optimisée pour les appels de procédures à distance (*Remote Procedure Call*, RPC). Le module de mémoire virtuellement partagée, appelé *DSM-PM2* pour *Distributed Shared Memory*, permet aux threads situés sur des nœuds différents de partager dynamiquement des pages de la mémoire virtuelle. Grâce à l'intégration de l'ensemble de ces modules, PM2 offre une fonctionnalité de *migration préemptive transparente* des threads entre nœuds distants. Cette fonctionnalité ouvre la possibilité d'installer au-dessus de PM2 un ordonnanceur global qui équilibre la charge dynamiquement en déplaçant les threads à travers l'ensemble du système.

L'ensemble de la conception de PM2 a été élaborée avec un souci de portabilité. La *suite* PM2 est actuellement disponible au-dessus de la plupart des systèmes : GNU/Linux, Solaris, ainsi que des variantes propriétaires (IBM et HP notamment). Le portage sur WindowsNT est en cours. L'interface de communication est disponible sur la plupart des interfaces utilisées pour l'interconnexion de grappes de calculateurs : TCP/IP bien sûr, mais aussi MPI, BIP pour le réseau gigabit Myrinet, SISCI pour les réseau SCI, etc. PM2 a en particulier été porté sur l'interface de communication VIA (*Virtual Interface Architecture*) actuellement en plein développement. Des évaluations de performance de VIA sur le réseau Giganet sont en cours.

L'interface réseau BIP/Myrinet Le *suite* logicielle BIP (*Basic Interface for Parallelism*) propose une interface optimisée pour le réseau *Myrinet* proposé par la société américaine *Myricom*. Cette interface est particulièrement adaptée au support d'environnement exécutifs multithreads tels que PM2 et/ou au support d'implémentation pour des bibliothèques de communication telles que MPI. La conception favorise les transmissions zéro-copies en espace utilisateur ainsi que l'utilisation intensive des capacités de calcul et de gestion mémoire offertes par la carte Myrinet. Ceci permet de soulager d'autant le processeur principal et donc d'augmenter considérablement la réactivité et les performances. Ces travaux ont déjà permis d'obtenir d'excellents résultats pour l'utilisation efficace des réseaux Myrinet et plusieurs dizaines d'équipes dans le monde utilisent la suite logicielle BIP comme base pour leurs propres développements (BIP, MPI-BIP) ou comme support réseau performant pour leurs applications (MPI-BIP, IP-BIP).

Nos défis de recherche

Communication multi-protocoles pour étendre les fonctionnalités de PM2, et tout particulièrement de son module de communication *Madeleine*, aux *grappes de grappes* constituées de sous-réseaux différents. L'objectif est d'obtenir ainsi des versions *portables* et *multi-protocoles* de couches de communication plus complexes, comme par exemple MPI.

Support des nœuds multiprocesseurs en espace utilisateur. Il s'agit ici de permettre à des bibliothèques de multithreading de niveau utilisateur comme *Marcel* de coopérer avec l'ordonnanceur noyau pour tirer parti des nœuds multiprocesseurs (*SMP nodes*)

et gérer correctement les appels bloquants. Le modèle proposé est une extension des *Scheduler Activations*.

Mémoire virtuellement partagée en environnement multithread. L'introduction d'une fonctionnalité de mémoire virtuellement partagée dans PM2 impose de repenser complètement les protocoles classiques, conçus dans un environnement monothread, pour les adapter au cas multithread : il faut alors prendre en compte des actions de cohérence *concurrentes*.

Entrées-sorties distribuées, en vue de l'utilisation de PM2 pour l'implémentation de serveurs multimédia sur des grappes de PC standards. Nos expériences (caches Web) ont montré l'importance déterminante de mécanismes d'entrée-sortie *asynchrones* étroitement intégrés avec l'ordonnanceur de threads.

Support exécutif haute performance. Il s'agit ici d'utiliser PM2 comme cible pour des compilateurs. Il devient alors possible de faire coopérer des analyses statiques avec des ajustements dynamiques. Une première expérience a été conduite avec un compilateur HPF, avec des résultats concluants, et nous travaillons actuellement sur un compilateur Java. D'autre part, PM2 peut aussi offrir un support exécutif très intéressant pour des environnements complexes comme des ORB Corba ou des environnements de gestion comme Globus. Ces travaux nécessitent une instrumentation précise des routines de base et des outils d'analyse sophistiqués pour permettre la localisation fine des sources d'inefficacité et la mise en place d'optimisations.

Optimisation des couches réseaux. L'efficacité de PM2 est entièrement dépendante de l'efficacité des couches réseaux sur lesquelles il s'appuie. L'optimisation fine de la bibliothèque de communication MPI au-dessus de l'interface BIP/Myrinet reste l'une de nos priorités à cause de son impact : plusieurs dizaines d'équipes l'utilisent à ce jour dans le monde. Parmi les travaux récents, citons le support optimisé des nœuds SMP et la mise en place de dispositifs de communications asynchrones au-dessus de BIP pour faciliter les recouvrements entre calculs et communications.

4 Domaines d'applications

Contexte Le parallélisme évolue et les domaines d'application se multiplient. Avec l'apparition de réseaux de stations de travail ou de piles de PC au bon rapport performance/prix, de nombreux utilisateurs se tournent vers des solutions parallèles. Outre les domaines d'application (on dépasse le traditionnel calcul scientifique pour aborder le secteur médical, bancaire, la sidérurgie, le textile, la publicité, la géographie, etc.), la nature des partenaires évolue (aux centres de recherche et développement des grands groupes publics ou privés s'ajoutent désormais les petites et moyennes entreprises). Ces nouveaux utilisateurs ont un besoin crucial d'environnements de programmation performants.

Calcul scientifique Nos objectifs se déclinent selon trois axes principaux.

- Concevoir et développer un environnement de programmation standard et portable (incluant des bibliothèques de calcul et de macro-communications, des outils de placement et

d'ordonnancement, des compilateurs-parallèliseurs, des environnements de type *Matlab*, etc.) pour machines parallèles à mémoire distribuée et réseaux de stations de travail.
- Mettre à disposition les plates-formes d'expérimentation développées en collaboration avec *SAM* pour les chercheurs et les industriels ; offrir des services d'ingénierie pour le portage d'applications existantes ou le développement de nouvelles applications et pour les activités de recherche associées.
- Assurer la formation des étudiants, chercheurs et ingénieurs et ainsi faciliter l'introduction du calcul parallèle au niveau des services de recherche et développement des entreprises.

Applications aux télécommunications Nos objectifs sont d'appliquer nos connaissances en algorithmique parallèle et distribuée à l'algorithmique des télécommunications afin de proposer des protocoles de réseaux efficaces. Cette approche est menée selon deux axes.
- Fournir des algorithmes parallèles efficaces fonctionnant sur réseaux de stations de travail ou de piles de PC répondant à des problèmes de télécommunication (comme le placement d'équipements ou l'allocation de fréquences par exemple). Notre but est d'apporter des algorithmes plus efficaces (en terme de temps ou d'optimisation) que ceux existants jusqu'ici.
- Proposer des algorithmes distribués qui mettent en œuvre des protocoles réseaux. Ces algorithmes sont directement implantés sur les réseaux de télécommunications en question. Ils doivent donc prendre en compte les aspects architecturaux et protocolaires des réseaux. Dans le cas des réseaux mobiles, les algorithmes proposés doivent s'adapter à la topologie dynamique des réseaux.

Pour ce domaine d'applications, un premier projet avec des partenaires industriels a démarré en octobre 2000 (ProxiTV).

Applications multimédia Nous développons des logiciels permettant de transformer des grappes d'ordinateurs standards en serveurs de cache Internet haute performance pour les têtes de réseau des boucles haut débit. Ce système, réalisé conjointement par *SAM* et le LIP, utilise des technologies issues du parallélisme pour garantir son extensibilité. Le système de cache intègre des fonctionnalités d'indexation en ligne et de filtrage. Un autre aspect novateur du système réside dans le support des flux audio et vidéo par des mécanismes de cache ou de miroir pour tenir compte de la part grandissante des documents multimédia dans le trafic Internet. Le projet inclut des phases d'expérimentation sur le réseau haut débit *Autoroutes Rhodaniennes de l'Information* de Rhône Vision Câble. Dans le cadre de ces expérimentations, *SAM* et le LIP mettront un serveur de cache extrêmement performant à disposition du centre serveur Érasme. Cette plate-forme permettra de vérifier l'adéquation des solutions proposées en termes de fonctionnalités et de performances et d'évaluer les bénéfices des infrastructures à haut débit.

Un autre type d'application est étudié dans le cadre du projet SPIHD (Services et Programmes Interactifs pour l'Internet Haut Débit). Plus précisément, il s'agit de définir un ensemble de services de télévision interactive pour l'Internet haut débit. Quatre composantes ont été identifiées pour mener à bien ce projet.

1. La production du contenu multimédia par les partenaires de l'audio-visuel (France3-Lorraine, la cinquième, CanalWeb).
2. Le développement de l'infrastructure logicielle (numérisation, serveur de tête de réseau, gestion et mise à jour des bases de données).
3. L'expérimentation en grandeur réelle sur une boucle locale à Nancy avec la société SEM-Câble de l'Est.
4. L'évaluation technique et économique de la solution pour un éventuel déploiement à plus large échelle.

La contribution du projet *ReMaP* se situe au niveau du point 2 où nous nous intéressons aux possibilités d'indexation de la base de données des vidéos.

5 Logiciels

5.1 Outils pour le calcul parallèle scientifique

Mots clés : parallélisation de code, calcul numérique, parallélisation automatique, serveurs de calculs, bibliothèques numériques.

Participants : Eddy Caron, Alain Darte, Frédéric Desprez, Isabelle Guérin-Lassous, Guillaume Huard, Martin Quinson, Frédéric Suter.

***Scilab* parallèle** *Scilab* est un logiciel de calcul scientifique de type Matlab développé par le projet Métalau à l'INRIA. Nous avons parallélisé ce logiciel dans le cadre de l'ARC INRIA OURAGAN.

Nos développements concernent la mise en place de serveurs de calcul efficaces accessibles depuis l'interface Scilab et l'interfaçage directement dans Scilab de bibliothèques numériques telles que ScaLAPACK ou PETSc. Autour des serveurs de calcul, nous avons développé les prototypes de deux outils utilisés pour la localisation de ressources logicielles et matérielles adaptées à la résolution de problèmes envoyés depuis Scilab. Le premier outil est une base de données de ressources logicielles (SLiM, *Scientific Library Metaserver*). Le second outil permet l'évaluation des performances des calculs sur les serveurs et des communications entre le client et les serveurs, et des serveurs entre eux (FAST, *Fast Agent System Timer*). Pour plus d'informations voir le site Web : `http://www.ens-lyon.fr/~desprez/OURAGAN/`.

Pastaga *Pastaga* (Plate-forme d'Analyse Statistique et Théorique d'Algorithme sur Graphes Aléatoires), développé par Guillaume Huard, est un petit logiciel d'évaluation d'heuristiques de pipeline logiciel qui nous a permis d'effectuer des tests de performance (virtuels, c'est-à-dire simulés) de nos techniques. Cet outil permet de spécifier des machines simples de type VLIW, de manipuler des graphes de dépendance réalistes (dont les caractéristiques sont celles de codes réels), de les générer aléatoirement mais de façon contrôlée (à l'aide de paramètres), de les visualiser (à l'aide de l'outil vcg) et de leur appliquer des algorithmes classiques de graphes. L'outil inclut des algorithmes de décalage d'instructions pour le pipeline logiciel et le

réordonnancement par étages. Il permet également de calculer les bornes classiques de performance pour le pipeline logiciel et de réunir des statistiques sur l'efficacité et le comportement des algorithmes étudiés. *Pastaga* est disponible sur simple demande et est diffusé sous licence GPL.

SSCRAP SSCRAP est une bibliothèque C++ qui met en œuvre les routines de base nécessaires pour les algorithmes écrits dans le modèle CGM (Coarse Grained Multicomputer) et fonctionnant sur plusieurs types d'architectures. Cette bibliothèque est le fruit d'une collaboration avec Jens Gustedt (projet Résédas, INRIA Lorraine) et Mohamed Essaïdi (projet Résédas, INRIA Lorraine). La version actuelle est basée sur MPI. Les algorithmes implantés et disponibles sont le list ranking, la somme parallèle préfixe, le tri et la contraction d'arbres. Ces implantations sont disponibles sur `http://www.loria.fr/~gustedt/cgm/`.

5.2 Environnement d'exécution PM2

Mots clés : multithreading distribué, migration, RPC, réseaux à haut débit, mémoire partagée distribuée.

Participants : Gabriel Antoniu, Olivier Aumage, Luc Bougé, Jean-François Méhaut, Raymond Namyst.

PM2 est un environnement multithread portable permettant d'exploiter efficacement les architectures distribuées haute performance (super-calculateurs, grappes de stations SMP interconnectées par réseau haut débit). Il se distingue par l'efficacité de sa gestion des processus légers, par ses fonctionnalités d'équilibrage dynamique de charge (migration de processus légers) ainsi que par son interfaçage efficace avec les protocoles de communication de très bas niveau. L'objectif de l'environnement PM2 est de définir un support d'exécution haute performance pour les applications parallèles ou les compilateurs de langages parallèles (HPF, Java) sur architectures distribuées (machines parallèles, grappes et bientôt grilles). La caractéristique majeure de ces applications est qu'il est difficile, voire impossible, de répartir statiquement les traitements et les données sur les processeurs de manière équilibrée.

Le modèle de programmation PM2 s'articule autour d'une décomposition des calculs en procédures activables par un mécanisme de type RPC (appel de procédure à distance). Ce découpage, potentiellement extrêmement fin, est pris en charge efficacement par le support exécutif de PM2. Afin de corriger les situations de déséquilibre, PM2 fournit un opérateur de migration permettant de déplacer les activités dynamiquement d'un processeur vers un autre. Cet opérateur s'appuie sur un mécanisme d'allocation iso-adresse assurant qu'un processus léger (et les données qu'il manipule) reste toujours logé dans la même zone d'adresses virtuelles des processus. En complément, PM2 inclut un noyau générique permettant la gestion d'une partie de l'espace mémoire en mode *virtuellement partagé* (DSM). Ce mécanisme permet aux processus légers d'une application PM2 de partager directement des données en mémoire sur des grappes homogènes.

Le support d'exécution s'appuie sur deux bibliothèques (Marcel et Madeleine) qui ont été développées pour PM2. Marcel est une bibliothèque de processus légers qui sont créés en contexte utilisateur (temps de commutation $< 1\,\mu$s) et dont l'exécution est prise en charge

par des processus noyaux permettant ainsi d'exploiter le parallélisme d'architectures de type SMP. Madeleine est une interface de communication qui se veut à la fois portable et efficace sur différents protocoles réseaux (temps de migration d'un processus léger : 26 μs sur le réseau SCI).

Récemment, une version adaptative multi-protocoles de Madeleine a été conçue permettant l'utilisation de plusieurs protocoles réseaux au sein d'une même application. En outre, lorsqu'un protocole sous-jacent offre plusieurs modes de transfert des données, Madeleine sélectionne automatiquement le plus approprié. Cette interface est actuellement opérationnelle sur les protocoles BIP (Myrinet), SISCI (SCI), MPI et TCP.

Les sources complets de la suite PM2 sont disponibles à l'URL `http://www.pm2.org`. L'environnement PM2 est en outre déposé à l'Agence de Protection des Programmes. Plusieurs équipes de recherches, en France et à l'étranger, utilisent PM2 ou certains de ses sous-composants : Rennes (Thierry Priol), Bordeaux (Jean Roman), UNH-Durham/USA (Philip Hatcher), Berlin/Allemagne (Frank Mueller).

5.3 Services de communications BIP

Mots clés : passage de messages, communication, réseaux à haut débit, Myrinet.

Participants : Loïc Prylli, Alice Bonhomme.

Vue d'ensemble BIP est un système logiciel fournissant plusieurs types de services de communications pour le réseau Myrinet.

La couche de plus bas niveau (*firmware* et bibliothèque hôte) s'interface directement avec le matériel. L'utilisation principale du système BIP se fait par l'intermédiaire de MPI-BIP, une implémentation complète du standard MPI-1 basée sur MPICH, permettant à la majeure partie des applications distribuées de tourner sur notre système sans effort de portage.

Plus de 200 sites distincts ont acquis un mot de passe pour rapatrier le système BIP. Les retours montrent que plusieurs dizaines d'équipes l'utilisent soit pour leurs propres développements de *middleware* (BIP, MPI-BIP), soit comme support réseau performant pour leurs applications (MPI-BIP, IP-BIP).

Les logiciels BIP sont actuellement disponibles pour l'ensemble des cartes Myrinet et pour les architectures x86/Alpha/PowerPC sous Linux (`http://www.ens-lyon.fr/LIP/RESAM/index_bip.html`).

Nouvelles fonctionnalités L'un des points d'étude actuels de BIP porte sur les techniques de notification des événements à l'application qui permettent de minimiser les interruptions. On peut se reposer sur la "scrutation active" soit à l'intérieur de l'application, soit de manière centralisée dans le noyau pour éviter à chaque processus de scruter individuellement. Mais dans certains cas il est plus intéressant voire nécessaire de payer le coût d'une interruption matérielle (éventuellement de manière retardée). Combiner et alterner entre ces techniques demande la création d'heuristiques permettant de choisir la stratégie à suivre à un instant donné.

5.4 Système de stockage vidéo

Mots clés : système de fichiers distribués, tolérance aux pannes, serveurs vidéos.

Participants : Loïc Prylli, Alice Bonhomme.

Dans le cadre d'une bourse CIFRE avec la société Sycomore-Aérospatiale-Matra (SAM), Alice Bonhomme a réalisé un système de stockage vidéo basé sur les principes suivants :

- chaque flux est réparti sur un ensemble de nœuds avec des blocs de parité pour assurer la redondance nécessaire à la tolérance aux pannes,
- en cas de panne, il n'y a pas de perturbation des clients, dans la mesure ou les blocs de parité sont préventivement lus, permettant l'alimentation du client, avant qu'une panne soit confirmée,
- l'architecture du système est conçu pour être facilement extensible, il n'y a pas de nœud maître.

Ce système de stockage distribué est intégré dans des produits de la société SAM destinés à la commercialisation.

5.5 Cache Web parallèle

Mots clés : cache web distribué, grappes, serveur internet.

Participants : Luc Bougé, Jean-François Méhaut, Jean-Christophe Mignot, Loïc Prylli.

Dans le cadre du projet *CHARM*, *ReMaP* a réalisé un cache web parallèle extensible (*scalable*) basé sur une grappe de PC. Le but était d'être extensible en terme de quantité de données cachées et de nombre de clients supportés. Le projet s'est terminé en novembre 2000 par une revue générale.

Le composant principal de ce cache est un routeur HTTP, permettant de répartir les requêtes HTTP entre plusieurs unités de cache séquentiels suivant une fonction de partitionnement des URL. L'utilisation des techniques de *VirtualServer* de Linux permet un première répartition *round-robin* des connections TCP vers l'ensemble des *routeurs HTTP* (aussi appelés *proxy-light*) qui redistribuent ensuite les requêtes après analyse du contenu de l'en-tête HTTP. Pour des raisons de performances, les communications entre le routeur HTTP et les unités de caches réparties sur les nœuds sont transmises via des messages MPI plutôt que sur une connexion TCP.

6 Résultats nouveaux

6.1 Algorithmique, bibliothèques et compilateurs

Participants : Olivier Beaumont, Vincent Boudet, Eddy Caron, Larry Carter, Philippe Combes, Frédérique Chaussumier, Alain Darte, Frédéric Desprez, Dominique Douthaut, Jeanne Ferrante, Isabelle Guérin-Lassous, Guillaume Huard, Arnaud Legrand, Jean-Yves L'Excellent, Martin Quinson, Fabrice Rastello, Yves Robert, Tanguy Risset, Frédéric Suter,

Nicolas Schabanel, Gil Utard.

Mots clés : Algorithmique hétérogène, algorithmique des télécommunications, environnement de programmation, bibliothèque, parallélisation, compilation. automatique.

Résumé : *Cette partie résume nos travaux récents concernant l'algorithmique, les bibliothèques et les compilateurs. Ces travaux sont à la fois théoriques et pratiques. D'une part, nous développons de nouvelles méthodes d'ordonnancement et d'allocation en algèbre linéaire hétérogène et en télécommunications, de nouvelles techniques de distribution des données pour des bibliothèques de calcul distribué et de nouvelles stratégies de transformation de codes (parallélisation automatique, pipeline logiciel, synthèse de circuits). D'autre part, nous développons également des outils de programmation parallèle, à travers les prototypes logiciels* Scilab, Pastaga *et* SSCRAP.

6.1.1 Algorithmique sur réseau hétérogène de processeurs

Pour distribuer les tableaux d'une application à paralléliser sur un réseau hétérogène de stations de travail, la solution bloc-cyclique traditionnelle n'est pas adaptée. Pour prendre en compte les différentes vitesses des processeurs, on peut penser à une distribution dynamique des données. Mais les stratégies dynamiques peuvent conduire à de mauvais résultats pour deux raisons : (i) le coût des communications liées à d'éventuelles redistributions rendues nécessaires par l'allocation des tâches ; (ii) l'inactivité forcée des processeurs à cause des contraintes de dépendances. Soulignons que la deuxième contrainte (ii) devient critique pour des plates-formes de calcul hétérogène. Comme nous l'avons montré, les stratégies dynamiques gloutonnes habituellement employées sont vouées à l'échec dans ce contexte, pour peu que les calculs des processeurs ne soient pas indépendants. Très vite, l'exécution ne progresse qu'au rythme du processeur le plus lent.

Nous avons proposé une approche statique par phases de calcul, en modifiant l'allocation après chacune des phases pour tenir compte d'éventuels changements de vitesse des processeurs. Au sein de chaque phase, nous optimisons la charge des processeurs en fonction de leurs vitesses (ou des estimations courantes de celles-ci) et nous minimisons les communications.

Ces travaux initiés en 1999 et 2000 ont été poursuivis en 2001. Les principaux résultats de cette année sont résumés ci-dessous.

Algèbre linéaire et complexité En suite directe aux travaux des années précédentes, nous nous sommes intéressés à la mise en œuvre de noyaux d'algèbre linéaire dense, comme le produit de matrices, sur les grilles bidimensionnelles de processeurs identiques reliés par un réseau de communications non homogène (le temps nécessaire au transfert d'un bloc de matrices entre deux processeurs dépend de ces processeurs). Nous montrons que déterminer une organisation des processeurs en grille minimisant les communications est un problème NP-complet.

Par ailleurs, nous avons amorcé l'étude des redistributions de données pour les noyaux d'algèbre linéaire adaptés aux plates-formes hétérogènes. La vitesse des différents processeurs pouvant varier au cours du temps sur ce type de plates-formes, il est important de mettre en

œuvre des stratégies de redistributions efficaces afin de maintenir un bon équilibrage de charge tout au long du calcul. La stratégie hybride (ni complètement statique ni complètement dynamique) que nous proposons consiste à redistribuer les données après des phases d'équilibrage statique bien délimitées. Nous présentons également un algorithme optimal (sous certaines hypothèses) pour la redistribution des données lors du calcul d'un produit de matrices.

Paradigme maître-esclave Nous nous sommes intéressés au paradigme maître/esclaves pour des plates-formes hétérogènes. Nous supposons que les communications ont lieu de façon exclusive (à travers un bus, par exemple). Nous donnons un algorithme polynomial qui donne une solution optimale au problème de l'allocation des tâches lorsqu'une seule communication est nécessaire avant le traitement des tâches sur les différents processeurs. Lorsqu'une communication avant et une communication après le traitement des tâches sont nécessaires, nous montrons que le problème est aussi difficile qu'un autre problème dont la complexité est ouverte. Dans ce dernier cas, nous présentons un algorithme d'approximation polynomial garanti. Enfin, nous présentons des algorithmes asymptotiquement optimaux quand des communications sont nécessaires avant ou avant et après le traitement de chaque tâche.

Allocation de tâches indépendantes sur des configurations hétérogènes En collaboration avec Larry Carter et Jeanne Ferrante, tous deux professeurs à l'Université de San Diego et en visite au LIP de mars à juin 2001, nous avons étudié le problème de l'allocation d'un grand nombre de tâches indépendantes et de taille identique sur des plates-formes de calcul hétérogènes (les vitesses des processeurs et des liens de communications sont arbitraires). Nous avons défini un modèle de base, avec une architecture en arbre et un recouvrement calcul/communication, et en utilisant un protocole 1-port (envoi et réception simultanés, mais sur un seul lien à la fois). Pour ce modèle, nous avons déterminé le débit maximal d'un nœud en régime stationnaire et nous en avons déduit la meilleure stratégie d'allocation qui s'avère *bandwidth-centric* : si les bandes passantes entre le père et ses fils sont suffisamment élevées, alors tous les nœuds peuvent être utilisés à plein régime ; dans le cas contraire, les tâches doivent être allouées en priorité aux fils dont le temps de communication avec le père est le plus petit, sans tenir compte de leur puissance de calcul.

Nous avons également montré comment déduire de ces résultats la meilleure stratégie d'allocation pour d'autres modèles architecturaux, en faisant varier les capacités de recouvrement et/ou en faisant appel à un modèle multi-port.

Ces résultats ouvrent la voie à des travaux plus ambitieux, en liaison avec le paradigme maître/esclaves, pour l'étude des stratégies de distribution pipelinées sur les plates-formes hétérogènes.

6.1.2 Serveurs de calculs et optimisation d'algorithmes numériques parallèles

Serveurs de calculs Le but de l'ARC OURAGAN consistait à développer une version parallèle du logiciel *Scilab*. Notre but est de conserver l'interactivité de cet outil tout en améliorant les performances et en cachant le plus possible l'emploi du parallélisme à l'utilisateur. Il s'agit de permettre au programmeur d'avoir différents niveaux d'utilisation du parallélisme : soit le parallélisme est totalement caché grâce à une surcharge des opérateurs d'algèbre linéaire, soit

le programmeur a un contrôle sur la distribution des données et des calculs. Deux approches ont été retenues : soit nous dupliquons des processus Scilab sur les divers processeurs mis à notre disposition, soit nous utilisons des serveurs de calculs disposés sur les grappes accessibles via le net.

Dans le premier modèle, après avoir intégré la bibliothèque de passage de message PVM, nous avons travaillé sur l'interfaçage de bibliothèques parallèles comme ScaLAPACK (version in-core et out-of-core). Il s'agit d'ajouter à Scilab des types de données distribuées et de surcharger les opérateurs arithmétiques standards par des opérations matricielles utilisant des bibliothèques parallèles. Cette approche a plusieurs avantages puisque qu'elle permet un développement assez rapide grâce à l'intégration des bibliothèques directement dans l'outil mais, par contre, elle nécessite des modifications des types si nous voulons ajouter d'autres bibliothèques.

Dans le second modèle, nous utilisons une approche client-agent-serveur. Un client souhaitant effectuer un calcul demande à un agent quel est le serveur le plus adapté pour effectuer ce calcul (à la fois en terme de capacité logicielle et de puissance de calcul ou de capacité mémoire). Après avoir consulté les serveurs dont il a la responsabilité, l'agent décide (grâce à une heuristique d'ordonnancement) quel est le serveur qui doit effectuer le calcul et le signale au client. Le client envoie alors ses données au serveur qui effectue le calcul et retourne le résultat. Nous sommes partis du logiciel Netsolve développé à l'université du Tennessee. Netsolve est un logiciel qui fonctionne déjà selon ce modèle mais qui demande à être amélioré. Dans un premier temps, nous avons ajouté à Netsolve la persistence de données. Plutôt que de ramener les résultats à chaque fois sur le client, nous laissons les données sur place et nous les réutilisons dans les calculs futurs. Cela permet d'améliorer les performances en réduisant la latence. Ensuite nous avons travaillé sur deux outils annexes : SLiM et FAST. SLiM (*Scientific Libraries Metaserver*) est une base de données permettant à l'agent de trouver sur les serveurs la bibliothèque qui permettra de résoudre un problème. FAST (*Fast Agent System Timer*) est un outil qui permet de dire à l'agent quel est le serveur le plus adapté en terme de performances et de capacité mémoire. Utilisant le logiciel NWS (*Network Weather Service*), FAST peut calculer le coût de migration des données entre le client et les serveurs ou entre les clients entre eux. Il utilise également des benchmarks exécutés au démarrage du serveur qui permettent d'avoir un modèle des coûts de calcul pour les diverses routines disponibles. Ces modèles sont conservés dans une base de données qui envoie les résultats à l'application cliente qui lui demande. Un cache a été mis en place qui permet d'éviter la latence d'appel à la base de données.

Cette architecture logicielle (*Scilab*, Netsolve optimisé et FAST) a été testée et validée sur le réseau VTHD qui relie les grappes présentes dans les URs INRIA par un réseau à 2.5 Gb/s.

Nous travaillons maintenant sur une couche logicielle qui permettra de mettre en place un environnement de type ASP (Application Service Provider). DIET (Distributed Interactive Enginnering Toolbox) permettra, grâce à CORBA, de mettre en place plus simplement un logiciel utilisant des serveurs d'applications. Plusieurs applications sont actuellement à l'étude, notamment autour des modèles numériques de terrain (laboratoire LST de l'ENS Lyon), la simulation de circuits électroniques (laboratoire IRCOM) et d'autres applications de chimie (Chimie Nancy) et de physique (Physique ENS et Lyon 1, LAN). Ces travaux sont réalisés à des degrés divers dans le cadre du projet RNRT VTHD++, du projet RNTL GASP (labellisé en 2001) et de l'ACI Grid Grid-ASP.

Ces travaux sont effectués en collaboration avec les projets Métalau (INRIA Rocquencourt), Résédas (INRIA Lorraine) et les laboratoires LaBRI (Bordeaux), LaRIA (Amiens) et LIFC (Besançon).

Optimisation d'algorithmes numériques parallèles Nous avons poursuivi nos travaux sur l'utilisation des recouvrements calculs/communications dans les algorithmes parallèles. Nos travaux précédents concernaient des algorithmes réguliers utilisant des schémas pipelines relativement simples (pipelines mono-dimensionnels). Ensuite, nous nous sommes intéressés à des schémas pipelines simples (toujours mono-dimensionnels) sur des structures de données irrégulières (algorithme du Shear-Warp) et enfin à des schémas pipelines multi-dimensionnels sur des structures de données régulières. La difficulté vient du fait qu'un recouvrement existe déjà entre les calculs de par la distribution multi-dimensionnelle. Le gain n'est plus aussi important qu'avec les pipelines mono-dimensionnels. Nous avons commencé par évaluer le gain issu de tels algorithmes. Dans ce but, nous avons étudié une application «benchmark» du programme américain ASCI, le Sweep3D. Cette application effectue des calculs par vagues sur des matrices à trois dimensions. Il existe de nombreuses manières de pipeliner les calculs et les communications et de distribuer les données. On peut utiliser une distribution mono, bi ou tridimensionnelle et plusieurs dimensions peuvent être pipelinées.

Les algorithmes parallèles peuvent être séparés en deux catégories : les algorithmes utilisant un paradigme de programmation *data-parallèle* et ceux utilisant le *parallélisme de tâches*. Dans le premier modèle, les données utiles au calcul sont découpées selon la grille virtuelle de processeurs et ceux-ci effectuent les mêmes opérations sur des données différentes ; il s'agit donc d'une approche SPMD. Dans le second modèle, le programme parallèle est découpé en tâches et ces tâches sont réparties sur les processeurs selon les dépendances qui les lient. Nous travaillons maintenant sur l'algorithmique parallèle mixte. Dans ce modèle, nos programmes sont constitués de tâches qui sont elles-mêmes data-parallèles. Notre premier algorithme cible est le produit de matrices utilisant les méthodes de Strassen ou Winograd. Les parallélisations précédentes utilisaient généralement soit le data-parallélisme, soit le parallélisme de tâches. Nous avons prouvé que nous pouvions avoir un gain en performances grâce au parallélisme mixte en optimisant le placement des données et les redistributions entre les sous-grilles de processeurs.

6.1.3 Algorithmique des télécommunications

Cette activité a démarré l'année dernière dans *ReMaP* suite aux recrutements d'Isabelle Guérin Lassous et de Nicolas Schabanel.

Qualité de service dans les réseaux ad hoc Un réseau ad hoc est un réseau sans fil dans lequel il n'y a aucune infrastructure fixe, les ordinateurs mobiles pouvant être des routeurs potentiels. Ces réseaux ont fait l'objet d'un groupe de travail à l'IETF appelé Manet. Les travaux sur ces réseaux portent principalement sur le routage.

En ce qui concerne la qualité de service, très peu de protocoles ont été proposés. De plus, tous ces protocoles se basent sur une connaissance locale maintenue par chaque mobile (nombre de paquets présents dans le mobile, qualité des liens avec ses voisins directs, etc.), mais ne

tiennent absolument pas compte des problèmes d'interférences qui interviennent très souvent dans ces réseaux. En effet, un mobile peut très bien gêner la communication d'un autre mobile qui ne se trouve pourtant pas dans sa zone d'émission. Ce phénomène a pour effet de diminuer le débit d'émission des mobiles communicants (même si ceux-ci n'ont aucun voisin direct qui communique), voire, sous certaines configurations, d'empêcher complètement certains mobiles de communiquer.

Nous nous sommes d'abord concentrés sur le problème de réservation de bande passante qui est un problème fondamental dans les réseaux ad hoc. En effet, avant de pouvoir assurer d'autres paramètres de qualité de service aux mobiles, il est important que les mobiles disposent de bande passante pour pouvoir communiquer. Nous avons tout d'abord proposé une modélisation de la réutilisation spatiale qui a permis de transcrire le problème de réservation de bande passante dans les réseaux ad hoc prenant en compte les interférences en un problème d'optimisation (le problème du sac-à-dos multi-dimensionnel) classique. Nous avons montré que les simplifications apportées par le cadre des réseaux radio laissent le problème de réservation de bande passante NP-complet. Nous avons proposé de premières heuristiques et calculé les bornes associées concernant la taille de la bande passante utilisée.

Cette modélisation, si elle présente un intérêt théorique pour mettre en évidence les possibilités et les limites du problème de réservation de bande passante dans les réseaux ad hoc, peut difficilement être mise en pratique car elle nécessite un contrôle centralisé qui est difficile et coûteux à réaliser dans de tels réseaux. Nous avons donc conçu un protocole distribué de réservation de bande passante qui prend en compte la notion d'interférences. Ce protocole, appelé *BRuIT* (Bandwidth Reservation under InTerferences influence), est basé sur l'échange périodique de messages qui permettent de maintenir une base des mobiles qui peuvent le brouiller et leurs réservations correspondantes. Chaque mobile utilise cette connaissance pour accepter ou refuser de nouvelles réservations de bande passante. Les premières simulations effectuées avec le simulateur de réseaux NS montrent que la signalisation engendrée reste limitée, que le protocole limite très fortement les délais des communications, qu'il est suffisamment réactif à la mobilité et qu'il permet de maintenir les débits des applications acceptées.

Étude de la couche MAC dans les réseaux ad hoc Grâce à des simulations sous NS, nous avons mis en évidence des configurations de réseaux ad hoc complètement inéquitables pour certains mobiles. En effet certains mobiles, brouillés par des mobiles situés dans leur zone d'interférences, ne peuvent plus du tout accéder au médium radio alors qu'il n'y a aucun mobile dans sa zone de réception. Le protocole d'accès au médium radio utilisé dans le standard 802.11 (et implanté dans NS) est dont très inéquitable sous ces configurations. De plus, aucun des protocoles d'accès au médium proposés pour les réseaux ad hoc ne prend en compte ces phénomènes d'interférences.

D'autres simulations ont montré que la modification de certains paramètres (comme la taille de la fenêtre de contention) pouvait permettre aux mobiles fortement brouillés d'accéder un peu plus au médium. Nous sommes donc en train d'élaborer un protocole qui, comme pour le protocole *BRuIT*, permet de mettre à jour périodiquement des informations sur les brouilleurs potentiels et d'ajuster grâce à cette liste les paramètres nécessaires de la couche MAC pour pouvoir avoir accès au médium radio de manière plus équitable.

Routage dans les réseaux Nous avons travaillé sur les problèmes de routage dans les réseaux hétérogènes par l'intermédiaire du projet européen ProxiTV auquel nous participons. Ce projet a pour objectif de mettre en place une infrastructure proxy pour la télévision interactive sur Internet. Notre travail au sein de ce projet est de proposer des algorithmes de routage des portails aux serveurs proxy.

Dans un premier temps, nous avons considéré que la fonction de routage était fixée. Cette hypothèse est complètement valide car, en pratique, nous n'aurons pas accès aux tables de routage des routeurs de l'opérateur partenaire. Nous avons proposé des algorithmes qui optimisent la bande passante quelle que soit la topologie. Un des algorithmes proposés route les données sans aucune connaissance de la topologie, de manière *on-line* en supposant qu'une sous-route est une route donnée par la fonction de routage et que cette fonction ne fait pas de *source-routing*. L'autre algorithme proposé pour l'optimisation de la bande passante fonctionne aussi sous n'importe quelle topologie, mais il fonctionne de manière *off-line* avec une connaissance de la topologie du réseau au niveau du point d'entrée (portail). Il nécessite aussi de supposer qu'une sous-route est une route.

Par ailleurs, nous nous sommes intéressons au problème de routage qui optimise la latence. C'est un problème difficile même dans le cas simple de l'anneau avec la topologie du réseau connue au point d'entrée.

6.1.4 Transformations de programmes

À l'heure actuelle, le parallélisme est exploité à tous les niveaux, bien sûr dans les machines clairement identifiées comme parallèles, mais également au sein même des microprocesseurs, microprocesseurs généraux superscalaires, VLIW (Very Long Instruction Word) et également processeurs dédiés. Pour toutes ces plates-formes, aussi différentes soient-elles, l'étude des transformations de programmes révélant du parallélisme et leur automatisation est notre centre principal d'intérêt. Nos dernières études ont portées, entre autres, sur une transformation très courante mais peu étudiée du point de vue algorithmique, le *décalage d'instructions*. Cette transformation a de nombreuses applications, à la fois en synthèse d'architectures et en parallélisation automatique : nos principaux résultats la concernant portent sur la détection de boucles parallèles et le pipeline logiciel.

Décalage d'instructions et boucles parallèles Nous avons cherché à déterminer les possibilités de parallélisation de boucles à l'aide de simples décalages. Cette approche permet de trouver des ordonnancements que les techniques classiques ne trouvent pas (en tirant parti des dépendances indépendantes de la boucle, en d'autres termes de séquentialité interne au corps de la boucle) et conduit à une transformation simple et efficace du programme, ce qui n'est pas toujours le cas lors de transformations plus complexes (par exemple, les transformations affines plus générales posent des problèmes d'élimination de gardes). Nous avons établi la NP-complétude du problème (résultat en contradiction – sauf si P=NP – avec un algorithme (erroné) polynomial présenté dans la littérature) et avons proposé une formulation exacte de celui-ci sous forme d'un système de contraintes linéaires entières.

Décalage d'instructions et pipeline logiciel Notre étude concernant le pipeline logiciel a porté sur l'approche dite décomposée qui consiste à considérer un ordonnancement cyclique (effectuant le pipeline logiciel) comme la combinaison d'un décalage d'instructions (ou retiming) et d'un ordonnancement acyclique (compaction de boucles). L'intérêt de ce découpage en deux phases est notamment de pouvoir tirer parti des résultats existants tant du côté des techniques d'ordonnancement cyclique que des techniques de décalage proches algorithmiquement des optimisations de flots dans les graphes. D'un point de vue théorique, nous avons proposé un algorithme polynomial permettant de minimiser le nombre total de contraintes restant lors de la phase de compaction et qui peut très simplement se combiner à la minimisation du chemin critique ou à une contrainte plus faible par ajout d'arcs dans le graphe de dépendances. D'un point de vue pratique, nous avons confronté notre technique à des tests d'efficacité sur un large ensemble de graphes de dépendance, à l'aide d'un petit logiciel que nous avons développé (voir la section 5.1).

Décalage d'instructions et ordonnancement par étages Le problème de la plupart des algorithmes de pipeline logiciel (et un problème intrinsèque au parallélisme lui-même) est celui de l'utilisation des registres. Généralement, plus un programme contient de parallélisme, plus il faut disposer d'un grand nombre de registres pour stocker les valeurs intermédiaires du calcul. Cependant, une mauvaise transformation peut également augmenter inutilement le nombre de registres requis. Le réordonnancement par étages est une technique permettant de modifier un ordonnancement cyclique afin de diminuer ses besoins en registres sans modifier sa performance. Jusqu'alors, les seules solutions exactes connues au problème mettaient en œuvre une résolution d'un nombre exponentiel de programmes linéaires de taille exponentielle, de plus la complexité du problème demeurait inconnue. Nous avons pu prouver que le problème est en fait NP-complet au sens fort et avons proposé une amélioration des solutions connues en formulant le problème général comme un seul programme linéaire ayant un nombre polynomial de contraintes, ainsi qu'une heuristique polynomiale produisant une solution garantie par rapport à l'optimal.

HPF et le multi-partitionnement Dans le cadre d'une collaboration avec l'Université de Rice, nous avons développé une nouvelle stratégie de répartition des données, appelée multi-partionnement, qui permet pour de nombreuses applications scientifiques effectuant des "vagues" de calculs dans chaque dimension sur des données multi-dimensionnelles, de garantir un équilibrage des charges parfait. Cette technique a été complètement implémentée dans le compilateur dHPF de Rice. Jusqu'à présent, le multi-partitionnement était limité en dimension 3 à un nombre de processeurs carré (en dimension d à un nombre de processeurs de la forme n^{d-1}). Nous avons levé cette hypothèse grâce au développement d'une théorie des allocations "modulo" et donné un algorithme pour calculer un multi-partitionnement optimal dans le cas général.

Transformation de codes pour la synthèse de circuits Dans le cadre d'une collaboration avec les HP Labs, Palo Alto, et plus précisément le projet PICO (Program In Chip Out), nous avons mis au point de multiples optimisations et transformations de codes destinées à la

synthèse automatique de circuits à partir de noyaux C, transformations de boucles, méthodes d'ordonnancement, optimisations de la mémoire, transferts entre circuits, etc. Ces résultats sont pour l'instant confidentiels.

6.2 Environnements d'exécution parallèles et distribués

Participants : Gabriel Antoniu, Olivier Aumage, Alice Bonhomme, Luc Bougé, Philippe Combes, Vincent Danjean, Jean-Christophe Mignot, Raymond Namyst, Loïc Prylli.

Résumé :

Les activités dans ce thème ont été cette année consacrées à l'évolution de l'environnement d'exécution PM2 d'une part et à l'évolution de certains aspects du système BIP d'autre part.

L'activité autour de PM2 s'est déroulée suivant trois directions principales : l'amélioration de la structure de base (meilleur support des réseaux haute performance, meilleure intégration threads/communications, prise en compte des nœuds SMP), l'extension des fonctionnalités (mémoire virtuelle partagée) pour l'exécution distribuée de byte-codes *Java et l'évolution vers les grappes hétérogènes (gestion des communications multi-protocoles). L'ensemble de ces travaux a été répercuté sur la suite logicielle PM2 (PM2, Madeleine, Marcel et DSM-PM2).*

En ce qui concerne BIP, de nouveaux mécanismes de recouvrement calcul/communication ont été étudiés dans MPI-BIP et MPI-GM. D'autres améliorations au niveau du module noyau de BIP permettent d'améliorer sa sécurité et sa fonctionnalité.

Enfin au niveau applicatif, des réalisations ont été effectuées dans le domaine des serveurs vidéo multimédia et des caches parallèles pour le Web.

Cette année marque à la fois le renforcement de l'environnement PM2 en tant que support exécutif pour les grappes de stations et son ouverture sur de nouvelles architectures plus complexes ainsi que de nouvelles applications. La structure de la suite PM2 s'appuie d'une part sur une couche de communication générique (*Madeleine II*) optimisée pour les réseaux rapides et d'autre part sur une bibliothèque de threads mixte (Marcel) capable d'exploiter les architectures SMP de manière très efficace. Sur cette base, nous avons travaillé cette année à l'amélioration de divers aspects concernant l'intégration des threads et des communications.

Concernant le domaine applicatif, un système de stockage distribué haute performance pour la vidéo a été mis au point par Alice Bonhomme en collaboration avec la société Sycomore-Aérospatiale-Matra (SAM). Ce système est tolérant aux pannes, extensible et offre un système redondant basé sur la parité. Cette réalisation constitue le cœur de la thèse d'Alice Bonhomme soutenue en octobre 2001.

Un cache Web parallèle extensible a été développé dans le cadre du projet CHARM. Les points forts de cette réalisation sont :

- aiguillage asymétrique avec l'utilisation de *Linux Virtual Server* permettant de passer à l'échelle sans être limité par le débit maximal de sortie d'une seule machine ;

- décomposition du domaine d'URL entre les unités de stockage par la mise en œuvre d'une première analyse succincte de la requête permettant de rediriger la requête de manière appropriée ;
- utilisation d'un système de threads spécifiques avec un cache et l'utilisation des primitives *post/wait* de SGI qu'on pourrait comparer dans l'avenir au système Marcel plus activations ;
- utilisation de communications haute performance basées sur MPI entre le *proxy-light* et les unités de stockage Squid.

6.2.1 Extensions de l'ordonnanceur Marcel/PM2

Participants : Luc Bougé, Vincent Danjean, Raymond Namyst.

Activations. Nous avons proposé une extension du modèle des *Scheduler Activations* dans le contexte particulier des applications de calcul sur machines SMP (travaux de Vincent Danjean). L'idée du modèle original consiste à étendre le noyau du système afin d'offrir à un ordonnanceur de threads de niveau utilisateur un support permettant la gestion efficace des appels systèmes bloquants. Ce modèle souffrait cependant de multiples limitations, d'un manque de généricité et surtout d'une mise en œuvre intrinsèquement peu efficace. En privilégiant un type d'exploitation *mono-application* de la machine, nous avons proposé et implanté une nouvelle version de ce mécanisme capable de traiter la totalité des appels système tout en optimisant fortement la gestion et le nombre d'interactions depuis le noyau vers l'espace utilisateur (ce type d'interaction, symétrique d'un appel système classique, est appelé *upcall*). L'implantation a été effectuée dans le noyau Linux 2.2.x et la bibliothèque multithread Marcel. Sur une telle plate-forme, l'exécution des applications PM2 peut s'effectuer sans recourir aux classiques opérations de scrutation pour la plupart des protocoles réseaux. Cette fonctionnalité est maintenant opérationnelle et elle va être intégrée dans la distribution standard.

Scrutations dirigées par l'ordonnanceur. Nous avons proposé une interface unificatrice pour réaliser l'intégration fine des entrées-sorties (en particulier les opérations liées aux communications) avec l'ordonnanceur de threads Marcel. Dans le cas de primitives d'entrées-sorties bloquantes (utilisant les interruptions), le mécanisme d'activations décrit précédemment est automatiquement utilisé sous Linux, ou alors la délégation de l'opération à un thread noyau est effectuée pour les autres systèmes. Dans le cas contraire, c'est-à-dire si les opérations requièrent une scrutation active de la part du processeur, alors l'ordonnanceur de threads est sollicité pour prendre en charge les opérations de scrutation et pour éventuellement factoriser les opérations de même nature. Ceci permet non seulement d'assurer une période de scrutation garantie, mais aussi d'éliminer de nombreux changements de contexte inutiles lorsqu'une scrutation réseau est effectuée. Il en résulte une bien meilleure réactivité des applications PM2 qui peuvent ainsi traiter les requêtes réseaux au plus tôt. Cette fonctionnalité est maintenant opérationnelle. Une interface générique a été conçue pour permettre à l'utilisateur de spécifier de manière *externe* les opérations *interne* de Marcel pour gérer les requêtes, grouper les scrutations, etc. Il est en particulier possible de gérer la scrutation conjointe de *plusieurs* interfaces

réseaux de type différents : par exemple, TCP/Fast Ethernet et BIP/Myrinet.

6.2.2 Communications hétérogènes en Madeleine/PM2

Participants : Olivier Aumage, Luc Bougé, Philippe Combes, Raymond Namyst.

Madeleine sur architectures hétérogènes. Notre travail s'est articulé autour de l'évolution de la bibliothèque de communication *Madeleine II* vers le support efficace des architectures de type *grappes de grappes* (travaux d'Olivier Aumage). Cette étude a nécessité le développement d'outils auxiliaires spécialisés (chargeur, contrôleur de session, routeur) afin de maîtriser la complexité de la mise en œuvre de sessions de calcul distribuées sur de multiples grappes de stations. Ces travaux ont été menés conjointement à la phase de finalisation et de diffusion de la bibliothèque *Madeleine II* dans son orientation initiale, à savoir le support de communications multi-protocoles sur des grappes homogènes de stations de travail. Dans ce cadre, *Madeleine II* a fait l'objet d'une refonte partielle de son architecture interne. La nouvelle version en préparation sera nommée *Madeleine III*.

Routage au sein des grappes de grappes. L'exploitation efficace d'architectures hétérogène de type *grappes de grappes* exige que le sous-système de communication fournisse des fonctionnalités de routages intégrées. Le point crucial concerne bien évidemment les machines *passerelles* qui possèdent plusieurs cartes d'interfaces et qui doivent assurer rapidement le transfert des messages d'un type de réseau à un autre. Cela signifie soutenir un débit élevé et préserver une latence faible. Nous avons donc conçu un mécanisme permettant à Madeleine de réaliser un routage automatique très efficace de manière interne, c'est-à-dire sans que les données ne remontent dans les couches logicielles supérieures (travaux d'Olivier Aumage). Outre son efficacité (les débits observés montrent une quasi-saturation du bus PCI sur des grappes de PC), ce mécanisme apporte aussi une certaine souplesse d'utilisation puisque des threads applicatifs peuvent également s'exécuter sur les machines passerelles. Toujours dans une optique de flexibilité d'utilisation, le modèle plat de *réseau virtuel* proposé par *Madeleine III* permet l'exploitation immédiate de configurations multi-grappes complexes par des applications distribuées fondées sur le paradigme *passage de messages* et la compatibilité ascendante avec les applications *Madeleine II*. Ce modèle plat est complété par un système d'information interne décrivant avec précision l'ensemble des détails de la topologie multi-grappe sous-jacente, autorisant de ce fait l'élaboration d'algorithmes optimisés prenant en compte la localisation des passerelles inter-réseau ou la nature des équipements matériels de communication. L'adaptation et l'optimisation d'applications à *Madeleine III* s'effectue donc de manière entièrement incrémentale. Le point le plus délicat — dans un contexte aussi hétérogène et distribué — reste cependant l'orchestration du lancement *conjoint* de *Madeleine III* sur les diverses grappes, de l'initialisation ordonnée des pilotes de communication et de l'établissement synchronisé des connections : un soin particulier a donc été apporté à la mise en œuvre de cet aspect, en tenant particulièrement compte des exigences d'*extensibilité*.

Intégration de Madeleine dans d'autres bibliothèques de communication. La bibliothèque Madeleine, bien que proposant une interface générique au programmeur, possède des caractéristiques remarquables en terme d'efficacité avec notamment un temps de transfert

minimal de moins de 4 μs sur un réseau SCI (le réseau lui-même ayant une latence d'environ 2 μs). Elle a donc logiquement été intégrée comme plate-forme de communication dans plusieurs logiciels : bien évidemment dans PM2, le support d'exécution multithread distribué développé au sein de l'équipe. Nous avons développé une collaboration étroite avec le projet PARIS, IRISA (Christian Perez, Alexandre Denis) pour l'utilisation de *Madeleine II* comme support pour les ORB Corba haute performance qui y sont développés. Les premiers résultats sont très encourageants.

6.2.3 Support générique des fonctionnalités DSM en PM2

Participants : Gabriel Antoniu, Luc Bougé, Raymond Namyst.

DSM-PM2, une bibliothèque de mémoire virtuellement partagée. Nos discussions avec Assaf Schuster (Technion, Haïfa) et Frank Mueller (Humboldt Univ., Berlin) nous ont conduits à envisager d'utiliser PM2 et son iso-allocateur comme outils de base pour un système de gestion mémoire virtuellement partagée *(Distributed Shared Memory, DSM)* spécialisé pour le multithreading haute performance et la migration dynamique de threads. Nous avons donc conçu et implanté une bibliothèque de mémoire virtuellement partagée multithread (DSM-PM2, travaux de Gabriel Antoniu). L'originalité de cette réalisation est que tous les mécanismes DSM sont compatibles avec l'exécution de multiples threads : il est par exemple possible de gérer l'occurrence de plusieurs défauts de pages concurrents sur un même nœud. À notre connaissance, cette fonctionnalité n'est présente dans aucune bibliothèque DSM à part DSM-Threads (Mueller).

Une plate-forme portable pour l'implémentation des protocoles de cohérence multithreads. L'originalité de DSM-PM2 est sa conception *générique*. La bibliothèque est en fait un *harnais* supportant plusieurs modèles de cohérence : pour l'instant, la bibliothèque propose la cohérence séquentielle, la cohérence relâchée ainsi que la cohérence Java, avec deux implémentations pour chacune d'elles. En outre, l'utilisateur peut définir ses propres modèles de cohérence et/ou implanter ses propres protocoles pour une cohérence fixée. Notons qu'en ce qui concerne la cohérence séquentielle, un protocole original basé sur la migration de threads est fonctionnel et permet d'envisager des stratégies adaptatives reposant sur l'utilisation alternée de la migration des pages et de la migration des threads. À notre connaissance, seule la DSM Millipede (Schuster) propose une telle fonctionnalité. Une étude très précise des performances ainsi obtenues a été réalisée, fondée notamment sur le *benchmark* FFT de la suite Splash-2 incontournable dans ce domaine. Ce travail constitue le cœur de la thèse de Gabriel Antoniu soutenue en novembre 2001.

Intégration dans l'environnement Java Hyperion. Le système DSM-PM2, et notamment son support du modèle de cohérence mémoire Java, constitue le socle de l'environnement d'exécution Java distribué *Hyperion* développé par Phil Hatcher (Univ. New Hampshire, Durham, USA). Un premier prototype est actuellement opérationnel. L'objectif est comme toujours de maintenir la portabilité des performances sur la plus grande gamme d'architecture de nœuds, de protocoles de communication et de réseaux d'interconnexion possible. Des mesures de performances très précises ont été réalisées et une comparaison approfondie a été conduite avec le système Manta, Université libre

d'Amsterdam (Bal, Kielmann). Ce travail est présenté en détail dans la thèse de Gabriel Antoniu.

6.2.4 MPI au-dessus de PM2

Participants : Guillaume Mercier, Loïc Prylli, Raymond Namyst, Philippe Raoult.

MPICH-Madeleine est une version de MPICH utilisant *Madeleine II* comme support pour les communications. L'intérêt de la démarche réside dans l'exploitation par MPICH des capacités multi-protocoles de *Madeleine* via une seule implémentation. Les protocoles supportés actuellement sont : TCP au-dessus de Fast-Ethernet, BIP au-dessus de Myrinet et enfin SISCI au-dessus du réseau SCI. Les performances obtenues au-dessus de ces trois types de matériel sont très bonnes ; les comparaisons effectuées en termes de débit avec d'autres implémentations (mono-protocoles) de MPI au-dessus de réseaux similaires montrent que notre version de MPICH multi-protocole atteint des niveaux de performances équivalents et/ou supérieurs à ceux obtenus par ces autres implémentations. Cette année, nous avons intégré un mécanisme de *forwarding* automatique des données sur les nœuds passerelles, permettant ainsi à une application MPI d'échanger des messages entre n'importe quels nœuds d'une architecture hétérogène de manière transparente.

6.2.5 Évolutions de l'interface de communication BIP/Myrinet

Participant : Loïc Prylli.

Recouvrement calcul/communications. Nous avons poursuivi les travaux autour de l'implémentation sur l'hôte des protocoles qui vont s'exécuter en *tâche de fond* pendant que le processeur exécute des calculs non liés au réseau. Ces travaux sont destinés en particulier à concilier scrutation sans interruption et recouvrement calcul/communication dans des bibliothèques de communication comme MPI. Nous avons étudié les performances obtenues sur des applications lorsque l'on fait varier l'heuristique qui avait été proposée initialement. Si l'heuristique simple donne toujours des résultats acceptables, ceux-ci indiquent qu'il est difficile de choisir automatiquement la stratégie optimale à employer et plaident donc pour l'instauration d'un paramètre de configuration laissé à la charge de l'utilisateur.

6.2.6 Support exécutif pour les entrées-sorties multimédia

Participants : Alice Bonhomme, Jean-Christophe Mignot, Loïc Prylli.

Système de stockage pour la vidéo. Dans le cadre d'une bourse CIFRE (Alice Bonhomme) avec la société Sycomore-Aérospatiale-Matra (SAM), un système de stockage distribué haute performance pour la vidéo a été réalisé. Le point fort de ce système est la combinaison entre la tolérance aux pannes, y compris l'absence de perturbation temporaire de la distribution des flux en cas de panne, l'extensibilité (aucune opération

centralisée) et un système de redondance basé sur la parité plus économique que la duplication. Ce système est d'ores et déjà utilisé pour des démonstrations industrielles de produits destinés à la commercialisation. Cette réalisation constitue le cœur de la thèse d'Alice Bonhomme soutenue en octobre 2001.

7 Contrats industriels (nationaux, européens et internationaux)

7.1 *LHPC*

Participants : Alice Bonhomme, Luc Bougé, Jean-Christophe Mignot, Loïc Prylli.

Le *LHPC* (Laboratoire pour les Hautes Performances en Calcul) est un laboratoire commun de recherche sur les ordinateurs massivement parallèles et le calcul à haute performance créé entre l'INRIA (projet *ReMaP*), l'ENS Lyon, le CNRS et la société *SAM*. Il s'appuie sur le contrat de plan état-région avec la Région Rhône-Alpes. Le *LHPC* a été officiellement inauguré le 2 décembre 1996. La convention de collaboration a été renouvelée au 1er janvier 1999, pour une période de quatre ans.

La vocation du *LHPC* est d'être un centre de compétence dans le domaine du calcul parallèle et de réunir des partenaires d'origines variées autour d'un projet académique et industriel aux larges ambitions. Les objectifs du *LHPC* peuvent se résumer selon 4 axes principaux.

- Participer à la réalisation d'une plate-forme de calcul modulaire et extensible.
- Concevoir et développer un environnement de programmation portable mais permettant l'exploitation optimale de la plate-forme d'expérimentation.
- Mettre la plate-forme d'expérimentation à la disposition des chercheurs et des industriels et leur offrir des services d'ingénierie pour le portage d'applications existantes ou le développement de nouvelles applications, ainsi que pour les activités de recherche associées.
- Assurer la formation continue des étudiants, chercheurs et ingénieurs.

Les plates-formes de calcul les plus récentes du *LHPC* sont des piles de PC : pile de Pentium Pro + Myrinet (12 processeurs) ; pile de Power PC + Myrinet (16 processeurs) ; pile de Pentium II + SCI (16 processeurs) ; pile de Pentium II + Myrinet (8 processeurs) ; pile de Alpha + Myrinet (4 processeurs) ; serveur multimédia (16 processeurs, 400 Giga-octets de disque). Ces plates-formes vont être reliées entre elles par des liens rapides pour former des collections de grappes (*grappes de grappes*). L'objectif est de permettre des expérimentations au niveau algorithmique et au niveau système dans ce nouveau contexte.

Projet CHARM Le projet CHARM (*Cache haut débit pour les autoroutes multimédia*), financé par le SERICS, programme *Autoroutes de l'information*, s'est terminé en novembre 2000 par une revue générale.

Projet SPIHD Le projet SPIHD (*Services et Programmes pour l'Internet Haut Débit*) est mené en collaboration avec *SAM*, des professionnels de l'audiovisuel (France 3, CanalWeb, la Cinquième) et des partenaires de la région lorraine (le LORIA et SEM-Câble de l'Est). L'objectif du projet est de développer une nouvelle approche de production et de diffusion de contenus multimédia et de journaux télévisés sur l'Internet haut débit et les boucles locales. Les

expérimentations sont menées sur Nancy et sa région. Les tâches affectées à ReMaP concernent l'étude des nouvelles technologies de codage/décodage vidéo. La revue de la première année a eu lieu en décembre 2000. Le projet y a présenté ses études sur les techniques de compression, d'encodage et de décodage vidéo, ainsi que sur les techniques d'indexation vidéo et les principaux formats en émergence dans ce domaine. Pour 2001, ReMaP n'avait pas de tâche spécifique dans le projet SPIHD.

7.2 Contrat INRIA/Microsoft

Participants : Raymond Namyst, Jean-François Méhaut, Vincent Danjean, Olivier Aumage.

ReMaP participe, avec trois autres projets INRIA (Apache, Reso, Sirac), à un programme de collaboration avec la société Microsoft. L'objectif est de porter un ensemble de logiciels issus des quatre projets INRIA sur le système d'exploitation Windows 2000 et d'étudier les performances obtenues. En ce qui concerne ReMaP, le logiciel PM2 et ses composants internes (Marcel, Madeleine, DSM-PM2) ont été proposés à Microsoft. Le noyau de processus légers Marcel et la bibliothèque de communication Madeleine ont ainsi été portés sur Windows 2000 et des comparaisons de performance avec le système Linux ont exhibé un comportement similaire sur les deux systèmes. Le portage de PM2 devrait se terminer cette année. Cette collaboration avec Microsoft a en outre permis de renforcer les liens entre les projets INRIA impliqués. Le responsable de ce contrat au sein du projet ReMaP est Raymond Namyst.

7.3 Accord-cadre INRIA avec la société Alcatel

Participants : Raymond Namyst, Jean François Méhaut, Vincent Danjean, Olivier Aumage.

Un contrat de coopération entre l'entreprise Alcatel et l'INRIA (projet lyonnais ReMaP et projet rennais PARIS) a débuté en septembre 2001. Cette collaboration porte sur l'étude et la conception de mécanismes de communication pour la réalisation de bus logiciels performants destinés au support de routeurs logiciels implantés sur des grappes de PC. Il s'agit d'étendre les fonctionnalités de Madeleine de façon à offrir des services de fiabilité des communications, de dynamicité des configurations et de tolérance aux pannes. D'autre part, il s'agit d'élaborer une mise en œuvre de CORBA capable d'exploiter finement des technologies de réseaux rapides telles que VIA, Myrinet ou encore SCI. Plus précisément, il s'agit d'étudier le couplage de Madeleine avec l'interface de transport générique OCI (Open Communication Interface) dont l'intérêt principal réside dans la réutilisabilité de son implantation : on obtient une solution indépendante des ORB compatibles et ne nécessitant pas d'accès à leur code source. Le responsable de ce contrat au sein du projet ReMaP est Raymond Namyst.

7.4 Contrat Myricom

Participant : Loïc Prylli.

Le LIP développe des relations régulières avec la société Myricom (spécialisée dans le

matériel réseau pour les grappes de PC) depuis 1998, à travers différents contrats de courte durée. Le contenu de ces contrats portent essentiellement sur la fourniture d'une expertise pour le système d'exploitation Linux et pour les couches de communications MPI. Cette relation a été poursuivie cette année à travers un séjour de deux mois de Loïc Prylli chez Myricom. En plus des obligations contractuelles, ces relations sont l'un des facteurs permettant d'envisager une donation de matériel Myricom dans le cadre de la plate-forme *Grappe 200 PC* à l'INRIA.

8 Actions régionales, nationales et internationales

8.1 Actions nationales

Un grand nombre d'actions nationales du projet ReMaP se font dans le cadre du GDR ARP (*Architecture, Réseaux et systèmes, Parallélisme*) dirigé par L. Bougé, avec l'aide de M. Diaz (LAAS, Toulouse) et D. Litaize (IRIT, Toulouse).

GDR ARP, thème iHPerf. Le projet ReMaP participe activement aux activités du thème iHPerf du GDR ARP (*Architectures, réseaux et systèmes, parallélisme*) sur l'algorithmique et les outils pour le parallélisme dans les applications régulières et irrégulières : L. Bougé, F. Desprez, R. Namyst, L. Prylli, Y. Robert. URL : `http://www.prism.uvsq.fr/public/jfcollar/ihperf.html`.

GDR ARP, action Grappes. L. Bougé, R. Namyst, L. Prylli et leurs thésards participent aux activités de cette action dirigée par Jean-Louis Pazat, Projet PARIS, IRISA. URL : `http://www.irisa.fr/grappes`.

GDR ARP, action TAROT. I. Guérin Lassous et N. Schabanel participent au groupe *TAROT* du GDR ARP sur les télécommunications. I. Guérin Lassous a organisé et animé une réunion TAROT en mars 2001. ReMaP apporte une compétence algorithmique dans ce groupe de recherche animé récemment par Éric Fleury de l'INRIA/LORIA. URL : `http://wwwhds.utc.fr/TAROT/prog-03-01.txt`.

GDR ARP, thème Ordonnancement. A. Darte participe aux activités du thème *Ordonnancement* du GDR ARP qui s'intéresse aux problèmes d'ordonnancement de toute nature et qui est dirigé par Philippe Chrétienne (LIP6).

ACI Jeunes Chercheurs 2000. I. Guérin Lassous fait partie de l'Action Concertée Incitative *Jeunes chercheurs* 2000 intitulée *Modélisation et optimisation de réseaux locaux sans fil* et dirigée par Stéphane Ubéda (CITI/INSA de Lyon). Cette action, créée pour 3 ans en septembre 2000, réunit le laboratoire CITI de l'INSA de Lyon, le projet Hipercom de l'INRIA Rocquenourt et le projet ReMaP. Cette action a pour but d'étudier et de proposer des outils d'aide à la planification de réseaux locaux sans fil.

RNTL VTHD, 2 ans, 2000-2001. E. Caron, P. Combes, F. Desprez et R. Namyst participent au projet RNTL VTHD rassemblant plusieurs équipes de recherche françaises autour de l'exploitation du réseau à "Vraiment Très Haut Débit" (2.5 Gb/s) reliant plusieurs centres INRIA et centres de recherche de France Telecom. Les recherches s'articulent autour de l'exploitation de ce réseau à plusieurs niveaux : protocoles, middlewares, applications. URL : `http://www.vthd.org`.

ACI Grid RMI, 2 ans, 2002-2003. R. Namyst participe à une Action Concertée Incitative sur le thème de la Globalisation des Ressources Informatiques et des Données (GRID). Il s'agit du projet RMI (Objets distribués haute performance pour la grille de calcul) dont le responsable est Christian Perez. Un des objectifs est de concevoir une plate-forme à objets distribués capable d'exploiter de multiples technologies réseaux de manière transparente.

ACI Grid GridASP, 3 ans, 2002-2005. F. Desprez est le coordinateur de l'ACI Grid GridASP (Grid Application Service Provider) à laquelle participent également E. Caron, J.-Y. L'Excellent et G. Utard. Il s'agit d'un projet pluridisciplinaire dont le but est de fournir des services de calcul à haute performance à des chercheurs d'autres discipline (physiciens, chimistes, mathématiciens appliqués, géologues, électroniciens, etc.).

ACI Grid CGP2P, 3 ans, 2002-2005 : *Calcul Global peer-to-peer.* Il s'agit d'un projet logiciel dont l'objectif est de définir une plate-forme de calcul global sur internet de type `SETI@home` pair à pair, i.e. que tous les participants ont accès aux ressources de calcul et de stockage. Il s'agit d'un projet multi-sites (LRI+LAL+ASCI (Orsay), LIFL (Lille), LaRIA (Amiens), IMAG (Grenoble) et LIP (Lyon)). Ce projet est décomposé en 5 sous-projets (applications et interfaces utilisateurs, sécurité des ressources et des applications, stockage et fouille de données, communications et ordonnancements, interopérabilité et vérification théorique). Le coordonnateur est Franck Cappello (LRI). Gil Utard est responsable du sous-projet stockage et fouille de données.

ACI Grid Grid2, 3 ans, 2002-2005 Les membres de ReMaP participent au projet d'animation GRID2 (Groupe de Rencontres, d'Information et de Discussion sur la Globalisation des Ressources Informatiques et des Données) piloté par Jean-Louis Pazat. Plus précisément, nous sommes responsables du thème "Modèles et algorithmique" qui fédère cinq équipes.

8.2 Actions financées par la commission européenne

Projet ProxiTV. Le projet ProxiTV (*A proxy infrastructure for Internet interactive TV*) est un projet financé par la communauté européenne qui répond à l'action IV du programme IST (Information Society Technology). Ce projet a commencé au 1er octobre 2000 et dure deux ans. Les partenaires de ce projet sont EADS-Sycomore, WebfreeTV, Jet2web (filiale de Telekom Austria), Eurosport et l'INRIA. Au sein de l'INRIA, Éric Fleury (INRIA Lorraine, projet Résédas) et Isabelle Guérin Lassous (INRIA Rhône-Alpes, projet ReMaP) participent à ce projet. ProxiTV a pour objectif de mettre en place une infrastructure proxy pour la télévision interactive sur Internet. Le rôle de l'INRIA est de proposer des algorithmes de routage dans un environnement hautement hétérogène de manière optimale. L'optimalité est définie comme une fonction de coût qui peut être basée sur le prix de la route choisie ou sur la longueur de la route choisie ou sur d'autres critères à définir par les partenaires. Un démonstrateur doit aussi être réalisé.

8.3 Relations bilatérales internationales

Contrat NSF-INRIA, USA. F. Desprez et J.-Y. L'Excellent collaborent avec le projet Aladin de l'Irisa, le laboratoire LIMA-IRIT à Toulouse, le LaBRI à Bordeaux, l'Université

du Minnesota, l'Université de l'Indiana, le Lawrence Berkeley laboratory. Ce projet a pour but de développer des préconditionneurs robustes et parallèles pour la résolution de grands sytèmes numériques. Nous nous intéressons plus particulièrement aux aspects algorithmiques de la parallélisation de ces solveurs.

Programme d'Action Intégrée Aurora, France/Norvège I. Guérin Lassous est responsable du projet franco-norvégien Aurora du côté français. Ce projet implique des chercheurs de l'INRIA (Lorraine et Rhône-Alpes) et des chercheurs de l'équipe de Jan Arne Telle de l'Université de Bergen. Ce projet concerne les problèmes d'allocation de fréquences dans les réseaux cellulaires de grande taille.

Contrat NSF/INRIA, USA. L. Bougé et R. Namyst collaborent avec l'équipe de Phil Hatcher et Robert Russell, Dept. Comp. Science, Univ. New Hampshire, Durham, NH, USA, dans le domaine des supports d'exécution pour les compilateurs parallèles. Phil Hatcher a en particulier développé un compilateur data-parallèle C* et un environnement d'exécution *distribué* pour Java appelé *Hyperion*. Cette collaboration avait déjà été soutenue par un contrat NSF/INRIA conclu à l'automne 1997 pour 2 ans. Un autre contrat a été accepté en décembre 2000 sur la suite de ce travail. Il s'agit de mettre en œuvre l'environnement distribué *Hyperion* pour Java développé par Phil Hatcher au-dessus de la couche DSM (*Distributed Shared Memory*) de PM2 développée par Gabriel Antoniu.

Programme INRIA *Équipe associée à l'étranger.* Luc Bougé (dans le projet PARIS à compter du 1er septembre 2001) et Raymond Namyst ont déposé une demande dans le cadre du programme *Associated Research Team Abroad* lancé par l'INRIA au printemps 2001. Ce dossier concernait la collaboration menée depuis 1995 avec l'équipe de Phil Hatcher et Robert Russell, Dept. Comp. Science, Univ. New Hampshire, Durham, NH, USA, soutenue notamment par des contrats NSF/INRIA. Cette demande a été acceptée, avec un financement de 100 kF/an. Le projet principal de rattachement est le projet PARIS.

Projet ProCoPe, Université Humboldt, Berlin. Luc Bougé a obtenu un financement bilatéral franco-allemand ProCoPe sur l'année 2001 pour soutenir la collaboration avec l'équipe de Franck Mueller, Université Humboldt, Berlin, dans le domaine des systèmes DSM pour les environnements d'exécution multithreads. Franck Mueller est le concepteur du système DSM-Threads, l'un des plus avancé dans ce domaine. Gabriel Antoniu avait fait un séjour l'an passé dans son équipe et l'un des étudiants de Mueller était venu passer quelque temps à Lyon en retour. Malheureusement, le départ de Franck Mueller au Lawrence Livermore National Lab (LLNL) n'a pas permis de continuer ces échanges.

Projet ProCoPe, Université de Passau, Allemagne. Luc Bougé participe à une collaboration avec Paul Feautrier, Projet A3 de l'INRIA Rocquencourt, et Christian Lengauer, Université de Passau, Allemagne, consacrée à la programmation parallèle haute performance en Java. Le système Java Hyperion développé avec Phil Hatcher doit servir de support d'exécution pour les programmes Java produits par des techniques de parallélisation automatique. Un financement bilatéral ProCoPe sur l'année 2001 a été obtenu pour soutenir cette collaboration.

Contrat LIAPUNOV INRIA/Université de Moscou, Russie O. Beaumont, A. Legrand et Y. Robert collaborent avec l'Académie des Sciences Russe dans le cadre d'un

projet de l'Institut franco-russe A.M. Liapunov d'informatique et de mathématiques appliquées sur la distribution des données sur des réseaux hétérogènes de stations pour des problèmes d'algèbre linéaire. En 2001, le projet a été prolongé pour une année supplémentaire. Olivier Beaumont a été invité pendant une semaine en mai 2001 à l'Université de Moscou dans le cadre du projet.

9 Diffusion de résultats

9.1 Animation de la communauté scientifique

Responsabilité d'animation

GDR CNRS ARP. L. Bougé dirige depuis 2000 le Groupement de recherche (GDR) CNRS *Architecture, réseaux et systèmes, parallélisme* (ARP). Ce GDR est rattaché au département STIC. Il constitue l'un des 5 *GDR d'animation* du département. Comme les autres GDR d'animation, ARP est en renouvellement au 1er janvier 2002.

Coordination inter-GDR. L. Bougé assure depuis cette année la coordination (informelle !) des 5 GDR d'animation STIC, à la suite de Malik Ghallab. Il a en particulier coordonné les discussions avec la direction du département concernant le renouvellement de ces GDR.

ASP MENRT iHPerf'98. L. Bougé est responsable avec Jean-Marc Geib et Brigitte Plateau de l'Action de soutien sur programme (ASP) intitulée *Initiative informatique 1998 pour les hautes performances* (iHPerf'98).

ACI GRID du Ministère de la recherche. L. Bougé est membre du Conseil scientifique de l'ACI GRID, lancée en avril 2001. Cette ACI, consacrée à la *globalisation des ressources informatiques et des données,* est dirigée par Michel Cosnard, INRIA Sophia.

Département STIC du CNRS Yves Robert, membre du conseil de département STIC du CNRS, a animé le groupe de réflexion "Mathématiques et Informatique".

Comités de rédaction, de pilotage et de programme

Luc Bougé est vice-président du *Steering Committee* de la conférence internationale *Euro-Par* sur le parallélisme depuis 1995. Euro-Par 2001 avait lieu cette année à Manchester, avec plus de 200 participants venus du monde entier. Il a été coordinateur du *Comité de pilotage* des Rencontres francophones annuelles du Parallélisme (*RenPar*) jusqu'au printemps 2001, avant de laisser cette responsabilité à Jean-Louis Pazat. Depuis 1999, RenPar s'est associé à 2 autres conférences francophones : SympA (architecture) et CFSE (systèmes d'exploitation) pour organiser un événement commun annuel. Pour 2001, l'édition s'est tenue dans le cadre des *premières rencontres des Sciences et technologies de l'information* (ASTI 2001), à La Villette, en avril. L. Bougé assure la coordination des trois conférences.

L. Bougé a participé aux comités de programme suivants : *HIPS'01*, 6th International Workshop on High-Level Parallel Programming Models and Supportive Environments, 23-27 avril 2001, San Francisco, Californie (en marge de IPDPS 2001) ; *RTSPP'01*, 5th

Workshop on Runtime Systems for Parallel Programming, 23-27 avril 2001, San Francisco, Californie (en marge de IPDPS 2001) ; *OCT'2001*, The 2001 International Workshop on Object and Component Technologies for Cluster Computing, Brisbane, Australie, 16-18 mai 2001 (en marge de CCGrid'2001) ; *RenPar 2001*, Rencontres Francophones du Parallélisme, des Architectures et des Systèmes, Paris La Villette, France, 24-27 avril 2001 (dans le cadre de ASTI 2001) ; *Cluster 2001*, IEEE International Conference on Cluster Computing, Newport Beach, Californie, USA, 8-11 octobre 2001 ; *HIPS 2002*, 7th International Workshop on High-Level Parallel Programming Models and Supportive Environments, 15-19 avril 2002, Ft. Lauderdale (en marge de IPDPS 2002).

L. Bougé est membre du *Editorial Advisory Board* de la revue *Journal of Scientific Programming* depuis mars 2001. L. Bougé a servi comme expert pour l'appel à projet 2001 du RNTL.

Yves Robert fait partie de l'Editorial Board de *Integration, the VLSI Journal* (North Holland) dans la section *Algorithms and Architectures*. Il est membre de l'Editorial Board de *Int. Journal Supercomputer Applications* (MIT Press). Il est l'éditeur européen de *Parallel Processing Letters* (World Scientific Publishing). À compter du 1er janvier 2002, il sera éditeur de *IEEE Trans. Parallel and Distributed Systems*.

Y. Robert a été membre des comités de programme de EuroPar'2001, Manchester, UK (chair du workshop *Scheduling and load balancing*) et de Euro PVM-MPI 2001, Santorini Island, Grèce.

Alain Darte est membre du comité de rédaction de la revue *TSI* (Technique et science informatiques).

Frédéric Desprez fait partie du comité de programme du journal *Parallel and Distributed Computing Practices* (`http://www.cs.okstate.edu/~pdcp`) dont l'éditeur en chef est M. Paprzycki.

F Desprez, J.-F. Méhaut et E .Fleury (Résédas, LORIA) ont organisé le workshop *Metacomputing Systems and Applications* en marge de la conférence ICPP'2001 (Valence, Espagne, septembre 2001).

9.2 Enseignement universitaire

Responsabilités d'organisation

École doctorale MathIF et DEA DIF. L. Bougé est responsable adjoint de l'*École doctorale de mathématiques et d'informatique fondamentale* (ED MathIF) créée à Lyon à la rentrée 1999. Cette école rassemble le DEA de mathématiques pures, le DEA d'analyse numérique et le DEA d'informatique fondamentale (DEA DIF) qui fait suite au DIL en ce qui concerne le LIP et l'équipe ReSAM de B. Tourancheau.

Magistère d'Informatique et Modélisation. R. Namyst est responsable de la 2^{e} année du magistère d'informatique et modélisation.

Concours d'entrée à l'ENS Lyon Alain Darte et Yves Robert se partagent depuis quelques années la responsabilité d'organisation du concours d'entrée informatique de l'ENS Lyon.

Enseignement

DEA d'informatique de Lyon. En 2000-2001, plusieurs membres du projet enseignent au DEA d'informatique fondamentale (DIF) : O. Beaumont (*Algorithmique parallèle et hétérogène* et R. Namyst (*Supports d'exécution parallèle et distribuée*). URL : `http://www.ens-lyon.fr/DIF/`.

DEA ID (Orsay). Raymond Namyst effectue la moitié d'un cours de DEA intitulé *Supports d'exécution parallèles et distribués* dans le DEA ID d'Orsay. L'autre moitié est assurée par Jean-François Méhaut, membre extérieur de ReMaP.

INSA de Lyon I. Guérin Lassous a donné des cours sur les réseaux ad hoc en cinquième année de l'INSA de Lyon au sein du département Télécoms, Réseaux et Usages.

9.3 Autres enseignements

Yves Robert a été nommé Professeur associé d'exercice partiel à l'École Polytechnique pour l'année 2000–2001.

Frédéric Desprez a été invité à donner un cours à Supelec dans le cadre de la formation continue sur les environnements pour la parallélisation d'applications numériques.

Alain Darte a été examinateur de l'oral de mathématiques et informatique des concours d'entrée 1999 et 2000 de l'ENS-Lyon. Un recueil des exercices posés lors de ces deux sessions vient de paraître chez Dunod (voir les références bibliographiques).

Nicolas Schabanel a présenté un cours intitulé "Algorithmic Issues in Wireless Data Delivery" lors de l'Ecole d'été "DIMACS Summer School on Foundations of Wireless Networks and Applications" organisée par S. Muthukrishnan, B. Badrinath et M. Adler à DIMACS (Rutgers University, New Jersey) en août 2001.

9.4 Participation à des colloques, séminaires, invitations

Yves Robert a été invité à donner des séminaires au Computer Science Department des Universités UCLA et USC de Los Angeles.

Yves Robert a co-organisé avec A. Kalinove et A. Lastovetsky le workshop *Algorithms and tools for parallel computing on heterogeneous clusters* à la conférence PDPTA'2001, Las Vegas.

Raymond Namyst a été invité à effectuer un cours sur le thème « exploitation efficace des grappes de PC, ou l'importance d'une bonne intégration des communications et du multithreading » aux *Journées Thèmes Emergents : Cluster Computing* parrainée par le chapitre français de l'ACM.

Luc Bougé a été invité à faire un exposé dans le cadre de l'inauguration du *Réseau métropolitain universitaire* (RMU) de Lyon en juin 2001.

Nicolas Schabanel a présenté au DIMACS Mixer organisé en Octobre 2001 à ATT Research (Florham Park, New Jersey) un exposé sur le thème "Broadcasting Data on a Wireless Medium".

Nicolas Schabanel a été invité à présenter ses travaux sur le thème "Simple Algorithms to Minimize Service Time in Data Broadcast Networks" lors du "DIMACS Mini-Workshop on Quality of Service Issues in the Internet" organisé par Funda Ergun et Bulent Yener à DIMACS (Rutgers University, New Jersey).

Alain Darte a participé au colloque Compilers for Parallel Computers (CPC'01) organisé par Mike O'Boyle à Edinburgh en juin 2001.

Alain Darte a séjourné 4 mois (d'octobre 2000 à janvier 2001) à l'Université de Rice (Houston, USA), dans l'équipe de Ken Kennedy et John Mellor-Crummey. Il y a donné quatre séminaires sur les algorithmes de détection de parallélisme, les systèmes d'équations récurrentes et diverses optimisations de codes.

Alain Darte a été employé de Hewlett Packard pendant 5 mois (de février 2001 à juin 2001), dans l'équipe de Bob Rau (HP Labs, Palo Alto, USA). Il y a donné deux séminaires sur diverses optimisations de codes. Il a également donné un séminaire sur le "multi-partitioning", dans le cadre du groupe de travail "lattice theory", organisé par Gadiel Seroussi et Hendrik Lenstra.

Frédéric Desprez a donné un tutorial sur HPF, MPI et OpenMP avec Franck Cappello et Fabien Coehlo à l'Ecole d'hiver iHPerf qui s'est tenue à Aussois en décembre 2000.

Frédéric Desprez a donné un séminaire intitulé "DIET : un environnement extensible des serveurs de calcul" aux Journées JTE sur le "cluster computing" à Besançon en octobre 2001.

10 Bibliographie

Livres et monographies

[1] A. Darte, S. Vaudenay, *Algorithmique et optimisation : exercices corrigés*, *Sciences Sup*, Dunod, 2001, ISBN 2-10-005643-3.

Thèses et habilitations à diriger des recherches

[2] G. Antoniu, *DSM-PM2 : une plate-forme portable d'implémentation de protocoles de cohérence multithread pour MVP*, Thèse de doctorat, ENS Lyon, France, LIP, novembre 2001, en français.

[3] A. Bonhomme, *Conception d'un système de stockage distribué et tolérant aux pannes, pour un serveur de vidéo à la demande*, Thèse de doctorat, ENS Lyon, France, LIP, octobre 2001, en français.

[4] F. Desprez, *Calcul numérique : des bibliothèques aux environnements de metacomputing*, thèse de doctorat, Université Claude Bernard de Lyon 1, juillet 2001, Habilitation à diriger des recherches, en français.

Articles et chapitres de livre

[5] G. Antoniu, L. Bougé, P. Hatcher, M. MacBeth, K. McGuigan, R. Namyst, « The Hyperion system : Compiling Multithreaded Java Bytecode for Distributed Execution », *Parallel Computing 27*, 10, septembre 2001, p. 1279–1297.

[6] O. Aumage, L. Bougé, A. Denis, L. Eyraud, J.-F. Méhaut, G. Mercier, R. Namyst, L. Prylli, « A Portable and Efficient Communication Library for High-Performance Cluster Computing (extended version) », *Cluster Computing*, 2001, Special Issue on the Cluster 2000 Conference. À paraître.

[7] O. Aumage, L. Bougé, J.-F. Méhaut, R. Namyst, « Madeleine II : A Portable and Efficient Communication Library for High-Performance Cluster Computing », *Parallel Computing*, mars 2001, À paraître. Version étendue de [22].

[8] O. Beaumont, V. Boudet, A. Petitet, F. Rastello, Y. Robert, « A Proposal for a Heterogeneous Cluster ScaLAPACK (Dense Linear Solvers) », *IEEE Trans. Computers 50*, 10, 2001, p. 1052–1070.

[9] O. Beaumont, V. Boudet, F. Rastello, Y. Robert, « Matrix Multiplication on Heterogeneous Platforms », *IEEE Trans. Parallel Distributed Systems 12*, 10, 2001, p. 1033–1051.

[10] O. Beaumont, A. Legrand, F. Rastello, Y. Robert, « Dense Linear Algebra Kernels on Heterogeneous Platforms : Redistribution Issues », *Parallel Computing*, 2001, à paraître.

[11] O. Beaumont, A. Legrand, F. Rastello, Y. Robert, « Static LU Decomposition on Heterogeneous Platforms », *Int. Journal of High Performance Computing Applications*, 2001, à paraître.

[12] V. Boudet, F. Rastello, Y. Robert, « Alignment and Distribution is not (Always) NP-Hard », *J. Parallel and Distributed Computing 61*, 2001, p. 501–519.

[13] E. Caron, S. Chaumette, S. Contassot-Vivier, F. Desprez, E. Fleury, C. Gomez, M. Goursat, E. Jeannot, D. Lazure, F. Lombard, J. Nicod, L. Philippe, M. Quinson, P. Ramet, J. Roman, F. Rubi, S. Steer, F. Suter, G. Utard, « Scilab to Scilab$_{//}$: The OURAGAN Project », *Parallel Computing 11*, 27, octobre 2001, p. 1497–1519.

[14] A. Darte, Y. Robert, F. Vivien, « Loop Parallelization Algorithms », *in : Compiler Optimizations for Scalable Parallel Systems : Languages, Compilation Techniques, and Run Time Systems*, S. Pande (éditeur), *Lecture Notes in Computer Science, 1808*, Springer Verlag, 2001, p. 141–172.

[15] A. Darte, R. Schreiber, B. R. Rau, F. Vivien, « Constructing and Exploiting Linear Schedules with Prescribed Parallelism », *ACM Transactions on Design Automation of Electronic Systems 7*, 1, à paraître.

[16] T. Kielmann, P. Hatcher, L. Bougé, H. Bal, « Enabling Java for High-Performance Computing : Exploiting Distributed Shared Memory and Remote Method Invocation », *Communications of the ACM 44*, 10, octobre 2001, p. 110–117, Numéro spécial sur Java for High Performance Computing.

[17] J.-C. Mignot, « Cache Web : un état de l'art des techniques et prototypes », *TSI 20*, 6, 2001, p. 719–748.

[18] F. Rastello, Y. Robert, « Automatic Partitioning of Parallel Loops with Parallelepiped-Shaped Tiles », *IEEE Trans. Parallel Distributed Systems*, à paraître.

Communications à des congrès, colloques, etc.

[19] G. Antoniu, V. Bernardi, L. Bougé, « Extension de la plate-forme DSM-PM2 pour le support de protocoles de cohérence relâchée multithreads », *in : Actes des Rencontres francophones du parallélisme (RenPar 13)*, p. 175–180, Paris, La Villette, avril 2001.

[20] G. Antoniu, L. Bougé, « DSM-PM2 : A Portable Implementation Platform for Multithreaded DSM Consistency Protocols », *in : Proc. 6th International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS '01)*, *Lect. Notes in Comp. Science, 2026*, Springer-Verlag, p. 55–70, San Francisco, avril 2001.

[21] G. Antoniu, P. Hatcher, « Remote Object Detection in Cluster-Based Java », *in : Proc. 15th Intl. Parallel and Distributed Processing Symposium, 3rd Int. Workshop on Java for Parallel and Distributed Computing (JavaPDC '01)*, p. 104, San Francisco, avril 2001.

[22] O. Aumage, L. Bougé, A. Denis, J.-F. Méhaut, G. Mercier, R. Namyst, L. Prylli, « A Portable and Efficient Communication Library for High-Performance Cluster Computing »,

in : IEEE Intl Conf. on Cluster Computing (Cluster 2000), p. 78–87, Technische Universität Chemnitz, Saxony, Allemagne, novembre 2000.

[23] O. Aumage, L. Eyraud, R. Namyst, « Efficient Inter-Device Data-Forwarding in the Madeleine Communication Library », *in : Proc. 15th Intl. Parallel and Distributed Processing Symposium, 10th Heterogeneous Computing Workshop (HCW 2001)*, p. 86, San Francisco, avril 2001.

[24] O. Aumage, G. Mercier, R. Namyst, « MPICH/Madeleine : A True Multi-Protocol MPI for High-Performance Networks », *in : Proc. 15th International Parallel and Distributed Processing Symposium (IPDPS 2001)*, IEEE, p. 51, San Francisco, avril 2001.

[25] O. Beaumont, V. Boudet, A. Legrand, F. Rastello, Y. Robert, « Geometrical Problems Arising from the Design of Parallel Algorithms on Heterogeneous Platforms », *in : Approximation and On-line Algorithms*, E. Bampis, K. Jansen, C. Kenyon (éditeurs), *LNCS*, Springer Verlag, à paraître.

[26] O. Beaumont, V. Boudet, A. Legrand, F. Rastello, Y. Robert, « Heterogeneous Matrix-Matrix Multiplication, or Partitioning a Square into Rectangles : NP-Completeness and Approximation Algorithms », *in : EuroMicro Workshop on Parallel and Distributed Computing (EuroMicro'2001)*, IEEE Computer Society Press, p. 298–305, 2001.

[27] O. Beaumont, V. Boudet, Y. Robert, « The Iso-Level Scheduling Heuristic for Heterogeneous Processors », *in : PDP'2002, 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, IEEE Computer Society Press, à paraître.

[28] O. Beaumont, A. Legrand, Y. Robert, « Master-Slave Tasking with Heterogeneous Processors », *in : 2001 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2001)*, CSREA Press, p. 857–863, 2001.

[29] O. Beaumont, A. Legrand, Y. Robert, « The Master-Slave Paradigm with Heterogeneous Processors », *in : Cluster'2001*, D. Katz, T. Sterling, M. Baker, L. Bergman, M. Paprzycki, R. Buyya (éditeurs), IEEE Computer Society Press, p. 419–426, 2001.

[30] O. Beaumont, A. Legrand, « Data Allocation Strategies for Dense Linear Algebra on Two-Dimensional Grids with Heterogeneous Communication Links », *in : 2001 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2001)*, CSREA Press, 2001.

[31] K. Bertet, C. C., I. Guérin Lassous, L. Viennot, « Impact of Interferences on Bandwidth Reservation for Ad Hoc Networks : A First Theoretical Study », *in : Proceedings of the IEEE Globecom 2001*, San Antonio, USA, 2001. À paraître.

[32] A. Bonhomme, « Scalability Issues in a Reliable Distributed Video Storage System », *in : Proceedings of the IASTED International Conference on Multimedia Systems and Applications (IMSA)*, p. 418–423, Hawai, août 2001.

[33] V. Boudet, Y. Robert, « Scheduling Heuristics for Heterogeneous Processors », *in : 2001 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2001)*, CSREA Press, p. 2109–2115, 2001.

[34] E. Caron, « Inversion matricielle parallèle out-of-core », *in : Treizièmes Rencontres Francophones du Parallélisme des Architectures et des Systèmes*, Paris, La Villette, avril 2001.

[35] D. Chavarría-Miranda, A. Darte, R. Fowler, J. Mellor-Crummey, « Generalized Multipartitioning », *in : Second Annual Los Alamos Computer Science Institute (LACSI) Symposium*, Santa Fe, NM, octobre 2001.

[36] A. Darte, Y. Robert, F. Vivien, « Loop Parallelization Algorithms », *in : Compiler Optimizations for Scalable Parallel Systems : Languages, Compilation Techniques and Run Time Systems*, *LNCS 1808*, Springer Verlag, p. 141–171, 2001.

[37] F. Desprez, M. Quinson, F. Suter, « Dynamic Performance Forecasting for Network Enabled Servers in a Metacomputing Environment », *in : Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2001)*, *3*, CSREA Press, p. 1421–1427, juin 2001.

[38] F. Desprez, F. Suter, « Mixed Parallel Implementations of the Top Level Step of Strassen and Winograd Matrix Multiplication Algorithms », *in : Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS'01)*, San Francisco, avril 2001.

[39] F. Desprez, F. Suter, « Produit de matrices, Strassen et parallélisme mixte », *in : Treizièmes Rencontres Francophones du Parallélisme*, Paris, La Villette, avril 2001.

[40] D. Dhoutaut, D. Laiymani, « A CORBA-Based Architecture for Parallel Applications : Experimentations with the WZ Matrix Factorization », *in : Proceedings of the first IEEE/ACM Internatianol Symposium on Cluster Computing and the Grid*, p. 442–449, 2001.

[41] I. Guérin Lassous, J. Gustedt, M. Morvan, « Graphs According a Coarse Grained Approach : Experiments with PVM and MPI », *in : Proceedings of European PVM/MPI Users' Group Meeting (EuroPVM/MPI 2000)*, J. D. P. K. N. Podhorszki (éditeur), *LNCS*, 1908, Springer Verlag, p. 72–79, Hongrie, septembre 2000.

[42] I. Guérin Lassous, J. Gustedt, « Portable List Ranking : an Experimental Study », *in : Proceedings of Workshop on Algorithm Engineering (WAE 2000)*, *LNCS*, 1982, Springer Verlag, p. 111–123, Allemagne, septembre 2000.

[43] I. Guérin Lassous, E. Thierry, « Generating Random Permutations in the Parallel Coarse Grained Models Framework », *in : Proceedings of the International Conference on Principles of Distributed Systems (OPODIS 2000)*, F. Butelle (éditeur), *International Journal of Informatics*, p. 1–16, Paris, décembre 2000.

[44] G. Huard, « Parallélisation de boucles par décalage d'instructions », *in : Actes des Rencontres francophones du parallélisme (RenPar 13)*, Paris, La Villette, avril 2001.

[45] F. Lombard, M. Quinson, F.Suter, « Une approche extensible des serveurs de calcul », *in : Treizièmes Rencontres Francophones du Parallélisme des Architectures et des Systèmes*, p. 79–84, Paris, La Villette, avril 2001.

[46] M. Quinson, « Un outil de modélisation de performances dans un environnement de metacomputing », *in : 13ième Rencontres Francophones du Parallélisme des Architectures et des Systèmes*, p. 85–90, Paris, La Villette, avril 2001.

Rapports de recherche et publications internes

[47] O. Beaumont, L. Carter, J. Ferrante, A. Legrand, Y. Robert, « Bandwidth-Centric Allocation of Independent Tasks on Heterogeneous Platforms », *rapport de recherche n°2001-25*, LIP, ENS Lyon, France, juin 2001.

[48] A. Bonhomme, « Cost Analysis of a Distributed Video Storage System », *Rapport de recherche n°RR2001-23*, LIP, ENS Lyon, Lyon, France, juin 2001.

[49] C. Chaudet, I. Guérin Lassous, « BRuIT : Bandwidth Reservation under InTerferences influence », *rapport de recherche n° 2001-29*, LIP/ENS Lyon, juillet 2001.

[50] A. Darte, G. Huard, « Loop Shifting for Loop Parallelization », *rapport de recherche n°RR2000-22*, LIP, ENS-Lyon, France, 2000, accepté à STACS2002.

[51] A. Ferreira, I. Guérin Lassous, K. Marcus, A. Rau-Chaplin, « Parallel Computation on Interval Graphs : Algorithms and Experiments », *rapport de recherche n° 2000-43*, LIP/ENS Lyon, décembre 2000.

[52] J.-C. MIGNOT, « Distribution de contenus multimédias par flot continu : état de l'art », *Rapport de recherche n°RR2000-40*, LIP, ENS Lyon, Lyon, France, novembre 2000, Rapport de contrat CHARM. Aussi disponible comme rapport de recherche RR-4073 de l'INRIA Rhône-Alpes, `http://www.inria.fr/rrrt/rr-4073.html`.

[53] J.-C. MIGNOT, « Les nouveaux codecs : un état de l'art », *Rapport de recherche n°RR2000-41*, LIP, ENS Lyon, Lyon, France, décembre 2000, Rapport de contrat SPIHD. Aussi disponible comme rapport de recherche RR-4074 de l'INRIA Rhône-Alpes, `http://www.inria.fr/rrrt/rr-4074.html`.