
Projet ReMaP

Régularité et Parallélisme Massif

Localisation : *Ecole Normale Supérieure de Lyon – Inria Rhône-Alpes*

Mots-clés : data-parallélisme, parallélisation automatique, environnements, bibliothèques.

1 Composition de l'équipe

Responsable scientifique

Yves Robert, Professeur ENS Lyon et IUF

Personnel Inria

Frédéric Desprez, CR

Secrétaire

Valérie Roger-Ubéda, adjoint administratif ENS Lyon, 20% sur le projet

Personnel des établissements partenaires (URA LIP-Imag)

Luc Bougé, Professeur ENS Lyon

Alain Darté, CR CNRS

Eric Fleury, ATER ENS Lyon

Pierre Fraigniaud, CR CNRS

Jean-Christophe Mignot, IR CNRS, 50% sur le projet

Bernard Tourancheau, CR CNRS

Karine Van Heumen, IR contractuelle ENS Lyon, 20% sur le projet

Ingénieur expert

Nicolas Bert, du 17 juillet 1995 au 31 janvier 1996

Chercheurs doctorants

David Cachera, allocataire moniteur normalien

Pierre-Yves Calland, allocataire région Rhône-Alpes

Johanne Cohen, allocataire DRET

Stéphane Domas, allocataire MENESR

Christian Perez, allocataire moniteur normalien

Loïc Prylli, allocataire moniteur normalien

Cyril Randriamaro, allocataire DRET

Georges-André Silber, allocataire DRET

Sandrine Vial, allocataire MENESR

Frédéric Vivien, allocataire moniteur normalien

2 Présentation du projet

Le projet *ReMaP* est un projet commun entre le CNRS, l'ENS Lyon, et l'Inria (UR Inria Rhône-Alpes), localisé dans les locaux du LIP, à l'École Normale Supérieure de Lyon. *ReMaP* a pour objectif de développer les techniques de programmation d'algorithmes réguliers sur les systèmes parallèles, en étudiant aussi bien les environnements de programmation et les langages adaptés que les bibliothèques et les compilateurs.

Les machines parallèles à mémoire distribuée, ainsi que les réseaux de stations de travail, font coopérer plusieurs centaines de processeurs, et apparaissent aujourd'hui comme la seule réponse technique et économique aux besoins croissants de puissance de calcul, de taille mémoire et de débit d'entrées/sorties, maintenant nécessaires pour résoudre de nombreux problèmes scientifiques. Les algorithmes dits réguliers sont les seuls pour lesquels il soit possible à l'heure actuelle de développer un véritable environnement de programmation permettant de manipuler des programmes indépendamment de la taille des données traitées. Par ailleurs, ce type d'algorithmes recouvre une bonne partie des noyaux de base de l'algorithmique matricielle et du traitement d'images et du signal.

ReMaP tend à montrer que le modèle de programmation data-parallèle est un cadre adéquat pour concevoir les algorithmes, les environnements de programmation, et les techniques de compilation, d'optimisation et de validation associées, qui permettent de maîtriser les architectures massivement parallèles ou distribuées existantes.

3 Actions de recherche

La présentation du projet *ReMaP* s'articule autour de deux axes : les techniques de parallélisation (semi)-automatique, et les environnements de programmation.

3.1 Parallélisation automatique et algorithmique parallèle

Participants : Alain Darté, Frédéric Desprez, Yves Robert

Nous décrivons nos travaux liés aux techniques de parallélisation en trois rubriques : les avancées théoriques en compilation-parallélisation, l'environnement intégré d'aide à la parallélisation *TransTool*, et les résultats obtenus en algorithmique parallèle.

3.1.1 Régularité et transformations de programmes

L'un des objectifs majeurs de la plate-forme *TransTool* [90] (voir section 3.1.2) est de développer et d'intégrer de nouvelles stratégies permettant de transformer (semi-) automatiquement des portions de code séquentiel (principalement des boucles à la Fortran) en codes annotés par des directives HPF (*High Performance Fortran*). 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.

Les maîtres-mots de notre recherche sont, plus que jamais, régularité et parallélisme : *description* régulière des dépendances, c'est-à-dire de la séquentialité, *détection* du parallélisme dans les structures régulières, *exploitation* régulière de ce parallélisme. Nous avons organisé nos recherches suivant ces mêmes trois axes. Les sections suivantes donnent un bref aperçu des problèmes que nous avons abordés et de nos résultats majeurs.

a) Dépendances et régularité. Une transformation de code est valide si elle ne change pas la sémantique du code, c'est-à-dire si elle respecte certaines contraintes de précédence (appelées dépendances) entre calculs. En général, on exprime ces dépendances par une structure finie, qui peut fournir une description exacte des dépendances du code séquentiel lorsque le code lui-même est régulier, mais

qui en est, en général, une sur-approximation. Plus l'approximation est fine, plus il est coûteux de la construire, mais plus grand est l'ensemble des transformations valides et plus grandes sont les possibilités d'optimisation. Cependant, pour un objectif donné, élargir l'ensemble des transformations valides peut être inutile s'il contient déjà la transformation optimale.

Pour tenter de mieux comprendre ce problème de représentation par une structure régulière, nous avons caractérisé, pour une représentation donnée, l'algorithme le plus simple permettant de détecter le parallélisme maximal implicite décrit par la structure régulière. Nous avons montré l'optimalité d'un algorithme bien connu, l'algorithme d'Allen et Kennedy, pour la représentation par niveaux de dépendance [109, 85]. Nous avons proposé une classification des algorithmes de détection de parallélisme, en fonction de leur optimalité pour une représentation donnée [107].

Le lien existant entre la complexité d'une structure régulière et le parallélisme maximal qu'elle peut implicitement décrire est maintenant parfaitement clair pour la représentation par un graphe étiqueté par niveaux (algorithme d'Allen et Kennedy) et pour la représentation par vecteurs de direction, sans structure de graphe (algorithme de Wolf et Lam). Le problème reste ouvert pour les dépendances affines, du fait du saut de complexité entre les différentes représentations des dépendances. On peut cependant en dire plus en définissant une structure régulière intermédiaire, les graphes réduits polyédriques : ceux-ci permettent d'exploiter la structure de graphe et l'information sur les vecteurs de distance. Nous avons proposé un algorithme optimal pour cette représentation [110].

b) Détection du parallélisme. Nous avons poursuivi nos travaux sur la détection du parallélisme dans les structures régulières. Notre but est d'exposer automatiquement au programmeur le parallélisme maximal qui peut être détecté au niveau des boucles de son code et qui peut être également décrit par des boucles. Dans le contexte d'HPF, ceci revient à annoter automatiquement le code, lorsque cela est possible, par des directives `!hpf$ independent`. Les principales techniques permettant de réaliser cet objectif sont des techniques d'ordonnement. Nos principaux résultats concernant les ordonnements mono-dimensionnels figurent dans [83]. La prise en compte de structures plus complexes telles que les graphes réduits polyédriques (voir paragraphe précédent) nous a amenés à développer des techniques d'ordonnements multi-dimensionnels [108, 110, 84].

Les méthodes de parallélisation citées plus haut sont essentiellement des techniques de recherche de parallélisme à grain fin, au niveau des boucles. Elles conviennent plutôt à des machines vectorielles ou à des machines parallèles à mémoire partagée. Pour certains types de machines telles que les processeurs VLIW, un grain encore plus fin est nécessaire, et le pipeline logiciel (*software pipelining*) est utile. Nous en avons proposé une nouvelle heuristique pour le pipeline logiciel avec contraintes de ressources [105] (la meilleure garantie à ce jour). Pour d'autres types de machines telles que les machines à mémoires distribuées, un grain plus gros peut être nécessaire si le degré de parallélisme est trop faible et les techniques dites de "tiling" sont utiles. Nous poursuivons actuellement nos recherches dans ce sens.

c) Exploitation du parallélisme. Détecter le parallélisme d'une application est une chose, l'exploiter en est une autre. Une directive telle que `!hpf$ independent` est une indication pour le compilateur que certains calculs *peuvent* être faits en parallèle. Elle ne précise pas où sont les données et où sont effectués les calculs. Pour cela, le programmeur, le compilateur ou le pré-compilateur doivent posséder des techniques de placement des données et des calculs, qui prennent en compte un modèle de machine-cible et son modèle de communications.

La communauté scientifique est toujours à la recherche de la "bonne façon" d'aborder ce problème. En l'absence d'une solution définitive, nous avons, nous aussi, étudié de nombreux problèmes de placement, sous des aspects très divers, du fait de la variété de modèles de machines physiques : évaluation d'expressions de tableaux [97, 80] permettant de recouvrir calculs et communications sur des tableaux de processeurs, placement "modulaire" [82] pour tenter de formaliser les placements cycliques, placement avec contraintes de ressources [114, 91] pour des réseaux de type systolique, génération automatique de directives de placement pour HPF pour la minimisation des communications [115, 92, 90, 113].

Les travaux précédents visent à prendre en compte le coût des communications *entre* processeurs. Une autre optimisation importante est la minimisation de la mémoire *d'un* processeur. En effet, pour exploiter le parallélisme d'une application (souvent écrite pour machines séquentielles avec le souci de réutiliser la mémoire), on est souvent contraint d'"expanser" la mémoire, c'est-à-dire par exemple de transformer des scalaires en tableaux. Caractériser l'expansion minimale est un problème difficile. Nous donnons des éléments de réponse dans [103, 104].

3.1.2 Projets HPFIT et *TransTool*

L'utilisation du langage HPF pour la parallélisation de codes industriels s'avère compliquée et peu efficace si elle n'est pas utilisée de manière adéquate. Peu d'outils existent pour aider l'utilisateur dans la recherche de parallélisme et dans l'insertion semi-automatique de directives HPF.

Nous avons démarré les projets HPFIT¹ et *TransTool*² autour de la parallélisation semi-automatique d'applications scientifiques écrites en Fortran77 [100, 101].

Un des problèmes rencontrés par les développeurs d'outils de parallélisation est la réécriture complète de nombreux composants d'outils tels que l'éditeur, l'analyseur syntaxique, l'analyseur de dépendances, le débogueur, l'analyseur de performance (*profiler*), etc. Ces développements sont coûteux en temps et en main d'œuvre et nous avons choisi pour notre projet de partir de composants de base existants et gratuits. De nombreux outils puissants sont déjà disponibles, mais malheureusement ils ne sont pas intégrés et leur utilisation reste donc problématique. Le projet HPFIT est né de la collaboration avec le LaBRI à Bordeaux et le GMD/SCAI à Bonn. Récemment, l'équipe de J.L. Dekeyser au LIFL est venu rejoindre les trois laboratoires fondateurs. HPFIT intégrera donc un certain nombre d'outils développés dans différents laboratoires autour d'un même outil de parallélisation : *TransTool*.

TransTool, noyau du projet HPFIT et développé au LIP dans le cadre du projet *ReMaP*, est un outil de restructuration semi-automatique de programmes écrits en F77 vers HPF. Il contient un éditeur (XEmacs), un analyseur syntaxique (issu des outils de compilation utilisés par Adaptor), un analyseur de dépendances (Petit de l'université du Maryland) et un noyau d'optimisation et de parallélisation automatique. Ce noyau contient un certain nombre de développements théoriques et logiciels effectués dans le cadre du projet *ReMaP*.

La version 1.0 de *TransTool* est en cours de dépôt.

3.1.3 Algorithmique parallèle

Notre but est de développer une expertise dans la parallélisation d'algorithmes. Nous travaillons principalement sur l'algèbre linéaire dense et sur la parallélisation d'applications utilisant de tels noyaux de calculs.

ScaLAPACK est une bibliothèque parallèle d'algèbre linéaire portable et efficace développée à l'université du Tennessee. Elle peut être optimisée et complétée. Cette année, nous avons terminé l'optimisation de la factorisation *LU* de ScaLAPACK en donnant une méthode de calcul de taille optimale des blocs et en optimisant les communications à l'aide de recouvrements calculs/communications [135, 116].

Nous avons commencé une collaboration avec F. Tisseur de l'université de St-Etienne dans le cadre de la parallélisation d'un nouvel algorithme de recherche de valeurs propres pour les matrices symétriques, l'algorithme de Yau et Lu. Cet algorithme contient de nombreux noyaux qui peuvent être parallélisés efficacement. La bibliothèque ScaLAPACK ne contient pas toutes les routines nécessaires. Une routine de produit de matrices symétriques a donc été étudiée et développée. En tenant compte de la symétrie de la matrice, nous pouvons diviser le temps d'exécution par deux dans la majorité des cas.

¹High Performance Fortran Integrated Tools

²Translation Tool

Les communications sont la principale source de perte de performances lors de la parallélisation d'un algorithme sur une machine à mémoire distribuée. Il convient donc de les recouvrir le plus possible, mais ce recouvrement conduit à des codes compliqués et peu portables. De plus, dans le cas des calculs pipelinés, cette optimisation conduit à des problèmes théoriques importants pour le choix du grain de calcul optimal. Une bibliothèque avait donc été développée pour simplifier l'utilisation des recouvrements : les LOCCS (*Low Overhead Communication and Computation Subroutines*). Nous avons amélioré la bibliothèque LOCCS et bâti la bibliothèque OPIUM (*Optimal Packet Size Computation Methods*) de calcul de grain optimal pour la bibliothèque LOCCS. Si, dans le cas d'une complexité linéaire des calculs et des communications, ce calcul est simple, ce n'est plus vrai pour des degrés supérieurs ou des valeurs logarithmiques. Nous avons donc développé des algorithmes pour le calcul des tailles de paquets dans le cas général [112].

L'utilisation d'un langage de haut niveau comme High-Performance Fortran (HPF) relègue la majeure partie des optimisations au niveau du compilateur et du support d'exécution. L'utilisation d'une bibliothèque comme les LOCCS doit donc être intégrée au compilateur : c'est pourquoi nous avons effectué ce travail en collaboration avec Thomas Brandes (GMD/SCAI), au sein du compilateur Adaptor [102].

3.2 Outils/environnements pour la programmation parallèle

Participants : Luc Bougé, Pierre Fraigniaud, Bernard Tourancheau

Nous présentons ici nos résultats en matière de bibliothèques de communication, d'outils pour la programmation parallèle, et d'équilibrage de charge.

3.2.1 Communications structurées et routage

Les travaux effectués au sein de cette action de recherche depuis le début du projet *ReMaP* ont pour principal objectif de soulager le programmeur parallèle de tout ce qui touche à l'acheminement de ses messages entre les différents processeurs mis en jeu par son application. Cet objectif se réalise au moyen de bibliothèques de communications et par l'utilisation de fonctions de routage efficaces. Nos recherches ont été principalement menées dans les directions suivantes :

- Algorithmes de communications structurées ;
- Algorithmes d'approximation pour les communications ;
- Routage *wormhole* et diffusion partielle (*multicasting*) ;
- Routage compact.

La parallélisation de problèmes utilisant des structures de données régulières fait souvent apparaître des mouvements de données réguliers entre les processus parallèles. C'est la raison qui a conduit par exemple les concepteurs de PVM et de MPI à introduire des procédures de communications collectives directement accessibles au sein de leur environnement. Les deux premiers thèmes cités ci-dessus ont pour objectif la réalisation de procédures de communications structurées optimisées en fonction du type de machine utilisée (voir la section 3.2.1 "bibliothèque de macro-communications"). Les deux derniers thèmes sont dédiés à la réalisation de communications quelconques entre les processeurs d'une machine parallèle ou les noeuds d'un réseau. Bien que le projet *ReMaP* ait surtout pour objectif de profiter de la régularité des problèmes pour générer des communications structurées, il persiste toujours des mouvements non réguliers nécessitant une fonction de routage efficace.

a) Algorithmes de communications structurées. Le parallélisme régulier induit des communications régulières (diffusion de pivot dans la méthode de Gauss, transposition de matrices, échange total entre les processeurs dans la méthode de Durand-Kerner). Une grande partie de notre activité consiste

en l'optimisation de ce type de communication en fonction des paramètres de la machine (mode de routage, topologie, etc.).

Ces dernières années, nous avons focalisé notre recherche sur les nouveaux modes de routage disponibles au sein d'une machine parallèle ou d'un réseau : routage *wormhole*, commutation de circuits, routage *cut-through* (machines parallèles, protocole ATM, réseaux optiques), bus, etc. Ces modes ont en commun la possibilité de communications distantes dont le coût n'est, pour la plupart, que faiblement lié à la longueur du chemin parcouru.

Nous avons une bonne expérience dans le mode point-à-point (voir [95]). Nos recherches récentes sur les nouveaux modes de routage ont donné lieu à publication, en particulier [106]. Cette étude est directement liée à la réalisation de bibliothèques de macro-communication telle que celle réalisée dans le cadre de l'action industrielle avec la société Matra (LHPC).

Perspectives. Les problèmes théoriques soulevés par ce type d'étude sont nombreux et forment un domaine de recherches à eux seuls. Nous avons cependant choisi d'orienter nos recherches dans une thématique plus directement liée aux protocoles de communication dans les réseaux.

b) Algorithmes d'approximation pour les communications. La plupart des problèmes de communication globales structurées donnent lieu à des problèmes de décision NP-complets. Cependant, les topologies envisagées comme candidates plausibles pour interconnecter les processeurs d'une machine parallèle ne dépassent guère la douzaine. Ce fait a permis une étude quasi exhaustive des problèmes de communications globales structurées en parallélisme.

Deux événements relativement récents ont remis en cause cet état de fait. D'une part, la plupart des machines parallèles sont (ou devraient être) multi-utilisateurs sous forme de partage de l'espace. Ce partage spatial détruit la structure régulière de l'ensemble des processeurs alloués à un utilisateur. D'autre part, l'utilisation pour le calcul parallèle de réseaux comprenant des processeurs aux caractéristiques très diverses ne permet pas de concevoir des algorithmes reposant sur une topologie fixe et homogène. Tout ceci conduit à la recherche d'algorithmes d'approximation susceptibles de fournir, quelle que soit l'architecture sous-jacente, un algorithme efficace de communications structurées. Nous avons ainsi conçu plusieurs méthodes d'approximation pour l'échange total pour des réseaux quelconques ou pour des grilles partielles [128], dans un mode point-à-point. Dans le cas de la diffusion, nous avons donné une méthode originale permettant d'obtenir des protocoles *furtifs* et optimaux en temps en mode *cut through*.

Perspectives. Notre objectif principal est maintenant de formaliser l'hétérogénéité des réseaux afin d'en déduire les conséquences sur les coûts et les méthodes de communication.

c) Routage *wormhole* et diffusion partielle. Toute machine parallèle distribuée moderne et tout réseau dispose d'un mécanisme de routage automatique des messages de leur source vers leur destination. Un des modes les plus utilisés dans les machines parallèles est le mode *wormhole* permettant d'optimiser à la fois le délai d'acheminement des messages (circuit plutôt que commutation de message) et l'emploi des ressources utilisées pour les communications (nombre de canaux proportionnel à la longueur du message).

Le principal problème lié à ce mode de commutation est la possibilité d'interblocage, particulièrement en mode adaptatif. Nous avons caractérisé la présence d'interblocage dans un cadre très général contenant la plupart des types de fonctions de routage usuelles [138].

Ce résultat peut également se généraliser à la diffusion partielle, communication typique du contrôle et des barrières de synchronisation. Nous avons de plus proposé un ensemble de protocoles de diffusion partielle optimisée pour la grille.

Perspectives. Une partie de nos résultats dépasse le simple cas du *wormhole*. Certains pourraient en effet se généraliser au cas de réseaux ATM (en particulier la problématique de la diffusion partielle). Nous comptons travailler dans cette direction dans les mois à venir.

d) Routage compact. Ce domaine de recherche est certainement celui de cette action dont les retombées pratiques peuvent paraître les plus faibles. Il répond cependant à une question fondamentale sur la “connaissance” dont doit disposer un routeur afin de permettre des communications efficaces. De fait, une partie significative des recherches sur le routage compact ont des applications significatives. Le fameux routage *par intervalle* n’est qu’une méthode parmi d’autres de routage compact.

Nous avons établi la complexité de la quantité d’information nécessaire à un routeur pour un routage optimal. Ce résultat montre que si des méthodes spécifiques telles que le routage par intervalle ou le routage booléen peuvent être utilisées efficacement dans certain cas, router dans un réseau arbitraire peut nécessiter de stocker une table de routage. Il établit qu’il n’existe pas de méthode de routage compacte universelle [124, 93, 125].

Perspectives. Nous travaillons actuellement en collaboration avec le LaBRI sur une formalisation plus générale du routage afin de tenir compte de la multitude de protocoles utilisés sur les réseaux généralistes.

e) Bibliothèque de macro-communications Les travaux menés au sein de cette tâche ont consisté en l’optimisation sur la machine CAPITAN de Matra Cap-Système (MCS) des procédures de communications structurées (*collective communications*) accessibles sous MPI : diffusion, échange-total, distribution, concentration, barrière, etc.

La première partie de ce travail a consisté à modéliser le comportement des communications sur la machine CAPITAN, puis à déduire de cette modélisation des algorithmes efficaces de communications structurées, et enfin à expérimenter les algorithmes. Ces trois phases sont maintenant terminées et ont donné lieu à un rapport interne MCS. Les performances obtenues sont encourageantes. Par exemple, notre algorithme de diffusion se révèle plus de deux fois plus rapide que celui natif de la machine.

Perspectives. Nous terminerons au cours de l’année 1997 le développement d’un prototype de bibliothèque de procédures de communications structurées accessibles sur CAPITAN à partir de MPI.

3.2.2 Environnement d’exécution et bibliothèques

Nos recherches et développements ont pour but de créer des outils logiciels qui facilitent la mise au point des programmes et l’exploitation des machines parallèles.

Environnement d’exécution. Dans le cadre de la collaboration EUROTOPS avec Matra Cap-Systèmes, nous avons réalisé un environnement d’exploitation multi-utilisateur (interface et fonctionnalités du type Intel Paragon) adapté à la machine, originellement temps-réel, CAPITAN. Nos recherches concernent la construction d’une surcouche du système d’exploitation parallèle, et son optimisation pour réduire le surcoût logiciel portant sur la latence des communications.

D’autre part, pour évaluer et optimiser des architectures nouvelles ou non encore accessibles, nous avons réalisé un simulateur distribué de machines parallèles [129]. Nos recherches portent essentiellement sur la simulation du réseau de communication lors de l’exécution d’applications parallèles pour fournir une évaluation réaliste du temps d’exécution de l’application. La version 1 de ce simulateur est en cours d’intégration dans *TransTool*.

Bibliothèques. Dans le cadre de nos collaborations CNRS-PICS avec l'université du Tennessee à Knoxville, nous avons réalisé un logiciel de redistribution de données matricielles pour les cas bloc-cyclique [117]. Ce logiciel est intégré à la librairie scientifique parallèle ScaLAPACK.

D'autre part, dans le cadre de notre collaboration CEE-KIT avec les universités de Mons et Oujda, et d'un contrat avec ITV Wind Concept, nous validons une approche incrémentale de la parallélisation de code grâce aux redistributions de données ci-dessus et aux méthodes algorithmiques développées dans l'action décrite Section 3.1.3. Ces travaux ont permis la parallélisation d'un code de simulation 3D d'écoulement de fluide autour d'une aile.

Réseaux haut débit pour le calcul parallèle. Nous nous sommes intéressés aux problèmes soulevés par l'utilisation des réseaux haut débit pour le calcul parallèle ou distribué.

Pour cela, nous avons étudié le comportement d'une plate-forme ATM du commerce (FORE) en réseau local. Nous avons modélisé les communications types du calcul parallèle, et déterminé précisément les goulots d'étranglement dus à la conception du pilote des cartes réseaux et à la bande passante mémoire du processeur associé.

3.2.3 Equilibrage de charge et parallélisme de données

Nous nous sommes intéressés à la faisabilité de l'intégration d'un service d'équilibrage de charge dans un langage à parallélisme de données. Nous avons choisi dans un premier temps le langage HPF (*High Performance Fortran*), car il existe des compilateurs dont les sources sont disponibles, par exemple Adaptor, et pour établir une synergie avec le projet HPFIT.

Nous avons d'abord étudié les processus légers dans les langages data-parallèles du point de vue de son coût. Notre approche est d'utiliser les processus légers (*threads*) comme unité de migration [140]. Il est apparu que la surcharge était tout à fait acceptable. La deuxième partie de l'année a été consacrée à la mise en œuvre et à l'évaluation expérimentale d'un code data-parallèle (Gauss avec pivots) utilisant la migration de processus légers [139].

Le choix de l'environnement multithread sous-jacent est un point crucial de ce travail. En effet, l'utilisation des threads comme unité de migration impose à celui-ci d'intégrer la migration de threads. À l'heure actuelle, Athapascan (projet Apache) n'intègre pas encore cette fonctionnalité de manière adéquate pour nous, et nous utilisons donc l'environnement PM2 du LIFL (Jean-Marc Geib).

L'originalité de notre approche est de coupler des techniques d'équilibrage de charge dynamique avec des techniques d'analyse statique de programmes data-parallèles réguliers. Il s'agit d'extraire des informations en vue de paramétrer le module d'équilibrage de charge suivant le code. En particulier, les nids de boucles "presque réguliers", avec des indirections et/ou des tests par exemple, sont de bons candidats pour notre approche. Notre objectif pour cette année est la réalisation d'un prototype générant du code multithread avec équilibrage de charge guidé par des annotations du programme source. Dans un deuxième temps, nous étudierons comment ces annotations peuvent être générées plus ou moins automatiquement par le compilateur.

Nous avons récemment noué des contacts avec Philip Hatcher, univ. New Hampshire, très impliqué dans le développement du langage data-parallèle C* (*Data Parallel C Environment*, DPCE). Ceci nous conduit à appliquer notre expertise au développement d'environnements de programmation pour ce langage. La plate-forme *TransTool* offre une bonne base de départ, en adaptant à C* certains développements faits pour HPF. Une proposition de projet NSF/Inria, C*IT, est en cours de rédaction et sera prochainement soumise.

4 Actions industrielles

4.1 Projet Eureka EuroTOPS et LHPC

Le projet Eureka EuroTOPS a pour but le développement d'outils pour la parallélisation d'applications industrielles sur machines parallèles à mémoire distribuée. Ce projet européen comprend des équipes d'experts de l'industrie, du monde académique, et inclut également un constructeur de machines, Matra Cap-Systèmes ainsi que des utilisateurs finaux en provenance de l'industrie européenne. Les partenaires français sont l'ENS Lyon, l'Inria, le CNRS, Simulog, et ESI. Les machines cibles sont la plate-forme du LHPC (machine CAPITAN installée au LIP) et les réseaux de stations.

Financé pour cinq ans, ce projet vient de terminer avec succès sa troisième année avec le développement de nouveaux outils, conçus en collaboration avec les utilisateurs, pour la parallélisation et le portage d'applications sur plate-forme parallèle. Les outils sont bâtis sur les standards de communication PVM et MPI, et sont capables de supporter une plate-forme hétérogène de processeurs (voir section 3.2.2).

Quelques applications ont été portées avec succès sur la plate-forme logicielle développée au LIP, notamment

- PAM-CRASH, PAM-FLOW, etc. de ESI, pour la simulation de mécanique des fluides et des solides,
- MAXWELL/VF de SIMULOG, un solveur électromagnétique,
- ainsi que d'autres applications de Matra Cap-Systèmes.

La collaboration s'étend à tous les niveaux du logiciel d'exploitation des machines CAPITAN. Au niveau de l'OS, une surcouche a été mise au point pour atteindre des fonctionnalités de dynamique et d'usage en mode multi-utilisateur (CAPDYN). Au niveau des outils, plusieurs travaux de *ReMaP* ont été finalisés dans EUROTOPS, notamment le logiciel de monitoring PIMSY/VIST. Des recherches sont en cours pour la simulation d'architectures parallèles (PARSIM) et la parallélisation automatique (*TransTool*).

Ces travaux ont maintenant lieu dans le cadre du LHPC, laboratoire commun installé à l'ENS Lyon entre Matra Cap Systèmes, l'ENS Lyon, le CNRS et l'Inria.

Dans le cadre d'EuroTOPS, il faut enfin souligner notre collaboration avec Simulog pour l'ajout de nouvelles fonctionnalités au logiciel PARTITA. Cette collaboration est rendue possible par l'octroi d'une bourse de la région Rhône-Alpes à P.-Y. Calland.

4.2 Action FITT

Un projet régional FITT (Fond Industriel de Transfert Technologique) est mené avec la société Alex-Informatique (Annecy) pour la commercialisation du simulateur T9000+C104.

4.3 Action ITV Wind Concept

Cette collaboration (menée également en liaison avec Georges Perrot, professeur PAST au LIP) a pour but la mise au point d'un logiciel d'écoulement 3D de fluide et sa parallélisation (voir section 3.2.2).

5 Actions nationales et internationales

5.1 Actions nationales

Les chercheurs de *ReMaP* participent activement au GDR PRS. L. Bougé est directeur adjoint de PRS. P. Fraigniaud est responsable de l'action RUMEUR. F. Desprez participe à l'action CAPA, B. Tourancheau à l'action EXEC, et L. Bougé, A. Darté et Y. Robert, à l'action Paradigme.

5.2 Actions internationales

Nous dressons la liste de nos collaborations pour lesquelles des accords formels ou des contrats ont été établis.

5.2.1 Europe de l'ouest

Nous avons poursuivi notre collaboration avec Thomas Brandes (GMD/SCAI) autour du projet *Trans-Tool* et des optimisations des communications dans les compilateurs HPF.

Dans le cadre du projet Esprit Européen HCM (*Human Capital and Mobility*), le projet "MAP" permet la collaboration entre le projet *ReMaP* et les chercheurs de Paderborn, Barcelone, Rome, Nice et Southampton autour du développement d'outils d'aide aux communications et plongements. Le responsable est B. Monien (Paderborn).

Le projet PICASSO crée une collaboration bilatérale entre le projet *ReMaP* et l'université Polytechnique de Valencia (Espagne) avec J. Duato autour des communications en mode wormhole.

Le projet Procope France/Allemagne est mené avec l'université de Passau (avec C. Lengauer). La deuxième année se termine. Le thème est la compilation des langages data-parallèles.

5.2.2 Afrique

Nous menons un projet CEE-KIT avec l'université d'Oujda (Maroc) et l'université de Mons (Belgique), ainsi qu'un projet CNRS-CNRMaroc avec l'université d'Oujda (Maroc). Ces projets ont pour but la parallélisation de noyaux de base de l'algèbre linéaire (décomposition de matrices, recherche de valeurs propres...)

5.2.3 Amérique

Nous menons une collaboration CNRS-PICS avec l'équipe de Jack Dongarra à l'université du Tennessee de Knoxville (UTK). Ce projet a pour but la parallélisation de noyaux de base de l'algèbre linéaire issus de la bibliothèque LAPACK (bibliothèque ScaLAPACK, communication) et des travaux autour de PVM. Nous menons également une collaboration NSF-Inria avec UTK autour des environnements de programmation parallèle. La coopération des chercheurs de *ReMaP* avec UTK se poursuit depuis plusieurs années : séjours de chercheurs du LIP à UTK et réciproquement, organisation d'une conférence annuelle dans le cadre d'un contrat CNRS-NSF depuis 3 ans. Cette coopération est l'un des axes moteurs du projet et revêt une importance capitale, car elle nous permet d'apporter notre pierre aux logiciels qui sont des "standards-de-fait" pour la communauté. La seule autre alternative serait de développer en interne des outils qui auraient bien du mal à sortir de notre cercle restreint et à trouver des utilisateurs.

Un projet NSF/Inria est en cours de dépôt avec P. Hatcher de l'université du New-Hampshire (Durham), autour d'un environnement de programmation pour C*.

Enfin, B. Tourancheau et F. Desprez participent au projet PARALIN entre le Chili, l'Espagne et la France pour le développement d'applications parallèles dans l'industrie minière.

5.3 Organisation de conférences

Euro-Par'96, conférence internationale qui a regroupé plus de 350 participants à l'ENS Lyon du 26 au 29 août 1996, a été organisée par L. Bougé, P. Fraigniaud, et Y. Robert (en collaboration avec Anne Mignotte, du LIP). Les actes ont été publiés par Springer Verlag. Tous les permanents de ReMaP ont été "local chair" d'un des workshops d'Euro-Par'96.

La conférence internationale IEEE *Application Specific Array Processors* s'est tenue à Strasbourg en juillet 1995. Y. Robert a été co-responsable du comité de programme avec Patrice Quinton, et co-éditeur des actes publiés par IEEE *Computer Science Press* en 1995.

A. Darte, en collaboration avec Guy-René Perrin (Strasbourg), a organisé l'école de printemps PRS 1996 sur le parallélisme de données (actes publiés par Springer Verlag).

B. Tourancheau a été co-organisateur de Euro-PVM'95, à Lyon (actes publiés par Hermès). Il a également été co-organisateur du workshop *Environment and Tools for Parallel Scientific Computing*, 21–23 août 1996, Faverges de la Tour (actes publiés par SIAM).

5.4 Comités de programme

Les chercheurs de ReMaP ont été membres des comités de programme suivants :

- L. Bougé, RenPar'8, Bordeaux (1996), et STACS'96 ;
- A. Darte, CCS'95, Brest ;
- F. Desprez, RenPar'8, Bordeaux (1996), et EuroPVM'96, Munich (1996) ;
- P. Fraigniaud, IWIN'95, SIROCCO'96, et PODC'96 (*ACM Symp. on the Principles of Distributed Computing*) ;
- Y. Robert, *High Performance Computing Symposium'95*, Montreal ;
IEEE Conference on Application Specific Array Processors, Chicago (1996) ;
High Performance Computing Symposium'96, Ottawa ;
ICSA Parallel and Distributed Computing Systems PDCS'96, Dijon et
Irregular'96, Santa Barbara ;
- B. Tourancheau, EuroPVM'96, Munich, et *European Parallel Tools Meeting*, ONERA, Paris (1996).

5.5 Comité de rédaction et numéros spéciaux

P. Fraigniaud est membre du comité de rédaction de TSI.

Y. Robert fait partie du comité de rédaction de "Integration, the VLSI Journal" (North Holland) dans la section "Algorithms and Architectures", et de "Int. Journal Supercomputer Applications" (MIT Press). Il est l'éditeur européen de "Parallel Processing Letters" (World Scientific Publishing).

B. Tourancheau est membre du comité de rédaction de *Calculateurs Parallèles* (Hermès).

F. Desprez, L. Prylli et B. Tourancheau sont éditeurs d'un numéro spécial de la revue *Calculateurs Parallèles* sur PVM.

Santosh Pande, J. Ramanujam et Y. Robert sont éditeurs d'un numéro spécial de la revue PPL sur les techniques avancées de compilation-parallélisation.

L. Bougé, P. Fraigniaud, Anne Mignotte, et Y. Robert sont éditeurs d'un numéro spécial de la revue TCS qui rassemble les meilleures contributions, révisées et étendues, d'Euro-Par'96.

6 Diffusion des résultats

6.1 Actions d'enseignement

B. Tourancheau et L. Prylli ont effectué des enseignements sur la technologie ATM (DESS réseau de l'UCB Lyon, magistère de l'ENS Lyon). A. Darté, F. Desprez et P. Fraigniaud proposent chacun un cours de recherche au DEA d'informatique de Lyon, dont L. Bougé est responsable. Bien sûr, les enseignants chercheurs du projet participent au magistère de l'ENS Lyon, UCB Lyon et UJF Grenoble.

6.2 Conférences invitées

Les membres de *ReMaP* ont donné plusieurs conférences invitées :

- L. Bougé, centre Charles Hermite, CRIN/Inria Lorraine, 1996,
- A. Darté, ETFA'95, Paris, SPDP'96 Workshop on compilers, New Orleans, Dagstuhl seminar on automatic parallelization, 1996, et IMA Minesota, 1996,
- P. Fraigniaud, SIROCCO'96, Sienna, et IWBG'96 (*International Workshop on Broadcasting and Gossiping*, Eugene Oregon),
- Y. Robert, "Journée calcul parallèle", Clermont-Ferrand, 1995, SPDP'96 Workshop on compilers, New-Orleans, et *Dagstuhl seminar on automatic parallelization*, 1996.

6.3 Actions de formation

F. Desprez et Y. Robert ont organisé trois journées de formation sur les outils du parallélisme à l'ENS Lyon. Ces trois journées étaient destinées à tous les ingénieurs et chercheurs souhaitant acquérir une première expérience de programmation des machines massivement parallèles à mémoire distribuée. L'accent a été mis sur les outils standards du domaine public : PVM, MPI, ScaLAPACK et HPF. Cours théoriques et travaux pratiques ont été dispensés aux 15 personnes présentes, par les membres du projet.

F. Desprez a mis en place sur Lyon le groupe de travail ParaMAP³ dont le but est de faire se rencontrer les experts du parallélisme et les chercheurs d'autres disciplines ayant besoin de puissance de calcul. Durant cette année, huit exposés ont été présentés autour de thèmes aussi divers que le placement d'antennes radios, les systèmes d'informations géographiques ou l'imagerie médicale.

P. Fraigniaud est intervenu dans le cadre du Greco Programmation : "Communication dans les réseaux d'interconnexion", et dans le cadre de la formation continue de Supélec.

6.4 Animations scientifiques

Nous avons participé aux journées "Science en fête" à l'ENS Lyon, en octobre 1996 (exposé de L. Bougé : "le parallélisme, pour quoi faire?").

6.5 Diffusion de logiciels

ReMaP a mis dans le domaine public la version 2 du logiciel `_PDGEMR()` pour la redistribution de données dans la librairie SCALAPACK.

ReMaP a déposé le logiciel CAPDYN, surcouche d'exploitation de machines parallèles.

ReMaP va mettre en accès sous FTP anonyme le simulateur T9000+C104. Celui-ci est utilisé par l'université de Clermont-Ferrand, pour la recherche.

³Parallélisme Massif et Applications

7 Publications

Livres et monographies

- [71] L. BOUGÉ, P. FRAIGNIAUD, A. MIGNOTTE, Y. ROBERT (réd.), *EuroPar'96, Lecture Notes in Computer Science, 1123-1124*, Springer Verlag, 1996.
- [72] P. CAPPELLO, C. MONGENET, G. PERRIN, P. QUINTON, Y. ROBERT (réd.), *Application-specific array processors ASAP'95*, IEEE Computer Science Press, 1995.
- [73] J. DE RUMEUR (réd.), *Communication dans les réseaux de processeurs*, Masson, Paris, 1994.
- [74] J. DONGARRA, M. GENGLER, B. TOURANCHEAU, X. VIGOUROUX (réd.), *EuroPVM*, Hermès, Lyon, France, septembre 1995.
- [75] J. DONGARRA, B. TOURANCHEAU (réd.), *Environments and Tools for Parallel Scientific Computing III*, SIAM Press, 1996.
- [76] M. GENGLER, S. UBÉDA, F. DESPREZ, *Initiation au Parallélisme : Concepts, Architectures et Algorithmes, Manuels informatiques*, Masson, 1995, 2-225-85014-3.
- [77] G.-R. PERRIN, A. DARTE (réd.), *The Data Parallel Programming Model, Lecture Notes in Computer Science, 1132*, Springer Verlag, 1996.

Articles et chapitres de livre

- [78] J.-C. BERMOND, P. FRAIGNIAUD, J. PETERS, «Antepenultimate broadcasting», *Networks* 26, 1995, p. 125–137.
- [79] J.-C. BERMOND, P. FRAIGNIAUD, «Broadcasting and Gossiping in de Bruijn Networks», *SIAM Journal on Computing* 23, 1, 1994, p. 212–225.
- [80] V. BOUCHITTÉ, P. BOULET, A. DARTE, Y. ROBERT, «Evaluating array expressions on massively parallel machines with communication/computation overlap», *Int. J. Supercomputer Applications and High Performance Computing* 9, 3, 1995, p. 205–219.
- [81] A. DARTE, F. DESPREZ, J.-C. MIGNOT, Y. ROBERT, «TransTOOL: A restructuring tool for the parallelization of applications using High Performance Fortran», *Journal of the Brazilian Computer Society*, to appear.
- [82] A. DARTE, M. DION, Y. ROBERT, «A characterization of one-to-one modular mappings», *Parallel Processing Letters* 5, 1996, p. 145–157.
- [83] A. DARTE, Y. ROBERT, «Affine-by-statement scheduling of uniform and affine loop nests over parametric domains», *J. Parallel and Distributed Computing* 29, 1995, p. 43–59.
- [84] A. DARTE, F. VIVIEN, «Revisiting the decomposition of Karp, Miller, and Winograd», *Parallel Processing Letters* 5, 4, 1995, p. 551–562.
- [85] A. DARTE, F. VIVIEN, «On the optimality of Allen and Kennedy's algorithm for parallelism detection in nested loops», *Journal of Parallel Algorithms and Applications*, 1996 (to appear, Special issue on Optimizing Compilers for Parallel Languages).
- [86] F. DESPREZ, J. DONGARRA, B. TOURANCHEAU, «Performance Study of LU Factorization with Low Communication Overhead on Multiprocessors», *Parallel Processing Letters* 5-II, 1995.
- [87] F. DESPREZ, M. GARBEY, «Numerical Simulation of a Combustion Problem on a Paragon Machine», *Parallel Computing* 21, 1995, p. 495–508.
- [88] F. DESPREZ, B. TOURANCHEAU, «Basic routines for the Rank-2k update: 2D torus vs reconfigurable network», *Parallel Computing* 21, 1995, p. 353–372.

- [89] F. DESPREZ, S. UBÉDA, «Le Parallélisme dans l'Industrie : Rêve ou Réalité ?», *Technique et Science Informatique* 15, 5, 1996, p. 643–647.
- [90] M. DION, C. RANDRIAMARO, Y. ROBERT, «Compiling affine nested loops: how to optimize the residual communications after the alignment phase?», *J. Parallel and Distributed Computing*, to appear.
- [91] M. DION, T. RISSET, Y. ROBERT, «Resource-constrained scheduling of partitioned algorithms on processor arrays», *Integration, the VLSI Journal* 20, 1996, p. 139–159.
- [92] M. DION, Y. ROBERT, «Mapping affine loop nests», *Parallel Computing*, to appear.
- [93] P. FRAIGNIAUD, C. GAVOILLE, «Memory Requirement for Universal Routing Schemes», *Journal of Distributed Computing*, (to appear).
- [94] P. FRAIGNIAUD, C. LAFOREST, «Minimum gossip bus networks», *Networks* 27, 1996, p. 239–251.
- [95] P. FRAIGNIAUD, E. LAZARD, «Methods and Problems of Communication in Usual Networks», *Discrete Applied Mathematics* 53, 1994, p. 79–133.
- [96] P. FRAIGNIAUD, A. LIESTMAN, D. SOTTEAU, «Open problems», *Parallel Processing Letters* 3, 4, 1994, p. 507–524.

Communications à des congrès, colloques, etc.

- [97] V. BOUCHITTÉ, P. BOULET, A. DARTE, Y. ROBERT, «Heuristics for the evaluation of array expressions on state-of-the-art massively parallel machines», in : *Algorithms and Parallel VLSI Architectures III*, M. Moonen, F. Catthoor (réd.), North Holland, p. 319–330, 1995.
- [98] P. BOULET, T. BRANDES, «Evaluation of Automatic Parallelization Strategies for HPF Compilers», in : *High-Performance Computing and Networking*, H. Liddell, A. Colbrook, B. Hertzberger, P. Sloot (réd.), *Lecture Notes in Computer Science, 1067*, Springer Verlag, p. 778–783, Brussels, Belgium, avril 1996.
- [99] P. BOULET, «Bouclettes: a Fortran Loop Parallelizer», in : *High-Performance Computing and Networking*, H. Liddell, A. Colbrook, B. Hertzberger, P. Sloot (réd.), *Lecture Notes in Computer Science, 1067*, Springer Verlag, p. 784–791, Brussels, Belgium, avril 1996.
- [100] T. BRANDES, S. CHAUMETTE, M.-C. COUNILH, A. DARTE, J. MIGNOT, F. DESPREZ, J. ROMAN, «HPFIT: A Set of Integrated Tools for the Parallelization of Applications Using High Performance Fortran: Part I: HPFIT and the Transtool Environment», in : *Third Workshop on Environments and Tools for Parallel Scientific Computing*, J. Dongarra, B. Tourancheau (réd.), SIAM, Faverges, August 1996.
- [101] T. BRANDES, S. CHAUMETTE, M.-C. COUNILH, A. DARTE, J. MIGNOT, F. DESPREZ, J. ROMAN, «HPFIT: A Set of Integrated Tools for the Parallelization of Applications Using High Performance Fortran: Part II: Data Structures Visualization and HPF Support for Irregular Data Structures with Hierarchical Scheme», in : *Third Workshop on Environments and Tools for Parallel Scientific Computing*, J. Dongarra, B. Tourancheau (réd.), SIAM, Faverges, August 1996.
- [102] T. BRANDES, F. DESPREZ, «Implementing Pipelined Computation and Communication in an HPF Compiler», in : *Europar'96 Parallel Processing, Lecture Notes in Computer Science, 1123*, Springer Verlag, p. 459–462, August 1996.
- [103] P.-Y. CALLAND, A. DARTE, Y. ROBERT, F. VIVIEN, «On the removal of anti and output dependences», in : *Application Specific Systems, Architectures and Processors*, J. Fortes, C. Mongenet, K. Parhi, V. Taylor (réd.), IEEE Computer Society Press, p. 353–364, 1996.
- [104] P.-Y. CALLAND, A. DARTE, Y. ROBERT, F. VIVIEN, «Plugging anti and output dependence removal techniques into loop parallelization algorithms», in : *Environments and Tools for Parallel Scientific Computing III*, J. Dongarra, B. Tourancheau (réd.), SIAM Press, to appear.
- [105] P.-Y. CALLAND, A. DARTE, Y. ROBERT, «A new guaranteed heuristic for the software pipelining problem», in : *Tenth ACM International Conference on Supercomputing*, ACM Computer Science Press, p. 261–269, 1996.

- [106] J. COHEN, P. FRAIGNIAUD, J. KONIG, A. RASPAUD, «Complexité de la diffusion en mode commutation de circuits», in: *RenPar'8*, Univ. Bordeaux 1, 1996.
- [107] A. DARTE, F. VIVIEN, «A classification of nested loops parallelization algorithms», in: *INRIA-IEEE Symposium on Emerging Technologies and Factory Automation*, IEEE Computer Society Press, p. 217–224, 1995.
- [108] A. DARTE, F. VIVIEN, «Revisiting the decomposition of Karp, Miller and Winograd», in: *Application Specific Array Processors ASAP 95*, IEEE Computer Society Press, p. 13–25, 1995.
- [109] A. DARTE, F. VIVIEN, «On the optimality of Allen and Kennedy's algorithm for parallelism detection in nested loops», in: *Europar'96 Parallel Processing, Lecture Notes in Computer Science, 1123*, Springer Verlag, 1996.
- [110] A. DARTE, F. VIVIEN, «Optimal fine and medium grain parallelism detection in polyhedral reduced dependence graphs», in: *PACT'96*, IEEE Computer Society Press, 1996 (to appear).
- [111] V. DEMIAN, F. DESPREZ, H. PAUGAM-MOISY, M. POURZANDI, «Parallel Implementation of RBF Neural Networks», in: *Europar'96 Parallel Processing, Lecture Notes in Computer Science, 1124*, Springer Verlag, p. 243–250, August 1996.
- [112] F. DESPREZ, P. RAMET, J. ROMAN, «Optimal Grain Size Computation for Pipelined Algorithms», in: *Europar'96 Parallel Processing, Lecture Notes in Computer Science, 1123*, Springer Verlag, p. 165–172, August 1996.
- [113] M. DION, C. RANDRIAMARO, Y. ROBERT, «How to optimize residual communications ?», in: *10th International Parallel Processing Symposium*, IEEE Computer Society Press, p. 382–391, 1996.
- [114] M. DION, T. RISSET, Y. ROBERT, «Resource-constrained scheduling of partitioned algorithms on processor arrays», in: *EuroMicro Workshop on Parallel and Distributed Processing*, IEEE Computer Society Press, p. 571–580, 1995.
- [115] M. DION, Y. ROBERT, «Mapping affine loop nests: new results», in: *HPCN Europe 1995*, B. Hertzberger, G. Serazzi (éd.), *LNCS 919*, Springer Verlag, p. 184–189, 1995.
- [116] S. DOMAS, F. DESPREZ, B. TOURANCHEAU, «Optimization of the ScaLAPACK LU Factorization Routine Using Communication/Computation Overlap», in: *Europar'96 Parallel Processing, Lecture Notes in Computer Science, 1124*, Springer Verlag, p. 3–10, August 1996.
- [117] J. DONGARRA, C. RANDRIAMARO, L. PRYLLI, B. TOURANCHEAU, «Array Redistribution in ScaLAPACK using PVM», in: *EuroPVM users' group*, Hermès, 1995.
- [118] E. FLEURY, P. FRAIGNIAUD, «Strategies for multicasting in meshes», in: *23rd International Conference on Parallel Processing (ICPP'94)*, 1994.
- [119] E. FLEURY, P. FRAIGNIAUD, «Analysis of Deadlock-Free Path-Based Wormhole Multicasting in Meshes in Case of Contention», in: *6th Symposium on the Frontiers of Massively Parallel Computing (Frontiers'96)*, 1996.
- [120] E. FLEURY, M. GRAMMATIKAKIS, M. KRAETZEL, «Continuous routing on high speed switches STC104 vs. telegraphos switch», in: *PARCELLA 96 - Seventh International Workshop on Parallel Processing by Cellular Automata and Arrays*, Akademie Verlag Berlin, 1996.
- [121] E. FLEURY, M. PICQUENDAR, «Massively parallel machine based on T9000 and C104», in: *Transputers'94*, L. L. M. Becker, M. Tréhel (éd.), IOS Press, p. 120–134, Besançon, France, septembre 1994.
- [122] P. FRAIGNIAUD, C. GAVOILLE, «A Characterisation of Networks supporting Linear Interval Routing», in: *13th Annual ACM Symposium on Principles of Distributed Computing (PODC'94)*, A. PRESS (éd.), p. 216–224, août 1994.
- [123] P. FRAIGNIAUD, C. GAVOILLE, «Optimal Interval Routing», in: *Parallel Processing: CONPAR '94 - VAPP VI*, B. Buchberger, J. Volkert (éd.), *854 of Lecture Notes in Computer Science*, Springer-Verlag, p. 785–796, septembre 1994.

- [124] P. FRAIGNIAUD, C. GAVOILLE, «Memory Requirement for Universal Routing Schemes», in : 14th Annual ACM Symposium on Principles of Distributed Computing (PODC'95), A. PRESS (réd.), p. 223–230, août 1995.
- [125] P. FRAIGNIAUD, C. GAVOILLE, «Local Memory Requirement of Universal Routing Schemes», in : 8th Annual ACM Symposium on Parallel Algorithms and Architecture (SPAA), ACM PRESS (réd.), p. 183–188, juin 1996.
- [126] P. FRAIGNIAUD, C. LAFOREST, «Disjoint spanning trees of small depth», in : *Parallel Computing: trends and application (ParCo'94)*, F. P. G. Joubert, D. Trystram, D. Evans (réd.), Elsevier Science, p. 105–112, 1994.
- [127] P. FRAIGNIAUD, J. PETERS, «Structured Communications in Torus Networks», in : *28th Hawaii International Conference on System Sciences*, IEEE, p. 584–593, 1995.
- [128] P. FRAIGNIAUD, S. VIAL, «Approximation Algorithms for Information Dissemination Problems», in : *IEEE Second International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP-96)*, I. S. Section (réd.), p. 155–162, juin 1996.
- [129] L. PRYLLI, B. TOURANCHEAU, «Distributed simulation of parallel computers», in : *PTOOLS conference*, NASA Ames research center, Palo-Alto, USA, mai 1995.
- [130] L. PRYLLI, B. TOURANCHEAU, «Calcul parallèle sur réseau ATM de stations de travail», in : *RENPAR'8*, Roman (réd.), Université de bordeaux, 1996.
- [131] L. PRYLLI, B. TOURANCHEAU, «Distributed simulation of parallel computers», in : *29th Annual Simulation Symposium*, IEEE computer society press, New Orleans, USA, avril 1996.

Rapports de recherche et publications internes

- [132] P. BOULET, M. DION, «Code generation in Bouclettes», *Research Report n°95-43*, Laboratoire de l'Informatique du Parallélisme, Nov 1995, To appear in proceedings 5th Euromicro Workshop on Parallel and Distributed Processing, 22–24 Jan 1997.
- [133] T. BRANDES, S. CHAUMETTE, M.-C. COUNILH, A. DARTE, J. MIGNOT, F. DESPREZ, J. ROMAN, «HPFIT: A Set of Integrated Tools for the Parallelization of Applications Using High Performance Fortran», *rapport de recherche n°96-28*, LIP ENS Lyon, 1996.
- [134] V. DEMIAN, F. DESPREZ, H. PAUGAM-MOISY, M. POURZANDI, «Parallel Implementation of RBF Neural Networks», *Rapport de Recherche n°96-11*, LIP ENS Lyon, 1996.
- [135] F. DESPREZ, S. DOMAS, B. TOURANCHEAU, «Optimization of the ScaLAPACK LU Factorization Routine Using Communication/Computation Overlap», *Rapport de Recherche n°96-17*, LIP - ENS Lyon, 1996.
- [136] F. DESPREZ, S. DOMAS, B. TOURANCHEAU, «Optimization of the ScaLAPACK LU factorization routine using communication/computation overlap», *rapport de recherche*, LIP-ENS-Lyon, 1996.
- [137] J. DONGARRA, L. PRYLLI, C. RANDRIAMARO, B. TOURANCHEAU, «Array Redistribution in ScaLAPACK using PVM», *rapport de recherche n°CS-95-310*, University of Tennessee, octobre 1995.
- [138] E. FLEURY, P. FRAIGNIAUD, «Deadlocks in adaptive wormhole routing», *rapport de recherche n°94-09*, LIP-CNRS, ENS-Lyon, France, 1994.
- [139] C. PEREZ, «Load balancing HPF programs by migrating virtual processors», *rapport de recherche n°RR96-??*, LIP, ENS Lyon, octobre 1996, To appear.
- [140] C. PEREZ, «Utilisation de processus légers pour l'exécution de programmes à parallélisme de données : étude expérimentale.», *rapport de recherche n°RR96-09*, LIP, ENS Lyon, avril 1996.
- [141] L. PRYLLI, B. TOURANCHEAU, «Efficient Block Cyclic Data Redistribution», *rapport de recherche n°2766*, INRIA Rhone-Alpes, 1995.

- [142] L. PRYLLI, B. TOURANCHEAU, «Fast Array Redistribution on MIMD computers», *rapport de recherche n°95-39*, LIP-ENS Lyon, 69364 Lyon, France, 1995.
- [143] L. PRYLLI, B. TOURANCHEAU, «Distributed simulation of parallel computers», *rapport de recherche n°2767*, INRIA Rhone-Alpes, 1996.
- [144] L. PRYLLI, B. TOURANCHEAU, «Parallel computing on ATM network», *rapport de recherche*, LIP-ENS Lyon, 1996, in progress.

8 Abstract

The aim of the *ReMaP* project is to bring new contributions and software tools for massively parallel computing applications. Target machines are distributed memory parallel computers and networks of workstations. Target applications are those where regularity is a key characteristic, such as dense linear algebra or volumic image processing.

Research topics are the following:

- Massively parallel algorithms (design, analysis, implementation),
- Environment tools (computing libraries, communication libraries, parallel HPF-like compilers),
- Programming methodologies (specification languages, data-parallelism),
- Automatic parallelization techniques.

