ARC ResCapA : Équipe lyonnaise

Rapport d'activité 1999*

15 février 2000

1 Personnes engagées dans l'action

Permanents : Luc Bougé, Raymond Namyst, Jean-François Méhaut (tous les trois dans le projet ReMaP, LIP, ENS Lyon) ; Laurent Lefèvre (maintenant dans le pré-projet RESO, Laboratoire RESAM, Univ. Claude Bernard Lyon 1).

Thésards : Christian Perez (thèse soutenue en décembre 1999), Gabriel Antoniu (2e année en 1999–2000), Olivier Aumage (1re année en 1999–2000).

Stagiaires de DEA : Gabriel Antoniu (1998) ; Alice Bonhomme (1998, Laboratoire RESAM), Olivier Aumage (1999), Mickaël Gauthier (1999) ; Vincent Danjean (2000), Alexandre Denis (2000), Nathalie Viollet (2000).

Autres Stagiaires : Guillaume Mercier, Nicolas Boullis et Georges Da Costa (Magistère MIM, 1999), Encolpe Degoute (Maîtrise d'informatique 1999, en stage dans le laboratoire RESAM).

2 Présentation synthétique de notre contribution

Notre compétence. Notre groupe *ParaDigme* s'intéresse au sein du projet INRIA ReMaP aux environnements d'exécution hautes performances pour les grappes de PC : bibliothèques de programmation multithread (PM2), bibliothèques de communication portables (Madeleine), bibliothèques de mémoire virtuellement partagée (Dosmos, DSM-PM2), protocoles zéro-copies (BIP/Myrinet).

Nos engagements. Dans le cadre de l'action ResCapA, nous avions proposé d'appliquer l'ensemble de notre compétence à l'étude de la technologie d'interconnexion SCI. Notre objectif était le suivant.

- Mettre en place une plate-forme SCI au LIP, ENS Lyon, en nous appuyant fortement sur l'expérience de nos partenaires grenoblois (Sirac) et rennais (Caps/Paris).
- Porter sur cette plate-forme notre environnent PM2, notamment la bibliothèque de communication Madeleine, permettant ainsi l'exécution de tous les logiciels construits au-dessus de PM2 : bibliothèques de mémoire virtuellement partagée DOSMOS et DSM-PM2, compilateur HPF/PM2, etc.

*Contact : Luc Bougé, LIP, ENS Lyon, 46 Allée d'Italie, 69364 Lyon Cedex 07. Luc.Bouge@ens-lyon.fr

– Mettre à la disposition de nos partenaires de ces couches logicielles (notamment Madeleine) pour qu'ils puissent y adosser leurs propres développements sans avoir à "affronter" les détails bas-niveau de SCI.
– Comparer notre approche de SCI orientée vers la *portabilité* en mode *utilisateur* avec des approches directes développées chez nos partenaires grenoblois et rennais.

Nos résultats. Nous avons atteint pleinement ces objectifs.

Nous avons installé au LIP une grappe de 8 PC bi-processeurs PentiumPro 450 MHz interconnectés par un réseau SISCI/SCI. L'aide de nos partenaires a été décisive pour la réussite de cette opération. Leurs expériences (parfois douloureuses...) nous ont permis d'éviter plusieurs écueils difficiles et surtout de bénéficier d'une assistance technique efficace et précieuse.

PM2 est aujourd'hui complètement opérationnel sur cette grappe, et toutes les applications de l'équipe ont été immédiatement installées sans aucune difficulté. Citons en particulier le *compilateur HPF Adaptor* version 6 (Thomas Brandes, GMD) dans le cadre de la thèse de Christian Perez, et une version d'un environnement d'exécution *Java distribué Hyperion* (Phil Hatcher, University New Hampshire) dans la cadre de celle de Gabriel Antoniu. Nous pensons être à ce jour la seule équipe au monde à disposer d'un compilateur HPF et d'un *exécuteur* Java au-dessus de SCI. Des études ont été faites pour le portage de l'environnement de partage virtuelle de mémoire Dosmos.

Le portage de PM2/Madeleine sur SCI a été réalisé à partir de la version initiale de Madeleine, conçue essentiellement pour BIP/Myrinet. Des performances honorables ont été obtenues, mais sa structure très orientée vers les communications par messages ne nous a pas permis d'utiliser *pleinement* les caractéristiques originales de SCI.

Cette expérience nous a conduits à repenser complètement l'architecture de la bibliothèque de communication Madeleine, de manière à pouvoir tirer parti de la capacité d'adressage de SCI et des transferts DMA entre la carte d'interface et la mémoire de l'hôte. Cette version est appelée *Madeleine II*. Elle sera prochainement diffusée. Ceci fait l'objet de la thèse de Olivier Aumage. Cette réflexion a été conduite en commun avec Benoît Planquelle, équipe Espace, Lille, qui travaille aussi sur les bibliothèques de communication multi-protocoles pour les grappes de grappes de PC.

L'ensemble de ces réalisations a été mis à la disposition de nos partenaires. Le portage d'Athapascan au-dessus de Madeleine a été étudié par Jacques Briat, projet Apache. Il rendrait Athapascan immédiatement utilisable sur une grappe de PC SCI. Yvon Jégou, projet Caps/Paris, a porté sa bibliothèque de partage virtuel de mémoire MOME sur Madeleine avec de bons résultats. La bibliothèque portable DSM-PM2 est développée en liaison avec David Mentré, Caps/Paris, qui travaille sur un sujet similaire avec une approche directe. Les travaux sur DSM-PM2 et Madeleine II ont été influencés par le travail de Paul Koch et Emmanuel Cecchet sur SciOS. Un portage de PM2 sur SciOS a été envisagé.

Utilisation du financement ResCapA. Le financement ResCapA a été utilisé pour des missions. Il a aussi permis l'indemnisation forfaitaire des stages de DEA de Gabriel Antoniu (LIP, 1998), Olivier Aumage (LIP, 1999), Mickaël Gauthier (LIP, 1999). Deux de ces trois stagiaires sont maintenant en thèse dans l'équipe sur des allocations MENRT.

Notre avis sur SCI. Nous n'avons pas connu de problème important à l'installation du matériel SCI (un commutateur et huit cartes d'interface) fourni par Dolphin. Seule une carte d'interface était défaillante à son installation, elle nous a été rapidement échangée par Dolphin. Depuis sa mise en service il y a un an, le matériel s'est révélé être d'une grande fiabilité et même tolérant aux pannes de la climatisation de la salle machine !

Depuis le mois de février 1999, deux versions des drivers Dolphin (IRM et SISCI) ont été fournies par Dolphin. L'essentiel de nos travaux et développements a été réalisé sur l'interface SISCI. Nous présentons plus en détail les réalisations dans la section suivante. La documentation électronique accompagnant les drivers s'est révélée suffisante pour leurs installations. Nous tenons en particulier à souligner que la distribution du driver SISCI est accompagnée d'une suite assez complète d'exemples qui testaient et démontraient les différentes fonctionnalités SISCI. Ces programmes nous ont aussi permis de tester le bon état de marche du matériel. Les fonctionnalités présentes dans SISCI sont multiples et variées (load/store, DMA, interruptions) et ouvrent de nouvelles perspectives pour les couches de communication.

On retrouve également cette grande variété au niveau de VIA et de ses différents modes de communication (Send/receive, RDMA, completion queues). En ce qui concerne VIA, nous regrettons que Dolphin n'ait pas poursuivi son effort de distribution de cette interface. Les discussions avec les responsables Dolphin étaient quelque peu frustrantes et pouvaient se résumer à ce qui suit : "VIA, on sait faire, c'est prêt dans nos cartons, mais on attend une demande forte du marché". Nous pensons que la demande s'alimente aussi par des offres un peu avant-gardistes !

Les performances mesurées ont été conformes à ce que nous nous attendions. La technologie de réseau SCI/Dolphin était annoncée avec une latence très faible (quelques microsecondes) et une bande passante limitée à environ 80 Mo/s. Cette tendance s'est confirmée sur notre plateforme, à la fois sur les programmes d'évaluation fournis avec le driver SISCI, mais aussi sur les différentes versions de PM2 et Madeleine qui ont été développées. On peut par exemple noter que la latence migration d'un petit thread PM2 a été mesurée à 25 μs, contre 56 μs sur réseau Myrinet. Nous reviendrons plus précisément sur ces résultats dans les sections suivantes.

Le driver SISCI donne la possibilité aux programmeurs d'accéder directement au processeur DMA de la carte. Ce mécanisme est intéressant car il constitue un bon support pour le recouvrement des communications. Les performances du mécanisme DMA de SISCI sont aujourd'hui assez décevantes (de l'ordre de 30 Mo/s). Nous pensons que ces performances sont principalement dues au driver, en particulier parce que le programmeur doit pré-enregistrer ses données dans un espace mémoire réservé aux opérations DMA. Il est raisonnable de penser que des progrès significatifs seront effectués avec les prochaines versions de SISCI.

Une des approches préconisées par l'environnement est de tenter minimiser les interactions avec le système d'exploitation. En ce qui concerne les communications, nous avons privilégié les études sur des bibliothèques comme BIP qui permettent de communiquer efficacement sur des réseaux Myrinet en contournant le système d'exploitation. Au niveau des threads et de la bibliothèque Marcel, les mécanismes sont implémentés en espace utilisateur et permettent d'afficher des performances de tout premier plan. Un des intérêts de la technologie SCI est de permettre la communication entre machines grâce à une simple opération de lecture ou d'écriture en mémoire, donc sans interaction avec le système. L'interface de communication Madeleine I a été portée sur SISCI et les bons résultats s'expliquent donc aussi par les interactions réduites avec le système.

A Présentation détaillée des résultats

A.1 Installation de la grappe SCI

La configuration SCI installée au LIP est constituée d'une grappe de 8 PC. Chaque PC est un bi-Pentium II cadencé à 450 MHz, équipé d'un chipset 440 BX, de 128 Mo de mémoire et d'une carte SCI Dolphin D310/PCI 32 bits. Le réseau SCI s'organise autour d'un commutateur Dolphin D512 à 4 ports. À chaque port est connecté un anneau (bi-directionnel) de deux PC.

Chaque machine est administrée sous Linux 2.2.10 SMP. Les pilotes utilisés pour la gestion des cartes SCI sont les suivants.

Dolphin IRM version 1.9.5 (alpha 3). Ce pilote est nécessaire au fonctionnement des cartes. Il fournit une interface de programmation très basique orientée "*ioctl + mmap*" pour l'établissement des projections mémoires entre les machines.

Dolphin SISCI version 1.9.2. Ce pilote s'appuie sur l'IRM en fournissant une API de plus haut niveau beaucoup plus agréable à utiliser. Entre autre, il existe une interface très complète pour l'utilisation des transferts DMA et le déclenchement d'interruptions à distance.

Note : Les noyaux Linux ont étés configurés pour réserver une zone statique de 8 Mo de mémoire physique (patch *Bigphysarea*).

Globalement, nous n'avons rencontré que quelques difficultés mineures au cours de l'installation de cette configuration et durant son exploitation. Cependant, l'immaturité des pilotes fournis par Dolphin a beaucoup ralenti les activités de développement en ce qui concerne les couches supérieures. La distribution "standard" de ces pilotes (disponibles sur le site Web de Dolphin) était d'ailleurs systématiquement inutilisable sur notre système : nous avons dû demander à Dolphin un accès privilégié aux sources des pilotes de manière à pouvoir les recompiler correctement sur notre système. D'autre part, l'absence de documentation des pilotes de Dolphin se révèle gênante lors de l'exploration de certaines fonctionnalités avancées de SCI.

Quelques performances. Les performances brutes obtenues en utilisant l'interface du pilote SISCI sont tout à fait conformes à celles annoncées par Dolphin. Nous avons ainsi mesuré une latence minimale de 2.5 μs (écriture d'un entier par le processeur dans une page distante) et une bande passante d'environ 72 Mo/s (toujours en mode "*store*") pour des transferts de 4 Mo. Il est à noter que les performances d'un transfert d'un bloc de données contiguës sont très sensibles à la taille du bloc et à son alignement mémoire. En particulier, il est beaucoup plus efficace de transférer 64 octets que d'en transférer 1 ou 63 ! Cette caractéristique est intimement liée au mode de tamponnage des cartes PCI-SCI ainsi qu'à l'utilisation du tampon de "*write combining*" du Pentium. En tout état de cause, une primitive est heureusement fournie dans l'interface SISCI pour effectuer les transferts de données le plus efficacement possible en tenant compte de ces paramètres matériels.

Tous ces tests et mises au point ont été en grande partie effectués par Guillaume Mercier et Nicolas Boullis au cours de leur projet de première année du Magistère d'informatique et modélisation (MIM).

Évaluation. Malgré une (courte) période de mise au point délicate des différents pilotes de Dolphin, on peut dire que le bilan de notre grappe SCI est très positif. Sauf dans le cas d'applications erronées faisant mauvais usage des pilotes, nous n'avons expérimenté aucun "plantage" intempestif de la grappe depuis sa mise en exploitation (depuis plus d'un an). D'autre part, les performances sont très bonnes en ce qui concerne les petits/moyens messages : la latence minimale est très faible et la demi-bande passante est atteinte très rapidement.

Il demeure cependant certains points moins positifs. Le premier est que beaucoup de fonctionnalités prévues dans l'interface du pilote SISCI ne sont pas encore implantées et aucune date de disponibilité n'est d'ailleurs annoncée. C'est particulièrement gênant car cela bride les performances de certains schémas de communication où les zones de transfert de données ne sont pas connues à l'avance (impossibilité de "projeter" des zones mémoires quelconques). A notre sens, il n'y a pourtant aucun obstacle majeur à la réalisation des fonctionnalités manquantes.

Le second point est certainement le plus important. Les différentes expériences que nous avons menées ont montré que l'obtention de bonnes performances est conditionnée par une utilisation extrêmement fine des différentes possibilités d'optimisation des cartes PCI-SCI et des caches du processeur. A l'opposé, ces expériences montrent qu'une utilisation naïve des capacité d'adressage des cartes mène à des performances catastrophiques. Aussi, il nous semble illusoire d'espérer obtenir des performances correctes en utilisant SCI pour mettre en place une *mémoire virtuelle partagée* de manière transparente entre les différents nœuds d'une grappe...

A.2 Portage de PM2/Madeleine sur SISCI/SCI

Les travaux de recherche menés au LIP concernant le réseau SCI s'articulent autour de l'environnement multithread PM2. L'objectif est double. Il s'agit d'une part d'étudier l'implantation efficace d'un modèle de communication orienté RPC sur un réseau à capacité d'adressage ainsi que d'essayer de "dégager" certaines fonctionnalités propres à ce type de réseau pour les faire remonter au niveau d'une API de haut niveau telle que celle de PM2 (par exemple pour fournir une DSM à ce niveau). D'autre part, il s'agit d'étudier finement l'intégration des communications avec un modèle d'exécution multithread. Dans ce cas, aux problèmes de choix de la méthode de transfert (*load/store* vs *DMA*) s'ajoutent ceux liés à la sélection de la méthode de détection des messages entrants (*scrutation* vs *interruptions*).

Ces travaux ont donc naturellement débuté par une première tentative de portage de l'environnement PM2 au-dessus d'un réseau SCI en utilisant l'interface du pilote SISCI de Dolphin. C'est précisément la bibliothèque Madeleine, qui constitue la couche de communication de PM2, qui a été portée sur SISCI. Ce travail a fait suite à la pré-étude menée par Guillaume Mercier et Nicolas Boullis (afin de déterminer si SISCI fournissait une interface adaptée) et a été réalisé par Georges Da Costa pendant son stage de première année de Magistère. Ce travail a été un succès en aboutissant à un portage complètement opérationnel de la bibliothèque Madeleine sur SISCI.

Les performances brutes de Madeleine sont très bonnes. La latence est de 6 μs et la bande passante de 70 Mo/s. Les performances de PM2 bénéficient d'ailleurs directement de ces caractéristiques : la migration d'un *thread* via SCI a ainsi été mesurée à 25 μs (record mondial) !

Cependant, à cause de l'incomplétude actuelle des fonctionnalités du pilote SISCI, il ne nous a pas été possible d'éviter une recopie (en réception) pour chaque paquet de données transmis. Les bonnes performances obtenues en bande passante (par rapport aux performances brutes) sont dues à l'utilisation d'un mode de transfert en "pipeline" maximisant le recouvrement des différentes étapes. Mais il est clair que les performances de Madeleine pourraient être meilleures si les pilotes implantaient toutes les fonctionnalités prévues.

En outre, ces travaux ont mis en lumière une orientation trop "*message passing*" de l'interface de portabilité de Madeleine. En quelque sorte, les performances résultantes sont bridées par une interface trop restrictive inadaptée aux réseaux de type *put/get*. A partir de cette constatation, nous avons entrepris la conception de *Madeleine II* qui offre une structure bien plus générique (cf Section A.3).

A.3 Applications développées

Compilateur HPF Adaptor/PM2. Le compilateur Adaptor, développé par T. Brandes (GMD), supporte en très grande partie le langage High Performance Fortran 2. Nous avons intégré dans Adaptor une fonctionnalité d'équilibrage de charge découplé du code utilisateur. Celle-ci est basée sur la migration de processus légers fournie en particulier par PM2. Le portage d'Adaptor au dessus de PM2 a été validé sur le réseau Myrinet en utilisant les protocoles TCP et BIP. Pour exécuter un programme HPF compilé avec notre version d'Adaptor en utilisant le réseau SCI, il n'a fallu que compiler PM2 pour SCI. Aucune modification n'a été faite dans Adaptor. Madeleine a parfaitement rempli son objectif d'assurer la portabilité. Des mesures préliminaires montrent des performances très proches entre l'utilisation de Madeleine au-dessus de SCI et au dessus de Myrinet. En conclusion, nous avons été en mesure d'utiliser SCI pour exécuter des programmes HPF dès que Madeleine a été opérationnelle pour ce réseau.

Bibliothèque de gestion mémoire virtuellement partagée : DSM-PM2. DSM-PM2 est une extension de PM2 permettant de supporter l'illusion d'une mémoire partagée lors d'une exécution en environnement distribué. DSM-PM2 supporte plusieurs modèles de cohérence et propose de multiples protocoles pour leur implémentation, qui utilisent des mécanismes alternatifs : réplication des pages, migration des threads, migration des threads, etc. Pour cela, DSM-PM2 utilise efficacement les composants de PM2 (bibliothèque de threads migrants Marcel, bibliothèque de communication Madeleine) tout en profitant de la portablilité de ces composants. Grâce à Madeleine, DSM-PM2 est disponible sur des grappes reliées par SISCI/SCI, BIP/Myrinet, TCP/Myrinet. Dans le tableau ci-dessous nous présentons en exemple les temps de réplication d'une page (en μs) sur les trois architectures :

Operation/Protocoles	SISCI/SCI	BIP/Myrinet	TCP/Myrinet
Preparing a page request	1	2	2
Transmitting the request	17	30	190
Processing the request	1	2	2
Sending back a 4 kB page	85	134	412
Installing the page	12	24	24
Total	**116**	**192**	**630**

La première colonne correspond à des mesures effectuées sur une grappe de de noeuds Pentium II (450 MHz) sous Linux 2.2.10 reliés par SISCI/SCI. Les chiffres des colonnes suivantes ont été obtenus sur une grappe de Pentium Pro (200MHz) sous Linux 2.2.13 utilisant BIP, respectivement TCP. On remarquera que les meilleurs résultats sont obtenus avec SISCI/SCI.

Exécutif Java distribué : Hyperion/PM2. L'API de DSM-PM2 est conçue d'une part pour une utilisation directe et d'autre part vise à être utilisée par des compilateurs de langages de haut niveau comme Java et OpenMP. Afin de tester la généricité de notre approche nous avons interconnecté DSM-PM2 avec le système de compilation Java distribué *Hyperion*, développé par Phil Hatcher à l'Université du New Hampshire. Nous présentons ci-dessous quelques résultats préliminaires obtenus lors de l'exécution d'une application Java de coloriage de graphe.

Nombre de noeuds	PPro 200MHz TCP			PPro 200MHz BIP			PII 450MHz SCI		
	T	jS	P	T	jS	P	T	jS	P
1	2.8	2.9	1.0	2.8	2.9	1.0	1.2	2.9	1.0
2	1.8	4.4	1.6	1.9	4.2	1.5	0.9	3.9	1.3
4	1.4	5.7	2.0	1.6	5.0	1.8	0.7	5.0	1.7
6	1.2	6.6	2.3	1.4	5.7	2.0	0.5	7.0	2.4
8	1.1	7.2	2.5	1.1	7.2	2.5	—	—	—

Pour chaque plate-forme on présente les paramètres suivants. T : temps d'exécution en secondes ; jS : *Java speedup* calculé par rapport au temps d'exécution sur l'interprète Linux Java (8.0 s sur le PPro 200 MHz et 3.5 s sur le PII 450 MHz) ; P : *parallélisabilité*, calculée par rapport au temps d'exécution du programme parallèle sur un noeud. De nouveau, les meilleurs résultats sont obtenus sur la plate-forme SCI. Il faut cependant tenir compte de la fréquence d'horloge double dans ce cas.

Dosmos/PM2. Les travaux du Laboratoire RESAM sur les réseaux à capacité d'adressage concernent l'étude préliminaire menée pour le portage du système de Mémoire Distribuée Virtuellement Partagée DOSMOS sur un réseau à capacité d'adressage SCI.

L'environnement de programmation parallèle DOSMOS (Distributed Shared Memory System) est fondé sur un système de mémoire distribuée virtuellement partagée (MDVP). Il permet à des tâches de partager des objets de manière structurée et hiérarchique (notion de groupes de processus). Fondé sur un découpage structurel de la programmation parallèle, l'environnement DOSMOS s'adapte parfaitement à certaines classes d'applications (applications distribuées irrégulières, applications coopératives, applications modulaires ou fondées sur une redondance logicielle, etc.). Prenant racine sur des bibliothèques de communications standards (PVM, MPI) il permet une adaptation des applications à une vaste gamme d'architectures matérielles. Ce système a ainsi été adapté et évalué sur des architectures parallèles ainsi que des grappes de PC reposant sur des réseaux haut-débit (Myrinet).

L'intérêt d'une architecture matérielle fournissant un support pour de la mémoire partagée entre noeuds distants, par l'intermédiaire d'un réseau à capacité d'adressage, semble donc particulièrement adapté pour des systèmes de MDVP. La contribution de RESAM a donc porté sur l'évaluation de la plate-forme SCI du LIP et l'adaptation du système DOSMOS pour un portage sur un réseau à capacité d'adressage.

Évaluation des performances de communications. Les réseaux SCI permettent de disposer de faibles latences de communications et sont fondés sur l'accès transparent à des espaces mémoires distants. Ils reposent ainsi sur une bonne gestion des espaces mémoires partagés. Notre évaluation a ainsi montré que le non-respect d'alignements mémoire, pour l'échange de données, peut dégrader les performances de communications de 50%.

Lancement de tâches. L'intérêt d'une mémoire partagée matérielle a pris tout son aspect pour le portage de DOSMOS avec notamment un placement des tâches et un partage de la mémoire optimisées en fonction des besoins des applications.

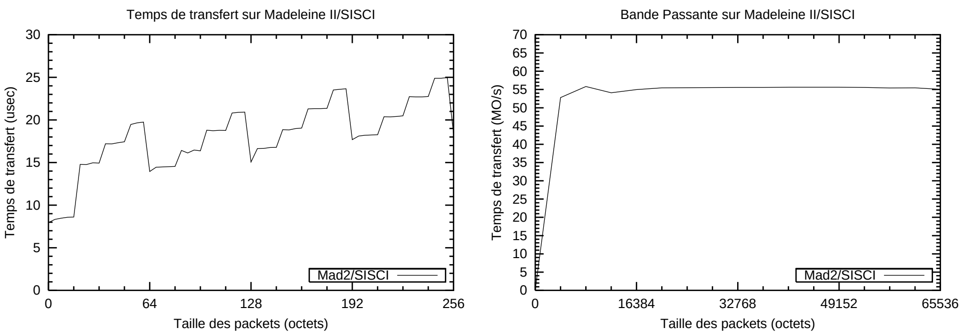

FIG. 1: Temps de transfert et bande passante sur Madeleine II/SISCI

Étude de l'extensibilité d'architecture. De plus, l'étude menée sur l'utilisation de commutateurs dans un réseau SCI montre que le coût de transfert des données dans ces commutateurs n'est pas négligeable pour les communications entre noeuds distants. L'extensibilité de ce type de plate-forme est donc difficilement maîtrisable si l'on veut fournir de bonnes performances pour les communications.

La difficulté pour garantir des performances honorables et constantes pour les opérations de communications les plus simples a été l'un des facteurs les plus limitatifs pour le portage du système de Mémoire Distribuée Virtuellement Partagée DOSMOS. Les expérimentations et évaluations effectuées sur la plate-forme du LIP démontrent que le principe de réseau Scalable Coherent Interface semble particulièrement intéressant comme alternative aux modèles de programmation à base d'échange de messages difficiles pour les programmeurs d'applications distribuées. Mais les produits disponibles actuellement (en termes de protocoles de communications, et d'API) ne permettent que d'extraire difficilement les performances et fonctionnalités que l'on pourrait attendre de ce type de matériel.

Bibliothèque de communication multi-protocoles : Madeleine II. Cette expérience nous a conduits à repenser complètement l'architecture de la bibliothèque de communication Madeleine, de manière à pouvoir tirer parti de la capacité d'adressage de SCI et des transferts DMA entre la carte d'interface et la mémoire de l'hôte. La nouvelle version de Madeleine, *Madeleine II*, intègre de manière générique cette nouvelle approche. Il est en fait possible de faire collaborer au sein d'un même programme différents protocoles (SCI, Myrinet, TCP/IP, etc.), et même différentes stratégies d'utilisation d'un même protocole (par exemple en SCI, utiliser ou pas le DMA de la carte SCI), le choix entre les stratégies étant fait *dynamiquement* au niveau de chaque bloc de données en fonction de la taille, des exigences d'efficacité, etc. Cette nouvelle structure a également été portée très rapidement sur le protocole VIA proposé par Compaq, Intel et MicroSoft.

Les mesures présentées sur la Figure 1 indiquent des performances honorables en temps de transfert de petits messages. Les bandes passantes maximales obtenues sont en revanche quelque peu décevantes, à environ 55 Mo/s, mais elles restent toutefois bien meilleures que celles mesurées lors des transfert par DMA (35 Mo/s). Cette même Figure 1 met en outre en évidence la sensibilité de SCI à la forme des données transférées : les temps de transferts de petits messages varient considérablement suivant l'alignement ou non des données sur des frontières puissance de 2, principalement en raison du fonctionnement du mécanisme de *write combining* des processeurs de type Pentium mis en œuvre par SCI. Un mécanisme parfois délicat à maîtriser...

Lancer de rayon parallèle (POV-RAY). Dans le cadre d'un projet TTN (*Technology Transfert Node*) entre le LIP et une entreprise lyonnaise spécialisée dans le rendu informatique réaliste de mobilier, nous avons réalisé une version parallèle et multithread du moteur de lancer de rayon POV-RAY. L'implantation a été effectuée directement en appui sur PM2. Son originalité réside dans sa capacité à équilibrer dynamiquement la charge en migrant les différents threads de calcul de manière préemptive et *transparente* pour l'application.

Cette application a été validée sur les différentes grappes de PC bi-processeurs du LIP et en particulier sur notre grappe SCI. Une démonstration a d'ailleurs été effectuée lors d'une réunion ResCapA à l'ENS Lyon.

Publications de l'équipe dans le cadre de l'ARC

[1] Gabriel Antoniu, Luc Bougé, Philip Hatcher, Mark MacBeth, Keith McGuigan, and Raymond Namyst. Compiling multithreaded Java bytecode for distributed execution, January 2000. Submitted for publication.

[2] Gabriel Antoniu, Luc Bougé, Philip Hatcher, Mark MacBeth, Keith McGuigan, and Raymond Namyst. Implementing Java consistency using a generic, multithreaded DSM runtime system. In *Parallel and Distributed Processing. Proc. Intl Workshop on Java for Parallel and Distributed Computing.*, volume ? ? of *Lect. Notes in Comp. Science*, pages ? ?– ? ?, Cancun, Mexico, May 2000. Held in conjunction with IPDPS 2000. IEEE TCPP and ACM, Springer-Verlag. To appear.

[3] Gabriel Antoniu, Luc Bougé, and Raymond Namyst. An efficient and transparent thread migration scheme in the PM2 runtime system. In *Parallel and Distributed Processing. Proc. 3rd Workshop on Runtime Systems for Parallel Programming (RTSPP '99)*, volume 1586 of *Lect. Notes in Comp. Science*, pages 496–510, San Juan, Puerto Rico, April 1999. Held in conjunction with IPPS/SPDP 1999. IEEE TCPP and ACM SIGARCH, Springer-Verlag.

[4] Gabriel Antoniu, Luc Bougé, and Raymond Namyst. DSM-PM2 : a multi-protocol DSM layer for the PM2 multithreaded runtime system. In J. Chassin de Kergommeaux, B. Lecussan, and J.-F. Méhaut, editors, *Proc. 2nd Workshop on Parallel Computing for Irregular Applications (WPCIA2)*, Toulouse, France, January 2000. Held in conjunction with HPCA-6. No publicly distributed proceedings.

[5] Gabriel Antoniu, Luc Bougé, Raymond Namyst, and Christian Perez. Compiling data-parallel programs to a distributed runtime environment with thread isomigration. In *The 1999 Intl Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA '99), Technical Session on parallel and distributed languages : mechanisms, implementations, and tools*, volume 4, pages 1756–1762, Las Vegas, NV, June 1999. Invited paper.

[6] Gabriel Antoniu, Luc Bougé, Raymond Namyst, and Christian Perez. Compiling data-parallel programs to a distributed runtime environment with thread isomigration. In *8th Workshop on Compilers for Parallel Computers (CPC 2000)*, Aussois, France, January 2000. Updated version of [5].

[7] Gabriel Antoniu, Luc Bougé, and Christian Perez. Generic load balancing for HPF programs : Application to the Flame Simulation kernel. In *The 3rd Annual HPF User Group Meeting (HUG '99)*, Redondo Beach, California, August 1999. Electronic proceedings available at `http://www.icase.edu/hug99/`.

[8] Gabriel Antoniu and Christian Perez. Allocation iso-adresse pour une migration préemptive de processus légers : application à un compilateur HPF. In *Actes des Rencontres francophones du parallélisme (RenPar 11)*, pages 43–48, IRISA, Rennes, France, June 1999.

[9] Gabriel Antoniu and Christian Perez. Using preemptive thread migration to load-balance data-parallel applications. In *Euro-Par '99 : Parallel Processing*, volume 1685 of *Lect. Notes in Comp. Science*, pages 117–124, Toulouse, France, August 1999. Springer-Verlag.

[10] Olivier Aumage. Une librairie de communication portable et adaptative pour réseaux haut-débit. Rapport de stage de DEA, DEA d'informatique de Lyon, Univ. Claude Bernard, Lyon 1, France, June 1999.

[11] Olivier Aumage, Luc Bougé, and Raymond Namyst. A portable and adaptative multi-protocol communication library for multithreaded runtime systems. In *Parallel and Distributed Processing. Proc. 4th Workshop on Runtime Systems for Parallel Programming (RTSPP '00)*, volume ? ? of *Lect. Notes in Comp. Science*, pages ? ?– ? ?, Cancun, Mexico, May 2000. Held in conjunction with IPDPS 2000. IEEE TCPP and ACM, Springer-Verlag. To appear.

[12] Luc Bougé, Phil Hatcher, Raymond Namyst, and Christian Perez. A multithreaded runtime environment with thread migration for a HPF data-parallel compiler. In *The 1998 Intl Conf. on Parallel Architectures and Compilation Techniques (PACT '98)*, pages 418–425, Paris, France, October 1998. IFIP WG 10.3 and IEEE.

[13] Luc Bougé, Jean-François Méhaut, and Raymond Namyst. Madeleine : an efficient and portable communication interface for multithreaded environments. In *Proc. 1998 Int. Conf. Parallel Architectures and Compilation Techniques (PACT '98)*, pages 240–247, ENST, Paris, France, October 1998. IFIP WG 10.3 and IEEE.

[14] Luc Bougé, Jean-François Méhaut, and Raymond Namyst. Efficient communications in multithreaded runtime systems. In *Parallel and Distributed Processing. Proc. 3rd Workshop on Runtime Systems for Parallel Programming (RTSPP '99)*, volume 1586 of *Lect. Notes in Comp. Science*, pages 468–182, San Juan, Puerto Rico, April 1999. Held in conjunction with IPPS/SPDP 1999. IEEE TCPP and ACM SIGARCH, Springer-Verlag.

[15] Luc Bougé, Jean-François Méhaut, and Raymond Namyst. Madeleine : An efficient and portable communication interface for RPC-based multithreaded environments. Research Report RR1999-55, LIP, ENS Lyon, Lyon, France, December 1999. Also available as Research Report RR-3845, INRIA Rhône-Alpes. Submitted for publication.

[16] Luc Bougé, Jean-François Méhaut, Raymond Namyst, and Loïc Prylli. Using the VI Architecture to build distributed, multithreaded runtime systems : a case study. In *Proc. 2000 ACM Symposium on Applied Computing (SAC 2000)*, Villa Olmo, Como, Italy, March 2000. ACM Special Interest Group on Applied Computing (SIGAPP), ACM. To appear.

[17] Franck Capello, Daniel Litaize, Jean-François Méhaut, Christine Morin, Serge Petiton, and Denis Trystram. Nouveaux challenges pour la prochaine décennie : le pétaflops (1000 téraflops). *Techniques et sciences informatique (TSI)*, 1999. Article invité, numéro spécial RenPar 11. À paraître.

[18] Benjamin Gaidioz. Extension d'une interface de communication haute-performance sur des protocoles à tampons pré-alloués. Rapport de stage MIM1, Magistère d'informatique et modélisation (MIM), ENS Lyon, LIP, ENS Lyon, September 1998. Stage effectué sous la direction de Raymond Namyst, Luc Bougé et Robert Russell.

[19] Patrick Geoffray, Laurent Lefèvre, Cong-Duc Pham, Loïc Prylli, Olivier Reymann, Bernard Tourancheau, and Roland Westrelin. High-speed LANs : New environments for parallel and distributed applications. In *Euro-Par '99 : Parallel Processing*, volume 1685 of *Lect. Notes in Comp. Science*, pages 633–642, Toulouse, France, August 1999. Springer-Verlag.

[20] Jean-François Méhaut. *Supports d'exécution pour le calcul parallèle et distribué*. Habilitation à diriger les recherches, Univ. Claude Bernard, Lyon, LIP, ENS Lyon, January 2000. To appear.

[21] Jean-François Méhaut and Yves Robert. Algorithms and tools for (distributed) heterogeneous computing : A prospective report. Research Report RR1999-36, LIP, ENS Lyon, Lyon, France, August 1999.

[22] Christian Perez. *Compilation des langages à parallélisme de données : gestion de l'équilibrage de charge par un exécutif à base de processus légers*. Thèse de doctorat, ENS Lyon, France, LIP, December 1999.

[23] Benoît Planquelle and Jean-François Méhaut. Communications multi-protocoles et réseaux à haut débit. In *11es Rencontres francophones du parallélisme (RenPar 11)*, pages 73–78, IRISA, Univ. Rennes 1, Rennes, INRIA, June 1999. Voir [25] pour une présentation plus complète en anglais.

[24] Benoît Planquelle, Jean-François Méhaut, and Nathalie Revol. Multi-cluster approach with PM2. In *The 1999 Intl Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA '99), Technical Session on Cluster Computing Technologies, Environments and Applications*, volume 2, pages 779–785, Las Vegas, NV, June 1999.

[25] Benoît Planquelle, Jean-François Méhaut, and Nathalie Revol. Multi-protocol communications and high speed networks. In *Euro-Par '99 : Parallel Processing*, volume 1685 of *Lect. Notes in Comp. Science*, pages 139–143, Toulouse, France, August 1999. Springer-Verlag.

[26] Gaëtan Scotto di Apollonia, Jean-François Méhaut, and Raymond Namyst. SCI and distributed multithreading : The PM2 approach. In H. Hellwagner and A. Reinefeld, editors, *Scalable Coherent interface : Technology and applications, Proc. of SCI Europe '98*, pages 177–183, Bordeaux, France, September 1998. Cheshire Henbury Publishing. Sponsored by SINTEF and Esprit SCI Working Group (EP 22582).