De l'utilisation de ATM pour le calcul distribué

Eric Dillon

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

De l'utilisation de ATM pour le calcul distribué

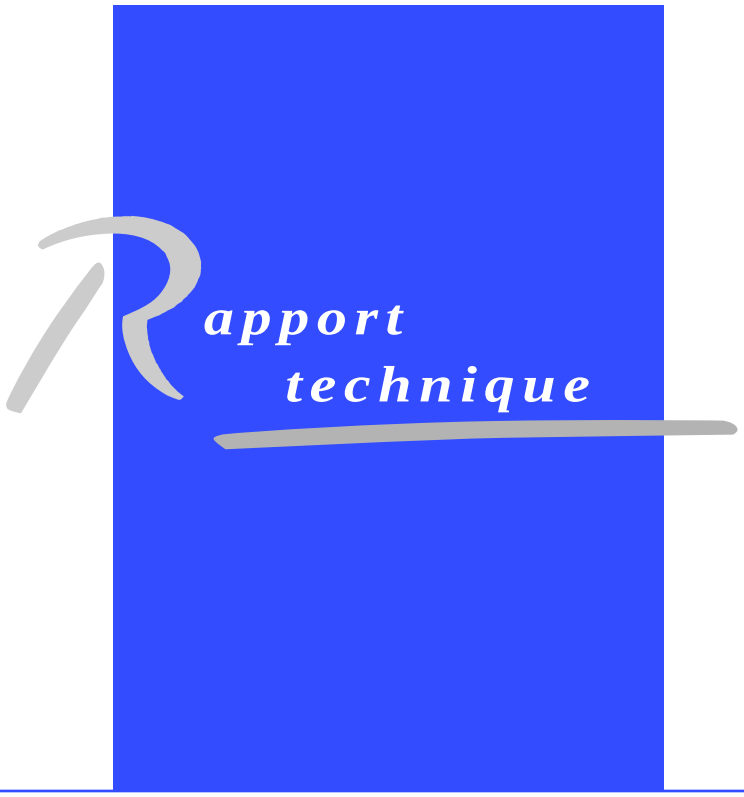
Eric Dillon

Eric.Dillon@loria.fr

N 188

Janvier 1996

PROGRAMME 1



ISSN 0249-6399

De l'utilisation de ATM pour le calcul distribué

Eric Dillon
Eric.Dillon@loria.fr

Programme 1 — Architectures parallèles, bases de données, réseaux et systèmes distribués
Projet RESEDAS

Rapport technique n° 188 — Janvier 1996 — 19 pages

Résumé : Dans le cadre des calculs distribués sur réseau de stations, le réseau Ethernet reste le moyen d'interconnexion le plus utilisé. L'apparition récente de réseaux dits à "haut-débit" comme ATM semble offrir une alternative intéressante. Ce document envisage son utilisation à travers une synthèse d'articles à propos d'études de performances concernant les communications à travers un réseau ATM. Ces articles mettent en évidence le fait qu'ATM reste sans doute prometteur dans ce cadre, mais aussi qu'il reste encore quelques problèmes à résoudre avant d'obtenir entière satisfaction. Ces résultats sont complétés par une étude de performances locale.

Mots-clé : ATM, calculs distribués, réseaux de stations, échange de messages

(Abstract: pto)

Unité de recherche INRIA Lorraine
Technôpole de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY (France)
Téléphone : (33) 83 59 30 30 – Télécopie : (33) 83 27 83 19
Antenne de Metz, technopôle de Metz 2000, 4 rue Marconi, 55070 METZ
Téléphone : (33) 87 20 35 00 – Télécopie : (33) 87 76 39 77

White Paper on the Use of ATM within Distributed Network Computing

Abstract: In the scope of Distributed Network Computing, Networks of Workstations are still often used with Ethernet as interconnection medium. Lately, new generation networks have appeared as a good alternative for distributed network computing. Among them, ATM networks seem interesting, more particulary for high performance computing. This document overviews a set of experiments carried out on ATM networks for distributed computing across the world. They point out some issues in using ATM within this area.

Key-words: ATM, Distributed Network Computing, Networks of Workstations, Message Passing

Chapitre 1

Les calculs distribués sur réseau de stations

1.1 Motivations

Les réseaux de stations (NOWs [1]) semblent offrir un potentiel de calcul intéressant face aux machines parallèles à mémoire distribuée. Malheureusement, ces "machines de calcul virtuelle" ne pouvaient jusqu'ici pas convenir à tout type d'application, essentiellement à cause des mauvaises performances des réseaux d'interconnexion utilisés. Le champ d'application était donc réduit à des applications "peu communicantes", pour ne pas dire "peu intéressantes" car la technologie des réseaux utilisés datait pratiquement des années 70.

En effet, le réseau d'interconnexion de stations le plus répandu est encore actuellement le réseau Ethernet, qui offre un débit limité dans le meilleur des cas à une dizaine de Mbits/s, et qui, en plus, oblige à un partage du médium par toutes les stations du réseau. Dans ce cas, le réseau devient vite une ressource critique et les communications inter-tâches un véritable goulot d'étranglement pour les applications distribuées. Dans le même ordre d'idée, même un réseau de plus haut débit comme FDDI (Fiber Distributed Data Interface), qui permet d'atteindre un débit proche de 100 Mbits/s peu rapidement être saturé s'il est partagé par plusieurs stations de travail.

La première approche face à la lenteur des communications a donc été de réduire au maximum les communications entre tâches, mais ce mode de programmation ne peut empêcher un certain degré de communication, si bien que, même si le canal de communication n'est pas saturé, le délai de transmission entre machines devient tout aussi pénalisant.

Depuis peu, le concept de réseau haut-débit a fait son apparition avec en tête de file le réseau *ATM*[de Prycker 91] (Asynchronous Transfert Protocol). Avec ces nouveaux réseaux sont apparus d'abord de nouveaux champs d'applications à l'image du multimédia et de la

1. Network of Workstations

vidéo à la demande sur ATM. Cependant, grâce aux débits qu'il permet d'atteindre a priori, le réseau ATM pourrait fournir un moyen efficace de communiquer entre stations de travail dédiées au calcul distribué.

Ce document évoque plusieurs expériences dans le domaine du calcul distribué avec ATM. Elles permettent de mettre en évidence l'aspect prometteur de ATM dans le domaine des calculs distribués, mais elles révèlent aussi certaines lacunes qu'il pourrait être intéressant de combler.

Le plan de ce texte s'articule de la manière suivante: les paragraphes suivants rappellent les besoins fondamentaux lorsque l'on simule une machine parallèle à mémoire distribuée sur un réseau de stations de travail. Un rapide portrait de ATM est brossé dans le paragraphe 1.3. Le chapitre 2 présente un ensemble de résultats correspondant à des expériences récentes d'utilisation d'un cluster ATM à travers plusieurs centres de calcul hautes performances. Enfin, le dernier chapitre évoquera un ensemble de solutions à explorer pour contribuer à améliorer l'effet "haut débit" de ATM sur le calcul distribué.

1.2 Le parallélisme par échange de messages

Lorsque l'on commence à parler de parallélisme, une seule chose compte: la performance. C'est en effet la raison essentielle qui justifie le temps (parfois très long) passé à parallèliser une application séquentielle.

Plus précisément, on peut remarquer qu'il existe essentiellement trois paramètres qui interfèrent avec les performances d'une application parallèle indépendamment de la "qualité" intrinsèque de design de l'application:

1. La puissance des processeurs utilisés pour réaliser les calculs.
2. La vitesse des réseaux d'interconnexion de ces processeurs.
3. Le degré d'interaction qui existe entre les deux.

Le premier paramètre est en dehors des préoccupations évoquées dans ce document, on peut néanmoins dire que la puissance intrinsèque des processeurs croît très rapidement, offrant un potentiel de calcul parallèle non négligeable. Le troisième paramètre concerne plutôt l'aspect matériel du problème et ne sera, lui non plus, pas envisagé ici. Enfin, Le second paramètre nous intéresse plus particulièrement ici. En effet, lorsque l'on modélise le temps que prend une application parallèle, il convient de distinguer deux parties:

$$T_{execution} = T_{calcul} + T_{communications} \tag{1.1}$$

Le temps de calcul est incompressible, par contre le temps de communication peut se décomposer encore en plusieurs parties.

Le modèle généralement utilisé pour caractériser un médium de communication est linéaire et comporte deux paramètres:

1. Un temps de *latence d'envoi*. Ce temps est indépendant de la taille du message envoyé. Il est défini comme le temps nécessaire pour envoyer un message vide. Il correspond

au temps de traversée des différentes couches réseau. Ce temps est incompressible et apparaît lors de chaque échange. Il sera noté T_{lat}.

2. Un coefficient correspondant au débit du médium de communication. Ce coefficient, noté T_b correspond au temps supplémentaire nécessaire pour véhiculer un octet [2].

L'ensemble des communications effectuées au sein d'une application distribuée pourra donc se noter comme:

$$T_{communications} = N_c(T_{lat} + L_c T_b) \tag{1.2}$$

où N_c sera le nombre total de communications effectuées, et L_c sera le volume moyen (en octets) échangé au travers des messages.

L'équation 1.1 permet de mesurer le poids réel que peuvent prendre les communications d'une application parallèle. En effet, elle signifie brutalement que, si l'on veut atteindre 90% des performances des processeurs, l'algorithme doit permettre d'atteindre un rapport calcul-communication tel que $T_{calcul} \geq 9T_{communications}$. Ce qui signifie en clair que le réseau de communication doit effectivement être rapide pour pouvoir atteindre ce rapport, mais que dans tous les cas il ne sera pas utilisé pendant 90% du temps ! La première conclusion à tirer est donc que le réseau d'interconnexion doit nécessairement posséder un débit élevé si l'on veut obtenir un minimum d'efficacité sans se restreindre à des applications "ennuyeuses" qui "ne communiquent pas".

Cependant, il est possible d'avoir une autre approche vis à vis des algorithmes parallèles. Une méthode "en vogue" vise à améliorer les performances d'une application distribuée en utilisant la technique du recouvrement calcul-communication, i.e. utiliser le maximum de CPU pendant que des "gros" messages sont en transit sur le réseau. Cette méthode met en évidence l'importance du temps de latence du réseau. En effet, dans le meilleur des cas, l'équation 1.1 s'écrit:

$$T_{execution} = max(T_{calcul} + N_c T_{lat}, N_c L_c T_b)$$

Il s'en suit que pour une utilisation efficace du processeur, il suffit que les temps de calculs et communications soient voisins et que $T_{calcul} \gg N_c T_{lat}$. Autrement dit, dans ce cas, pour utiliser un processeur à 90%, il faut que $\frac{T_{calcul}}{N_c} \geq 9T_{lat}$. Ici, le temps de latence du réseau devient prépondérant.

En conclusion, le débit du médium de communication n'est pas le seul paramètre essentiel dans le cadre du calcul distribué. Enfin, l'impact des capacités du réseau est d'autant plus important que l'on se place, non pas sur un processeur mono-utilisateur dédié à une tâche comme dans tout ce qui précède, mais sur une station de travail où le processeur est partagé par plusieurs tâches !

2. Le temps de propagation n'intervient pas explicitement puisque l'on se place dans le cadre de réseaux locaux.

1.3 Évolution vers des réseaux rapides

Le premier réseau utilisé pour l'interconnexion de stations de travail est Ethernet. Depuis, plusieurs alternatives sont apparues avec la naissance de réseaux dits "haut débit". Ce paragraphe brosse le portrait rapide de ces réseaux de manière à les reclasser les uns par rapport aux autres.

1.3.1 Les réseaux à "partage de médium"

Le premier réseau fut donc le célèbre réseau Ethernet. Né dans les années 70, il est encore actuellement le moyen le plus utilisé lorsqu'il s'agit de relier deux stations par un réseau local.

Son débit est assez limité puisqu'il plafonne à une dizaine de Mbits/s. De plus, il fait partie de la catégorie des réseaux que nous appellerons à *ressources partagées*, i.e. la bande passante totale est de 10 Mbits/s mais toutes les stations utilisent obligatoirement le même médium, d'où partage de ressources.

Cependant, son grand âge lui permet d'avoir certaines qualités malgré tout. En effet, la technologie Ethernet est maintenant plus que bien maîtrisée, si bien que toutes les couches, à la fois matérielles et logicielles, ont bénéficié d'un vaste choix d'optimisation.

Une première évolution est apparue avec le réseau FDDI (Fiber-Distributed Data Interface). Il se place dans la même classe de réseaux que Ethernet. Bien sûr, le débit atteint est bien plus important puisqu'il est de l'ordre de 100 Mbits/s, mais le principe de partage des ressources est le même, ce qui implique des problèmes de saturation du médium de la même manière que pour Ethernet.

Ces deux réseaux manquent donc de "scalability": la bande passante du réseau ne peut pas augmenter avec le nombre de processeurs connectés, d'où saturation rapide.

1.3.2 Les réseaux de nouvelle génération

Les réseaux précédents sont sur le point de laisser la place à des réseaux dits à haut-débit. Ces réseaux ne sont plus basés sur un partage du médium, mais sur une architecture autour d'un switch.

Ces réseaux se répandent de plus en plus dans le monde du calcul distribué, même s'ils n'y étaient pas tous dédiés au départ. Parmi les plus célèbres, on retrouve des réseaux tels que HIPPI (HIgh Performance Parallel Interface), Fiber Channel, ou encore ATM (Asynchronous Transfert Mode).

ATM est sans aucun doute le réseau haut-débit le plus célèbre actuellement. Il a été introduit dans le cadre des réseaux longue distance à intégration de services, mais semble pouvoir être aussi très prometteur pour le calcul distribué. Proposé par le CCITT, le principe de ATM est basé sur l'échange de cellules de petite taille (53 octets) à des débits élevés (multiples de 51.84 Mbits/s), le plus populaire étant le débit OC-3 (155.52 Mbits/s).

un réseau ATM est basé sur l'utilisation d'un switch. Chaque station est reliée à ce switch qui se charge d'établir les connexions. En conséquence, il n'y a pas de partage du médium

(comme dans les cas de Ethernet ou FDDI) mais simplement un partage du switch qui est en général "fait pour".

ATM est caractérisé par une structure en couche qui va de la couche physique (fibre optique ou simple paire torsadée) à une couche d'adaptation notée AAL. Il existe 5 couches d'adaptations définies par le CCITT:

1. AAL type 1: offre des services à débit constant dédié à des transmissions classiques de la voix.
2. AAL type 2: offre des services de transferts d'informations vidéo et audio à des débits variables.(Maintient une relation de temps entre la source et la destination)
3. AAL type 3: offre des services orientés connexion, et des fonctions de signalisation sur le réseau.
4. AAL type 4: version "sans connexion" de AAL3 (souvent regroupée avec AAL3)
5. AAL type 5: couche d'adaptation minimale pour un échange de PDUs.

Outre ce support pour plusieurs classes de services, ATM possède 2 propriétés spécifiques:

- Un mode orienté connexion: ATM fournit une connexion virtuelle entre deux processus qui voudraient communiquer. Toutes les cellules d'un même envoi suivent le même chemin, encore appelé chemin virtuel.
- Un débit de transfert élevé.

On retrouve ces propriétés dans d'autres réseaux haut débits. *Fiber Channel* [Lin 94] semble lui aussi être assez prometteur pour l'utilisation comme réseau d'interconnexion de stations dédiées au calcul distribué. Il offre deux techniques de commutation, le *packet switching* et le *circuit switching*, à des vitesses de transmission variées: 25.805 Moctets/s, 51,61 Moctets/s, et 103.22 Moctets/s. *HIPPI* [Hsieh 96] offre lui aussi des services équivalents avec deux débits possible de 800 ou 1600 Mbits/s sur une distance limitée à 25m.

Ces réseaux haut débits, en particulier ATM, semblent naturellement très intéressants pour réaliser l'interconnexion entre des stations de travail, et bénéficier de vitesses de transfert élevées. Le paragraphe suivant présente différent exemples d'utilisation qui mettent en évidence quelques problèmes posés par ATM.

Chapitre 2

Utilisation de réseaux haut débit: ATM

Plusieurs travaux ont été réalisés pour tester les avantages et inconvénients que pourrait amener l'utilisation de ATM dans le cadre du calcul distribué. Nous présenterons d'abord une synthèse des principaux résultats avant de présenter nos propres résultats "locaux".

2.1 Quelques pré-dispositions ?

Les réseaux haut débit possèdent indéniablement des atouts qui peuvent faire d'eux des candidats sérieux pour l'utilisation dans le calcul distribué.

En particulier, les transferts à haut débit ! Grâce à l'architecture basée sur l'utilisation d'un switch, il suffit d'augmenter le nombre de liaisons physiques pour augmenter le débit. Cette propriété de "scalability" permet d'augmenter le nombre de stations connectées sans saturer le réseau, tout en atteignant des débits proches du Gbit.

Ces réseaux disposent potentiellement d'un temps de latence faible, puisqu'ils utilisent des connections dédiées, basées sur l'utilisation d'un switch. Malheureusement, comme le montrent les résultats suivants, les réseaux traditionnels comme Ethernet fournissent encore un temps de latence réellement plus rapide. La première explication peut être donnée par le fait que la technologie utilisée dans le cadre des réseaux haut débit devient de plus en plus mature, mais les interfaces et en particuliers les softwares mis en jeux nécessitent encore quelques étapes de mise au point et d'optimisation.

En conséquence, la question de l'utilisabilité de ATM au sein du calcul distribué ne doit pas se poser en terme de technologie mais plutôt en terme de d'interface utilisée au dessus d'ATM, de protocole, et donc de logiciel.

2.2 Synthèse des principaux résultats

Cette section s'appuie essentiellement sur les travaux réalisés dans 3 centres de recherches différents:

- à l'université du Minnesota par l'équipe de David H.C. Du [Chang 95][Lin 95].
- à Oak Ridge Nat. Lab. par Al Geist, l'un des pères de PVM. [Zhou 95]
- au NASA Lewis Research Center, à Cleveland, Ohio. [Dowd 95]

D'autres travaux existent concernant l'utilisation de ATM pour les calculs distribués, notamment à l'université du Michigan [Huang 94], ou encore à l'université de Californie à Berkeley [Anderson 95]. Ces travaux sont basés sur des considérations qui ne seront pas envisagées ici.

2.2.1 Etude de performances dans différents domaines

Les résultats peuvent être envisagés suivant trois points de vue:

- celui du débit atteignable sur une liaison ATM avec des outils d'échange de messages divers.
- celui du temps de latence d'une liaison ATM avec ces mêmes outils.
- celui du gain réel dans le cas d'algorithmes de calculs complets (plus seulement des résultats bruts sur le plan des communications) exécutés sur un cluster ATM.

Beaucoup de chiffres existent dans la littérature sur les débits, les temps de latences et autres, qu'il est possible d'atteindre avec ATM. Mais ils dépendent tous de l'architecture réelle du réseau, des machines reliées elles-même à un switch donné, et même de l'API utilisée.

Tous ces chiffres ne peuvent être équivalents car lorsque l'on mesure de telles performances, beaucoup de paramètres entrent en ligne de compte (les machines, les bus, les cartes d'interface, le système, etc.). Tous les résultats seront donc à prendre de manière relative, en tant que *comparaisons* lorsqu'un seul paramètre n'aura varié. [Lin 95] présente d'ailleurs des résultats permettant de quantifier l'impact de ces différents paramètres sur les performances brutes:

- Le poids de l'interface: le passage d'une interface Fore serie-200 à une interface serie-100 provoque une chute du débit maximum de 70,08 Mbits/s à 11,52 Mbits/s sur le même Sun Sparc 2.
- Le poids de la machine: de manière évidente une machine plus rapide permet d'obtenir de meilleures performances, même si quelques exceptions restent inexpliquées.

– Le poids du switch: le temps de propagation du signal à travers le switch ASX-100 et les interfaces serie-200 permet d'obtenir des valeurs variant de 9 μsec à 11 μsec d'interface à interface.

Conclusion, puisque tous les résultats qui suivent n'ont pas tous été obtenus sur le même site, il est impossible de comparer ces résultats entre eux.

ATM et le débit

Le premier point d'étude de ATM peut se faire à travers les avantages qu'il offre au niveau du débit brut des informations échangées.

Les premiers résultats présentés ont été obtenus en utilisant une émulation des sockets BSD sur trois architectures réseau différentes: Ethernet, ATM et FDDI. [Lin 95] présente une comparaison sur les débits bruts atteignables sur ces différents réseaux, ils sont regroupés dans la table 2.1.

Réseau	Débit max (Mbits/s)	$n_{1/2}$ Octet
ATM	16,7	3204,
FDDI	17,2	6818
Ethernet	8,4	6482

TAB. 2.1 – *Débits bruts ATM, Ethernet et FDDI*

Même si ces valeurs sont relativement faibles (mesures effectuées avec de "vieux" Sun), elles permettent de montrer que les débits de ATM et FDDI sont équivalents, au détail prêt que ATM peut garantir une qualité de service, alors que FDDI peut rapidement être saturé. Le second paramètre, note $n_{1/2}$ correspond à la taille des messages à envoyer pour atteindre un débit égal à la moitié du débit maximum. ATM est donc réellement plus efficace que les deux autres puisqu'il atteint sont débit limite plus rapidement.

Pour compléter ces résultats qui attestent du "plus" apporté par ATM sur le plan du débit, [Zhou 95] compare l'utilisation de PVM sur un réseau ATM et sur Ethernet. La table 2.2 présente les résultats obtenus dans ce cas. Dans cette étude, ils ont essentiellement ré-écrit PVM en l'interfaçant avec les différentes APIs disponibles sur ATM. Les résultats sont ici aussi très encourageants: l'utilisation complète de ATM (le cas AAL5) permet de doubler le débit limite de PVM.

Enfin, il faut noter que [Chang 95] obtient des résultats comparables sur un site différent (Cf table 2.3).

En conclusion, sur le plan débit d'échange des informations, ATM est réellement plus efficace, et constitue donc un avantage dans le cadre du calcul distribué.

ATM et le temps de Latence

Cependant, sur le plan du temps de latence, ATM n'est plus aussi efficace.

API utilisée	Réseau	Débit Max (Mbits/s)
PVM-AAL5	ATM	**13,3**
PVM-AAL4	ATM	8,88
PVM-TCP	ATM	8,42
PVM-TCP	Ethernet	**6,60**

TAB. 2.2 – *Débits avec PVM sur ATM et Ethernet*

PVM utilisé	Débit Max (Mbits/s)	$n_{1/2}$ (Octets)
PVM-ATM avec AAL5	**27.202**	7867
PVM-ATM avec AAL4	26.627	8239
PVM/TCP/ATM	20.826	7649
PVM/TCP/Ethernet	**8.312**	1945

TAB. 2.3 – *Débits avec PVM sur ATM et Ethernet*

Si on reprend la même expérience comparant Ethernet, FDDI et ATM, les résultats sont complètement inversés (Cf table 2.4). Dans ce cas, le temps de latence d'envoi double lorsque l'on passe de Ethernet à ATM.

Réseau	Latence (μs)
ATM	**1960**
FDDI	1833
Ethernet	**1053**

TAB. 2.4 – *Latences brutes de ATM, Ethernet et FDDI*

Bien sûr ces résultats sont confirmés avec l'utilisation de PVM. La table 2.5 présente les résultats de [Chang 95]. Ils ont été obtenus sur une architecture visiblement plus efficace, mais malgré cela, l'utilisation de ATM fait augmenter le temps de latence.

PVM utilisé	Temps de Latence (μs)
PVM-ATM avec AAL5	**1905**
PVM-ATM avec AAL4	1903
PVM/TCP/ATM	1839
PVM/TCP/Ethernet	**1541**

TAB. 2.5 – *Latences avec PVM sur ATM et Ethernet*

Donc, contrairement à ce qui se passe sur le plan du débit, le passage à un réseau ATM fait encore considérablement augmenter le temps de latence d'envoi des messages, ce qui peut se révéler très pénalisant pour certaines classes d'application de calculs distribués.

Un cas réel

Les deux paragraphes précédents s'articulent tous les deux autour des performances brutes des fonctions de communication. Malheureusement, ce genre de résultats ne permettent pas toujours d'extrapoler correctement sur des résultats d'applications réelles. [Lin 95] présente un dernier ensemble de résultats en comparant l'exécution de deux algorithmes "populaires" dans le domaine des calculs distribués. Il s'agit d'un algorithme de produit parallèle de matrices et d'une résolution parallèle d'équations différentielles. Les résultats présentés ne permettent cependant pas de tirer de réelles conclusions car le nombre de stations est trop limité pour mettre en évidence les atouts de ATM, et les algorithmes sont optimisés pour ne pas "trop" communiquer. Ce qui tend à privilégier Ethernet.

2.3 Résultats locaux

Enfin, cette section présente les résultats que nous avons obtenus sur le site local du LORIA.

L'environnement de tests était constitué de 2 stations HP735 (125 Mhz et 64 Mo de RAM) reliées grâce à un switch de Fore Systems fournissant donc une liaison à 155Mbits/s, et grâce à une liaison Ethernet.

Les tests ont été conduits de manière classique en mesurant les temps d'aller-retour sur le réseau au moyen d'un programme d'ÉCHO.

Afin de préciser les mesures, les tests ont été effectués à la fois sur les couches "basses" de communication, i.e. les couches d'adaptation ATM et la couche TCP, et sur des couches plus "hautes" comme PVM [PVM Team 94] et une implantation de MPI [MPI Forum 95]. La version 3.3.9 de PVM a été utilisée pour les tests, ainsi que la version 1.0.10 de MPICH dans le cas de MPI.

2.3.1 Au niveau réseau

Les premiers résultats concernent donc les performances mesurées sur les couches réseaux disponibles à la fois au-dessus de ATM et de Ethernet.

Les résultats concernant les temps de latence (Cf table 2.6) confirment les résultats obtenus sur d'autres sites. Il est malgré tout intéressant de remarquer que les valeurs "brutes" sont les plus basses de ce rapport. Ceci tend à prouver que les performances, sur le plan des communications, des stations HP utilisées ici sont nettement supérieures à celles des stations SUN utilisées par ailleurs. Cependant, malgré ce "nivellement" des valeurs, les temps de latence obtenus sur ATM restent décevant par rapport à celui obtenu par la combinaison TCP/Ethernet. En particulier, l'émulation de TCP sur AAL4/5 se révèle plus lente que

sa version originale sur Ethernet. De plus, en utilisant directement l'API fournie par Fore Systems, le temps de latence n'est que de 210μsec, ce qui est moins que TCP/Ethernet, mais la différence n'est pas suffisante compte tenu que l'API Fore ne fournit pas un service fiable comme TCP, et que, par conséquent, un traitement supplémentaire doit être effectué pour gérer les erreurs de transmission et le contrôle de flux.

API utilisée	Latence (μsec)
AAL4&5/ATM	210
TCP/AAL4&5	280
TCP/Ethernet	248

TAB. 2.6 – *Latences brutes*

En ce qui concerne les débits maximums[1], ATM confirme son efficacité par rapport à Ethernet, mais les mesures révèlent que seule une faible partie de la bande passante de ATM peut être exploitée. Une étude plus approfondie devra permettre de dire si cette limite est fixée par le matériel ou par le logiciel. La table 2.7 résume ces résultats. TCP exploite en effet près de 95% de la bande passante de Ethernet, valeur qui n'est visiblement pas encore envisageable avec ATM !

API utilisée	Débit Max (Mbits/s)
TCP/AAL4&5	60
TCP/Ethernet	9.50

TAB. 2.7 – *Débits bruts*

2.3.2 Au niveau Message Passing

Les derniers résultats concernent l'utilisation de bibliothèques de communication au dessus des deux réseaux ATM et Ethernet.

De manière évidente, l'introduction de ces "sur-couches" font chuter les résultats. Concernant les temps de latence, les bibliothèques se retrouvent au même niveau indépendamment du réseau. Ce qui est, encore une fois décevant pour ATM. La table 2.8 résume ces résultats[2].

Enfin, au chapitre des débits maximums, pas de surprises, ATM confirme une fois de plus son avantage. (Cf table 2.9).

1. Il n'a été possible de mesurer un débit maximum à partir de l'API Fore car elle est limitée à l'utilisation de message de faible taille. Par conséquent, les mesures de débits ne pouvaient être fiables.

2. Ici encore, MPICH/AAL4/5 n'a pu être mesuré car la conception de MPICH ne permet pas de forcer l'utilisation d'une interface donnée. Dans notre configuration de mesures, seule l'interface ATM a pu être utilisée.

Bibliothèque utilisée	Latence (μ sec)
PVM/TCP/ATM	487
PVM/TCP/Ethernet	490
MPI(mpich)/TCP/ATM	473

TAB. 2.8 – *Latences, Message Passing*

Bibliothèque utilisée	Débits max (Mbits/s)
PVM/TCP/Ethernet	7.5
PVM/TCP/ATM	46
MPI(mpich)/TCP/ATM	43

TAB. 2.9 – *Débits max, Message Passing*

2.4 Conclusion

Il faut donc tirer deux conclusions quant à l'utilisation de ATM pour le calcul distribué:

- ATM permet d'augmenter le débit lors des échanges.
- ATM augmente aussi le temps de latence d'envoi.

En conséquence, l'utilisation de ATM dans le cadre des calculs distribués doit n'être encore réservé à des applications précises, des applications qui font beaucoup d'envois de gros messages qui pourront bénéficier de la bande passante de ATM. Par contre, il serait illusoire de penser, sous prétexte que ATM est un réseau rapide, qu'il devient possible d'utiliser un cluster ATM comme une machine parallèle, sans se préoccuper des problèmes de communication.

Néanmoins, l'utilisation de ATM reste sans aucun doute prometteuse, à condition que l'on apporte quelques solutions au problème posé par le temps de latence.

Chapitre 3

Conclusions

3.1 Pistes à explorer

Il est possible d'explorer plusieurs pistes pour remédier au problème de la latence.

Première solution, changer quelque peu le mode d'échange des messages. C'est notamment ce qui a été fait à Berkeley avec la généralisation aux clusters ATM de la méthode de communication par messages actifs [Martin 95][von Eicken 94]. Cette méthode permet de réduire le temps de latence mais conduit aussi à un changement dans le mode de programmation.

Seconde solution, considérer que l'API livrée actuellement par Fore est essentiellement orienté "débit". En effet, ATM est avant tout dédié à des envois massifs de son ou d'image, où ce qui compte est essentiellement le débit, et la bande passante du réseau. Cette approche soulève un problème plus général. En effet, l'API de Fore ne correspond pas actuellement à une norme. Des travaux sont sans nul doute en cours au sein de l'ATM Forum, mais rien n'existe encore à l'heure actuelle: d'où problème.

Il semble donc envisageable de définir une API différente de celle de Fore. Ce travail passe par plusieurs étapes:

1. Acquérir une connaissance de base sérieuse sur les principes d'ATM.
2. Évaluer et analyser les performances précises d'un système comme celui qui se trouve ici.
3. Proposer des approches différentes pour améliorer le temps de latence et le débit puisque par rapport aux débits et latences annoncés dans les normes ATM, le potentiel existe réellement.

Une telle démarche a été suivie dans le cadre d'une étude sur le réseau Fiber Channel [Lin 94]. Après différentes modélisation des performances de chaque "couche" réseau, des

solutions ont été trouvées conduisant à une amélioration de l'ordre de 75% pour le débit et de 16% pour le temps de latence.

Plusieurs défauts sont déjà reconnus à l'API Fore actuelle: les principaux étant que la MTU[1] de cette API est de 4 Ko, ce qui oblige à des opérations segmentation/Re-assemblage programmées directement par l'utilisateur... et donc non optimisées et que la transmission n'est pas fiable. Il s'agit donc aussi de trouver un meilleur compromis entre interface de base performante mais difficile à utiliser, et interface d'un peu plus haut niveau mais peu performante.

3.2 Une justification économique

Enfin, le projet "NOW" de l'université de Berkeley a fourni une dernière justification à l'utilisation de ATM comme médium d'interconnexion entre stations de travail. Ils ont mené une étude mettant en rapport les performances et le coût de différentes configurations.

Ils ont évalué les performances d'une application de modélisation de pollution atmosphérique sur plusieurs configurations matérielles différentes:

- sur un CRAY C90 à 16 processeurs.
- sur une Paragon à 256 processeurs.
- et sur un réseau de 256 RS6000 utilisant Ethernet ou ATM dans différentes configurations:
  - Liaison Ethernet, file system séquentiel et PVM.
  - Liaison ATM, file system séquentiel et PVM.
  - Liaison ATM, file system parallèle et PVM.
  - Liaison ATM, file system parallèle et API dédiée.

Leurs résultats sont résumés dans la table 3.1.

Configuration	Temps d'exécution	Prix
Cray C-90	27s	30M$
Paragon	46s	10M$
RS6000 + Eth	27374s	4M$
RS6000 + ATM	2211s	5M$
idem + FS parallèle	205s	5M$
idem + API dédiée	21s	5M$

TAB. 3.1 – *Rapport entre prix et performances*

1. Maximum Transfert Unit

Dans ce cas, l'utilisation de ATM entraine une augmentation de 25% du prix mais permet une accélération d'un facteur 10 ! Enfin, l'utilisation d'une API dédiée combinée à un file system parallèle permet d'obtenir des résultats étonnants.

3.3 Conclusion

Ce papier dresse un état des choses concernant l'utilisation de ATM dans le cadre du calcul distribué. Il met en évidence le fait qu'ATM peut s'avérer être une bonne solution tant sur le plan des performances que sur le plan économique.

Malheureusement, pour obtenir des communications rapides entre stations, il reste encore à régler plusieurs problèmes, notamment celui du temps de latence introduit par ATM.

L'utilisation efficace de ATM dans le calcul distribué passe donc par une phase d'étude et d'analyse, avant d'envisager de définir une API dédiée au calcul parallèle.

Bibliographie

[Anderson 95] T.E. Anderson, D.E. Culler and D.A. Patterson. A case for now (networks of workstations. *IEEE Micro*, 15(1):54–64, February 1995.

[Chang 95] S. Chang, D. Du, J. Hsieh, R. Tsang and M. Lin. Enhanced pvm communications over high-speed lan. *IEEE Parallel and distributed Technology*, pages 20–32, fall 1995.

[de Prycker 91] Martin de Prycker. *Asynchronous Transfert Mode: solution for broadband ISDN*. Ellis horwood series in computer communications and networking, 1991.

[Dowd 95] P.W. Dowd, S.M. Srinidhi, E. Blade and R. Claus. Issues in atm support of high performance geographically distributed computing. In IEEE Computer Society Press, editor, *First Internationnal Workshop on High-Speed Network Computing*, pages 19–28, Santa-Barbara California, April 1995.

[Hsieh 96] J. Hsieh, D. Du and J. MacDonald. Experimental study of extended hippi connections over atm networks. *INFOCOM'96*, 1996. Submitted.

[Huang 94] Chengchang Huang and Philip K. McKinley. Communication issues in parallel computing across atm networks. Technical Report MSU-CPS-94-34, Michigan State University, June 1994. To appear in IEEE Parallel and Distributed Technology.

[Lin 94] M. Lin, J. Hsieh, D. Du and J. MacDonald. Performance of high-speed network i/o subsystems: case study of a fiber channel network. In *Supercomputing'94*, November 1994.

[Lin 95] M. Lin, J. Hsieh, D. Du and J. MacDonald. Distributed network computing over local atm networks. *IEEE, journal on Selected Areas in communications: special issue of ATM Lans*, 1995.

[Martin 95] Richard P. Martin. Hpam: An active message layer for a network of hp workstations. Technical report, University of Berkeley, 1995.

[MPI Forum 95] MPI Forum. MPI: A Message Passing Interface Standard. University of Tennessee, June 1995.

[PVM Team 94] PVM Team. PVM 3 users's guide and reference manual. Technical Report ORNL/TM-12187, Oak Ridge National Laboratory, September 1994. Available via NetLib.

[von Eicken 94] Thorsten von Eicken, Veena Avula, Anindya Basu and Vineet Buch. Low-latency communication over atm networks using active messages. In IEEE Micro, editor, *Hot Interconnects II*, Palo Alto, CA, August 1994.

[Zhou 95] Honbo Zhou and Al Geist. Fast message passing in pvm. In IEEE Computer Society Press, editor, *First Internationnal Workshop on High-Speed Network Computing*, pages 67–73, Santa-Barbara California, April 1995.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399