Ordonnancement et services différenciés pour réseaux rapides

Jérôme Clet-Ortega

RenPar'18 / SympA'2008 / CFSE'6
Fribourg, Suisse, du 11 au 13 février 2008

Ordonnancement et services différenciés pour réseaux rapides

Jérôme Clet-Ortega

Projet INRIA RUNTIME, LaBRI
Université Bordeaux I - 351, cours de la Libération - 33405 TALENCE - France

---

Résumé
Les évolutions dans le domaine du calcul intensif, tant au niveau matériel qu'au niveau niveau applicatif, entraînent inéluctablement un accroissement du nombre de flux devant être gérés par les interfaces de communication. La majorité de ces interfaces concentrent leurs efforts sur la réduction du temps de transmission global, mais ne répondent pas aux nombreux problèmes posés par le partage d'un petit nombre de cartes réseau par un grand nombre d'unités de calcul. Dans cet article, nous exposons l'implémentation d'une nouvelle stratégie à différenciation de services au sein de l'architecture de la bibliothèque pour réseaux rapides NewMadeleine capable d'arbitrer les flux de communication pour appliquer des optimisations agressives, s'accordant avec l'activité des cartes réseau. La stratégie prend en compte les besoins de chacun des flux (priorité, latence, etc.), ceci de façon dynamique, et dirige leurs paquets de données vers les politiques d'ordonnancement adéquates.

Mots-clés : Qualité de service - réseaux rapides - ordonnancement

---

1. Introduction

L'utilisation aujourd'hui répandue des grappes comme plate-formes préférentielles de calcul intensif place la prise en charge des communications, et donc les bibliothèques de communication, parmi les facteurs déterminants influençant les performances applicatives. De plus, les besoins de ce domaine évoluent avec les progrès du matériel, mais aussi avec les techniques et usages dans le développement d'applications. Sur le plan matériel, les processeurs des nœuds évoluent vers plus de cœurs, alors que le nombre de cartes réseau reste sensiblement le même. Cela introduit une concurrence de nombreux flux devant être multiplexés pour l'accès aux ressources d'interconnexion. Sur le plan logiciel, les applications sont de plus en plus développées par assemblage d'éléments logiciels divers (code propre de l'application, intergiciels, bibliothèques, etc.), voire de composants, et même par couplage de codes applicatifs distincts. Chacun de ces éléments introduit ses propres flots de données aux caractéristiques (structure, taille et fréquence des messages) et contraintes spécifiques (latence, débit, réactivité) devant être prises en compte dans le multiplexage des flux.
Or, les bibliothèques de communication actuelles pour les réseaux rapides ne permettent pas d'exprimer et de tenir compte de telles contraintes différenciées entre les flux. Dans cet article, nous proposons une évolution de notre bibliothèque NewMadeleine [2] dont l'ordonnanceur de requêtes réseau est désormais capable d'intégrer des contraintes propres à chaque flux dans son processus d'optimisation des échanges sur le réseau.

2. Équité et favoritisme dans les réseaux rapides

2.1. Pression croissante sur le multiplexage des communications

Des décennies de croissance soutenue nous avaient habitués à voir la puissance des machines progresser de pair avec la fréquence des processeurs. Diverses barrières technologiques récemment atteintes remettent en cause cette tendance. Pour maintenir la croissance de la puissance de calcul, les concepteurs de processeurs se tournent vers la multiplication des unités de traitement physiques avec l'introduction des processeurs multicœurs, ainsi que vers celle des unités virtuelles avec l'utilisation de techniques telles que l'HyperThreading. Ces processeurs à cœurs multiples sont eux-même intégrés en nombre croissant dans les nœuds de calcul (SMP, NUMA).

Cette croissance ne s'applique cependant pas aux périphériques, et notamment aux cartes réseau. Le ratio unités de calcul/unités de communication amorce donc une phase de forte augmentation. Chaque unité de calcul étant susceptible de transmettre des données sur le(s) réseau(x), la concurrence pour l'accès aux unités de communication devient naturellement plus forte.

Par ailleurs, les techniques de développement logiciel évoluent. Le code purement applicatif ne représente souvent qu'une fraction de l'ensemble constituant une application. Le reste est constitué d'intergiciels, de bibliothèques spécialisées, de supports exécutifs d'environnement de programmation ou de compilateurs de langage parallèle, de gestionnaires de *monitoring* (visualisation interactive de l'évolution d'un calcul), de *steering* (modification interactive de paramètres d'un calcul pour le guider ou l'orienter vers une solution préférentielle) ou de *checkpointing* (sauvegarde régulière pour reprise après panne), par exemple ([1, 6]).

Dans cet assemblage de codes, chaque élément est susceptible de communiquer, soit avec d'autres instances de lui-même soit avec d'autres éléments. Cette évolution des pratiques de construction d'applications a donc également pour incidence d'augmenter la concurrence dans l'accès aux unités d'interconnexion. En outre, chaque flot de communication établi présente des caractéristiques propres suivant qu'il sous-tend une re-distribution de données (fort volume, recherche de débit élevé) dans un contexte de couplage de codes par exemple, une commande interactive (volume moyen ou faible, recherche de réactivité élevée) pour interagir avec l'application au cours de son exécution, ou une synchronisation (volume faible, recherche de faible latence). Pour répondre à ces transformations du paysage matériel et logiciel du calcul intensif, les bibliothèques de communication doivent s'adapter.

Un certain consensus s'est peu à peu dégagé entre les divers acteurs de recherche sur la gestion des communications [4, 3, 8, 7] autour de l'utilisation d'une pile réseau à *trois niveaux* fondamentaux : interface matérielle, abstraction, interface logicielle. Au niveau le plus bas, les bibliothèques réseau telles que MX de Myricom ou Elan de Quadrics assurent l'*interface spécifique* avec le matériel. Au niveau le plus haut, les interfaces telles que MPI ou les environnements de programmation fournissent le *modèle* de programmation utilisé par les développeurs d'application. Enfin, le niveau intermédiaire a la responsabilité de fournir l'*abstraction*. Tout en proposant une interface générique, il assure la *projection* des requêtes de communication du modèle sur les commandes de l'interface de bas niveau.

Les modalités de cette projection sont l'enjeu d'importants efforts de recherche actuellement. Par nécessité, tout d'abord, puisque le multiplexage d'un nombre croissant de flux de communication rend cette étape critique. Par intérêt, ensuite, car l'expression des requêtes de communication offre une grande latitude d'optimisation et certaines requêtes peuvent être réordonnées et/ou fusionnées avec bénéfice[2]. Les bibliothèques de niveau intermédiaire ont donc avant tout un rôle d'ordonnanceur et d'optimiseur de requêtes, et de la politique mise en œuvre dépendent les performances.

2.2. La bibliothèque de communication NewMadeleine

Dans NewMadeleine [2] nous avons donc proposé une nouvelle bibliothèque de communication basée sur un moteur d'optimisation SchedOpt à politiques interchangeables de façon à pouvoir expérimenter différentes stratégies. La bibliothèque NewMadeleine est développée au sein de l'équipe RUNTIME pour le support exécutif PM2. Contrairement aux bibliothèques de communication existantes, elle présente l'avantage de pouvoir appliquer dynamiquement des stratégies d'ordonnancement et d'optimisation génériques. NewMadeleine déclenche des opérations sur la liste des requêtes en cours, en fonction de l'état des cartes réseau, de l'état de la machine (lorsqu'un processeur devient inoccupé, etc.), de ce que l'application demande (recouvrement ou attente), et ceci pour que les communications progressent de la meilleure façon possible.

L'architecture de NewMadeleine se décompose en trois couches qui correspondent aux différentes étapes de l'évolution d'un paquet.

La **couche de collecte** est en charge de récupérer les données fournies par les différents flux. Elle les encapsule et rajoute les informations servant à leur identification (identifiant de l'expéditeur, numéro de séquence, etc.) Les données collectées sont alors placées soit sur une liste spécifique au type de réseau renseigné par l'application, soit sur une liste générale dans le cas par défaut, pour permettre à l'ordonnanceur de répartir les données sur toutes les cartes réseau disponibles.

La **couche intermédiaire** est chargée de l'ordonnancement des données et de l'optimisation de leur transfert. Elle s'organise pour former des paquets à fournir aux cartes réseaux disponibles à partir de ces données tout en respectant les contraintes applicatives et en garantissant de bonnes performances. C'est cette couche qui assure la mise en place des protocoles réseau assurant le bon déroulement des communications (rendez-vous, etc.)

La **couche de transfert** est la plus proche du matériel. Elle se charge de vérifier l'état d'occupation des cartes réseau et de signaler à la couche supérieure la disponibilité de celles-ci pour l'envoi de nouveaux paquets. Elle transmet également les données en réception à la couche intermédiaire. Une fenêtre de travail se constitue donc tout naturellement, dans laquelle vont s'accumuler les données lorsque les cartes réseau sont déjà occupées.

Au cœur de cette mécanique, le moteur d'optimisation de NewMadeleine va projeter les paquets des multiples flux applicatifs sur les unités de multiplexage selon différents paramètres (taille de la fenêtre de travail, caractéristiques du réseau, etc.) L'ordonnanceur programmable SchedOpt permet, au travers de l'écriture de stratégies, de définir la fonction d'optimisation globale, et ceci selon un objectif particulier. Une stratégie peut ainsi décider d'agréger certains paquets entre eux pour former une unique requête, ou découper un paquet pour le transmettre sur plusieurs liens de communication.

Notre objectif est donc, dans ce cadre, d'étendre l'ordonnanceur SchedOpt de la bibliothèque de communication NewMadeleine de la capacité d'appliquer des stratégies distinctes aux multiples flots de données. L'idée est d'adapter le concept de services différenciés, issu de l'Internet et des travaux en télécommunication, au contexte spécifique des réseaux rapides.

2.3. Travaux connexes

L'application de techniques de services différenciés ou de support de qualité de service aux réseaux rapides utilisés pour les grappes de calcul a été relativement peu abordée en recherche jusqu'à présent. Quelques équipes en ont néanmoins étudié certains aspects dans le courant des années 90. Dans [5], les concepteurs de l'interface de Fast Messages [8] présentent une étude comparée de diverses approches de qualité de service dans les réseaux rapides pour la réalisation de FM-QoS. Cependant, l'objectif poursuivi est assez différent du nôtre puisqu'il concerne la mise en œuvre de serveurs de forte capacité en termes de nombre de connexions (serveurs vidéo, par

exemple). Les solutions y étant explorées se situent de manière préférentielle au niveau matériel, et notamment au niveau des commutateurs. Dans [10], la plate-forme QUIC reste sur le même créneau des serveurs hautes performances que FM-QoS quant à la finalité, mais se rapproche plus de nos préoccupations dans sa mise en œuvre au sein d'une pile logicielle. Cependant, l'approche retenue repose pour l'essentiel sur une importante « délocalisation » du travail d'ordonnancement au sein des cartes de communication, ce qui va à l'encontre de la recherche de portabilité qui nous intéresse ici. De plus, le modèle proposé repose sur la fourniture, par l'application, de routines d'ordonnancement distinctes sur les différents flots, sans que soit mentionné l'éventuel support d'une optimisation globalisée multi-flots contrairement à la solution que nous proposons.

3. Stratos : la stratégie d'ordonnancement orientée qualité de service

La solution proposée ici permet d'exploiter les informations fournies en amont en implantant, au cœur de la bibliothèque NewMadeleine, un module d'optimisation chargé de mixer tous les flux applicatifs de façon cohérente, c'est-à-dire en adéquation avec leurs besoins.

3.1. Le contexte particulier des réseaux rapides

Les réseaux rapides utilisés pour les grappes ont comme caractéristique une faible latence, ce qui implique que le temps de décision imparti aux algorithmes d'ordonnancement est beaucoup plus faible dans notre contexte que dans celui des réseaux classiques. Par ailleurs, les communications dans les réseaux rapides sont considérées comme fiables par rapport aux réseaux traditionnels.
De plus, contrairement à un routeur qui peut ordonnancer les paquets de ses files selon une échelle temporelle précise, une bibliothèque de communication s'exécute sur un ou plusieurs processeurs qu'elle partage avec les processus applicatifs. Dans ces conditions, offrir des garanties strictes (de bande passante ou de temps de latence maximal, par exemple) n'est pas réellement envisageable. Nous pouvons uniquement fournir des garanties relatives d'un flux par rapport aux autres flux : garantir une latence plus élevée, un débit plus important, une priorité globale, une meilleure réactivité, etc. Fort heureusement, les garanties relatives sont beaucoup plus utiles pour les problèmes étudiés dans notre cadre de travail (favoriser/préserver un flux par rapport à d'autres, par exemple) que les garanties absolues.

3.2. Vers une intégration des services différenciés dans les bibliothèques de communication pour réseaux rapides

Le principe de notre solution s'inspire de celui de la manipulation des flux. Différents besoins sont exprimés par les flux applicatifs et l'objectif est de diriger ces flux vers les mécanismes qui fournissent une réponse à ces besoins. À cette fin, nous implantons au cœur de la bibliothèque de communication une nouvelle stratégie à différenciation de services1. L'idée est d'utiliser les étiquettes (ou « tags ») qui identifient chaque flux. Toutes les données qui appartiennent à un flux, ou même à un groupe de flux, sont associées à un traitement particulier (agrégation, placement dans des listes, etc.), et c'est ce traitement qui détermine le service alloué aux flots. Ce sont les applications qui réalisent l'association flux-traitement au travers des méthodes de l'interface utilisateur. La stratégie implantée, dénommée Stratos, va « aiguiller » les flux vers les files d'attentes liées à des politiques d'ordonnancement. Chaque politique a la charge de fournir un traitement spécifique aux paquets qu'elle gère. C'est donc à l'intérieur de ces politiques que l'on va mettre en place les algorithmes inspirés des travaux sur la qualité de service.

3.3. Fonctionnement de Stratos

Plusieurs politiques ont été d'ores et déjà mises en place dans le moteur d'optimisation de NewMadeleine, chacune correspondant à un algorithme d'ordonnancement particulier.

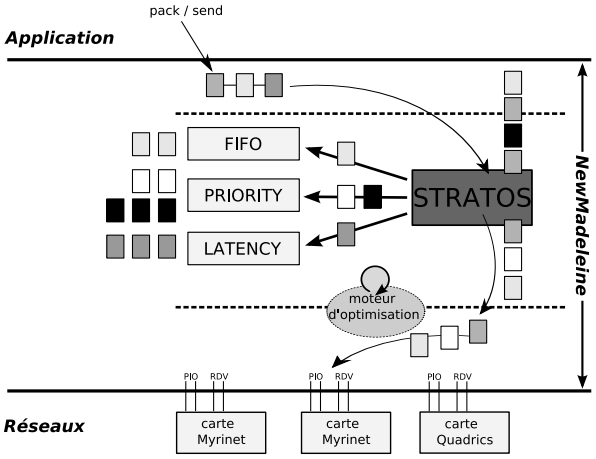


FIG. 1 – Architecture de la stratégie à différenciation de services.

PRIORITY : Dans cette politique, plusieurs files d'attente sont disponibles (actuellement trois), et correspondent chacune à un niveau de priorité. Les flux de cette politique sont alors placés dans la file de la priorité qui leur a été affectée. De ce fait, lorsque la fonction d'optimisation est appelée par la stratégie, les paquets de plus haute priorité sont transmis en premier.
LATENCY : Cette politique n'est pas directement issue des travaux sur la qualité de service puisque son objectif n'est pas de garantir une latence minimale, mais plutôt de privilégier les flux qui délivrent des paquets de petite taille. On retrouve une notion de priorité, non plus entre chaque flux, mais qui peut être au sein d'un même flux. En général, les applications délivrant les flux de ce type vont vouloir se transmettre des petites données en urgence.
RATE : Il s'agit de la politique inverse à la précédente étant donné que ce sont les paquets nécessitant une demande de rendez-vous qui vont être privilégiés. En effet, ce type de paquet va occuper beaucoup plus de bande passante que ceux transportant moins de données. Pour élaborer cet algorithme, les demandes de rendez-vous sont délivrées en priorité. Ainsi, le récepteur pourra émettre son acquittement plus rapidement que dans le cas où aucune distinction n'est faite.
FIFO : Aucun traitement particulier n'est appliqué dans cette politique. Tous les paquets sont placés dans une même liste et le premier arrivé est le premier transmis. C'est le fonctionnement de base de la stratégie principale de NEWMADELEINE. Cette politique est utilisée pour tous les flux qui n'ont pas spécifié de politique particulière.

D'autres politiques sont en projets comme par exemple l'allocation préférentielle de ressources de multiplexages matérielles (*endpoints* MX, *tports* Quadrics, etc.) à certains flux ou groupes de flux sur critères applicatifs.

4. Implantation

Afin d'implanter ce nouveau module au sein de NEWMADELEINE, plusieurs détails sur la mise en oeuvre des politiques de la stratégie ont été abordés.

4.1. Compromis sur l'implantation des politiques

La question de la récursivité dans l'utilisation des stratégies d'ordonnancement de NEWMADELEINE a été le point de départ de notre réflexion. L'idée de base étant d'appliquer un traitement d'ordonnancement/optimisations sur les flux, les stratégies déjà implémentées dans NEWMADE-

LEINE se sont avérées correspondre à ce schéma. Il est donc assez naturel, d'un point de vue conceptuel, d'introduire une notion de méta-stratégie permettant d'appliquer récursivement (et sélectivement) des stratégies d'ordonnancement sur des sous-ensembles de flux. Par ailleurs, il a fallu faire un choix entre utiliser directement les structures de données de la bibliothèque NEWMADELEINE ou alors mettre en œuvre des structures de données plus complexes intégrant les informations sur les garanties propres aux données transportées. Le nombre de flux concurrent étant certes croissant, mais tout de même limité dans le domaine des réseaux rapides, il était moins coûteux de prendre uniquement en compte, de manière globale, les informations de contraintes au niveau de la bibliothèque plutôt que d'intégrer une politique à chaque requête.

4.2. Garantie du niveau de réactivité

Une autre notion qu'il est important d'aborder est la gestion de la réactivité. Nous avons effectué un travail préliminaire de prototypage afin de permettre la mise en relation des politiques du gestionnaire d'événements PIOMAN (PM2 IN/OUT MANAGER [9]) avec les besoins diversifiés des flux de données. Le module PIOMAN est un gestionnaire d'entrées/sorties interagissant avec NEWMADELEINE et MARCEL, la bibliothèque de threads du support exécutif PM2. L'objectif de PIOMAN est d'offrir aux bibliothèques de communication un service de scrutation garantissant un temps de réaction indépendant de l'activité des threads de calcul. Le principe de cette architecture logicielle se base sur un serveur d'événements asynchrones qui se charge de choisir la meilleure méthode à appliquer pour détecter une communication réseau selon les préférences de l'application. C'est-à-dire soit attendre l'événement avec un appel bloquant qui permet des temps de réaction très courts (mécanisme d'interruption) mais induit un léger surcoût en temps, soit utiliser une méthode de scrutation qui peut s'avérer plus efficace dans certains cas (processeur libre). Dans le deuxième cas, la scrutation est entièrement gérée par le serveur d'événements.
Pour permettre à l'application de choisir le niveau de réactivité des flots de communication, nous proposons d'intégrer la notion de priorité dans les requêtes traitées par PIOMAN. Dans chaque requête, un degré de priorité définit la fréquence de scrutation de la détection de terminaison. Lorsque plusieurs événements sont détectés le serveur va réveiller le thread de communication de la requête de plus haute priorité. Cette méthode permet une réactivité accrue pour les requêtes des flux concernés. La modification s'intègre complètement à l'environnement de notre stratégie et n'implique pas l'écriture d'une politique particulière. L'utilisateur doit uniquement préciser quel degré de réactivité il désire pour un flux particulier.

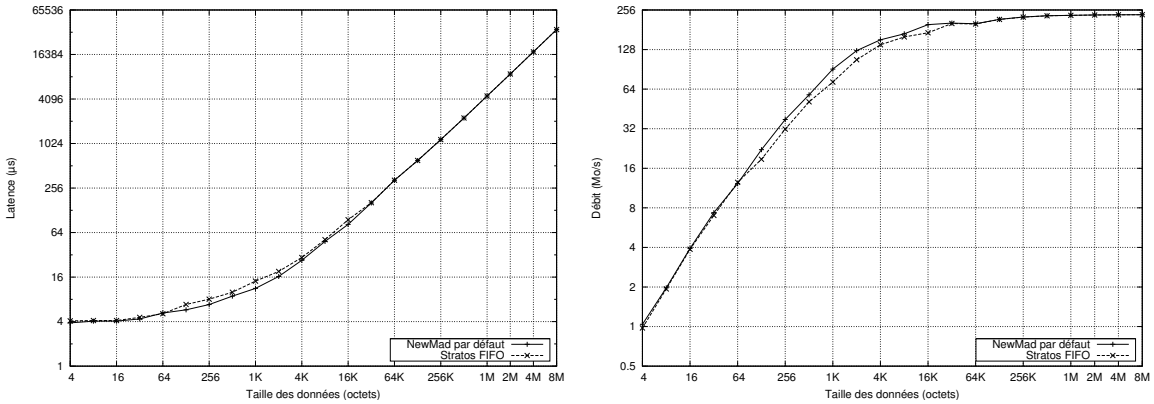
5. Évaluation

Évaluons les performances non plus « brutes » mais relatives aux besoins propres à chaque type de flux. Les tests présentés ici ont pour objectif de comparer les performances entre le mixage des flux opéré par la stratégie principale de NEWMADELEINE (qui a déjà prouvé son efficacité) et celui opéré par les politiques d'ordonnancement proposés dans cet article.

Non-régression

Nous commençons par vérifier que les mécanismes et traitements mis en place comme la recherche de la politique associée à une requête ou la gestion des acquittements pour le cas de priorités, n'affectent pas les performances « brutes » de NEWMADELEINE. Ainsi nous effectuons ici un simple test de type « ping-pong » qui consiste en une série de 2000 allers-retours sur le réseau (afin de lisser les éventuelles fluctuations), avec des données de taille croissante (de 4 octets à 8 Mo). Pour une taille donnée, le temps moyen d'un transfert est obtenu à partir du temps écoulé entre le premier envoi et la dernière réception divisé par deux et par le nombre de tours

de boucles effectués. Que ce soit au niveau de la latence ou au niveau du débit on observe que les performances sont sensiblement les mêmes. Pour les flux ne spécifiant pas de contraintes spécifiques, on garde ainsi les mêmes performances.



(a) Ping-Pong (latence)

(b) Ping-Pong (débit)

FIG. 2 – NewMadeleine par défaut vs Stratos FIFO.

Priorité

Le second test de performance concerne la garantie de priorité d'un (ou même plusieurs) flux par rapport aux autres flux transitant par l'interface de communication. Les performances de certains flux applicatifs prioritaires (les flux de données critiques, par exemple) utilisant NEWMADELEINE ne doivent pas être altérées en raison de la présence d'autres flux moins prioritaires, internes à l'application ou bien générés par d'autres applications . Le test évalue la latence et le débit pour différentes tailles de messages (de 4 octets à 8Mo). Le déroulement des opérations est le suivant :

1. La priorité maximale est donnée au flux `0` et les autres flux ont la priorité la plus basse.
2. Chacun des flux sous-prioritaires envoie une requête d'une taille donnée (les envois sont réalisés en mode non bloquant).
3. Le flux `0` envoie une requête de la même taille et attend le retour du nœud distant.

Nous comparons ensuite le temps mis entre l'envoi du message du flux `0` et la réception du second message prioritaire. Les valeurs observées dans la figure 3 comparent le cas où aucune priorité ne différencie les flux (utilisation de la politique principale de NEWMADELEINE) et celui où le flux dont on calcule les performances est prioritaire, ceci pour différents nombres de flux sous-prioritaires. Le cas particulier où un seul flux (le `0`) transmet sur le réseau nous fournit les valeurs « étalons ». Il s'agit du cas idéal pour le flux `0`. Nous constatons que la gestion des priorités faites dans STRATOS permet de conserver les performances pour le flux `0`.

6. Conclusion

Tant sur le plan matériel que sur le plan applicatif, les communications dans les réseaux rapides se complexifient et de plus en plus de flux peuvent être générés en parallèle, dépassant ainsi les capacités de multiplexage physique des cartes réseaux. Les bibliothèques de communication réalisent ainsi un multiplexage logiciel et optimisent principalement le temps de transmission global. Cependant les nombreux flux parcourant les réseaux rapides peuvent exprimer des besoins

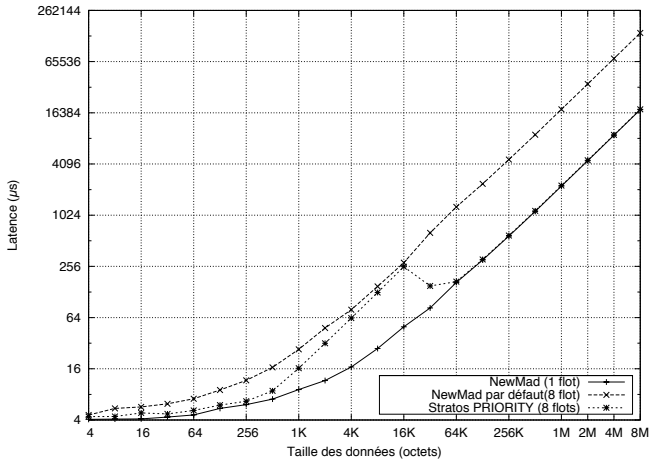


FIG. 3 – Transmission d'un flux prioritaire - NewMadeleine par défaut vs Stratos PRIORITY.

différents selon de multiples contraintes applicatives (paradigme utilisé, etc.) Dans cet article, nous présentons l'implantation, au coeur de la bibliothèque NEWMADELEINE, d'une nouvelle stratégie d'ordonnancement axée sur la différenciation de services. Cette stratégie utilise le moteur d'optimisation de la bibliothèque pour réaliser un mixage dynamique des flux afin de leur garantir à chacun un traitement approprié à ses besoins (réactivité, priorité, etc.)
Dans un avenir proche, il serait intéressant d'étendre la gestion globale du mixage des flux au moment où une carte réseau devient disponible. Pour le moment, la stratégie est de fournir un paquet provenant d'une politique d'ordonnancement à chaque fois que la carte en fait la demande. L'alternance se réalise par un tourniquet sur les politiques actives. Une idée serait de tenter d'agréger les flux en provenance des différentes politiques tout en restant en accord avec les contraintes propres aux politiques. L'utilisation du lien de communication en serait maximisée.

Bibliographie

1. « Projet Lego ». `http://graal.ens-lyon.fr/LEGO/`.
2. O. AUMAGE, E. BRUNET, N. FURMENTO et R. NAMYST. « NewMadeleine : a Fast Communication Scheduling Engine for High Performance Networks ». Dans *CAC 2007 : Workshop on Communication Architecture for Clusters, held in conjunction with IPDPS 2007*, Long Beach, California, USA, March 2007.
3. O. AUMAGE, E. BRUNET, G. MERCIER et R. NAMYST. « High-Performance Multi-Rail Support with the NewMadeleine Communication Library ». Dans *HCW 2007 : the Sixteenth International Heterogeneity in Computing Workshop, held in conjunction with IPDPS 2007*, Long Beach, California, USA, March 2007.
4. E. BRUNET. « NewMadeleine : ordonnancement et optimisation de schémas de communication haute performance (version étendue de Perpi'06). ». *Technique et Science Informatiques*, 2008. To appear.
5. A. A. CHIEN et J. H. KIM. « Approaches to Quality of Service in High-Performance Networks ». *Lecture Notes in Computer Science*, 1417, 1998.
6. A. ESNARD. « *Analyse, conception et réalisation d'un environnement pour le pilotage et la visualisation en ligne de simulations numériques parallèles* ». Informatique, Université de Bordeaux 1, décembre 2005.
7. W. GROPP, E. LUSK et A. SKJELLUM. *Portable Parallel Programming with the Message Passing Interface*. MIT Press, seconde édition, 1999.
8. S. PAKIN, V. KARAMCHETI et A. A. CHIEN. « Fast Messages : Efficient, portable communication for workstation clusters and MPPs ». *IEEE Concurrency*, 5(2) :60–73, /1997.
9. F. TRAHAY. « Gestion de la réactivité des communications réseau ». Mémoire de dea, Université Bordeaux 1, juin 2006.
10. R. WEST, R. KRISHNAMURTHY, W. K. NORTON, K. SCHWAN, S. YALAMANCHILI, M.-C. ROSU et V. SARAT. « QUIC : A Quality of Service Network Interface Layer for Communication in NOWs ». Dans *Heterogeneous Computing Workshop*, pages 199–208, 1999.