Impacts des effets NUMA sur les communications haute performance dans les grappes de calcul

Stéphanie Moreaud

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

Impacts des effets NUMA sur les communications haute performance dans les grappes de calcul

Stéphanie Moreaud
Laboratoire Bordelais de Recherche en Informatique, INRIA Bordeaux - Sud-Ouest
Stephanie.Moreaud@labri.fr

Résumé
La multiplication des processeurs et des cœurs dans les machines a conduit les architectes à renoncer aux bus mémoire centralisés. Les effets NUMA (*Non-Uniform Memory Access*), surtout connus pour leur impact sur l'efficacité de l'ordonnancement de processus, révèlent également une influence sensible sur les performances des entrées-sorties. Dans cet article, nous présentons une évaluation de leur incidence sur les performances réseau dans les grappes de calcul, en exhibant leur ampleur et leur aspect parfois asymétrique sur le débit. Nous proposons une solution de placement automatique et portable des tâches de communication dans la bibliothèque NEWMADELEINE qui permet d'obtenir des performances analogues à celles d'un placement manuel, en utilisant des informations de topologie collectées auprès du système.

Mots-clés : Communications, NUMA, placement, réseaux rapides, grappes.

1. Introduction et contexte

L'émergence des grappes de calcul dans le domaine du calcul haute performance, il y a une quinzaine d'années, a engendré, presque paradoxalement, un regain d'intérêt pour les « petites » machines multiprocesseurs dont les grappes sont majoritairement constituées. Ces machines multiprocesseurs ont l'avantage d'être peu onéreuses et restaient, jusqu'il y a peu, assez faciles à programmer grâce à leur architecture symétrique simple. Toutefois, en raison des contraintes de dissipation thermique qui limitent la fréquence des processeurs, on assiste actuellement à une multiplication des « cœurs » au sein de ces machines. C'est la nouvelle voie de développement qu'ont choisie les constructeurs pour accroître les performances des processeurs contemporains. Cette prolifération des cœurs a renforcé l'organisation hiérarchique de la mémoire et la multiplication des caches, pour éviter le goulot d'étranglement créé par les accès concurrents aux bancs centralisés. Au sein de telles architectures, les temps d'accès mémoire dépendent de l'emplacement physique des cœurs, des caches et des bancs mémoire.
De nombreuses études sur les effets du placement des tâches et de la mémoire sur ces machines NUMA ont été menées, notamment dans le contexte de l'ordonnancement. Avec l'éclatement de la mémoire, le placement et l'ordonnancement des threads devient crucial. En effet, un thread doit être placé près des données qu'il manipule pour éviter les accès distants, et l'importance du placement augmente significativement avec la taille et le facteur NUMA du système [4]. Les affinités des threads pour le partage des caches et les accès mémoire jouent également un rôle notable sur les performances générales, et des travaux exposent l'importance de la localité des données [11] ou encore l'intérêt de faire migrer des pages pour conserver les affinités mémoire [5].
Les architectures NUMA sont de plus en plus présentes en calcul haute performance, où de nombreux nœuds sont connectés par des réseaux rapides tels que INFINIBAND, MYRI-10G ou encore QSNET II. Ces technologies, conçues initialement pour de petits nœuds SMP, connectent désormais des nœuds

comprenant de plus en plus de processeurs et de cœurs qui partagent les mêmes périphériques réseau. L'obtention de performances optimales dans ces grappes de nœuds hiérarchiques passe par la maîtrise des effets NUMA et par l'optimisation des communications entre les nœuds. Il est clair que les performances des communications peuvent être sensibles aux caractéristiques matérielles, nécessitant des réglages minutieux [1], et l'influence du placement des tâches de communication peut être important. En particulier, les effets NUMA peuvent influer sur les communications, notamment MPI, à l'intérieur de grosses machines [6]. Les contrôleurs d'entrées-sorties se trouvent souvent physiquement plus proches de certains processeurs et bancs mémoire que d'autres, pouvant entraîner des variations de temps d'accès. Des systèmes d'exploitation tels que LINUX disposent d'informations détaillées quant au placement du matériel et fournissent aux différents sous-systèmes logiciels la possibilité de les utiliser [8]. Au niveau des applications utilisateur, des outils (`libnuma` [7]) permettent ainsi de forcer le placement des threads et de la mémoire. Ceux-ci sont systématiquement utilisés dans les mesures de performances réseau impliquant des machines NUMA pour garantir des performances optimales et reproductibles, ce qui montre leur sensibilité au placement.
Bien qu'assez connu, le problème du placement des communications n'a jamais été étudié en détails. Pouvoir prédire les variations des performances des communications et comprendre leurs causes exactes devrait permettre d'envisager des stratégies permettant d'éviter leur dégradation. Nous proposons donc une étude des effets NUMA sur les performances des communications sur réseaux rapides. Notre objectif est dans un premier temps de quantifier ces effets, que nous appellerons par extension NUIOA (*Non Uniform Input/Output Access*), et de déterminer les conditions dans lesquelles ils sont significatifs. Nous verrons donc en sections 2 et 3 les impacts du placement sur le débit puis sur la latence, aux niveaux des transferts locaux et applicatifs. Dans un second temps nous proposerons une solution de placement automatique des tâches de communication et de la mémoire, décrite en section 4, pour minimiser les effets néfastes et maximiser les performances en fonction des besoins applicatifs.

2. Mise en évidence des effets NUIOA sur le débit des communications

Pour estimer l'influence du placement des threads et de la mémoire sur le débit, nous avons réalisé une série de tests réseau : sur les transferts locaux (entre la carte réseau et la mémoire) pour mesurer les impacts NUIOA au niveau de la machine, et sur des communications réelles pour évaluer leurs répercussions au niveau applicatif.

2.1. Impact sur le débit des transferts locaux

Nous avons étudié l'impact NUIOA sur les transferts locaux en examinant les performances des transferts DMA (*Direct Memory Access*) successivement sur différents processeurs de la machine. Nous nous sommes pour cela appuyés sur les tests `mx_dmabench` et `qsnet2_dmabench` fournis par les constructeurs MYRICOM et QUADRICS qui effectuent des transferts vers et depuis la carte réseau en mode DMA, et avons créé un test analogue pour INFINIBAND. Ces tests ont été effectués sur différentes architectures NUMA (Figure 1 (a) et (b)) dotées de processeurs AMD OPTERON 1,8 GHz et disposant des interfaces réseau MYRICOM MYRI-10G, QUADRICS QSNET II (ELAN4) et INFINIBAND 8x (MELLANOX INFINIHOST III).
Les résultats correspondants, donnés en Table 1, révèlent une baisse de performance pour un placement distant (sur un nœud autre que le plus proche de l'interface réseau) d'autant plus significative que le débit est élevé. Les résultats pour QSNET II ne montrent pas de variation relative au placement tandis que ceux pour INFINIBAND et MYRI-10G témoignent d'une dégradation de débit de près de 25%. Cette perte est en outre asymétrique, puisque les lectures DMA ne semblent pas affectées alors que les écritures le sont. L'observation des impacts NUMA sur de simples copies mémoire confirment que la dégradation liée au placement augmente avec le débit théorique, et que l'écriture en souffre bien plus que la lecture.

		Placement proche	Placement distant	Impact
MYRI-10G	Lecture	1308	1295	- 1 %
	Écriture	1518	1162	- 24 %
INFINIBAND	Lecture	*1075*	*1075*	0 %
	Écriture	1392	1071	- 23 %
QSNET II	Lecture	879	879	0 %
	Écriture	925	925	0 %
Acces mémoire	Lecture	2696	1871	-31 %
du processeur	Écriture	4765	2952	-38 %

TAB. 1: Impact du placement NUMA sur le débit (Mo/s) pour des transferts locaux par DMA de la mémoire vers la carte (écriture), de la carte vers la mémoire (lecture), et des accès mémoire sur des machines 2 nœuds NUMA.

2.2. Répercussions au niveau applicatif

Nous avons examiné les répercussions de la chute de débit lors des transferts locaux au niveau applicatif. La Figure 2 montre l'impact NUIOA lors d'un usage extrême de la bande passante, un ping-pong multi-rail sur la plateforme NEWMADELEINE [3], pour des machines multi-OPTERON deux nœuds disposant des réseaux MYRI-10G et QSNET II (Figure 1(a)). Chaque message de l'application est transmis de façon transparente (*stripping*) sur l'ensemble des réseaux disponibles en fonction de leur performances respectives, entraînant une augmentation conséquente de la bande passante. Lorsque la taille des mes-

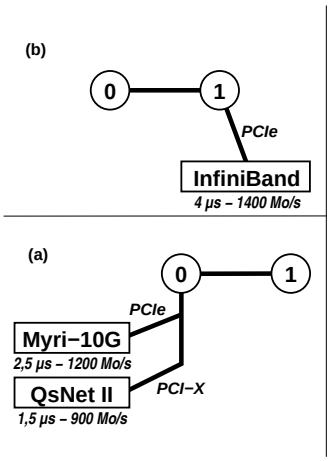


FIG. 1: Plateforme de test : architectures multi-processeurs AMD OPTERON 1,8 GHz bi-cœurs, deux et huit nœuds.

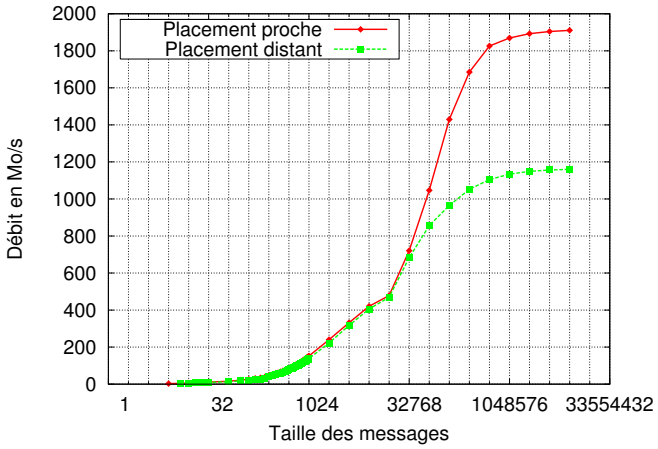


FIG. 2: Débits d'un ping-pong multirail sur la plateforme NEWMADELEINE en fonction du placement sur des machines 2 nœuds.

sages atteint 32 Ko, le placement distant commence à limiter les performances, engendrant une dégradation pouvant aller jusqu'à plus de 40% par rapport au débit maximum obtenu (environ 2 Go/s pour des messages de 8 Mo) avec un placement sur le nœud le plus proche du périphérique réseau. Nous avons constaté des effets similaires avec divers modes de communication et réseaux. Avec un placement distant, le débit du ping-pong multirail (agrégation des débit des réseaux) est inférieur à celui d'un ping-pong monorail sur MYRI-10G (1200 Mo/s). Nous supposons que cela tient d'une congestion due au

fait que les deux contrôleurs d'entrées-sorties sont utilisés simultanément et connectés sur le même lien HYPERTRANSPORT (bus mémoire des OPTERON).

Les tests de ping-pong monorail effectués sur les différentes interfaces de notre plateforme de tests présentent des résultats variés. Alors que les mesures des transferts locaux sur le réseau MYRI-10G révélaient d'importantes modifications de débit, celles-ci ne sont pas retranscrites au niveau applicatif pour lequel le débit est moindre (1200 Mo/s contre 1500 Mo/s lors des transferts locaux). Nous n'observons pas de changement sur QSNET II. Sur le réseau INFINIBAND, qui offre un débit plus élevé, la dégradation reste conséquente, avec une chute supérieure à 20%. Il semble donc que seuls les débits élevés (au delà de 1 Go/s) subissent une détérioration relative au placement.

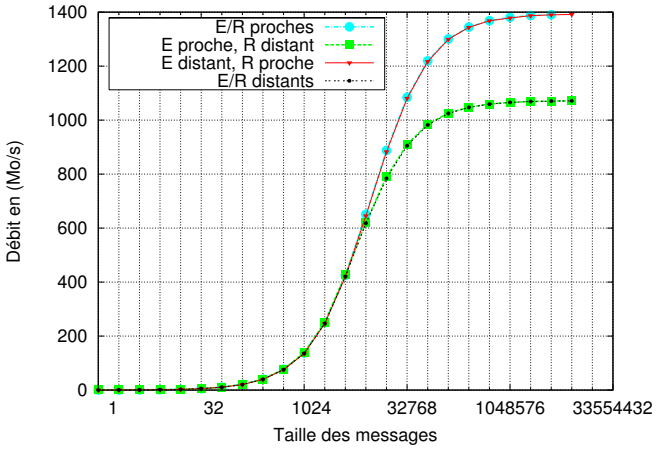


FIG. 3: Variation du débit d'un transferts RDMA sur le réseau INFINIBAND en fonction des placements NUMA côté émetteur (E) et récepteur (R) sur une machine deux nœuds.

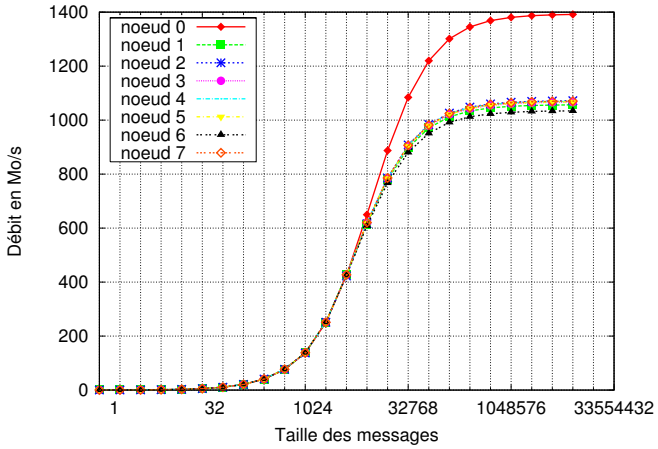


FIG. 4: Variation du débit d'un transferts RDMA sur le réseau INFINIBAND en fonction du placement côté récepteur sur une machine multi-OPTERON à huit nœuds NUMA.

Pour enrichir nos expérimentations nous avons développé des tests de RDMA (*Remote Direct Memory Access*) sur le réseau INFINIBAND, effectués sur des machines deux nœuds pourvues de ce réseau (Figure 1(b)) dont les résultats sont présentés en Figure 3. Les mesures de débit pour divers placements montrent des effets similaires à ceux décrits précédemment, avec une saturation avant 1100 Mo/s et une perte de 25% par rapport au débit maximum. Elles soulignent la répercussion au niveau applicatif des effets asymétriques présentés en section 2.1, jusqu'alors invisible dans les tests bidirectionnels. Nous observons que le placement de la mémoire cible de l'écriture RDMA importe, contrairement à celui du côté émetteur. Nous avons constaté des réactions semblables dans le cas d'une lecture RDMA. Il apparaît donc que seul l'emplacement du tampon mémoire dans lequel les données sont « déposées » importe. Cet effet inattendu est provoqué par une saturation du bus HYPERTRANSPORT. Ses spécifications semblent en effet indiquer que les nombres de tampons matériels de requêtes et de réponses peuvent différer.

La Figure 4 étend ces résultats à notre machine à huit nœuds NUMA (Figure 1(c)) et indique que le débit d'une écriture RDMA ne décroît pas avec une augmentation du nombre de liens HYPERTRANSPORT traversés. Si le placement est déjà distant, augmenter la distance à l'interface réseau ne réduit pas davantage les performances. Ce résultat est valable dans le seul cas où la machine n'est pas chargée. Dans le cas contraire, nous avons observé que les contentions sur les liens HYPERTRANSPORT entraînent une baisse conséquente des performances avec la distance.

3. Effets NUIOA sur la latence des communications

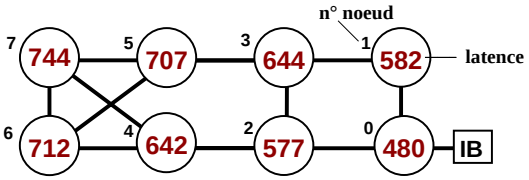
Nous avons poursuivi notre étude en examinant l'impact NUIOA sur la latence lors des transferts locaux et applicatifs.

3.1. Influence sur la latence des transferts locaux

Obtenir des communications avec une latence faible implique de réduire au maximum le chemin critique des données. La meilleure méthode pour le transfert de petits messages est généralement de les écrire par PIO (*Programmed Input/Output*) dans l'interface réseau côté émetteur, et que l'autre interface les place dans la mémoire par DMA côté récepteur. Si l'implémentation de cette stratégie peut varier, les effets NUIOA attendus sont les mêmes : plus le nombre de liens HYPERTRANSPORT traversés est grand, plus la latence devrait être élevée. La Figure 5(a) présente les résultats de tests de PIO, correspondant aux délais entre l'envoi d'une requête à l'interface réseau commandée par PIO, et le moment ou l'événement correspondant est notifié en mémoire. Ces résultats sont obtenus grâce aux tests `mx_piobench` et

	placement proche	placement distant	Surcoût
MYRI-10G	1739	1794	55
QSNET II	1610	1670	60
INFINIBAND	464	559	95

(a) Latences relatives au placement pour de petites requêtes à différentes interfaces réseau sur une machine 2 nœuds.



(b) Latences relatives au placement pour un aller-retour PIO entre un processeur et l'interface INFINIBAND sur une machine huit nœuds.

FIG. 5: Impact du placement NUMA sur la latence (en nanosecondes).

`qsnet2_dmatest` des constructeurs MYRICOM et QUADRICS, et à un test « maison » pour le réseau INFINIBAND. Ces outils sont spécifiques à chaque interface et leurs résultats n'ont pas vocation à être comparés entre eux. Ils fournissent toutefois une méthode similaire pour évaluer l'impact NUIOA du placement sur la latence. Nous notons un surcoût entre 55 et 95 nanosecondes pour un aller-retour entre le processeur et l'interface réseau, soit un coût de près de 40 ns pour la traversée simple d'un lien HYPERTRANSPORT. L'extension de ces expériences à notre machine à 8 nœuds (Figure 5(b)) confirme bien une augmentation de la latence de près de 40 ns par traversée d'un lien HYPERTRANSPORT qui traduit la hiérarchie de la machine, et expose des singularités de routage que nous ne détaillerons pas ici.

3.2. Répercussions au niveau applicatif

Au niveau applicatif, l'examen des latences des communications sur le réseau MYRI-10G révèle une influence négligeable du placement NUMA, avec un surcoût d'environ 25 ns pour une latence théorique de 2,5 μs, soit une fluctuation de 2% par aller-retour. Le surcoût est d'autant plus négligeable sur INFINIBAND puisque la latence est plus élevée. Le réseau QSNET II montre une influence NUIOA plus grande puisque sa latence théorique est très inférieure, entre 1 et 2 μs selon le mode de communication. Avec l'utilisation des opérations *Put/Get* natives, la latence brute est très proche de 1 μs et l'impact d'un placement distant atteint 100 ns de chaque côté entraînant une différence globale de 20%. Cet impact peut être conséquent pour des applications très sensibles à la latence, sur de grandes machines telles que notre machine huit nœuds. D'un point de vue général, la perte occasionnée par un placement distant est du même ordre que celle d'un défaut de cache et peut être considérée négligeable dans la plupart des cas.

4. Implémentation d'une solution de placement automatique

Nous avons vu en section 2 et 3 que le placement des tâches de communication et des tampons associés a un impact réduit sur la latence mais peut prendre des proportions considérables sur le débit. Pour garantir des communications optimales, il est indispensable de placer les tâches qui en sont responsables au plus près des cartes réseau. Les processus utilisés lors des tests réseau sont toujours placés manuellement pour en tirer les meilleures performances, mais aussi pour les rendre reproductibles. En effet, une tâche non liée à un processeur peut par exemple subir une migration aléatoire par l'ordonnanceur lors du réveil d'un processus démon. À l'opposée, un placement manuel est contraignant car l'utilisateur doit prendre connaissance de l'architecture matérielle et connaître l'emplacement des périphériques. De plus, un tel placement ne peut convenir aux cas complexes des applications multithreadées effectuant des communications irrégulières, pour lesquelles un compromis entre le placement proche des périphériques pour les communications et l'exploitation de l'ensemble des processeurs des différents nœuds NUMA est nécessaire. La portabilité des performances devrait être garantie par le système d'exploitation ou les intergiciels grâce à des mécanismes de placement des tâches et des buffers. Nous proposons dans cette section une solution de placement automatique dans la bibliothèque de communication NEWMADELEINE [2].

4.1. Trouver le nœud le plus proche de chaque carte réseau

Effectuer un placement proche d'une interface nécessite de savoir quel nœud NUMA en est le plus près physiquement. Les tests de transferts locaux présentés dans les sections 3.1 et 2.1, et tout particulièrement ceux de latence nous permettent de détecter l'emplacement physique de la carte, mais les contraintes qu'ils imposent (machine totalement non chargée et droits privilégiés) les rendent inadaptés à un usage courant. L'idéal serait d'exposer la topologie NUMA aux intergiciels qui seraient chargés du placement. Certains systèmes d'exploitation connaissent parfois la position NUMA des périphériques. Nous avons intégré un patch dans le noyau LINUX 2.6.22 pour exposer le nœud NUMA de chaque interface, de sorte que les intergiciels puissent utiliser facilement les outils `libnuma` pour placer les tâches. Nous avons implémenté cette idée dans l'intergiciel NEWMADELEINE, développé dans notre équipe. Les attributs fournis par le système ne peuvent pas être immédiatement traduits en informations de placement exploitables au niveau applicatif, puisque les applications manipulent des descripteurs virtuels de haut niveau, pointant sur des canaux de communication cachés par le système. La correspondance entre ces descripteurs et les périphériques physiques ne peut être faite que par les pilotes réseau. Les pilotes de QSNET II et ETHERNET ne le permettent pas et nous ne disposons actuellement d'aucun moyen d'obtenir l'emplacement de la carte réseau du système. Par contre, le pilote INFINIBAND fournit cette équivalence [10], et nous avons ajouté un support explicite dans le pilote MX [9] pour obtenir l'attribut NUIOA associé à un canal de communication sur le réseau MYRI-10G.

4.2. Placement automatique : mise en oeuvre dans NEWMADELEINE

Une fois que la bibliothèque NEWMADELEINE connaît la position physique des cartes réseau, elle doit placer les tâches et la mémoire correctement. Il est important d'effectuer ce placement au bon moment. Trop hâtif, il risquerait de forcer le placement de ressources qui n'ont pas lieu de l'être, par exemple des threads ou des zones mémoire non mis en jeu dans les communications. Un placement tardif pourrait quant à lui imposer une migration délicate de ressources complexes telles que des pages verrouillées en mémoire physique en vue d'un transfert DMA. La bibliothèque doit donc d'abord interroger chaque pilote pour obtenir l'attribut NUIOA de toutes les cartes, puis effectuer le placement en conséquence, et enfin initialiser les drivers aux bons endroits. Cette initialisation entraîne alors un placement correct et définitif des ressources spécifiques aux pilotes, en particulier les tampons mémoire utilisés par la suite lors de chaque transfert DMA. Par contre, l'intergiciel doit être capable d'exposer ces informations de placement à l'application pour qu'elle puisse allouer correctement ses tampons propres. Enfin, il faudra

à terme être capable d'utiliser ces informations par exemple pour migrer les tâches près des cartes réseau lors de la soumission de requêtes.
Nous avons implémenté cette stratégie dans NEWMADELEINE et vérifié que nous obtenons toujours les performances optimales (équivalentes à celles d'un placement manuel). Chaque application utilisant les réseaux MYRI-10G ou INFINIBAND est automatiquement placée sur le nœud NUMA le plus proche du périphérique correspondant. Lorsqu'une technologie de réseau qui ne fournit pas les attributs NUIOA est utilisée, NEWMADELEINE part du principe qu'aucun nœud n'est plus proche de l'interface réseau que les autres et ne prend donc pas cette interface en compte dans sa prise de décision de placement.

4.3. Cas des applications multirail

Les applications multirail nécessitent d'être placées avec attention puisque les débits attendus sont élevés (jusqu'à 2 Go/s) et peuvent subir une baisse de 40% (Figure 2). Ce placement est d'autant plus complexe que les interfaces réseau utilisées n'exposent pas forcément les mêmes affinités NUIOA. Il est commun que les machines disposent de plusieurs bus d'entrées-sorties connectés à des nœuds NUMA distincts. Pour gérer les multiples interfaces réseau mises en jeu dans les communications multirail, la bibliothèque doit comparer les attributs NUIOA de chaque réseau et éventuellement avoir à en privilégier certains si des conflits d'affinité apparaissent. Nous avons implémenté cette idée dans NEWMADELEINE. En cas de conflit, les performances des différents réseaux peuvent être un bon critère de choix. Une application qui nécessite avant tout une latence faible devrait être par exemple plutôt placée près de l'interface d'un réseau QSNET II tandis qu'une application nécessitant un haut débit devrait plutôt être proche de l'interface d'un réseau INFINIBAND. Une application demandant une forte utilisation du processeur devrait quand à elle être établie sur un processeur libre plutôt qu'à proximité d'une interface réseau. Il est donc nécessaire que l'application puisse donner des indices de placement en fonction de ses besoins. Dans le cas commun d'un multirail homogène avec des réseaux similaires connectés sur différents nœuds NUMA, aucun placement proche d'une interface ou d'une autre ne sera à priori préférable.

5. Conclusions et perspectives

Cet article présente une étude des effets NUIOA sur les communications dans des grappes de calcul. Les effets NUIOA, connus dans le contexte de l'ordonnancement et du placement des threads et des données, n'ont à notre connaissance pas fait l'objet d'étude approfondie dans le cadre des machines distribuées avec communications sur réseaux rapides. Nous avons présenté une évaluation de l'impact NUMA du placement des tâches et des données sur les performances des communications sur réseaux rapides, lors des transferts locaux et au niveau applicatif. Les mesures montrent que le placement des tâches sur un nœud NUMA loin de l'interface réseau entraîne une chute de performance. Alors que la latence augmente légèrement avec la distance entre les éléments de la machine (environ 40 ns par lien HYPERTRANSPORT), le débit souffre d'une chute dramatique pour des transferts effectuant une utilisation intensive de la bande passante, avec une perte de plus de 40% pour 2 Go/s. Nous avons montré que les effets NUIOA sur le débit sont asymétriques puisque seules les zones mémoire cibles semblent nécessiter un placement rigoureux sur le nœud NUMA le plus proche de l'interface réseau. Nous avons enfin proposé une implémentation de placement automatique dans la plateforme de communications NEWMADELEINE. Nous avons exposé les attributs NUIOA pertinents des périphériques réseau MYRI-10G et INFINIBAND pour l'application et les utilisons pour placer une tâche de communication proche de ce périphérique. Cette implémentation garantit la portabilité des performances en plaçant correctement la tâche de communication de façon automatique, et permet d'obtenir des performances aussi bonnes que dans le cas d'un placement manuel, même dans le cas de communications multirail.
Nous souhaitons désormais étudier l'impact des effets NUIOA sur différentes technologies de réseau, en les regardant précisément pour divers modes de communications (PIO, copie et DMA, enregistrement

mémoire et DMA), qui diffèrent par le taux d'implication des ressources (processeur en PIO, mémoire en DMA) et dont les contraintes de placement peuvent ainsi varier. Nous projetons également d'examiner ces effets sur d'autres architectures, et plus particulièrement celles basées sur ITANIUM 2, avec un bus mémoire hiérarchique et plusieurs bus d'entrées-sorties, ou encore sur la technologie attendue INTEL QUICKPATH qui devrait bientôt généraliser l'adoption d'architectures du type HYPERTRANSPORT. Nous envisageons par ailleurs d'étudier divers systèmes d'interconnexion réseau, notamment INFINIPATH dont la connectivité HTX pourrait créer des effets NUIOA autres que ceux visibles sur les bus PCI-X et PCIE. L'implémentation du placement automatique dans la bibliothèque de communication NEWMADELEINE constitue un premier pas dans la conception d'un placement adaptatif relatif à la topologie des machines. À terme nous souhaitons utiliser des informations de topologie détaillées pour gérer le placement des tâches au travers de machines complexes, en considérant de potentielles congestions sur les bus. Nous espérons pouvoir effectuer un placement automatique pour des applications complexes mixant des sections parallèles OPENMP (qui distribuent beaucoup de threads sur la machine) et des communications MPI (qui nécessitent un nombre restreint de threads dont le placement proche des interfaces réseau est crucial). Des indications fournies par l'application devraient pouvoir assister l'intergiciel lors des décisions de placement, pour influencer celles-ci en fonction des besoins dominants en latence, débit ou encore en disponibilité processeur. Un de nos principaux objectifs sera enfin d'intégrer ces informations de placement dans un ordonnanceur tel que MARCEL [12], pour guider les threads de communication vers les interfaces réseau, adaptant ainsi l'ordonnancement à la topologie des systèmes d'entrées-sorties.

Bibliographie

1. L. ARBER et S. PAKIN. « The Impact of Message-Buffer Alignment on Communication Performance ». *Parallel Processing Letters*, 15(1) :49–65, mars 2005.
2. O. AUMAGE, E. BRUNET, N. FURMENTO et R. NAMYST. « NewMadeleine : a Fast Communication Scheduling Engine for High Performance Networks ». Dans *Proceedings of the Workshop on Communication Architecture for Clusters (CAC 2007), held in conjunction with IPDPS 2007*, Long Beach, CA, mars 2007.
3. O. AUMAGE, E. BRUNET, G. MERCIER et R. NAMYST. « High-Performance Multi-Rail Support with the NewMadeleine Communication Library ». Dans *Proceedings of the Sixteenth International Heterogeneity in Computing Workshop (HCW 2007), held in conjunction with IPDPS 2007*, Long Beach, CA, mars 2007.
4. T. BRECHT. « On the Importance of Parallel Application Placement in NUMA Multiprocessors ». Dans *Proceedings of the Fourth Symposium on Experiences with Distributed and Multiprocessor Systems (SEDMS IV)*, San Diego, CA, Sept 93.
5. R. Chandra *et al.*. « Scheduling and page migration for multiprocessor compute servers ». Dans *Proceedings of the sixth international conference on Architectural support for programming languages and operating systems table of contents*, pages 12–24, San Jose, CA, 1994.
6. S.R. Alam *et al.*. « Characterization of Scientific Workloads on Systems with Multi-Core Processors ». Dans *Proceedings of the International Symposium on Workload Characterization (IISWC)*, San Jose, CA, 2006.
7. A. KLEEN. « A NUMA API for LINUX », avril 2005. `http://www.novell.com/collateral/4621437/4621437.pdf`
8. C. LAMETER. « Local and Remote Memory : Memory in a Linux/NUMA System », juillet 2006. `http://kernel.org/pub/linux/kernel/people/christoph/pmig/numamemory.pdf`
9. Myricom, Inc. « *Myrinet Express (MX) : A High Performance, Low-Level, Message-Passing Interface for Myrinet* », 2006. `http://www.myri.com/scs/MX/doc/mx.pdf`
10. « OpenIB Alliance ». `http://www.openib.org`
11. M. STECKERMEIER et F. BELLOSA. « Using Locality Information in Userlevel Scheduling ». Rapport Technique TR-95-14, University of Erlangen-Nürnberg – Computer Science Department, décembre 1995.
12. S. THIBAULT. « A Flexible Thread Scheduler for Hierarchical Multiprocessor Machines ». Dans *Proceedings of the Second International Workshop on Operating Systems, Programming Environments and Management Tools for High-Performance Computing on Clusters (COSET-2)*, Cambridge, MA, juin 2005.