Impacto da rede de interconexão para a formação eficiente de agregados de computadores

Luiz C. Pinto, Rodrigo P. Mendonça, M. A. R. Dantas

Laboratório de Pesquisa em Sistemas Distribuídos (LaPeSD)
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil
{luigi, rodrigop, mario}@inf.ufsc.br

Resumo. *Este artigo apresenta uma análise do impacto da rede de interconexão sobre o desempenho de agregados computacionais. Para tanto, são analisados dois algoritmos de larga utilidade no meio científico, de diferentes granularidades, sobre duas redes de interconexão distintas: Ethernet e Myrinet. Os resultados empíricos indicam que os ganhos em função da rede de interconexão são crescentes apenas para o algoritmo de granularidade fina. Sendo assim, deve-se dispor atenção especial sobre a relação entre granularidade do algoritmo e rede de interconexão para a formação eficiente de agregados computacionais adequados às necessidades de aplicações científicas com desempenho positivamente diferenciado.*

1. Introdução

O surgimento de dispositivos de interconexão com alta taxa de transmissão e a popularização dos computadores pessoais tornaram viável a implementação de sistemas distribuídos como os agregados computacionais. É uma opção de baixo custo se comparado às máquinas paralelas convencionais. Nestes tipos de sistema de computação, a rede de interconexão e o software de comunicação são componentes decisivos para a formação de um sistema de alto desempenho. Recentemente, com a redução de custo, dispositivos de interconexão diferenciados vêm ganhando espaço no mercado e na pesquisa científica. Exemplos clássicos são: Myrinet [Boden et al 1995], Quadrics [Petrini et al 2002] e Infiniband [Cassiday 2000]. Poderosos agregados computacionais optam por estas tecnologias de interconexão mais eficientes. No entanto, os ganhos dependem de outros fatores redutores do desempenho.

O objetivo deste trabalho é desenvolver um estudo experimental, visando a formação mais adequada de agregados computacionais quanto ao elemento rede de interconexão. Para tanto, o comportamento de dois meios de interconexão distintos, Ethernet e Myrinet, serão observados com a execução de dois algoritmos paralelos de granularidades distintas. O tempo de execução dos algoritmos será tomado como critério para a análise [Jordan e Alaghband 2003]. Os algoritmos escolhidos como objetos desta análise são exemplos amplamente usados no meio científico: multiplicação de matrizes (MM) [Dongarra et al 2007] e transposição de matrizes (TM) [Choi et al 1995].

Enfim, com base nos resultados empíricos, apresenta-se uma posição mais precisa sobre o custo do fator comunicação para a formação de um agregado computacional, cuja configuração deve ser adequada às necessidades de alto desempenho de aplicações científicas e também restritas ao melhor custo/benefício.

O artigo está organizado como segue. A seção 2 introduz uma visão geral em torno dos aspectos teóricos relevantes, particularizando-os ao sistema em questão. Neste sentido, características gerais e específicas da biblioteca de comunicação adotada são apresentadas. A mesma seção 2 ainda apresenta os trabalhos correlatos. Na seção 3, a contribuição deste trabalho é descrita, seguida da caracterização do ambiente onde os experimentos são realizados e do método de coleta dos resultados. Os resultados dos experimentos são explicados na seção 4. Finalmente, na seção 5, a conclusão justifica os resultados empíricos em função do objetivo deste estudo, e apresenta trabalhos futuros.

2. Visão Geral

2.1. Algoritmos paralelos

Os ganhos em desempenho com a implementação paralela de algoritmos seqüenciais e monoprocessados são evidentes. Essa diferença é intuitiva especialmente quando pensamos em sistemas distribuídos, pois têm a capacidade de executar múltiplos processos seqüenciais simultaneamente. Tais sistemas estão baseados em uma arquitetura MIMD (*multiple instruction, multiple data*), isto é, cada unidade de processamento opera uma parte do aplicativo, denominada processo, independentemente do que está ocorrendo nas outras unidades [Jordan e Alaghband 2003].

Multicomputadores e multiprocessadores são desenvolvidos com base na arquitetura MIMD [Foster 1995]. Mais especificamente, o presente estudo tem seu foco nos multicomputadores, ou seja, no modelo de memória distribuída, para o qual cada unidade de processamento tem sua própria memória, e não no modelo de memória compartilhada. Em ambos os modelos, a rede de interconexão é responsável pela conectividade entre os processos e, por isso, sua configuração tem importância fundamental sobre o desempenho do sistema distribuído [Jordan e Alaghband 2003].

Por outro lado, novas fontes de custo são inseridas com a implementação de algoritmos paralelos [Jordan e Alaghband 2003]. Interessam para este estudo, apenas os custos relativos à comunicação que, em última instância, refere-se à granularidade dos algoritmos em execução. Convém definir de forma mais precisa as características dos algoritmos escolhidos assim como o mecanismo de comunicação do sistema em questão.

A granularidade de um algoritmo é explicada como a razão entre o tempo de comunicação sobre o tempo de computação gastos na sua execução [Kumar et al 1994]. Ademais, o grau de paralelismo de um algoritmo é determinado essencialmente pelo grau de dependência de dados entre os processos [Jordan e Alaghband 2003]. O grau de dependência de dados relaciona-se com o algoritmo em si, com suas características qualitativas. Para este estudo, esta característica merece atenção especial, pois um grau de dependência não-nulo significa, sobretudo, necessidade de comunicação.

O algoritmo de multiplicação de matrizes (MM) [Dongarra et al 2007] é um exemplo clássico de granularidade grossa e não apresenta dependência de dados entre os processos, permitindo um alto grau de paralelismo. Já a transposição de matrizes (TM) [Choi et al 1995] exemplifica um algoritmo de granularidade fina pois apresenta alto grau de dependência de dados e, por isso, a ordem de execução deve ser preservada em meio a trocas de informações entre os processos. Portanto, o grau paralelismo de um algoritmo depende do grau de dependência dos dados [Pacheco 1996].

O modelo de memória distribuída do sistema em estudo utiliza-se do mecanismo de passagem de mensagem para a interação entre os processos. Alguns exemplos são: PVM [Geist et al 1996] e MPI [MPI Forum 1994]. No entanto, as bibliotecas baseadas no padrão MPI vêm tomando o lugar das outras implementações, de modo que se afirmou como padrão *de facto*. Além disso, é portável, eficiente e confiável, enquanto o PVM é portável mas não oferece um desempenho ótimo [Dongarra et al 2000].

2.2. Bibliotecas de comunicação MPI

Bibliotecas de comunicação baseadas no padrão MPI (*Message Passing Interface*) fornecem um ambiente de comunicação entre processos para a execução de programas distribuídos. As funções de comunicação operam de forma transparente, podendo ser: ponto-a-ponto ou coletivas [MPI Forum 1994]. Ainda podem ser classificadas como bloqueantes ou não-bloqueantes [Foster 1995].

Existem diversas implementações de bibliotecas de comunicação baseadas no padrão MPI. A MPICH [Gropp et al 1996] é uma iniciativa pioneira do Argonne National Laboratory que já dispõe da implementação MPICH2, baseada no padrão revisado MPI-2 [MPI Forum 1997]. Outro exemplo é o audacioso projeto OpenMPI [Gabriel et al 2004] que tem o objetivo de ser a melhor implementação MPI disponível.

A maioria das implementações apresentam versões para diversas redes de interconexão como Ethernet, Myrinet [Boden et al 1995], e Quadrics [Petrini et al 2002]. Assim também ocorre com a implementação MPICH, que foi escolhida para configurar o sistema porque é uma biblioteca consagrada e preocupada com o fator desempenho. Entretanto, em função de uma incompatibilidade de versão da rede Myrinet à disposição com a implementação MPICH2 foi necessário adotar a implementação MPICH1.

2.3. MPI no ambiente Ethernet: MPICH1

Na implementação MPICH1 para Ethernet, a comunicação MPI se dá por meio de um canal TCP, que é um protocolo orientado à conexão residente no sistema operacional. Ou seja, quando o processo faz um pedido de comunicação para a biblioteca MPICH1, o pedido é repassado ao sistema operacional. [Gropp et al 1996]. Assim, o sistema operacional é chamado à execução sempre que for necessário enviar ou receber pacotes de rede, gastando tempo de processamento com o serviço de comunicação, pois concorre com o programa distribuído para a utilização do processador principal.

Na rede Ethernet, de topologia em barra, o meio físico é compartilhado pelas máquinas conectadas àquele segmento de rede. Assim, quanto mais pedidos de comunicação concomitantes, mais colisões acontecem em nível de acesso ao meio físico e mais retardo é inserido na comunicação [Foster 1995]. Quanto ao MPICH, é importante destacar que as comunicações coletivas são implementadas por definição com base nas comunicações ponto-a-ponto [Gropp et al 1996]. Assim, a vantagem de uma configuração nativamente multi-ponto como a Ethernet não é aproveitada.

2.4. MPI no ambiente Myrinet: MPICH-GM

Diferentemente do MPICH1 para Ethernet, a implementação MPICH-GM favorece a configuração nativamente ponto-a-ponto da rede de interconexão Myrinet, de topologia *mesh* [Boden et al 1995]. Isto significa que há uma conexão física dedicada de cada ponto

para com todos os outros pertencentes àquele segmento de rede e, portanto, não existem problemas de colisão.

Outra diferença dá-se em nível de protocolos de comunicação, pois quando a biblioteca MPICH-GM recebe um pedido de comunicação de um processo, o que acontece é uma chamada direta à biblioteca de comunicação GM. Ela fornece a interface de acesso à rede Myrinet e implementa um serviço de rede não-orientado à conexão e externo ao sistema operacional. Adicionalmente, as placas de rede Myrinet são equipadas com processador e memória próprios, o que as torna capazes de encarregar-se do processamento de pacotes de rede. Portanto, o caminho de comunicação é desviado do sistema operacional e, em sua quase totalidade, dos processador e memória principais da máquina [Boden et al 1995]. Em termos de desempenho, essa abordagem na arquitetura de interconexão possibilita ganhos diferenciados. Na literatura, tal abordagem é conhecida como Virtual Interface Architecture (VIA) [Dunning et al 1998].

2.5. Trabalhos Correlatos

A consolidação do paradigma de passagem de mensagem, consagrado pelo padrão MPI, traz uma preocupação sobre o impacto da utilização de redes mais eficientes na interconexão de agregados computacionais. Qian, Afsahi e Zamani [Qian et al 2004] apresentam um extenso estudo analítico sobre o desempenho da Myrinet com base puramente em funções de comunicação da biblioteca MPICH. Majumber e Rixner [Majumber e Rixner 2004] comparam o desempenho das redes Ethernet e Myrinet sobre três diferentes implementações do padrão MPI, duas para Ethernet e uma para Myrinet. Para tanto, apresentam o estudo com base na análise de funções de comunicação MPI e de um software de *benchmarking* simulando algoritmos de características obscuras.

3. Contribuição: Granularidade da Aplicação X Rede de interconexão

O presente estudo apresenta o impacto do ambiente de interconexão em termos de desempenho visando a formação eficiente de agregados computacionais para aplicações científicas. Para tanto, não basta investigar apenas o comportamento da rede de interconexão ou da biblioteca de comunicação entre processos, pois as aplicações podem ser de naturezas diferentes e definir necessidades diversas. Sendo assim, diferentemente dos trabalhos correlatos, a análise das características dos algoritmos é relevante. Portanto, tendo em vista o objetivo posto, o comportamento de algoritmos com graus de paralelismo distintos, isto é, de granularidades fina e grossa, são observados sobre redes de interconexão de características díspares como a Ethernet e a Myrinet.

3.1. Ambiente Experimental

Os experimentos deste artigo foram realizados em um agregado de cinco computadores: quatro máquinas IBM NetFinity 3500, exercendo a função de escravas; e uma máquina mestre mais potente que as escravas. A máquina mestre foi dotada de mais memória e poder de processamento, como se pode ver na tabela 1. Dessa forma, a caracterização dos ambientes de comunicação não fica limitada às capacidades da máquina mestre, pois o processo mestre executa o algoritmo e também é responsável pelos escravos, segundo o modelo SPMD (*single program, multiple data*). Neste modelo de programação paralela, todos os processos são fontes de um mesmo programa mas operam sobre diferentes porções de dados [Jordan e Alaghband 2003]. Além disso, todas as máquinas operam com

sistema Linux Ubuntu 6.06, baseado no kernel versão 2.6.17-11. Apenas processos básicos do sistema estão operantes, ou seja, as máquinas estão ociosas à espera dos experimentos. Com estas medidas, evitam-se questões relativas ao balanceamento de carga no sistema.

Tabela 1. Configuração do agregado computacional.

	Mestre (1)	**Escravas (4)**
Processador	AMD Athlon XP 2200+ 1.8 GHz	Intel Pentium III 600 MHz
Memória	512 Mb	256 Mb

Este agregado computacional está interconectado por dois meios físicos distintos, isto é, por duas redes de interconexão: Ethernet (placas 100BASE-T) e Myrinet (placas M3S-PCI64B). Em ambos os casos, o agregado dispõe de um segmento dedicado de rede, o que significa que está isolado de ruído proveniente de outros serviços de rede.

Além das diferenças em relação à topologia, Ethernet em barra e Myrinet em *mesh*, a latência e a taxa de transferência foram medidas para ambos os ambientes de interconexão segundo o *round-trip time*. Para tanto, um algoritmo ping-pong foi implementado com base nas bibliotecas MPICH para cada ambiente de interconexão. O tamanho das mensagens utilizadas para estas medições corresponde ao tamanho, em bytes, de uma matriz N x N formada de elementos do tipo de dados inteiro (4 bytes).

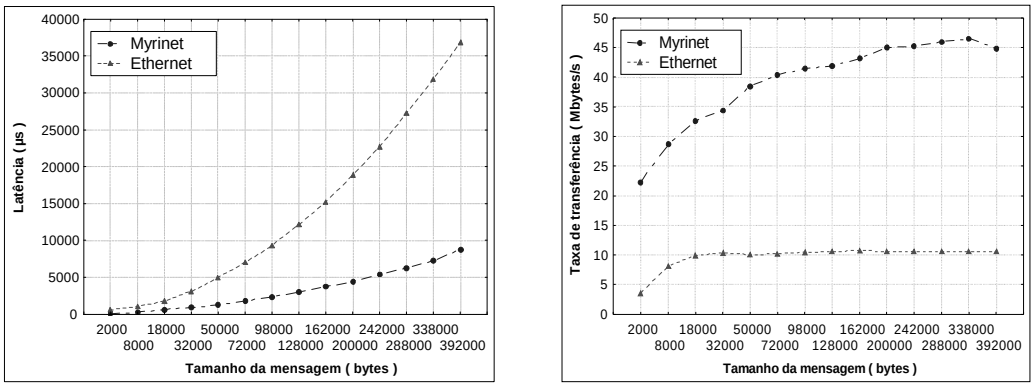

Figura 1. Latência (à esq.) e taxa de transferência (à dir.) para Ethernet e Myrinet.

Segundo a figura 1, é visto que o agregado computacional vale-se de um ambiente de interconexão mais eficiente, o Myrinet, e de outro mais limitado, o Ethernet. Uma diferença crucial entre a Ethernet e a Myrinet reside no modelo de arquitetura adotado, visto que as placas Myrinet dispõem de processador e memória dedicados ao serviço de rede, como foi mencionado anteriormente. Em adição, tomou-se cuidado ao adotar versões equivalentes da implementação MPICH1 para ambos os ambientes.

O ambiente experimental se constitui também em nível de aplicação. Foram adotados dois algoritmos largamente utilizados em aplicações científicas: a multiplicação de matrizes (MM), de granularidade grossa em função do grau nulo de dependência de dados e da alta demanda de processamento; e a transposição de matrizes (TM), de granularidade fina em função do alto grau de dependência de dados, uma vez que caracteriza-se pela redistribuição de elementos da matriz. Optou-se por implementações consagradas no meio acadêmico, publicadas por Peter Pacheco [Pacheco 1996].

3.2. Experimentos

Os experimentos foram executados igualmente para ambos os algoritmos e redes de interconexão em estudo. Em cada ambiente de interconexão, cada algoritmo foi executado 30 vezes para cada tamanho de entrada, a fim de obter uma média mais precisa, amenizando resultados acidentalmente díspares. Ademais, um processo é executado por máquina. Falhas de processo foram desconsideradas pois não ocorreram.

Como resultados empíricos, foram coletados o tempo de execução do algoritmo [Jordan e Alaghband 2003], tomando como base o processo mestre. A média e os valores máximo e mínimo coletados estão indicados. Tomou-se como variável o tamanho da entrada: uma matriz quadrada gerada aleatoriamente, de tamanho N x N, com N variando de 50 em 50, até 700. O limite superior em 700 foi obrigatório, sendo este o valor mais alto suportado pelas máquinas do agregado.

4. Resultados empíricos

O tempo de execução do algoritmo MM em função do tamanho das matrizes a serem multiplicadas está ilustrado na figura 2, à esquerda. Como esperado, o tempo de execução dos processos comunicando-se pela Ethernet é superior ao da Myrinet.

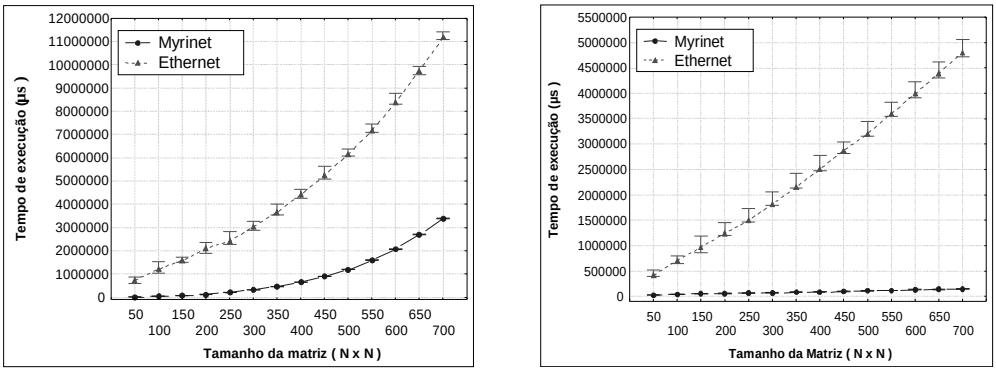

Figura 2. Tempo de execução: MM (esq.) e TM (dir.) em função do crescimento da matriz.

Da mesma forma, à direita na figura 2, tem-se o tempo de execução do algoritmo TM em função da matriz a ser transposta. O resultado esperado é observado pois as execuções sobre o ambiente Myrinet gastaram muito menos tempo do que no ambiente Ethernet. Esta enorme diferença é resultado da alta dependência de dados do algoritmo TM, que demanda uma grande quantidade de comunicações para sua consecução.

Em termos absolutos, os resultados da execução do algoritmo TM sobre a rede Myrinet se distanciam muito mais dos resultados obtidos sobre a Ethernet do que na execução do MM. Ou seja, pela comparação dos gráficos da figura 2, a impressão é de que o ambiente de interconexão Myrinet tem um maior impacto positivo no desempenho do algoritmo TM, de granularidade fina, do que do MM, de granularidade grossa.

Para melhor apresentar os ganhos em desempenho na execução dos algoritmos MM e TM sobre o ambiente Myrinet em relação ao Ethernet, a figura 3 ilustra esta relação em termos destes ganhos (*speedup)*.

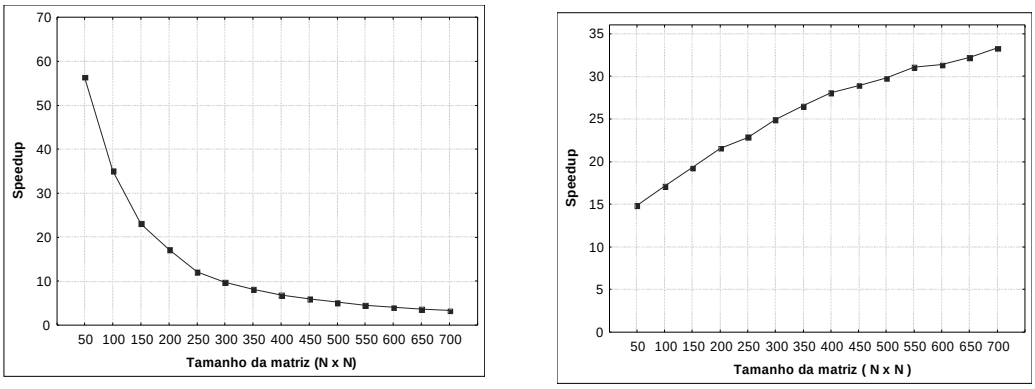

Figura 3. Ganho em desempenho: MM (à esq.) e TM (à dir.) em função do crescimento da matriz de entrada.

Observa-se, à esquerda na figura 3, que o ganho em desempenho diminui com o aumento da matriz de entrada. Isto se deve à granularidade grossa do algoritmo MM. Como não há dependência de dados, a necessidade de comunicação é menor e, por isso, o ganho em função da rede de interconexão tende à equivalência.

Por outro lado, à direita na figura 3, vê-se o efeito contrário, pois o ganho em desempenho relativo ao algoritmo TM, de granularidade fina, mostra-se em crescimento com o aumento da matriz de entrada. Isto leva à observação de que a dependência de dados, ao provocar maior demanda de comunicação entre processos, enaltece as características notadamente superiores da rede Myrinet em detrimento da Ethernet.

Esta última análise vem para assegurar a impressão notada em termos absolutos dos tempos de execução na figura 2. A comparação dos gráficos relativos aos ganhos em desempenho, na figura 3, confirma portanto que, no caso de algoritmos de granularidade fina, a contribuição do ambiente de interconexão para um desempenho positivamente diferenciado é maior do que para algoritmos de granularidade grossa.

5. Conclusões e trabalhos futuros

Com base neste estudo empírico, é possível determinar um impacto positivo da rede de interconexão Myrinet no desempenho das aplicações distribuídas, seja de granularidade fina ou grossa, em função do crescimento dos dados de entrada. Ademais, os ganhos no desempenho de aplicações de granularidade fina crescem expressivamente com o aumento do problema quando interconectados por uma rede mais eficiente como a Myrinet. Por outro lado, o desempenho de aplicações de granularidade grossa mostram certa independência quanto à eficiência da rede de interconexão.

Conseqüentemente, a granularidade da aplicação se mostrou como fator determinante da parcela de contribuição para o desempenho por parte do ambiente de interconexão. Portanto, o impacto da rede de interconexão está em relação direta com a granularidade das aplicações científicas a serem utilizadas no agregado. Ou seja, a determinação do ambiente de interconexão mais adequado à formação eficiente de um agregado computacional deve levar em conta as especificidades das aplicações.

Como trabalho futuro, indica-se a execução destes experimentos sobre outros dispositivos de interconexão de alta taxa de transferência como Quadrics [Petrini et al

2002] e Infiniband [Cassiday 2000]. Considera-se também um estudo aprofundado sobre a variação da relação computação/comunicação com base nos experimentos realizados.

Referências

Boden N., Cohen D., Felderman R., Kulawik A., Seitz C., Seizovic J. e Su W. (1995) "Myrinet: A Gigabit-per-second Local Area Network", IEEE Micro, 15(1):29–36.

Gropp W., Lusk E., Doss N. e Skjellum A. (1996) "A High-performance, Portable Implementation of the MPI Message Passing Interface Standard", Parallel Computing.

Foster I. (1995) "Designing and Building Parallel Programs". http://www-unix.mcs.anl.gov/dbpp/.

Jordan H. e Alaghband G. (2003) "Fundamentals of Parallel Processing". Prentice Hall.

Kumar V., Grama A., Gupta A.,Karypis G. (1994) "Introduction to Parallel Computing". Benjamin/Cummings. http://www-users.cs.umn.edu/~karypis/parbook/.

Pacheco P. (1996) "Parallel Programming with MPI". Morgan Kaufmann.

Geist G., Kohl J. e Papadopoulos P. (1996) "PVM and MPI: A comparison of features". Calculateurs Paralleles, 8(2).

Petrini F., Chun Feng W., Hoisie A., Coll S. e Frachtenberg E. (2002) "The Quadrics Network: High-Performance Clustering Technology". IEEE Micro, 22(1):46–57.

Message Passing Interface Forum (1994) "Special issue: MPI: a message passing interface standard". The International Journal of Supercomputer Applications and High Performance Computing.

Message Passing Interface Forum (1997) "MPI-2: Extensions to the Message-Passing Interface". http://www.mpi-forum.org/docs/mpi-20-html/mpi2-report.html.

Gabriel E., Fagg G., et al (2004) "Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation". 11th European PVM/MPI Users' Group Meeting.

Cassiday D. (2000) "InfiniBand Architecture Tutorial". Hot Chips 12.

Dunning D., Regnier G., McAlpine G., Cameron D., Shubert B., Berry F., Merritt A. M., Gronke E. e Dodd C. (1998) "The Virtual Interface Architecture". IEEE Micro.

Dongarra J., Pineau J., Robert Y., Shi Z., Vivien F. (2007) "Revisiting Matrix Product on Master-Worker Platforms". IEEE IPDPS.

Choi J., Dongarra J. e Walker D. (1995) "Parallel Matrix Transpose Algorithms on Distributed Memory Concurrent Computers". Parallel Computing, 21(9):1387–1405.

Dongarra J., Fagg G., Hempel R. e Walker D. (2000) "Message Passing Software Systems". Encyclopedia of Electrical and Engineering, John Wiley & Sons Inc.

Qian Y., Afsahi A. e Zamani R. (2004) "Myrinet Networks: A Performance Study". IEEE Network Computing and Applications, pp. 323-328.

Majumder S. e Rixner S. (2004) "Comparing Ethernet and Myrinet for MPI communication". Proceedings of the 7th Workshop on Languages, Compilers, and Run-Time Support For Scalable Systems.