

Análise de desempenho e eficiência de um *cluster* de computadores e interconexão de redes utilizando bibliotecas matemáticas

Felippe Vieira Zacarias^{1,2}, Josemar Rodrigues de Souza^{1,2}

¹Centro Integrado de Manufatura e Tecnologia SENAI CIMATEC
Centro de Supercomputação para Inovação Industrial (CSII)
Av. Orlando Gomes 1845, Piatã, 41.650-010
Salvador, Bahia, Brasil

²Universidade do Estado da Bahia (UNEB)
Núcleo de Arquitetura de Computadores e Sistemas Operacionais (ACSO)
Rua Silveira Martins, 2555, Cabula.
Salvador-BA, 41195-001
Salvador, Bahia, Brasil

{fvzacarias, josemarsbr}@gmail.com

Resumo. Medir e avaliar o desempenho dos computadores paralelos é uma grande oportunidade de descobrir falhas e oportunidades de melhoria. Este artigo apresenta uma avaliação de desempenho de um cluster de computadores interconectados sobre uma rede Switched Ethernet fazendo uso da aplicação HPL (High Performance Linpack) e comparando o uso das bibliotecas ATLAS e MKL sobre a perspectiva de desempenho em Gflops. Além disso são utilizados como fatores para aquisição dos resultados a taxa de transferência da rede e versões das bibliotecas compiladas com e sem otimização para a plataforma de hardware, demonstrando que versões otimizadas e redes de maior taxa de transferência geram um uso mais eficiente dos recursos de hardware.

1. Introdução

Com uma necessidade cada vez maior de computação, as atividades científicas e de engenharia requisitam cada vez mais arquiteturas de *software* e *hardware* que reduzam o tempo necessário para a solução de seus problemas que demandam altas capacidades de recursos computacionais. A Computação de alto desempenho ou HPC, surge então desta demanda e da necessidade de substituição da tecnologia de fabricação dos computadores, que atingiu um patamar limitado [Boratto et al. 2012]. Não basta ampliar a frequência de operação de um processador sem aumentar sua capacidade de comunicação com a memória principal ou mesmo com os demais dispositivos com o qual interage. Sob esta ótica, o desenvolvimento de um computador está fisicamente relacionado com seus componentes e a forma de como estão organizados.

Como alternativa atraente encontra-se o *Clustering Computing* [Spector 2000] que podem, a um custo relativamente baixo, garantir o desempenho necessária viabilizando assim a execução das aplicações demandantes. Um *cluster* se caracteriza por ser um conjunto de computadores interconectados através de uma rede de computadores para a troca de informação e de recursos, operando sobre o controle de *softwares* específicos. Esse

software faz com que o conjunto opere como se fosse apenas um único equipamento com capacidade mais elevada, escalabilidade mais elevada e custo proporcional a capacidade computacional [Cuenca et al. 2005].

Uma das razões mais importantes para se medir e avaliar o desempenho dos computadores paralelos é saber se o desempenho atual pode ser melhorado [Zacarias et al. 2013], contudo com a evolução das arquiteturas heterogêneas e *multicores*, torna-se difícil comparar o desempenho desses sistemas especialmente quando análises detalhadas e projeções são desejadas [Kogge and Dysart 2011]. O crescimento acelerado da tecnologia e o aumento da complexidade dos sistemas computacionais fazem da avaliação de desempenho uma questão não muito simples de ser resolvida.

Neste sentido, este trabalho apresenta o resultado de uma análise do impacto da interconexão de rede e da biblioteca matemática no resultado da aplicação de um *benchmark* em um *cluster* de computadores. O trabalho está estruturado da seguinte forma: na seção 2 serão abordados os principais tipos e dispositivos de redes para *cluster* de computadores, na seção 3 será abordada o *benchmark* e as bibliotecas utilizadas na análise, na seção 4 e 5 serão abordados respectivamente os resultados experimentais e as conclusões.

2. Redes de Interconexão

A configuração de um *cluster* pode variar em termos de opções arquiteturais utilizadas: nós computacionais, redes de interconexão e sistemas de armazenamento são as principais delas. Segundo [Navaux and Rose 2011], a rede de interconexão desempenha um papel muito importante influenciando diretamente na eficiência da troca de informações nos multicomputadores. É desejável canais de comunicação com baixa latência e máxima largura de banda para a construção de soluções distribuídas [Coelho 2012], que por sua vez devem atender a requisitos como escalabilidade, facilidade de atualização, alta disponibilidade e custo/benefício.

A rede de interconexão denominada estática é onde existe uma ligação direta e dedicada entre dois componentes e é utilizada, na maioria dos casos, em multicomputadores [Navaux and Rose 2011]. Já as redes em que as conexões são feitas sobre demanda, são denominadas dinâmicas. Máquinas paralelas possuem quase sempre estruturas regulares com ligações homogêneas, mas interligar diretamente todos os processadores torna-se inviável quando o seu número aumenta, desta forma padrões são definidos para a interligação dos mesmos. Neste caso, a topologia determina as características da rede [Navaux and Rose 2011].

2.1. Topologias de Redes

No caso ideal, o padrão de interconexão da topologia corresponde exatamente ao padrão de comunicação da aplicação paralela que executa na máquina [Navaux and Rose 2011], porém não há como prever *a priori* os padrões de comunicação de todas as aplicações que irão executar no ambiente. Neste caso pode-se avaliar a topologia em relação a sua adequação a um algoritmo.

Na topologia em **barramento** os processadores compartilham o mesmo canal de comunicação. É a alternativa de menor custo, porém por se tratar de um canal compartilhado por todas as possíveis conexões, tem baixa tolerância a falhas e é altamente bloqueante [Rose 2001]. Quando um processador necessita comunicar-se com outro, ele aguarda

até que o barramento esteja livre para propagar sua mensagem. Em uma comunicação simultânea, colisões são detectadas e os processadores voltam a tentar a comunicação após um período de tempo determinado aleatoriamente.

A **árvore binária** favorece a execução de algoritmos divisão e conquista (*divide and conquer*) [Navaux and Rose 2011]. Seu diâmetro, maior distância entre dois componentes quaisquer da rede, cresce linear em relação a altura h da árvore e de forma logarítmica em relação ao número de nós [Rose 2001]. Suas desvantagens são em caso de falha de um nó resulta na perda da ligação com toda sua sub-árvore e o gargalo introduzido quando metade da árvore precisa conversar com a outra.

Uma topologia muito utilizada em máquinas paralelas é a **malha bidimensional** [Navaux and Rose 2011]. Nesta topologia, os processadores tem um canal de comunicação direto com o vizinho. Malhas são adequadas aos problemas nos quais uma estrutura de dados bidimensional tem que ser processada de forma particionada [Rose 2001], algoritmos que utilizam cálculos científicos e manipulação de matrizes se destacam. Uma variação utilizada consiste em interligar as extremidades da malha de forma cíclica, formando um **torus bidimensional**.

2.2. Dispositivos de Interconexão de Rede

A maioria dos *clusters* de alto desempenho utilizam soluções SAN (*System Area Network*) [Watanabe et al. 2007]. SAN são redes de conexões de alto desempenho que podem conectar *cluster* de computadores com alta largura de banda e baixa latência. Por outro lado, de acordo com [Coelho 2012], após a popularização dos *cluster beowulf*, a utilização da *Ethernet* em pequenas redes locais tornou-se muito popular, mas esforços na eliminação de gargalos de comunicação e a maximização da velocidade de transmissão estão presentes em tecnologias como Myrinet, infiniband, Quadrics.

O padrão **Ethernet** é a tecnologia mais usada em redes locais e seu preço não é muito elevado. Especificada no padrão IEEE 802.3, provê velocidade de transmissão de 10 Mbit/s. Como todos os computadores de uma rede *ethernet* estão conectados a uma mesma linha de transmissão, eles competem pelo acesso usando o protocolo *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD). Com este protocolo qualquer máquina está autorizada a emitir sobre a linha a qualquer momento, porém cada máquina verifica antes se não existe nenhuma comunicação na linha antes de emitir. No caso de envio simultâneo as duas máquinas interrompem sua comunicação e esperam um prazo aleatório para o reenvio.

Com a necessidade do aumento de desempenho das redes locais surgiram os padrões **Fast Ethernet** que levam o tráfego de dados à taxa nominal de 100 Mbit/s e o padrão **Gigabit Ethernet** cujos dados podem trafegar a taxa de 1 Gigabit/s. Atualmente as instalações *Ethernet* utilizam de *switches* para criar um "canal dedicado" entre o emissor e o receptor. Desta forma não há colisões e o protocolo CSMA/CD, em alguns casos, é usado como controle de fluxo para a prevenção de estouro no uso dos buffers internos do *switch*.

A **Myrinet** é um padrão público aberto, publicado e registrado na ANSI [Dantas 2005]. Essa tecnologia possui uma pequena sobrecarga de protocolo comparado com o *ethernet* padrão. Os enlaces de comunicação entre portas do *switch* e placa de rede *full duplex* alcançam até 2 Gigabits/s. Possui baixa latência, *switches crossbar* com monitoramento para aplicação de alta disponibilidade [Secco 2004] e pode escalar dezenas

de milhares de nós com caminhos alternativos.

A **QsNet**, também conhecida como *Quadrics Network*, consiste de dois grandes blocos de *hardware*: uma interface de rede programável chamada Elan e um *switch* de alto desempenho e baixa latência chamado Elite [Petrini et al. 2001]. Os dispositivos do tipo Elite são interligados em forma de topologia *Fat-tree* alcançando uma possibilidade de interligação de milhares de dispositivos de comutação e grande escalabilidade. A interface de rede Elan4 é composta de três grandes módulos: o processador RISC, o sistema de memória e um subsistema de entrada/ saída, tendo a taxa de transferência da ordem de 900Mbits/s [Dantas 2005].

Segundo [Coelho 2012], **Infiniband** é uma especificação para protocolos de comunicação gerida pelo grupo *Infiniband Trade Association* composto por fornecedores líderes de mercado. Este protocolo fornece uma interface de comunicação com elevadas taxas de transferência e *links* seriais ponto-a-ponto bidirecionais além de suporte a *failover* e *QoS* (*Quality of Service*).

3. Aplicação Utilizada

Um *benchmark* é um programa que é usado para efetuar testes de desempenho do sistema computacional, e tem como objetivo avaliar a influência da arquitetura no desempenho. Um teste de *benchmark* é o processo que consiste na execução de um determinado programa ou carga de trabalho em uma máquina ou sistemas específicos, medindo o desempenho resultante. A aplicação utilizada para os testes foi o *benchmark* HPL, que é a aplicação padrão utilizada pelo projeto TOP500 [Meuer 2013] para ranquear o desempenho dos supercomputadores pelo mundo.

3.1. HPL

O HPL (*High-Performance Linpack*) resolve um sistema de equações lineares denso do tipo $A.x = b$, onde A é uma matriz densa gerada aleatoriamente de dimensão $N \times N$; e onde x e b são vetores de tamanho N . A matriz A é primeiramente fatorada como sendo o produto $A = L.U$, onde L e U representam as matrizes triangulares inferior e superior respectivamente [Dongarra and Whaley 2008].

Esse *benchmark* permite ao usuário dimensionar o tamanho do problema e otimizar o *software* a fim de obter o melhor desempenho possível da máquina. Esse desempenho não reflete o desempenho "completo" de um determinado sistema, como nenhum número nunca pode. Ele, no entanto, reflete o desempenho de um sistema dedicado para resolver um sistema de equações lineares densa. Desde que o problema seja muito regular, o desempenho alcançado é bastante elevado, e o desempenho dá uma boa aproximação do desempenho máximo [Dongarra and Whaley 2008].

O algoritmo usado para resolver o sistema de equações no procedimento de referência devem cumprir a fatoração LU com pivotamento parcial. Em particular, o número de operações para o algoritmo deve ser $(2/3)n^3 + O(n^2)$ em ponto flutuante e precisão dupla [Dongarra and Whaley 2008], onde a fatoração requer $(2/3)n^3$ operações e as duas soluções triangulares totalizando n^2 operações cada, com a fase de fatoração dominando o tempo de computação a medida que N aumenta. O desempenho obtida via HPL é abaixo do desempenho teórica pois deve-se levar em conta aspectos como o algoritmo, o tamanho do problema, a linguagem de alto nível, a implementação, o nível de esforço humano

usado para otimizar o programa, a capacidade do compilador para otimizar, a idade do compilador, o sistema operacional, a arquitetura do computador e as características de *hardware* [Dongarra and Whaley 2008].

3.2. Bibliotecas Matemáticas

O HPL requer a instalação da biblioteca **BLAS** (*Basic Linear Algebra Subprograms*) que é uma coleção de rotinas que proveêm as bases para implementar programas que façam operações com matrizes e vetores. As funcionalidades da biblioteca são divididas em três níveis: no primeiro estão as funcionalidades que executam operações de escalares com escalares, escalares com vetores e vetores com vetores; no nível dois encontram-se as operações entre matrizes e vetores; e o nível três implementa operações de matrizes com matrizes [Milani 2010].

A BLAS é disponibilizada gratuitamente e pode ser livremente incorporada em softwares comerciais. Além disso, existem versões otimizadas para diversas máquinas, ou seja, versões otimizadas dependentes de máquina, desenvolvidas pelas fabricantes e fornecedores de software, como por exemplo o *Intel Math Kernel Library* (MKL) que é a implementação da Intel (R) (trademark) otimizada para processadores de sua arquitetura. Possui implementações sequenciais das rotinas, altamente otimizadas apenas em nível de instruções, e versões multithread. É possível também utilizar a biblioteca ATLAS (*Automatically Tuned Linear Algebra Software*), a qual permite gerar automaticamente uma versão otimizada da BLAS para a arquitetura escolhida [Whaley and Petitet 2005].

4. Resultados Experimentais

4.1. Caracterização do ambiente de execução

Foram realizadas medições do desempenho do *cluster* usando o *benchmark* descrito na seção 3.1 enquanto era variado o tipo de rede (*Fast Ethernet* e *Gigabit Ethernet*), as bibliotecas matemáticas (ATLAS versão 3.10.1 e MKL versão 11.04.183) utilizadas para compilar o *benchmark* e o suporte a *multithread* na compilação. Os testes foram realizados no *cluster* de codinome gabi, constituído de quatro máquinas HP ProLiant DL120 G6, com processador Intel (R) Xeon X3440 2,53 GHZ, placa de rede NetXtreme BCM5723 *Gigabit Ethernet*, 8GB de memória RAM, sistema operacional Linux 3.2.0-23-*generic* X86_64 GNU/Linux Ubuntu 12.04LTS, compilador gcc versão 4.6.3, conectadas através do *switch* CISCO modelo SF 300-24P utilizando cabos par trançado UTP categoria 5e.

Para realizar os testes é necessário que o valor de N , ordem do sistema linear utilizado na aplicação, seja limitado pela quantidade de memória total do sistema. Geralmente 90% de memória configura-se como uma boa escolha de N , pois se o valor de N for muito baixo, não irá resultar em trabalho suficiente para os processadores, o que dará resultados ruins e baixa eficiência. Se o valor de N exceder a capacidade de memória ou não deixar memória suficiente para os outros processos do sistema operacional, ocorrerá *swap* e o desempenho irá cair sensivelmente. Utilizando o *HPL Calculator* [Sindi 2009] pode-se estimar o valor de $N=58494$ para os testes realizados nas quatro máquinas do *cluster*, com a eficiência estimada de aproximadamente 90%.

4.2. Análise dos resultados obtidos

Antes de iniciar os testes na arquitetura apresentada anteriormente, a largura de banda da rede foi medida utilizando o Netperf versão 2.6.0 para saber se a aplicação saturou ou

não os enlaces da rede já que o *benchmark* apresenta alta taxa de comunicação entre os processos. O Netperf é um software que provê a largura de banda a partir do teste entre dois hosts em uma rede, transferindo dados TCP de um cliente para o servidor. Para a rede *Fast Ethernet*, a largura de banda obtida foi de 94 Mb/s, já para os computadores conectados via rede *Gigabit Ethernet* o resultado obtido foi de 941 Mb/s, o que evidencia a sobrecarga de cabeçalhos inseridos na comunicação TCP/IP sobre as tecnologias *Ethernet*.

Para ambas as redes, foram configurados dois cenários: o cenário 1 com o *benchmark* HPL compilado sem suporte a *multithread* e o cenário 2 com o suporte a *multithread*. Na execução do *benchmark* foi analisado o tráfego na interface de rede das máquinas participantes do processamento utilizando a ferramenta ifstat versão 1.1, que é uma ferramenta de monitoramento de interfaces de rede que reúne estatísticas dos contadores internos do *kernel* do sistema operacional. Parte dos dados obtidos desta ferramenta foram utilizados na montagem dos gráficos, além disso todos os testes foram realizados cinco vezes e o melhor e pior resultado foram excluídos da média. Os primeiros testes foram realizados na rede *Fast Ethernet* e os resultados pode ser visto nas figuras 1 e 2.

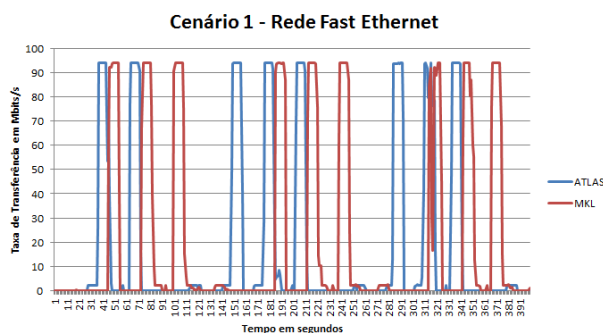


Figura 1. Tráfego na rede durante execução do Benchmark na rede Fast Ethernet no cenário 1.

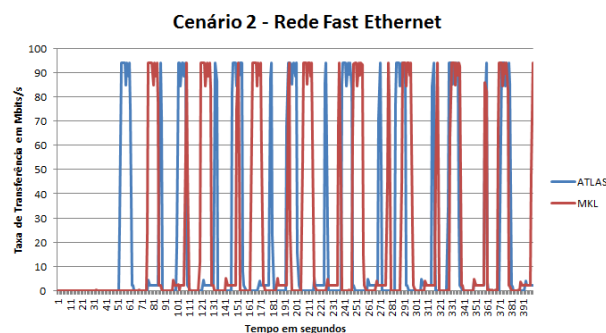


Figura 2. Tráfego na rede durante execução do Benchmark na rede Fast Ethernet no cenário 2.

Pode-se notar que a comunicação no cenário 1 na rede *Fast Ethernet* para ambas bibliotecas satura o enlace de comunicação entre duas máquinas consumindo toda a largura de banda disponível para esta comunicação por um período perceptível. O que ocorre no cenário 2, apesar da saturação do enlace, é uma sensível diminuição do tempo de comunicação entre as máquinas isto devido ao fato de que ao compilar o *benchmark* com suporte a *multithread* diminui-se o número de processos na execução e comunicação, conseqüentemente a divisão do trabalho para um número menor de trabalhadores cria uma carga maior fazendo com que os processos não fiquem ociosos esperando sua comunicação para o envio/recebimento de trabalho junto ao *master*.

De acordo com a tabela 1 o desempenho teórico (*Rpeak*) [Zacarias et al. 2013], que é o pico de desempenho teórico que a máquina consegue atingir representando o limite superior do desempenho, calculado para as quatro máquinas do *cluster* foi de 161,9 Gflops, o resultado obtido nos testes com a rede *Fast Ethernet* atinge apenas 34% do desempenho teórico no melhor cenário (cenário 2 utilizando MKL). Apesar disso o aumento do desempenho obtido com ATLAS no cenário 2 em relação ao cenário 1 foi um pouco maior do que 100%.

Rede de interconexão	Rpeak	Cenário 1 ATLAS	Cenário 1 MKL	Cenário 2 ATLAS	Cenário 2 MKL
Fast Ethernet	161,9	24,9	33,2	51,2	55,2
Gigabit Ethernet	161,9	51,1	114,2	104,9	123,9
Ganho Gigabit/Fast	-	2,05	3,43	2,04	2,24

Tabela 1. Resultado dos testes em Gflops nas redes de interconexão.

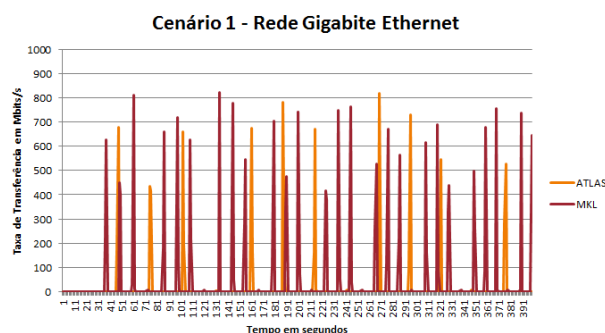


Figura 3. Tráfego na rede durante execução do Benchmark na rede Gigabit Ethernet no cenário 1.

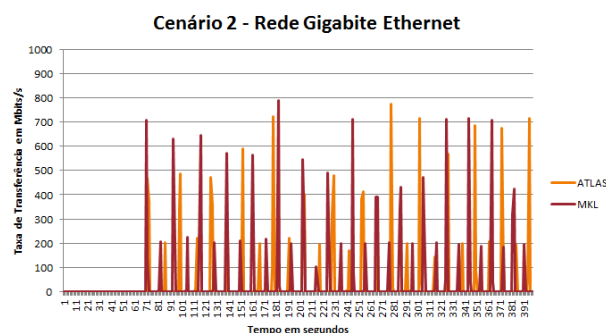


Figura 4. Tráfego na rede durante execução do Benchmark na rede Gigabit Ethernet no cenário 2.

Colocando as máquinas conectadas pela rede *Gigabit Ethernet* e observando a comunicação na execução da aplicação nas figuras 3 e 4, em todos os testes a largura de banda do enlace de comunicação da rede foi suficiente para a aplicação não permitindo a saturação da mesma. É válido observar que no cenário 2 (figura 4) a comunicação não ultrapassou os 800Mbits/s.

Comparando a primeira linha (Rede *Fast Ethernet*) da tabela 1 com a segunda linha (Rede *Gigabit Ethernet*), pode-se notar que o desempenho entre os cenários melhorou muito. Os cenários 1 e 2 usando a biblioteca ATLAS na rede *Gigabit Ethernet* obteve respectivamente um desempenho aproximadamente 2 vezes melhor do que o mesmo cenário de teste na rede *Fast Ethernet*, enquanto que os cenários 1 e 2 utilizando o MKL foi aproximadamente 3,4 e 2,2 vezes melhor.

A diferença de desempenho entre as bibliotecas nos cenário 1 e 2 utilizando a rede *Fast Ethernet* apesar de pequena é perceptível, sendo a diferença entre os cenários 1 mais expressiva que a do cenário 2. Já na rede *Gigabit Ethernet* a diferença entre as bibliotecas torna-se mais evidente já que a interferência do enlace de rede é ultrapassada então a biblioteca que mais exigir do *hardware* expressará os melhores resultados. Neste caso sendo o *cluster* dotado de processadores Intel (R) a biblioteca da fabricante obteve os melhores resultados para qualquer teste em relação a biblioteca ATLAS, além de obter o maior desempenho com aproximadamente 76% do desempenho teórico (Cenário 2 MKL).

É notável também que os cenários 2 obtiveram resultados melhores que o seus respectivos cenários 1. Isso demonstra que aplicações que são executadas em ambientes distribuídos com processadores *multicores* podem se beneficiar do uso do modelo de programação híbrida para alcançar o tão desejado aumento de desempenho de suas aplicações [da Costa and de Souza 2013].

Rede de interconexão	Cenário 1 Ganho MKL/ATLAS	Cenário 2 Ganho MKL/ATLAS
Fast Ethernet	5,1%	2,4%
Gigabit Ethernet	38,9%	11,7%

Tabela 2. Ganho de desempenho ao utilizar a biblioteca MKL em relação a biblioteca ATLAS no mesmo cenário de teste.

5. Conclusão

Este trabalho apresentou uma análise de desempenho e eficiência de um *cluster* de computadores e interconexão de redes utilizando bibliotecas matemáticas em um *benchmark* no *cluster* de computadores situado no centro de supercomputação para inovação industrial no SENAI CIMATEC objetivando o conhecimento de suas potencialidades e fraquezas para propor melhorias em sua arquitetura no futuro e estimular outras pessoas que possuam uma arquitetura semelhante possam fazer novos projetos deste tipo para desenvolver esta área da computação.

Como pode ser observado no artigo, a utilização de bibliotecas otimizadas e uma rede com alta taxa de transferência geram uma utilização da arquitetura de maneira mais eficiente. No caso específico, ao utilizar a rede *Gigabit Ethernet* foi possível até obter um desempenho cerca de 3 vezes maior que o desempenho obtido nas mesmas condições com a rede *Fast Ethernet*. Em relação as bibliotecas matemáticas, ao utilizar a biblioteca MKL foi possível obter um aumento de 38,9% no desempenho em relação a utilização da biblioteca ATLAS no mesmo cenário de teste ,conforme pode ser visto na tabela 2. Em todos os testes o desempenho obtido na rede *Gigabit Ethernet* sempre foi melhor que o desempenho obtido na rede *Fast Ethernet*, bem como o desempenho da biblioteca MKL foi melhor em ambas as redes do que o desempenho da biblioteca ATLAS.

Como trabalhos futuros pretende-se aprofundar a análise na importância de outros elementos no desempenho final da aplicação paralela no *cluster*. Além disso pretende-se analisar também o uso de diferentes paradigmas de programação paralela, diferentes paradigmas de comunicação e diferentes aplicações e seus desempenhos na arquitetura atual.

Referências

- Boratto, M., Coelho, L., and Barreto, M. (2012). Distributed and parallel computing on multicore and multi-gpu systems. *XIII Simpósio em Sistemas Computacionais (WSCAD-SSC)*.
- Coelho, S. A. (2012). Introdução a computação paralela com o openmpi. *Simpósio Mineiro de Computação, 2012, Juiz de Fora. Escola Regional de Informática de Minas Gerais*, pages 24–44.
- Cuenca, J., Garcia, L., Gimenez, D., and Dongarra, J. (2005). Processes distribution of homogeneous parallel linear algebra routines on heterogeneous clusters. *IEEE Computer Society*, pages 1– 10.
- da Costa, A. L. L. and de Souza, J. R. (2013). Apcm: An auto-parallelism computational model. *Journal of Computer Science and Technology, La plata*.

- Dantas, M. (2005). *Computação Distribuída de Alto Desempenho: Redes, Clusters e Grids Computacionais*. Rio de Janeiro, 1ª edition.
- Dongarra, J. and Whaley, R. (2008). Hpl - a portable implementation of the high-performance linpack benchmark for distributed-memory computers. Disponível em: <http://www.netlib.org/benchmark/hpl/>. Version 2.0.
- Kogge, P. M. and Dysart, T. J. (2011). Using the top500 to trace and project technology and architecture trends. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11*, pages 28:1–28:11, New York, NY, USA. ACM.
- Meuer, H. (2013). Top500 supercomputers. Disponível em: http://www.top500.org/project/top500_description/.
- Milani, C. R. (2010). Computação verificada aplicada à resolução de sistemas lineares intervalares densos em arquiteturas multicore. Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2010. [Orientador: Profº Dr. Luiz Gustavo Leão Fernandes, Co Orientadora: Profª Dra. Mariana Luderitz Kolberg].
- Navaux, P. O. A. and Rose, C. A. F. D. (2011). Fundamentos das arquiteturas para processamento paralelo e distribuído. *XI Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul - 2011 - Porto Alegre, RS*, pages 22 – 59.
- Petrini, F., chun Feng, W., Hoisie, A., Coll, S., and Frachtenberg, E. (2001). The quadrics network (qsnet): High-performance clustering technology. In *Proceedings of the 9th IEEE Hot Interconnects*.
- Rose, C. A. F. D. (2001). Arquiteturas paralelas. Minicurso palestrado na 1ª Escola Regional de Alto Desempenho ERAD 2001.
- Secco, F. R. (2004). Uma Família de Abstrações MYRINET para o EPOS. Master's thesis, Federal University of Santa Catarina, Florianópolis. B.Sc. Thesis.
- Sindi, M. (2009). Top500 hpl calculator. Disponível em: <http://hpl-calculator.sourceforge.net>.
- Spector, D. H. (2000). *Building Linux Clusters*. O'Rilley Media, 1 st edition.
- Watanabe, K., Otsuka, T., Tsuchiya, J., Nishi, H., Yamamoto, J., Tanabe, N., Kudoh, T., and Amano, H. (2007). A network interface controller chip for high performance computing with distributed pcs. *IEEE Transactions on Parallel and Distributed Systems*, 18:1282–1295.
- Whaley, R. C. and Petitet, A. (2005). Minimizing development and maintenance costs in supporting persistently optimized blas. *Softw. Pract. Exper.*, 35(2):101–121.
- Zacarias, F. V., de Souza, J. R., Júnior, A. L. M., and de Oliveira, P. M. (2013). Avaliação de desempenho do cluster gabi. *ERBASE 2013, Itabaiana, Sergipe*, 1.