O Impacto da Interconexão de Rede no Desempenho de Programas Paralelos

Anderson M. Maliszewski[1,2], Eduardo Roloff[1], Dalvan Griebler[2,3], Philippe O. A. Navaux[1]

[1]Instituto de Informática, Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, Brasil

[2]Laboratório de Pesquisas Avançadas para Computação em Nuvem (LARCC),
Faculdade Três de Maio (SETREM), Três de Maio, Brasil

[3]Escola Politécnica, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS),
Porto Alegre, Brasil

{ammaliszewski, eroloff, navaux}@inf.ufrgs.br

dalvan.griebler@acad.pucrs.br

Abstract. *The performance of parallel applications depends on two primary components of the environment; processing power and network interconnection. In this work, the impact of a high performance interconnect in parallel programs on a homogeneous cluster of servers interconnected by Gigabit Ethernet 1 Gbps and InfiniBand FDR 56 Gbps was evaluated. Therefore, a characterization of NAS Parallel Benchmarks concerning computation, communication, and execution cost using Microsoft Azure instance pricing model was performed. The results showed that in highly network-dependent applications, performance could be significantly improved by using InfiniBand at a better execution cost, even with the higher instance price.*

Resumo. *O desempenho de aplicações paralelas depende de dois componentes principais do ambiente; o poder de processamento e a interconexão de rede. Neste trabalho, foi avaliado o impacto de uma interconexão de alto desempenho em programas paralelos em um cluster homogêneo de servidores interconectados por Gigabit Ethernet 1 Gbps e InfiniBand FDR 56 Gbps. Foi realizada uma caracterização do NAS Parallel Benchmarks em relação à computação, comunicação e custo de execução em instâncias da Microsoft Azure. Os resultados mostraram que, em aplicações altamente dependentes de rede, o desempenho pode ser significativamente melhorado ao utilizar InfiniBand a um custo de execução melhor, mesmo com o preço superior da instância.*

1. Introdução

No atual cenário da computação, uma crescente demanda tem sido vista por poder computacional em aplicações de computação científica. Para preencher essa lacuna, a computação de alto desempenho (*High Performance Computing* - HPC) oferecida por *clusters* de servidores e pelos modelos *"as a Service"* da computação em nuvem continuam sendo muito importantes. Como esses sistemas costumam usar *Message Passing Interface* (MPI) como padrão para desenvolvimento de aplicações paralelas, e as características de comunicação dessas aplicações variam conforme sua finalidade específica, a

rede tem uma forte influência sobre o desempenho geral. Assim, a interconexão deve ser capaz de garantir uma alta largura de banda de comunicação, bem como baixa latência, para lidar com os requisitos das aplicações e não tornar-se o gargalo de todo o sistema [Escudero-Sahuquillo et al. 2015, Kamburugamuve et al. 2017].

Muitos avanços, bem como novos padrões de rede, foram criados para atender a esses requisitos [Maliszewski et al. 2019]. No entanto, a interconexão de rede ainda está diretamente relacionada a gargalos no desempenho de aplicações paralelas que executam em sistemas distribuídos [Roloff et al. 2017, Moura and Hutchison 2016]. Consequentemente, para evitar perdas de desempenho, a interface de comunicação de tais sistemas foi redesenhada para permitir uma experiência de alto desempenho, notavelmente o InfiniBand (IB), amplamente adotado pela comunidade HPC, e o Omni-Path, relativamente novo em comparação com o InfiniBand. Além disso, mesmo com sua disparidade de desempenho entre os sistemas HPC, as redes Gigabit Ethernet (ETH) ainda são utilizadas em uma porcentagem não desprezível da maioria dos *clusters* HPC básicos.

Visualizando as interconexões utilizadas em *clusters* inseridos na lista Top 500[1] (Figura 1) percebe-se que 123 dos 500 supercomputadores usam IB como interconexão primária, sendo superados apenas pela família de interconexão Ethernet (por exemplo, 1GbE, 10GbE) com 272. No entanto, quando a lista Top 500 é vista apenas pelos seus 100 primeiros sistemas, o IB tem uma presença mais significativa que a Ethernet, representando 43% do total de interconexões utilizadas, sugerindo que suas características de desempenho o tornam mais adequado para grandes instalações de supercomputadores. Assim, neste trabalho, as interconexões InfiniBand e Gigabit Ethernet foram avaliadas usando o mesmo *cluster* físico de servidores, executando os *benchmarks* paralelos do conjunto NPB. Foi realizado o rastreamento das aplicações para avaliar suas características de computação e comunicação e posteriormente comparadas com o tempo de execução. Usando esses dados, as aplicações foram agrupadas em quatro grupos, começando com as altamente dependentes de rede até o grupo sem dependência de rede. Além disso, foi apresentada a métrica de custo de execução usando o modelo de precificação da instâncias da Microsoft Azure.

O restante deste artigo está organizado da seguinte forma: A Seção 2 apresenta a metodologia, com detalhes de *hardware* e *software*, bem como a especificação do *benchmark* e as medidas de rastreamento. A Seção 3 apresenta os resultados de desempenho em relação ao rastreamento, tempo de execução e custo de execução das aplicações com diferentes interconexões de rede. Seção 4 mostra os trabalhos relacionados. No final, na Seção 5 são apresentadas as conclusões e planos futuros.

2. Metodologia

Esta seção descreve a configuração experimental em relação aos detalhes de *hardware/software*, juntamente com os detalhes de rastreamento usados na avaliação.

2.1. Especificações do *Cluster*

Os experimentos foram realizados com dois nós idênticos, cada qual com pico teórico de desempenho de 0.7 TFlop e composto por dois processadores Intel®Xeon®E5-2650 v3

[1] `https://www.top500.org/`

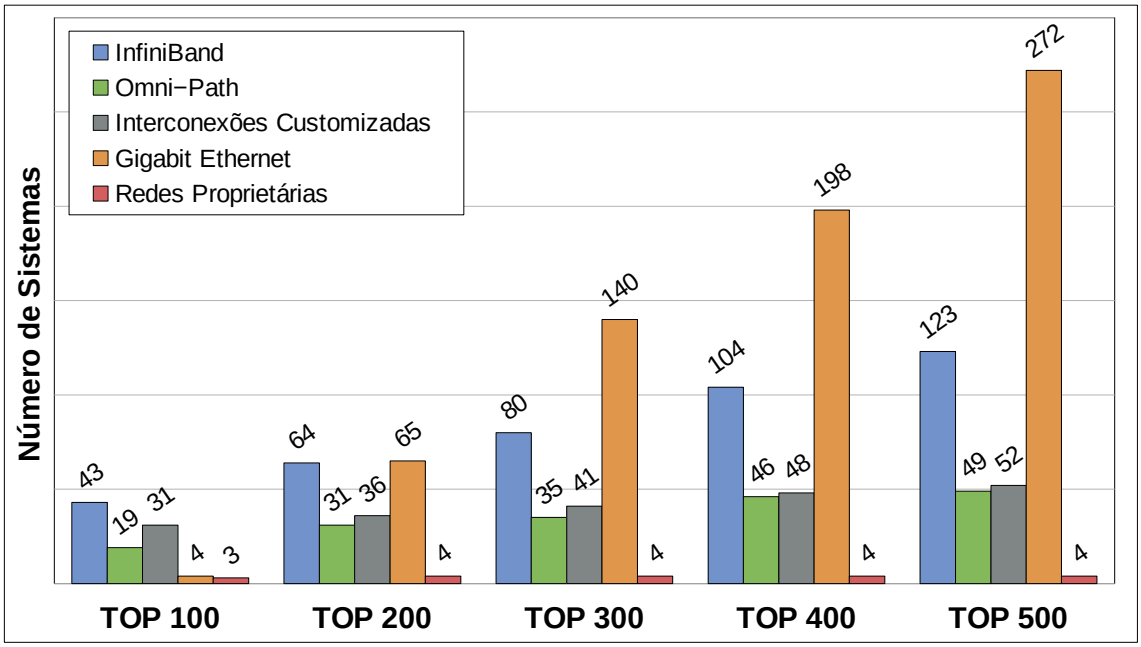

Figura 1. Família de interconexões de rede usadas pelos sistemas do Top 500. Dados da última avaliação realizada em julho de 2019. Atualizado de Zahid [Zahid 2017].

(Q3'14) Haswell 2,3 GHz, 20 núcleos (10 por CPU) com *Hyper-Threading* habilitado resultando em 40 *threads* e 128 GB de memória RAM DDR4. Cada núcleo tem caches L1 (32KB de instrução e 32KB de dados) e L2 (256KB). A cache L3 (256 MB) é compartilhada entre todos os núcleos. Além disso, cada nó tem uma Mellanox MT27600 *Channel Adapter* (CA) configurada para o InfiniBand 56 Gb/s 4X FDR ConnectX-3 com a versão de *firmware* 10.16.1038. Os nós são conectados por meio de um comutador Mellanox SX6036 FDR e um comutador genérico de 1 Gbps. As especificações de *software* tem o Ubuntu Server 18.04 64-bit (kernel 4.15.0-48) como o sistema operacional (SO), Open MPI 2.1.1, compilador GCC/GNU Fortran 7.4.0 e OFED 4.6-1.0.1.1.

2.2. Especificações do *Benchmark*

A avaliação foi feita usando a implementação paralela MPI do conhecido conjunto de *benchmarks* de simulação aerodinâmica numérica NPB [Bailey et al. 1991], recentemente atualizado para a versão 3.4. O NPB foi projetado para avaliar o desempenho de diferentes *hardwares* e *softwares* em sistemas de computação paralela. Foram utilizadas todas as aplicações do conjunto original de *benchmarks* nesta avaliação, que por sua vez é composto por cinco *kernels* (IS, EP, CG, FT e MG) e três pseudo-aplicações (BT, SP e LU).

Como a quantidade computacional de recursos cresceu exponencialmente, o NPB também aumentou a entrada de dados em seus *benchmarks*, que em sua própria terminologia são conhecidos como classes. Neste artigo, foram utilizadas as classes A, B, C (consideradas tamanhos de problema padrão que aumentam 4X a entrada de dados de uma classe para outra), e D (tamanho de problema grande que aumenta a entrada de dados em 16X), assim como a *flag* `-O3`, `mpifort` e `mpicc` para compilar os códigos Fortran e C, respectivamente. Os experimentos com NPB foram repetidos 30 vezes no *cluster* físico em máquinas idênticas (descritas na Seção 2.1) e apresentam o desempenho

em tempo de execução médio com o desvio padrão relacionado. Todas as experiências utilizaram 64 processos devido a requisitos específicos de algumas aplicações do NPB (por exemplo, o número de processadores deve ser uma raiz quadrada ou uma potência de dois). Na Tabela 1 é fornecida uma breve descrição dos *benchmarks* NPB, com seu foco e respectiva linguagem na qual foi escrito.

Nome	Descrição	Foco	Linguagem
BT	Bloco Tridiagonal	Ponto Flutuante	Fortran
CG	Gradiente Conjugado	Comunicação Irregular	Fortran
EP	Desordenamento Paralelo	Ponto Flutuante	Fortran
FT	Transformação de Fourier	Equações Diferenciais	Fortran
IS	Ordenação de Inteiros	Valores Inteiros	C
LU	Cálculo Triangular	Comunicação Regular	Fortran
MG	Multigrid	Comunicação Regular	Fortran
SP	Pentadiagonal Escalar	Ponto Flutuante	Fortran

Tabela 1. Visão geral dos *benchmarks* do NPB usados na avaliação.

2.3. Procedimento de Rastreamento

Para rastrear as aplicações e expor o comportamento de comunicação MPI, foi utilizado a versão 3.0 do *software* Score-P[2] que é responsável pela introdução e instrumentação de código nos *benchmarks* NPB-MPI durante a compilação em uma execução isolada. Após essa instrumentação, para garantir que os eventos sejam rastreados, a variável de ambiente `export SCOREP_ENABLE_TRACING=true` foi setada em ambos os nós e, em seguida, a execução dos *benchmarks* é feita normalmente. Ao final desta etapa, os rastreamentos são criados no formato OTF2[3] (*Open Trace Format Version* 2) e precisam ser convertidos para o formato Pajé. Para converter os traços, foram utilizadas a ferramenta Akypuera[4], mais especificamente a ferramenta `otf22paje`[5]. Depois de concluir a conversão, obtém-se um arquivo com formato .rastro, que para esta avaliação ainda precisa ser convertido em um .csv com a ferramenta `pj_dump`[6]. Por fim, o arquivo .csv contendo todas as informações de rastreamento foi analisado na linguagem estatística R.

3. Resultados

Os resultados foram classificados em três vertentes. A primeira são as medidas de rastreamento das aplicações NPB-MPI, nos quais seus padrões MPI são expostos. A segunda mostra a avaliação de desempenho das mesmas aplicações usando diferentes tecnologias de interconexão, nas quais seu tempo de execução é mostrado. A última é composta pelo cálculo do custo de execução no qual foi usado o modelo de precificação de instâncias da Microsoft Azure em comparação com os resultados obtidos no *cluster* físico.

[2] `https://www.vi-hps.org/projects/score-p/`
[3] `https://silc.zih.tu-dresden.de/otf2-current/html/`
[4] `https://github.com/schnorr/akypuera`
[5] `https://github.com/schnorr/akypuera/wiki/OTF2WithAkypuera`
[6] `https://github.com/schnorr/pajeng/wiki/pj\_dump`

3.1. Medidas de Rastreamento

O rastreamento da execução de todas as aplicações do conjunto NPB foi realizado utilizando as ferramentas e metodologia explicadas anteriormente. Usando as informações do rastreamento, as aplicações foram agrupadas de acordo com seu comportamento em quatro grupos diferentes, que estão representadas na Figura 2 usando uma aplicação de cada grupo. Foram plotados o percentual das três operações mais representativas, sendo elas a computação, comunicação (composta por operações de comunicação MPI) e inicialização (operação `MPI_Init`, que como próprio nome relata, inicializa o ambiente de execução e é principalmente representado em aplicações com pequeno tempo de execução) nas classes de entrada de dados A, B, C e D.

O primeiro grupo é composto por aplicações altamente dependentes da rede, nas quais estão incluídos o IS e o FT. Conforme mostrado na Figura 2, tem-se o perfil do IS e sua porcentagem das três operações representativas explicadas anteriormente. No IS, a operação de inicialização é vista como representativa nas três primeiras classes, principalmente porque essa aplicação possui um tempo de execução rápido. No entanto, na classe D, o IS tem um aumento no tempo de execução causado pelo aumento de entrada de 16X comparado à classe C, e a operação de inicialização mostra um percentual não representativo do tempo de execução geral. A operação de cálculo diminui até a classe C e tem um leve aumento na classe D. Esse é um fator determinante, no qual, se existe um aumento de entrada nesse grupo de aplicações, não necessariamente aumenta a porcentagem de cálculo. Por outro lado, a operação de comunicação aumenta sua proporção juntamente com a classe de entrada de dados. Esse comportamento corrobora a ideia de caracterizar essa aplicação como altamente dependente da rede.

O segundo grupo ilustra o comportamento de aplicações dependentes da rede. No perfil, foi identificado que o CG e MG compartilham esse padrão. Na Figura 2(b), descreve-se o perfil do CG e, como pode ser visto, a operação de inicialização é representativa apenas na entrada de classe A onde a aplicação tem uma execução rápida (menos de um segundo). Além disso, esse tempo de execução rápido também reflete na porcentagem das operações de comunicação e computação dessa classe. A comunicação mostra uma alta porcentagem nas classes B e C e é perceptível que esta começa a diminuir. A operação de computação, por sua vez, começa a aumentar desde a classe B e fica em cerca de 30% na classe D. Com essa informação de perfil, infere-se que neste grupo de aplicações a interconexão de rede ainda exerce influência significativa no desempenho geral.

O terceiro grupo é composto pelas aplicações BT, SP e LU, as quais, de acordo com o perfil criado, são classificadas como de baixa dependência de rede. Para representar o comportamento desse grupo, tem-se o perfil do BT mostrado na Figura 2(c). Como pode ser visto, o BT possui um padrão linear nas operação de comunicação e computação. A operação de inicialização só pode ser vista nas classes A e B, não sendo representativa nas classes maiores, como C e D. A operação de comunicação começa a diminuir desde a classe A até a classe D, ao contrário da computação, que aumenta desde a classe A até a D. Esse comportamento enfatiza que a interconexão de rede não influencia significativamente se entradas enormes forem usadas, por exemplo.

O último grupo é composto apenas pela aplicação EP, descrita como não dependente da rede. Esse comportamento pode ser visto na Figura 2(d), no qual a computação domina todas as classes de entrada. A operação de inicialização é vista principalmente

em entradas mais baixas, porque o EP teve um tempo de execução rápido, representado por menos de um segundo nas classes A, B e C (detalhes podem ser vistos na Tabela 2). Também é perceptível uma pequena porcentagem de comunicação, na qual o EP executa principalmente as rotinas `MPI_Bcast` e `MPI_Allreduce`, para transmitir a mensagem do processo com *rank "root"* para os demais, e combinar esses valores, distribuindo o resultado, respectivamente.

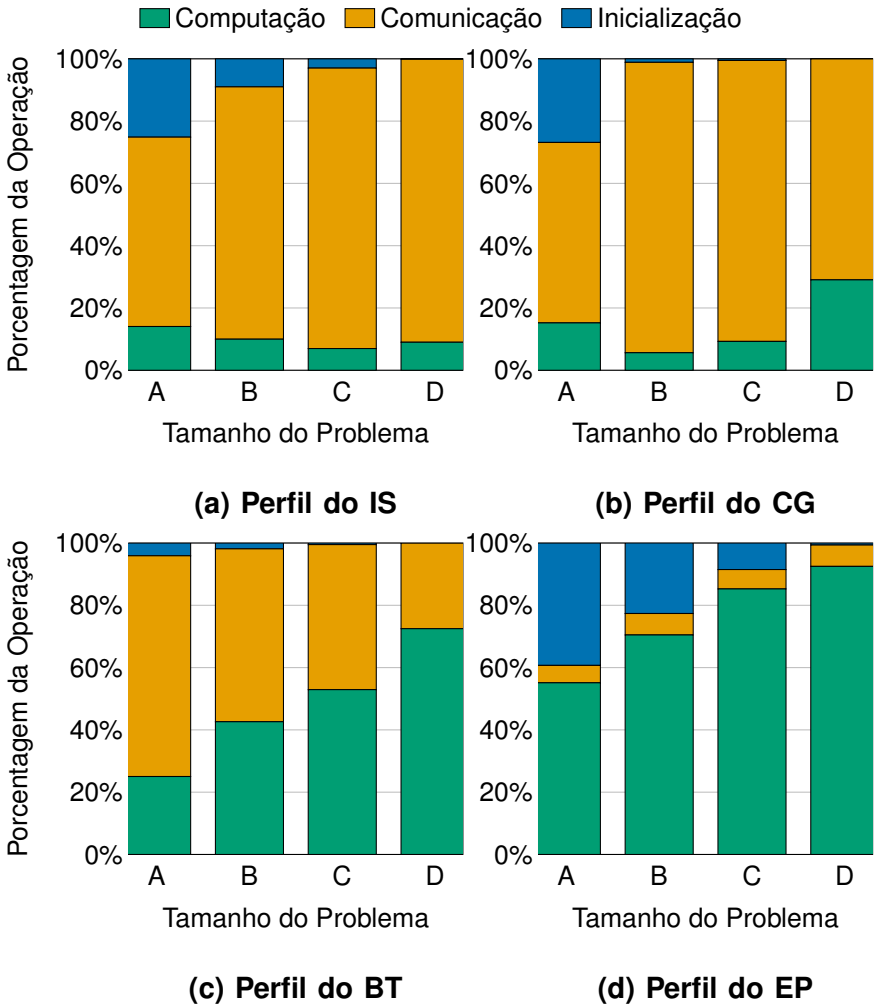

Figura 2. Perfil de quatro aplicações representando o comportamento dos quatro grupos: Altamente dependentes de rede, dependentes de rede, baixa dependência de rede, e não dependente de rede.

3.2. Avaliação do Desempenho

Os resultados de avaliação de desempenho são apresentados na Tabela 2 na qual as aplicações do NPB (previamente descritos no perfil) foram agrupadas. No primeiro estão inclusos o IS e o FT. Essas aplicações têm sua execução baseada nas rotinas `MPI_Alltoallv` e `MPI_Alltoall` (envia dados de todos para todos os processos), respectivamente, e o tempo de computação não aumenta significativamente à medida que a entrada da classe aumenta. Como pode ser visto, na entrada de classe A, o resultado não pode ser considerado representativo devido ao seu baixo tempo de execução, que

representa uma pequena carga no sistema. Na classe B, IB teve um desempenho 32% e 418% melhor que ETH, nos *benchmarks* IS e FT, respectivamente. Quando a entrada aumenta para a classe C, essa porcentagem aumenta para 301% e 648% pelas mesmas aplicações. Finalmente, quando essas aplicações são executados com a classe D, obtêm uma diferença de desempenho representada em 936% no IS e 566% no FT. Assim, neste tipo de aplicações, uma interconexão com baixa latência e alto desempenho é essencial e tem uma influência extrema em seu desempenho.

O grupo de aplicações dependentes de rede é composto por CG e MG. Essas aplicações tem em comum o padrão de uso das rotinas `MPI_Send` (executa um bloqueio de envio) e `MPI_Wait` (espera por um pedido MPI para completar). Como pode ser visto na Tabela 2, CG e MG melhoraram seu desempenho principalmente nas menores entradas. Novamente, na entrada de classe A, as aplicações têm um tempo de execução baixo (menos de 1 segundo para InfiniBand e Ethernet), portanto, não podem ser considerados como representativas. Por outro lado, na classe B, o tempo de execução de ambas as aplicações diferem significativamente, sendo o IB com desempenho de 939% no CG e 204% no MG melhor que no ETH. Quando aumenta-se a entrada para a classe C, essa diferença diminui para 489% no CG e 140% no MG. Finalmente, na classe D, a diferença entre InfiniBand e Ethernet cai para 140% e 45%, em CG e MG, respectivamente. Assim, embora os resultados tenham mostrado na classe A, em que a entrada de dados não foi suficiente para estressar o *hardware* significativamente, a influência da interconexão de rede é vista nas entradas maiores, enfatizando sua importância no desempenho.

O terceiro grupo é composto de aplicações com baixa dependência de rede, nas quais se encontram o BT, LU e SP. BT e SP têm padrões MPI semelhantes, representados pelas rotinas `MPI_Wait` (espera por um pedido MPI) e `MPI_Waitall` (espera por todas as solicitações MPI), enquanto que no LU, a rotina MPI mais representativa é `MPI_Recv` (bloqueio de recebimento para uma mensagem). Por outro lado, todos eles compartilham o mesmo padrão de execução, quase linear, no qual a comunicação diminui à medida que a entrada de dados e a computação aumentam. Assim, em todas as classes, percebe-se que a diferença entre IB e ETH também diminui linearmente. Esse padrão sugere que, se tivermos uma entrada muito grande nessas aplicações, uma rede mais rápida não poderá influenciar significativamente o desempenho, principalmente porque a comunicação é reduzida pela metade e a computação dobra quando aumenta-se a entrada de dados.

Por fim, o quarto grupo não tem dependência de rede e é composto apenas pela aplicação EP. Como é sugerido em seu nome (Embaraçosamente Paralelo), o EP quase não tem comunicação, e sua execução é dominada pela computação. Observa-se na Tabela 2, que a diferença no tempo de execução entre InfiniBand e Gigabit Ethernet é insignificante. Assim, para este tipo de aplicação, uma rede mais rápida não influencia significativamente em seu desempenho.

Além disso, para uma melhor visualização dos ganhos de desempenho em porcentagem na comparação entre InfiniBand e Gigabit Ethernet, plotou-se uma aplicação de cada grupo na Figura 3. Como pode ser visto, o FT melhora seu desempenho usando IB à medida que a entrada aumenta (altamente dependente da rede). O CG também melhora seu desempenho usando uma interconexão mais rápida, mas quando a entrada aumenta, o ganho diminui (dependente da rede). Por outro lado, o SP tem um declínio contínuo no ganho utilizando IB, sugerindo que com entradas maiores a interconexão não influenciará

Nome	Classe A			Classe B			Classe C			Classe D		
	IB	ETH	%	IB	ETH	%	IB	ETH	%	IB	ETH	%
IS	0,06	0,80	1293,60%	2,53	3,37	32,80%	3,07	12,32	301,18%	19,32	200,21	936,46%
FT	2,85	2,42	-15,04%	5,16	26,76	418,47%	14,40	107,77	648,62%	326,49	2176,72	566,71%
CG	0,11	0,77	604,27%	3,08	32,04	939,47%	11,68	68,78	489,09%	497,93	1197,15	140,43%
MG	0,08	0,22	186,64%	0,35	1,06	204,21%	2,59	6,24	140,60%	68,97	100,29	45,41%
BT	2,38	7,43	212,09%	9,67	17,97	85,78%	40,23	58,58	45,61%	847,54	919,78	8,52%
SP	2,02	11,07	446,85%	8,06	29,06	260,29%	38,36	81,11	111,47%	1144,77	1296,45	13,25%
LU	1,48	2,62	77,11%	6,40	8,71	36,11%	25,12	29,02	15,49%	470,76	483,84	2,78%
EP	0,31	0,29	-3,92%	1,18	1,16	-1,50%	4,66	4,66	0,04%	75,02	75,27	0,34%

Tabela 2. Tempo de execução (média em segundos) e porcentagem de diferença entre IB e ETH nas classes A, B, C e D do conjunto de *benchmarks* NPB executados com 64 processos (números positivos nas porcentagens mostram que o InfiniBand é melhor e negativos que o Gigabit Ethernet).

significativamente o desempenho (baixa dependência de rede). EP do último grupo não mostra qualquer diferença entre as interconexões, principalmente porque é CPU-*Bound*.

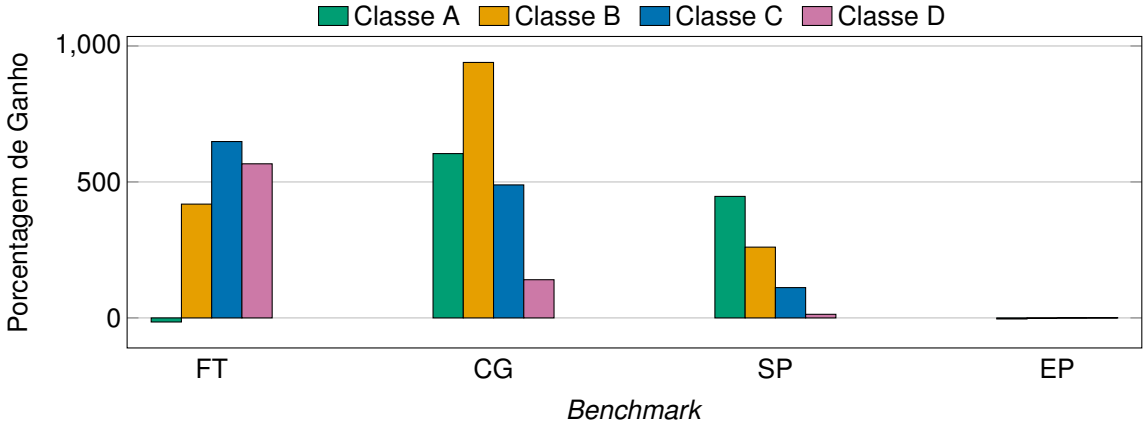

Figura 3. Percentual de ganho de desempenho da interconexão InfiniBand comparado com Ethernet.

3.3. Custo de Execução

Os testes foram executados usando um *cluster* físico sem custos de alocação. No entanto, sabendo que o custo operacional é um dos problemas mais relevantes para a computação em nuvem, estimou-se o cenário mais próximo ao mostrado neste trabalho, disponível na nuvem pública. O provedor Microsoft Azure foi utilizado como parâmetro porque possui as instâncias A10 e A8, com as mesmas especificações de *hardware*, (8 núcleos e 56 Gigabytes de memória), sendo a única diferença entre a interconexão de rede da instância, na qual a A8 é baseada no InfiniBand e a instância A10 usa Ethernet. Como esse é o cenário utilizado nos experimentos, usou-se os custos dessas instâncias para analisar o custo de execução.

O custo por hora de uma instância A8 é US$ 0,975 e o custo por hora de uma instância A10 é US$ 0,78. Os experimentos foram conduzidos usando dois nós, então tem-se o custo total dos nós InfiniBand como US$ 1,95 por hora e US$ 0,0325 por minuto, e os nós Ethernet custam US$ 1,56 por hora e US$ 0,026 por minuto. Para obter o custo de execução, foram multiplicados o tempo de execução pelo custo de execução; é necessário normalizar os valores, então o tempo de execução foi calculado em minutos.

Nome	IB Tempo. Exec	IB Custo. Exec	ETH Tempo. Exec	ETH Custo. Exec	Diferença	%
IS	0,32	US$ 0,01	3,33	US$ 0,08	US$ 0,07	700%
FT	5,44	US$ 0,17	36,27	US$ 0,94	US$ 0,77	452,94%
CG	8,29	US$ 0,26	19,95	US$ 0,51	US$ 0,25	96,15%
MG	1,14	US$ 0,03	1,67	US$ 0,04	US$ 0,01	33,33%
BT	14,12	US$ 0,45	15,32	US$ 0,39	US$ -0,06	-13,33%
SP	19,07	US$ 0,61	21,60	US$ 0,56	US$ -0,05	-8,20%
LU	7,84	US$ 0,25	8,06	US$ 0,20	US$ -0,05	-20%
EP	1,25	US$ 0,04	1,25	US$ 0,03	US$ -0,01	-25%

Tabela 3. Tempo de execução normalizado (em minutos), custo de execução e percentual de diferença (números positivos mostram que o InfiniBand foi mais barato, do contrário, que o Gigabit Ethernet foi mais barato) do conjunto de *benchmarks* NPB classe D.

Na Tabela 3 são mostrados os resultados do custo de execução juntamente com o respectivo tempo de execução, diferença de custo por minuto e porcentagem de diferença para cada aplicação da classe D apenas, porque seu tamanho é mais aproximado das aplicações HPC em execução nos ambiente de produção. Como pode ser visto, mesmo considerando que a instância A8 com IB é 25% mais cara que a instância A10 com Ethernet, as aplicações com maior dependência na rede mostraram menores custos de execução. Esse é o caso de IS, FT e CG que obtiveram os melhores resultados no InfiniBand com custo de execução menor que o da Ethernet em cerca de 700%, 452% e 96%, respectivamente, principalmente devido a diferença no tempo de execução. IS tem uma menor diferença na execução de custos, representada em aproximadamente US$ 0,07. Por outro lado, o FT mostra a maior diferença com US$ 0,77. CG mostra a segunda melhor diferença em cerca de US$ 0,25 por minuto.

MG é a última aplicação com melhor custo de execução no InfiniBand, porém, a diferença é menor, em torno de US$ 0,01. O BT é o *benchmark* com maior diferença de custo, neste caso negativa (US$ -0,06), sendo portanto não financeiramente viável executar na instância com interface IB. SP, LU e EP também tem um melhor custo de execução na instância com interface Ethernet. SP e LU apresentam um desempenho melhor em tempo de execução ao usar a interface IB. No entanto, como a instância InfiniBand é mais custosa que a instância Ethernet, seu custo de execução é melhor na Ethernet. A aplicação EP tem o mesmo desempenho em ambos os ambientes e consequentemente pior resultado no custo de execução no Infiniband em comparação (-25%) com o Ethernet. Dessa forma, percebe-se que para a execução da aplicação ser financeiramente viável na instância com InfiniBand é necessário que haja um desempenho consideravelmente superior no tempo de execução da aplicação.

4. Trabalhos Relacionados

Como o InfiniBand é a interconexão *de facto* adotada para HPC, muitos trabalhos abordaram e a caracterizaram na literatura. Portanto, considerou-se como trabalhos relacionados aqueles que realizam avaliações com InfiniBand e outras interconexões em *clusters bare-metal*/físico, bem como em ambientes virtualizados/nuvens. Por exemplo, Vienne et al., [Vienne et al. 2012] fez uma avaliação abrangente de várias interconexões de alto desempenho, incluindo 10/40 GbE, InfiniBand 32 Gbps *Quad Data Rate* (QDR), InfiniBand 54 Gbps *Fourteen Data Rate* (FDR) e 10/40 GigE *RDMA over Converged Ethernet* (RoCE).

Os experimentos foram conduzidos em ambiente HPC e computação em nuvem, usando o conjunto de *benchmarks* NPB, assim como o TestDFSIO e HBase para visualizar o impacto das interconexões sobre o desempenho de HPC, HDFS básico e *benchmarks* de computação em nuvem.

Por outro lado, Liu et al. [Liu et al. 2004] avaliou o MPI sobre o InfiniBand propondo um novo design para obter melhor escalabilidade, explorando o padrão de comunicação de aplicações usando uma implementação baseada em RDMA com o MVAPICH, que beneficia não apenas mensagens grandes, mas também pequenas e de controle. Eles fornecem medidas de latência e largura de banda. Além disso, para corroborar sua implementação, eles usaram os *benchmarks* do conjunto NPB com classes A e B. Alles et al. [Alles et al. 2018] avaliaram e compararam o desempenho de contêineres Docker e Singularity, tendo como *baseline* o ambiente nativo, para execução de aplicações HPC. O trabalho foi motivado no estudo das desvantagens e melhorias que ocorrem usando técnicas de virtualização baseada em contêiner. Foram utilizadas as aplicações EP com classe B do conjunto de *benchmarks* NPB, o simulador de terremotos Ondes3D e uma aplicação MPI que realiza Ping-Pong para avaliar a latência de redes. Os experimentos foram executados com até 256 núcleos em até 64 nós, com 4 processos MPI por nó usando a interconexão 1 Gb no *cluster* Graphene no Grid5000.

Outros trabalhos concentraram sua avaliação em experimentos usando gerenciadores de nuvens IaaS (*Infraestructure as a Service*) e virtualização baseada em *hypervisor* e contêineres. Por exemplo, Ruivo et al. [Ruivo et al. 2014] integrou o OpenNebula com o InfiniBand usando o *Single-root input/output virtualization* (SR-IOV) e máquinas virtuais baseadas em Kernel (KVM). Além disso, sua abordagem inclui avaliações sobre o pior cenário (pequenas mensagens) em latência e largura de banda com *microbenchmarks* e o Linpack. Da mesma forma, Chakthranont et al. [Chakthranont et al. 2014] integrou o CloudStack com o InfiniBand e conduziu uma avaliação de desempenho em *cluster* virtual e físico usando os *microbenchmarks* da Intel, HPC Challenge, OpenMX e Graph500. Finalmente, Zhang et al. [Zhang et al. 2016] avaliou o desempenho do *hypervisor*, contêiner e cluster físico com (SR-IOV) e PCI *passthrough* integrada com InfiniBand. A avaliação utilizou aplicações HPC representativas e *benchmarks* como o Graph500, conjunto NPB, LAMMPS e SPECMPI 2007.

Em contraste, este trabalho realizou uma comparação entre as interconexões InfiniBand FDR e Gigabit Ethernet, com foco nas aplicações MPI das classes A, B, C e D do conjunto de *benchmarks* NPB. Também rastreou-se o comportamento das aplicações para entender melhor como estas funcionam e como uma interconexão mais rápida influencia o desempenho geral relacionado à entrada de dados crescente. Além disso, foi realizada a avaliação de custo de execução, nas quais as aplicações tiveram seu tempo de execução precificado em instâncias da nuvem publica Microsoft Azure.

5. Conclusão e Trabalhos Futuros

O desempenho de interconexão de rede continua sendo um aspecto crucial dos ambientes de HPC. Para aplicações com um alto nível de dependência de rede, esta atinge ou até supera, o mesmo nível de importância que o poder de processamento. Vários esforços já foram feitos no desenvolvimento de novas tecnologias de rede para HPC [Maliszewski et al. 2019]. Normalmente, essas técnicas não são amplamente

acessíveis por toda a comunidade, e em grupos de pesquisa, muitas vezes, tecnologias comercialmente disponíveis são usadas.

Neste trabalho, motivou-se a análise do perfil das aplicações em relação à quantidade de computação e comunicação, visando criar, a longo prazo, uma metodologia para explorar a interconexão da rede com base nas necessidades da aplicação. Mostrou-se que as aplicações podem ser agrupadas em quatro grupos com necessidades distintas, desde aqueles com alta dependência de rede até aplicações sem dependência de rede.

Os resultados evidenciaram que a interconexão desempenha um papel crucial nas aplicações MPI, que tem como base a execução na maior utilização da rede. Por exemplo, o *kernel* IS aumenta a comunicação à medida que a entrada aumenta e, com esse comportamento, executa aproximadamente 936% melhor com InfiniBand FDR 56 Gbps do que a Ethernet padrão de 1 Gbps. Por outro lado, para aplicações que são CPU-*Bound*, como o *kernel* EP, uma interconexão mais rápida não influencia para se obter um desempenho aprimorado. Neste trabalho, como pode ser visto na Figura 2, o EP foi a única aplicação que não melhorou seu desempenho devido à utilização do IB.

Além disso, com base no desempenho aprimorado das aplicações que usaram a rede, experimentou-se a influência do uso de RDMA, que lê dados diretamente da memória principal de um computador e grava esses dados diretamente na memória principal de outro computador, sem envolver o processador, cache ou mesmo o sistema operacional (SO) [Liu et al. 2004]. Assim, essa técnica melhora significativamente o desempenho, liberando recursos. A análise do custo de execução também corroborou a importância da execução do rastreamento das aplicações, nas quais, com a escolha correta do ambiente e o conhecimento dos requisitos da aplicação, pode-se obter um significativo ganho de desempenho em tempo de execução, assim como poupar recursos financeiros no caso da utilização da computação em nuvem.

Para trabalhos futuros, com base nos resultados obtidos, pretende-se desenvolver um modelo de decisão utilizando SDN (*software-defined network*) para alocar de acordo a necessidade/característica da aplicação, o ambiente mais adequado para sua execução, em termos de desempenho e custo, quando o custo é aplicável ou estatísticas de rede, como congestionamento da rede ou caminho mais curto. Além disso, como a execução foi realizado em um *cluster* físico, essa avaliação pode ser feita em nuvens e ambientes virtualizados usando técnicas bem adotadas em ambientes HPC do mundo real, como o SR-IOV e PCI *passthrough*. Por fim, pode-se reproduzir a mesma metodologia usando aplicações de outros domínios HPC, como previsão do tempo.

Agradecimentos

Este trabalho foi parcialmente apoiado pelo projeto "GREEN-CLOUD: Computação em Cloud com Computação Sustentável" (# 16/2551-0000 488-9), da FAPERGS e CNPq Brasil, programa PRONEX 12/2014, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - *Finance Code* 001, e pelo projeto FAPERGS 01/2017-ARD PARAELASTIC (No. 17/2551-0000871-5). Agradece-se também ao projeto RICAP, parcialmente financiado pelo *Ibero-American Program of Science and Technology for Development* (CYTED), convênio de subvenção nº 517RT0529.

Referências

[Alles et al. 2018] Alles, G. R., Carissimi, A., and Schnorr, L. M. (2018). Assessing the Computation and Communication Overhead of Linux Containers for HPC Applications. In *Symposium on High Performance Computing Systems (WSCAD)*.

[Bailey et al. 1991] Bailey, D. H., Barszcz, E., Barton, J. T., Browning, D. S., Carter, R. L., Dagum, L., Fatoohi, R. A., Frederickson, P. O., Lasinski, T. A., Schreiber, R. S., Simon, H. D., Venkatakrishnan, V., and Weeratunga, S. K. (1991). The NAS Parallel Benchmarks; Summary and Preliminary Results. In *ACM/IEEE Conference on Supercomputing (SC)*.

[Chakthranont et al. 2014] Chakthranont, N., Khunphet, P., Takano, R., and Ikegami, T. (2014). Exploring the Performance Impact of Virtualization on an HPC Cloud. In *International Conference on Cloud Computing Technology and Science (CloudCom)*.

[Escudero-Sahuquillo et al. 2015] Escudero-Sahuquillo, J., Gran, E. G., Garcia, P. J., Flich, J., Skeie, T., Lysne, O., Quiles, F. J., and Duato, J. (2015). Efficient and Cost-Effective Hybrid Congestion Control for HPC Interconnection Networks. *Transactions on Parallel and Distributed Systems (TPDS)*.

[Kamburugamuve et al. 2017] Kamburugamuve, S., Ramasamy, K., Swany, M., and Fox, G. (2017). Low Latency Stream Processing: Apache Heron with Infiniband/Intel Omni-Path. In *International Conference on Utility and Cloud Computing (UCC)*.

[Liu et al. 2004] Liu, J., Wu, J., and Panda, D. K. (2004). High Performance RDMA-based MPI Implementation over InfiniBand. *International Journal of Parallel Programming (IJPP)*.

[Maliszewski et al. 2019] Maliszewski, A. M., Vogel, A., Griebler, D., Roloff, E., Fernandes, L. G., and Navaux, P. O. A. (2019). Minimizing Communication Overheads in Container-based Clouds for HPC Applications. In *Symposium on Computers and Communications (ISCC)*.

[Moura and Hutchison 2016] Moura, J. and Hutchison, D. (2016). Review and Analysis of Networking Challenges in Cloud Computing. *Journal of Network and Computer Applications (JNCA)*.

[Roloff et al. 2017] Roloff, E., Diener, M., Gaspary, L. P., and Navaux, P. O. A. (2017). HPC Application Performance and Cost Efficiency in the Cloud. In *Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*.

[Ruivo et al. 2014] Ruivo, T. P. P. D. L., Altayo, G. B., Garzoglio, G., Timm, S., Kim, H. W., Noh, S., and Raicu, I. (2014). Exploring Infiniband Hardware Virtualization in OpenNebula towards Efficient High-Performance Computing. In *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*.

[Vienne et al. 2012] Vienne, J., Chen, J., Wasi-Ur-Rahman, M., Islam, N. S., Subramoni, H., and Panda, D. K. (2012). Performance Analysis and Evaluation of InfiniBand FDR and 40GigE RoCE on HPC and Cloud Computing Systems. In *Symposium on High-Performance Interconnects (HOTI)*.

[Zahid 2017] Zahid, F. (2017). *Network Optimization for High Performance Cloud Computing*. PhD thesis, Faculty of Mathematics and Natural Sciences, University of Oslo, Oslo, Norway.

[Zhang et al. 2016] Zhang, J., Lu, X., and Panda, D. K. (2016). Performance Characterization of Hypervisor-and Container-Based Virtualization for HPC on SR-IOV Enabled InfiniBand Clusters. In *International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*.