Avaliação Comparativa de Escalabilidade de Aplicações de Alto Desempenho em Cluster Físico e na Nuvem

Thiago B. de Oliveira[1], **Ariel Lauber de P. Silva**[1],
Italo T. da Cunha[1], **Paulo Afonso P. Júnior**[1]

[1]Bacharelado em Ciência da Computação – Universidade Federal de Goiás (UFG)
Regional Jataí – Câmpus Cidade Universitária – BR 364, KM 195, 3800
CEP 75801-615 – Jataí – GO – Brasil

thborges@ufg.br, {ariellauber,italo.tiago}@gmail.com, paulojunior@jatai.ufg.br

Abstract. *The performance analysis of High Performance Computing (HPC) applications has recently been facilitaded by the use of cloud platforms. The grants offered by infrastructure providers to research projects in universities has also been contributing on this migration. However, the performance of HPC applications depends heavily on the Input/Output support of the platform, mostly the intercommunication network between virtual machines. In this study, we compared the performance of selected applications of the NPB NAS Parallel Benchmark and real applications for geographic queries processing, on a physical cluster of commodities machines and another similar cluster on the Microsoft Azure cloud. Our tests demonstrated a speed-up up to 2.7 times lower in the cloud for cpu-bound applications. Despite the superior quality of the processors in the cloud, our tests have also shown a run time up to 3.0 times higher in the cloud.*

Resumo. *A análise de desempenho de aplicações de computação de alto desempenho (HPC) foi recentemente facilitada pelo uso de plataformas de nuvem. Os subsídios concedidos pelos fornecedores de infra-estrutura para projetos de pesquisa nas universidades também tem contribuído neste sentido. No entanto, o desempenho de aplicativos HPC depende fortemente do suporte de Entrada/Saída da plataforma, principalmente da rede de intercomunicação entre as máquinas virtuais. Neste estudo, o desempenho de aplicativos selecionados da suíte NPB NAS Parallel Benchmark e aplicações reais para o processamento de consultas geográfica foi comparado, em um* cluster *físico de máquinas comuns e outro* cluster *semelhante na nuvem Microsoft Azure. Nossos testes demonstraram um* speed-up *até 2,7 vezes menor na nuvem para aplicações limitadas por CPU. Apesar da qualidade superior dos processadores na nuvem, nossos testes também mostraram um tempo de execução até 3,0 vezes maior na nuvem.*

1. Introdução

O aumento recente da disponibilidade de plataformas de nuvem, principalmente fornecedores de infraestrutura como serviço (IaaS), tem causado uma migração da análise de aplicações de computação de alto desempenho (HPC). Aplicação que antes eram testadas exclusivamente em *clusters* nos centros de computação nas universidades são hoje testadas em *clusters* virtuais na nuvem, como pode ser evidenciado na literatura recente

da área. Os subsídios concedidos pelos fornecedores de infra-estrutura para projetos de pesquisa nas universidades também tem contribuído neste sentido[2].

O uso de plataformas de computação em nuvem é vantajoso devido a três fatores principais: *i*) a facilidade de configurar e manter *clusters* para execução de aplicações de computação de alto desempenho, *ii*) a inexistência do custo inicial para adquirir a infraestrutura necessária e, *iii*) a Elasticidade, que proporciona o aumento da infraestrutura sob demanda, fato primordial para o teste de aplicações distribuídas de alto desempenho.

No entanto, o desempenho dessas aplicações está intimamente ligado ao desempenho do subsistema de Entrada/Saída (E/S) da plataforma, principalmente da rede de intercomunicação entre as máquinas virtuais. Segundo [Rixner 2008], a implementação do compartilhamento da interface de rede nos Monitores de Máquinas Virtuais (VMM - Do Inglês, *Virtual Machine Monitors*) ou *Hypervisors* é complexa, devido à necessidade de multiplexação de pacotes em *software*[3], e proteção de memória de pacotes[4] entre as Máquinas Virtuais (MV). Esse processamento extra, em *software*, prejudica o desempenho das aplicações virtualizadas.

No estudo de [Rixner 2008], foi avaliada a abordagem de compartilhamento de E/S de rede exclusivamente por *software* e empregando Acesso de Rede Direto e Concorrente (CDNA, *Concurent Direct Network Access*)[5]. Esta abordagem usa interfaces de rede físicas com múltiplas filas[6] [Intel 2007]. A diferença de vazão entre as duas abordagens chegou a 370% para transmissão e 126% para recepção, à medida que a quantidade de máquinas virtuais variou de 1 a 24. Esta avaliação confirmou que o suporte do *hardware* à virtualização é essencial para E/S de rede, assim como o já consolidado suporte à virtualização nas Unidades Centrais de Processamento (UCP's) modernas.

Desde a publicação deste estudo, outras tecnologias de compartilhamento de E/S em máquinas virtuais surgiram, como Intel VT-x [Abramson et al. 2006] e AMD-V IOMMU (AMD) [AMD 2009]. Estas tecnologias disponibilizam uma Unidade de Gerênciamento de Memória exclusiva para Entrada e Saída (IOMMU), com tabelas de páginas de memória de E/S identificadas, que possibilitam a proteção de memória pelo próprio *hardware*, com o mínimo de interferência do VMM.

Apesar destes avanços, alguns aspectos da virtualização ainda causam preocupação atualmente. Um deles é a interferência de máquinas virtuais vizinhas, alocadas na mesma máquina física. Esta estratégia é empregada pelos provedores de computação em nuvem, com objetivo de reduzir custos e o consumo de recursos naturais [Ferreira et al. 2014]. O isolamento entre as máquinas virtuais, ainda hoje, é um problema reportado por alguns artigos como [Mei et al. 2013, Tudoran et al. 2012, Barker and Shenoy 2010, Hill et al. 2011]. Este fato, se não observado adequa-

[2]A Microsoft e a Amazon possuem programas universitários de apoio à pesquisa em suas respectivas plataformas de computação em Nuvem: Azure e *Elastic Compute Cloud* (EC2).

[3]O tráfego de rede é naturalmente "não solicitado". Um pacote pode ser recebido a qualquer momento e o VMM deve entregá-lo para uma máquina virtual específica, destinatária do pacote.

[4]Uma máquina virtual pode arbitrariamente transmitir ou receber pacotes de outras MV's, se o VMM não garantir o isolamento.

[5]Estratégia de compartilhamento de rede implementada no *Hypervisor* Xen na época da publicação.

[6]A tecnologia mais atual da Intel chama-se VMDq *Virtual Machine Device Queues*, que provê suporte ainda mais específico para máquinas virtuais vizinhas.

damente, compromete a repetibilidade dos experimentos, conforme discutido por [Luszczek et al. 2012].

Nosso trabalho avalia a interferência da plataforma de computação em nuvem na escalabilidade de aplicações distribuídas de alto desempenho e compara a escalabilidade das mesmas aplicações em um *cluster* físico. Foram avaliados aspectos que podem comprometer a validação de experimentos científicos de escalabilidade, que avaliam principalmente o *speed-up* de aplicações distribuídas e paralelas, conduzidos nestas plataformas. Os experimentos na nuvem foram realizados na plataforma Microsoft Azure. As principais contribuições de nosso trabalho são:

- Avaliação comparativa de aplicações de alto desempenho entre *cluster* físico e em plataforma de nuvem para identificar possíveis diferenças de tempo de execução e *speed-up*;
- Avaliação de aplicações reais e aplicações científicas de *benchmark* conhecidas, e
- Identificação de comportamentos das aplicações que interferem na escalabilidade, tanto na nuvem quanto no *cluster* físico.

O restante do artigo está organizado da seguinte forma: Na Seção 2, são apresentados trabalhos correlatos que realizaram experimentos que evidenciam a falta de isolamento e variabilidade de desempenho em plataformas de nuvem. Na Seção 3, a configuração das instâncias de máquinas virtuais na nuvem é apresentada, juntamente com a descrição das aplicações testadas e como foram conduzidos os experimentos. Na Seção 4, os resultados dos experimentos são apresentados e discutidos e, por fim, a Seção 5, apresenta a conclusão do trabalho e direções para trabalhos futuros.

2. Trabalhos Correlatos

Apesar de um grande esforço em medir o desempenho das plataformas de computação em nuvem, fato que pode ser notado devido a grande quantidade de publicações recentes (Veja [Mei et al. 2013]), a quantidade de variáveis que interferem na comparação é muito grande e isso dificulta a avaliação completa das plataformas. Como afirmado por [Li et al. 2012], por ser uma área de estudo em desenvolvimento, não há uma taxonomia de avaliação amplamente aceita e muitos trabalhos confundem termos ou realizam experimentos inconsistentes para avaliar determinadas características.

A maioria dos trabalhos analisam comparativamente as implementações de VMM existentes, como [Walters et al. 2008, Huber et al. 2011, Luszczek et al. 2012]. Nota-se uma instabilidade em relação às conclusões, devido a constante evolução da implementação das tecnologias de virtualização e da recente disponibilização de *hardware* com suporte à virtualização (Intel VT-d e AMD-V). Alguns trabalhos avaliam o desempenho de diferentes tecnologias de virtualização (paravirtualização, virtualização completa, e outras) como [Younge et al. 2011]. Atualmente, os principais VMM's existentes implementam somente a tecnologia de paravirtualização ou virtualização completa.

O estudo de [Ferreira et al. 2014] concluiu que o aumento da quantidade de MV por máquinas físicas é o fator mais relevante na redução do desempenho de máquinas virtuais vizinhas, devido à maior concorrência por recursos na máquina física. A falta de controle da vizinhança por parte do usuário da plataforma, e a falta de isolamento por parte do VMM, comprometem a repetibilidade dos experimentos, conforme discutido em [Luszczek et al. 2012, Tudoran et al. 2012, Mei et al. 2013].

Para demonstrar a variação de desempenho devido a este compartilhamento, [Tudoran et al. 2012] avaliou comparativamente uma nuvem privada, Nimbus e a nuvem pública, Azure. A conclusão do estudo em relação ao desempenho de rede foi que há uma variabilidade considerável numa janela de tempo de uma semana. Os desvios padrões do percentual de variação relatados foram de $24,2\%$ para a nuvem Nimbus contra $52,3\%$ para instâncias ExtraLarge[7] e $120,7\%$ para Small[8] na nuvem Azure.

Resultados que reafirmam a falta de isolamento entre máquinas virtuais vizinhas são apresentados em [Mei et al. 2013]. [Barker and Shenoy 2010] estuda este comportamento em aplicações multimídia, sensíveis a flutuação de latência. Os autores identificaram uma degradação de até 75% em aplicações limitadas por disco (*disk-bound*) e afirmaram que a interferência em aplicações sensíveis a latência de rede poderia ser reduzida através de configuração específica do VMM para a aplicação, mas não completamente eliminada. [Hill et al. 2011] evidenciou nos experimentos momentos esporádicos onde a velocidade de execução de aplicações em uma máquina virtual reduz-se até 4vezes. Ocorreram também falhas que comprometem a execução das tarefas em até 16% das vezes devido a *timeouts* de rede.

Poucos estudos avaliam a consequência destas conclusões na escalabilidade de aplicações distribuídas em *clusters* de máquinas virtuais. O trabalho mais relevante encontrado foi [Expósito et al. 2013]. Nele, aplicações de alto desempenho selecionadas da suite NPB NAS Parallel Benchmark [Bailey et al. 1994] foram avaliadas na plataforma EC2 da Amazon. No entanto, não foi realizado um estudo comparativo com *clusters* físicos. Apesar de existirem relatórios de execução da mesma suite divulgados na Internet, a comparação direta é difícil devido a heterogeneidade das máquinas do *cluster* físico e da nuvem.

Nosso estudo compara o *speed-up* de aplicações de alto desempenho entre um *cluster* virtual em plataforma de computação em nuvem e um *cluster* físico, com o objetivo de quantificar e qualificar o quanto a virtualização e a plataforma podem comprometer a validação de experimentos científicos de escalabilidade, especificamente na plataforma de nuvem Microsoft Azure.

3. Metodologia de Avaliação

Para realizar os experimentos, um conjunto de aplicações foram selecionadas, observando quatro fatores principais:

1. O *speed-up* da aplicação, de forma a ter valores representativos, tanto *speed-up* próximo de nulo quanto próximo a linear;
2. O tipo de aplicação, se é um aplicativo real ou uma aplicação de *benchmark* consolidada na literatura observando experimentos similares;
3. A diversidade das aplicações em relação ao recurso computacional mais empregado: rede ou CPU, e
4. Se a aplicação explora múltiplos níveis de paralelismo, ou seja, além do paralelismo inerente a distribuição, a aplicação também realiza processamento paralelo *multithread*, preparada para uso de máquinas *multicore*.

[7]ExtraLarge instance = 8 vCPUs e 14 GB RAM, interface de rede exclusiva.
[8]Small instance = 1 vCPU e 1,75 GB RAM, interface de rede compartilhada.

Para a aferição do *speed-up* das aplicações, *clusters* de tamanhos variados foram configurados. O objetivo da escolha dos tamanhos foi aumentar a taxa de utilização de comunicação na interface de rede, observando que este é um importante, muitas vezes o principal, limitador de *speed-up* em sistemas distribuídos. Ao testar um *cluster* com 8 CPU's ou vCPU's, por exemplo, preferiu-se utilizar 8 máquinas com 1 core cada, de forma que as 8 máquinas efetuem comunicação entre si pela interface de rede.

A Seção 3.1 detalha as aplicações utilizadas nos experimentos e o tamanho do *cluster* e instâncias de MV utilizadas na nuvem é detalhado na Seção 3.2.

3.1. Aplicações executadas nos experimentos

Um conjunto de aplicações da suite NPB NAS Parallel Benchmark foram usadas nos experimentos. As aplicações, também chamadas *kernel's*, são escritas em linguagem C e Fortran, utilizando a especificação de comunicação distribuída MPI (*Message Passing Interface*). A implementação de MPI utilizada no *cluster* foi o Open MPI[9]. As aplicações escolhidas são as que mais usam banda de rede durante sua execução, conforme indicado em [Bailey et al. 1994, Expósito et al. 2013]. São elas:

- IS_c - Ordenação de Inteiros. Uso de Allreduce e comunicação ponto-a-ponto na inicialização.
- MG_c - Multi-Grid numa sequência de Meshes, comunicação.
- CG_c - Resolve um sistema linear esparso com o método do Gradiente Conjugado. Comunicação ponto-a-ponto intensiva.
- FT_c - Transformada Rápida de Fourier 3D. Uso extensivo de primitivas Alltoall que sobrecarregam a rede.

O *c*, em *IS_c* e demais, representa o tamanho da instância da aplicação compilada. Este tamanho foi escolhido de forma a estressar completamente os recursos das instâncias de *cluster* selecionadas. A instância *b*[10] seria pequena e não estressaria completamente o *cluster*. Já a instância *d* implicaria em aumento substancial do tempo de execução dos testes, sem oferecer em contrapartida uma diferença relevante para o experimento que justificasse sua escolha.

Embora a suite NPB seja referência para mensurar o desempenho de máquinas paralelas (com memória compartilhada ou distribuída), os próprios autores afirmam que durante a sua implementação foram utilizadas instâncias simplificadas de programas reais, que possuem casos melhores de balanceamento de carga [Bailey et al. 1994, pág. 4]. Como o balanceamento do *cluster* interfere no *speed-up* de aplicações distribuídas, experimentamos também uma aplicação real: um banco de dados distribuído para processamento de dados espaciais (DGEO). Este banco de dados foi desenvolvido em um projeto de pesquisa em nossa universidade, utilizando linguagem C e Go, com *threads* nativas para explorar múltiplos níveis de paralelismo e comunicação distribuída através de protocolos próprios, usando serialização Gob, sobre *sockets* TCP. A aplicação DGEO foi usada para o processamento de consultas de junção espacial (*spatial join*), de forma paralela e distribuída.

[9] www.open-mpi.org

[10] Na suite existem instâncias *Standard*: *A*, *B* e *C*, *Large*: *D*, *E* e *F*, além de *S* e *W*. Estas duas últimas são consideradas pequenas para o desempenho de máquinas atuais.

Uma consulta de junção espacial processa conjuntos de dados (chamados *datasets* ou *shapefiles*) de diferentes tamanhos, executando algoritmos de geometria computacional para filtragem de predicados. O processamento destas consultas emprega um uso considerável de rede, mas é limitado por CPU (*cpu-bound*). Durante o processamento distribuído destas consultas, os dados dos *datasets* são distribuídos pelo *cluster* e a distribuição realizada impacta no *speed-up* da aplicação. Algumas consultas espaciais possuem *speed-up* bons e outras *speed-ups* ruins. Três consultas espaciais foram empregadas nos testes (Q_a, Q_b e Q_c), cada um com um *speed-up* particular, escolhidas de forma a representar o comportamento de aplicações distribuídas pouco escaláveis, $\frac{1}{4}linear$ escaláveis, ou seja, a escalabilidade é $\frac{1}{4}$ da escalabilidade teórica máxima, e $\frac{1}{2}linear$ escaláveis.

A cada teste executado, todas os processos de SO referentes às aplicações foram encerrados e reiniciados, com o objetivo de não reaproveitar *caches* da aplicação, do sistema operacional ou de *hardware*. Nenhuma das aplicações utilizam dados armazenados em disco na inicialização, e não foi necessário, portanto, limpar a *cache* de disco do Sistema Operacional.

Os tempos de inicialização das aplicações foram descartados, e somente o tempo de processamento efetivo do algoritmo distribuído foi considerado. No caso da suite NPB, o que acontece na inicialização das aplicações pode ser visto em [Bailey et al. 1991]. Na aplicação DGEO, o tempo de inicialização corresponde à inserção e distribuição dos *datasets* pelo *cluster*, para o posterior refinamento do predicado espacial. Os fragmentos de *datasets* distribuídos são armazenados completamente em memória.

Cada aplicação foi executada cinco vezes. Os valores apresentados na seção de resultados foram calculados ignorando o maior e o menor tempo encontrado, e a média dos outros três valores foi calculada. Nos testes efetuados no *cluster* virtual, as medições foram comparadas em dias e momentos diferentes, para garantir que os tempos não sofreram interferências momentâneas da plataforma e para possibilitar a repetibilidade dos experimentos.

3.2. Configuração dos *Clusters* utilizados

Em nossos experimentos, tanto o *cluster* físico quanto o virtual utilizaram um conjunto de 16 máquinas. O *cluster* físico utilizou um conjunto de máquinas do Laboratório de Redes de Computadores LARC, do Curso de Ciência da Computação na Regional Jataí da Universidade Federal de Goiás. O *cluster* virtual foi configurado na plataforma Azure, com instâncias de MV de tamanho A1 a A3. A Tabela 1 detalha o *hardware* completo das máquinas físicas e virtuais.

O tamanho das instâncias do *cluster* variou de 1 a 64 núcleos: de 1 máquina, com apenas um núcleo, até 16 máquinas com 4 núcleos cada. Os tamanhos de instância foram escolhidos de forma a usar ao máximo a interface de rede disponível, conforme descrito na Seção 3. A Tabela 2 detalha a composição de cada instância: quantidade de núcleos, quantidade de máquinas, máquina virtual escolhida na nuvem e o custo total de cada instância, em R$. O custo foi empregado na avaliação da relação custo *x* escalabilidade.

Para limitar a quantidade de núcleos nas máquinas físicas, nas instâncias de *cluster* com 1 ou 2 núcleos de *CPUs*, os núcleos foram desabilitados através de comando específico do Linux (`echo 0 > /sys/devices/system/cpu/cpu1/online`), de forma que o núcleo fosse totalmente desabilitado durante a execução.

Tabela 1. Detalhes do *Hardware* dos *clusters* utilizados na avaliação.

Item	*Cluster* Físico	*Cluster* Virtual
Quantidade de Máquinas	16	16
CPU	Intel Core i5 3330 3 GHz	Intel Xeon E5-2660 2.2 GHz
Núcleos	4	4[a]
Cache	6MB	20MB
RAM	DDR3 SDRAM 4 GB 1.333 MHz	Não documentado[b]
Rede	1 Gbps	1 Gbps
Sistema Operacional	Ubuntu Server 14.4 LTS	Ubuntu Server 14.4 LTS
Switch	D-Link DGS-1210-28P Gigabit	Não documentado[c]

[a]O processador Xeon E5-2660 possui 8 núcleos. Em nosso experimento, duas máquinas virtuais podem ser vizinhas no mesmo chassi.

[b]O tamanho é particular da instância e foi limitado de acordo com cada instância escolhida, para refletir a mesma quantidade da máquina física.

[c]Como é comum entre os provedores de plataforma de nuvem, a descrição completa dos equipamentos físicos não é fornecida.

Tabela 2. Tamanho das instâncias utilizadas nos experimentos.

Núcleos	Máquinas	Núcleos p/ máq	Instância Azure	Preço p/ hora (R$)
1	1	1	A1	0,168
2	2	1	A1	0,336
4	4	1	A1	0,672
8	8	1	A1	1,344
16	16	1	A1	2,688
32	16	2	A2	5,376
64	16	4	A3	10,760

Na nuvem Azure, as máquinas virtuais foram desligadas e redimensionadas com a quantidade de vCPUs indicada em cada instância. Como no Azure a quantidade de memória aumenta à medida que a quantidade de vCPUs aumenta, para prover o mesmo comportamento em relação à disponibilidade de memória, criamos um aplicativo que reserva a quantidade de memória excedente (em relação às máquinas físicas) através da chamada de sistema `mlock`. Este aplicativo foi executado antes de cada teste com tamanho diferente de instância.

4. Avaliação de Desempenho

Esta seção avalia os resultados obtidos, conforme a descrição dos cenários na seção anterior. Dividimos a avaliação em quatro subseções: A Seção 4.1 avalia o tempo de execução das aplicações, considerando as diferenças entre o *hardware* físico e o virtual, a Seção 4.2 apresenta a escalabilidade aferida nos dois *clusters*, a Seção 4.3 compara o *speed-up* individual de cada aplicação e por fim, a Seção 4.4 faz uma avaliação de custo *x* benefício em relação ao aumento do tamanho do *cluster* virtual e a escalabilidade obtida.

4.1. Avaliação do Tempo de Execução das aplicações

Apesar das diferenças entre o *hardware* do *cluster* físico e da nuvem, ou seja, *i*) apesar do maior *clock* de CPU no cluster físico (3 Ghz *x* 2.2 Ghz) e *ii*) o processador próprio de servidores, Xeon, com *cache* de CPU maior (20 MB *x* 6 MB), porém compartilhado em

ambiente de computação em nuvem, os experimentos identificaram, na média entre todas as execuções, que o tempo de execução das aplicações *cpu-bound* é até 3,06 vezes maior no ambiente Azure (ta) que no físico (tf), como apresentado na Figura 1.

Somente a diferença percentual de tamanho de CPU, que é de 36,3% (2.2 Ghz x 3 Ghz), não pode justificar a diferença de tempo de execução ($\approx 300\%$). Como estas aplicações não são intensivas de disco, memória e nem rede, a diferença se deve ao próprio ambiente compartilhado na nuvem, reafirmando os resultados já reportado nos trabalhos de [Mei et al. 2013, Barker and Shenoy 2010, Hill et al. 2011].

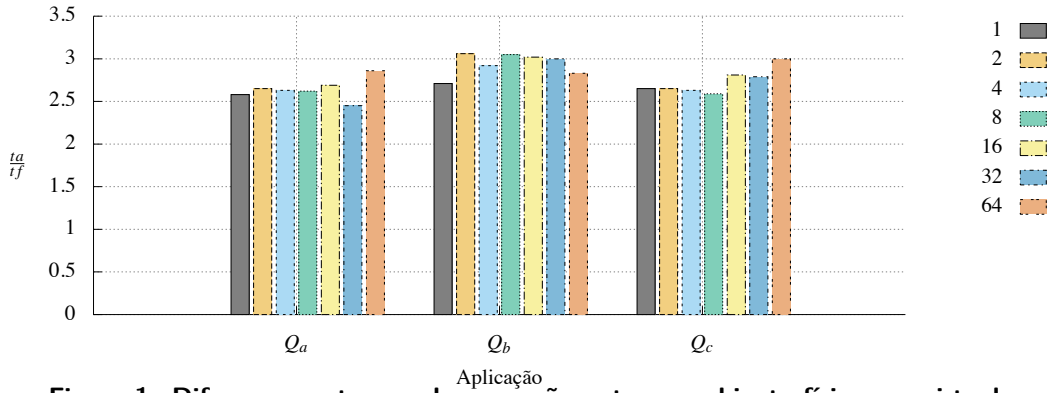

Figura 1. Diferença no tempo de execução entre o ambiente físico e o virtual para aplicações *cpu-bound*.

Nas aplicações da suite NPB (Figura 2), que são *network-bound*, esta diferença é menor e apresenta um comportamento menos constante, com desvio padrão maior entre os tamanhos de instância do *cluster* (0,5864), se comparado ao caso anterior (0,1798). Outro aspecto observado é que, enquanto na aplicação IS_c a tendência é o aumento da diferença (de ≈ 1 com 1 núcleo a ≈ 3 com 64), nas outras três, FT_c, MG_c e CG_c, a tendência é diminuir. Esta diferença se deve ao comportamento de uso de rede destas aplicações.

A diferença fundamental entre estas aplicações é que, em IS_c, à medida que a quantidade de máquinas aumenta, a comunicação necessária se mantém a mesma e é dividida entre mais interfaces de rede, prevalecendo o comportamento das aplicações anteriores ($Q_{a..c}$). Nas outras três aplicações FT_c, MG_c e CG_c, a comunicação necessária entre os processos aumenta, à medida que o número de máquinas também aumenta. Esse aumento na comunicação atinge o limite da capacidade do *switch* do *cluster*, fazendo com que a diferença de tempo de execução diminua. O *cluster* virtual é beneficiado pela ocorrência de máquinas vizinhas, onde parte da comunicação ocorre apenas em *loopback* no próprio *hypervisor*.

A conclusão em relação ao tempo de execução é que, quando a aplicação é *cpu-bound* há um *overhead* significante na plataforma de nuvem. Em aplicações onde o uso de CPU é limitado pela comunicação (E/S de rede) este *overhead* diminui. Se considerarmos a diferença do *clock* e do tipo de CPU que beneficiam os testes no *cluster* Azure, este número aumentaria em $\approx 30\%$, o que é uma diferença relevante a ser considerada, por exemplo, no cálculo do custo da contratação de recursos na nuvem.

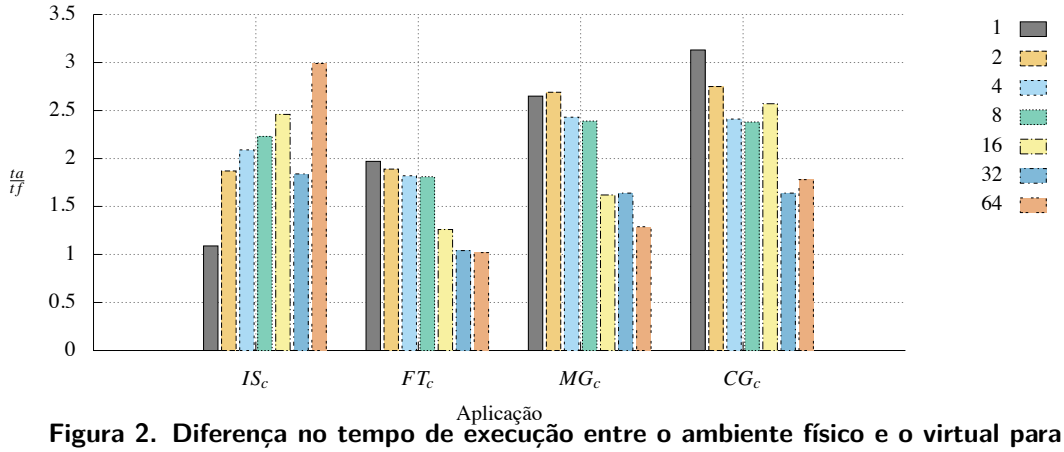

Figura 2. Diferença no tempo de execução entre o ambiente físico e o virtual para aplicações *network-bound*.

4.2. Escalabilidade das Aplicações

Nesta seção, uma visão geral da escalabilidade das aplicações, tanto no *cluster* físico quanto no virtual é apresentada. De forma geral, os resultados são coerentes com a escalabilidade reportada em trabalhos relacionados.

A Figura 3 apresenta o *speed-up* de cada uma das aplicações testadas, em cada instância do *cluster*, de 1 a 64 núcleos. A aplicação que apresentou o melhor *speed-up* foi Q_c (próximo de $\frac{1}{2}$ linear). As demais aplicações possuem *speed-up* menores, variando entre quase nulo (Q_a) e $\frac{1}{2}$ linear (Q_c). Q_b é um exemplo de escalabilidade próxima a $\frac{1}{4}$ linear. Nenhuma das aplicações apresentou comportamento massivamente paralelo.

Ainda, na Figura 3, é possível notar que apesar de algumas aplicações manterem o comportamento de escalabilidade entre os dois *clusters*, outras possuem diferenças significativas, principalmente as aplicações da suite NPB. Uma comparação detalhada, aplicação à aplicação, é realizada na Seção 4.3.

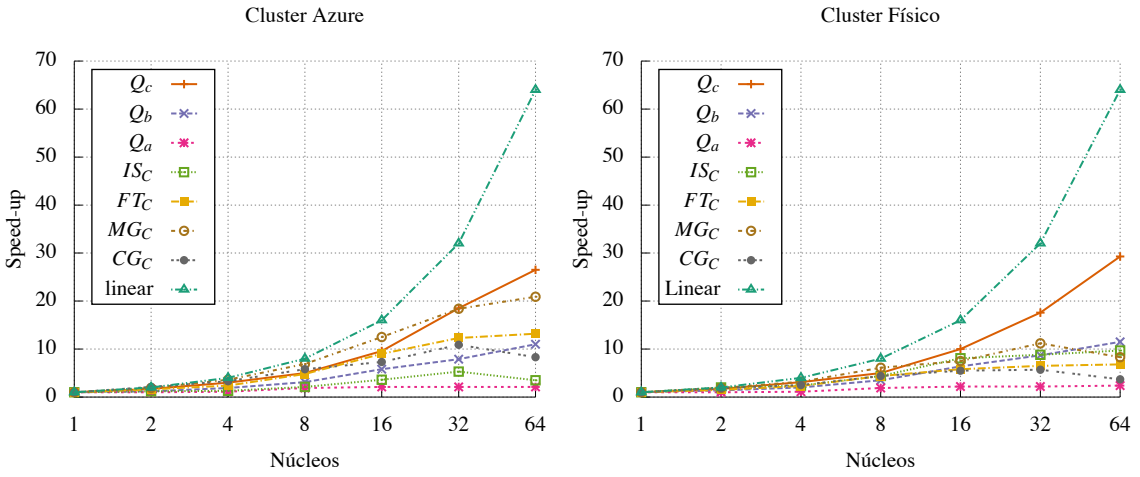

Figura 3. *Speed-up* das aplicações executadas e *speed-up* linear para referência.

4.3. Comparação de Speed-up Individual

Esta seção apresenta a avaliação de escalabilidade, analisando individualmente o *speed-up* de cada aplicação na nuvem e no *cluster* físico. Apesar do tempo das aplicações ser

maior na nuvem, conforme apresentado na seção anterior, mesmo com um maior tempo de execução na nuvem, a aplicação pode apresentar *speed-up* similar à plataforma física, à medida que o tamanho da instância do *cluster* aumenta. O objetivo desta análise é demonstrar se a avaliação científica da escalabilidade de aplicações distribuídas pode ser comprometida pelo fato de ser realizada na nuvem.

Três comportamentos distintos, em relação ao *speed-up*, foram encontrados durante a análise: *i*) a aplicação testada se comporta de forma similar, *ii*) o *speed-up* é significativamente maior no físico e *iii*) o *speed-up* é significativamente maior na nuvem. Dicutiremos os três a seguir.

A Figura 4 apresenta os *speed-ups* das aplicações Q_a, Q_b e Q_c. Para a aplicação Q_a, que por natureza possui *speed-up* próximo de nulo, há uma diferença maior no final da curva (instâncias com 16, 32 e 64 núcleos). Isso indica que aplicações com baixa escalabilidade tendem a acentuar ainda mais a redução do *speed-up* na nuvem.

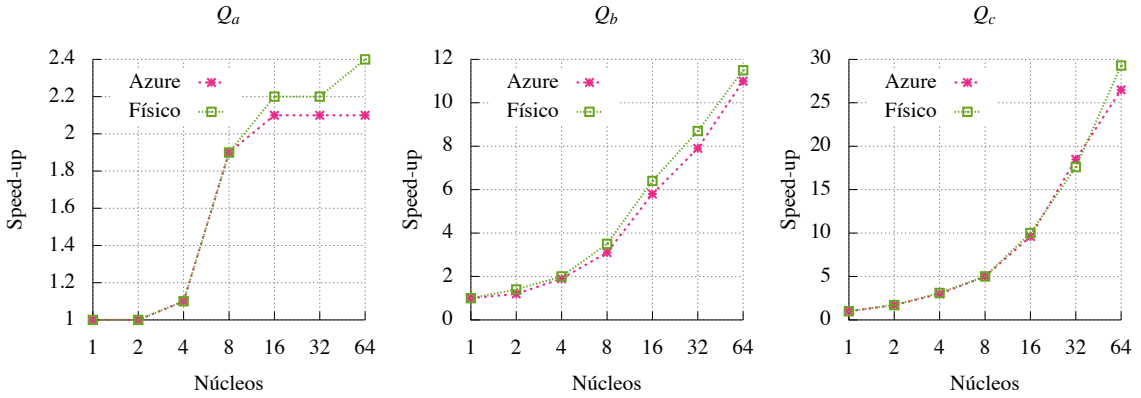

Figura 4. *Speed-up* inferior na nuvem, para aplicações $Q_{a..c}$.

As aplicações Q_b e Q_c, *speed-up* $\frac{1}{4}$ e $\frac{1}{2}$ linear, respectivamente, demonstraram uma escalabilidade muito similar entre o *cluster* nuvem e o físico. A maior diferença de *speed-up* entre as duas curvas (no gráfico da Figura 4, onde as curvas se separam) varia de $0,40$ ($\approx 10\%$ menor que o físico) a $2,86$ ($\approx 11,2\%$ menor que o físico).

Outro comportamento observável é que a curva tende a acentuar a diferença nas instâncias maiores (Q_a e Q_c, de 32 para 64 núcleos). A características destas aplicações é que ocorre uma maior localidade de dados quando há mais núcleos para processar o algoritmo, reduzindo a comunicação necessária e aumentando o uso de CPU do *cluster*. Isso provoca uma redução no tempo final de execução e consequentemente uma diferença de *speed-up*. Este comportamento é ainda mais acentuado na avaliação da aplicação IS_c.

A diferença de *speed-up* da aplicação IS_c é apresentada na Figura 5. Nela, a diferença de speed-up entre as duas curvas chega a ser de $6,15$ pontos na instância de 64 núcleos ($2,74$ vezes menor que o físico, $\approx 63,5\%$ menor). Esta aplicação acentua ainda mais o comportamento observado para $Q_{a..c}$. A diferença ocorre devido ao comportamento da aplicação: a comunicação da ordenação distribuída ocorre apenas no início e final do processamento, enquanto a CPU é usada durante todo o tempo da execução. O tempo de execução de aplicações *cpu-bound*, portanto, não é só maior na nuvem, mas também aumenta em comparação à execução no *cluster* físico, à medida que o tamanho da instância também aumenta, e isso se reflete na diferença de *speed-up* apresentada.

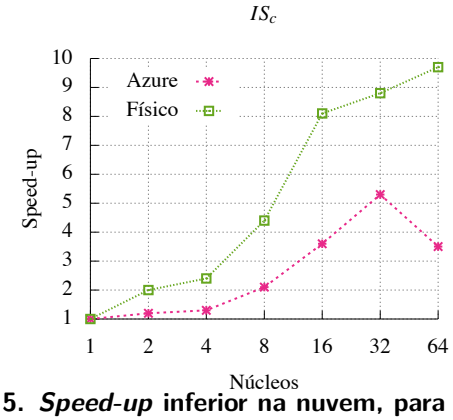

Figura 5. *Speed-up* inferior na nuvem, para aplicação IS_c.

Por outro lado, para as aplicações *network-bound*, há uma diferença de escalabilidade positiva na nuvem, ou seja, o comportamento inverte-se em relação às aplicações anteriores. Este cenário é ilustrado na Figura 6. Nas aplicações FT_c, MG_c e CG_c, quanto maior a quantidade de núcleos, maior é a comunicação na rede, devido a troca de mensagens de um núcleo com todos os outros. A aplicação que apresentou maior diferença foi MG_c: diferença de *speed-up* de $8,72$ pontos ($\approx 106\%$ melhor que o físico) na instância de 64 núcleos. Na média entre as instâncias de 16, 32 e 64 núcleos, a diferença é de $3,29$ pontos ($\approx 52,1\%$ melhor que o físico).

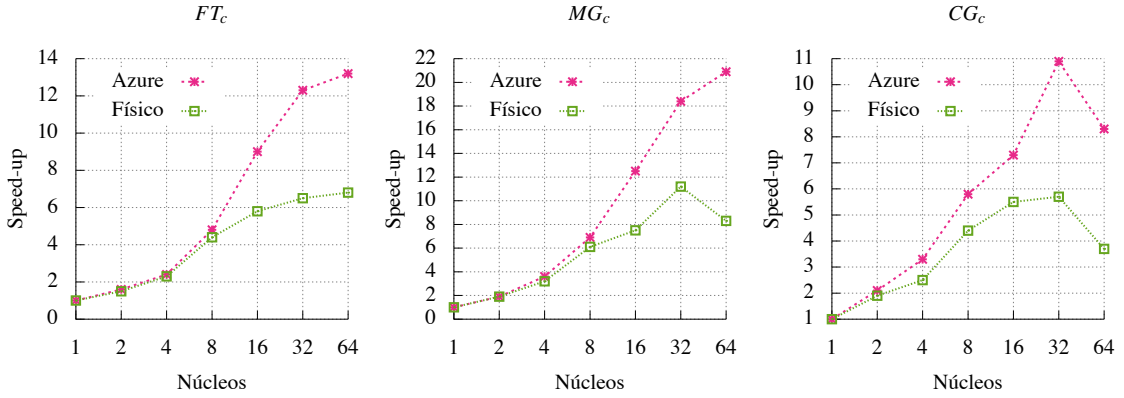

Figura 6. *Speed-up* superior na nuvem, para aplicações FT_c, MG_c e CG_c.

No *cluster* físico, a partir de 16 núcleos, há uma degradação do *speed-up* devido o uso total da capacidade do *switch*. Na nuvem, a escolha das MV justifica a continuidade do *speed-up*: as MV $A2$ e $A3$, usadas nas instâncias maiores do *cluster* (32 e 64), possuem maior banda de rede. Além disso, conforme destacado na seção anterior, a existência de máquinas virtuais vizinhas também contribui para a diferença, devido ao uso de *loopback* no VMM. No entanto, o limite da banda também foi atingido na consulta CG_c, de 32 para 64 núcleos, onde há uma queda de *speed-up* acentuada (Figura 6).

Esta é uma limitação que a plataforma de nuvem impõe ao pesquisador. Para se testar instâncias maiores de *cluster*, é necessário o uso de instâncias maiores de MV. Deve-se, portanto, atentar-se a estes detalhes na metodologia, para não comprometer a avaliação de escalabilidade de aplicações distribuídas. Outra possibilidade é usar instâncias de MV iguais, e avaliar a escalabilidade dobrando o número de máquinas e não de núcleos. Porém, aplicações puramente distribuídas, que não exploram o paralelismo multinível, como é o caso da suite NPB devem ser tratadas em particular.

4.4. Avaliação de Preço e Escalabilidade na Nuvem

Realizar uma comparação do custo total de aquisição, configuração e manutenção de um *cluster* físico com um *cluster* em plataforma de nuvem está fora do escopo deste artigo, devido à quantidade de variáveis a serem analisadas e a necessidade de observar um horizonte de tempo de vida da aplicação. Nosso objetivo nesta seção é somente comparar a escalabilidade da aplicação com o custo dos recursos utilizados na nuvem.

O único recurso empregado na plataforma de nuvem foram as MV's. Conforme pode-se notar na Tabela 2, o aumento do custo é linear, ou seja, a preços atuais (Abril/2015), paga-se exatamente o dobro do valor pelo dobro de máquinas. Não há cobrança de banda de rede entre as MV's situadas no mesmo *datacenter*.

Nenhuma das aplicações testadas é embaraçosamente paralela, com *speed-up* próximo ou igual a linear, e portanto, nenhuma delas apresenta vantagem proporcional entre custo e aumento de desempenho. No entanto, normalizando o *speed-up* de cada aplicação individualmente é possível avaliar melhor o cenário.

Sejam $sp_{a,i}$ o *speed-up* da aplicação a em cada tamanho de instância i do *cluster*, e max_a, o maior *speed-up* da aplicação a em todas as instâncias, o gráfico da função $sp_{a,i}/max_i$, para cada aplicação $a = \{Q_a, Q_b, Q_c, IS_c, FT_c, MG_c, CG_c\}$ é apresentado na Figura 7. Na Figura, quando uma linha está acima da linha de referência (Linha custo), significa que há algum ganho de *speed-up* com o aumento da instância. E vice-versa, quando a linha está abaixo da linha custo, aumentar o tamanho da instância piora o desempenho da aplicação em relação ao tamanho exatamente anterior.

Observando o comportamento das linhas (Figura 7), somente a aplicação Q_c mantêm um aumento de *speed-up* em todas as instâncias e está acima da linha custo (iguala em 64). Todas as demais aplicações pioram ou mantêm o desempenho na mudança de 32 para 64 núcleos, cruzando com a linha de custo. Q_c, MG_c, CG_c e FT_c demonstram aumento de *speed-up* até 16 núcleos. A aplicação IS_c, o pior desempenho na nuvem, não apresentou vantagem desde a instância com 4 núcleos.

De forma geral, quanto pior foi o *speed-up* da aplicação na nuvem, pior também foi a avaliação em relação ao custo. Porém, apenas duas aplicações (Q_c e MG_c) apresentaram ganho de *speed-up* que justificaria o aumento da instância até certo tamanho, para reduzir o tempo de execução.

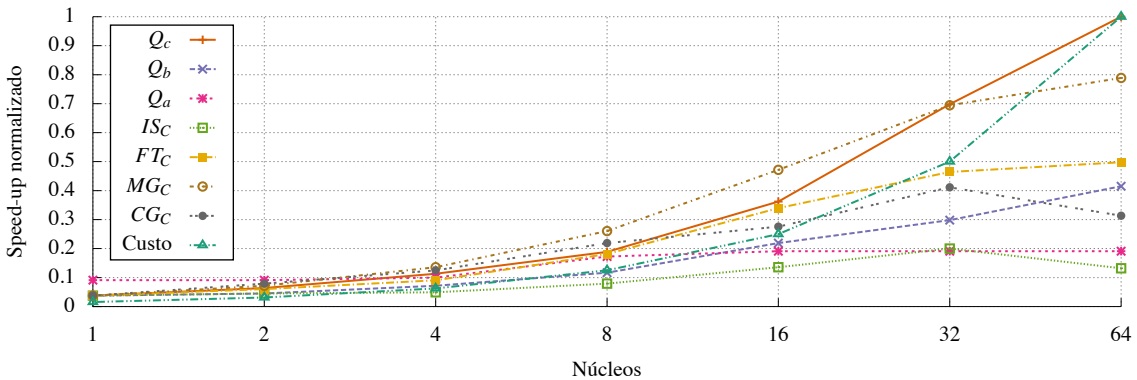

Figura 7. Speed-up das aplicações normalizado para comparação de vantagem de preço da nuvem.

5. Conclusão

Este trabalho avaliou a interferência da plataforma de computação em nuvem na escalabilidade e tempo de execução de aplicações distribuídas de alto desempenho e comparou o tempo de execução e a escalabilidade das mesmas aplicações em um *cluster* físico.

Para isso, foram utilizadas aplicações reais e de *benchmark* conhecidos da suite NPB NAS. As aplicações foram escolhidas para demonstrar diferentes tipos de uso de recursos computacionais: aplicações *cpu-bound*, *network-bound* e com diferentes níveis de *speed-up*. O intuito foi descobrir se, e como, a plataforma de nuvem compromete a execução de testes científicos de escalabilidade na nuvem.

Nossos experimentos demonstraram um *speed-up* até 2,7 vezes menor na nuvem, para aplicações limitadas por CPU, e apesar da qualidade superior dos processadores na nuvem, nossos testes também mostraram um tempo de execução até 3,06 vezes maior na nuvem. Aplicações limitadas por rede, *network-bound*, demonstram *speed-up* até 106% maior que o físico, devido à maior banda de rede disponível na plataforma de nuvem. Estes resultados evidenciam que é ainda mais importante detalhar a metodologia dos testes de aplicações distribuídas na nuvem e a análise deve observar a característica principal da aplicação para concluir sobre a sua escalabilidade.

Um esforço adicional é necessário em trabalhos futuros para generalizar ou expandir os resultados apresentados para outras plataformas de nuvem públicas e, também, para plataformas de nuvem privadas e virtuais. Nas plataformas de nuvem privadas, o objetivo seria avaliar se o controle total de *hardware* e *software* utilizados, interferem nos resultados obtidos.

Referências

Abramson, D., Jackson, J., Muthrasanallur, S., Neiger, G., Regnier, G., Sankaran, R., Schoinas, I., Uhlig, R., Vembu, B., and Wiegert, J. (2006). Intel Virtualization Technology for Directed I/O. *Intel Technology Journal*, Vol. 10 Is.

AMD, A. M. D. (2009). I/O Virtualization Technology (IOMMU) Specification Revision 1.26. *White Paper, AMD*, 1:2–11.

Bailey, D., Browning, D., Carter, R., Dagum, L., Fatoohi, R., Fineberg, S., Frederickson, P., Lasinski, T., Schreiber, R., Simon, H., and Others (1994). The NAS Parallel Benchmarks. *NASA Ames Research Center: Moffett Field, CA*.

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., and Others (1991). The NAS parallel benchmarks. *International Journal of High Performance Computing Applications*, 5(3):63–73.

Barker, S. and Shenoy, P. (2010). Empirical evaluation of latency-sensitive application performance in the cloud. *MMSys '10 Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, pages 35–46.

Expósito, R. R., Taboada, G. L., Ramos, S., Touriño, J., and Doallo, R. (2013). Performance analysis of HPC applications in the cloud. *Future Generation Computer Systems*, 29(1):218–229.

Ferreira, C. H. G., Ribeiro, J. a. B., Júnior, W. D. B., Estrella, J. C., Filho, D. M. L., and Peixoto, M. L. M. (2014). Identificação de gargalos de desempenho em ambientes virtuais para uso em computação em nuvem. In *Anais do WPerformance - CSBC 2014*, pages 1975–1988.

Hill, Z., Li, J., Mao, M., Ruiz-Alvarez, A., and Humphrey, M. (2011). Early observations on the performance of Windows Azure. *Scientific Programming*, 6640(2-3):121–132.

Huber, N., von Quast, M., Hauck, M., and Kounev, S. (2011). Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments. *Proceedings of the 1st International Conference on Cloud Computing and Services Science (CLOSER 2011)*, pages 563–573.

Intel (2007). Improving Network Performance in Multi-Core Systems. *White Pap. Intel Ethernet Controllers*, 1:1–4.

Li, Z., OBrien, L., Cai, R., and Zhang, H. (2012). Towards a Taxonomy of Performance Evaluation of Commercial Cloud Services. In *2012 IEEE Fifth International Conference on Cloud Computing*, pages 344–351. IEEE.

Luszczek, P., Meek, E., Moore, S., Terpstra, D., Weaver, V. M., and Dongarra, J. (2012). Evaluation of the HPC Challenge Benchmarks in Virtualized Environments. In *Proceedings 2011 International Conference on Parallel Processing*, volume 7156 of *Lecture Notes in Computer Science*, pages 436–445, Berlin, Heidelberg. Springer.

Mei, Y., Liu, L., Pu, X., Sivathanu, S., and Dong, X. (2013). Performance Analysis of Network I/O Workloads in Virtualized Data Centers. *IEEE Transactions on Services Computing*, 6(1):48–63.

Rixner, S. (2008). Network Virtualization: Breaking the Performance Barrier. *Queue*, 6(1):36.

Tudoran, R., Costan, A., Antoniu, G., and Bougé, L. (2012). A performance evaluation of Azure and Nimbus clouds for scientific applications. In *Proceedings of the 2nd International Workshop on Cloud Computing Platforms - CloudCP '12*, pages 1–6, New York, New York, USA. ACM Press.

Walters, J. P., Chaudhary, V., Minsuk, C., Guercio, S., and Gallo, S. (2008). A comparison of virtualization technologies for HPC. In *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, pages 861–868.

Younge, A. J., Henschel, R., Brown, J. T., von Laszewski, G., Qiu, J., and Fox, G. C. (2011). Analysis of Virtualization Technologies for High Performance Computing Environments. In *2011 IEEE 4th International Conference on Cloud Computing*, pages 9–16. IEEE.