

# Desempenho de OpenStack e OpenNebula em Estações de Trabalho: Uma Avaliação com Microbenchmarks e NPB

Carlos A. F. Maron<sup>1,2</sup>, Dalvan Griebler<sup>1,2</sup>, Claudio Schepke<sup>3</sup>, Luiz Gustavo Fernandes<sup>2</sup>

<sup>1</sup> Laboratório de Pesquisas Avançadas para Computação em Nuvem (LARCC),  
Faculdade Três de Maio (SETREM) – Três de Maio – RS – Brasil

<sup>2</sup> Programa de Pós-Graduação em Ciência da Computação,  
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS),  
Grupo de Modelagem de Aplicações Paralelas (GMAP) – Porto Alegre – RS – Brasil

<sup>3</sup> Laboratório de Estudos Avançados (LEA),  
Universidade Federal do Pampa (UNIPAMPA) – Alegrete – RS – Brasil

carlos.maron@acad.pucrs.br, dalvan.griebler@acad.pucrs.br,  
schepke@unipampa.edu.br, luiz.fernandes@pucrs.br

**Resumo.** *Nuvens IaaS (Infrastructure as a Service) fornecem recursos computacionais (memória, rede, unidade de processamento e armazenamento) sob demanda para a execução de aplicações. Trabalhos que avaliam o desempenho de nuvens IaaS se limitam à camada de virtualização e desconsideram a análise do impacto das ferramentas de gerenciamento. Em contraste, este artigo investiga o impacto delas, buscando identificar se há influências ou diferenças entre OpenStack e OpenNebula. Para tal finalidade usou-se cargas de trabalho intensivas (microbenchmarks) e aplicações científicas paralelas. Estatisticamente, os resultados obtidos mostram que OpenNebula foi 11,07% melhor usando microbenchmarks e 8,41% considerando somente as aplicações.*

**Abstract.** *IaaS (Infrastructure as a Service) clouds provide on-demand computing resources (i.e. memory, networking, storage and processing unit) for running applications. Studies that evaluate the IaaS cloud performance are limited to the virtualization layer and ignore the impact of management tools analysis. In contrast, our research investigates the impact of them in order to identify if there are influences or differences between OpenStack and OpenNebula. We used intensive workloads (microbenchmarks) and scientific parallel applications. Statistically, the results demonstrated that OpenNebula was 11.07% better using microbenchmarks and 8.41% with scientific parallel applications.*

## 1. Introdução

A computação em nuvem popularizou-se como modelo de processamento no decorrer dos últimos anos. O principal motivo para tal expansão é sua arquitetura bem definida e voltada para o negócio, sendo composta por modelos de serviços tais como IaaS (*Infrastructure as a Service*), PaaS (*Platform as a Service*) e SaaS (*Software as a Service*). Isso despertou o interesse de usuários que necessitam de infraestrutura de computação (armazenamento, processamento), ambiente de desenvolvimento (plataformas de desenvolvimento) e *softwares* específicos [JTC1/SC38 2014a, JTC1/SC38 2014b].

Para implantar uma nuvem e oferecer serviços é necessário a integração de diversas tecnologias. Além da infraestrutura computacional, é fundamental a utilização de ferramentas de gerenciamento. Em uma nuvem IaaS, por exemplo, as ferramentas devem ser capazes de gerenciar recursos de armazenamento, servidores, roteadores e redes, todos virtuais. Muitas técnicas e ferramentas de virtualização (ex: KVM, LVM, Open vSwitch) possibilitam aos usuários acesso a uma infraestrutura completa em nuvem. Neste caso, a virtualização proporciona um aproveitamento otimizado da infraestrutura, alocando diversos servidores virtuais no mesmo *host*, visando a alta escalabilidade e disponibilidade.

Diversos trabalhos encontrados na literatura avaliam as tecnologias presentes na camada de virtualização [Xavier et al. 2013, Hwang et al. 2013, Regola and Ducon 2010]. Outros trabalhos avaliam o desempenho de nuvens IaaS (base da pilha), mas direcionam a análise sobre o virtualizador presente na nuvem e ignoram a camada das ferramentas de IaaS [Navaux et al. 2012, Mehrotra et al. 2012]. Por outro lado, a iniciativa desta pesquisa também se diferencia pela utilização de estações de trabalho. Provedores de nuvem pública como a Google e Microsoft vem usando *hardwares* de baixo custo e menos robusto para reduzir o consumo energético do *data center* e custo aos seus clientes<sup>1</sup>. Em vista disso, em [Maron et al. 2014] foi realizado um estudo das características de configuração e funcionalidades, provando ser possível o funcionamento em ambiente deste tipo. Porém, o desempenho até então era desconhecido.

As pesquisas de [Dukaric and Juric 2013, Maron et al. 2014, Thome et al. 2013] vêm apontando que ferramentas *open source*, para o gerenciamento de nuvens IaaS, possuem características e abordagens diferentes, o que reforça a hipótese destas impactarem no desempenho das execuções em nuvem. Em [Vogel et al. 2016a] uma revisão do estado da arte em relação ao suporte a flexibilidade e resiliência das ferramentas de nuvem IaaS são apresentadas. Adicionalmente, alguns testes de desempenho foram realizados sem de fato investigar os motivos de tais diferenças [Vogel et al. 2016b]. Além disso, uma infraestrutura de nuvem robusta (composta por servidores) foi usada. Assim, este trabalho tem como objetivo investigar o impacto de OpenStack e OpenNebula no desempenho em uma infraestrutura de estações de trabalho. Este impacto é avaliado através de *microbenchmarks*<sup>2</sup> que usam intensivamente um determinado recurso computacional e *benchmarks* de aplicações científicas. Dessa forma, mantendo um *cluster* homogêneo e serviços básicos idênticos para ambas as ferramentas conseguimos avaliar o impacto delas sobre as execuções em nuvem.

As contribuições deste artigo são:

- **Uma comparação do desempenho de *microbenchmarks* e aplicações paralelas em nuvem privada com OpenStack e OpenNebula.** Apresentamos um conjunto de resultados de aplicações científicas executadas em nuvem privada. Também avaliamos cada aspecto computacional de maneira restrita. Nossa discussão considerou o impacto que as ferramentas de nuvem exercem sobre esse conjunto de resultados. Essa análise possibilita ao usuário uma visão geral sobre o desempenho das ferramentas que gerenciam essas nuvens IaaS privadas.

---

<sup>1</sup><http://www.networkcomputing.com/cloud-infrastructure/are-cloud-based-virtual-desktops-future/695379268>

<sup>2</sup>Usado para avaliar desempenho de recursos isolados, como: rede (Ex.latência e *throughput*), I/O, entre outros.

- **Uma análise estatística visando a comparação dos resultados.** Comprovamos através dos testes estatísticos o quão significativamente diferente são os resultados entre as nuvens. Diferentemente de outros trabalhos, esta análise possibilita ao usuário uma escolha mais justa de uma ferramenta de código aberto para computação em nuvem, uma vez que existe pouco material científico disponível que faça este tipo de análise sem um viés comercial.

O artigo está organizado em 6 seções. Na Seção 2 posicionamos este trabalho em relação à literatura. Na Seção 3 discutimos brevemente o cenário de aplicações paralelas. Na Seção 4 são elencadas as ferramentas de gerenciamento de nuvem, destacando os aspectos importantes de OpenStack e OpenNebula. Em seguida, na Seção 5, é explicada a metodologia adotada nos experimentos. A Seção 6 exibe e discute os resultados obtidos. Ao final, na Seção 7, apresenta-se a discussão dos resultados e trabalhos futuros.

## 2. Trabalhos Relacionados

Diversas pesquisas avaliam o desempenho no cenário de computação em nuvem. Neste artigo, nos posicionamos em relação aos trabalhos voltados ao modelo de serviço IaaS. Grande parte desses artigos concentra-se na análise de tecnologias de virtualização (KVM, XEN, OpenVZ, LXC, VServer, Hyper-V e vSphere) ou nas plataformas de nuvem pública (Azure, Amazon EC2 e Rackspace).

A Tabela 1 sintetiza os trabalhos relacionados às tecnologias de virtualização e de plataformas de nuvem. As seguintes características são elencadas: **Objetivo:** investiga o desempenho alcançado, onde cada trabalho concentra-se em detalhes específicos de avaliação dos virtualizadores ou das nuvens para HPC<sup>3</sup>. **Plataforma:** referencia os virtualizadores ou ambientes de nuvens avaliados. **Ambiente Base:** descreve as configurações das máquinas utilizadas. **Benchmarks:** cita os *benchmarks* utilizados nos experimentos.

**Tabela 1. Panorama geral dos trabalhos relacionados.**

Trabalhos	Objetivo	Plataforma	Ambiente Base	Benchmarks e microbenchmarks
Regola 2010	Virtualizadores para HPC	Xen, OpenVZ, KVM, Amazon EC2	4x Dell R610, 2x Xeon E5520 2.27 GHz com 24 GB RAM	NPB3.3-OMP, NPB3.3-MPI, IOZone
Navaux 2012	Nuven para HPC	Amazon EC2, Azure, Rackspace	Intel Xeon 8 cores 15 GB de RAM	NPB-MPI, NPB-OMP
Mehrotra 2012	Nuven para HPC	Amazon EC2	Intel Xeon X5570 2.93 GHz com 23 GB de RAM	NPB-MPI, Enzo, Cart3D, MITgcm, DDSCAT
Xavier 2013	Virtualizadores	Xen, OpenVZ, LXC, VServer	4x intel xeon E5520 2.22 GHz com 16 GB de RAM	NPB-OMP, NPB-MPI, LinkPack, IOZone, NetPipe, STREAM
Hwang 2013	Virtualizadores	Hyper-V, KVM, vSphere, Xen	Intel Xeon 5160 3.00 GHz com 8 GB RAM	Bytemark, RAMSpeed, Bonnie++, FileBench, NetPerf, FreeBench
Trabalho Atual	Nuven Privada Ferram. IaaS	OpenStack (KVM), OpenNebula (KVM)	8x Intel i5 650 3.20 GHz com 4 GB RAM	LINPACK, IPerf, IOzone, STREAM, NPB-OMP, NPB-MPI

O trabalho de [Regola and Ducon 2010] defende que os benefícios da virtualização em ambientes HPC são obtidos com a maximização da largura de banda e com perdas mínimas de CPU e I/O. A avaliação de desempenho deteve-se à rede e ao disco nos virtualizadores KVM e Xen (*Hypervisor*) e OpenVZ (*container*). Os resultados indicaram

<sup>3</sup>High Performance Computing

que OpenVZ apresenta os resultados mais próximos ao ambiente nativo<sup>4</sup> em termos de desempenho de CPU, I/O e aplicações paralelas. Na plataforma Amazon EC2 os testes mostraram que a nuvem torna-se uma alternativa economicamente viável para soluções de HPC. Os autores concluem que, para HPC, I/O é um aspecto que ainda precisa ser melhorado para que haja um melhor proveito da virtualização.

Em [Navaux et al. 2012] é avaliada aplicações de HPC nos principais provedores de nuvem pública nos quesitos: desempenho, eficiência de custo e desenvolvimento. O estudo sugere que a computação em nuvem pode ser uma alternativa para eliminar a necessidade de manter *clusters* permanentes. Nos experimentos, foram executados os *benchmarks* NAS e realizada uma análise das características referentes ao desenvolvimento e custo. Os autores concluem que a nuvem pode se tornar uma alternativa viável para executar aplicações de HPC, com algumas desvantagens na questão do desenvolvimento. No entanto, uma análise das características das aplicações deve ser feita para escolher o provedor em que elas terão o melhor desempenho. Considerando o desempenho e o custo-benefício, os autores estimaram que ambientes de nuvem podem ser 27% e 41% mais eficientes (no custo) do que *clusters*.

Em [Mehrotra et al. 2012] ocorre a avaliação do desempenho de aplicações paralelas na nuvem Amazon EC2. Como esta infraestrutura de nuvem oferece instâncias computacionais especializadas para HPC, os autores avaliaram se o ambiente é adequado para este tipo de aplicações. Além dos testes e avaliações de desempenho usando NAS, foram executadas algumas aplicações reais que estressavam os recursos de forma específica. Os resultados foram comparados com os obtidos em centros especializados de HPC. Ao final, os autores concluíram que, embora a Amazon EC2 tenha diminuído a sobrecarga na camada de virtualização, o desempenho das aplicações é significativamente afetado em função da rede utilizada. A pesquisa aponta que plataformas de nuvem, como a Amazon, não são um ambiente adequado para aplicações de HPC que exigem grande poder computacional e de comunicação.

O artigo de [Xavier et al. 2013] investiga o desempenho da virtualização por *container*. A experimentação foi conduzida com o objetivo de testar o isolamento de desempenho de processamento, memória, disco e rede. Adicionalmente, foram executados testes com aplicações paralelas usando o *benchmark* NPB. Os resultados indicam que: (I) HPC somente irá tirar proveito da virtualização se a sobrecarga de CPU, memória, disco e rede for reduzida; (II) todas as tecnologias de virtualização avaliadas apresentam desempenho próximo do nativo; (III) a principal diferença entre os virtualizadores é a implementação, que resulta em um isolamento pobre e pouca segurança para ambientes de HPC. Os autores acreditam que sistemas baseados em *container* ainda estão imaturos, pois o processamento foi o único quesito com bom desempenho. Além disso, LXC foi o melhor virtualizador para HPC.

Em [Hwang et al. 2013] são avaliados os virtualizadores Hyper-V, KVM, vSphere e Xen. Os autores comentam que, embora os processadores atuais têm melhorado o desempenho em relação ao uso de tecnologias de virtualização, a sobrecarga ainda é comum quando máquinas virtuais competem pelo mesmo recurso. Neste sentido, o artigo realizou uma série de testes de desempenho no isolamento de processador, memória, disco e rede.

---

<sup>4</sup>Considera-se o ambiente nativo aquele que não possui camada de virtualização e ferramenta de administração de nuvem.

As conclusões obtidas a partir dos testes são de que não existe um virtualizador perfeito e de que diferentes aplicações irão se beneficiar dos diferentes virtualizadores, dependendo das necessidades de desempenho e de características específicas das aplicações.

Assim como a maioria dos trabalhos relacionados, este artigo visa contribuir com a análise de desempenho de computação em nuvem. Porém, nossa análise está direcionada no comportamento dos ambientes de nuvem privada com OpenStack e OpenNebula, acreditando que os resultados das execuções das cargas de trabalho apresentam diferenças. Nossa discussão é feita a partir das características das ferramentas que estariam influenciando os resultados. Portanto, diferentemente dos outros trabalhos, neste em questão são levados em conta características de infraestrutura e implantação idênticas (Seção 5) para efetuar uma comparação justa entre as ferramentas. Como forma de solidificar os resultados, a análise estatística propõe encontrar as diferenças significativas entre os resultados.

### 3. Aplicações Paralelas

A computação científica é uma importante área interdisciplinar de pesquisa. Diversas ciências naturais e exatas utilizam a computação científica para a resolução de problemas como a modelagem da aerodinâmica de veículos, a simulação de condições climáticas e o cálculo do deslocamento de partículas poluentes em um fluido. Tudo isso é possível graças à utilização de computadores de alto desempenho, que fornecem condições de processamento razoáveis em relação à magnitude dos problemas tratados, explorando o paralelismo das aplicações [Andrews 2001].

Uma versão concorrente ou paralela de uma aplicação pode ser desenvolvida tanto para ambientes de memória compartilhada como distribuída. Para tanto, algumas das mais conhecidas para programação nestes ambientes são *Open Multi-Processing* (OpenMP), que permite a criação implícita de *threads* em trechos de código, e *Message-Passing Interface* (MPI), uma norma amplamente difundida para a criação de processos comunicantes.

Aplicações científicas podem beneficiar-se do uso de um ambiente de nuvem, uma vez que é possível acessar um ambiente dinâmico e com grande capacidade de processamento [Navaux et al. 2012]. O modelo de nuvem agrega diversas características de outros modelos de processamento, como processamento de larga escala, computação sob demanda, compartilhamento de recursos, além da escalabilidade, otimização de recursos e elasticidade que destacaram-se nesses ambientes.

Muitas iniciativas ou tendências para aplicações de alto desempenho tem sido apresentadas [Venner 2009]. Propostas como *Big Data* e *Hexascale* além da interface oferecida em *Hadoop* precisam contar com abstrações do ambiente físico, para prover dinamismo e, ao mesmo tempo, estabilidade na execução das aplicações (tolerância a falhas e continuidade do processamento). Diante disso, uma infraestrutura de nuvem pode ser uma boa alternativa para todas estas questões, maximizando o uso dos recursos computacionais disponíveis.

Neste trabalho, como forma de representar diversos tipos de aplicações de alto desempenho, adotou-se os *benchmarks* na versão paralela do NAS [Bailey et al. 1991]. Através destes *benchmarks* é possível avaliar características clássicas de alto desempenho como ambientes de memória compartilhada e distribuída. Também é possível configurar

aplicações que avaliam a infraestrutura computacional, como processador, rede (*Gigabit*), memória física.

Acredita-se que ferramentas de administração de nuvens podem impactar de forma distinta no desempenho das aplicações paralelas. Neste sentido, a próxima seção detalha alguns aspectos das ferramentas de administração avaliadas neste trabalho.

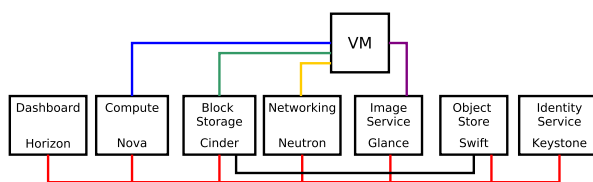
## 4. Ferramentas de Gerenciamento de Nuvens IaaS

Recursos computacionais de ambientes de nuvens podem ser oferecidos na camada do modelo de serviço IaaS. Para isso, existem diferentes ferramentas de gerenciamento que permitem aos usuários o acesso e a facilidade para controlar e utilizar estes recursos. Nesta seção são descritas as ferramentas avaliadas neste trabalho. Nossa escola foi baseada em [Thome et al. 2013], que avaliaram características operacionais de diversas ferramentas, onde OpenStack e OpenNebula foram consideradas as mais completas.

### 4.1. OpenStack

OpenStack é uma ferramenta para o gerenciamento de nuvens do tipo IaaS, responsável por controlar os recursos computacionais virtualizados. Do ponto de vista de infraestrutura como serviço, usuários podem acessar a interface WEB e configurar toda uma infraestrutura virtual. As funcionalidades básicas de criação e configuração são: máquinas virtuais, redes virtuais, roteadores, volumes de armazenamento e aplicação de regras de segurança.

A Figura 1 apresenta alguns dos principais componentes que a ferramenta OpenStack na versão IceHouse utiliza para administrar uma nuvem computacional. As linhas representam a conexão com os componentes e a VM (*Virtual Machine*). A seguir serão detalhados alguns desses componentes de acordo com [OpenStack 2016].



**Figura 1. Diagrama de componentes de OpenStack. Adaptado de [OpenStack 2016].**

O componente Nova gerencia a comunicação com os virtualizadores. Para isso, Nova forma um conjunto de vários sub-componentes, dos quais, os principais são: Nova-Scheduler (gerencia e aloca novas instâncias), Nova-API (exerce comunicação privilegiada entre componentes e comunicação com provedores externos), Nova-Compute (interage com o virtualizador, criando e terminando instâncias), Nova-conductor (gerencia interações com o banco de dados), Nova-consoleauth (gerencia autorizações de acessos), Nova-Cert (controla certificados X509), Nova-manage (serve para manutenção do Nova), Nova-novncproxy (realiza proxy VNC). Todos estes sub-componentes possuem funções ligadas ao gerenciamento das instâncias na nuvem e interagem com os virtualizadores.

Glance é responsável por controlar as imagens dos sistemas operacionais que serão provisionados em nuvem. Estas imagens são armazenadas pelo componente Swift.

Glance possui uma árvore de onze sub-componentes para seu funcionamento, por exemplo: Glance-manage, Glance-api, Glance-registry.

Neutron gerencia e oferece uma infraestrutura de rede. Para isso, possui 14 sub-componentes que gerenciam os níveis mais baixos desse serviço em uma nuvem, permitindo a criação de roteadores virtuais, controle de rotas e definição de agentes de roteamento externo.

Cinder gerencia o armazenamento da nuvem. O serviço permite criar unidades de armazenamento no formato LVM ou ZFS que podem ser anexadas nas instâncias virtuais (armazenamento em bloco). Swift também é um componente para gerenciar o armazenamento. Porém, é utilizado para dados não estruturados, por exemplo, no uso em grandes volumes de dados. Adicionalmente, OpenStack oferece o Trove, componente específico para provisionamento de banco de dados em nuvem.

Keystone implementa um sistema de segurança capaz de proteger as comunicações dos componentes da nuvem. Ele autentica e também autoriza a maior parte das solicitações realizadas pelos componentes.

É importante ressaltar que OpenStack possui componentes desenvolvidos por terceiros e que são agregados em sua estrutura, como: **RabbitMQ** - *middleware* responsável pela comunicação interna entre os componentes, utilizando o protocolo AMQP (*Advanced Message Queue Protocol*). **Open vSwitch(OVS)** - componente que implementa as interfaces de redes virtuais das instâncias. Em relação aos virtualizadores, OpenStack é compatível com KVM, LXC, vSphere, Xen e Hyper-V.

## 4.2. OpenNebula

OpenNebula é uma solução *Open Source* para a gerenciamento de ambientes de computação em nuvem, privadas, públicas ou híbridas. É compatível com as principais ferramentas de virtualização: KVM, VMware e Xen Server. Para a implantação desta ferramenta em um *cluster*, é necessário definir um *host* da infraestrutura como um *front-end*, o qual executará os principais serviços da ferramenta descritos a seguir e ilustrados na Figura 2: **ONED**: *daemon* principal de gerenciamento da ferramenta (rede, disco). **MMSCHED**: controle para alocação das máquinas virtuais entre os *hosts* físicos. **ONECCTD**: monitoramento de contas e recursos de usuários. **SUSTONE**: serviço da interface WEB da ferramenta. **EC2-query / OCCI**: gerencia APIs de comunicação com provedores públicas.

O protocolo SSH é parte fundamental para a comunicação entre os nós do cluster no OpenNebula, pois o ONED (OpenNebula *Daemon*) gerencia e monitora a infraestrutura através das conexões SSH. Recomenda-se o uso de NFS (*Network File System*) na instalação padrão do OpenNebula. Pois dessa forma, uma unidade de rede é compartilhada no *front-end* entre os nós do *cluster*. Assim, o armazenamento dos arquivos das máquinas virtuais é centralizado, oferecendo um serviço de *storage* como descreve a Figura 2. OpenNebula oferece outras alternativas compatíveis para o armazenamento, como: Lustre, GlusterFS, ZFS, GPFS e MooseFS.

Na questão da rede, OpenNebula usa o Dummy, que implementa *bridges* e trabalha com *drivers* para a configuração das redes e interfaces virtuais. Dummy não aplica nenhum tipo de controle ao tráfego de rede entre as instâncias. Além deste, OpenNebula

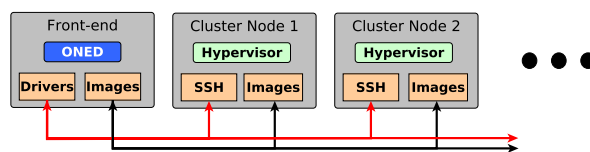


Figura 2. Diagrama básico do OpenNebula. Adaptado de [OpenNebula 2016].

permite a criação de redes virtuais através dos drivers fw, ebtables, 802.1Q, ovswhch e VMware [Toraldo 2013].

## 5. Metodologia de Avaliação

Neste trabalho foi utilizada a abordagem experimental para realizar a análise no isolamento de recursos e nas aplicações paralelas. A experimentação foi conduzida levando em consideração a execução de *benchmarks* e *microbenchmarks* e a análise estatística das métricas avaliadas. Para analisar o desempenho através de uma visão isolada, medimos separadamente o desempenho dos recursos de rede, memória e disco (*throughput*) e processador (capacidade de processamento). Para aplicações paralelas foram coletados os tempos de execução de cada *kernel* executado.

A infraestrutura utilizada é composta por 8 computadores idênticos, com processador Intel Core i5 650 3.20 GHz, memória RAM de 4 GB DDR3 1333 MHz, disco rígido de 500 GB, operando em SATA II e uma rede *gigabit*. Ubuntu *Server* 14.04 foi utilizado como sistema operacional em todas as instalações. Para configuração das ferramentas OpenStack (v.Icehouse) e OpenNebula (v.4.8.0), criou-se 2 *clusters* destinando 4 computadores para cada uma, onde ambas as ferramentas usaram o virtualizador KVM (v.2.0.0), Open vSwitch (v.2.0.2) e NFS (v.1.2.8). Deste modo, características de infraestruturas e implantação foram isoladas usando um ambiente idêntico. No ambiente de nuvem, as instâncias foram criadas utilizando a capacidade total do nodo. Assim, as características de cada uma foram: 4 vCPUs, 4 GB de RAM e disco virtual no tamanho de 10 GB.

Para a avaliação do isolamento de recursos, foram utilizados os seguintes *microbenchmarks*: **IPerf**: avaliação da rede, coletando os dados de *throughput* durante um tráfego de 10 s. **IOzone**: teste das unidades de armazenamento, considerando as funções *Read*, *Reread*, *Write*, *Rewrite* aplicado à um arquivo de 100 MB. **STREAM**: avaliação de memória, usando as funções *ADD*, *SCALE*, *TRIAD*, *COPY*. As políticas internas do STREAM endereçam uma matriz de  $2 \times 10^6$  elementos utilizando 45,8 MB da memória. Internamente o código é executado 10 vezes, e destes coleta-se o melhor resultado. **LINPACK**: teste do processador, usando uma matriz de  $6.000 \times 6.000$  elementos, para executar cálculos em ponto-flutuante.

Além de avaliar individualmente cada recurso de processamento, realizamos testes com aplicações de alto desempenho. Entende-se que estas aplicações exigem um nível maior de recursos do ambiente, simulando em proximidade cargas de trabalho reais. Para representar esse cenário de aplicações e servir como cargas de trabalho no ambiente, adotou-se os *benchmarks* disponibilizados na versão paralela do NAS [Bailey et al. 1991]. Com eles é possível avaliar características clássicas de alto desempenho como ambientes de memória compartilhada (OpenMP - *Open Multi-Processing*) e distribuída (MPI - *Message-Passing Interface*). A suíte escolhida foi NPB-3.3 (*NAS Parallel Benchmark*) compilada na classe B.



Os *kernels* executados foram: **IS** (*Integer Sort*): Avalia essencialmente o acesso aleatório de memória através de uma operação de ordenação. **EP** (*Embarrassingly Parallel*): usa tarefas paralelas sem dependências entre si. **CG** (*Conjugate Gradiente*): implementa um método numérico de resolução de sistemas lineares onde as matrizes são simétricas positivas-definidas. Há um acesso irregular de memória e padrão de comunicação. **MG** (*Multi Grid*): são feitas operações intercaladas sobre malhas de diferentes resoluções. Há distintos padrões de comunicação e uso intensivo de memória. **FT** (*Fourier Transform*): implementa a transformada rápida de Fourier 3D. Todas as *threads* ou processos se comunicam. **BT** (*Block Tri-diagonal*): resolve sistemas lineares esparsos tridiagonais em blocos. Exige comunicação entre processos vizinhos cartesianos. **SP** (*Scalar Penta-diagonal*): resolve sistemas lineares esparsos pentadiagonais. A comunicação cartesiana tem mais impacto que o processamento. **LU** (*Lower-Upper*): usa método de Gauss-Seidel para a resolução de sistemas lineares densos. Exige muito processamento e alguma comunicação.

Para a validação dos resultados foi utilizado o *software* estatístico SPSS [Field 2009] usando método de teste de hipótese com amostras de 40 execuções, comum em testes de *software* [Bussab and Morettin 2010]. A margem de confiança foi 95%, que indica se a variável *Sig* for menor que 0,05, os resultados da comparação são significativamente diferentes.

## 6. Resultados

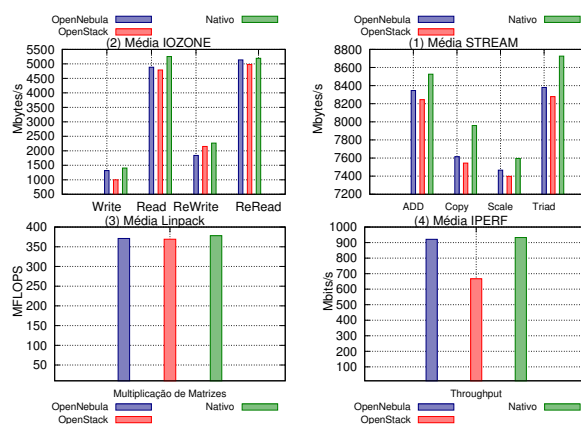
Esta seção apresenta os resultados obtidos na avaliação de desempenho e análise estatística da infraestrutura tanto no isolamento de recursos quanto das aplicações paralelas.

### 6.1. Desempenho dos *microbenchmarks*

A Figura 3 apresenta os resultados obtidos de largura de banda de acesso à memória (MB/s), disco (MB/s), rede (Mb/s) e processamento (MFlops). A sequência de colunas nos gráficos representam respectivamente os resultados obtidos por OpenNebula, OpenStack e ambiente nativo (sem virtualização).

O gráfico da Figura 3(1) demonstra o desempenho alcançado pelo *microbenchmark* STREAM em cada um dos ambientes. Para operações aplicadas pelo STREAM percebe-se uma perda no desempenho quando este é executado no ambiente gerenciado pelo OpenStack. O nível de controle de *software* aplicado pelo componente RabbitMQ da ferramenta torna o tempo de execução das operações maiores. Já em OpenNebula existe o componente ONED, que faz uma interface direta com o virtualizador.

Na Figura 3(2) são destacados os resultados obtidos nos testes de funções de armazenamento, avaliando escrita (*Write*), leitura (*Read*), reescrita (*Rewrite*) e releitura (*Re-read*). Na operação *Rewrite* o desempenho de OpenStack foi maior. Isso é justificado pela interface TAP (*Test Access Point*), que faz parte de uma *bridge* interna usada para a comunicação dos serviços de OpenStack. No caso da operação *Rewrite*, o privilégio de ter uma interface dedicada em OpenStack reflete o ganho nessa operação. As técnicas de reaproveitamento de cache em releituras e reescritas são naturalmente empregadas nas operações de disco. Dessa forma, as funções do IOzone são reaproveitadas pelos componentes das ferramentas, já que o armazenamento é distribuído e OpenStack tem componentes que otimizam a rede.



**Figura 3. Gráficos dos resultados no isolamento de recursos.**

O desempenho do armazenamento na nuvem ainda possui desafios, principalmente para a computação de alto desempenho, pois o acesso à disco compromete o tempo de execução das aplicações. Em nosso ambiente de testes, foram considerados recursos para melhorar o desempenho de acesso ao disco em ambas as nuvens. Neste caso, intimamente relacionado ao virtualizador, o recurso utilizado foi o *driver* virtIO do KVM, que paravirtualiza o acesso ao disco e torna as operações mais rápidas.

Na Figura 3(3) observa-se o desempenho da execução do algoritmo LINPACK. Neste cenário de execução, não foi possível identificar alguma característica das ferramentas que tenha influenciado as execuções. Contudo, acredita-se que os componentes do OpenStack em concorrência pelos recursos causaram essa intrusão no desempenho.

Na Figura 3(4) são exibidos os resultados de vazão de rede, para ambas as ferramentas usando o *microbenchmark* IPerf. OpenStack foi prejudicado devido a forma que ele implementa as interfaces virtuais das instâncias. O percurso de um pacote de rede ao sair de uma instância passa por 2 etapas até ser encapsulado no protocolo IP, gerando assim uma sobrecarga. A interface TAP no sistema operacional representa a conexão de rede da instância alocada na nuvem. Ela faz parte de uma *bridge* interna que agrega todas as interfaces TAP que são alocadas no nodo. A *bridge* interna possui um *link* com a *bridge* túnel que encaminha os pacotes para fora do nodo. Toda a comunicação entre as *bridges* interna e túnel são encapsuladas com protocolo GRE (*Generic Routing Encapsulation*). Entende-se que isso aumente o tempo de comunicação do pacote na rede do OpenStack, afetando o *throughput*. Na ferramenta OpenNebula cada instância tem uma interface virtual (VNET) que faz a comunicação direta.

Outro aspecto que tem prejudicado a comunicação na instância OpenStack foi a sobrecarga no *front-end*, pois o mesmo também foi utilizado como roteador do *cluster* OpenStack. Um ponto que também deve ser observado é a existência de componentes específicos para o gerenciamento da rede na ferramenta OpenStack. Neste caso, o Neutron é um serviço que realiza a gestão das redes virtuais da nuvem. Essa gestão permite que ele realize requisições aos *plugins* como o OVS e o utilitário netns (*network namespace*) do Linux. Em questões de infraestrutura, a utilização de um servidor específico para o nodo *network*, em alguns aspectos, poderia melhorar o desempenho de rede. Contudo, acabaria gerando um custo maior em equipamentos e consumo de energia.

A Tabela 2 apresenta a análise estatística da avaliação do isolamento dos recursos.

Para cada *micro-benchmark* são mostrados os resultados médios de execução e o desvio padrão, além de indicar se as diferenças de desempenho entre OpenStack e OpenNebula são significativas. Em todos os quesitos considerados houve diferença expressiva entre as duas ferramentas, uma vez que todos os valores de *Sig* foram menores que 0,05.

**Tabela 2. Resultados estatísticos OpenStack vs OpenNebula (microbenchmarks)**

Microbenchmark	Medida	Nativo		OpenStack ( $\Omega$ )		OpenNebula ( $\beta$ )		$\beta$ vs $\Omega$ <i>Sig.</i>
		$\bar{x}$ (40x)	$\sigma$ (40x)	$\bar{x}$ (40x)	$\sigma$ (40x)	$\bar{x}$ (40x)	$\sigma$ (40x)	
LINPACK	CPU	378,08	0,30	369,05	1,57	370,98	0,43	0,000
IPerf	Throughput	932,35	0,83	666,85	125,8	920,77	22,02	0,000
STREAM	ADD	8525,76	55,42	8244,39	45,71	8346,42	216,15	0,005
	COPY	7959,16	104,60	7544,98	58,08	7615,26	180,52	0,022
	SCALE	7595,65	40,76	7397,46	54,33	7466,60	205,35	0,027
	TRIAD	8725,63	72,46	8278,34	44,44	8379,81	205,45	0,005
IOzone	Write	1406,18	61,04	875,21	167,84	1317,75	351,19	0,000
	Rewrite	2265,48	49,05	2275,63	228,85	1839,34	151,79	0,000
	Read	5254,16	129,21	4528,18	323,15	4886,51	198,38	0,000
	Reread	5192,15	127,79	4820,10	195,09	5138,01	143,05	0,000

OpenStack teve melhor desempenho somente na operação *Rewrite* (21,21% melhor), enquanto OpenNebula apresentou melhores resultados no acesso a rede, memória, processamento e nas demais operações no teste de armazenamento. No geral, o ganho de OpenNebula representa em média 11,07% dos testes realizados. Nos testes de processamento e de rede, a diferença foi de 0,52% e 31,99%, respectivamente. Nos testes de memória a diferença fica entre 1,23% e 0,93%. Para os resultados de armazenamento, a média foi de 40,36%, 7,61% e 6,38% (*Write*, *Read*, *Reread*).

## 6.2. Desempenho em Aplicações Paralelas

Na Figura 4 e Figura 5 são apresentados os resultados do tempo de execução dos algoritmos NAS, utilizando os padrões de programação paralela OpenMP e MPI, respectivamente. Os testes usando OpenMP é instanciado 1, 2, 3 e 4 *threads*, enquanto que MPI são utilizados 1, 4, 8 (com exceção da suíte NPB-MPI SP e BT, com 9) e 16 processos. Em quase todos os casos, o tempo de execução decresce a medida que mais *thread*/processos são utilizados. Em algumas aplicações, devido as características dos algoritmos, o ganho de desempenho não é linear a medida que mais unidades de processamento são usadas.

Os gráficos da Figura 4 mostram que os *kernels* com maior independência entre as *threads* (gráfico EP) ou com maior carga de processamento e acesso a memória (principalmente gráficos FT e MG) apresentaram reduções significativas do tempo de execução a medida que mais *threads* são utilizadas. Resultados similares ocorrem quando os testes são realizados utilizando a interface MPI, conforme pode ser visto na Figura 5.

Para os demais casos, o desempenho é afetado pela sincronização entre as *threads* OpenMP ou pela comunicação entre processos MPI. O fato de cada processador ter dois *cores* físicos permite uma redução significativa no tempo de execução ao usar 2 *threads* ou processos. O uso de 3 *threads* nem sempre é uma boa alternativa, uma vez que existe concorrência das *threads* para acesso a ULA (unidade lógica e aritmética).

Os resultados apresentados na Figura 4 mostram que não existem grandes diferenças entre o desempenho das aplicações paralelas em memória compartilhada utilizando os dois ambientes de nuvem. No entanto, o desempenho de OpenNebula foi melhor nas aplicações OMP, com exceção de CG 2 e SP 2.

Os resultados apresentados na Figura 5 mostram que os tempos de execução são menores, em sua grande maioria, no ambiente OpenNebula. A única exceção expressiva

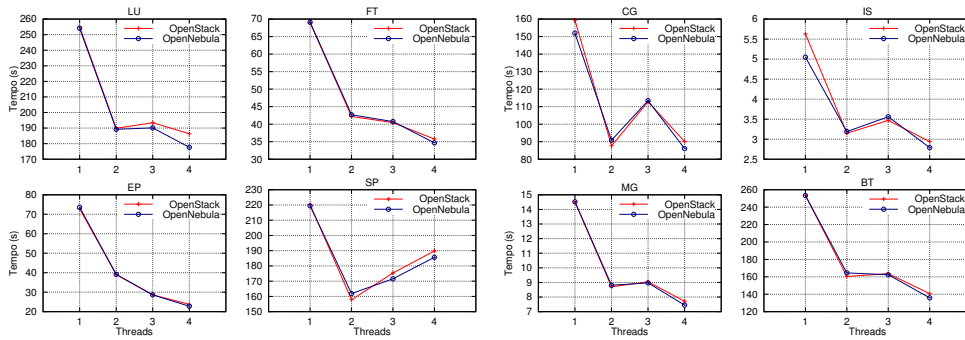


Figura 4. Análise dos resultados das aplicações usando OpenMP.

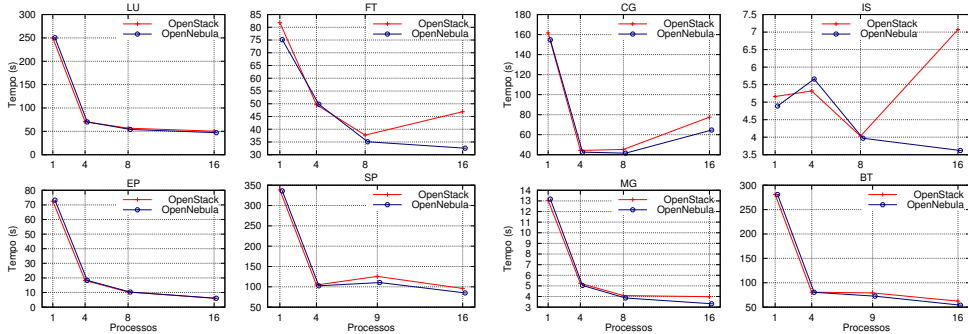


Figura 5. Análise dos resultados das aplicações usando MPI.

nos gráficos é o IS 4. De forma geral, a nuvem OpenNebula é expressivamente menor para os *kernels* que realizam um número maior de troca de mensagens, o que condiz com os resultados obtidos na avaliação da infraestrutura de rede (Figura 3(3)).

Tabela 3. Resultados estatísticos Nuvem vs Nuvem (NPB-MPI e OMP)

Bench.	Th.	OpenStack-OMP ( $\beta$ )		OpenNebula-OMP ( $\Omega$ )		$\beta$ vs $\Omega$
		$\bar{x}$ (40x)	$\sigma$ (40x)	$\bar{x}$ (40x)	$\sigma$ (40x)	
BT	1	252,74	1,07	249,47	3,02	0,000
	2	153,92	17,58	152,46	13,53	0,679
	3	163,33	0,49	162,51	0,68	0,000
	4	135,38	0,17	132,33	0,45	0,000
CG	1	156,32	20,53	140,68	34,07	0,026
	2	93,27	16,47	91,11	21,67	0,645
	3	114,05	0,13	109,21	13,97	0,035
	4	87,48	7,09	85,30	9,82	0,279
EP	1	72,27	0,35	72,09	0,30	0,021
	2	38,05	1,29	38,18	1,25	0,682
	3	28,73	0,35	28,41	0,38	0,001
	4	22,33	0,13	22,25	0,13	0,010
FT	1	68,25	0,11	67,80	0,36	0,000
	2	43,11	6,22	40,33	3,38	0,014
	3	41,11	0,29	40,92	0,24	0,005
	4	34,22	0,09	34,02	0,07	0,000
IS	1	5,60	0,11	4,91	0,66	0,000
	2	3,23	0,76	3,15	1,09	0,722
	3	3,80	0,07	3,53	0,24	0,000
	4	3,00	0,04	2,87	0,11	0,000
LU	1	255,94	0,59	253,56	1,71	0,000
	2	194,55	12,19	187,48	6,40	0,002
	3	195,90	0,74	195,21	1,21	0,004
	4	183,02	0,44	180,89	0,96	0,000
MG	1	14,35	0,01	14,30	0,03	0,000
	2	8,97	1,52	9,79	2,02	0,048
	3	9,10	0,02	9,08	0,04	0,006
	4	7,56	0,04	7,48	0,06	0,000
SP	1	220,30	0,27	218,39	0,98	0,000
	2	161,40	13,10	156,48	10,67	0,082
	3	176,85	0,84	175,28	1,22	0,000
	4	192,66	0,62	191,68	0,97	0,000
Proc.	1	280,96	1,80	280,89	2,70	0,889
	4	80,74	0,25	80,38	0,35	0,000
	9	79,17	1,10	72,51	0,68	0,000
	16	62,42	7,75	54,06	0,52	0,000
1	1	161,76	7,11	154,76	23,82	0,050
	4	44,29	0,36	42,60	0,50	0,000
	8	45,25	1,60	41,59	2,56	0,000
	16	77,52	1,59	64,65	2,01	0,000
2	1	72,54	0,34	73,22	0,44	0,000
	4	18,28	0,07	18,31	0,14	0,374
	8	10,36	0,63	10,34	0,54	0,849
	16	6,09	0,08	6,07	0,10	0,409
3	1	81,81	31,35	75,14	2,76	0,156
	4	49,65	0,91	49,80	0,16	0,320
	8	37,67	0,88	35,07	1,86	0,000
	16	46,89	1,04	32,54	0,55	0,000
4	1	5,16	0,31	4,89	0,51	0,008
	4	5,32	0,36	5,66	0,50	0,001
	8	4,03	0,27	3,97	0,40	0,387
	16	7,07	0,45	3,62	0,38	0,000
5	1	249,86	3,27	250,47	1,15	0,265
	4	70,62	1,98	70,12	1,49	0,240
	8	56,88	4,83	54,04	3,85	0,004
	16	50,30	2,58	47,18	2,37	0,000
6	1	13,05	0,07	13,17	0,10	0,000
	4	5,24	0,05	5,04	0,09	0,000
	8	4,09	0,36	3,86	0,43	0,017
	16	3,98	0,24	3,32	0,28	0,000
7	1	338,28	2,62	335,28	2,62	0,051
	4	104,46	0,48	102,55	0,63	0,000
	9	125,68	1,51	110,48	1,01	0,000
	16	95,92	4,10	84,92	0,60	0,000

A Tabela 3 apresenta os resultados de média, desvio padrão e diferenças significativas no desempenho das ferramentas, para cada um dos *kernels* considerados. Comparando-

se os resultados obtidos, percebe-se que não há diferenças significativas entre as ferramentas nas aplicações paralelas (destacados em verde) BT 2, CG 2 e 3 e 4, EP 2, IS 2 e SP 2, para as aplicações usando OpenMP. Em sistemas de memória distribuída, usando MPI, os casos BT 1, CG 1, EP 4 e 8 e 16, FT 1 e 4, IS 8, LU 1 e 4 e SP 1 também não apresentaram diferenças significativas. Por outro lado, os testes MPI MG 1, IS 4 e MG 1 foram os únicos casos em que o desempenho foi significativamente melhor em OpenStack. Desta forma, em apenas 6,25% (OMP 3,13% e MPI 9,38%) dos casos significativamente diferentes é que OpenStack foi melhor do que OpenNebula. Além disso, em 28,13% dos casos (OMP 21,88% e MPI 34,38%), em média, não é possível afirmar que existe diferença significativa no desempenho das aplicações paralelas.

A Tabela 3 mostra também que as diferenças no desempenho favorável ao OpenNebula nas execuções em MPI foram em média de 16,34%, com variação de 0,02% a 64,55% nos ganhos. Esta vantagem é compreendida através dos testes de isolamento de recursos, onde a nuvem OpenNebula teve ganhos mais expressivos nos testes de rede (31,99%). Já as diferenças favoráveis de OpenNebula com OMP foram em média de 4,08%, com variações de 0,22% a 13,13%.

## 7. Discussão dos Resultados e Trabalhos Futuros

Este artigo apresentou uma discussão sobre os resultados de desempenho obtidos em nuvens gerenciadas por OpenStack e OpenNebula implantadas em uma mesma infraestrutura. Inicialmente foram apresentados os trabalhos relacionados, posicionando esta pesquisa em relação às anteriormente efetuadas. Em seguida, as ferramentas de nuvens foram brevemente descritas. Ao final, apresentou-se os resultados obtidos a partir dos experimentos realizados.

Os ganhos de desempenho com os *microbenchmarks* (memória, rede, processamento e armazenamento (*Write, Read e Reread*)) executados em OpenNebula foram 11,07% melhores. Estes resultados são significativos considerando que foi utilizada uma mesma infraestrutura física, virtualizador e modo de implantação das ferramentas de gerenciamento. A escolha de cargas de trabalho para execução em nuvem, além de gerar tempo de processamento menor, interfere em outros aspectos, permitindo que mais tarefas ou usuários possam utilizar recursos de processamento. Além disso, podemos mencionar a diminuição do consumo energético, uma vez que há menos tempo de processamento/execução. Também há redução dos gastos com a implantação da infraestrutura da nuvem, já que, por exemplo, um número menor de unidades de processamento, em uma infraestrutura OpenNebula, pode atingir a mesma capacidade de processamento que uma infraestrutura maior, usando OpenStack.

Os resultados da infraestrutura ainda podem servir como apoio ao usuário sobre qual ferramenta escolher para sua implantação privada. Além disso, poderá ser feito otimizações (tempo e esforço) na etapa implantação, já que vários detalhes técnicos sobre as ferramentas são discutidos durante o trabalho.

Em relação às aplicações paralelas, o desempenho da rede apresentado no isolamento de recursos foi determinante para a vantagem da ferramenta OpenNebula nos testes com as aplicações MPI. Os resultados numéricos para OpenNebula foram em média 56,25% melhores e as aplicações usando OpenMP tiveram um benefício médio de 75%. A suíte NAS além de permitir emular o comportamento de diversas situações de

aplicações paralelas cotidianas, também serve para estimar o impacto no tempo de execução das aplicações nas ferramentas de gerenciamento de nuvem. Diante destes resultados, foi possível perceber que ferramentas para o mesmo propósito apresentam desempenho diferentes em uma nuvem privada.

Embora OpenNebula tenha se mostrado a melhor ferramenta em aspectos gerais, foi possível identificar que existem aspectos que podem ser otimizados com o objetivo de atingir melhores desempenhos, como outros modelos de implantação, ou formatos de discos e de redes. Em relação a OpenStack, soluções para reduzir o impacto no desempenho de rede, memória e processamento podem ser melhor investigados em função dos testes de isolamento de recursos e de atualizações subjacentes a versão IceHouse.

Como trabalhos futuros planeja-se: (I) avaliar diferentes domínios de aplicações (ex. corporativas, financeiras) para identificar limitações; (II) avaliar a escalabilidade e elasticidade dos recursos nas ferramentas; (III) Investigar os componentes das ferramentas OpenStack e OpenNebula para otimizações futuras. (IV) Incluir outras ferramentas de administração de nuvem, considerando também novos tipos de implantações voltados para o desempenho das aplicações submetidas em nuvem.

**Agradecimentos** Esta pesquisa foi realizada com o apoio do projeto HiPerfCloud<sup>5</sup> no LARCC<sup>6</sup>. Os autores agradecem o suporte financeiro da Abase Sistemas<sup>7</sup> e da Sociedade Educacional Três de Maio (SETREM)<sup>8</sup>.

## Referências

- [Andrews 2001] Andrews, G. R. (2001). *Foundations of Multithreaded, Parallel, and Distributed Programming*, volume 1. Addison-Wesley, Reading, Massachusetts.
- [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., et al. (1991). The NAS Parallel Benchmarks. *International Journal of High Performance Computing Applications*, 5(3):63–73.
- [Bussab and Morettin 2010] Bussab, W. d. O. and Morettin, P. A. (2010). *Estatística básica*. Saraiva.
- [Dukaric and Juric 2013] Dukaric, R. and Juric, M. B. (2013). Towards a Unified Taxonomy and Architecture of Cloud Frameworks. *Future Generation Computer Systems*, 29(5):1196–1210.
- [Field 2009] Field, A. (2009). *Discovering Statistics Using SPSS*. SAGE, Dubai, EAU.
- [Hwang et al. 2013] Hwang, J., Zeng, S., Wood, T., et al. (2013). A Component-Based Performance Comparison of Four Hypervisors. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 269–276. IEEE.
- [JTC1/SC38 2014a] JTC1/SC38, Technical Report, I. O. f. S. (2014a). ISO/IEC 17788:2014 Information technology - Cloud computing - Overview and vocabulary.

---

<sup>5</sup><http://hiperfcloud.setrem.com.br/>

<sup>6</sup><http://larcc.setrem.com.br/>

<sup>7</sup><http://www.abase.com.br/>

<sup>8</sup><http://www.setrem.com.br/>

- [JTC1/SC38 2014b] JTC1/SC38, Technical Report, I. O. f. S. (2014b). ISO/IEC 17789:2014. Information Technology-Cloud computing-Reference architecture.
- [Maron et al. 2014] Maron, C. A. F., Griebler, D., and Schepke, C. (2014). Comparação das Ferramentas OpenNebula e OpenStack em Nuvem Composta de Estações de Trabalho. In *14th Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS)*, pages 173–176, Alegrete, RS, Brazil. SBC.
- [Mehrotra et al. 2012] Mehrotra, P., Djomehri, J., Heistand, S., Hood, R., Jin, H., Lazanoff, A., Saini, S., and Biswas, R. (2012). Performance Evaluation of Amazon EC2 for NASA HPC Applications. In *Proceedings of the 3rd Workshop on Scientific Cloud Computing Date*, pages 41–50, Delft, The Netherlands. ACM.
- [Navaux et al. 2012] Navaux, P., Roloff, E., Diener, M., and Carissimi, A. (2012). High Performance Computing in the Cloud: Deployment, Performance and Cost Efficiency. In *Proceedings of the 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 371–378, Washington, DC, USA. IEEE Computer Society.
- [OpenNebula 2016] OpenNebula (2016). OpenNebula (Official Page) <<http://opennebula.org/>>. Last access in October, 2016.
- [OpenStack 2016] OpenStack (2016). OpenStack Roadmap (Official Page) <<http://openstack.org/software/roadmap/>>. Last access October, 2016.
- [Regola and Ducon 2010] Regola, N. and Ducon, J. C. (2010). Recommendations for Virtualization Technologies in High Performance Computing. pages 409 – 416, Indianapolis. IEEE.
- [Thome et al. 2013] Thome, B., Hentges, E., and Griebler, D. (2013). Computação em Nuvem: Análise Comparativa de Ferramentas Open Source para IaaS. In *11th Escola Regional de Redes de Computadores (ERRC)*, page 4, Porto Alegre, RS, Brazil. SBC.
- [Toraldó 2013] Toraldó, G. (2013). *OpenNebula 3: Cloud Computing*. Packt, Birmingham.
- [Venner 2009] Venner, J. (2009). *Pro Hadoop - Build Scalable, Distributed Applications in the Cloud*. Springer-Verlag, New York.
- [Vogel et al. 2016a] Vogel, A., Griebler, D., Maron, C. A. F., Schepke, C., and Fernandes, L. G. L. (2016a). Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack. In *24rd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 672–679, Heraklion Crete, Greece. IEEE.
- [Vogel et al. 2016b] Vogel, A., Maron, C. A. F., Griebler, D., and Schepke, C. (2016b). Medindo o Desempenho de Implantações de OpenStack, CloudStack e OpenNebula em Aplicações Científicas. In *16th Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS)*, pages 279–282, São Leopoldo, RS, Brazil. SBC.
- [Xavier et al. 2013] Xavier, M., Neves, M., Rossi, F., Ferreto, T., Lange, T., and Rose, C. D. (2013). Performance Evaluation of Container-based Virtualization for High Performance Computing Environments. In *21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 233–240, Belfast, UK. IEEE.