

**Reference:** Vogel, A.; Griebler, D. *Implantando, Avaliando e Analisando as Ferramentas para Gerenciamento de IaaS OpenStack, OpenNebula e CloudStack*. Laboratory of Advanced Researches on Cloud Computing (LARCC), Technical Report, 2016.

## Relatório Técnico de Pesquisa (Atividades de 2015)

Nome do Projeto:

# High Performance in Cloud (HiPerfCloud)

Avaliação dos Ambientes de Nuvem IaaS

**RT2: Implantando, Avaliando e Analisando as Ferramentas para Gerenciamento de IaaS OpenStack, OpenNebula e CloudStack**



ID do Documento:	LARCC-HiPerfCloud-RT2
Versão:	1.2
Autores:	Adriano Vogel, Dalvan Griebler
Objetivo:	Avaliação e análise das ferramentas de gerenciamento de nuvem CloudStack, OpenStack e OpenNebula
Tarefa:	Implantar, avaliar e analisar as ferramentas CloudStack, OpenStack e OpenNebula
Hardware:	3 <i>clusters</i> isolados foram criados usando 4 máquinas idênticas, cada uma com 24 GB de RAM (1333 MHz), processador Intel Xeon X5560 (quad-core 2.80GHz), discos SATA II (7200 RPM) e conectados em uma rede Gigabit (10/1000)
Ambiente:	Sistema Operacional (Ubuntu Server 14.04), Virtualizador (KVM e LXC), OpenStack (vers. Kilo), OpenNebula (vers. 4.8.0), CloudStack (vers. 4.5.2) Benchmarks de Isolamento (Iperf, IOzone, STREAM e LINPACK), Aplicações Paralelas (NPB-MPI e NPB-OMP) e de avaliação de rede (Hpcbench)
Softwares:	GNUPlot (Gráficos), Latex (Documentos)

---

Tarefa	Responsável	Instituição	Papel	Data
Criado por:	Dalvan Griebler	SETREM	Coordenador	09/01/2016
Editado por:	Adriano Vogel	SETREM	Pesquisador	09/02/2016
Revisado por:	Vera Lúcia Benedetti	SETREM	Colaboradora	09/03/2016
	Fauzi Shubeita	SETREM	Colaborador	09/03/2016
Aprovado por:	Dalvan Griebler	SETREM	Coordenador	15/03/2016
	Ildo Corso	ABASE	Colaborador	15/03/2016
Publicado:	LARCC	SETREM	Laboratório de Pesquisa	20/03/2016

## Log de Mudanças do Documento

<b>Versão</b>	<b>Autores</b>	<b>Instituição</b>	<b>Mudança</b>	<b>Data</b>
1	Dalvan Griebler	SETREM	Versão inicial	09/01/2016
1	Adriano Vogel	SETREM	Envio para revisão	09/02/2016
1	Adriano Vogel	SETREM	Envio para aprovação	27/02/2016
1	Adriano Vogel	SETREM	Versão final	15/03/2016
1.2	Dalvan Griebler	SETREM	Atualização no layout e informações	18/02/2017



## Lista de colaboradores internos e externos

A baixo é listado (em ordem alfabética) as pessoas que fizeram contribuições para este relatório técnico:

- Adriano Vogel (SETREM)
- Carlos A. F. Maron (SETREM)
- Claudio Schepke (UNIPAMPA)
- Dalvan Griebler (SETREM)



## Resumo Geral

O objetivo primário do **Projeto HiPerfCloud** (*High Performance in Cloud*) é avaliação de desempenho em ambientes de nuvens IaaS (*Infrastructure as a Service*) e analisar características de implantação e gerenciamento nas ferramentas disponíveis. Este documento apresenta a continuidade do RT1-2015 [23] e mostra novos resultados em implantações de nuvem privada baseadas em 3 ferramentas: OpenStack, OpenNebula e CloudStack.

## Contexto do Relatório

Este documento é o segundo Relatório Técnico *Implantação, Avaliação e Análise das Ferramentas para Gerenciamento de IaaS* relativo ao **Projeto HiPerfCloud** que apresenta resultados de infraestrutura (processador, memória, armazenamento e rede) e aplicações científicas executadas em ambientes de nuvem. Ainda, é apresentada uma análise da robustez das ferramentas e implantações em direção à redundância e alta disponibilidade para máquinas virtuais na nuvem.

## Estrutura do Relatório

Este documento inicialmente apresenta a estrutura geral. Posteriormente, computação em nuvem é contextualizada com as ferramentas de gerenciamento OpenStack, CloudStack e OpenNebula, seguida pela análise das características relacionadas ao suporte para flexibilidade e robustez das ferramentas. Ainda, as características dos ambientes implantados, experimentos e resultados são apresentados. Ao final, um estudo em direção de alta disponibilidade e redundância é apresentado e discutido.



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Visão Geral . . . . .	1
1.2	Terminologia . . . . .	1
1.3	Estrutura deste Documento . . . . .	2
<b>2</b>	<b>Computação em Nuvem IaaS</b>	<b>3</b>
2.1	Ferramentas <i>Open Source</i> para implantação de nuvens IaaS . . . . .	4
2.1.1	OpenStack . . . . .	4
2.1.2	OpenNebula . . . . .	4
2.1.3	Eucalyptus . . . . .	5
2.1.4	OpenQRM . . . . .	5
2.1.5	CloudStack . . . . .	5
2.1.6	Nimbus . . . . .	5
2.2	Taxonomia de ferramentas de IaaS . . . . .	6
2.3	Análise comparativa de ferramentas de IaaS . . . . .	6
2.3.1	Flexibilidade . . . . .	7
2.3.2	Resiliência . . . . .	7
<b>3</b>	<b>Desempenho em Nuvens Privadas IaaS</b>	<b>13</b>
3.1	Desempenho da Infraestrutura e Aplicações Científicas . . . . .	13
3.1.1	Implantações . . . . .	13
3.1.2	Metodologia dos testes . . . . .	15
3.1.3	Experimentos . . . . .	15
3.1.4	Resultados . . . . .	15
3.2	Avaliação do Desempenho de Rede em Ambientes de Nuvem . . . . .	17
3.2.1	Experimentos de Rede . . . . .	17
3.2.2	Desempenho de Rede . . . . .	17
<b>4</b>	<b>Um Modelo de Ambiente de Produção na Nuvem</b>	<b>21</b>
4.1	CloudStack . . . . .	21
4.2	GlusterFS . . . . .	21
4.3	Em Direção à Alta Disponibilidade . . . . .	22
4.3.1	Alta Disponibilidade baseada em alocação de VMs em <i>Clusters Homogêneos</i> . . . . .	22
4.4	Em direção a backup em nuvem privada . . . . .	22
<b>5</b>	<b>Conclusão</b>	<b>23</b>



# 1. Introdução

Este capítulo apresenta um olhar genérico e introdutório do que será discutido nesse documento, elencando o conteúdo dos capítulos e termos relevantes desse estudo.

## 1.1 Visão Geral

Neste documento ferramentas de código aberto para a implantação de nuvens do tipo IaaS são analisadas através de uma taxonomia e posteriormente as mais robustas são implantadas, e o desempenho das instâncias oferecidas pelas mesmas é analisado. Além do desempenho de aplicações, tópicos especiais (alta disponibilidade, armazenamento redundante, backup) para implantações de nuvem são discutidos.

## 1.2 Terminologia

- **Infraestrutura:** Representa os recursos de processamento: Memória RAM, armazenamento, rede e processador.
- **Aplicações Paralelas:** Área da computação de alto desempenho.
- **Benchmark:** Programa para teste específico de determinado recurso ou serviço.
- **Cluster:** Conjunto de computadores interligados por uma rede somando recursos.
- **OpenStack:** Ferramenta de código aberto para gerenciamento de infraestrutura de nuvem IaaS.
- **OpenNebula:** Ferramenta de código aberto para gerenciamento de infraestrutura de nuvem IaaS.
- **CloudStack:** Ferramenta de código aberto para gerenciamento de infraestrutura de nuvem IaaS.
- **Eucalyptus:** Ferramenta de código aberto para gerenciamento de infraestrutura de nuvem IaaS.
- **OpenQRM:** Ferramenta de código aberto para gerenciamento de infraestrutura de nuvem IaaS.
- **Nimbus:** Ferramenta de código aberto para gerenciamento de infraestrutura de nuvem IaaS voltado para ambientes científicos.
- **Iperf:** experimento para avaliação de rede.
- **IOzone:** experimento para avaliação de unidades de armazenamento e sistemas de arquivos.
- **STREAM:** experimento para avaliação da memória RAM.
- **LINPACK:** experimento para avaliação do processador.
- **NPB-MPI:** carga de trabalho composto por diferentes *kernels* de simulação de aplicações paralelas com bibliotecas MPI.

- **NPB-OMP**: carga de trabalho composto por diferentes *kernels* de simulação de aplicações paralelas com bibliotecas OMP
- **Hpcbench**: Experimento para avaliação do desempenho de rede, abrangendo protocolos TCP e UDP e possui diversos experimentos tanto de *throughput* como latência.
- **Taxonomia**: Método que estabelece critérios para classificar em algum determinado assunto.
- **Survey**: Método de pesquisa exploratório e quantitativo que faz um levantamento relacionado com algum tema selecionado.

## 1.3 Estrutura deste Documento

Este documento está organizado em cinco capítulos:

- Capítulo 1: Apresenta um visão geral deste documento.
- Capítulo 2: Nesta seção, encontra-se o referencial sobre a computação em nuvem e seus serviços, seguido pela apresentação de ferramentas e dos métodos usados para classificar e comparar as ferramentas de código aberto disponíveis.
- Capítulo 3: Apresenta a metodologia utilizada para a execução dos experimentos nos ambientes implantados e os resultados dos testes.
- Capítulo 4: Apresenta e discute sobre as implantações avançadas em direção a alta disponibilidade e redundância para evitar perdas de dados
- Capítulo 5: Conclusão do estudo a partir dos resultados e trabalhos futuros.

## 2. Computação em Nuvem IaaS

Com a consolidação das redes locais e posteriormente da rede global (internet) usuários individuais e corporações se tornaram capazes de acessarem um serviço computacional disponível em qualquer lugar do mundo. Com isso, o tráfego global de informação vem crescendo exponencialmente. Nos primórdios, recursos computacionais passaram a serem vendidos e disponibilizados através da internet, usando o conceito de computação em grade (*grid computing*), posteriormente já com o surgimento de tecnologias de virtualização, surgiu a computação em nuvem. Este paradigma combinava diversas tecnologias consolidadas (redes de computadores, virtualização, *grid e cluster computing*), e se tornou popular por abstrair a complexidade de configuração da infraestrutura e oferecendo máquinas virtuais sob demanda, que receberam o nome de instâncias. Os serviços de nuvem se popularizaram por oferecerem o pagamento de acordo com a utilização (dados armazenados, percentual de utilização de CPU, memória, tráfego de dados, etc) [3].

A venda de poder computacional para usuários oferecido por terceiros (provedores) passou a se chamar nuvem pública, pois qualquer um com cartão de crédito poderia em questão de minutos ter uma instância acessível, alocada em servidores ao redor do mundo e inacessível fisicamente [33]. O investimento inicial, que sempre foi um grande limitador de infraestrutura de TI por ser caro e sofrer desvalorização e deterioração, a nuvem tornou-se uma alternativa para a redução de custos, mais ágil e flexível provisão de recursos e gerenciamento de serviços de TI. Conseqüentemente, vem ocorrendo um aumento significativo na utilização de serviços da computação em nuvem, por usuários domésticos e setores corporativos, educacionais, governamentais e organizações comunitárias. A melhor utilização dos recursos, economia energética e eficiência pode colaborar para o planeta no conceito de TI verde [5].

Com o passar dos anos e aumento da utilização da nuvem pública diversas preocupações surgiram, principalmente as relacionadas com privacidade, segurança dos dados e desempenho. Armazenar dados corporativos vitais e confiar todo o poder de processamento e armazenamento à terceiros se tornou potencialmente perigoso. Além disso, diversas instituições possuíam hardware e ambientes de TI próprios. Além das preocupações com segurança da informação, diversas pesquisas mostraram preocupações com desempenho nesses ambientes. O uso de virtualização diminui o poder computacional e em alguns casos combinações equivocadas de tecnologias e *drivers* de infraestrutura degradam ainda mais o desempenho. Além disso, em ambientes de nuvem pública o hardware é compartilhado por máquinas virtuais de clientes distintos, cada um com suas cotas, mas mesmo competindo por recursos e uma VM podendo prejudicar o desempenho das demais. Por esses motivos surgiu a chamada nuvem privada, a qual é implantada dentro de uma instituição usando sua infraestrutura e fornecendo recursos computacionais exclusivamente para os usuários locais. Diversas ferramentas de gerenciamento de infraestrutura virtual estão disponíveis, algumas pagas e outras de código aberto e sem custos.

Nuvem privada também se tornou uma alternativa competitiva e largamente utilizada, e surgiu ainda a nuvem híbrida que é uma combinação entre nuvem pública e privada. Dessa forma, frequentemente serviços primordiais e delicados são executados em uma nuvem implantação de nuvem privada enquanto demais serviços secundários são migrados para instâncias de nuvem pública, pois é alternativa barata e flexível. Independente do tipo de nuvem, quando a provisão é de poder computacional (CPU, memória, armazenamento, rede, sistemas operacionais) faz parte do tipo de serviço de nuvem mais popular, IaaS (*Infrastructure as a Service*). Existem outros serviços de nuvem reconhecidos como padrões, PaaS (plataforma como serviço) e SaaS (Software como serviço), as plataformas ou software são executadas na nuvem dentro de um ambiente de IaaS, o que resulta em dependência de recursos e na grande importância dos serviços IaaS.

## 2.1 Ferramentas *Open Source* para implantação de nuvens IaaS

Atualmente existem diversas ferramentas de código aberto para gerenciamento de infraestrutura virtual de nuvem, algumas dessas soluções são usadas para implantação de nuvens privadas e outras suportam a configuração de um provedor de nuvem pública [35, 14, 38, 39]. Nesse documento foram consideradas todas as ferramentas de código aberto para nuvens IaaS. A praticidade do uso da virtualização que possibilitou a execução de vários sistemas operacionais no mesmo *hardware* se tornou um desafio no ponto de vista do gerenciamento e utilização. A virtualização que é voltada para o *hardware*, para construir um ambiente de nuvem recebeu uma camada de *software* adicional, que é representada pelas ferramentas de IaaS, simplificando e centralizando o gerenciamento de complexas infraestruturas computacionais.

As ferramentas de IaaS são comparadas usando uma taxonomia específica para essas soluções. A análise leva em conta o fato que diversas tecnologias necessárias em camadas diferentes são necessárias para implantação de ambientes de nuvem eficientes e confiáveis. Dessa forma, o comparativo é focado em robustez, que é o resultado da combinação do suporte para flexibilidade e resiliência encontrado nas ferramentas de IaaS. A seguir as ferramentas são apresentadas.

### 2.1.1 OpenStack

OpenStack é uma ferramenta *open source* altamente conhecida no cenário de IaaS, isso ocorre principalmente por seu projeto ter sido lançado pelas respeitadas Rackspace e NASA, e em seguida diversas gigantes do mundo da tecnologia uniram-se ao projeto. Um dos seus principais objetivos é a interoperabilidade e flexibilidade de implantações de nuvem, possuindo mais de 40 APIs que são componentes independentes que estão disponíveis para instalação. A escolha desses componentes é totalmente customizável de acordo com a demanda de cada arquiteto de nuvem [28].

Alguns componentes se destacam por serem imprescindíveis para uma ambiente de nuvem. Por exemplo, o keystone controla a autorização e autenticação de toda a infraestrutura virtual (demais componentes, serviços, usuários, etc) e precisa ser configurado e “amarrado” aos componentes para a nuvem funcionar [13]. O componente glance que oferece um repositório de imagens para a criação de instâncias na nuvem. A conectividade da nuvem pode ser implementada através do componente nova-network (básico, usando as *bridges linux*) ou usando o neutron (avançado, rede como serviço) [9]. O armazenamento para as instâncias pode ser implementado usando o componente Cinder (armazenamento em blocos LVM) ou usando o robusto swift (armazenamento em objetos). Outro componente relevante é o horizon que oferece interface gráfica (GUI) para usuários e administradores da nuvem. Diversos outros componentes podem ser adicionados a nuvem de forma sistemática [19] e toda a comunicação entre os serviços ocorre usando um mensageiro instantâneo, sendo o RabbitMQ o mais popular.

### 2.1.2 OpenNebula

OpenNebula surgiu como um projeto de código aberto para virtualização de *data centers* e provisão de serviços de nuvem. Uma característica específica é buscar por uma implantação simplificada e ao mesmo tempo eficiente [26], oferecendo ainda customização para usuários e desenvolvedores.

Essa ferramenta possui uma arquitetura modular, iniciando com os *drivers* básicos que implementar controle sob a virtualização (Libvirt, NFS, dummy) e no núcleo da ferramenta são implementadas as rotinas mais importantes (controle de serviços e usuários, interfaces de rede, monitoramento da infraestrutura, escalonador, transferência entre os nodos, gerencia de

armazenamento, entre outros) [36]. A maioria das tarefas são realizadas através da interação do componente oned com o sistema operacional nativo.

### 2.1.3 Eucalyptus

Eucalyptus um longo acrônimo de *Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems* [12] é considerado um *bundle* para implantação de ambientes de nuvem, que busca melhorar o gerenciamento da infraestrutura e provisão de serviços. Um aspecto único dessa ferramenta é possuir APIs voltada para integração com Amazon EC2, que diferentemente das demais ferramentas oferecem apenas um suporte para nuvem híbrida.

Eucalyptus possui uma arquitetura voltada para escalabilidade [22]. As principais partes de uma implantação com essa ferramenta são o controlador de nodo, de *cluster* e o de armazenamento. O controlador de nodo [2] é um serviço executado em cada nodo que aloca recursos para as instâncias e monitora a utilização. O controlador de *cluster* é geralmente executado no servidor que gerencia a nuvem, controlando e balanceando as requisições dos clientes. Ainda, o controlador e servidor de armazenamento oferece volumes virtuais para as instâncias e geralmente é um servidor dedicado.

### 2.1.4 OpenQRM

OpenQRM é uma solução para nuvem dividida em dois produtos, versão comunitário (*open source e grátis*) e *enterprise* (*open source e paga*). Este documento considera para fins de comparação a versão comunitária, que é uma ferramenta de simples instalação e que busca alta disponibilidade através da redundância [27].

[37] cita que OpenQRM possui uma arquitetura modular e suporta módulos de gerenciamento avançados (monitoramento, desempenho, escalabilidade). Os serviços são implementados na forma de *plugins* e possui uma interface gráfica para gerenciamento (administrador de nuvem) e um portal (para clientes).

### 2.1.5 CloudStack

Cloudstack [6] é um projeto para nuvens privadas e públicas focado em alta disponibilidade e flexibilidade. A infraestrutura possui 2 componentes imprescindíveis, o gerente e o agente de nuvem [32]. Enquanto o gerente controla a infraestrutura virtual e é instalado no servidor principal, o agente é instalado nos nodos escravos que são agrupados em clusters.

Essa solução possui uma interface gráfica completa, onde toda a infraestrutura pode ser montada, o isolamento entre usuários pode ocorrer através de zonas de disponibilidade, grupos de segurança ou através da rede (Vlan, túneis GRE, etc). o armazenamento dos volumes das instâncias é feito na zona primária e precisa de uma zona de armazenamento secundária para armazenar imagens de sistemas operacionais, *templates* e *snapshots*.

### 2.1.6 Nimbus

Nimbus [25] é uma ferramenta para nuvens IaaS focada em implantações de clusters para o meio científico. Nessas implantações os usuários instanciam máquinas virtuais através de um repositório de imagens (Cumulus) que também controla o armazenamento nos volumes virtuais. Essa ferramenta por ser de código aberto, busca atrair desenvolvedores para flexibilidade e escalabilidade do sistema.

A arquitetura dessa plataforma possui diversas partes [21]. Os clientes interagem com a nuvem através de interfaces de linha de comando (CLI), e podem usar *framework* compatíveis

com a nuvem pública da Amazon. Nimbus possui também um controlador de recursos que usa SSH para se comunicar com os nodos do *cluster* e máquinas virtuais.

## 2.2 Taxonomia de ferramentas de IaaS

Taxonomia pode ser definida como a ciência da categorização e classificação baseada em métodos pré-definidos e estruturados [20]. O [11] apresenta uma proposta de taxonomia para análise e comparação de ferramentas para implantação de nuvens IaaS. Dessa forma, a taxonomia conceitual foi aplicada nas ferramentas escolhidas para analisar o suporte a flexibilidade nas implantações de cada ferramenta.

A taxonomia [11] é dividida em camadas e as camadas são formadas por itens 2.1. Na base da taxonomia tem-se a camada de abstração de recursos que interage com a virtualização e oferece recursos virtuais (processamento, armazenamento e conectividade) para a nuvem. A camada de núcleo de serviços gerencia os clientes e aloca recursos através de serviços (repositório de imagens, medição e utilização, essa camada conta ainda com o escalonador de recursos da nuvem que controla principalmente a alocação de VMs.

A camada de gerenciamento [30, 31] é responsável por controlar as operações do ambiente de nuvem, principalmente através de grupos de usuários e do gerenciamento de recursos através de interfaces (gráficas, APIs ou CLIs). Outros aspectos do gerenciamento de nuvem que podem ser destacados é o suporte à elasticidade e orquestradores, para alocação em massa de recursos. Já na camada de suporte serviços adicionais são oferecidos, tais como banco de dados, transferências e mensagens entre os servidores, nuvens e redes.

A camada de controle implementa políticas de utilização do ambiente de nuvem. Os acordos de nível de serviço (SLA) garantem critérios e métricas de qualidade e controle, e a medição verifica quanto dos recursos está sendo utilizado. A camada de serviços agregados oferece opções extras como migração de VMs, e soluções para alta disponibilidade e portabilidade. Outra camada vital é a de segurança, que autentica, autoriza e verifica os recursos e usuários da nuvem.

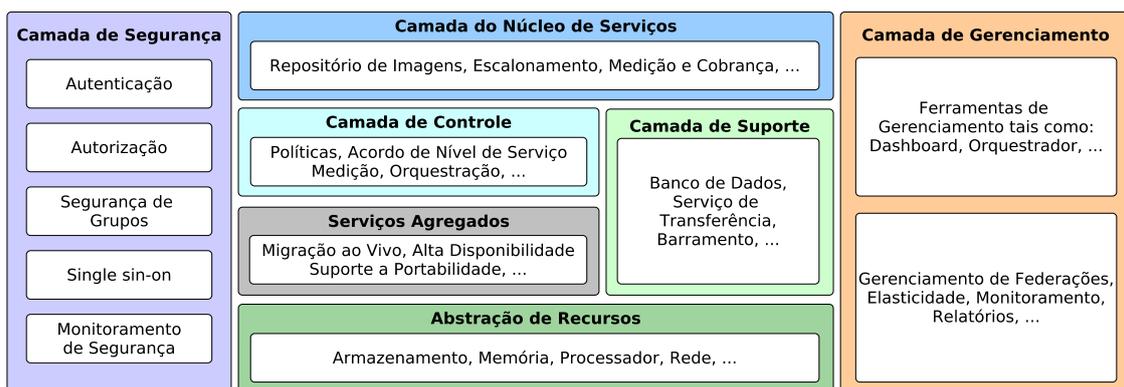


Figura 2.1: Taxonomia conceitual de IaaS. Adaptado de [11]

## 2.3 Análise comparativa de ferramentas de IaaS

As ferramentas de IaaS possuem objetivos e arquitetura diferentes, por isso se presume que possuam também contrastes no suporte para tecnologias e aplicações. Neste estudo a análise foi voltada para a robustez das ferramentas, que é dividida em flexibilidade e resiliência [34]. Usando métodos específicos, as próximas apresentam uma análise desses aspectos.

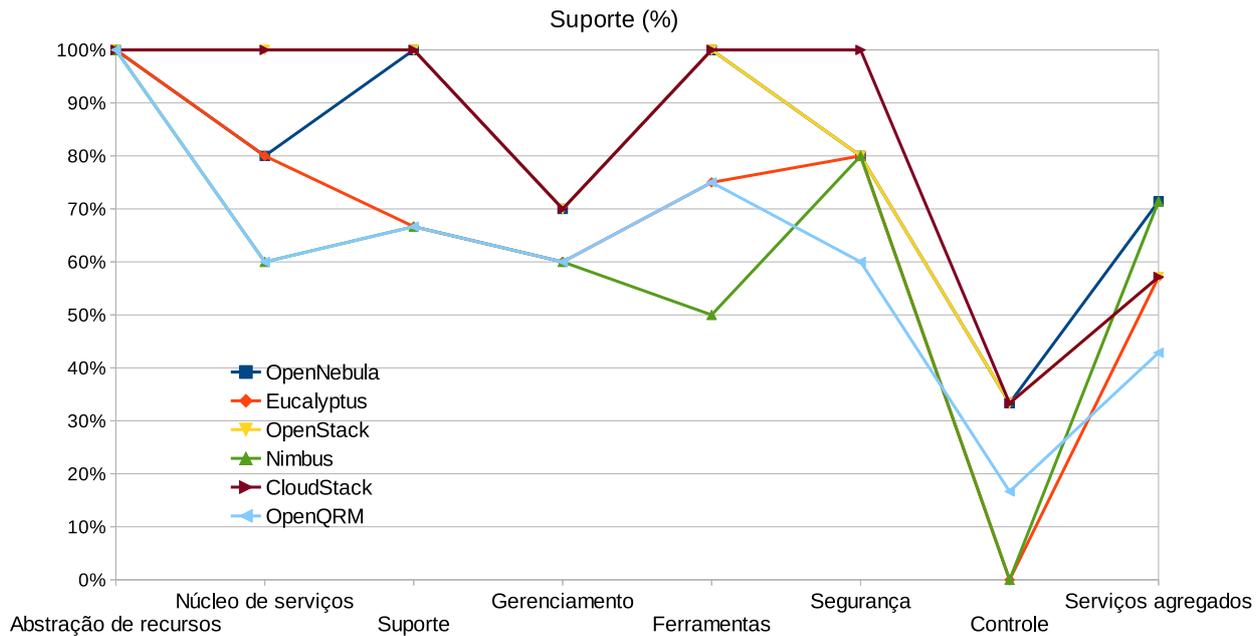


Figura 2.2: Flexibilidade das ferramentas de IaaS(%).

### 2.3.1 Flexibilidade

A flexibilidade de um sistema computacional é a capacidade de suportar diversos componentes, tecnologias e aplicações [3], esse suporte é relevante para atender as diferentes demandas de usuários e aplicações. As tabelas 2.1 e 2.2 apresentam a análise das ferramentas, o caractere "/" significa que naquele item não foi encontrada referência de suporte para as tecnologias relacionadas, a palavra "interno" significa que existe suporte, porém não está claro qual componente implementa, ou é do núcleo da ferramenta. Outra distinção usada foi a palavra "externo" que significa que o suporte pode ser suprido facilmente a partir da integração de uma tecnologia totalmente compatível.

A Figura 2.2 sumariza os níveis percentuais de flexibilidade de cada ferramenta, que é calculado a partir dos resultados das tabelas 2.1 e 2.2. A taxonomia é dividida em camadas que são formadas por itens, o resultado da figura mostra de acordo com o numero de itens suportados por cada ferramenta em cada camada o relativo percentual (de 0 a 100%), por exemplo, para uma ferramenta atingir 100% em alguma camada precisa ter compatibilidade como todos os tópicos.

Analisando os resultados de flexibilidade, de um modo geral as ferramentas apresentam um suporte pobre nas camadas de controle, serviços e gerenciamento. As ferramentas para implantação de ambientes de nuvem supostamente deveriam aumentar suas preocupações com a flexibilidade para as implantações, pois muitas apresentam um foco comercial forte e acabam deixando de lado o aumento da compatibilidade com as tecnologias com o cenário de constante mudanças. CloudStack teve o nível mais alto de compatibilidade seguido por OpenStack e CloudStack.

### 2.3.2 Resiliência

O suporte computacional para resiliência é a habilidade de adaptação para mudanças e em caso de falhas manter a disponibilidade dos serviços[4]. Em ambientes de nuvem a possibilidade de manter os serviços disponíveis é fortemente relacionada com as tecnologias de infraestrutura

<b>Suporte</b>	<b>OpenNebula</b>	<b>Eucalyptus</b>	<b>OpenStack</b>
<b>Abstração de recursos</b>			
Computador	Oned	Interno	Nova
Armazenamento	Interno	Walrus	Object storage (Swift) /Block Storage (Cinder)
Volume	Interno	Storage Controller	Nova-Volume
Rede	Virtual Network Manager	Interno	Neutron/Nova-network
<b>Núcleo dos serviços</b>			
Identidade	Interno	IAM API	Keystone
Escalonador	Scheduler	Cluster Controller	Nova-scheduler
Repositório de imagens	Interno	Interno	Glance
Cobrança	/	/	ceilometer
<i>Logging</i>	Interno	Interno	Interno
<b>Suporte</b>			
<i>Bus</i> mensageiro	Interno/RabbitMQ	/	RabbitMQ
Banco de dados	sqlite/MySQL	Sqlite/HSQldb	MySQL/Galera/MariaDB/MongoDB
transferência	Interno	Node Controller	Nova Object store/cinder
<b>Gerenciamento</b>			
Gerenciamento de recursos	Interno	Interno	Nova
Gerenciamento de federação	/	/	/
Gerenciamento de Elasticidade	Auto-scaling	Elastic Load Balancing	Elastic Recheck
Usuários e grupos	Interno	Interno	Interno
SLA	/	/	/
Monitoramento	probe/ssh/OneGate	externo	externo
Relatório	code reporting	/	/
Gerenciamento de incidentes	/	/	externo
Gerenciamento de energia	externo	externo	Blueprint driver
Aluguel	externo	externo	externo
<b>Ferramentas</b>			
CLI	OpenNebula CLI	Euca2ools	OpenStack (CLI)
APIs	Public cloud and Plugins	Public cloud and Plugins	Public cloud and Plugins
<i>Dashboard</i>	Sunstone(Admin UI, User UI)	Admin UI, User UI	Horizon(Admin UI)
Orquestrador	onflow	/	heat
<b>Segurança</b>			
Autenticação	Basic Auth/OpenNebula Auth/x509 Auth/LDAP	LDAP/ AWS IAM	LDAP/Tokens(APIs)/X.509/HTTPD
Autorização	Auth driver	Interno	Keystone
Grupos de segurança	Interno	Interno	Interno
<i>Single sign-on</i>	/	/	/
monitoramento de segurança	externo	externo	externo
<b>Controle</b>			
controle de SLA	/	/	/
monitoramento de SLA	/	/	/
Medição	externo	/	ceilometer
Política de controle	/	/	/
Serviços de notificação	/	/	/
Orquestração	OneFlow	/	heat
<b>Serviços agregados</b>			
Zonas de disponibilidade	Interno	Interno	Interno
Alta disponibilidade	externo	externo	externo
Suporte híbrido	Amazon EC2/Microsoft Azure/IBM	Amazon AWS	HP Helion/Amazon EC2/IBM
<i>Live migration</i>	Interno	Interno	Interno
Portabilidade	/	/	/
Contextualização de imagens	one-context	/	/
Aplicações virtuais	/	/	/

Tabela 2.1: Flexibilidade(a).

<b>Suporte</b>	<b>Nimbus</b>	<b>CloudStack</b>	<b>OpenQRM</b>
<b>Abstração de recursos</b>			
Computador	workspace	Libcloud	interno
Armazenamento	Cumulus	interno	interno
Volume	interno	interno	interno
Rede	workspace Control	interno	interno
<b>Núcleo dos serviços</b>			
Identidade	/	IAM plugin	/
Escalonador	interno	interno	interno
Repositório de imagens	interno	interno	Image Shelf
Cobrança	/	CloudStack Usage	/
<i>Logging</i>	Workspace-service	interno	interno
<b>Suporte</b>			
<i>Bus</i> mensageiro	/	interno/RabbitMQ	/
Banco de dados	interno	MySQL	Postgres/MySQL
Transferência	Workspace-service	interno	interno
<b>Gerenciamento</b>			
Gerenciamento de recursos	interno	interno	interno
Gerenciamento de federação	/	/	/
Gerenciamento de Elasticidade	/	Elastic Load Balancing	/
Usuários e grupos	interno	interno	interno
SLA	/	/	/
Monitoramento	externo	externo	externo
Relatório	/	/	/
Gerenciamento de incidentes	/	interno	interno
Gerenciamento de energia	externo	externo	externo
Aluguel	externo	externo	externo
<b>Ferramentas</b>			
CLI	interno	cloudmonkey	interno
APIs	Public cloud and Plugins	Public cloud and Plugins	Public cloud and Plugins
<i>Dashboard</i>	/	Admin UI	Admin UI
Orquestrador	/	cloudstack Cookbook	/
<b>Segurança</b>			
Autenticação	X.509	SAML/LDAP	LDAP
Autorização	interno	SAML	interno
Grupos de segurança	interno	interno	/
Single sign-on	/	externo	/
Monitoramento de segurança	externo	externo	externo
<b>Controle</b>			
controle de SLA	/	/	/
monitoramento de SLA	/	/	/
Medição	/	/	/
Política de controle	/	/	/
Serviços de notificação	/	interno	interno
Orquestração	/	cloudstack Cookbook	/
<b>Serviços agregados</b>			
Zonas de disponibilidade	EC2 driver	interno	/
Alta disponibilidade	externo	externo	externo
Suporte híbrido	Amazon EC2	Amazon EC2	Amazon AWS
<i>Live migration</i>	interno	interno	interno
Portabilidade	/	/	/
Contextualização de imagens	Nimbus context	/	/
Aplicações virtuais	/	/	/

Tabela 2.2: Flexibilidade(b).

<b>Suporte</b>	<b>OpenNebula</b>	<b>Eucalyptus</b>
Virtualização Completa e Bare Metal	Xen, KVM, Vmware	VMware, KVM, Xen
Tecnologias de armazenamento	NFS, ssh, ceph	ceph, OSG
Formatos de discos	qcow2, vmfs, ceph, lvm, fslvm, raw, dev	qcow2, LVM, raw, vmdk
Rede	dummy, ebttables, VLAN, OVS, vmware	VLAN, bridge, DHCP
SO	Ubuntu, Debian, RedHat, SUSE, CentSO	CentOS, RHEL
Virtualização em Contêiner	Não	Não
Interface gráfica (GUI)	Sim	Sim
Storage de objetos	Sim	Sim
<b>Suporte</b>	<b>OpenStack</b>	<b>Nimbus</b>
Virtualização Completa e Bare Metal	Hyper-V, VMware, Xen, KVM, VirtualBox	Xen, KVM
Tecnologias de armazenamento	LVM, Ceph, Gluster, NFS, ZFS, Sheepdog	local, EC3
Formatos de discos	LVM, qcow2, raw, vhd, vmdk, vdi	qcow2
Rede	Neutron, and B.Switch, Brocade, OVS, NSX, PLUMgrid	DHCP, ebttables
SO	Debian, Ubuntu, RHEL, CentOS, Fedora, Suse	Debian, Ubuntu, RedHat, Gentoo, SUSE
Virtualização em Contêiner	Sim	Não
Interface gráfica (GUI)	Sim	Não
Storage de objetos	Sim	Não
<b>Suporte</b>	<b>CloudStack</b>	<b>OpenQRM</b>
Virtualização Completa e Bare Metal	Hyper-V, Xen, KVM, VMware, VirtualBox	KVM
Tecnologias de armazenamento	NFS, SMB, SolidFire, NetApp, Ceph, LVM	LVM, NFS, GlusterFS
Formatos de discos	LVM, VMDK, VHD, qcow2,	raw, qcow2
Rede	bridge, VLAN, DHCP, DNS, NVP, BigSwitch, OVS	bridge, VLAN, DHCP, DNS, TFTP
SO	Debian, Ubuntu, RHEL, CentOS	Debian, Ubuntu, RHEL, CentOS, OpenSuSE
Virtualização em Contêiner	Sim	Não
Interface gráfica (GUI)	Sim	Sim
Storage de objetos	Sim	Não

Tabela 2.3: Resiliência de ferramentas de IaaS.

envolvidas, essas tecnologias se propriamente configuradas que tornarão a tolerância à falhas possível [18]. Do ponto de vista das ferramentas, elas precisam suportar tecnologias atuais e relevantes para a implantação de ambientes complexos. Na tabela 2.3 as ferramentas são analisadas quanto ao suporte para tecnologias subjacentes de acordo com método de critérios.

O suporte à técnicas de virtualização se refere as tecnologias que são suportadas tanto em virtualização completa ou para-virtualização. Cada virtualizador utiliza meios específicos para emular recursos computacionais e fazer a conversão dos binários entre o hardware e as camadas de software, por isso é relevante o suporte de diversas soluções.

A solução de virtualização usada também impacta nas tecnologias de armazenamento, tipos de redes e formatos de discos suportados, pois cada virtualizador suporta tecnologias específicas. Os sistemas operacionais suportados impactam na possibilidade ou não de se usar virtualização em contêiner, que cria máquinas virtuais que compartilham o mesmo *kernel* do ambiente nativo, o isolamento de recursos ocorre geralmente via usuários e processos. As ferramentas de IaaS necessitam de *drivers* e bibliotecas para interagirem com tecnologias de contêiner. Ainda, o *storage* de objetos é uma forma robusta e inovadora de oferecer armazenamento para usuários, os discos são tratados como objetos e distribuídos de forma escalável pela rede para processamento e tratamento de grande volumes de dados e requisições de aplicações intensivas [8].

A Figura 2.3 apresenta o valor percentual de cada ferramenta nos tópicos. Os resultados da Tabela 2.3 que de acordo com o número de tecnologias suportadas por cada ferramenta em relação ao total de tecnologias apresenta o percentual, por exemplo o 100% ocorre quando todas

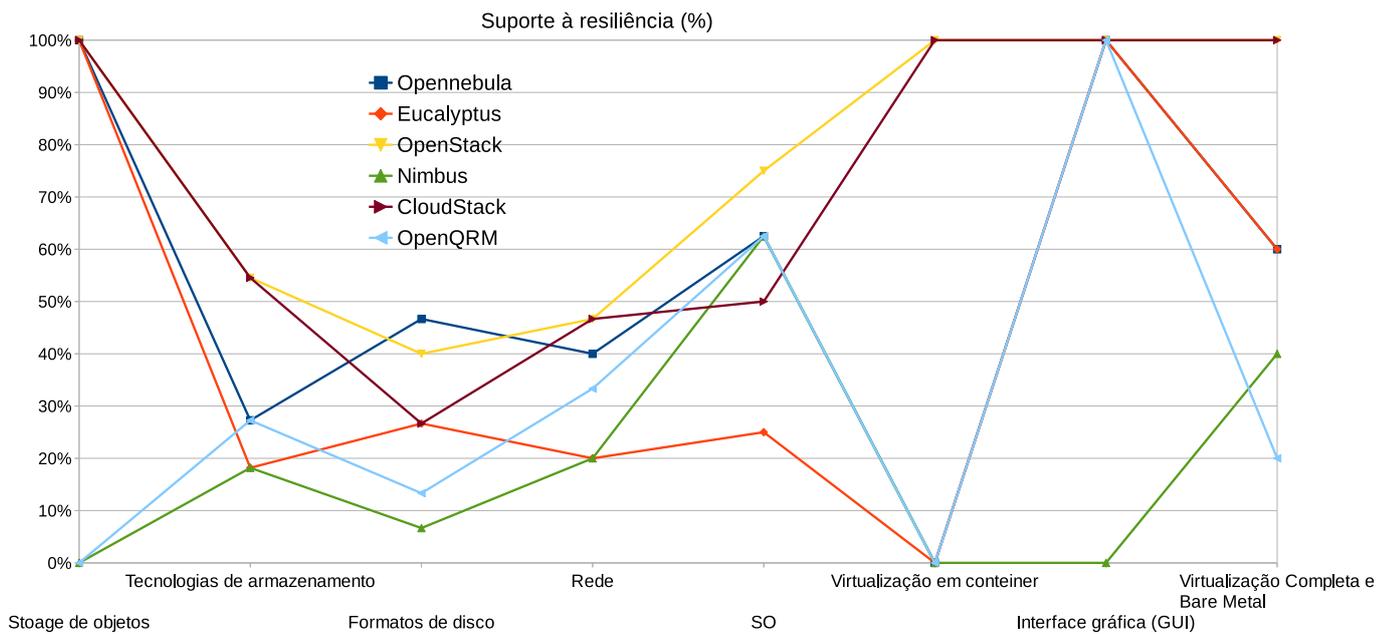


Figura 2.3: Resiliência das ferramentas de IaaS(%).

as tecnologias de algum tópico são suportadas por alguma ferramenta.

A ferramenta OpenStack alcançou os melhores resultados no suporte à resiliência, pois suporta diversas tecnologias de virtualização, armazenamento e redes. CloudStack aparece em seguida, e ambas suportam *storage* de objetos, virtualização em contêiner (usando LXC).



## 3. Desempenho em Nuvens Privadas IaaS

Como apresentado com Capítulo 2, as ferramentas para implantação de ambientes de nuvem suportam diversas tecnologias e configurações. Nesse seção, ambientes de nuvem usando ferramentas distintas foram implantados e o objetivo principal é comparar os resultados e apresentar um viés inovador nos resultados. Na Seção 3.1 são apresentados resultados da infraestrutura e aplicações científicas, e na Seção 3.2 são apresentados os resultados de diversos experimentos de rede (vazão e latência) nos ambientes de nuvem.

### 3.1 Desempenho da Infraestrutura e Aplicações Científicas

#### 3.1.1 Implantações

Para a execução de experimentos, três *clusters* isolados foram criados usando 4 máquinas idênticas, cada uma com 24 GB de RAM (1333 MHz), processador Intel Xeon X5560 (quad-core 2.80GHz), discos SATA II (7200 RPM) e conectados em uma rede Gigabit (10/1000). No ambiente nativo e nas instâncias foi utilizado o sistema operacional Ubuntu *Server* 14.04 (kernel 3.19.0). Optou-se por implementar os ambientes usando virtualização completa através do KVM (versão 2.0.0) e as ferramentas de nuvem OpenStack versão Kilo, OpenNebula 4.12 e CloudStack 4.5.2. Como foi utilizado 4 lâminas independentes nas 3 implantações distintas, os discos foram particionados usando *multiboot*.

A arquitetura das implantações foi similar. Usando as diferentes ferramentas de IaaS, cada ambiente teve um servidor agindo como gerente de nuvem, o qual controla a infraestrutura (recursos e usuários) e aloca as instâncias nos nodos. Controlados pelo gerente de nuvem, o virtualizador e agente de nuvem foram instalados em cada servidor (nodo) que ofereciam os recursos virtuais para as instâncias de nuvem. Ambas as implantações utilizaram os volumes distribuídos pela rede, OpenNebula e CloudStack usando o protocolo NFS (Network File System) e imagens QCOW e OpenStack usou o protocolo iSCSI (Internet Small Computer System Interface) e imagens LVM. Em ambas as implantações a conectividade entre as instâncias foi usando as *Bridges Linux* e o *driver* VirtIO. Para aumentar o poder de processamento, em cada servidor foi alocada apenas uma instância que tinha todos os recursos disponíveis.

#### Implantação da Ferramenta OpenStack

Para criação de instâncias virtuais e execução de experimentos com a ferramenta OpenStack, foi configurado um servidor como controlador de nuvem (armazenamento, rede, serviços, etc) e os nodos de nuvem nos quais as instâncias são alocadas e executadas. Nessa implantação os seguintes componentes do OpenStack e serviços foram instalados no controlador de nuvem:

- Servidor NTP- Para manter sincronizado os horários com os nodos da nuvem.
- Banco de dados - MySQL para armazenar informações da nuvem.
- Servidor RabbitMQ - Para coordenar operações e comunicação entre os serviços da nuvem.
- Keystone - Serviço de identidade que gerencia a autenticação de toda a nuvem.
- Glance - Servidor de imagens e *templates* de SO para os usuários da nuvem

- Nova - Gerenciamento de recursos
- Cinder - Oferece armazenamento em blocos para as VMs.
- Nova-Network - Implementa as redes virtuais para as instâncias
- Horizon - Oferece uma interface gráfica para gerenciamento da infraestrutura.

Nos nodos da nuvem foi instalado os pacotes de rede e configuradas as *bridges* e o componente *nova-compute* que interage com o nodo controlador da nuvem e com o *Libvirt* para execução dos serviços da nuvem. Detalhes adicionais da instalação, configuração e utilização da nuvem está disponibilizado como apêndice desse documento.

## Implantação da Ferramenta OpenNebula

O ambiente de testes OpenNebula foi instalado sob o sistema operacional nativo Ubuntu 14.04.3 amd64 e na versão do *kernel* Linux 3.19.0. As especificações dos servidores se encontra na seção 3.1.1 e foi usada uma partição de 600 GB configurada a partir de um RAID 5 (exemplo `/dev/md0p1`), tanto nos *frontend* quanto nos nodos da nuvem.

Um dos primeiros passos para a instalação do OpenNebula é atualizar o sistema e configurar as *bridges* Linux e o endereço IP de cada servidor. A versão instalado do OpenNebula foi a 4.12.0, por era a mais recente no momento da instalação. Nesta implantação, adotou-se o padrão recomendado pelo manual de instalação do OpenNebula no Ubuntu *Server* 14.04, utilizando o virtualizador KVM.

Os repositórios oficiais do OpenNebula para a versão foram adicionados para a *sources list* do sistema operacional. O servidor principal que gerencia a nuvem OpenNebula é chamado de *frontend*, a instalação pode ser feita pelos repositórios, que baixa e configura as bibliotecas, pacotes e *drivers* necessários. Entre os principais está o pacote OpenNebula, o OpenNebula-sunstone que é a interface web de gerenciamento e o servidor NFS que é o modo convencional de fornecer para o *cluster* virtual.

Nos nodos da nuvem, é instalado o pacote *opennebula-node* que comunica com o *Frontend* e também onde o cliente dos diretórios é montado. A interação com a nuvem pode ser através do *Sunstone* ou pelas interfaces de linha de comando (CLI). Diversas customização e configurações avançadas podem ser feitas, seguindo a documentação oficial [29].

## Implantação da Ferramenta CloudStack

Para comparar o desempenho entre as nuvens distintas as implantações são semelhantes, a instalação com a ferramenta CloudStack seguiu o modelo das demais ferramentas, com um servidor de controla a nuvem e os nodos da nuvem onde as VMs são alocadas. Na ferramenta CloudStack isso ocorre através do componente *cloudstack-management* e *cloudstack-agent* respectivamente.

A instalação do servidor que controla a nuvem foi iniciada com os pacotes básicos do sistema operacional, instalação do servidor NTP e do pacote *bridge-utils* (para implementar as interfaces de rede *bridges*). Após foi instalado o banco de dados MySQL para controlar as entradas do ambiente e os papéis da nuvem com o banco de dados. Em seguida é instalado o componente *cloudstack-management* que baixa todos os pacotes necessários para o gerenciamento da nuvem.

O nodo gerente da nuvem também foi usado para oferece armazenamento às instancias, nessa implantação foi instalado o servidor NFS e pelos menos dois diretórios precisam ser exportados (para a zona primária e secundária de armazenamento).

Nos nodos da nuvem é imprescindível a instalação do componente *cloudstack-agent* que baixa todos os pacotes necessários para a nuvem (EX: Libvirt, KVM, QEMU). Em cada nodo os diretórios NFS exportados no servidor precisam ser montados para serem acessíveis às VMs

executadas na nuvem. A interface gráfica pode ser acessada pelo servidor gerente da nuvem, e requer algumas configurações iniciais (Criação de redes, *clusters*, zonas, armazenamento).

### 3.1.2 Metodologia dos testes

A metodologia usada se baseou na forma com que a literatura lida com os mesmo experimentos computacionais, o renomados *micro-benchmarks* escolhidos foram executados 40 vezes e os resultados apresentam a média dos resultados e a comparação entre os ambientes de nuvem.

### 3.1.3 Experimentos

Os experimentos executados para analisar o desempenho da infraestrutura foram os seguintes:

- **LINPACK**: esse experimentos executa tarefas de uso exaustivo de CPU através de cálculos para resolução de matrizes[10], que foram do tamanho de 8000x8000.
- **STREAM**: nesse teste a banda de memória é medida usando vetores de kernel (*Add, Copy, Scale, Triad*) [24], usando blocos de dados maiores que o sistema de *cache*.
- **IOzone**: esse experimento avalia o desempenho de I/O do sistema de arquivos através de quatro operações, escrita, leitura, reescrita e releitura [16], as taxas de transferência foram analisadas usando um arquivo do tamanho de 5 GB.
- **IPerf**: esse teste faz uma medição básica de transferências de rede [17].

O desempenho de aplicações científicas foi avaliado nas três implantações de nuvem usando o renomado NAS [1] com os seguintes *kernels*:

- **Embarrassingly Parallel (EP)**: essa aplicação é caracterizada por independência de cada processo.
- **Multi Grid (MG)**: faz uso exaustivo de memória e comunicação através de operações sequenciais.
- **Integer Sort (IS)**: faz uso de memória de forma randômica com operações de ordenação.
- **Fourier Transform (FT)**: essa aplicação faz uso de comunicação para as operação de *Fourier*.

### 3.1.4 Resultados

Com a implantação dos ambientes planejados e criação das instâncias de nuvem foi possível a execução dos experimentos escolhidos na metodologia.

#### Desempenho da Infraestrutura

Os experimentos de infraestrutura foram executados em quatro ambientes distintos e isolados: ambiente nativo e instâncias de nuvem das ferramentas OpenNebula, OpenStack e CloudStack. A Figura 3.1 apresenta os resultados dos experimentos nos ambientes implantados.

De modo geral os resultados demonstraram perdas pequenas de desempenho comparando o ambiente nativo com o virtualizado, um exemplo são os resultados de processamento com o Linpack que foram próximos entre os ambientes. Por outro lado, os resultados de desempenho do armazenamento com o experimento IOzone tiveram mais variação, o que mostra a variação

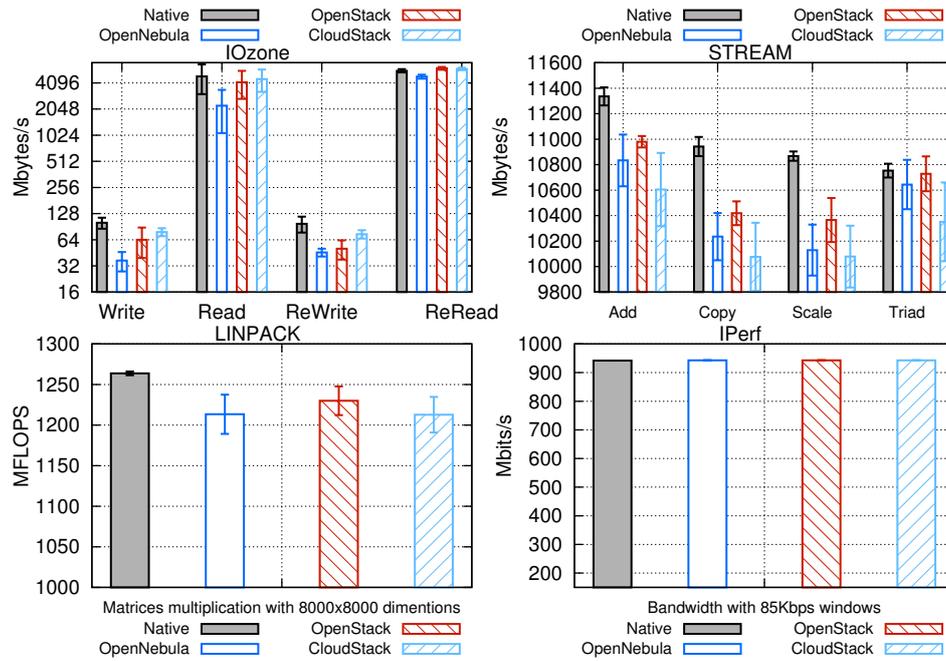


Figura 3.1: Desempenho da Infraestrutura.

devido a carga de trabalho. Os resultados de discos com a ferramenta OpenNebula foram piores que os demais ambientes, e nas operações de leitura e Releitura os resultados não foram muito diferentes, os contrastes ficaram nas operações de escrita e reescrita.

Os experimentos com memória foram um pouco melhores nas instâncias da ferramentas em comparação com OpenNebula e CloudStack. Por outro lado, o desempenho de rede foi muito próximo entre todos os ambientes.

## Desempenho de Aplicações Científicas

A Figura 3.2(A) mostra os resultados de desempenho de alguns *kernels* OMP, esses experimentos também foram executados em uma máquina do ambiente nativo e instâncias das ferramentas OpenStack, OpenNebula e CloudStack. Os resultados foram quase os mesmos nos diferentes ambientes, sem diferenças significantes.

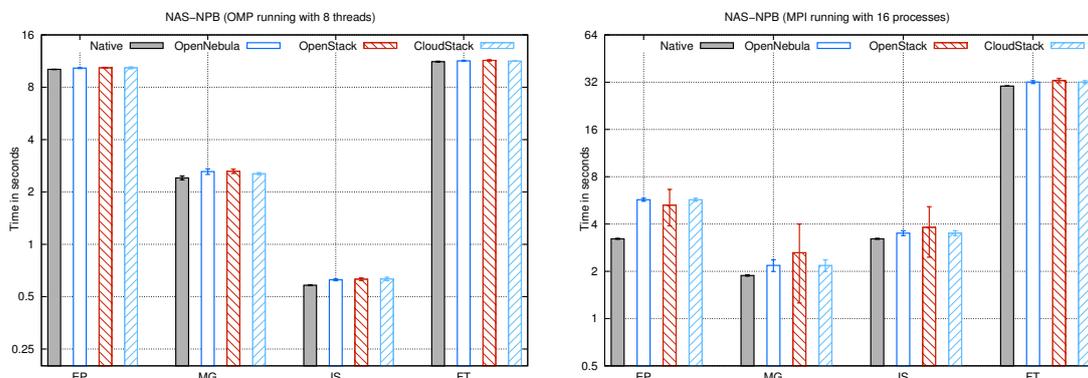


Figura 3.2: Desempenho NAS.

Também foram executados *kernels* do NAS-MPI nos ambientes implantados, esses experimentos foram executados repetidamente entre duas máquinas usando todos os recursos de processamento (até 16 processos) e o mesmo nos ambientes de nuvem, entre duas instâncias de

cada ferramenta. A Figura 3.2(B) apresenta os resultados dos *kernels* NAS, o ambiente nativo teve desempenho melhor que as máquinas virtuais. As instâncias executadas na ferramenta OpenStack tiveram um desvio padrão o que prejudicou o desempenho, enquanto CloudStack e OpenNebula tiveram um desempenho levemente melhor. Porém, as diferenças muito pequenas, o que demonstra que existem perdas de desempenho em ambientes virtualizados que usam memória distribuída enquanto com memória compartilhada o desempenho sofre um impacto menor.

## 3.2 Avaliação do Desempenho de Rede em Ambientes de Nuvem

Um aspecto relevante que tem um papel essencial para a computação em nuvem são as redes de computadores. Isso ocorre porque atualmente o processamento é distribuído colaborativamente entre nodos de *clusters*, sem conectividade isso seria impossível. Além disso, o diferencial da computação em nuvem é oferecer os recursos computacionais como serviços, o que só é possível graças as redes de computadores. Consequentemente, o desempenho e disponibilidade de aspectos de rede são muito importantes para garantir qualidade dos serviços e isso requer análises e entendimento.

### 3.2.1 Experimentos de Rede

As implantações dos ambientes nuvem foram semelhantes ao apresentado na Seção 3.1, a principal diferença é que os experimentos de rede foram também executados usando a virtualização em contêiner com LXC. Isso é possível através da integração entre a ferramenta de IaaS CloudStack e com servidores usando LXC para hospedar instâncias. Os experimentos foram executados ainda no ambiente nativo e com o virtualizador KVM com as ferramentas OpenStack, CloudStack OpenNebula. O *benchmark* Hpcbench [15] foi escolhido para execução com os seguintes experimentos:

- **Throughput Exponencial:** Esse experimento inicia com o tamanho de mensagem de 1 Byte e o tamanho do pacote é aumentado exponencialmente. Tamanhos diferentes impactam no *throughput* devido ao número de transferência necessárias, com tamanhos de mensagens maiores o *throughput* deve aumentar também.
- **Throughput Unidirecional:** Esse é o teste mais simples e mais usado em redes, sendo um nodo envia e outro recebe os pacotes.
- **Throughput Bidirecional:** Nesse experimento, dois nodos (cliente e servidor) enviam e recebem pacotes simultaneamente.
- **Latência:** Esse experimento considera o tempo de transferência de um pacote, quando é enviado, o tempo que demora até ele retornar.

### 3.2.2 Desempenho de Rede

A execução dos experimentos foi entre duas máquinas de cada ambiente implantado, e os resultados são apresentados relacionados com as transferências entre esses dois nodos, usando protocolos TCP e UDP. A Figura 3.3 apresenta os resultados nos cinco ambientes, fica evidente que tamanhos de pacotes pequenos degradam o *throughput*, que só aumenta com pacotes maiores. O ambiente nativo teve os melhores resultados seguido pelo ambiente CloudStack-LXC e pelas instâncias usando o KVM.

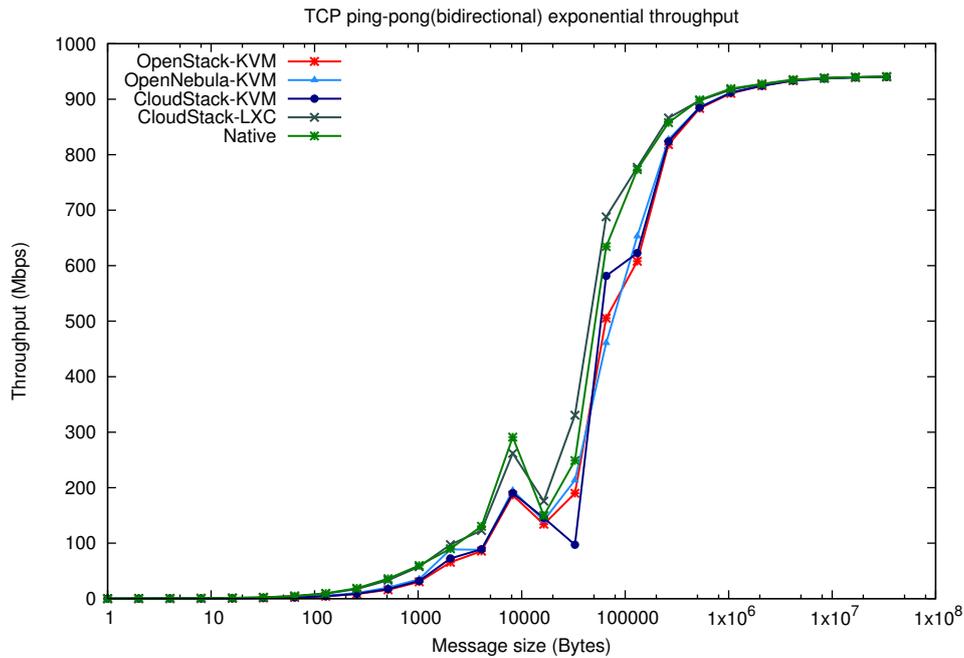


Figura 3.3: *Throughput* TCP Exponencial.

Na Figura 3.4 os resultados de *throughput* nos ambientes usando os protocolos TCP e UDP são mostrados. Comparando entre os ambientes, os resultados foram muito próximos, e o protocolo UDP foi um pouco melhor devido a não monitorar a transferência, focando no envio.

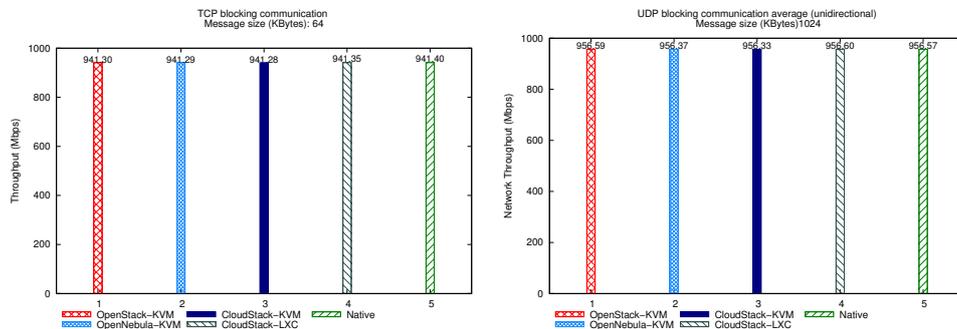


Figura 3.4: TCP e UDP *Throughput* Unidirecional.

A Figura 3.5 mostra os resultados customizando parâmetros dos pacotes. Os tamanhos de mensagem dos pacotes, quantidade de dados que pode ser transferida (MSS) e tamanho do *buffer* foram customizados para ver o desempenho dos ambientes quando as configurações de rede não são as usuais. Os resultados tiveram uma variação maior, o que diminui os contrastes (ganhos ou perdas) comparando os ambientes.

A Figura 3.6 apresenta os resultados de testes bidirecionais, quando os dois nodos enviam e recebem pacotes simultaneamente. Esse experimento apresentou resultados pobres usando o protocolo TCP devido ao controle de tráfego e verificação implementado pelo protocolo e também sofre influência do nível de prioridade do experimento nas filas do escalonador do sistema operacional. Usando o protocolo UDP, os resultados de desempenho foram significativamente melhores, a velocidade foi dobrada se comparada com a transferência unidirecional (Figura 3.4). A comparação entre as instâncias dos ambientes distintos mostra que a ordem se manteve, sendo o nativo o melhor, seguido pelo ambiente baseado em LXC e após, as VMs executadas usando o virtualizador KVM.

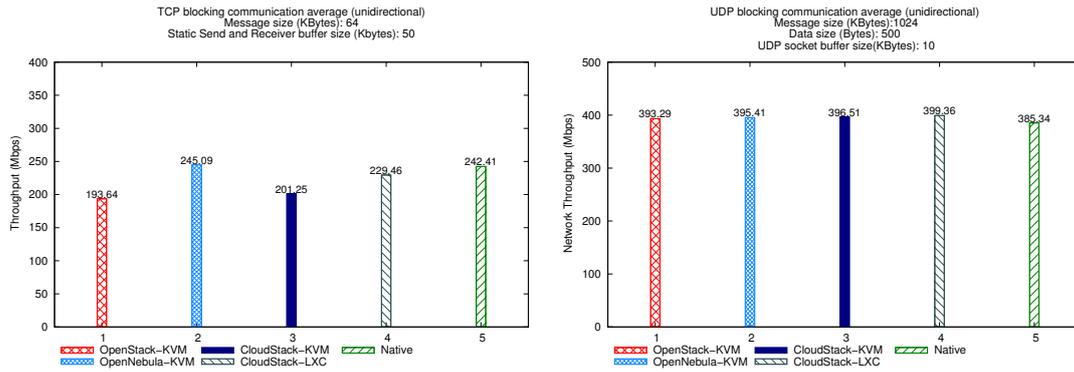


Figura 3.5: TCP e UDP *Throughput* Usando *Buffer* e tamanhos MSS customizados.

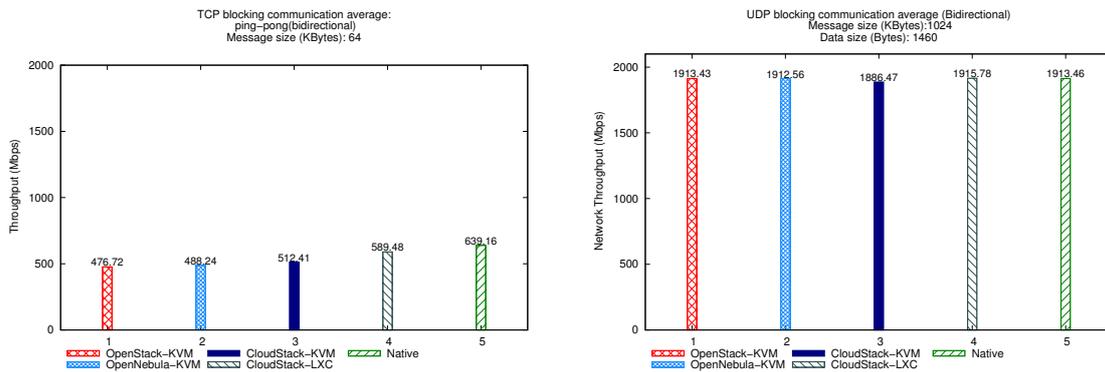


Figura 3.6: TCP e UDP Bidirecional *Throughput*.

Na Figura 3.7 são apresentados os resultados de latência TCP e UDP. Foi usado o RTT (*Round Trip Time*) para medir a latência, considerando o intervalo em micro-segundos entre uma instância iniciar uma conexão com outra, até receber o datagrama de resposta. Dessa forma, quanto menor o tempo, melhor é o ambiente. Contrastando com os resultados anteriores relacionados com o *throughput* de rede, na latência foram encontrados resultados significativamente diferentes entre os ambientes. As instâncias de nuvem usando as três ferramentas e o mesmo virtualizador KVM precisaram do dobro de tempo para completarem as operações, quando comparando com o ambiente nativo e o baseado em LXC. O ambiente nativo foi novamente o melhor, e o ambiente LXC usando as mesmas interfaces *bridges* de rede foi melhor que os ambientes que usaram o KVM.

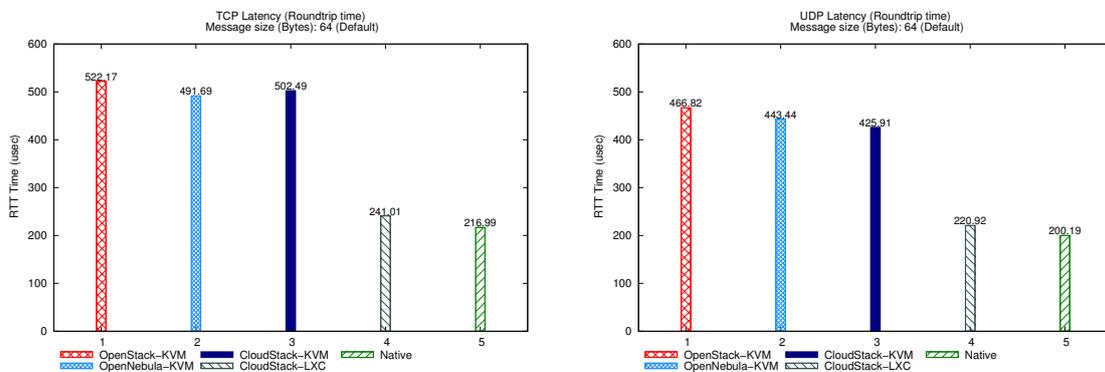


Figura 3.7: Latência.



## 4. Um Modelo de Ambiente de Produção na Nuvem

Os capítulos anteriores apresentaram análises de ferramentas de nuvem relacionados com robustez e desempenho. Esses resultados são relevantes para qualquer ambiente real de nuvem, pois a flexibilidade e resiliência das ferramentas impacta diretamente no ambiente, pois o suporte para integração com tecnologias e provisão de recursos para usuários define o quão eficiente é um ambiente de nuvem. O mesmo ocorre com o desempenho, o poder de processamento e I/O resulta na qualidade dos serviços oferecidos e executados na nuvem.

Além das implantações de nuvem que foram rodados os experimentos do capítulo anterior, existe uma enorme demanda por tolerância a falhas, redundância e backup e, conseqüentemente por alta disponibilidade. Ambientes de nuvem, pela flexibilidade e escalabilidade são recomendados para essas necessidades. Portanto, um ambiente de nuvem de produção foi implantado em direção de alta disponibilidade dos serviços na nuvem.

De acordo com os requisitos, o protótipo do ambiente foi planejado através da ferramenta de nuvem CloudStack devido a sua arquitetura flexível e voltada para alta disponibilidade. Pela confiabilidade e bom desempenho, o virtualizador KVM e formato de imagens QCOW foram escolhidos. Essas tecnologias raízes e demais secundárias foram selecionadas e escolhidas planejando uma implantação eficiente e tolerante à falhas. Outro aspecto muito importante para o ambiente de nuvem, por isso nesse foi escolhida GlusterFS integrada na nuvem e com volumes replicados para evitar perdas de dados. As próximas seções detalham como cada tecnologia e foi usada e integrada com o ambiente como um todo.

### 4.1 CloudStack

A ferramenta de IaaS CloudStack foi instalada seguindo a documentação oficial [7] usando o sistema operacional Ubuntu Server 14.04. Um servidor do ambiente foi configurado como gerente de nuvem, ou seja através do pacote *cloudstack-management*, esse nodo controla a utilização dos recursos, usuários e aloca a VMs nos nodos clientes agrupados em *clusters*. O agrupamento dos nodos numa nuvem CloudStack só é possível através da homogeneidade do ambiente, ou seja, todo o *hardware* e *softwares* precisam ser os mesmos. Um dos benefícios da utilização de clusters é a possibilidade de alta disponibilidade e balanceamento de carga.

Outro requisito para alta disponibilidade de VMs em uma nuvem CloudStack é armazenamento distribuído, ou seja, os volumes e ISOs usadas pelas máquinas virtuais precisam estarem acessíveis para todos os nodos do cluster. Com a infraestrutura instalada apropriadamente, as instâncias podem ser criadas com a opção "HA"habilitada.

### 4.2 GlusterFS

A tecnologia de armazenamento GlusterFS foi integrada com a nuvem CloudStack para oferecer volumes virtuais as instâncias virtuais. O lado do servidor do GlusterFS foi instalado em duas máquinas rodando o sistema operacional Ubuntu Server 14.04, dessa forma, ambas tiveram os pacotes do servidor instalados e posteriormente um volume replicado foi criado e sincronizado entre os servidores. O lado cliente é a ferramenta CCloudStack onde esse volume foi adicionado como zona primária de armazenamento (armazenamento dos discos das máquinas virtuais), e as informações ficam duplicadas em cada servidor para assim evitar perda de dados.

## 4.3 Em Direção à Alta Disponibilidade

Nas seções anteriores as principais tecnologias usadas na implantação do modelo de ambiente de produção foram apresentadas e foi detalhado como foram configuradas. Alta disponibilidade é algo complexo que depende de vários níveis de serviços e paridades. Iniciando no nível do *hardware*, falha de algum componentes pode resultar e indisponibilidade, a energia que é usada para manter os equipamentos ligados não pode falhar também. Nos nível está os sistemas operacionais e virtualizadores, que oferecem os recursos para os serviços da nuvem.

No caso de alguma falha nas camadas inferiores, as ferramentas para gerenciamento de infraestrutura de nuvem estão avançando para controlarem a disponibilidade, isso geralmente ocorre através da redundância dos servidores.

### 4.3.1 Alta Disponibilidade baseada em alocação de VMs em *Clusters* Homogêneos

A ferramenta CloudStack oferece uma forma de alta disponibilidade para as máquinas virtuais, através da alocações em servidores agrupados em *cluster*. Cada VM é alocado em um determinado servidor, e a ferramenta monitora a disponibilidade do servidor e em caso de alguma falha seja detectada a máquina é automaticamente transferida para outro servidor do mesmo *cluster*. Para isso, geralmente a VM precisa ser reiniciada, mas em questão de minutos os serviços executados retornam. É possível ainda oferecer redundância e sincronização entre as máquinas virtuais, para no caso de falha de alguma, outra continua a execução de determinada aplicação, exemplos de tecnologias são Heartbeat, Corosync, Keepalived e Pacemaker.

## 4.4 Em direção a backup em nuvem privada

Além da necessidade pela disponibilidade e confiabilidade em ambientes de produção, existe uma grande necessidade por integridade de dados virtuais. Uma técnica muito usada para evitar perdas de dados é realizar backup dos dados, sempre mantendo pelo menos uma cópia. Diversos provedores já oferecem o recurso de backup em nuvem, que pode ser contratado e pago de acordo com a utilização. Existe também a alternativa da instalação de instalar servidores próprios para backup, onde uma determinada tecnologia é instalada para oferecer esse serviço exclusivamente para um domínio.

Existem diversas soluções para backup, e nesse estudo consideramos as alternativas gratuitas e de código aberto. Nesse segmento, a solução mais popular é o bacula. Analisando essa tecnologia ficou evidente que é uma solução robusta e que atende várias demandas. Conseqüentemente, essa solução é mais complexa para implantar e gerenciar, além do fato que necessita para o correto funcionamento uma infraestrutura robusta e complexa. Por isso, foi testada uma BackupPC que é mais simplista e facilmente atende aos objetivos de um ambiente de nuvem privada (Backup de máquinas virtuais). Essa solução foi instalada so sistema operacional *Ubuntu Server* e foi habilitada também a interface gráfica para gerenciamento via *Web*, que possibilita a configuração dos clientes de backup, os agendamentos das tarefas de backup e restauração.

## 5. Conclusão

Os benefícios para a migração dos serviços computacionais para serviços de nuvem são evidentes. Os principais desafios são relacionados com desempenho, gerenciamento e segurança nesses ambientes. Enquanto os provedores de nuvem pública oferecem recursos computacionais sob demanda e pagos conforme a utilização, ferramentas de gerenciamento de infraestrutura surgiram oferecendo a alternativa de se implantar nuvens privadas, com níveis adicionais de controle, privacidade e customização. Essas ferramentas possuem profundos contrastes tanto na arquitetura, implantação, tecnologias e serviços suportados.

Por isso, as ferramentas de código aberto disponíveis para a implantação de ambiente de nuvem foram analisadas e comparadas quanto ao suporte para flexibilidade e resiliência em implantações. Os resultados mostraram a ferramenta CloudStack como a mais flexível e OpenStack mais resiliente. A combinação de flexibilidade e resiliência resulta em robustez, o que reflete no nível de compatibilidade e customização possível para um ambiente de nuvem, isso aliado ao fato de cada aplicação ter características e requisitos diferentes faz com que a robustez seja extremamente importante para QoS.

Além da avaliação das ferramentas, o desempenho das implantações usando o virtualizador KVM mostrou resultados próximos no desempenho da infraestrutura e de aplicações científicas. Em testes específicos de rede ficou evidente que o *throughput* entre ambientes de nuvem KVM, LXC e nativo são próximos. Por outro lado, a latência nas implantações de nuvem usando KVM foram altas em comparação com os ambientes nativos e LXC.

Uma análise e implantação de ambiente de nuvem em direção a alta disponibilidade para serviços na nuvem foi apresentada usando a ferramenta CloudStack. Nessa implantação, os nós são agrupados em *clusters* que formam zonas de disponibilidade, no caso de falha em algum dos nós, as máquinas virtuais são transferidas e reiniciadas em outro servidor. Esse nível de disponibilidade e tolerância a falhas atingido no nível de infraestrutura se combinado com redundância no nível das aplicações pode atingir facilmente elevados níveis de confiabilidade.



## Referências Bibliográficas

- [1] 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.: The nas parallel benchmarks. *International Journal of High Performance Computing Applications* 5(3), 63–73 (1991)
- [2] Baun, C., Kunze, M., Nimis, J., Tai, S.: *Cloud Computing: Web-Based Dynamic IT Services*. Springer Berlin Heidelberg (2011), <https://books.google.com.br/books?id=b1skkacCvoC>
- [3] Buyya, R., Vecchiola, C., Selvi, S.: *Mastering Cloud Computing*. McGraw Hill (2013), <https://books.google.com.br/books?id=VSDZAgAAQBAJ>
- [4] Castano, V., Schagaev, I.: *Resilient Computer System Design*. Springer International Publishing (2015), [https://books.google.com.br/books?id=4\\_RICAAAQBAJ](https://books.google.com.br/books?id=4_RICAAAQBAJ)
- [5] Chandrasekaran, K.: *Essentials of Cloud Computing*. Taylor & Francis (2014), <https://books.google.com.br/books?id=-GhYBQAAQBAJ>
- [6] Cloudstack: *Cloudstack roadmap* <<https://cloudstack.apache.org/>> (2015), last access Dec, 2015
- [7] CloudStack, A.: *CloudStack Installation*, <<https://docs.cloudstack.apache.org/projects/cloudstack-installation/en/4.6/>> (2016), last access Feb, 2016
- [8] Coyne, L., Gopalakrishnan, S., Sing, J., Redbooks, I.: *IBM Private, Public, and Hybrid Cloud Storage Solutions*. IBM Redbooks (2014), <https://books.google.com.br/books?id=Mjr5AwwAAQBAJ>
- [9] Denton, J.: *Learning OpenStack Networking (Neutron)*. Community experience distilled, Packt Publishing (2014), <https://books.google.com.br/books?id=iXrKBAAAQBAJ>
- [10] Dongarra, J., Luszczek, P., Petitet, A.: *The Linpack Benchmark: Past, Present and future*. eeeee (2001)
- [11] Dukaric, R., Juric, M.B.: *Towards a unified taxonomy and architecture of cloud frameworks*. *Future Generation Computer Systems* 29(5), 1196–1210 (2013)
- [12] Eucalyptus: *Eucalyptus: Open Source Private Cloud Software*, <<https://www.eucalyptus.com/eucalyptus-cloud/iaas>> (2015), last access Mar, 2015
- [13] Fifield, T., Fleming, D., Gentle, A., Hochstein, L., Proulx, J., Toews, E., Topjian, J.: *OpenStack Operations Guide*. O’Reilly Media (2014), <https://books.google.com.br/books?id=jQ5pAwAAQBAJ>
- [14] Hentges, E.L., Thomé, B.R.: *Análise e Comparação de Ferramentas Open Source de Computação em Nuvem para o Modelo de Serviço IaaS*. Master’s thesis, Undergraduate Thesis, Sociedade Educacional Três de Maio (SETREM), Três de Maio, RS, Brazil (December 2013)
- [15] Huang, B., Bauer, M., Katchabaw, M.: *Hpcbench - a Linux-based network benchmark for high performance networks*. In: *High Performance Computing Systems and Applications, 2005. HPCS 2005. 19th International Symposium on*. pp. 65–71 (May 2005)

- [16] Iozone: Iozone File System Benchmark (Official Page) <<http://iozone.org/>> (2006), last access in April, 2015
- [17] Iperf: What is Iperf? (2015), <http://iperf.fr/>, last access in Sept, 2015
- [18] ISO JTC1/SC38 Technical Report: ISO/IEC 17788:2014 Information Technology - Cloud Computing - Overview and Vocabulary (2014), [http://www.iso.org/iso/catalogue\\_detail?csnumber=60544](http://www.iso.org/iso/catalogue_detail?csnumber=60544), last access in Sept, 2015
- [19] Kumar, R., Gupta, N., Charu, S., Jain, K., Jangir, S.K.: Open Source Solution for Cloud Computing Platform Using OpenStack. *International Journal of Computer Science and Mobile Computing* 3(5), 89–98 (2014)
- [20] Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., Leaf, D.: NIST cloud computing reference architecture. NIST special publication 500, 292 (2011)
- [21] Mahmood, Z.: Continued Rise of the Cloud: Advances and Trends in Cloud Computing. *Computer Communications and Networks*, Springer London (2014), <https://books.google.com.br/books?id=114gBAAAQBAJ>
- [22] Marinescu, D.: Cloud Computing: Theory and Practice. Elsevier Science (2013), <https://books.google.com.br/books?id=mpcBw10nyIgC>
- [23] Maron, C.A.F., Griebler, D.: Avaliando o Desempenho das Ferramentas de Nuvem Privada OpenStack e OpenNebula. Tech. rep., Laboratory of Advanced Researches on Cloud Computing (LARCC) (2015)
- [24] Mccalpin, J.: The Stream Benchmark) <<https://www.cs.virginia.edu/stream/>> (1996), last access in April, 2015
- [25] Nimbus: About Nimbus, <<http://www.nimbusproject.org/about>> (2015), last access Mar, 2015
- [26] OpenNebula: OpenNebula roadmap <<http://opennebula.org//>> (2015), last access Dec, 2015
- [27] OpenQRM: OpenQRM Community Edition, <<http://www.openqrm-enterprise.com/products/community-edition.html>> (2015), last access Mar, 2015
- [28] OpenStack: OpenStack roadmap <<http://openstack.org/software/roadmap/>> (2015), last access Dec, 2015
- [29] Project, O.: Quickstart: OpenNebula 4.12 on Ubuntu 14.04 and KVM *http://docs.opennebula.org/4.12/design\_and\_installation/quick\_start/qs\_ubuntu\_vm* (2015), last access Dec, 2015
- [30] Roveda, D., Vogel, A., Griebler, D.: Understanding, Discussing and Analyzing the OpenNebula's and OpenStack's IaaS Management Layers. *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação (REABTIC)* 3(1), 15 (August 2015)
- [31] Roveda, D., Vogel, A., Maron, C.A.F., Griebler, D., Schepke, C.: Analisando a Camada de Gerenciamento das Ferramentas CloudStack e OpenStack para Nuvens Privadas. In: 13th Escola Regional de Redes de Computadores (ERRC). Sociedade Brasileira de Computação, Passo Fundo, Brazil (September 2015)

- [32] Sabharwal, N.: Apache CloudStack Cloud Computing. Community experience distilled, Packt Publishing (2013), [https://books.google.com.br/books?id=46m\\_ciDiCe8C](https://books.google.com.br/books?id=46m_ciDiCe8C)
- [33] Sosinsky, B.: Cloud Computing Bible. Wiley (2010), <https://books.google.com.br/books?id=hvv2pDEAbOEC>
- [34] Srivastava, S., Ciorba, F.M., Banicescu, I.: Employing a study of the robustness metrics to assess the reliability of dynamic loop scheduling (2011)
- [35] Thome, B., Hentges, E., Griebler, D.: Computação em Nuvem: Análise Comparativa de Ferramentas Open Source para IaaS. In: 11th Escola Regional de Redes de Computadores (ERRC). p. 4. Sociedade Brasileira de Computação, Porto Alegre, RS, Brazil (November 2013)
- [36] Toraldo, G.: OpenNebula 3 Cloud Computing. Community experience distilled, Packt Publishing (2012), <https://books.google.com.br/books?id=W0o0o8dIp9UC>
- [37] Vaquero, L.: Open Source Cloud Computing Systems: Practices and Paradigms: Practices and Paradigms. Premier reference source, Information Science Reference (2012), <https://books.google.com.br/books?id=my5V2JUW0U0C>
- [38] Vogel, A.: Surveying the Robustness and Analyzing the Performance Impact of Open Source Infrastructure as a Service Management Tools. Master's thesis, Undergraduate Thesis, Sociedade Educacional Três de Maio (SETREM), Três de Maio, RS, Brazil (August 2015)
- [39] Vogel, A., Griebler, D., Maron, C.A.F., Schepke, C., Fernandes, L.G.: Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack. In: 24th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). pp. 672–679. IEEE, Heraklion Crete, Greece (February 2016)

