



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Campus de São José do Rio Preto

Leandro Luis Pauro

Auditoria e Monitoramento de Eventos Inconsistentes em
Instâncias de Máquinas Virtuais em *IaaS* no Orquestrador
Apache CloudStack

SÃO JOSÉ DO RIO PRETO

2016

Leandro Luis Pauro

Auditoria e Monitoramento de Eventos Inconsistentes em
Instâncias de Máquinas Virtuais em *IaaS* no Orquestrador
Apache CloudStack

Orientador: Prof^a. Dr^a. Roberta Spolon

Dissertação de Mestrado elaborada junto ao Programa de Pós-Graduação em Ciência da Computação - Área de Concentração em Arquitetura de Computadores e Sistemas Distribuídos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

SÃO JOSÉ DO RIO PRETO

2016

Pauro, Leandro Luis.

Auditoria e monitoramento de eventos inconsistentes em instâncias de máquinas virtuais em IaaS no orquestrador Apache CloudStack / Leandro Luis Pauro. – São José do Rio Preto, 2016
102 f. : il.

Orientador: Roberta Spolon

Dissertação (mestrado) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Instituto de Biociências, Letras e Ciências Exatas

1. Computação. 2. Sistemas de computação virtual.
3. Computação em nuvem. 4. Redes de computadores - Monitorização. 5. Software gratuito. I. Universidade Estadual Paulista "Júlio de Mesquita Filho". Instituto de Biociências, Letras e Ciências Exatas. II. Título.

CDU – 681.3.025

Ficha catalográfica elaborada pela Biblioteca do IBILCE
UNESP - Câmpus de São José do Rio Preto

Leandro Luis Pauro

Auditoria e Monitoramento de Eventos Inconsistentes em
Instâncias de Máquinas Virtuais em *IaaS* no Orquestrador
Apache CloudStack

Dissertação de Mestrado elaborada junto ao Programa de Pós-Graduação em Ciência da Computação – Área de Concentração em Arquitetura de Computadores e Sistemas Distribuídos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de São José do Rio Preto.

Orientador: Prof^a. Dr^a. Roberta Spolon

Comissão Examinadora

Prof^a. Dr^a. Roberta Spolon
Universidade Estadual Paulista - Bauru
Orientador

Prof. Dr. Marcos A. Cavenaghi
HumberInstitute of Technology and AdvancedLearning, The Business School - Toronto
- Canadá

Prof. Dr. Ronaldo Augusto de Lara Gonçalves
Universidade Estadual de Maringá

Bauru, 06 de dezembro de 2016

Agradecimentos

Primeiramente a Deus que permitiu que tudo isso acontecesse, ao longo de minha vida.

Em segundo lugar, agradeço aos meus pais Evaristo Pauro e Ester Maria da Cruz Pauro, pela formação que me proporcionaram e sem eles não estaria aqui. À minha noiva Micheli Fabiana Salina, por toda paciência e compreensão.

Em especial, a minha professora orientadora Dr^a. Roberta Spolon, por ter acreditado, pelo auxílio, disponibilidade de tempo, material e dedicação a me conduzir nesta pesquisa.

Meu amigo e colega de mestrado Gustavo Cesar Bruschi que muito me ajuda nas dificuldades de conhecimento técnico, para o meu maior incentivador e amigo Henrique Pachioni Martins, pessoa que fez muito por mim não apenas em relação a minha carreira profissional, mas também em minha vida como pessoa.

Aos colegas de trabalho, em especial aos amigos Alessandro Martinez de Camargo e Miguel Neves, por entenderem e em dar o suporte necessário para dar andamento na realização desse trabalho.

Resumo

Cada vez mais a Computação em Nuvem é incorporada pelas empresas como forma econômica e viável de se disponibilizar recursos e serviços. No entanto, a confiabilidade operacional e a disponibilidade de recurso ainda causam preocupação em virtude de ocorrer a inatividade de algum serviço fornecido pela nuvem, o que pode gerar a perda de receitas e desconfiança do cliente. Assim, é crucial que se disponibilize ferramentas a este ambiente para realizar auditoria e monitoramento, a fim de prover a prevenção e a eliminação de inconsistências que possam provocar a indisponibilidade do serviço oferecido. Este trabalho apresenta a ferramenta de Auditoria e Monitoramento em Nuvem Orquestrador *Apache CloudStack* AMFC, que através do sincronismo das informações do estado atual com dados persistentes do orquestrador, realiza a eliminação de dados sem utilização e inconsistências, diminui o alertas de falso positivo e falso negativo e também proporciona menor custo para armazenamento de dados persistentes da nuvem. Sua eficácia foi evidenciada através da realização de validação manual comparada com o resultado obtido da execução da ferramenta a partir de casos de uso gerados no ambiente de teste controlado. Os resultados obtidos após a realização de 1.320 rotinas administrativas para instância de máquina virtual mostraram a identificação e eliminação das inconsistências na base de dados persistente, a redução do custo de armazenamento e conseqüentemente, uma base de dados íntegra, que oferece ao administrador da nuvem uma tomada de decisão com maior precisão para averiguar um problema que esteja ocorrendo no ambiente.

Palavras-chave: Computação em Nuvem, *opensource*, *CloudStack*, auditoria, monitoramento, inconsistência, instância, infraestrutura.

Abstract

Cloud Computing has been increasingly incorporated by companies as an economic and feasible mean to provide resources and services. However, operational reliability and resource availability are still cause for concern since there's the possibility of a cloud service going down, which can lead to loss of revenue and customer distrust. Thus, it is crucial to provide tools for performing auditing and monitoring in order to prevent and eliminate inconsistencies that may cause the unavailability of the service offered. This paper presents the Cloud Orchestrator Auditing and Monitoring Tool Apache CloudStack AMFC, which by synching information current status with the orchestrator persistent data deletes any unused data and inconsistencies, decreases the false positive and negative alerts and also provides lower cost for cloud persistent data storage. Its effectiveness has been demonstrated through manual validation compared to results obtained from running the tool in a controlled test environment. The results obtained after performing 1.320 administrative tasks for a virtual machine were the identification and elimination of inconsistencies in the persistent database, reducing storage costs and, consequently, resulting in an intact database. This enables the cloud administrator to make more accurate decisions when investigating a possible malfunction in the environment.

Keywords: *Cloud computing, open source, CloudStack, auditing, monitoring, inconsistency, instance, and infrastructure.*

Lista de Figuras

Figura 1 - Ambiente Computacional com Base na Computação em Nuvem	13
Figura 2 - Virtualização de Servidores	14
Figura 3 - Características de Computação em Nuvem	15
Figura 4 - Arquitetura da Computação em Nuvem.....	18
Figura 5 - Arquitetura do Orquestrador Eucalyptus	25
Figura 6 - Arquitetura do Orquestrador <i>OpenNebula</i>	26
Figura 7 - Plataforma Windows Azure	28
Figura 8 - Arquitetura do Orquestrador <i>OpenStack</i>	31
Figura 9 - Arquitetura do Orquestrador <i>Vmware VCloud</i>	34
Figura 10 - Arquitetura do <i>Apache CloudStack</i>	35
Figura 11 - Principais Características dos Orquestradores de Computação em Nuvem	39
Figura 12 - Diagrama da Metodologia para Auditoria e Monitoramento.....	52
Figura 13 - Utilização da ferramenta AMFC na nuvem	57
Figura 14 - Camada de Nível de Serviço de configuração da ferramenta AMFC.....	59
Figura 15 - Etapas do Monitoramento da Nuvem pela Ferramenta AMFC	60
Figura 16 - Fluxograma Módulo de Redundância não Controlada	62
Figura 17 - Gerenciamento de Instância de Máquina Virtual no Orquestrador.....	66
Figura 18 - Informações de Instância de Máquina Virtual em Dados Persistentes	67
Figura 19 - Atividades de Eventos no Orquestrador	67
Figura 20 - Dados Persistentes das Atividades de Eventos	67
Figura 21 - Gerenciamento de Instâncias de Máquinas Virtuais no Orquestrador.....	68
Figura 22 - Status de Instância de Máquina Virtual de Dados Persistentes.....	69
Figura 23 - Atividades de Eventos no Orquestrador	69
Figura 24 - Atividades de Eventos do Orquestrador em Dados Persistentes	69

Figura 25 - Ferramenta AMFC: Informações de Máquina Virtual e Atividades dos Eventos	72
Figura 26 - Modelo de Redundância de Dados no Orquestrador <i>Apache CloudStack</i> ...	73
Figura 27 - Gerenciamento de Armazenamento de Volumes	74
Figura 28 - Informação dos Volumes em Dados Persistentes.....	74
Figura 29 - Gerenciamento de Instâncias de Máquinas Virtuais	75
Figura 30 - Informações Inconsistentes de Volumes em Dados Persistentes	76
Figura 31 - Inconsistências de informações de volumes ferramenta AMFC.....	77
Figura 32 - Resultado do Modelo de Recurso Pequeno	78
Figura 33 - Resultado do Modelo de Recurso Médio.....	79
Figura 34 - Resultado do Modelo de Recurso Grande	79
Figura 35 - Comparativo de Inconsistências entre Recursos.....	80
Figura 36 - Layout de instalação do <i>CloudStack</i>	92
Figura 37 - SGBD <i>MySql</i> do arquestrador <i>CloudStack</i>	97
Figura 38 - <i>Hypervisor</i> <i>XenServer</i> instalado no orquestrador <i>Apache CloudStack</i> ...	100
Figura 39 - <i>Interface web</i> de gerenciamento do orquestrador <i>Apache CloudStack</i>	102

Lista de Quadros

Quadro 1- Definição das Questões de Segurança em Nuvens.....	21
Quadro 2 - Configuração dos computadores	50
Quadro 3 - Pseudocódigo da ferramenta AMFC	61
Quadro 4 - Tipos de Oferta de recurso	64
Quadro 5 - Modelo de testes oferta de recurso	65
Quadro 6 - Eliminação Manual para Validar as Informações sem Utilização no Ambiente de Teste	71
Quadro 7 - Eliminação Manual para Validar Informações de Volumes	77
Quadro 8 - Ganho de Armazenamento de Dados Persistentes	80
Quadro 9 - Equipamentos utilizados para o ambiente de protótipo da computação em nuvem	91

Lista de Abreviaturas

AMFC - Ferramenta de Auditoria e Monitoramento para Eliminar Inconsistências no Apache CloudStack

API - *Application Performance Interface*

AWS - *Amazon Web Service*

CLI - *Command Line Interface*

CSP - *Cloud Service Providers*

DDL - *Data Definition Language*

DHCP - *Dynamic Host Configuration Protocol*

DML - *Data Manipulation Language*

EBS - *Elastic Block Store*

EC2 - *Elastic Compute Cloud*

GNU - *General Public License*

GPL - *GNU Public Licence*

IaaS - *Infrastructure as a Service*

IEEE - *Institute of Electrical and Electronics Engineers*

IP - *Internet Protocol*

ISCI - *Internet Small Computer System Interface*

LDAP - *Lightweight Directory Access Protocol*

NASA - *United States National Aeronautics and Space Administration*

NAT - *Network Address Translation*

NFS - *Network File System*

NIST - *National Institute of Standards and Technology*

NoSQL - *Not Only SQL*

PaaS - *Platform as a Service*

S3 - *Simple Storage Service*

SaaS - *Software as a Service*

SAML - *Security Assertion Markup Language*

SDN - *Software-Defined Networking*

SGBD - Sistema de Gerenciamento de Banco de Dados

SLA - *Service Level Agreement*

TI - Tecnologia da informação

VLAN - *Virtual LAN*

VM - *Virtual Machine*

XACML - *Extensible Access Control Markup Language*

XCP - *Xen Cloud Platform*

Sumário

1	Introdução.....	8
1.1	Objetivo	9
1.2	Motivação	10
1.3	Organização do texto	10
2	Computação em Nuvem	12
2.1	Introdução à computação em nuvem	12
2.2	Arquitetura da computação em nuvem	16
2.3	Modelos de níveis de serviços em computação em nuvem	17
2.4	Segurança na computação em nuvem.....	20
2.5	Vantagens e riscos da computação em nuvem	22
2.6	Orquestradores de computação em nuvem	24
2.6.1	<i>Eucalyptus</i>	24
2.6.2	<i>OpenNebula</i>	26
2.6.3	<i>Xen Cloud Platform</i>	27
2.6.4	<i>Microsoft Windows Azure Platform</i>	28
2.6.5	<i>OpenStack</i>	30
2.6.6	<i>Amazon EC2 (Elastic Compute Cloud)</i>	32
2.6.7	<i>Vmware vCloud Suite</i>	33
2.6.8	<i>Apache CloudStack</i>	35
2.7	Considerações Finais	38
3	Auditoria e Monitoramento em Computação em Nuvem.....	40
3.1	Introdução à Auditoria e Monitoramento em Nuvem	40
3.2	Técnicas e Método para Auditoria e Monitoramento em Nuvem.....	40
3.3	Principais Características para uma Ferramenta de Auditoria e Monitoramento .	41

3.4 Normas para Auditoria e Monitoramento em Nuvem.....	43
3.4.1 ISO/IEC 17788	44
3.4.2 ISO/IEC 17789 sessão 8.5.12.1.....	44
3.4.3 NIST SP 500-292	45
3.5 Trabalhos Relacionados	46
3.6 Considerações Finais	48
4 Material e Métodos	49
4.1 Material	49
4.2 Hardware.....	49
4.3 Software	50
4.4 Métodos	50
Definição dos Parâmetros de Cálculo	54
4.5 Considerações Finais	55
5 Ferramenta de auditoria e monitoramento em nuvem AMFC.....	57
5.1 Arquitetura da ferramenta AMFC	57
5.2 Implantação na nuvem da ferramenta AMFC.....	59
5.2 Ambiente da nuvem auditado e monitorado pela ferramenta AMFC	60
5.3 Considerações Finais	63
6 Resultados Experimentais.....	64
6.1 Realização dos Testes e Resultados	64
6.2 Casos de uso realizados de auditoria e monitoramento AMFC	65
6.3 Cenário 1: Ambiente de teste e dados persistentes no gerenciamento de informações.....	66
6.4 Cenário 2: Monitorar Informação Inconsistente no Ambiente de Nuvem.....	74
5.5 Avaliação dos Resultados.....	78
6.6 Considerações Finais	81

7 Conclusões.....	82
7.1 Contribuições	83
7.2 Trabalhos Futuros	84
Referências Bibliográficas.....	85
Apêndice A - Instalação do Apache CloudStack	91
Apêndice B - Instalação do <i>XenServer</i>	98
Apêndice C - Instalação do <i>OpenFiler</i>	101
Apêndice D - Instalação do <i>XenCenter</i>	102

1 Introdução

Com o crescente uso computacional, aumenta-se a disponibilização de recursos sob demanda (como sistemas, aplicativos, armazenamento de dados) que podem ser acessados em qualquer lugar independente de plataforma, com alta disponibilidade e escalabilidade, tendo a mesma facilidade de tê-los instalados como em nossos próprios computadores. (HE; HE, 2011)

Sabahi (2011) define computação em nuvem como um ambiente de rede baseado no compartilhamento de recursos computacionais. Na verdade, nuvens são baseadas na *internet* e tentam disfarçar a complexidade para os usuários. Computação em Nuvem refere-se à entrega de serviços através da *internet* por meio de *hardware* e *software* nos ¹*datacenters*. Assim, empresas que fornecem nuvens se utilizam de tecnologias de virtualização, combinadas com suas habilidades para fornecer recursos de computação através de sua infraestrutura de rede.

A computação em nuvem é vista como uma provisão de recursos computacionais, de responsabilidade de empresas especializadas que realizam abstração de recursos em níveis que apenas especialistas venham se preocupar em gerenciá-los e mantê-los, sendo disponibilizados como serviços. (CARR, 2008)

É, portanto, um modelo que possibilita o acesso sob demanda a um *pool* compartilhado de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) de forma simples e eficiente na sua maximização e flexibilização destes recursos. Além disso, uma nuvem computacional é um ambiente redundante, resiliente e de escalabilidade, que mantém sua operabilidade apesar de um mau funcionamento de um ou mais dos seus componentes. (TAURION, 2009)

Convém observar que a computação em nuvem consiste em serviços que são comoditizados² e entregues de modo semelhante, tais como água, eletricidade, gás e telefonia. Nesse modelo, os usuários acessam os serviços baseados em suas necessidades, independentemente do local onde estes serviços estejam hospedados. (BUYYA et al., 2013)

¹*Datacenter* - Centro de processamento de dados, é um ambiente projetado para concentrar servidores, equipamentos de processamento e armazenamento de dados, e sistemas de ativos de rede, como *switches*, roteadores, e outros.

²Comoditizado - Mercadoria ou serviço que é fornecido por muitos agentes econômicos e, portanto tem seu preço pressionado para baixo pela extrema competição.

A computação em nuvem tornou-se economicamente viável para grandes empresas. No entanto, a confiabilidade operacional e disponibilidade de recursos ainda causam preocupação, em virtude de ocorrer a inatividade de algum serviço fornecido pela nuvem, que possam gerar a perda de receitas e a insegurança do cliente. Assim, é crucial que se disponibilize ferramentas a este ambiente, capazes de realizar a auditoria e o monitoramento, a fim de oferecer a prevenção e a eliminação de inconsistências que podem provocar a indisponibilidade do serviço oferecido. (ZHU et al., 2013)

Para se interagir com esta infraestrutura de nuvens, os serviços de monitoramento são essenciais, devido a versatilidade de aplicações e recursos que são simultaneamente acessados por vários usuários. Nuvens muitas vezes sofrem vários tipos de situações indesejadas, tais como degradações de desempenho, falhas de componentes ou invasões de segurança. Com isso é importante haver mecanismos de monitoramento e auditoria para nuvens, sendo possível avaliar os eventos apontados e posteriormente oferecer ações de prevenção e recuperação. (MDHAFFAR et al., 2013)

A utilização de um serviço de monitoramento de recursos é considerado um dos principais desafios para uma auditoria em nuvem, devido à falta de informações e o controle sobre as customizações dos parâmetros que descrevem os sistemas deste ambiente. É importante observar que as soluções de monitoramento atuais não são totalmente precisas, pois a coleta de informação que se realiza neste ambiente ocorre em áreas distintas e não integradas. (SUCIU et al., 2014)

Para que se consiga prover um ambiente confiável e permitir uma auditoria em tempo real, os serviços de monitoramento devem fornecer os resultados a partir da avaliação corrente do desempenho e o estado atual da nuvem. Para garantir esta confiabilidade, deve-se haver uma combinação de mecanismos que possam prover a prevenção, detecção e correção para este ambiente. (MEERA e GEETHAKUMARI, 2015)

1.1 Objetivo

Desenvolver uma ferramenta que realize a auditoria e monitoramento das rotinas administrativas em instâncias de máquinas virtuais para o ambiente de nuvem. Isso é obtido com o sincronismo das informações armazenadas na base de dados do orquestrador e as informações de seu estado atual, proporcionando ao administrador da nuvem suporte para tomada de decisão com maior precisão, realizando a eliminação de dados sem utilização e

eventos inconsistentes, diminuindo alertas de falso positivo e falso negativo e também oferecendo menor custos para armazenamento de dados persistentes da nuvem.

1.2 Motivação

A computação em nuvem se tornou uma alternativa encontrada pelas empresas na busca de redução de custos relacionados à tecnologia, através da diminuição de operações de T.I, normalmente tendo como ação inicial a virtualização de seus servidores. O gerenciamento de todos os elementos de software da computação em nuvem dimensionados sob demanda por meio da internet possibilita que as empresas tenham maior flexibilidade em relação a seus dados e suas informações.

Este ambiente deve ser auditado e monitorado no intuito de continuar oferecendo todos os benefícios que o modelo de computação em nuvem proporciona. Um ponto relevante encontrado é a forma como é monitorado, por meio de ferramentas que coletam apenas informações através de logs e em lugares separados na nuvem, podendo ocasionar informações inconsistentes e a geração de alertas de falsos positivos e falsos negativos. Isso induz o administrador do ambiente a cometer falhas no gerenciamento e administração dos recursos, podendo provocar até uma indisponibilidade de serviço.

É necessário que este monitoramento de informações na nuvem ocorra de forma sincronizada entre os dados persistentes do orquestrador e informações de estado atual do ambiente de nuvem com o intuito de proporcionar eliminação das informações inconsistentes, prevenção de alertas de falsos positivos e falsos negativos, maior precisão do administrador de infraestrutura sobre qual ação deve-se tomar e também menor custo para armazenamento de dados persistentes da nuvem.

1.3 Organização do texto

O texto está organizado em seis capítulos descritos a seguir:

- O Capítulo 2 apresenta uma revisão bibliográfica sobre Computação em Nuvem e também uma descrição sucinta sobre os principais orquestradores;
- No Capítulo 3 são apresentadas características, normas e trabalhos relacionados com auditoria e monitoramento em ambiente de computação em nuvem;

- O Capítulo 4 descreve a metodologia seguida nesta dissertação, os materiais e métodos utilizados;
- No Capítulo 5 são apresentados e discutidos os resultados dos experimentos realizados, e
- No capítulo 6 são apresentadas as conclusões embasadas pelos resultados obtidos, enumeradas as contribuições desta dissertação e as propostas para trabalhos futuros.

2 Computação em Nuvem

Esse capítulo apresenta detalhes da arquitetura de nuvem, definindo seus modelos, tipos de serviços, segurança, vantagens e riscos e também analisando os principais orquestradores e suas diferenças.

2.1 Introdução à computação em nuvem

A computação em nuvem é considerada um modelo de acesso a um *pool* compartilhado de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados e liberados com o mínimo de esforço no gerenciamento ou integração dos serviços pelo provedor, permitindo assim que os usuários acessem por meio da *internet* estes recursos de computação de acordo com a demanda, conforme necessário. (QAISAR et al., 2012)

Uma nuvem é um tipo de sistema paralelo e distribuído que consiste de uma coleção de computadores virtualizados e interconectados, provisionados de forma dinâmica e apresentados como um ou mais recursos computacionais unificados. Estes recursos são disponibilizados e controlados através de acordos relacionados aos serviços, estabelecidos entre um prestador e um consumidor, sendo definidos a partir de negociações entre as partes. Os recursos do computador são heterogêneos e geograficamente dispersos. (BUYAYA, 2008) e (NIST, 2014)

O conceito de computação em nuvem é oferecer o armazenamento e processamento dos dados fora do ambiente corporativo, dentro da grande rede e em estruturas conhecidas como centro de dados, otimizando o uso dos recursos. Os centros de dados irão fazer o papel de processar aplicações e armazenar os dados da organização que atuam em rede. (VERAS, 2012) Desse modo os arquivos em rede pode ser o mesmo que deixar os aplicativos e arquivos hospedados em uma nuvem que consiste em milhares de computadores e servidores, todos ligados entre si e acessíveis via *internet*. Com a computação em nuvem, tudo que um usuário faz é baseado na *web* em vez de ser em área de trabalho local.

O aumento da utilização do modelo de computação em nuvem em ambiente corporativo é crescente, pois é financeiramente menos oneroso para seus usuários. Isso ajuda reduzir a economia de recursos computacionais, visto que a promessa da computação em nuvem é prover os recursos de modo virtualizado com maior rapidez, sem ter a despesa de

aquisição de uma nova infraestrutura, licenciamento de *software* e treinamento de pessoal. (HASSAN, 2013)

A Figura 1 ilustra como a nuvem torna possível o acesso às suas informações de qualquer lugar e a qualquer momento, enquanto o modelo de computação tradicional obriga que o usuário acesse suas informações em um único local.

Figura 1 - Ambiente Computacional com Base na Computação em Nuvem



Fonte: Adaptado de Business Innovation Technologies Inc (2013).

Um exemplo de como os recursos providos pela computação em nuvem são disponibilizados através da *internet* é o *GoogleApps* da empresa Google³ que permite acessar serviços através do *browser* em milhões de computadores. Estes recursos são distribuídos e acessíveis a partir de qualquer momento e qualquer lugar em todo o mundo usando a *internet*, tendo-os como serviços sob demanda combinados para alcançar maior rendimento e ser capaz de resolver problemas de computação em grande escala. (SHAIKH e HAIDER, 2011)

A computação em *grid* e a computação em nuvem compartilham os mesmos objetivos de redução de custos, aumento de flexibilidade e confiabilidade através da utilização de *hardware* operado por terceiros. O *grid* é uma coleção de recursos computacionais e de comunicação utilizados para execução de aplicações, através de um tipo especial de *middleware*⁴ que permite o compartilhamento e gerenciamento, baseados nas necessidades dos usuários que utilizam os recursos atribuídos. A maior distinção entre os dois diz respeito à alocação de recursos. No caso do *grid*, tenta-se fazer uma distribuição uniforme de recursos, e

³ *Google* - Empresa multinacional de serviços *online* e *software* dos Estados Unidos. O *Google* hospeda e desenvolve uma série de serviços e produtos baseados na *internet*.

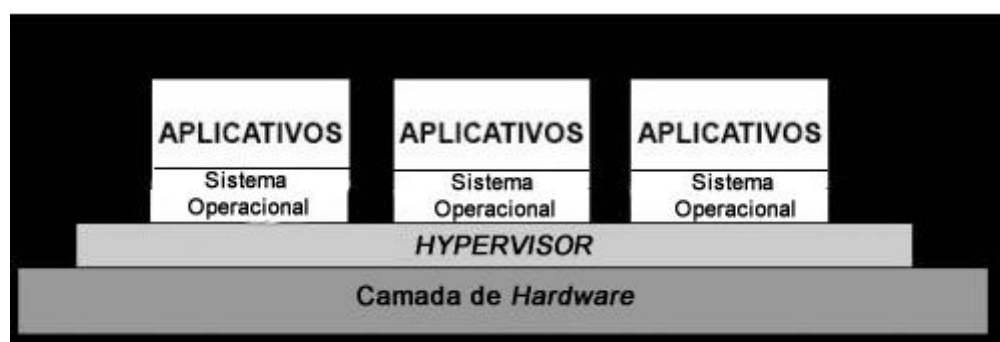
⁴ *Middleware* - Camada de *software* intermediária, utilizada para mover ou transportar informações e dados entre programas de diferentes protocolos de comunicação, plataformas e dependências do sistema operacional.

em um ambiente de computação em nuvem os recursos são alocados sob demanda. Outra diferença é referente a utilização dos recursos, pois a virtualização garante uma separação entre os recursos utilizados pelos vários usuários em ambientes de computação em nuvem. (ZHANG et al., 2010).

A virtualização é a maneira prática e rápida utilizada na criação dos recursos computacionais utilizados na computação em nuvem. Suas principais características são utilizar, gerenciar e prover de maneira otimizada a capacidade de dispositivos físicos, evitando assim a sua subutilização. Apesar de muitos acreditarem que ela é aplicada apenas a servidores, esse conceito pode ser estendido para *desktops*, aplicações e perfis de usuário. (VERAS, 2012).

A Figura 2 apresenta o exemplo do *VMWare ESX Server*, que é uma plataforma de virtualização (*hypervisor*), tendo este a função de criar e executar máquinas virtuais simultaneamente em uma única máquina física. É capaz de abstrair processador, memória, armazenamento e recursos de rede em várias máquinas virtuais e é usada para a consolidação de servidores, proporcionando altos níveis de desempenho, escalabilidade e robustez. A instalação é realizada diretamente no *hardware* do servidor, provendo virtualização abaixo do sistema operacional. O servidor físico é particionado em várias máquinas virtuais, que são cópias do sistema completo. Nelas é possível executar sistemas operacionais e aplicações *Windows*, *Linux*, *Solaris* entre outros, sem qualquer modificação. (VMWARE, 2013).

Figura 2 - Virtualização de Servidores



Fonte: Elaborado pelo autor com base em (VMWARE, 2013).

A computação em nuvem requer uma infraestrutura de computação dinâmica, tendo como base a infraestrutura física. Deve haver níveis de redundância para garantir altos níveis de disponibilidade, mas principalmente deve ser simples para estender o crescimento de uso que se exige. Dentre isso, é possível compreender sobre as características essenciais e quais vantagens e soluções são oferecidas pela computação em nuvem. (NIST, 2014)

Figura 3 - Características de Computação em Nuvem



Fonte: Adaptado de Veras (2012).

A Figura 3 ilustra as características essenciais de um ambiente em nuvem, que são apresentadas a seguir. (VERAS, 2012)

- **Serviço sob demanda:** O usuário pode adquirir unilateralmente os recursos computacionais, como tempo de processamento no servidor ou armazenamento na rede na medida em que necessite e sem precisar de interação humana com os provedores de cada serviço;
- **Amplo acesso à rede:** Os recursos computacionais encontram-se disponíveis através da *internet* e podem ser acessados através de mecanismos padronizados, que possibilitem o uso por plataformas heterogêneas, como por exemplo, telefones móveis, *tablets*, computadores pessoais ou outras tecnologias;
- **Pooling de recursos:** Os recursos computacionais do provedor (físicos ou virtuais) são organizados em um *pool* para servir múltiplos usuários, sendo alocados e realocados dinamicamente conforme a demanda do usuário. Como exemplos, recursos que incluem armazenamento, processamento, memória, largura de banda de rede e máquinas virtuais;
- **Elasticidade:** Os recursos podem ser adquiridos de forma rápida e elástica e, em alguns casos, automaticamente, caso haja necessidade de escalar com o aumento da demanda e liberar na retração dessa demanda. Para o usuário tudo deve ser transparente, dando a impressão de que ele possui recursos ilimitados, que

podem ser adquiridos (comprados) em qualquer quantidade e a qualquer momento, e

- **Medição de serviços:** Os sistemas de gestão utilizados para a nuvem, controlam, monitorizam e otimizam automaticamente o uso de recursos em cada tipo de serviço (processamento, armazenamento, largura de banda e ativos). O uso de recursos pode ser monitorado e controlado, possibilitando transparência para o provedor e para o usuário do serviço utilizado.

2.2 Arquitetura da computação em nuvem

O modelo de computação em nuvem depende das necessidades das aplicações que serão disponibilizadas. A restrição ou abertura de acesso depende do processo de negócios, do tipo de informação e do nível de visão desejado. Certas organizações não desejam que todos os usuários possam ter permissão para utilizar determinados recursos no seu ambiente de computação em nuvem. Surge assim, a necessidade de ambientes mais restritos, onde somente alguns usuários devidamente autorizados possam utilizar os serviços providos. (NIST, 2014)

A arquitetura da nuvem pode ser implantada de quatro formas, dependendo dos requisitos dos usuários e são descritas a seguir: (VAQUERO et al., 2009)

- **Nuvem Privada:** Para este modelo de infraestrutura a nuvem é proprietária ou alugada por uma única organização sendo exclusivamente executada pela mesma. Diferentemente de um *datacenter* privado virtual, a infraestrutura utilizada pertence ao provedor, e, portanto, ele possui total controle sobre como as aplicações são implementadas na nuvem. Uma nuvem privada é, em geral, construída sobre um *datacenter* privado;
- **Nuvem Pública:** Compreende uma infraestrutura de uma organização que vende serviços para o público em geral e pode ser acessada por qualquer usuário que conheça a localização do serviço. A implementação de uma nuvem pública considera questões fundamentais como desempenho e segurança. A existência de outras aplicações executadas na mesma nuvem permanece transparente tanto para os prestadores de serviços como para os usuários;
- **Nuvem Comunitária:** A infraestrutura deste modelo de nuvem é compartilhada por diversas organizações ou até uma terceira que normalmente possuem

interesses comuns, como requisitos de segurança, políticas, aspectos de flexibilidade e ou compatibilidade, e

- **Nuvem Híbrida:** Sua composição é realizada por pelo menos duas nuvens, que preservam as características originais do seu modelo, porém, estão interligadas por uma tecnologia que possibilita a portabilidade de informações e de aplicações.

Cada modelo de implantação requer uma particularidade para migração ou implantação do novo ambiente em nuvem; tudo depende do negócio, informação desejada e da visão comercial. (VERAS, 2012)

2.3 Modelos de níveis de serviços em computação em nuvem

A computação em nuvem distribui os recursos na forma de três modelos de níveis de serviços: aplicação como um serviço (*Software as a Service* - SaaS), plataforma como um serviço (*Platform as a Service* - PaaS) e infraestrutura como um serviço (*Infrastructure as a Service* - IaaS), que são descritos a seguir: (NIST, 2014)

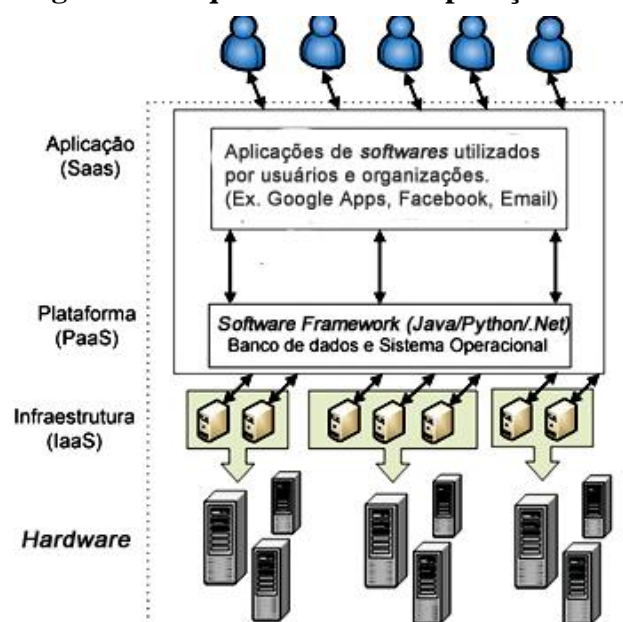
- **SaaS (*Software como serviço*):** O consumidor não administra ou controla a infraestrutura básica, incluindo nuvens de rede, servidores, sistemas operacionais, armazenamento, ou mesmo capacidades de aplicação individual, com a possível exceção de limitada aplicação específica e definições de configuração de utilizadores. De acordo com Veras (2012), aplicações de interesse de uma grande quantidade de usuários passam a ser hospedadas na nuvem como uma alternativa ao processamento local. As aplicações são oferecidas como serviços por provedores e acessadas pelos usuários por aplicações como o *browser*. Todo o controle e gerenciamento da rede, sistemas operacionais, servidores e armazenamento é feito pelo provedor de serviço;
- **PaaS (*Plataforma como serviço*):** É a camada intermediária, que oferece serviços à plataforma, além de proporcionar o ambiente para hospedagem de aplicativos do usuário. Neste caso, o usuário programador tem acesso e controle das configurações sobre as aplicações implantadas, ambientes para desenvolvimento, linguagens de programação e algumas configurações do sistema operacional. Para Vaquero et al. (2009) o objetivo é facilitar o

desenvolvimento de aplicações destinadas aos usuários de uma nuvem, criando uma plataforma que agiliza esse processo, e

- **IaaS (Infraestrutura como serviço):** A administração da infraestrutura e a localização é de responsabilidade do provedor, também fornece acesso com maior nível de configuração sobre os recursos disponibilizados, tais como servidores, roteadores, sistemas de armazenamento e outros recursos de computação. Para Veras (2012), é definida a capacidade que o provedor tem de oferecer uma infraestrutura de processamento e armazenamento de forma transparente. Neste cenário, o usuário não tem o controle da infraestrutura física, mas, através de mecanismos de virtualização, possui controle sobre os sistemas operacionais, armazenamento, aplicações instaladas e possivelmente um controle limitado dos recursos de rede.

Segundo Buyya (2008), a arquitetura de computação em nuvem é baseada em camadas, em que cada camada trata de uma particularidade na disponibilização de recursos. A Figura 4 apresenta cada uma dessas quatro camadas: a camada de *hardware/datacenter*, a camada de infraestrutura, a camada de plataforma e a camada de aplicação. Para Leimester et al., (2010) cada camada representa um nível de abstração, escondendo do usuário todos os componentes subjacentes e permitindo assim o acesso simplificado aos recursos ou funcionalidade, descritos a seguir.

Figura 4 - Arquitetura da Computação em Nuvem



Fonte: Elaborado pelo autor com base em (BUYYA, 2008).

- **Camada de *hardware*:** Em sua composição tem-se servidores, roteadores e *switches*. Está tipicamente implementada em um *datacenter*, com milhares de servidores organizados em *racks* e interconectados por roteadores e *switches*. Nessa camada compreende à configuração de *hardware*, tolerância a falhas, gerência de tráfego, energia, gerência de refrigeração. Dentro da computação em nuvem, esta é a única camada que não pode ser sublocada, pois não há virtualização dos recursos. Ou seja, a sublocação desta camada equivale ao aluguel físico dos equipamentos, caracterizando assim terceirização dos recursos físicos de computação, como máquinas, servidores e cabeamento, portanto, não se trata de computação em nuvem; (AVILES et al., 2012)
- **Camada de infraestrutura:** Visa o gerenciamento dos servidores físicos e o aproveitamento do potencial computacional. É essa camada que cede recursos para que as outras camadas possam utilizá-las. Quando esta camada é utilizada pelo consumidor, a computação em nuvem caracteriza-se no modelo de infraestrutura como serviço (IaaS). Em seu modelo tem-se um conjunto de recursos computacionais e de armazenamento através do particionamento dos recursos físicos utilizando tecnologias de virtualização (*Xen*, *KVM*, *VMWare*, entre outros). A atribuição dinâmica destes recursos pode ser realizada graças à virtualização;
- **Camada de plataforma:** Oferece a linguagem de programação para os desenvolvedores de aplicação na nuvem. O ambiente de *software* também oferece um conjunto de interfaces de programação de aplicações, conhecidas como *API* (*Application Performance Interface*). Sendo alocada para o usuário, caracteriza-se no modelo de plataforma como serviço (PaaS). Algumas de suas características relacionam-se ao uso do sistema operacional e *frameworks* de aplicação, que minimizam o ônus de disponibilizar aplicações diretamente sobre as VMs (Máquinas virtuais). Como exemplo, tem-se o *Google App Engine*, que opera no nível de plataforma para prover *APIs* (que suportam implementação de armazenamento / bancos de dados e lógica de aplicações *web* típicas), e
- **Camada de aplicação:** Geralmente é acessada por portais através da *internet* usando uma interface na qual o usuário interage quando usa a computação em nuvem. Caracteriza-se o modelo de *software* como serviço (SaaS). É a camada

mais visível ao usuário final, portanto a mais comumente usada como serviço. (AVILES et al., 2012)

2.4 Segurança na computação em nuvem

É um quesito que deve ser bem analisado para o ambiente da computação em nuvem, uma vez que existe o uso de seus respectivos serviços por terceiros, sendo utilizado para hospedar informações importantes ou executar operações críticas. Neste contexto, há também os quesitos legais que se deve dar atenção, pois quando estes dados são movidos para uma nuvem, os provedores podem optar em armazená-los em qualquer lugar (a localização física dos *datacenters* implica em estar sujeito a um conjunto de leis do país ou estado, que podem ser aplicadas para o gerenciamento das informações ali armazenadas).

As ameaças de segurança podem ocorrer no provisionamento de recursos e durante a execução de aplicativos distribuídos. Serviços de nuvem devem preservar a integridade dos dados e a privacidade do usuário e, ao mesmo tempo, devem aumentar a interoperabilidade entre vários prestadores de serviços em nuvem.

Os problemas de segurança são voltados especificamente para fornecedores de nuvem pública, uma vez que as organizações têm um maior controle de cada camada de segurança, quando o modelo de implantação de nuvem privada é usada. (NIST, 2014)

O Quadro1 apresenta um resumo das questões de segurança para o ambiente da computação em nuvem.

Quadro 1- Definição das Questões de Segurança em Nuvens.

Questão	Descrição
Governança / concordância	Concentra-se em políticas e procedimentos necessários para ser seguido pelas unidades organizacionais. Também levanta uma questão dos riscos de segurança da informação, uma vez que o risco empresarial ocorre devido à falta de controle dos serviços oferecidos pela nuvem, e recomenda o uso de ferramentas de auditoria e a implantação de programa de gestão de risco.
Confiança	Aborda sobre vários tópicos e questões de ameaças internas causadas por multi inquilino, propriedade de manutenção de dados, direitos de propriedade intelectual, risco da gestão, ganho de visibilidade e controle de segurança oferecido pelo CSP (Provedor de Serviços da Nuvem).
Arquitetura	Discute as questões relativas a <i>software</i> e sistemas utilizados pela plataforma de nuvem. A maior parte das questões discutidas relaciona-se as suas características de plataformas de computação em nuvem, que são completamente diferentes em comparação aos tradicionais <i>datacenter</i> . Para arquitetura são tratadas questões como segurança do <i>hypervisor</i> , proteção de rede virtual, imagens de máquinas virtuais e proteção do lado do usuário.
Gerenciamento do acesso e identidade	Aborda a verificação da identidade, autenticação e mecanismos de controle de acesso. É também recomendado o uso do <i>Security Assertion Markup Language</i> (SAML) para a troca de dados de autenticação e autorização entre domínios e <i>Extensible Access Control Markup Language</i> (XACML) para definição do controle de acesso.
<i>Software</i> de isolamento	Adverte sobre as ameaças associadas com multi inquilino, assim como do vetor de ataque.
Proteção das informações	Verifica a necessidade de privacidade de dados e isolamento, como as informações de diferentes usuários hospedadas em um mesmo <i>datacenter</i> na plataforma de computação em nuvem.
Disponibilidade	Discute sobre as ameaças que têm um impacto negativo para os recursos organizacionais.
Resposta de incidente	Foca na reativação de contramedidas para ataques de ameaças em um ambiente de nuvem.

Fonte: Adaptado de NIST (2014).

2.5 Vantagens e riscos da computação em nuvem

A computação em nuvem proporciona uma economia financeira convincente em TI, incluindo menores custos de implementação e manutenção; menos *hardware* para aquisição e apoio; eliminação do custo de energia, refrigeração, espaço físico e de armazenamento. A administração dos recursos a serem disponibilizados e utilizados para a empresa que adere ao modelo de nuvem fica a cargo do prestador de serviço que foi contratado; com isso, obtém-se uma redução nos custos operacionais e paga-se apenas pelo que é utilizado (serviço medido). A computação em nuvem também permite que as organizações aumentem sua competitividade devido a utilização de plataformas de computação flexíveis e ágeis e prevê escalabilidade e de alto desempenho, aplicativos e dados altamente confiáveis e disponíveis.

Segundo Kim (2009) a computação em nuvem oferece quatro importantes benefícios que seus consumidores podem levar em consideração:

- **Redução de investimento e custeio:** Os provedores são os proprietários e gerenciam todos os recursos computacionais, cabendo aos consumidores unicamente conectarem-se às nuvens. Neste contexto, os consumidores não necessitam fazer aquisição de uma grande quantidade de recursos de infraestrutura em computação para dar início a sua operação de TI, tampouco prover espaço físico para instalações ou ter um elevado custo de consumo de eletricidade, manutenção operacional e de remuneração de pessoal especializado;
- **Escalabilidade dinâmica:** Os consumidores podem aumentar ou reduzir o nível de utilização de serviços ou recursos computacionais de modo simples e com flexibilidade;
- **Menor preço relativo:** Em geral, o custo dos serviços na computação em nuvem são inferiores aos dispêndios demandados por uma operação própria, pois os consumidores pagam apenas pelos recursos ou serviços efetivamente utilizados, sem a necessidade de manter uma parcela de capacidade ociosa para fazer frente a situações de pico, e
- **Facilidade de acesso:** Pode-se acessar as nuvens e demandar serviços a qualquer momento e de qualquer lugar.

A computação em nuvem não é isenta de riscos ou completamente segura. Para os riscos em computação em nuvem, o problema seria a parada imprevista de algum evento, ou propriamente o objetivo do negócio. A computação em nuvem está em constante evolução, com desafios em relação a confiança, eficiência, segurança, custos e legislação, o que influencia o seu uso a partir da perspectiva do seu usuário final. Alguns riscos devem ser observados: (BUYYA et al., 2013)

- **Acesso compartilhado:** Refere-se à nuvem pública que é o modelo de multi inquilino, uma única instância lógica compartilhada por centenas ou milhares de usuários. Por conta dos riscos que envolvem o compartilhamento de recursos, vulnerabilidades de multi inquilino são muito preocupantes, porque uma falha pode permitir que outro inquilino ou atacante veja todos os outros dados ou assuma a identidade de outros usuários;
- **Vulnerabilidades virtuais:** Os servidores virtuais estão sujeitos aos mesmos ataques que atingem os servidores físicos, assim como novas ameaças estão explorando falhas do *hypervisor*;
- **Autenticação, autorização e controle de acesso:** A privacidade e a integridade das informações são itens de suma importância, pois, especialmente em nuvens públicas, existe uma grande exposição a ataques. Dentre as capacidades requeridas para evitar a violação das informações estão: a criptografia dos dados, o controle de acesso rigoroso e o sistema eficaz de gerenciamento de cópias de segurança, (KAUFMAN, 2009) e
- **Disponibilidade:** Para quem usa os serviços de nuvem pública, redundância e tolerância a falhas não estão sob seu controle. O usuário não tem a garantia que as medidas cabíveis, como a realização de *backups*, são realizadas. Uma possível solução para este problema seria implementar micro nuvens privadas, nas quais apenas os aplicativos importantes seriam disponibilizados, retornando à nuvem pública assim que o provedor volta a operar. (ARMBRUST et al., 2009)

A computação em nuvem, ainda projeta algumas contradições, no que se refere a seus conceitos, na padronização de comercialização e implementação. Atribuição de preço pelo tipo de serviço oferecido, provisionamento autônomo de serviços, análise e gerência de

tráfego, tecnologias para armazenamento e gerência de informações, localização e instanciação automática de máquinas virtuais e gerência de execução de *workflows* em IaaS concorrentes, são exemplos citados de contradições em meio ao ambiente de nuvem. (BUYYA, 2008)

2.6 Orquestradores de computação em nuvem

Os orquestradores de computação em nuvem são responsáveis por realizar a harmonização e o gerenciamento dos recursos virtuais e componentes de infraestrutura no ambiente de nuvem, permitindo a alocação de servidores, plataformas e aplicações pelo próprio cliente.

Nesta seção são descritos alguns orquestradores, apresentadas algumas de suas características básicas e uma visão geral de sua infraestrutura.

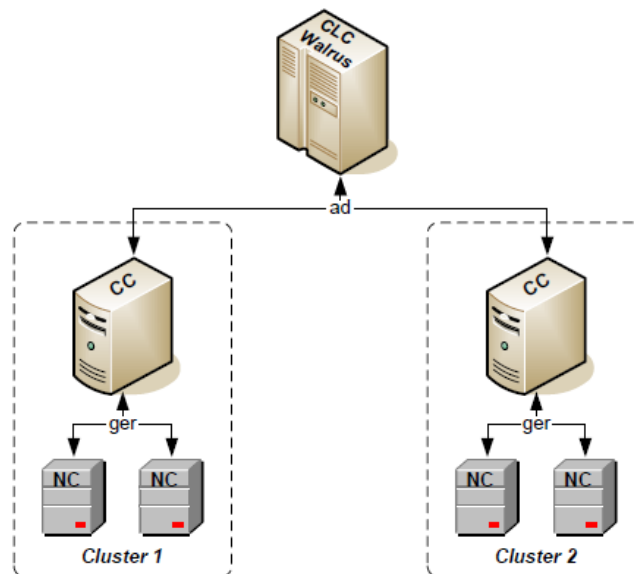
2.6.1 *Eucalyptus*

Eucalyptus é uma plataforma de computação em nuvem de código aberto que permite construir sua própria nuvem privada de uma empresa corporativa. (SOUSA et al., 2014) Aproveita a infraestrutura de TI existente para criar uma nuvem privada *self-service*, ou seja, possibilita ao usuário criar sua própria estrutura de nuvem. Uma Infraestrutura como Serviço (IaaS) pode ser ativada, abstraindo os recursos de computação, rede e armazenamento disponíveis. O *Eucalyptus* cria um *pool* de recursos escaláveis que pode dimensionar dinamicamente, dependendo da demanda da aplicação. Através de uma parceria com a *Amazon Web Services* (AWS)⁵, mantém a compatibilidade da API, possibilitando aos usuários deslocar cargas de trabalho entre ambientes AWS e *Eucalyptus*.

Conforme ilustrado na Figura 5, a arquitetura é composta pelos componentes *Node Controller*, *Cluster Controller*, *Cloud Controller*, *Storage Controller* e *Walrus*, descritos a seguir. (EUCALYPTUS, 2014)

⁵*Amazon Web Services* (AWS) oferece um conjunto completo de serviços de aplicativos e infraestrutura que permitem executar praticamente tudo na nuvem desde aplicativos empresariais, armazenamento em banco de dados, jogos sociais e aplicativos móveis.

Figura 5 - Arquitetura do Orquestrador Eucalyptus



Fonte: Adaptado pelo autor com base em (NURMI et. al., 2009).

- **Node Controller (NC)** é o componente que executa os recursos físicos de uma máquina virtual. Desta forma, pode-se alocar apenas um *Node Controller* numa máquina física e diversos outros *Node Controllers* rodando em máquinas virtuais. Cada *Node Controller* é responsável por controlar a inspeção, execução e o término da respectiva instância;
- **Cluster Controller (CC)** é o componente responsável por realizar o roteamento de pacotes entre as redes virtualizadas externa (pública) com a interna (privada). Desse modo, todas as manipulações realizadas nos recursos físicos alocados nos *Node Controllers* devem ser, necessariamente, gerenciadas pelo *Cluster Controller*;
- **Cloud Controller (CLC)** é a entrada para aos administradores, desenvolvedores e usuário final. O *Cloud Controller* é responsável por consultar o nó de controle para obter informações sobre os recursos físicos, tomar decisões de programação de alto-nível e implementá-las por meio de requisições aos controladores dos clusters;
- **Storage Controller** é o componente que implementa o acesso aos blocos de armazenamento em rede, como por exemplo *Amazon Elastic Block Storage (EBS)* e é capaz de interagir com sistemas de armazenamento, como o NFS e iSCSI. Um bloco elástico pode ser anexado a uma máquina virtual com Linux, porém todo o tráfego de disco é enviado para um local de armazenamento remoto, e

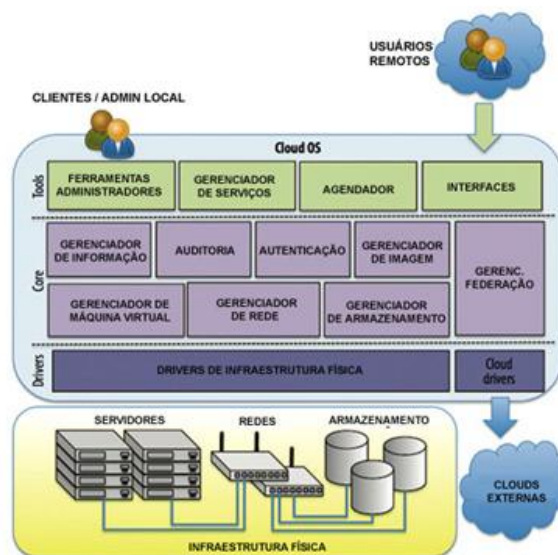
- **Walrus** é o módulo que permite aos usuários armazenar dados e organizá-los em *buckets*⁶. Além disso, no *Walrus* é possível inserir políticas de acesso aos *buckets* e aos dados armazenados. (NURMI et. al, 2009).

2.6.2 OpenNebula

O *OpenNebula* é um kit de ferramentas de computação em nuvem para gestão de infraestruturas de *datacenters* distribuídos. O *toolkit OpenNebula* gerencia uma infraestrutura virtual de um *datacenter* para construir implementações privadas, públicas e híbridas de infraestrutura como serviço (IaaS).

Este *toolkit* inclui recursos para integração, gerenciamento, escalabilidade, segurança e contabilidade. Ele também possui padronização, interoperabilidade e portabilidade, oferecendo aos usuários de nuvem e administradores a possibilidade de escolha de várias interfaces em nuvem (*Amazon EC2*, *OGF Open Cloud Computing interface* e *vCloud*) e *hypervisors* (*Xen*, *KVM* e *VMware*). Oferece uma solução simples, mas com recursos e flexibilidade para construir e gerenciar nuvens empresariais e *datacenters* virtualizados. Foi projetado para ser simples de instalar, atualizar e operar pelos administradores, e simples de ser utilizado pelos usuários finais. A arquitetura do *OpenNebula* é composta por três camadas para o gerenciamento da infraestrutura *Drivers*, *Core* e *Tools*, conforme pode ser apresentado na Figura 6. (OPENNEBULA, 2014)

Figura 6 - Arquitetura do Orquestrador OpenNebula



Fonte: Adaptado de *OpenNebula* (2014).

⁶*Buckets* - Criação de uma área de armazenamento de dados (*container*).

- **Drivers:** realiza a comunicação com a plataforma de sistema operacional, responsável por funções de criação, inicialização e desligamento de máquinas virtuais (VMs), armazenamento e provisionamento de VMs e monitora o estado operacional de máquinas físicas e virtuais;
- **Core:** gerencia os ciclos de vida das máquinas virtuais e realiza a criação de rede virtual dinamicamente com atribuição de endereço IP dinâmico para uma VM. Com isso, torna-se uma solução de rede transparente para os usuários, e
- **Tools:** são interfaces de linha de comando (CLI) com uso de *API's* que realizam a comunicação e gerenciamento de informações de usuários e máquinas virtuais. Há possibilidade de realizar interfaceamento direto na camada de *core*, proporcionando aos usuários externos acesso às funcionalidades de um recurso como, por exemplo, VM's.

2.6.3 Xen Cloud Platform

O *Xen Cloud Platform* é uma solução *opensource* e gratuita para projeto de virtualização de servidor e plataforma de computação em nuvem. Combina as funcionalidades do *hypervisor Xen* como segurança, armazenamento, tecnologias de virtualização de rede, e todas as ferramentas necessárias para construção do ambiente em computação em nuvem. Destina-se a apoiar as implantações de nuvem privada e pública, originalmente derivado do *XenServer*,⁷ que é uma solução *Xen* inventada pela Citrix. (ALMURAYH e SEMWAL, 2013)

Segundo *Citrix* (2014), o XCP inclui o *Xen Hypervisor*, a funcionalidade de *API toolstack Xen* com funcionalidades como: capacidade de gerenciar *pools* de sistemas *host*, suporte para repositórios de armazenamento avançadas, suporte para multi inquilino, suporte para garantias de SLA, entre outros. Além disso, ele possui integração com outros orquestradores de nuvem, como o *Apache CloudStack*, *OpenNebula* e *OpenStack*. Algumas de suas características são:

- Ciclo de vida da VM através de *snapshots*, ponto de verificação e migração;
- *Pools* de recursos: armazenamento flexível e rede;
- Acompanhamento de eventos: progresso e notificação;

⁷*XenServer* é uma distribuição comercial da Citrix para XCP, onde está licenciado sob GNU *General Public License* (GPL2).

- Capacidade de atualização automática de pacotes;
- Monitoramento de desempenho em tempo real e alerta, e
- Suporte integrado e modelos para os usuários *Windows* e Linux.

2.6.4 Microsoft Windows Azure Platform

Suporta aplicativos criados no *NET Framework* e outras linguagens comuns suportadas em sistemas *Windows*, como C#, *Visual Basic*, C++ e outros. *Windows Azure* suporta programas de uso geral, ao invés de uma única classe de computação. Os desenvolvedores podem criar aplicações web utilizando tecnologias como o ASP.NET e *Windows Communication Foundation* (WCF). Estes aplicativos são executados como processos em segundo plano independentes, ou aplicações que combinam os dois. Essa tecnologia foi desenvolvida para oferecer suporte a aplicativos que são escalonáveis, confiáveis e cuja operação é de baixo custo. (AZURE, 2014)

O orquestrador *Windows Azure*, conforme apresenta a Figura 7, pode ser usado por aplicativos executados na nuvem e por aplicativos executados em sistemas locais e consiste em três componentes e cada um deles fornece um conjunto específico de serviços para usuários em nuvem. Oferece um ambiente baseado em *Windows* para execução de aplicativos e armazenamento de dados em servidores e nos *datacenters* *SQL Azure* e também fornece serviços de dados na nuvem baseado em *SQL Server*. Enquanto que o *NET Services* oferece serviços de infraestrutura distribuídos para aplicativos baseados em nuvem e locais. (ZHANG, 2015)

Figura 7 - Plataforma Windows Azure



Fonte: Adaptado de *Windows Azure* (2014).

Ainda segundo *Azure* 2014, algumas de suas principais características são:

- **CDN (Rede de distribuição de conteúdo):** solução de *cache* distribuído ao redor do mundo. O objetivo principal é reduzir a latência de comunicação entre as aplicações;
- **HPC (High Performance Computing):** é um serviço de computação de alto desempenho. Neste modelo é cobrada a hora de computação de acordo com o tipo de servidor escolhido. Exemplos de aplicações comumente encontradas em HPC são aquelas que utilizam código assíncrono e *multithreads*⁸;
- **SQL Azure, Azure Tables e Blogs:** recursos para armazenamento de dados disponíveis no *Windows Azure*. *SQL Azure* é o modelo relacional de armazenamento de dados. *SQL Azure* foi desenvolvido com os mesmos recursos do *SQL Server* tradicional, assim, é possível executar praticamente todas as operações já tradicionalmente realizadas no ambiente *on-premises*⁹. Disponibiliza também um modelo de armazenamento de dados não estruturados, chamado de *Azure Tables*. Profissionais acostumados a trabalharem com *NoSQL* tem suas necessidades atendidas com este modelo;
- **Service Bus:** recurso que possibilita a integração entre diferentes sistemas através de serviços diversos. *Service Bus* pode ser entendido como um “*hub* de integração”;
- **Windows Azure Connect:** recurso responsável por garantir a qualidade de conexão entre a nuvem e o(s) ambiente(s) *on-premise*(s) ao(s) qual(is) ocorrerá esta(s) conexão(es);
- **Active Directory:** ao sofrer consideráveis melhorias em diversos aspectos o *Access Control Service* (ACS) tornou-se o que hoje, sendo conhecido como *Windows Azure Active Directory*. Em linhas gerais, WAAD é um *hub* dedicado a disponibilização de serviços para autenticação de usuários, e
- **Media Services:** recurso dedicado a produção e compartilhamento de recursos de mídia (vídeo e áudio).

⁸*Multithreads* - É a capacidade que o sistema operacional possui de executar várias *threads* (conjunto de tarefas existentes em um ou mais programas, executadas ao mesmo tempo pelo processador) simultaneamente sem que uma interfira na outra. Estas *threads* compartilham os recursos do processo, mas são capazes de ser executadas de forma independente.

⁹*On-premises*- Sistemas ou aplicativos instalados em um recurso local, como um computador, sem a necessidade de acesso a internet para manter seu funcionamento.

2.6.5 OpenStack

O orquestrador de nuvem do *OpenStack* é um *software* de código aberto, capaz de gerenciar os componentes de múltiplas infraestruturas virtualizadas, assim como o sistema operacional gerencia os componentes de computadores. O *OpenStack* é chamado de Sistema Operacional da Nuvem, por cumprir o mesmo papel em maior escala. É considerada uma plataforma de *software* por fornecer *APIs* que em conjunto são capazes de controlar todos os recursos disponíveis na oferta dessa infraestrutura, como máquinas virtuais, rede, armazenadores e balanceadores de carga. Até mesmo um painel de controle *web* está presente no *OpenStack*, em sua maior parte escrita em linguagem *Python*¹⁰. (OPENSTACK, 2014)

O *OpenStack* é uma coleção de projetos de *software* de código aberto que as empresas ou provedores de nuvem podem utilizar para configurar e executar a sua própria computação em nuvem e infraestrutura de armazenamento. O projeto visa construir uma comunidade *opensource* com pesquisadores, desenvolvedores e empresas. É uma das maiores comunidades de desenvolvimento de computação em nuvem *middleware*¹¹ de código aberto do mundo, suporta virtualização com *KVM*, *XEN*, e *HyperV*, usando o emulador *QEMU*. Na implementação, o *libvirt*, biblioteca C / C ++, é usado para se comunicar com o *hypervisor*¹² da camada de *middleware* (WEN et al., 2012).

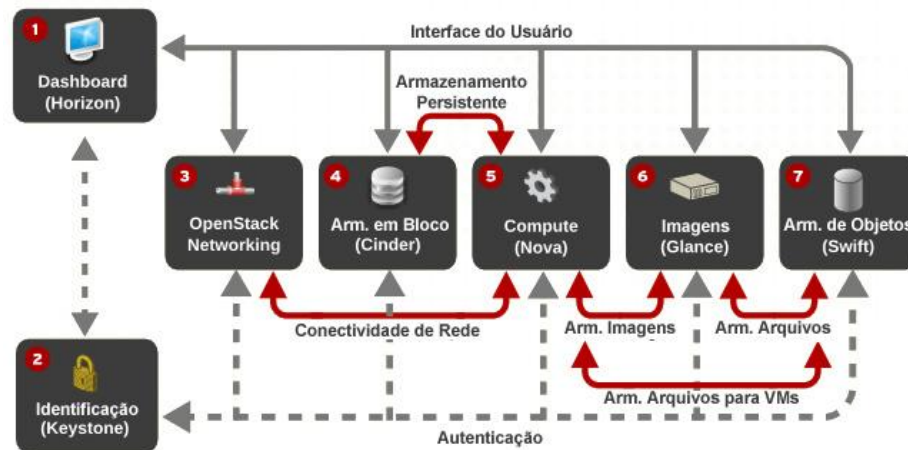
Como uma plataforma de infraestrutura virtual, o *OpenStack* gerencia um grupo de recursos virtuais interdependentes, como máquinas virtuais, redes virtuais e volumes, e define os vários níveis do detalhamento de recurso virtual. A Figura 8 apresenta a arquitetura do *OpenStack*, dividida em componentes descritos a seguir: (CASTILLO et al., 2013).

¹⁰*Python* - uma linguagem de programação de alto nível, interpretada, imperativa, orientada a objetos, funcional e de tipagem dinâmica.

¹¹*Middleware* - Camada adicional de *software* situada entre o nível de aplicação e o nível que consiste no sistema operacional, oculta da melhor maneira possível a heterogeneidade das plataformas das aplicações.

¹²*Kypervisor* (Monitor de Máquinas Virtuais) -É uma plataforma de *software* que permite executar múltiplos sistemas operacionais simultaneamente em uma máquina física (*host*).

Figura 8 - Arquitetura do Orquestrador *OpenStack*



Fonte: Adaptado de WEN *et al.* (2012).

- **Swift:** Permite armazenar e recuperar arquivos que não estejam montados em um diretório. É um sistema de armazenamento em longo prazo para dados permanentes ou estáticos;
- **Horizon:** Provê uma interface *web* modular para todos os componentes do OpenStack. Com a interface *web* visual é possível gerenciar mais facilmente os componentes da nuvem;
- **Nova (gerenciamento da nuvem):** É o *software* que controla a infraestrutura de IaaS, alocando ou liberando recursos computacionais, como rede e autenticação;
- **Glance (serviço de imagens):** Fornece um catálogo e repositório para imagens de disco virtuais. Estas imagens são mais utilizadas no Nova, que gerencia estas imagens;
- **Cinder (armazenamento de blocos):** Fornece um armazenamento de blocos para as máquinas virtuais;
- **Neutron (antigo Quantum - serviço de rede):** Provisiona serviços de rede para os componentes que são gerenciados pelo Nova. Permite aos usuários criar e anexar sua interface a ele, e
- **Keystone (serviço de identidade):** Fornece autenticação e autorização para todos os serviços do OpenStack.

Segundo Hu e Yu (2013), o *OpenStack* é projetado para fornecer o máximo de flexibilidade sem requisitos de *hardware* ou de *software* proprietário, com a capacidade de integração com sistemas legados e tecnologias de terceiros. Além disso, é projetado para gerenciar e automatizar *pools* de recursos de computação e pode trabalhar com tecnologias amplamente disponíveis de virtualização, como *bare metal*¹³ e computação de alta performance (HPC). Os administradores geralmente implantam o *OpenStack* usando um dos vários *hypervisors* suportados em um ambiente virtualizado. O *KVM* e o *XenServer* são as escolhas mais populares para tecnologia de *hypervisor*, além de serem recomendados para a maioria dos casos de uso.

2.6.6 Amazon EC2 (*Elastic Compute Cloud*)

EC2 (*Elastic Compute Cloud*) é um serviço que permite ao usuário alugar os recursos computacionais da *Amazon* e executar máquinas virtuais sobre os *datacenter* da empresa. (Amazon, 2014). Algumas de suas principais características são:

Endereços *Elastic IP*: Refere-se a endereços de IP's estáticos projetados para computação em nuvem dinâmica. Um endereço *Elastic IP* está associado à uma conta de usuário e não a uma instância específica, sendo que o próprio usuário controla esse endereço até que se decida liberá-lo. *Amazon EC2* permite que o usuário gerencie os problemas com sua instância ou *software* através de remapeamento do seu endereço *Elastic IP* para uma instância de substituição. Além disso, pode-se opcionalmente configurar o registro de DNS reverso de qualquer um dos seus endereços *Elastic IP*.

***Amazon VPC*:** Permite que as empresas conectem sua infraestrutura existente a um conjunto de recursos computacionais isolados da *Amazon*, através de uma conexão de rede privada virtual (VPN). Assim, é possível expandir seus recursos de gestão existentes, tais como serviços de segurança, *firewalls* e sistemas de detecção de intrusão para incluir seus recursos a este orquestrador.

Instâncias de armazenamento denso: ideal para clientes que precisam de densidade de armazenamento alta por instância. Possuem E/S sequencial elevada para aplicações com uso intensivo de dados como um armazenamento de dados com processamento paralelo

¹³*Bare Metal* - É um *hypervisor* de tipo 1, *software* de virtualização que interage diretamente com a camada de *hardware*.

massivo (MPP¹⁴), computação distribuída do *MapReduce*¹⁵ e do *Hadoop*¹⁶ e processamento de *logs* e dados que podem obter benefícios com as instâncias de armazenamento denso.

O EC2 fornece a capacidade de colocar ocorrências em vários locais. No EC2 os locais são compostas por regiões e zonas de disponibilidade. Essas regiões consistem de uma ou mais zonas de disponibilidade dispersas geograficamente. Imagens de máquinas EC2 são armazenadas e retiradas da *Amazon Simple Storage Service (Amazon S3)*. S3 armazena dados como "objetos" que são agrupados em "cubos". Cada objeto contém de 1 *byte* a 5 *gigabytes* de dados, seus nomes de objetos são essencialmente caminhos URI (*Uniform Resource Identifier*). Os cubos devem ser criados explicitamente antes de serem utilizados e podem ser armazenados em várias regiões. Os usuários podem escolher uma região para otimizar a latência, minimizar os custos, ou atender às exigências regulamentares (ZHANG, 2015).

2.6.7 *Vmware vCloud Suite*

O *VMware vCloud Suite* é uma ferramenta comercial de código fonte fechado, para criação e gerenciamento de uma nuvem privada do *VMware vSphere*, que é a plataforma de virtualização de servidores da empresa, contendo o *Hypervisor ESXi*, da própria VMware. É o orquestrador para computação em nuvem da empresa VMware, que permite que os clientes migrem sua carga de trabalho sob demanda a partir de sua nuvem privada de *hypervisor* da VMware para a nuvem pública também com uso dos *hypervisors* VMware. (VMWARE, 2014)

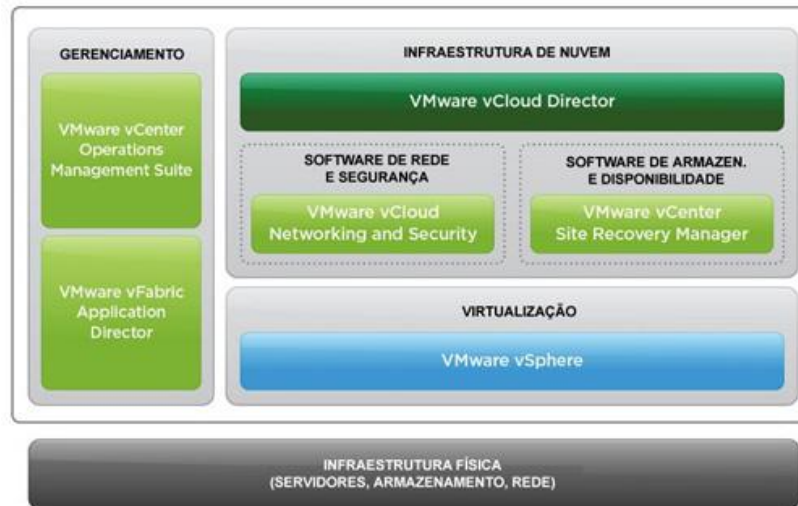
Conforme ilustrado na Figura 9 é composto de diversos módulos que trabalham de forma integrada, baseado em produtos como o *VMware vCenter Operations*, *VMware vFabric Application Director*, *VMware vCloud Networking and Security* e *VMware Recovery Manager*.

¹⁴MPP - É o processamento de um programa de forma coordenada em vários processadores.

¹⁵*MapReduce* - Estrutura que permite escrever aplicativos que processam grandes quantidades de dados estruturados e desestruturados em paralelo, em um *cluster* de milhares de máquinas, de maneira confiável e tolerante a falhas.

¹⁶*Hadoop* - É uma estrutura em código aberto para armazenamento e processamento distribuídos de grandes conjuntos de dados em *hardware*.

Figura 9 - Arquitetura do Orquestrador VMware VCloud



Fonte: Adaptado de VMware (2014).

- **VMware vCenter Operations:** proporciona uma abordagem integrada ao gerenciamento de desempenho do ambiente, através de painéis de controle pré-integrados e configuráveis para realizar a coleta e medições em tempo real;
- **VMware vFabric Application Director:** ferramenta que realiza o desenvolvimento e o provisionamento de soluções ou aplicativos de forma simplificada e padronizada a topologia de uma nuvem IaaS;
- **VMware vCloud Networking and Security:** fornece a entrega de serviços em rede através da criação de dispositivos virtuais como VPN (Rede privada virtual), *firewall*, balanceamento de carga, endereço de rede (NAT) e DHCP, e
- **VMware Recovery Manager:** evita o ponto de acesso em *storage* que esteja subutilizado ou superutilizado. Proporciona mecanismos de posicionamento nas máquinas virtuais e de balanceamento de carga com base em I/O e capacidade de espaço. É utilizado para agregar recursos de armazenamento, habilitando o posicionamento dos arquivos do disco de uma máquina virtual e o balanceamento de cargas de trabalhos existentes.

Por se tratar de uma solução proprietária, o orquestrador não disponibiliza que se realize interfaceamento de seus módulos com uso de outras ferramentas a fim de monitorar e analisar mensagens de comunicação geradas pela *Web vSphere* por exemplo. Com isso, torna-se difícil sistematicamente analisar e monitorar estas mensagens para mensurar o desempenho

adequado da solução de problemas que venham a ocorrer em um ambiente de computação em nuvem. (LIANG et al., 2011)

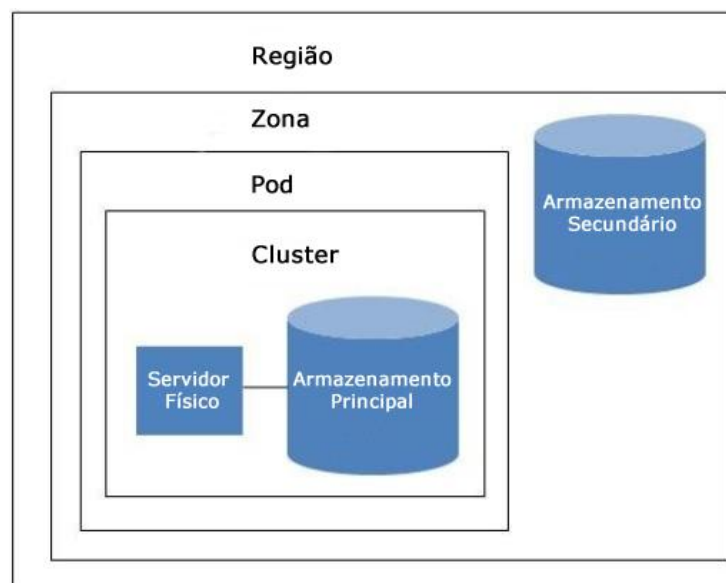
2.6.8 Apache CloudStack

O *Apache CloudStack* é um *software* de código aberto projetado para oferecer serviços de nuvem pública, e por muitas empresas para fornecer uma infraestrutura local (privada) em nuvem, ou como parte de uma solução de nuvem híbrida. Inclui toda a lista de recursos que a maioria das organizações precisam com uma nuvem *IaaS*: gerenciamento de máquinas virtuais, rede como serviço, gerenciamento de usuários e conta e uma *API* nativa e aberta. Atualmente, suporta os principais *hypervisors*: *VMware*, *KVM*, *XenServer*, *Xen Cloud Platform (XCP)* e *Hyper-V*.

O orquestrador integra e gerencia esses recursos virtuais para construir plataforma de nuvem e fornece serviços de nuvem transparente para os usuários. Ele é responsável pelo mapeamento de recursos virtuais para recursos de *hardware* e possibilita o isolamento de segurança entre os usuários. (CLOUDSTACK, 2014)

Conforme apresentado na Figura 10, esse orquestrador é compatível com os recursos de infraestrutura existentes. O design modular permite que o usuário personalize o funcionamento do sistema conforme necessário. Além disso, o *Apache CloudStack* fornece *API* que é compatível com a *AWS EC2* e *S3* para as organizações que desejam implantar nuvens híbridas. (CHEN e XU, 2013)

Figura 10 - Arquitetura do Apache CloudStack



Fonte: CLOUDSTACK (2014).

Os recursos de *hardware* são abstraídos e provisionados em recursos de computação através do gerenciamento e armazenados por meio do uso de *hypervisors*. O *Apache CloudStack* integra e gerencia esses recursos virtuais para construir uma plataforma de nuvem, fornecendo serviços de nuvem transparente para os usuários. Ele é responsável pelo mapeamento de recursos virtuais para recursos de *hardware* e possibilita o isolamento de segurança entre os usuários. (CHEN e XU, 2013)

A seguir é apresentada uma descrição dos componentes que fazem parte da arquitetura do orquestrador *Apache CloudStack*. (BARKAT et. al, 2014)

- **Servidor Físico:** Um *host* é uma representação de um servidor físico, este é virtualizado por *hypervisors* (por exemplo, *Xen*, *KVM*, *vSphere*, *Hyper-V*, *VMWare*), que por sua vez faz o gerenciamento e os serviços a serem provisionados por meio de máquinas virtuais;
- **Cluster:** Dentro de um *cluster*, os *hosts* são acolhidos em um mesmo *pool* computacional com armazenamento primário, em uma mesma sub rede de *IPs*. Para o armazenamento primário, aceita qualquer tipo de armazenamento suportado pelo *hypervisor*. Pode ter mais de um dispositivo de armazenamento principal;
- **Pod:** É uma coleção de diferentes *clusters* ligados através de *switch* na camada 2 e sua representação é de um *rack* de servidor. Os *hosts* estão no mesmo *pod* com a mesma sub rede e não é visível para o usuário final;
- **Zona:** É uma coleção de *pods*. A vantagem do uso da *zone* está no isolamento e na redundância, permitindo com isso que os operadores de nuvem tenham várias *zones* de disponibilidade dentro de um determinado *datacenter*. Os *pods* de uma zona utilizam o armazenamento secundário, através de *switch* na camada 3;
- **Região:** Remete-se a maior unidade organizacional do orquestrador *Apache CloudStack*. As *regions* contêm várias *zones*, distribuídas em localidades geograficamente distintas de cada uma delas, e
- **Management Server:** Responsável pelo gerenciamento de todos os recursos da infraestrutura na nuvem através de *APIs* ou via administração de interface *web*. A implantação deste gerenciador pode ser realizada em um servidor físico ou por uma

máquina virtual. Para uma alta disponibilidade no ambiente do orquestrador, este gerenciador pode ser duplicado, desde que o utilizador visualize as mesmas informações independente de qual servidor acesse. Também existe a necessidade da instalação de um banco de dados, para que o *management server* seja persistente.

Em questão de armazenamento faz-se necessário a configuração de um *host* físico subdividido em dois tipos principais de *storages*: *primary storage* e *secondary storage*, descritos a seguir.

- **Armazenamento Primário:** Este *storage* é associado a um *cluster* ou uma *zone*. Para um mesmo *cluster*, pode-se implantar vários *primary storage*. Este tipo de *storage* é basicamente utilizado pelo *hypervisor* em nível de armazenamento contendo a instalação das máquinas virtuais. Este *storage* interage diretamente com as aplicações na máquina virtual, podendo acarretar um alto custo em termos de entrada/saída para suas operações, e por essa razão deve ser implementado em um meio físico próximo aos seus *hosts* da nuvem, e
- **Armazenamento Secundário:** Utilizado para armazenamento de *template*, *snapshot*, *ISO storage* e ocorrência de eventos.
 - **Imagens *ISO storage*:** utilizada no momento em que um usuário necessita da criação de uma máquina virtual.
 - ***Template*:** é a imagem base do sistema operacional que o usuário pode escolher quando criar uma nova instância. Há possibilidade de se incluir informações adicionais a sua configuração, tais como instalação de aplicativos, e
 - ***Snapshot*:** usado como *backup* de dados e recuperação de serviços.
- ***Network*:** Suporta a utilização de diferentes dispositivos de rede física (por exemplo *NetScaler*, *F5 BIG-IP*, *Juniper*, *SRX*, entre outros). Os usuários tem a possibilidade de escolher dois tipos de cenários de rede: básico e avançado.
 - **Cenário básico:** Trata-se de redes em estilo *AWS*. Provê uma rede única a cada instância de máquina virtual com um endereço *IP* diretamente da rede. O isolamento dos *guests* pode ser provido através de *switch* de camada 3, e

- **Cenário avançado:** É mais flexível para definição de *guest* na rede, por exemplo, assim o administrador pode criar redes múltiplas para serem utilizadas pelo *guest*, por exemplo.

2.7 Considerações Finais

Esse capítulo apresentou um levantamento sobre computação em nuvem, suas principais características, virtualização, modelos de infraestrutura, diferentes recursos e arquitetura, modelos de níveis de serviços, segurança, vantagens e riscos.

A Figura 11 apresenta uma síntese entre os principais orquestradores de nuvem analisados neste capítulo, demonstrando tipo de serviço, versões, tipos de códigos, arquitetura, *hypervisors*, *API*, *drivers* autenticação, algoritmo de escalonamento, movimentação instâncias tempo real, configuração de rede para instâncias, sistema operacional hospedeiro e autenticação para conexão à instância.

Entre os orquestradores abordados, ressalta-se o *Apache CloudStack*, tendo seu maior impulsionador a empresa Citrix, contando com comunidades de desenvolvimento *opensource* e a *Apache Software Foundation*. Outro ponto a ser observado é a questão desta nuvem ser o orquestrador de IaaS (Infraestrutura como Serviço) de empresas de grande porte em ambiente de produção. (CLOUDSTACK, 2015)

Figura 11 - Principais Características dos Orquestradores de Computação em Nuvem

	OpenStack	CloudStack	Eucalyptus	OpenNebula	Xen Cloud Platform	Amazon Web Service	Microsoft Azure	Vmware vCloud Suite
Tipo de Serviço	IaaS	IaaS	IaaS	IaaS	IaaS	IaaS	IaaS	IaaS
Versões código	Aberto	Aberto	Proprietário/Aberto	Aberto	Proprietário/Aberto	Proprietário	Proprietário	Proprietário
Arquitetura	Público/Híbrido/Privado	Público/Híbrido/Privado	Privado/Híbrido	Privado/Híbrido	Público/Híbrido	Público	Público	Público/Híbrido
Hypervisors	KVM,XEN,ESXi(Sim (com ou sem vCenter),Hyper-V, LXC, Docker(não).	KVM,XEN,ESXi(Sim (só com vCenter), Hyper-V(não), LXC(não), Docker(não).	Xen, KVM, VMware	EC2, Xen, KVM e Vmware	Xen Hypervisor	Xen, podem ser criadas com ferramentas livres Interfaces HTTP	Windows Server 2008/2012 Hyper-V	VMware ESX, ESXi
API	Nativa, Amazon,EC2/S3/EBS(parcial), OGF OCCi, vCloud (não)	Nativa, Amazon,EC2/S3/EBS, OGF OCCi (não encontrado), vCloud (não)	Padrão evaa2tools, gerenciamento de interface via web	XML-RPC, libvirt, EC2 (consulta) e Cloud OpenNebula API para acesso público	XenAPI for XenServer/XCP, libvirt for KVM	Padrão EC2	Padrão Microsoft, armazenamento em modo NTFS	VMware vCloud Director e do API REST
Drivers	LDAP/AD	LDAP/AD	LDAP/AD	LDAP/AD	LDAP/AD	LDAP	LDAP	LDAP
Autenticação								
Algoritmo de escalonamento	Sim, nova-scheduler	Sim	Sim	Sim	sim	Sim	Sim	Sim
Movimentação instâncias tempo real	Sim	Sim	Não localizado	Não localizado	sim	Sim	Sim	Sim
Configuração de rede para instâncias	SDN, vLANS, Simples (FlatManager e FlatDHCPManager), IPs Elásticos (Floating IPs) e Grupos de segurança	SDN, vLANS, Simples (FlatManager e FlatDHCPManager), IPs Elásticos (Floating IPs) e Grupos de segurança	Estático, DHCP e VLAN	Rede pode ser definida com suporte para Etable. Configuração dos nodes via IP	Estático, DHCP e VLAN	Fornecido o Elastic IP fornecido e após isso é associado pelo Amazon Route 53 (gerencia entradas de	Windows Azure Connect (WAC), com utilização da metodologia conhecida como	Estático, DHCP e VLAN
Sistema Operacional Hospedeiro	GNU/Linux	GNU/Linux	GNU/Linux	GNU/Linux	GNU/Linux	CentOS e o Apache	Windows Server 2008 e 2012/Linux (algumas distribuições)	2008 e 2012/Linux (algumas distribuições)
Autenticação p/ conexão à	Chaves SSH, Credencial X509	Chaves SSH, RSA e Credencial X509	Chaves SSH, Credencial	Chaves SSH, RSA e Credencial X509	Chaves SSH e SSL	Chaves SSH e SSL	Chaves SSL e TLS	Chaves SSL e TLS

Fonte: Elaborado pelo autor.

3 Auditoria e Monitoramento em Computação em Nuvem

Neste capítulo são apresentados os principais conceitos sobre auditoria e monitoramento em nuvem. A auditoria é utilizada para rastrear a execução de rotinas administrativas em instâncias de máquinas virtuais, utilização de dados intermediários e procedimentos aplicados. Essa informação é essencial para o entendimento, descoberta, validação de dados e processos. O monitoramento deve ser efetuado em intervalos de tempos contínuos ou discretos, realizando a medição dos eventos consistentes e inconsistentes das instâncias de máquinas virtuais disponibilizadas no ambiente de nuvem.

3.1 Introdução à Auditoria e Monitoramento em Nuvem

Ferramentas de auditoria e monitoramento devem oferecer mecanismos adicionais, que mantenham o ambiente de computação em nuvem confiável, realizem a verificação das informações com segurança, permitam eliminação de inconsistências, e promovam a disponibilidade dos recursos a seus usuários. Para realizar esta análise faz-se necessário o entendimento de conceitos, como auditoria e monitoramento em nuvem. (ISMAIL et. al, 2015)

Auditoria em computação em nuvem refere-se a garantir que a nuvem atenda às expectativas desejadas, com o alcance dos objetivos organizacionais de forma eficaz e consiga realizar de forma eficiente o monitoramento do consumo de recursos através da coleta de informações e avaliação das evidências. (WAQAS et. al, 2013)

O monitoramento tem como objetivo garantir que um conjunto de sistemas, processos e procedimentos sejam implementados na plataforma de computação em nuvem. O objetivo é proporcionar uma resposta imediata quanto a questões de alterações injustificadas ou a eventos relacionados ao ambiente de nuvem. (ISMAIL et. al, 2015)

3.2 Técnicas e Método para Auditoria e Monitoramento em Nuvem

Há muitos aspectos para realização de auditoria e monitoramento em nuvem, como infraestrutura, integridade de dados, segurança, desempenho e privacidade. A seguir são apresentados algumas técnicas e métodos utilizados nesse trabalho. (MEERA e GEETHAKUMARI, 2015)

- **Técnica da introspecção de máquina virtual:** A virtualização é a espinha dorsal da computação em nuvem, e portanto é crucial oferecer técnicas que realizem não somente o monitoramento de forma estática, mas também em tempo real deste ambiente, através da VMI¹⁷ (*Virtual Machine Introspection*), colabora com a tradicional técnica forense, para realizar uma investigação de falha ocorrida em máquinas virtuais em tempo real;
- **Técnica de medição de integridade de máquina virtual:** Tem como objetivo medir o formato de ELF¹⁸ (*Executable and Linkable Format*), executa em um sistema operacional e permite realizar uma terceira etapa, que é verificar se esses ELF são confiáveis. (YAQIANG et. al, 2014);
- **Método de proveniência de dados:** Auxilia no rastreamento da origem dos dados e, por consequência, auxilia no gerenciamento de informações coletadas em ambientes de sistemas distribuídos, e
- **Técnica de auditoria de consistência de dados:** Garante a consistência no armazenamento em nuvem, que envolve a entrega de dados de armazenamento como um serviço que necessita garantir o acesso ubíquo¹⁹, para isso o provedor da nuvem utiliza técnicas de replicação de dados em vários servidores distribuídos geograficamente. (DIVYA e REDDY, 2015)

3.3 Principais Características para uma Ferramenta de Auditoria e Monitoramento

O ambiente de nuvem compreende vários desafios como versatilidade, escalabilidade, interoperabilidade, segurança, diagnóstico e automação. Na maioria dos casos, a segurança é considerada como um parâmetro extra para se estender o desenvolvimento de *software* ou configuração do ambiente e não faz parte essencial do processo.

No entanto, a segurança é indispensável para qualquer ambiente computacional, em virtude de prover qualidade e impedir que ocorram danos ou riscos a fatores essenciais como, por exemplo, controle de acesso, integridade de informações, entre outros. Um dos requisitos

¹⁷ VMI: É o processo utilizado para analisar o estado do sistema dinamicamente, sendo este a máquina virtual (VM), como por exemplo, quais processos estão sendo executados ou quais conexões de rede estão ativas.

¹⁸ ELF: É um padrão para arquivos executáveis, arquivos-objeto e bibliotecas.

¹⁹ Acesso ubíquo: Promove a ideia de que os computadores estarão em todos os lugares e em todos os momentos auxiliando o ser humano sem que ele tenha consciência disso.

para ser aplicada é o monitoramento que deve compreender os mecanismos de análise que possam proporcionar ao administrador do provedor o acompanhamento e controle das informações. (MADHUBALA, 2015)

Para auditar a conformidade de segurança um sistema automatizado deve coletar todas as informações relevantes a partir de diferentes fontes dentro de uma nuvem, converter os dados coletados em um formato consistente e reconstruir qualquer falta de correlação entre estes dados, antes de incluí-los em uma ferramenta para análise. (MAJUMDAR, et. al, 2015)

O ambiente de computação em nuvem enfrenta problemas que precisam ser monitorados e analisados, para prover a confiabilidade e evitar falhas que possam causar a indisponibilidade de algum serviço. A seguir são apresentadas algumas das principais características que os sistemas de monitoramento devem conter. (IRIMIE e PETCU, 2015).

- **Escalabilidade:** O sistema de monitoramento deve ser composto por diferentes módulos, que possam ser replicados em função da carga e o nível de tolerância a falhas, para que seja também executado em outras instâncias que irão receber esta nova carga;
- **Portabilidade:** O sistema de monitoramento deve ser transportado com facilidade entre os provedores de nuvem e plataformas de todos os seus componentes do sistema a serem implementados na máquina virtual;
- **Inquilinidade:** O sistema deve possibilitar que diversos recursos de vários usuários possam ser monitorados ao mesmo tempo. Neste sentido, usuários podem realizar seus trabalhos sem que este monitoramento comprometa seu resultado;
- **Interoperabilidade:** O sistema pode realizar o monitoramento dos recursos de diferentes provedores de nuvem, desde que não permita à exposição da informação destes recursos durante a realização deste processo;
- **Customizabilidade:** Permite que clientes realizem customizações através de alteração de parâmetro, por exemplo, para verificação específica ou outra alteração que ocorra no ambiente;
- **Extensibilidade:** A arquitetura permite a configuração de novas ferramentas de monitoramento, para que possam auxiliar e facilitar o entendimento das informações coletadas a serem analisadas a partir deste ambiente assim, é preciso manter a compatibilidade na realização da execução do processo solicitado, e

- **Acessibilidade:** Todos os componentes, de preferência da ferramenta, devem ser desenvolvidos com utilização de linguagem *opensource*, sendo possível ser adaptado à necessidade específica através de alteração do código fonte.

Vários provedores oferecem soluções próprias para monitoramento dos recursos de nuvem, visando com isso proporcionar maior integridade das informações, diminuir os incidentes e prover a segurança para o ambiente. A seguir são apresentadas algumas dessas ferramentas de monitoramento e auditoria utilizadas em nuvem. (PARK et. al, 2012)

- **CloudWatch:** Ferramenta proprietária da AWS orquestrador *Amazon*, permite a visualização de informações sobre padrões de acesso, a utilização de recursos e métricas de desempenho. Possibilita coleta de dados como tráfego de rede, uso da CPU e utilização do disco. O acesso é realizado via linha de comando ou *API* de serviço *web*;
- **AzureWatch:** Ferramenta proprietária para os clientes do orquestrador *Windows Azure*, monitora a integridade e a disponibilidade, possibilita criar alertas para receber notificações quando a disponibilidade do serviço for prejudicada e personalizar os gráficos de métricas;
- **Rackspace Cloud Monitoring:** Ferramenta proprietária que substituiu o *Cloudkick*, monitora serviços de nuvem da *Rackspace* e serviços de outros fornecedores. Possibilita a criação de alertas, utiliza gráficos gerados a partir da coleta de métricas do ambiente para identificar tendências e padrões no consumo de serviços, (RACKSPACE,2013) e
- **CloudClimate:** Ferramenta de monitoramento gratuita, possível de ser utilizada pelos orquestradores da *AWS*, *GoGrid CloudServers* e *NewServers*. Os tipos de dados são métricas relacionadas ao desempenho da CPU, uso de memória e acesso a disco, que são coletados com a ferramenta *PRTG Network Monitor* (software de monitoramento de rede). Armazena estas informações em um único local para que o usuário realize comparação do desempenho destes fornecedores, que são exibidos em forma de gráficos. (CLOUDCLIMATE, 2013)

3.4 Normas para Auditoria e Monitoramento em Nuvem

As normas sugerem procedimentos, métodos e ferramentas que devem ser utilizadas no ambiente de computação em nuvem, para manter a segurança, confiabilidade e integridade de

um recurso virtual, para manter a disponibilidade de seus serviços oferecidos aos usuários da nuvem e como realizar a prevenção de possíveis indisponibilidades ou falhas deste serviço a ser disponibilizado.

3.4.1 ISO/IEC 17788

Apresenta uma visão geral e vocabulário da computação em nuvem, para aspectos importantes de comportamentos ou recursos, que precisam ser coordenados na implementação dos papéis de um sistema de computação em nuvem consistente e seguro. Os comportamentos e recursos chaves para a auditoria e monitoramento são:

- **Auditabilidade:** Capacidade de coletar informações para realizar análise dos eventos que possam demonstrar às operações e o funcionamento da utilização de um serviço de nuvem, com a finalidade de realização de uma auditoria, e
- **Segurança física e segurança de aplicativos:** Incluem requisitos, como autenticação, autorização, disponibilidade, confidencialidade, gerenciamento de identidade, integridade, auditoria, monitoramento, resposta a incidentes e gerenciamento de política de segurança. (ISO, 2014)

3.4.2 ISO/IEC 17789 sessão 8.5.12.1

Para atividades de gerenciamento para o ambiente de computação em nuvem não são apenas aspectos condicionados aos prestadores de serviços em nuvem, mas também afetam aos usuários, parceiros de serviços em nuvem e suas sub-funções.

A segurança dos recursos no ambiente de computação em nuvem inclui atividades de gestão e administração, que são utilizados para o controle do uso e prestação de serviços e recursos subjacentes. Como forma de prevenir problemas na segurança a estes recursos e serviços, são apresentados alguns requisitos que devem ser implementados nas atividades de gestão e administração, conforme a seguir: (ISO, 2014)

- Ferramentas que permitem realizar detecção precoce e diagnóstico, verificar a estabilidade do serviço de nuvem e reportar os problemas relacionados com os recursos;
- Segurança na autenticação das informações, relatórios de atividades, monitoramento de sessão e inspeções de pacotes na rede, e

- Implementação de *firewall*, detecção de ataques maliciosos e prevenção para os sistemas dos prestadores de serviços em nuvem.

3.4.3 NIST SP 500-292

Para o gerenciamento dos serviços em nuvem, pois inclui todas as funções que estão relacionadas a seus serviços, visando atender à necessidade na gestão e operação dos serviços exigidos ou propostos pelos usuários. Pode ser descrita a partir da perspectiva do suporte ao negócio, provisionamento e configuração, e do ponto de vista das exigências de portabilidade e interoperabilidade, como apresentado a seguir: (NIST, 2011)

Suporte ao Negócio

- Gestão de clientes: Gerenciar contas de usuários, perfis, relacionamentos com clientes, fornecer um ponto de contato e resolver os problemas dos clientes;
- Gestão de contratos: Administração dos contratos de serviços para instalação, negociação e encerramento;
- Gerenciamento de inventário: Configurar e gerenciar catálogos de serviços;
- Contabilidade e Faturamento: Gerenciar informações de faturamento do usuário, enviar declarações de faturamento, processo de recebimento de pagamentos e detalhamento de faturas, e
- Informação e auditoria: Monitorar as operações do usuário e emissão de relatórios.

Provisionamento e configuração.

- Rápido provisionamento: Implementação automática de sistemas de nuvem com base nos serviços, recursos e capacidades que são requisitados;
- Alterações dos Recursos: Ajustar a atribuição de configuração dos recursos visando sua manutenção, melhoria e também ao ser adicionado novos nós na nuvem;
- Monitoramento e Relatório: Auditar e monitorar os recursos virtuais, monitoramento de operações e eventos na nuvem e geração de relatórios de desempenho;

- Medição: Apresentar a medição da capacidade de algum nível de abstração apropriado para o tipo de serviço (por exemplo, armazenamento, processamento, largura de banda e contas de usuários ativos), e
- Gestão de SLA (acordo de nível de serviço): Engloba a definição de contrato de SLA (esquema básico como os parâmetros de QoS (qualidade de serviço)), monitoramento de SLA e execução SLA de acordo com as políticas definidas.

Exigências de portabilidade e interoperabilidade

- Portabilidade dos dados: É a capacidade dos usuários na nuvem de copiar objetos de dados para dentro ou fora de uma nuvem ou usar um disco para a transferência de dados em massa, e
- Interoperabilidade: é o serviço que tem a capacidade para seus usuários da nuvem em utilizar seus dados e serviços entre vários provedores de nuvem com uma interface de gestão unificada.

Dentre as atividades de gerenciamento no ambiente de nuvem, o monitoramento exerce um papel fundamental no acompanhamento do sistema. É fundamental para detectar falhas de funcionamento e também serve como base para atividade de planejamento e melhorias.

Por isso, acompanhar essas mudanças e desenvolver ou estender modelos e técnicas de monitoramento é uma atividade constante, desempenhando um papel importante não apenas técnico e gerencial, mas também de agente propagador de novas técnicas e impulsionador de novas tecnologias.

3.5 Trabalhos Relacionados

Nesta sessão são apresentados alguns trabalhos relacionados que demonstram a importância em relação a segurança, a confiabilidade e integridade de dados para se evitar a indisponibilidade dos serviços no ambiente de nuvem, através da auditoria e monitoramento em nuvem.

O trabalho desenvolvido por Xu et al. 2014 realiza coleta de informações dos *logs* de instâncias criadas na nuvem, proveniente de um processo de atualização sem interrupção. Com a utilização das ferramentas Redis, Logstash, Elasticsearch e Kibana, estes *logs*

distribuídos podem ser armazenados em um único *log* central. Neste *log* são aplicadas expressões regulares, algoritmos e *API's*. Tais aplicações permitem a detecção de 91,95% dos erros nas informações analisadas e um aumento na precisão de diagnóstico de causa raiz de 97,13%.

O trabalho de XU et al., 2015 realiza o algoritmo de aprendizagem estatística *SVM* (*Support Vector Machine*), sendo utilizado um interfaceamento por meio de *API's* na ferramenta de monitoramento *CloudWatch* da *AWS* que fornece uma série de métricas da nuvem como (*I/O*, *CPU*, redes etc) para serem analisados por este algoritmo. Com isso, determina quais os eventos que devem ser desativados, proporcionando aos administradores que não recebam alarmes de falsos positivos e falsos negativos ou até mesmo uma inundação de alarmes de diferentes canais sobre o mesmo evento. Com este modelo implementado obteve-se taxa de precisão de 0,421 (84,7%).

Os autores XIAOJIANG e YANLEI, 2013 apresentam a importância do uso de um *framework* para monitoramento de recursos, assim como o que é utilizado pelo orquestrador *OpenStack opensource*, que monitora o desempenho de métricas na utilização de *CPU*, uso de memória e rede de *I/O* de recursos físicos e virtuais. Foi desenvolvido um subsistema de monitoramento de recursos, que armazena informações coletadas em um banco de dados de arquitetura *NoSQL*, para prover otimização na gestão das informações. Os autores enfatizam que nos testes realizados, por estas informações estarem armazenadas em uma base de dados *NoSql*, não houve falhas na integração e armazenamento, funcionou sem problemas, manteve a integridade dos dados, além de ser mais elástico.

LIN et al., 2015 apresenta em seu trabalho que o acesso a todos os recursos e serviços oferecidos na nuvem são através de autenticação. Foi implementado um *framework* que monitora este acesso que se inicia através do *login*, com interfaceamento de *API's* com as informações do banco de dados na nuvem. Através desta coleta de informações pelo *login* no ambiente é realizada a identificação do nível de usuário que está realizando este acesso. Com isso, faz-se uma separação destas informações para serem armazenadas em *logs* diferentes, para impedir que fiquem juntas no mesmo *log* as informações de administradores e usuários, sendo que a criação destes *logs* na base de dados obedece a critérios de privilégios e determina quais recursos e serviços são acessados.

O trabalho de SALEH et al., 2013 utiliza o conceito de CEP (*Complex Event Processing*) de como detectar um evento ou determinar o padrão de evento que ocorreu ou

não, coordena o tempo de ação de resposta e descobre padrões complexos entre múltiplos fluxos de dados do evento. Esse conceito é utilizado principalmente para o BPM (*Business Process Management*) e áreas afins. Através do CEP foi desenvolvido um *framework* que realiza o monitoramento automatizado das métricas coletadas como CPU, memória e disco, a partir da utilização de sensores que possibilitam gerenciar e ajustar os recursos em tempo real, para atender as SLAs (Acordo de Nível de Serviço) do provedor enquanto estes recursos ainda não estão provisionados.

3.6 Considerações Finais

Este capítulo apresentou conceitos importantes sobre a auditoria e o monitoramento de serviços e recursos em nuvem, evidenciando a importância na implementação de mecanismos e ferramentas de monitoramento do ambiente de nuvem, como é demonstrado por algumas normas da Sessão 3.4. Foi evidenciado o surgimento de falhas capazes de impedir a continuidade e disponibilidade dos serviços que são oferecidos aos usuários da nuvem.

Um ponto relevante apresentado na Sessão 3.3 é o tipo de dado que as ferramentas de monitoramento captam, sendo a maioria apenas métricas de desempenho.

No capítulo seguinte são apresentados os materiais e métodos utilizados neste trabalho e são listados os métodos utilizados para coleta de dados e monitoramento.

4 Material e Métodos

Neste capítulo são apresentados o material utilizado nos experimentos e a metodologia adotada para realizar o monitoramento e auditoria entre os dados persistentes do orquestrador com as informações de estado atual do ambiente de nuvem.

4.1 Material

Para o desenvolvimento do trabalho foi utilizada a configuração de equipamentos de rede que é apresentada na sessão *hardware*. A configuração de programas e ferramentas para o desenvolvimento é tratada na Sessão *software*.

4.2 Hardware

Para realização dos testes no ambiente controlado de pesquisa, foram utilizados cinco computadores da marca Lenovo ThinkCentre M58 3.0GHz 4GB 500GB, sendo um servidor de gerenciamento do orquestrador *management server* com plataforma linux e o orquestrado *Apache Cloudstack* versão 4.4.2, dois computadores com o *hypervisor* que realizam a criação das máquinas virtuais, e dois computadores para armazenamento de dados:

- *Management Server*: Plataforma Linux CentOS 6.6 64 bits, orquestrador *Apache Cloudstack* versão 4.4.2;
- *Hypervisor software XenServer* 6.2, para prover os recursos de CPU, memória, armazenamento e de rede necessários para hospedar as máquinas virtuais;
- Armazenamento de dados primário com uso do *software Openfiler* 2.99 Linux CentOS, sendo utilizado para o armazenamento dos volumes de discos para todas as máquinas virtuais sendo executadas em *hosts* na nuvem, e
- Armazenamento de dados secundário com a utilização do *Openfiler* 2.99 Linux CentOS, com o objetivo de armazenar dados como *templates*, imagens ISO e *snapshots* de volumes de disco.

O Quadro 2 apresenta a configuração de rede para o orquestrador *Apache Cloudstack*.

Quadro 2 - Configuração dos computadores

Computador	Função	Rede Privada
CSGL01	Orquestrador (<i>Management server</i>)	192.168.30.1
CSGL02	<i>Hypervisor</i>	192.168.30.2
CSGL03	<i>Hypervisor</i>	192.168.30.3
CSGL04	Armazenamento de dados primário	192.168.30.4
CSGL05	Armazenamento de dados secundário	192.168.30.5
CSGL06	<i>XenCenter</i>	192.168.30.6

Fonte: Elaborado pelo autor.

4.3 Software

Para o ambiente de desenvolvimento foram utilizados os seguintes *softwares* na implementação e execução dos experimentos:

- *Apache CloudStack Cloudmonkey 5.3.2*: Ferramenta utilizada para realizar a extração das informações da nuvem; (APACHE,2015)
- *MySQL Workbench 6.3 CE*: Ferramenta utilizada na conexão com o SGBD MySQL do orquestrador, que auxiliou no desenvolvimento de consultas em linguagem SQL. Foi possível realizar a comparação e validação das informações armazenadas na nuvem (dados persistentes), com o resultado apresentado após a execução da ferramenta AMFC, e (MYSQL, 2015)
- *IDLE Python 2.7*: Através deste *software* foi realizado o desenvolvimento da ferramenta AMFC, com uso da linguagem *python* possibilitando assim a sincronização dos dados persistentes do SGBD *Mysql*, com as *API's* do ambiente de nuvem do orquestrador *Apache CloudStack*. (PYTHON, 2015).

4.4 Métodos

Os métodos para auditoria e monitoramento do ambiente em computação em nuvem tiveram como base três principais trabalhos que proporcionaram contribuições e demonstraram possibilidades de realizar melhoria neste processo. As normas NIST SP 500-292, ISO/IEC 17788 e ISO/IEC 17789 também fundamentam a importância de se auditar e monitorar as atividades operacionais e eventos dos recursos virtuais da nuvem, provendo um importante papel em se detectar falhas de funcionamento deste ambiente para prevenir a

indisponibilidade do serviço que é oferecido a seus usuários, conforme demonstrado na Sessão 3.4.

Foram realizados testes com base em XU et al., 2014 que demonstraram o tipo de rotina que pode ser adotada na geração de informações a serem analisadas e coletadas pela ferramenta AMFC. Como melhoria, o resultado da coleta dessa informação é apresentado em tempo real entre ambiente atual e dados persistentes.

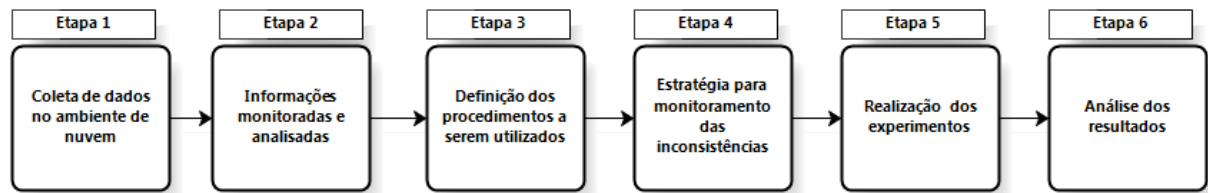
O trabalho de XU et al., 2015 contribuiu em evidenciar a importância da eliminação de eventos inconsistentes no ambiente de nuvem, provocando os falsos positivos e falsos negativos. Quanto à melhoria, foi possível realizar uma integração por meio de *API's* com a base de dados persistentes, para eliminação dos eventos inconsistentes.

SALEH et al., 2013 contribuiu para demonstrar como é o processo na detecção de um evento ou na determinação do padrão deste evento na análise a um fluxo de dados em um ambiente de nuvem. Como melhoria, a ferramenta AMFC não ficou restrita à análise de algumas métricas, e sim, faz a auditoria e monitoramento dos recursos providos por este ambiente de nuvem.

Dentro deste contexto, foi desenvolvida uma ferramenta que realize à auditoria e monitoramento de dados inconsistentes na nuvem de forma contínua através da sincronização das informações do ambiente de teste da nuvem com as informações de dados persistentes (base de dados).

Com base nas normas apresentadas na Sessão 3.4 foram definidas as regras a serem seguidas no intuito de atender as sugestões para realizar a auditoria e monitoramento em sistemas de nuvem de forma contínua e proporcionar a segurança, integridade, confiabilidade para a gestão do negócio no ambiente de computação em nuvem. A Figura 12 apresenta as etapas da metodologia utilizada para a realização deste trabalho.

Figura 12 - Diagrama da Metodologia para Auditoria e Monitoramento



Fonte: Elaborado pelo autor.

Etapa 1 - Coleta de dados no ambiente de nuvem: A coleta dos dados para análise e monitoramento é proveniente da execução de oito diferentes tipos de rotinas de manutenção para uma instância de máquina virtual, realizadas através da console de administração do orquestrador. A partir dessas rotinas, há uma geração de informações que serão armazenadas em base de dados persistentes e posteriormente serão monitoradas e analisadas.

Etapa 2 - Informações monitoradas e analisadas: Foi possível realizar o monitoramento e análise, a fim de que se verificar os dados inconsistentes que podem ocasionar os falsos negativos (evento²⁰ que não gera um alerta ao administrador, sendo que deveria) e falsos positivos (evento que gera um alerta ao administrador, sendo que não deveria) no ambiente de computação em nuvem; a validação da sincronização dos dados persistentes armazenados no SGBD *Mysql*, localizadas no servidor de gerenciamento do orquestrador, com as informações do ambiente de teste da nuvem. São apresentadas as informações inconsistentes e também as informação não utilizadas pelo ambiente de teste, para em seguida serem eliminadas na base de dados persistentes;

Etapa 3 - Definição dos procedimentos a serem utilizados: Os procedimentos foram definidos a partir de oito diferentes tipos de rotinas de manutenção para uma instância de máquina virtual. A definição dessas rotinas foi motivada pela pesquisa através de outros orquestradores que as realizam por meio de duas formas, a planejada, que seria na criação de uma nova instância, ou a não planejada, quando ocorre uma alteração na configuração da instância ou ocorre uma atualização. (AZURE, 2014) e (AMAZON, 2014) Posterior à execução destas rotinas, o ambiente de nuvem gerou informações, podendo estas serem consistentes ou inconsistentes, para serem auditadas e monitoradas. e

²⁰ Evento: Uma ocorrência que acontece de forma completa, ou não acontece, instantânea, em um ponto específico do tempo e que tem significado para o seu local de interesse. Representa também o resultado de uma ação. A ocorrência de um evento pode provocar uma reação, que seria uma ação (ou conjunto de ações) a ser tomada.

Etapa 4 - Estratégia para monitoramento das inconsistências: A definição de inconsistência se caracteriza neste trabalho pela informação divergente apresentada entre os dados coletados, sendo a estratégia realizada através da auditoria e monitoramento por meio de sincronização das informações do ambiente de nuvem teste, com as informações persistentes (banco de dados).

Para identificação desta divergência que leva à inconsistência de dados, foram definidas duas regras para serem verificadas:

- Redundância não controlada: A partir dos estudos de conceitos de banco de dados, demonstra que o gerenciamento da informação deve ser realizado através da ferramenta AMFC e não pelo usuário, conforme descrito na Sessão 5.3, (ELMASRI e NAVATHE, 2005) e
- Violação de integridade na nuvem: Com os estudos realizados no ambiente de nuvem foi verificado que o ambiente de teste e dados persistentes não devem apresentar violação quanto à integridade de suas informações. Para o caso em que a manutenção de uma instância de máquina virtual é realizada pelo administrador, como por exemplo, a eliminação de volumes de dados de uma máquina virtual, ainda está sendo contabilizada como um recurso em uso pelo ambiente de teste, conforme descrito na Sessão 5.4.

A interface de administração do orquestrador *Apache CloudStack* foi utilizada para realizar as rotinas de manutenção de uma instância de máquina virtual.

Etapa 5 - Realização dos experimentos: Nesta etapa pôde-se demonstrar os experimentos realizados e a análise dos resultados do ambiente controlado e privado de computação em nuvem, para se atestar a importância de se realizar o desenvolvimento e implementação de uma ferramenta de auditoria e monitoramento contínuo para o ambiente de nuvem, descrita com maiores detalhes na Sessão 5.

Etapa 6 - Análise dos resultados: Descrito na Sessão 5.5 que apresenta avaliação dos resultados obtidos por meio da quantidade de inconsistências encontradas e o ganho de armazenamento de dados, sendo provenientes da eliminação destas inconsistências e das informações que não são mais utilizadas pelo ambiente de teste.

Definição dos Parâmetros de Cálculo

Foram definidos parâmetros para realização de cálculo de inconsistências por oferta de recurso e proporcionalidade de ganho em armazenamento de dados persistentes, para se obter o planejamento da coleta de dados, sumarizar e interpretar os dados. A fim de realizar avaliação dos resultados os parâmetros são: (ACHCAR e RODRIGUES, 1998)

- Parâmetros de cálculo de inconsistências por oferta de recurso.
 - atn (ambiente de teste da nuvem): Refere-se à quantidade inicial de recursos virtuais na finalização da configuração do ambiente de teste, como por exemplo, usuários, roteadores e máquina *proxy*. Neste sentido, não há nenhum recurso virtual criado manualmente (administrador), apenas recursos do ambiente que não são contabilizados pela análise e monitoramento da ferramenta;
 - qdp (quantidade dados persistentes): Apresenta a quantidade de registros armazenados na base de dados persistente (consistente e inconsistente), e
 - tir (total de inconsistência por recurso): Após definidos os parâmetros de qdp e atn é realizada a operação de subtração entre estes valores e, com isso, é determinado o total de inconsistências encontradas no ambiente de teste da nuvem. A equação utilizada é:

$$\text{qdp} - \text{atn} = \text{tir} \quad (1)$$

- reor (rotinas executadas por oferta de recurso): No Quadro 4 são descritos na Sessão 6.1 os tipos de oferta de recursos utilizados juntamente com as rotinas de manutenção para uma instância de máquina virtual. Também na Sessão 6.1 no Quadro 5 é descrito um modelo de teste realizado para cada tipo de oferta criada;

Com isso, é realizado o cálculo de inconsistências por oferta de recurso, em que se define o total da quantidade de inconsistências mediante o aumento da quantidade do número de cada uma das rotinas. Definida pela equação:

$$\frac{\text{tir}}{\text{reor}} \quad (2)$$

- Parâmetros de cálculo para proporcionalidade de ganho em armazenamento de dados persistentes;
 - tidp (tamanho de instância de dados persistentes): É apresentado o valor total da instância que contém os dados persistentes da nuvem, havendo nesta instância dados consistentes e inconsistentes, (MYSQL, 2010) e
 - gar (ganho de armazenamento por rotina) Valor apurado a partir do cálculo realizado na operação de subtração entre os valores de tidp e tir . A equação apresentada é:

$$\text{tidp} - \text{tir} = \text{gar} \quad (3)$$

- ord (oferta de recurso em disco): Conforme apresentado no Quadro 4 na coluna disco (GB).

Através destes parâmetros, foi possível determinar a proporcionalidade de ganho por oferta de recurso. Definida pela equação a seguir:

$$\left(\frac{\mu_{\text{gar}}}{\text{ord}}\right) * 100 \quad (4)$$

4.5 Considerações Finais

Este capítulo apresentou os materiais e os métodos adotados no trabalho, a metodologia, descrita em cinco etapas de monitoramento e análise do ambiente em nuvem, a definição dos procedimentos a serem utilizados, a estratégia para monitoramento das inconsistências, a realização dos experimentos e a análise dos resultados.

Foi ressaltado quanto à necessidade e importância em se auditar e monitorar os recursos que são providos no ambiente de computação em nuvem, por meio de normas descritas na Sessão 3.4 NIST SP 500-292, ISO/IEC 17788 e ISO/IEC 17789, que visam apresentar sugestões e regras para prevenir a ocorrência de falhas na disponibilidade dos serviços na nuvem. O estudo de trabalhos de outros autores contribuiu para o entendimento de como se detectar e realizar a prevenção e posteriormente a eliminação apenas dos eventos

inconsistentes, que possam causar uma falha do serviço oferecido ou provocar inundação de falsos positivos e falsos negativos que induzem o administrador a cometer erro em uma tomada de decisão.

Com isso foi possível realizar o desenvolvimento de uma ferramenta *opensource* que procurasse atender às necessidades apresentadas, oferecendo melhorias de coleta de informação em tempo real entre ambiente de teste e armazenamento dos dados persistentes e não ficar restrita apenas à análise de algumas métricas já definidas.

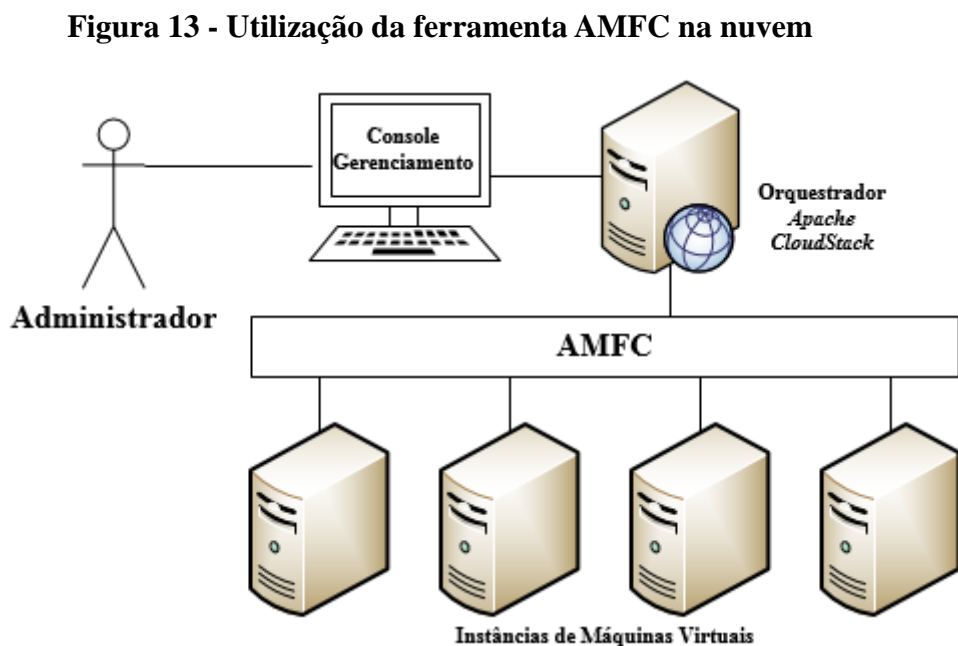
Por fim, foi apresentada a definição de parâmetros para realização de cálculo de inconsistências por oferta de recurso e proporcionalidade de ganho em armazenamento de dados persistentes, para assim ser contabilizado.

5 Ferramenta de auditoria e monitoramento em nuvem AMFC

Neste capítulo a ferramenta proposta, nomeada de AMFC - Ferramenta de Auditoria e Monitoramento para Eliminar Inconsistências no Apache CloudStack, é descrita desde a sua concepção à sua implementação. Para isso, foram criados alguns fluxogramas para facilitar o seu entendimento, assim como o comportamento em sua execução neste ambiente de nuvem.

5.1 Arquitetura da ferramenta AMFC

A Figura 13 apresenta a arquitetura da ferramenta quanto à interação do usuário administrador no ambiente de nuvem, que é responsável por executar as oito rotinas de manutenção para uma instância de máquina virtual através da console de gerenciamento do orquestrador Apache CloudStack, sendo essas rotinas definidas na Sessão 4.4 Métodos no item Etapa 2 - Definição dos procedimentos a serem utilizados e sua configuração no ambiente de nuvem. A seguir, são descritas cada uma dessas oito rotinas:



Fonte: Elaborado pelo autor

- Criar disco de volume de dados: Na criação de uma instância de máquina virtual o administrador tem a possibilidade de realizar a configuração do tamanho da oferta de disco para armazenamento de dados desta máquina virtual;
- Anexar disco de volume de dados: Posterior à criação de uma instância de máquina virtual, o administrador tem a possibilidade de realizar a configuração de um espaço em disco adicional para armazenamento de dados dos usuários, os quais se utilizarão desta máquina assim que ela estiver finalizada;
- Migrar instância para outro host: O administrador pode realizar a migração de uma instância de máquina virtual entre diferentes hosts do ambiente de nuvem;
- Reiniciar instância: Nesta rotina o administrador reinicia a instância de máquina virtual já criada, no caso de haver algum problema de disponibilizar um serviço ou para realizar uma manutenção quanto a sua configuração;
- Parar execução de instância: A instância de máquina virtual está funcionando normalmente, o administrador executa a parada desta máquina, no entanto ainda consta no ambiente de nuvem do orquestrador;
- Apagar instância: Para esta rotina a instância de máquina virtual é apagada do ambiente de nuvem;
- Recuperar instância: Após uma instância de máquina virtual ser apagada, é possível realizar sua recuperação, juntamente com toda a sua configuração que estava em uso.
- Iniciar instância: Esta rotina é executada quando a instância de máquina virtual está parada.

A ferramenta AMFC é configurada no orquestrador Apache CloudStack como descrito na Sessão 4.2 Hardware no item *Management Server*. Há possibilidade também de configurá-la através de agendador de tarefas no sistema operacional Linux como, por exemplo, a utilização do serviço *crontab*²¹, podendo-se inicializar mediante definição de horários ou de

²¹ Crontab: Um serviço da plataforma Linux que é carregado durante o processo de *boot*, permite agendar a execução de comandos e processos de maneira repetitiva ou apenas uma única vez.

forma contínua, realizando a coleta das informações geradas a partir das rotinas de manutenção executadas em cada instância de máquina virtual.

5.2 Implantação na nuvem da ferramenta AMFC

A implantação da ferramenta AMFC é realizada pelo provedor do ambiente da nuvem localizada na máquina *Management Server* de plataforma Linux sendo esta o orquestrador Apache CloudStack. A Figura 14, apresenta a camada de IaaS, sendo este o local que é realizada a coleta de dados pela ferramenta AMFC, modelo que abriga o conjunto de componentes virtualizados (por exemplo, instância de máquina virtual ou de armazenamento em rede) disponíveis para os consumidores como um serviço terceirizado, utilizado para criar e executar aplicativos sem a compra de componentes físicos de um computador.

Figura 14 - Camada de Nível de Serviço de configuração da ferramenta AMFC

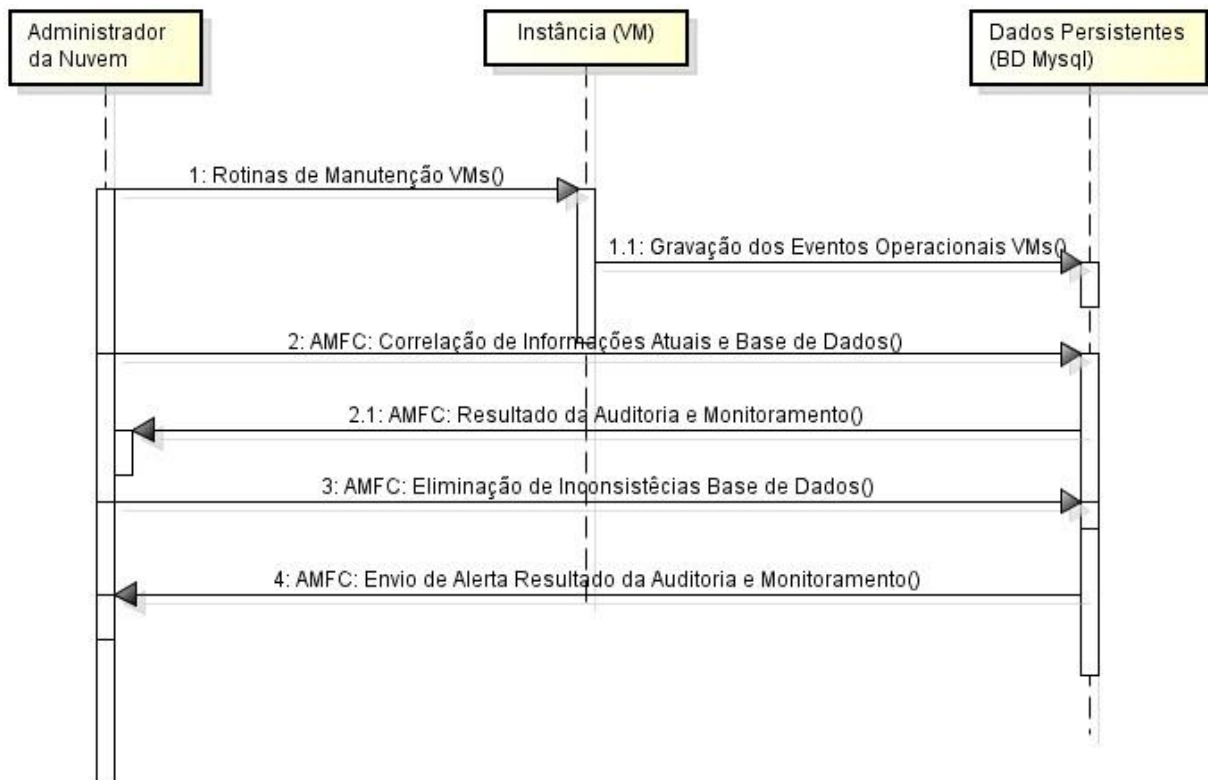


Fonte: Elaborado pelo autor com base em (BUYYYA et al. 2013)

5.2 Ambiente da nuvem auditado e monitorado pela ferramenta AMFC

Na Figura 15 são apresentadas as etapas de auditoria e monitoramento, iniciando-se pelo administrador da nuvem até o envio das inconsistências encontradas neste ambiente.

Figura 15 - Etapas do Monitoramento da Nuvem pela Ferramenta AMFC



Fonte: Elaborado pelo autor

- 1: Rotinas de Manutenção VMs: São realizadas as rotinas de manutenção de recurso máquina virtual pelo administrador do ambiente de nuvem, descritas na Sessão 5.1;
 - 1.1: Gravação dos Eventos Operacionais VMs: Posteriormente à Etapa 1 o orquestrador gera as informações dos eventos operacionais da máquina virtual e os grava na base de dados persistente;
- 2: AMFC: Correlação de Informações Atuais e Base de Dados: Nessa Etapa a ferramenta AMFC realiza a correlação das informações coletadas do ambiente de teste com as informações de dados persistentes;

- 2.1 AMFC: Resultado da Auditoria e Monitoramento: Após conclusão da Etapa 2, é apresentado no terminal da plataforma, para o administrador do ambiente de nuvem, a quantidade de registros inconsistentes encontrados;
- 3 AMFC: Eliminação de Inconsistências Base de Dados: Essa etapa apresenta os registros no terminal da plataforma, devem ser eliminadas as inconsistências conforme resultado apresentado na Etapa 2.1, já definidos em linguagem *SQL-ANSI*²² para sua execução, e
- 4 AMFC: Envio de Alerta Resultado da Auditoria e Monitoramento: Essa etapa realiza o envio de um e-mail para o administrador do ambiente de nuvem, apresentando estatística de quantos registros foram eliminados e como está o ambiente de teste comparado com os dados persistentes.

O Quadro 3 apresenta um trecho do pseudocódigo da ferramenta AMFC relacionado ao módulo de monitoramento de redundância não controlada no ambiente de nuvem. Para maior entendimento, foi elaborado seu fluxograma conforme Figura 16. A seguir são explicadas quais as principais etapas no processo de auditoria e monitoramento deste módulo.

Quadro 3 - Pseudocódigo da ferramenta AMFC

```

Algoritmo "Trechopseudocodigo AMFC"
  db,host,user,passwd,banco:caracter
  api,apichave,chavenuvem,orquestrador:caracter
  VarLista_vm_db,VarUsoApi:caracter
  Varbancob,VarNuvem:inteiro
  //Importar módulos de conexão com o mysql e do orquestrador
  Leia (importa)
  //Chamadas de acesso de apis do orquestrador
  Leia (api)
  Leia (apichave)
  Leia (chavenuvem)
  //Conexão com base de dados do orquestrador
  Leia (host)
  Leia (user)
  Leia (passwd)
  Leia (banco)
  orquestrador <- api + apichave + chavenuvem
  db <- host + user + passwd + banco
  //Modulo de monitoramento de redundância não controlada na nuvem
  VarLista_vm_db = db
  VarUsoApi = orquestrador.ListaVMs
  Varbancob = 0
  VarNuvem = 0

```

²² *SQL/ANSI (Structured Query Language/ American National Standards Institute)*: Linguagem de pesquisa declarativa padrão para gerenciamento de dados que interage com os principais bancos de dados baseados no modelo relacional.

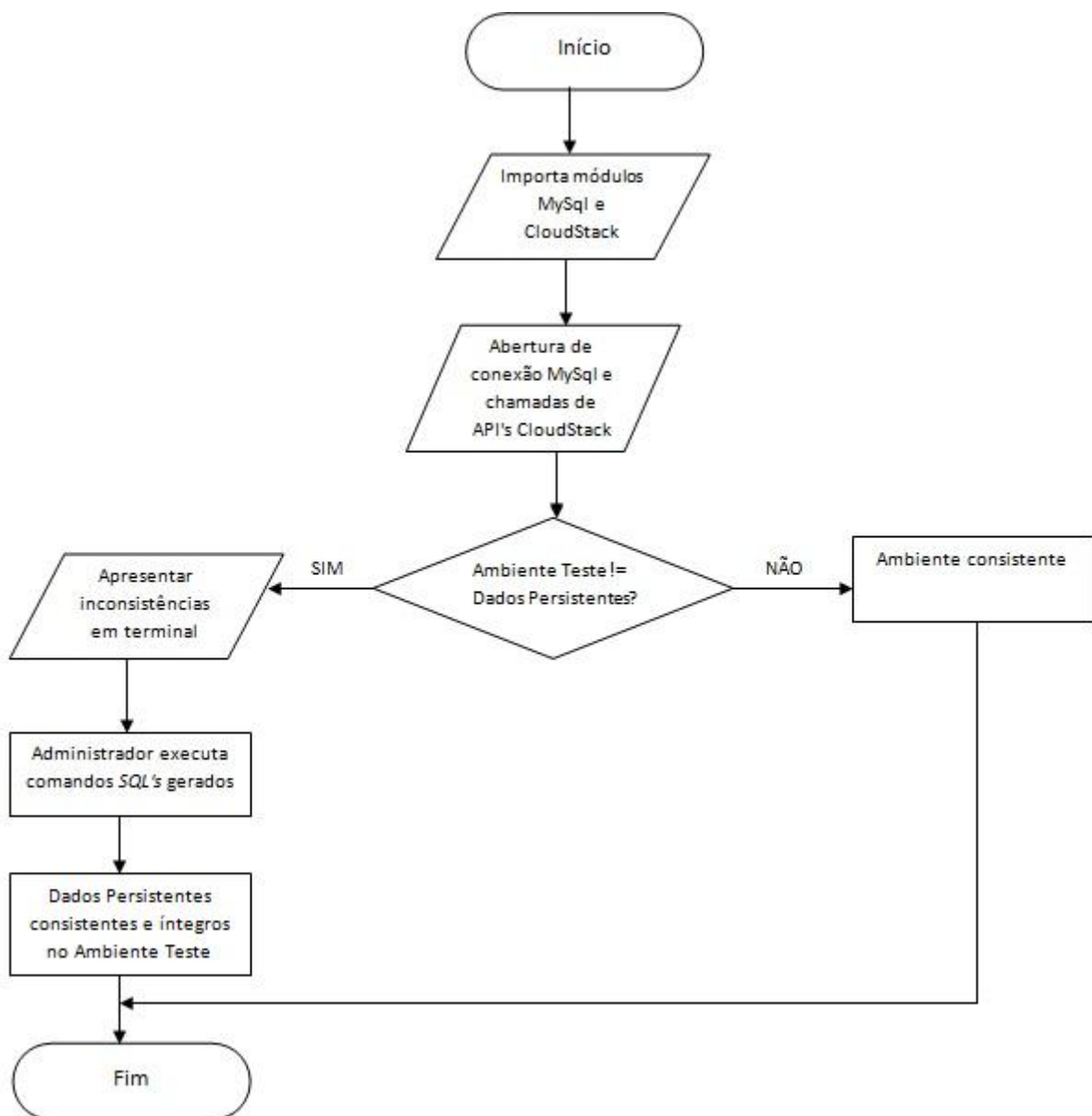
```

para Infor_linha de VarLista_vm_db faca
para VM de VarUsaApi faca
se Infor_linha == VM entao
  Escreva ("Ambiente Consistente" )
Senão
  enquanto VarNuvem < (VarLista_vm_db)
    Escreva ("Geração de Comandos SQL's")
  fimenquanto
fimpara
fimpara
fim
Fimalgoritmo

```

Fonte: Elaborado pelo autor.

Figura 16 - Fluxograma Módulo de Redundância não Controlada



Fonte: Elaborado pelo autor.

- **Entrada:** Inicia-se pela importação dos módulos que contem as funções e definições para utilização do banco de dados Mysql e orquestrador CloudStack. Posteriormente utiliza-se a URL que contém as chamadas de APIs: Ex: `http://localhost:8080/client/api`, no ambiente do orquestrador. Também faz-se necessário realizar a geração da API key e a secret por meio de uma conta de usuário, criada neste orquestrador, com isso é possível ter acesso ao gerenciamento dos recursos e automatização de ações neste ambiente.
- **Procedimento:** Nesta etapa é possível utilizar os comandos de APIs. Neste caso, está sendo monitorada a redundância não controlada, que são provenientes das informações de uma instância de máquina virtual em dados persistentes com o ambiente de teste. São utilizadas duas condições, a primeira realiza uma comparação dos dois ambientes, que se forem iguais não apresenta nenhuma mensagem de inconsistência, enquanto na segunda condição, é verificado se o ambiente de teste está diferente das informações persistentes e se imprime mensagem de inconsistência.
- **Saída:** Quanto à forma de apresentação do resultado desta auditoria e monitoramento da inconsistência encontrada, gera-se em comando *SQL-ANSI* no terminal do administrador, para depois ser executado, e isso faz com que os dados persistentes do orquestrador fiquem consistentes e íntegros.

5.3 Considerações Finais

Este capítulo apresentou a implementação da ferramenta AMFC na camada de nível de serviço IaaS e a explicação das oito rotinas de manutenção para uma instância de máquina virtual, exemplificando alguns dos componentes virtualizados, contidos neste local.

A realização do processo de auditoria e monitoramento em nuvem foi dividida em etapas para melhor entendimento e por fim, o pseudocódigo do módulo de monitoramento de redundância não controlada foi explicado a partir do fluxograma e descrição das principais etapas.

6 Resultados Experimentais

Neste capítulo são apresentados os experimentos realizados e a análise dos resultados do ambiente controlado e privado de Computação em Nuvem, utilizando o orquestrador *Apache CloudStack*. (CLOUDSTACK, 2015)

Para a coleta de informações do ambiente controlado, foram utilizados com base nos métodos descritos na Seção 4.4, e na avaliação dos resultados, foram utilizadas as equações como mostra também na Sessão 4.4 em Definição dos Parâmetros de Cálculo. Posteriormente será apresentado a ferramenta de monitoramento do ambiente, e como foi implementada, os testes desenvolvidos e os resultados finais obtidos.

6.1 Realização dos Testes e Resultados

Foram realizados testes com base nos trabalhos de (XU et al., 2014) e (XU et al., 2015), que destacam a importância de se diagnosticar os eventos inconsistentes que ocorrem na nuvem em menor tempo possível, com isso, realizar a prevenção de falsos positivos e falsos negativos, que podem provocar até a indisponibilidade de um serviço, em virtude da ação que um administrador pode tomar em relação à informação destes eventos.

A configuração padrão de instâncias foi definida conforme apresentado no Quadro 4. Os tamanhos das instâncias de máquinas virtuais foram limitados devido ao ambiente disponível para os testes.

Quadro 4 - Tipos de Oferta de recurso

Recurso	Núcleo	CPU (Mhz)	Memória (MB)	Disco (GB)
Grande	2	2000	2048	100
Médio	1	1000	1024	20
Pequeno	1	500	512	05

Fonte: Elaborado pelo autor.

Os testes desse experimento foram executados em oito diferentes tipos de rotinas de manutenção para uma instância de máquina virtual (a definição dessas rotinas foi descrita na Sessão 4.4).

As rotinas de manutenção foram realizadas através da própria *interface* de administração do orquestrador *Apache CloudStack*, podem ser verificadas na Sessão 5.1 arquitetura da ferramenta AMFC e também descrição explicando o que cada uma realiza.

Para cada tipo de oferta de recurso, foram criadas dez instâncias, sendo que em cada uma, foi realizado um número de repetições (por exemplo, na instância chamada VM3, foram realizadas três vezes o número de cada uma das rotinas citadas, com isso, esta VM3 obteve um total de 24 rotinas executadas ou seja (número de execuções X número de cada rotina)) e assim por diante até a VM10, como demonstrado no Quadro 5. Foram executadas 440 rotinas por recurso.

Quadro 5 - Modelo de testes oferta de recurso

Instâncias	Execuções	Quantidade de rotinas
VM1	1	8
VM2	2	16
VM3	3	24
VM4	4	32
VM5	5	40
VM6	6	48
VM7	7	56
VM8	8	64
VM9	9	72
VM10	10	80

Fonte: Elaborado pelo autor

Totalizando todos os recursos, foram criadas 30 instâncias de máquinas virtuais com 1.320 rotinas administrativas executadas.

Para cada instância que atingia a quantidade de rotinas conforme Quadro 5, realizou-se sua eliminação e, após isso, foi executada a ferramenta AMFC.

6.2 Casos de uso realizados de auditoria e monitoramento AMFC

Para demonstrar como a ferramenta AMFC realiza a auditoria e monitoramento através da sincronização das informações do ambiente de nuvem de teste com as informações persistentes (banco de dados), foram realizadas alterações na ferramenta possibilitando assim a apresentação das informações monitoradas da nuvem após sua execução no ambiente de forma manual.

Nas seções seguintes são apresentados dois cenários de testes. No primeiro cenário foi demonstrado como a ferramenta realiza sincronização das informações e elimina somente aquilo que não está no ambiente da nuvem. Já no segundo cenário, é apresentado um caso de uma inconsistência monitorada deste ambiente. Para ambos os cenários, foram determinadas algumas etapas como segue:

- Ambiente de teste do orquestrador *Apache CloudStack*: Não possui nenhum recurso ou serviço configurado no ambiente de nuvem;
- Criação e configuração de instâncias de máquinas virtuais: Com a utilização da console de gerenciamento do orquestrador, serão realizadas as execuções das rotinas administrativas descritas na Sessão 6.1;
- Sem utilização da ferramenta: De forma manual realiza-se o desenvolvimento da solução para eliminação das informações divergentes do ambiente, e
- Com utilização da ferramenta: Através do resultado apresentado pela ferramenta, é possível realizar uma comparação das informações que deveriam ser eliminadas com a etapa sem utilização da ferramenta (manual).

6.3 Cenário 1: Ambiente de teste e dados persistentes no gerenciamento de informações

A Figura 17 apresenta o ambiente inicial do orquestrador Apache CloudStack, especificamente a console de gerenciamento de instâncias de máquinas virtuais, sendo que não há nenhuma máquina criada ou configurada.

Figura 17 - Gerenciamento de Instância de Máquina Virtual no Orquestrador



Fonte: CLOUDSTACK (2015).

Figura 18 - Informações de Instância de Máquina Virtual em Dados Persistentes

The screenshot shows a MySQL query: `select id, name, uuid, instance_name, state, type from cloud.vm_instance;` The result grid displays the following data:

	id	name	uuid	instance_name	state	type
▶	1	s-1-VM	96c349ed-9a63-4c6d-a257-ff936a1eec23	s-1-VM	Running	SecondaryStorageVm
	2	v-2-VM	f4b54545-2440-4fa0-8c21-3762dc87bbd6	v-2-VM	Running	ConsoleProxy
	4	r-4-VM	11b1ea66-1f75-4af0-9980-4739b5f6274f	r-4-VM	Running	DomainRouter
*	NULL	NULL	NULL	NULL	NULL	NULL

Fonte: MYSQL (2015).

Para a Figura 18 é apresentada a tabela `vm_instance` do SGBD Mysql do orquestrador, que armazenam as informações de instâncias de máquinas virtuais criadas e configuradas conforme demonstrado na Figura 17. Convém observar que as informações apresentadas na Figura 18, referem-se a instâncias pertencentes e de uso da infraestrutura do orquestrador.

Figura 19 - Atividades de Eventos no Orquestrador

The screenshot shows the CloudStack 'Eventos' page. The visualization is set to 'Eventos'. The table displays the following data:

	Descrição	Nível	Type	Domínio	Conta	Data
<input type="checkbox"/>	user has logged in from IP Address 177.99.132.65	INFO	USER.LOGIN	ROOT	admin	16 Jul 2016 21:04:31
<input type="checkbox"/>	user has logged in from IP Address 177.99.132.65	INFO	USER.LOGIN	ROOT	admin	16 Jul 2016 11:51:34

Fonte: CLOUDSTACK (2015).

Figura 20 - Dados Persistentes das Atividades de Eventos

The screenshot shows a MySQL query: `select id, uuid, description, level, type, state from cloud.event;` The result grid displays the following data:

	id	uuid	description	level	type	state
▶	189	810dab92-f07e-4ab8-91ee-918e09c5e597	user has logged in from IP Address 177.99.132.65	INFO	USER.LOGIN	Completed
	190	09f08df1-2713-4533-a8df-2cd188c48aa5	user has logged in from IP Address 177.99.132.65	INFO	USER.LOGIN	Completed
*	NULL	NULL	NULL	NULL	NULL	NULL

Fonte: MYSQL (2015).

A Figura 19 apresenta dados de eventos gerados do ambiente inicial e, como pode-se verificar, são apenas eventos de *login* do usuário admin. Na Figura 20 é apresentada a tabela event do *SGBD Mysql* do orquestrador, contendo as mesmas informações demonstradas na Figura 19 da nuvem.

Cada instância de máquina virtual foi configurada seguindo a oferta de recurso conforme descrita na Sessão 6.1 no Quadro 4. Foram realizadas rotinas administrativas de manutenção descritas na Sessão 5.1 Arquitetura da ferramenta AMFC.

Figura 21 - Gerenciamento de Instâncias de Máquinas Virtuais no Orquestrador

Home > Instâncias >

Filtrar por Todos

<input type="checkbox"/>	Nome	Nome interno	Nome de exibição	Nome da zona	Estado
<input type="checkbox"/>	VMRECG-ELI	i-2-21-VM	VMRECG-ELI	csgl_zone	● Running
<input type="checkbox"/>	VMRECM-01	i-2-26-VM	VMRECM-01	csgl_zone	● Running
<input type="checkbox"/>	VMRECM	i-2-25-VM	VMRECM	csgl_zone	● Running
<input type="checkbox"/>	VMRECM-STOP	i-2-24-VM	VMRECM-STOP	csgl_zone	● Stopped
<input type="checkbox"/>	VMRECP-STOP	i-2-23-VM	VMRECP-STOP	csgl_zone	● Stopped
<input type="checkbox"/>	VMRECP-ELI	i-2-22-VM	VMRECP-ELI	csgl_zone	● Running

Fonte: CLOUDSTACK (2015)

A Figura 21 apresenta o gerenciamento de instâncias de máquinas virtuais do orquestrador *Apache CloudStack*. Posterior à realização das rotinas administrativas para uma destas instâncias, serão eliminadas as VMs VMRECG-ELI e VMRECP-ELI e com isso será apresentado um exemplo de como a ferramenta realizou o monitoramento e eliminou as informações inconsistentes do ambiente de nuvem de teste.

Figura 22 - Status de Instância de Máquina Virtual de Dados Persistentes

id	name	uuid	instance_name	state	type
4	r-4-VM	11b1ea66-1f75-4af0-9980-4739b5f6274f	r-4-VM	Running	DomainRouter
1	s-1-VM	96c349ed-9a63-4c6d-a257-ff936a1eec23	s-1-VM	Running	SecondaryStorageVm
2	v-2-VM	f4b54545-2440-4fa0-8c21-3762dc87bbd6	v-2-VM	Running	ConsoleProxy
21	VMRECG-ELI	589dfa84-1fec-4326-bcdb-b87a0f9043f3	i-2-21-VM	Expunging	User
25	VMRECM	88379122-44d2-46dd-b38a-6fb02e3f8fb3	i-2-25-VM	Running	User
26	VMRECM-01	5d7c3622-3ee6-44c3-9c65-5d7d3d22f272	i-2-26-VM	Running	User
24	VMRECM-STOP	26e4648d-d59d-4a08-b744-a61aa714b8e7	i-2-24-VM	Stopped	User
22	VMRECP-ELI	1f71c2b4-4551-4be7-b651-c7b67c1303e1	i-2-22-VM	Expunging	User
23	VMRECP-STOP	925dd74f-5dd0-466f-b984-f1d230c06980	i-2-23-VM	Stopped	User
NULL	NULL	NULL	NULL	NULL	NULL

Fonte: MYSQL (2015)

A Figura 22 demonstra a tabela vm_instance que oferece informações conforme a Figura 21, sendo que as VMs eliminadas do ambiente são identificadas através da coluna id com as informações 23 e 22.

Figura 23 - Atividades de Eventos no Orquestrador

Descrição	Nível	Type	Domínio	Conta	Data	Visi
user has logged in from IP Address 191.250.141.50	INFO	USER.LOGIN	ROOT	admin	19 Jul 2016 23:46:25	
user has logged in from IP Address 191.250.141.50	INFO	USER.LOGIN	ROOT	admin	19 Jul 2016 23:13:18	
Successfully completed destroying Vm. Vm Id: 22	INFO	VM.DESTROY	ROOT	admin	16 Jul 2016 21:53:08	
Successfully completed revoking firewall rule. Vm Id: 22	INFO	FIREWALL.C...	ROOT	admin	16 Jul 2016 21:53:08	

Fonte: CLOUDSTACK (2015)

Figura 24 - Atividades de Eventos do Orquestrador em Dados Persistentes

id	uuid	description	level	type	state
227	009f479e-0ed6-48b8-bd5f-af3a60235d79	Successfully created entity for deploying Vm. Vm Id: 21	INFO	VM.CREATE	Created
190	09f08df1-2713-4533-a8df-2cd188c48aa5	user has logged in from IP Address 177.99.132.65	INFO	USER.LOGIN	Completed
236	0f14a698-ad8a-460e-9588-3e4eb4514e59	starting Vm. Vm Id: 23	INFO	VM.CREATE	Scheduled
235	16486e60-e719-4f40-b230-c4cd34d1c0f6	Successfully created entity for deploying Vm. Vm Id: 23	INFO	VM.CREATE	Created
239	1720b2fb-4a7e-4801-90c1-c4b52f9be7be	Successfully created entity for deploying Vm. Vm Id: 24	INFO	VM.CREATE	Created
247	18b8f345-3ee5-4fda-a12f-e40769ceab8e	stopping Vm. Vm Id: 23	INFO	VM.STOP	Started
249	1cab44c0-9bae-4fc1-a8c0-fa5815a90936	Successfully created entity for deploying Vm. Vm Id: 25	INFO	VM.CREATE	Created
244	23c7a794-aa58-4784-a240-da2592e0bb04	stopping Vm. Vm Id: 24	INFO	VM.STOP	Started
238	3328af0e-64e1-4971-8e8e-984ffce33f6b	Successfully completed starting Vm. Vm Id: 23	INFO	VM.CREATE	Completed
248	39d04bb6-dcca-4f05-bfe7-fdb1f8413f52	Successfully completed stopping Vm. Vm Id: 23	INFO	VM.STOP	Completed
266	3bd953a4-b84c-4908-9715-383c75353e5a	Successfully completed destroying Vm. Vm Id: 22	INFO	VM.DESTROY	Completed
230	457481a3-9737-4bcd-aeaf-4b639f3e1f71	Successfully completed starting Vm. Vm Id: 21	INFO	VM.CREATE	Completed
254	54550db6-ef09-4103-a581-6c53d0a2ba9e	starting Vm. Vm Id: 26	INFO	VM.CREATE	Scheduled
234	66947958-c3f2-44d1-8e00-f9e9ecd7e049	Successfully completed starting Vm. Vm Id: 22	INFO	VM.CREATE	Completed
228	6ca75274-9199-4aa7-b0dc-0190715d9c1e	starting Vm. Vm Id: 21	INFO	VM.CREATE	Scheduled
261	6eaa5bee-8e0b-4e6a-be92-8d95e86caae2	Successfully completed destroying Vm. Vm Id: 21	INFO	VM.DESTROY	Completed
252	70a6f36d-8046-4c81-975b-c6842e50dbe4	Successfully completed starting Vm. Vm Id: 25	INFO	VM.CREATE	Completed
256	7185201a-e381-47c6-ba50-250dff32f72a	Successfully completed starting Vm. Vm Id: 26	INFO	VM.CREATE	Completed

Fonte: MYSQL (2015)

Para a Figura 23 e Figura 24 são demonstrados algumas atividades de eventos do orquestrador gerados a partir das rotinas de manutenção administrativa, executadas através do gerenciamento das instâncias demonstradas na Figura 21.

Verifica-se que após a eliminação das instâncias VMRECG-ELI e VMRECP-ELI, suas informações e também os eventos gerados para cada uma das rotinas de manutenção administrativas continuam armazenadas na base de dados persistentes do orquestrador, sendo que estas informações não têm relevância e utilidade ao ambiente de teste e ao administrador da nuvem.

Caso a eliminação destas informações armazenadas no banco de dados do orquestrador procedesse de forma manual, poderiam ser realizadas conforme apresentado no Quadro 6.

Quadro 6 - Eliminação Manual para Validar as Informações sem Utilização no Ambiente de Teste

ID Instância	Máquina Virtual Eliminada	Eliminação Instância de Máquina Virtual	ID Evento	Eliminação de Eventos Gerados por VMs
21	VMRECG-ELI	delete from cloud.vm_instance where id = '589dfa84-1fec-4326-bcdb-b87a0f9043f3';	227	delete from cloud.event where id = '009f479e-0ed6-48b8-bd5f-af3a60235d79';
			230	delete from cloud.event where id = '457481a3-9737-4bcd-aeaf-4b639f3e1f71';
			228	delete from cloud.event where id = '6ca75274-9199-4aa7-b0dc-0190715d9c1e';
			261	delete from cloud.event where id = '6eaa5bee-8e0b-4e6a-be92-8d95e86caae2';
22	VMRECP-ELI	delete from cloud.vm_instance where id = '1f71c2b4-4551-4be7-b651-c7b67c1303e1';	266	delete from cloud.event where id = '3bd953a4-b84c-4908-9715-383c75353e5a';
			234	delete from cloud.event where id = '66947958-c3f2-44d1-8e00-f9e9ecd7e049';

Fonte: Elaborado pelo autor.

No Quadro 6 são apresentadas as informações que devem ser eliminadas deste ambiente de nuvem de teste, correlacionando as informações dos eventos gerados com as instâncias de máquinas virtuais eliminadas, para que não haja problema em eliminar alguma informação que esteja em uso pelo ambiente.

Figura 25 - Ferramenta AMFC: Informações de Máquina Virtual e Atividades dos Eventos

```

VERIFICAR INCONSISTENCIAS ENCONTRADAS
delete from cloud.vm_instance where uuid = '589dfa84-1fec-4326-bcdb-b87a0f9043f3';
delete from cloud.vm_instance where uuid = '1f71c2b4-4551-4be7-b651-c7b67c1303e1';
VMs no Orquestrador:4 Registros Inconsistentes Base de Dados:2
VERIFICAR INCONSISTENCIAS ENCONTRADAS
delete from cloud.event where uuid = '009f479e-0ed6-48b8-bd5f-af3a60235d79';
delete from cloud.event where uuid = '3bd953a4-b84c-4908-9715-383c75353e5a';
delete from cloud.event where uuid = '457481a3-9737-4bcd-aeaf-4b639f3e1f71';
delete from cloud.event where uuid = '66947958-c3f2-44d1-8e00-f9e9ecd7e049';
delete from cloud.event where uuid = '6ca75274-9199-4aa7-b0dc-0190715d9c1e';
delete from cloud.event where uuid = '6eaa5bee-8e0b-4e6a-be92-8d95e86caae2';

```

Fonte: Elaborado pelo autor.

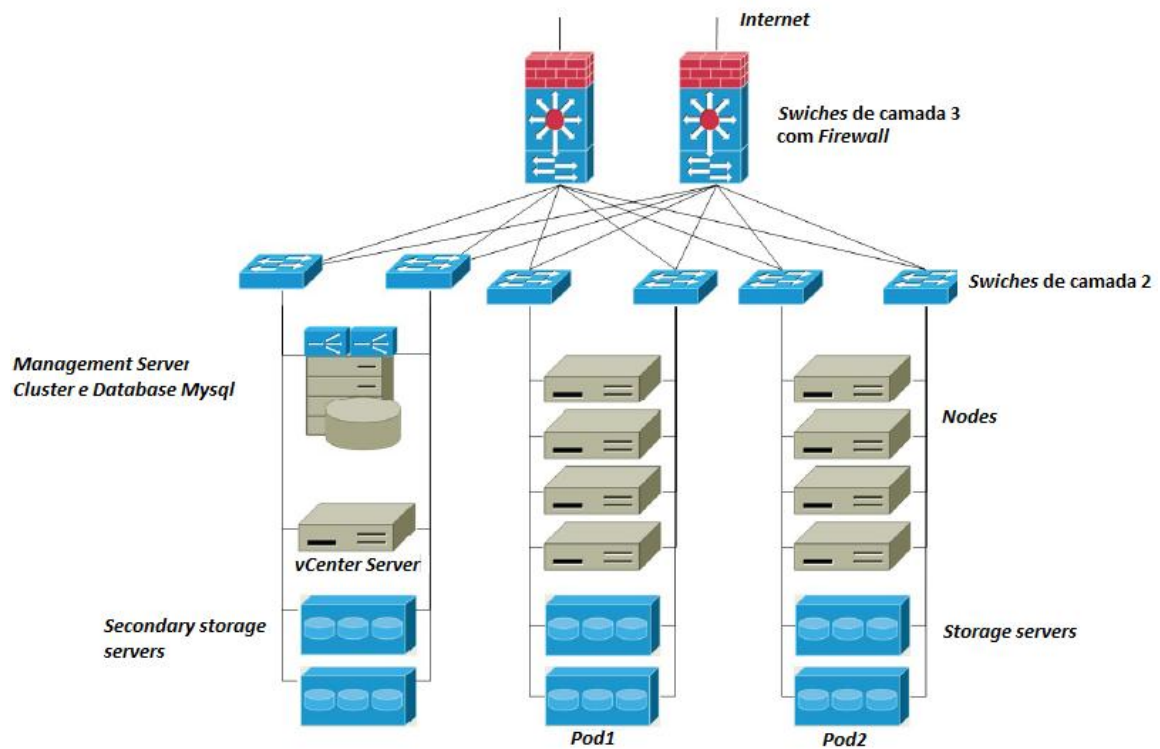
A Figura 25 apresenta a execução da ferramenta AMFC que a partir do monitoramento deste ambiente mostra as informações que devem ser eliminadas. A validação do resultado apresentado pela ferramenta fica comprovado devido ser o mesmo do Quadro 6, em que foi realizada de forma manual. A ferramenta elimina apenas as informações que não estão em utilização pelo ambiente de teste.

Convém observar que a AMFC apresenta a quantidade de recursos em utilização no ambiente de teste da nuvem e os registros que são considerados como informações irrelevantes, que devem ser eliminadas através desta sincronização por meio de *API's* entre o ambiente de teste da nuvem com as informações persistentes.

A importância de realizar esta política de eliminação de registros antigos ou não utilizados pode ser compreendida na implementação da estratégia de redundância de dados para tolerância a falhas na nuvem. Caso não seja adotado neste cenário de dados volumosos, pode ocasionar baixo desempenho para replicação dos dados e aumento dos custos no armazenamento destas informações irrelevantes ao negócio.

Como apresentado por Zhang e Chen (2011), a tolerância a falhas é fundamental para os sistemas de armazenamento que se estendem por várias nuvens ou múltiplos *pods* como apresentado na Figura 26 que necessitam estar conectados. Para ambos, é condicionado o uso da estratégia de redundância de dados e o esquema de reparação.

Figura 26 - Modelo de Redundância de Dados no Orquestrador *Apache CloudStack*



Fonte: CLOUDSTACK (2014)

A Figura 26 apresenta uma estratégia de redundância do orquestrador *Apache CloudStack*. No servidor de gerenciamento da nuvem que está o SGBD *MySQL*, cuja carga de dados foi dividida entre outros servidores de *storage*, existindo assim a redundância dos servidores, ocasionando uma alta disponibilidade neste ambiente.

6.4 Cenário 2: Monitorar Informação Inconsistente no Ambiente de Nuvem

A Figura 27 apresenta o gerenciador de armazenamento de volumes do orquestrador *Apache CloudStack*, sendo que não há nenhum volume criado.

Figura 27 - Gerenciamento de Armazenamento de Volumes



Fonte: CLOUDSTACK (2015)

Na Figura 27 é apresentada a tabela volumes do SGBD Mysql do orquestrador, que armazena as informações da Figura 28. As informações apresentadas são dos volumes criados para cada instância, pertencente e de uso da infraestrutura do orquestrador.

Figura 28 - Informação dos Volumes em Dados Persistentes

```

1 • select id,instance_id,name,uuid,size,volume_type from volumes;

```

id	instance_id	name	uuid	size	volume_type
1	1	ROOT-1	7789ef43-1f14-48ac-a1a0-bdaa29ee0d72	2621440000	ROOT
2	2	ROOT-2	46a6a9cb-348a-4da7-a499-60bd281ce064	2621440000	ROOT
4	4	ROOT-4	a7386df3-ba13-450a-9851-c483fd7c40ce	2621440000	ROOT
* NULL	NULL	NULL	NULL	NULL	NULL

Fonte: MYSQL (2015)

Figura 29 - Gerenciamento de Instâncias de Máquinas Virtuais

Home > Instâncias >

Filtrar por Todos

<input type="checkbox"/>	Nome	Nome interno	Nome de exibição	Nome da zona	Estado
<input type="checkbox"/>	VMRECM-STOP	i-2-31-VM	VMRECM-STOP	csgl_zone	Stopped
<input type="checkbox"/>	VMRECP	i-2-30-VM	VMRECP	csgl_zone	Running
<input type="checkbox"/>	VMRECM	i-2-29-VM	VMRECM	csgl_zone	Running
<input type="checkbox"/>	VMRECP-ELI	i-2-28-VM	VMRECP-ELI	csgl_zone	Running
<input type="checkbox"/>	VMRECG-ELI	i-2-27-VM	VMRECG-ELI	csgl_zone	Running

Fonte: CLOUDSTACK (2015)

Na Figura 29 é apresentado o gerenciamento de algumas instâncias de máquinas virtuais criadas do orquestrador *Apache CloudStack*, sendo que para cada uma são configurados e criados volumes de raiz e dados, sendo estes:

- Volume de raiz: Associados a uma instância no momento de sua criação, contém o sistema de arquivos da instância e dispositivos de inicialização, e
- Volume de dados: Criados e associados ou não a uma instância, armazenamento adicional de dados disponível para utilização dos usuários, mediante oferta configurada pelo administrador. (CLOUDSTACK, 2015).

A configuração dos discos de armazenamento de volume de dados está descrita conforme Quadro 4 na coluna disco.

Para as instâncias houve a realização das rotinas administrativas de manutenção descritas na Sessão 5.1 Arquitetura da ferramenta AMFC, após isso, foram eliminadas as VMs VMRECG-ELI e VMRECP-ELI.

Figura 30 - Informações Inconsistentes de Volumes em Dados Persistentes

The screenshot shows a MySQL query result grid. The query is: `select id,instance_id,name,uuid,size,volume_type from volumes;` The result grid displays the following data:

id	instance_id	name	uuid	size	volume_type
1	1	ROOT-1	7789ef43-1f14-48ac-a1a0-bdaa29ee0d72	2621440000	ROOT
2	2	ROOT-2	46a6a9cb-348a-4da7-a499-60bd281ce064	2621440000	ROOT
4	4	ROOT-4	a7386df3-ba13-450a-9851-c483fd7c40ce	2621440000	ROOT
44	27	ROOT-27	75d14a3e-2f7c-494c-89a6-afc0110a9913	21474836480	ROOT
45	NULL	DATA-27	60130694-1f2c-474d-8d79-56b44f1ed96b	107374182400	DATADISK
46	28	ROOT-28	d4e2284a-5a54-4fa6-84a6-7ba2cdef69c3	21474836480	ROOT
47	NULL	DATA-28	4b9f35e0-78e0-4598-b358-82caed33a054	5368709120	DATADISK
48	29	ROOT-29	5481f9ac-4b55-467d-bcac-1f4e48d18365	21474836480	ROOT
49	29	DATA-29	c0592583-16d5-461a-b645-e2dd2eec6887	21474836480	DATADISK
50	30	ROOT-30	4b0e03a1-cf6a-47ca-8453-5e03ce47e6bc	21474836480	ROOT
51	30	DATA-30	4e630d42-0b20-4c19-a311-c579ee68b7b6	5368709120	DATADISK
52	31	ROOT-31	232db8d7-8cd9-46c4-a019-43d60634815a	21474836480	ROOT
53	31	DATA-31	deb04b2-37a0-47f7-a6ad-afb99debfea9	21474836480	DATADISK
NULL	NULL	NULL	NULL	NULL	NULL

Fonte: MYSQL (2015)

Como a eliminação das duas instâncias de máquinas virtuais, os volumes raiz foram também eliminados, no entanto, para os volumes de dados não houve esta eliminação e sim a retirada da informação no campo `instance_id`, que referencia a instância virtual ao volume. Além disso, estes volumes de dados não continham nenhuma informação de usuário, porém continuavam no ambiente de nuvem como uma informação inconsistente conforme Figura 30.

Caso a eliminação destas informações armazenadas no banco de dados do orquestrador procedesse de forma manual, poderiam ser realizadas conforme apresentado no Quadro 7.

Quadro 7 - Eliminação Manual para Validar Informações de Volumes

ID Instância	Máquina Virtual Eliminada	Eliminação Instância de Máquina Virtual	ID Volume	Eliminação de Volumes por VMs Eliminadas
27	VMRECG-ELI	delete from cloud.vm_instance where id = '5ff2e9c9-0ee9-4f31-8ead-fe78366dd0cc';	44	delete from cloud.volumes where id = '75d14a3e-2f7c-494c-89a6-afc0110a9913';
			45	delete from cloud.volumes where id = '60130694-1f2c-474d-8d79-56b44f1ed96b';
28	VMRECP-ELI	delete from cloud.vm_instance where id = '65dc695c-a59d-4eef-9c2f-729651cf7bf0';	46	delete from cloud.volumes where id = 'd4e2284a-5a54-4fa6-84a6-7ba2cdef69c3';
			47	delete from cloud.volumes where id = '4b9f35e0-78e0-4598-b358-82caed33a054';

Fonte: Elaborado pelo autor.

Conforme apresenta a Figura 31 na execução da ferramenta AMFC foi realizada auditoria e monitoramento do ambiente de nuvem, que demonstra as inconsistências encontradas.

Figura 31 - Inconsistências de informações de volumes ferramenta AMFC

```

VERIFICAR INCONSISTENCIAS ENCONTRADAS
delete from cloud.volumes where uuid = '75d14a3e-2f7c-494c-89a6-afc0110a9913';
delete from cloud.volumes where uuid = '60130694-1f2c-474d-8d79-56b44f1ed96b';
delete from cloud.volumes where uuid = 'd4e2284a-5a54-4fa6-84a6-7ba2cdef69c3';
delete from cloud.volumes where uuid = '4b9f35e0-78e0-4598-b358-82caed33a054';
Volumes no Orquestrador:6 Registros Inconsistentes Base de Dados:4

```

Fonte: Elaborado pelo autor.

Verifica-se também que a sincronização entre ambiente de teste e dados persistentes está de acordo, em virtude da comparação com a Figura 30, sendo que para cada instância há dois volumes criados. Com isso, temos seis volumes em utilização pelo ambiente de nuvem e quatro volumes que não estão sendo mais utilizados pela nuvem. Na questão da validação do resultado apresentado pela ferramenta, fica comprovado em virtude do que mostra o Quadro 7 que são as mesmas informações a serem eliminadas, porém de forma manual.

5.5 Avaliação dos Resultados

Na implementação da ferramenta AMFC, a quantidade de inconsistências é contabilizada por meio da Equação 1 descrita na Sessão 4.4 em Definição dos Parâmetros de Cálculo. Posteriormente a este resultado foi definido o cálculo de inconsistências por oferta de recurso pela Equação 2, também descrita na Sessão 4.4 em Definição dos Parâmetros de Cálculo, que apresenta o resultado para cada tipo de modelo de recurso.

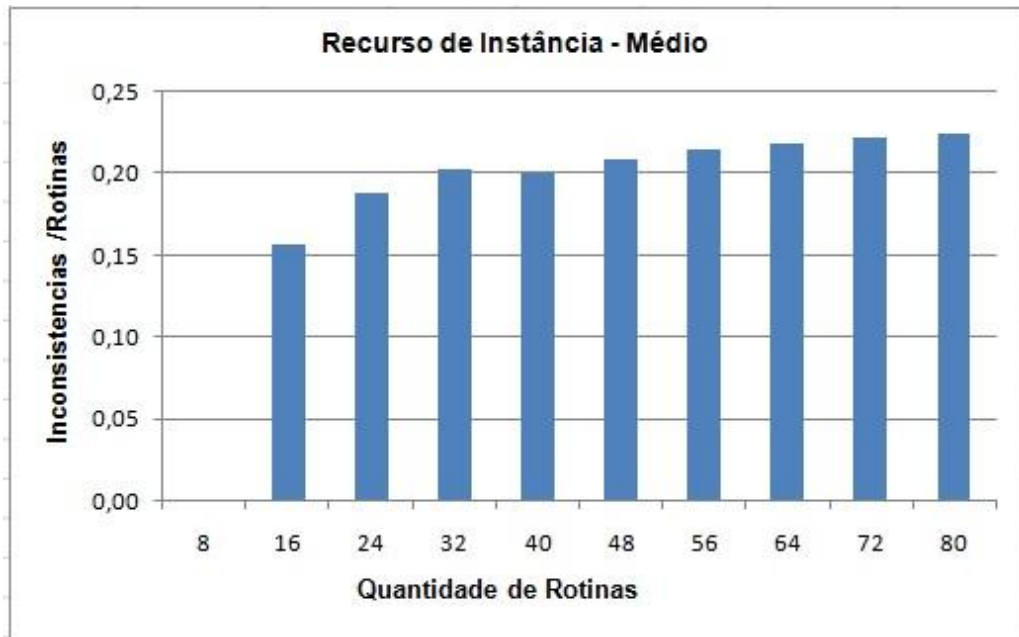
Para os modelos de recurso de instância de máquina virtual, apresentados nas Figuras 32, 33 e 34, foram descritos na Sessão 6.1 e configurados seguindo o Quadro 4. Também na Sessão 5.1 Arquitetura da ferramenta AMFC são descritas as rotinas de manutenção que foram realizadas para cada um destes modelos de recurso.

A Figura 32 apresenta o modelo de recurso de instância de máquina virtual pequeno, sendo possível observar que, com o aumento do número das rotinas de manutenções realizadas para este modelo de recurso, ocasionou uma maior incidência de inconsistências geradas no ambiente.

Figura 32 - Resultado do Modelo de Recurso Pequeno



Fonte: Elaborado pelo autor.

Figura 33 - Resultado do Modelo de Recurso Médio

Fonte: Elaborado pelo autor.

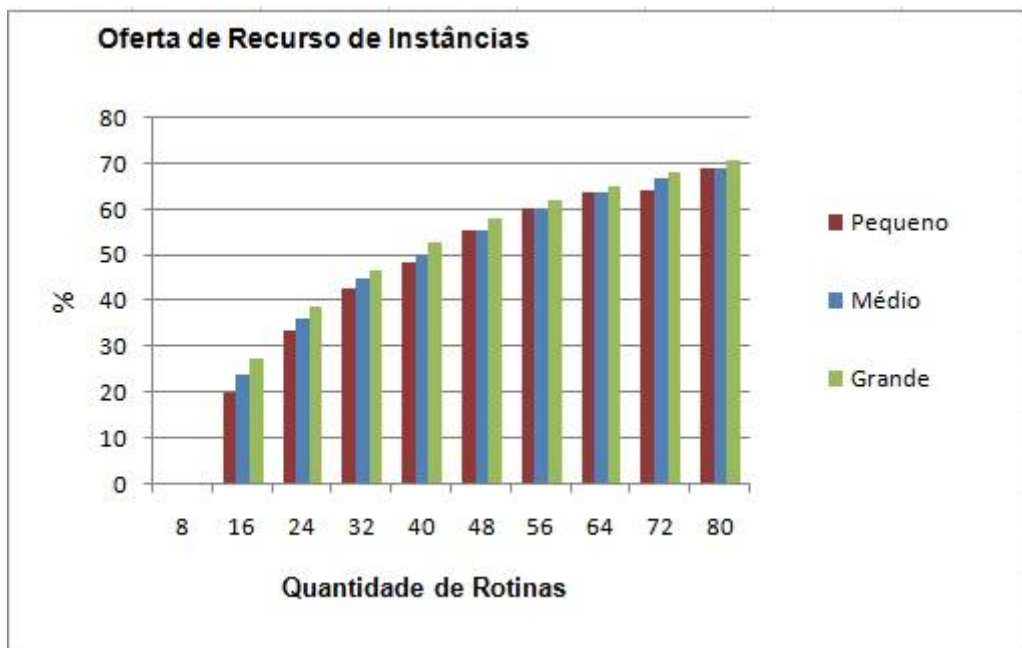
Comparando a Figura 33 e a Figura 34, constata-se que foi realizado o mesmo número de repetições de rotinas de manutenções. Nota-se que são modelos de recursos diferentes. Nesse ponto vale ressaltar que houve um aumento no número de inconsistências geradas no ambiente, o que significa que a configuração do recurso também é um fator que impacta no aumento das inconsistências geradas.

Figura 34 - Resultado do Modelo de Recurso Grande

Fonte: Elaborado pelo autor

A Figura 35 apresenta um comparativo entre os três modelos de recursos. Conforme pode ser visualizado na figura, cada recurso teve um aumento de inconsistências para o número de rotinas administrativas que foram realizadas. É importante ressaltar que foi possível identificar que o aumento da incidência de inconsistências geradas no ambiente é proveniente do número de rotinas administrativas que se realiza e também ao modelo de recurso configurado.

Figura 35 - Comparativo de Inconsistências entre Recursos



Fonte: Elaborado pelo autor

Quadro 8 - Ganho de Armazenamento de Dados Persistentes

Recurso	Disco (GB)	Disco Convertido (KB)	Ganho em proporcionalidade
Pequeno	5	5242880	0,00057983
Médio	20	20971520	0,00019836
Grande	100	104857600	0,00004120

Fonte: Elaborado pelo autor.

Por fim, o Quadro 8 apresenta o ganho de armazenamento de dados proporcionado pela ferramenta para cada tipo de recurso, sendo utilizado conforme Equação 4, descritos na Sessão 4.4 em Definição dos Parâmetros de Cálculo. O ganho foi proveniente da eliminação das inconsistências encontradas e também das informações que não seriam mais utilizadas

pelo ambiente de teste e que não comprometessem ao negócio, conforme Equação 3 descrita também na Sessão 4.4 em Definição dos Parâmetros de Cálculo.

Vale ressaltar que foi realizada conversão no disco de determinado recurso de GB (gigabyte) para KB (kbytes), em virtude da apresentação dos valores mediante cálculo efetivado.

Alguns dos desafios da computação em nuvem é oferecer ferramentas ou serviços que realizem o monitoramento contínuo da utilização de seus recursos, gerenciar e ajustá-los, para que possam atender as solicitações de SLAs (Acordo de Nível de Serviço) em tempo real. (SALEH et al., 2013).

6.6 Considerações Finais

Este capítulo apresentou o desenvolvimento da ferramenta AMFC, o tamanho das instâncias de máquinas virtuais utilizadas, que tiveram que ser limitadas em virtude do ambiente disponível para testes, a definição das rotinas de manutenção para geração de informações a serem coletadas e todas as etapas de monitoramento que é realizado pela ferramenta no ambiente de nuvem.

No entanto, não coube apenas demonstrar as etapas de desenvolvimento, mas sim, comprovar e evidenciar que esta ferramenta realiza o processo de auditoria e monitoramento em nuvem. Foi realizada a validação, através de casos de uso na eliminação de informações sem relevância e utilidade ao ambiente de teste e ao administrador da nuvem. Manteve-se a preocupação em não eliminar informação que estivesse sendo utilizada pelo ambiente de teste, e realizou-se o gerenciamento da informação para evitar a redundância não controlada e também a informação inconsistente que provoca a violação de integridade entre ambiente de teste com os dados persistentes, como por exemplo, um recurso que já foi eliminado seja contabilizado como se estivesse em uso pelo ambiente.

Na avaliação e resultado da ferramenta, observou-se que cada recurso teve um aumento da incidência de inconsistências geradas no ambiente a partir do número de rotinas administrativas que foram realizadas e também ao modelo de recurso configurado. Além disso, obteve-se ganho de armazenamento de informações através da eliminação das inconsistências encontradas e também das informações que não seriam mais utilizadas pelo ambiente de teste.

7 Conclusões

Esta dissertação realizou o desenvolvimento da ferramenta AMFC para auditoria e monitoramento que realizasse o sincronismo das informações atuais da nuvem com os dados persistentes, utilizando o orquestrador *Apache CloudStack*, o *hypervisor XenServer* e o sistema de armazenamento *NAS Openfiler*, além da realização de testes para eliminação das inconsistências e de informações sem uso para o ambiente, sendo possível obter ganho em armazenamento de dados persistentes para o ambiente de nuvem privado.

Foi ressaltada a importância da realização de auditoria e monitoramento em uma nuvem IaaS de forma contínua, no intuito de manter a segurança, confiabilidade, integridade das informações e prevenção de eventos que possam se tornar falsos positivos e falsos negativos.

Através dos testes realizados pela ferramenta AMFC no ambiente de computação em nuvem pode-se constatar importantes conclusões, conforme segue:

- A AMFC proporcionou que as informações inconsistentes geradas a partir de rotinas administrativas de uma máquina virtual fossem eliminadas, impedindo que os administradores recebam inundações de alertas de falso positivo e falso negativo conforme trabalho (XU et al., 2015), evitando assim que administradores tomem decisões precipitadas quanto a configuração e manutenção do ambiente.
- Com a eliminação de informações inconsistentes na base de dados persistentes, os administradores obtêm uma maior facilidade e precisão em diagnosticar um evento falho. Comparado ao trabalho (XU et al., 2014), a facilidade e precisão se evidenciam para o administrador quando este necessita averiguar um problema que está ocorrendo no ambiente, devido a base de dados estar íntegra e sem inconsistências, pois a busca da informação demanda menos tempo de consulta.
- O trabalho de (LIN et al., 2015) apresenta preocupação na segurança da geração de *logs* em conter informações de usuários e de administradores que estejam armazenadas em uma mesma tabela na base de dados. Ocasionalmente vulnerabilidade caso um invasor consiga ter acesso a esta tabela de *log*, pois poderá obter informações administrativas da nuvem. A ferramenta AMFC realiza auditoria e monitoramento de uma instância de máquina virtual, independente do nível de usuário, no entanto, não armazena e nem cria *logs* de informações monitoradas.

7.1 Contribuições

Ao final deste trabalho, as seguintes contribuições podem ser apresentadas:

- A ferramenta AMFC, que permite a eliminação de informações sem relevância e utilidade ao ambiente de teste, possibilitando o gerenciamento das informações para se evitar a redundância não controlada.
- Foi possível realizar um comparativo, conforme descritos nas Sessões 3.5 e 3.3, que apresentam análises e monitoramento das informações a partir de *logs* distribuídos na nuvem, integração com outros frameworks de monitoramento de nuvem. Entretanto, não apresentam a informação em comparação com o ambiente de teste, ou seja, o administrador ao utilizar a ferramenta AMFC, consegue ter o resultado das informações que precisa do estado de teste da nuvem. Com isso, a tomada de decisão se torna mais rápida e em tempo real.
- Através do estudo realizado neste trabalho, foi possível entender como se detectar e posteriormente prevenir os eventos inconsistentes gerados que podem ocasionar os falsos positivos e falsos negativos, que induzem o administrador a cometer erros na gestão do ambiente. Viabilizou-se assim, uma nova estratégia de auditoria e monitoramento que correlaciona as informações atuais do ambiente com os dados persistentes em tempo real.
- A utilização da ferramenta na estratégia de redundância no orquestrador é importante para os sistemas de armazenamento, que se estendem por várias nuvens ou múltiplos *pods*, com a eliminação de informações sem relevância, pois evita a queda do desempenho na replicação dos dados e aumento dos custos de armazenamento.

7.2 Trabalhos Futuros

Para incentivar a continuação desse trabalho, são apresentadas algumas idéias para a ferramenta AMFC na criação de novas soluções:

- **Comportamento dos serviços na nuvem:** Realizar acompanhamento do comportamento dos serviços que estão sendo disponibilizados pelos recursos de *VM's*. Definir as informações a serem coletadas, armazená-las em uma base de dados *NoSql*, posteriormente avaliar o que causou uma indisponibilidade e sugerir correção. Poderia ser utilizada como uma base de dados de conhecimento para auxiliar o administrador.
- **Envio de Alertas para Equipamentos Móveis:** a ferramenta poderia enviar alertas do ambiente monitorado também para equipamentos móveis.
- **Ampliação da Auditoria e Monitoramento no Ambiente:** a ferramenta poderia realizar a auditoria e monitoramento de outros recursos e serviços na nuvem.
- **Mecanismo de *Live Migration*:** a ferramenta poderia realizar a migração do volume de dados do usuário de uma instância de máquina virtual, para outro local de armazenamento.

Referências Bibliográficas

- ACHCAR, A. J.; RODRIGUES, J. **Introdução à Estatística para Ciências e Tecnologia**. 1ª ed. São Paulo: USP, 1998. p.25-41, p.84-132
- ALMURAYH, A., SEMWAL, S. **Controlling Xen Cloud platform via smart phones**. Information Conference on Reuse and Integration (IRI), 2013 IEEE 14th International, vol., no., pp. 676 - 683, 14-16 Aug. 2013.
- AMAZON, **Centro de Arquitetura da AWS**, Acesso em: <<http://aws.amazon.com/pt/architecture/>>, 02 de novembro de 2014.
- APACHE. **CloudStack cloudmonkey CLI in Apache Documentation**. Acesso em: <<https://cwiki.apache.org/confluence/display/CLOUDSTACK/CloudStack+cloudmonkey+CLI>>, 30 de Novembro de 2015.
- ARMBRUST, M.; FOX, A.; GRIFFITH, R.; JOSEPH, A. D.; KATZ, R.; KONWINSKI, A.; LEE, G.; PATTERSON, D.; RABKIN, A.; STOICA, I.; ZAHARIA, M. **Above the Clouds: A Berkeley View of Cloud Computing**. UC Electrical Engineering and Computer Sciences University of California at Berkeley, vol., no., pp. 01 - 21, 10 Feb. 2009.
- AVILES, M.; RUTNER, P.; DICK, G. **Logistics management: Opportunities in the cloud**. SAIS 2012 Proceedings. Atlanta, n.3, p. 11-17, Jan. 2012.
- AZURE, W. **Centro de documentos**. Acesso em: <<http://azure.microsoft.com/pt-br/documentation/>>, 02 de novembro de 2014.
- BARKAT, A.; SANTOS, A., D. ; HO, T., T., N. **Open Stack and Cloud Stack: Open Source Solutions for Building Public and Private Clouds**. Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2014 16th International, vol., no., pp. 429 - 436, 22-25 Sept. 2014.
- BUSINESS INNOVATION TECHNOLOGIES INC. **Ambiente computacional com base na computação em nuvem**. Acesso em: <<http://www.businnovatech.com/cloud-computing.html>>, 01 de julho de 2014.
- BUYYA, R. **Introduction to the IEEE Transactions on Cloud Computing**. IEEE TRANSACTIONS ON CLOUD COMPUTING, 2013, vol. 01, no., pp. 3 - 21, 25 Nov.2013.
- BUYYA, R., YEO, C. VENUGOPAL, S. **Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities**. Conference on High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International, vol., no., pp. 5 - 13, 25-27 Sept. 2008.
- CARR, N. **The Big Switch: Rewiring the World, from Edison to Google**. Norton & Company, 1 ed. USA: W. W. Norton & Company, 2008.

CASTILLO, J. A. ,L. ; MALLICHAN, M.; AL-HAZMI, Y. **OpenStack Federation in Experimentation Multi-cloud Testbeds**. 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, vol.2, no., pp. 51 - 56, 2-5 Dec. 2013.

CHEN, Q. ; XU, F. **Study on Water Information Cloud of Nanjing Based on CloudStack**. Symposium on Distributed Computing and Applications to Business, Engineering & Science (DCABES), 2013 12th International, vol., no., pp. 132 - 136, 2-4 Sept. 2013.

CITRIX, **The Xen Project is Built for Cloud Computing**. Acesso em: <<http://www.xenproject.org/users/cloud.html/>> , 02 de novembro de 2014 .

CLOUDCLIMATE. **Cloud Hosting and Cloud Storage Performance Dashboard**. Acesso em: <<http://www.cloudclimate.com/>>, 04 de Julho de 2016.

CLOUDSTACK. **Conferencia para usuários corporativos, desenvolvedores e fornecedores de serviços relacionados ao Apache CloudStack™**. Acesso em: <<http://cloudstackday.com.br/>>, 04 de Abril de 2015.

CLOUDSTACK. **Open Source Cloud Computing**. Acesso em: <<http://cloudstack.apache.org/>>, 02 de Abril de 2015.

DIVYA, S., E.; REDDY, K. K. **Maintenance of Cloud Data for Verification of Consistency**. INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ENGINEERING AND SCIENCE. vol.3, no.6, pp. 1921-1925, Jan. 2015.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 4ª ed. São Paulo: Pearson Addison Wesley, 2005.

EUCALYPTUS. **Open source AWS compatible private cloud software**. Acesso em: <<https://www.eucalyptus.com/>> , 02 de Novembro de 2015.

HASSAN, S.A.Z. **SONA: A Service Oriented Nodes Architecture for Developing Cloud Computing Applications**. International Conference on Advanced Computing and Communication Systems (ICACCS), vol., no., pp. 1 - 6, 19-21 Dec. 2013.

HE, Z.; HE, Y. **Analysis on the security of cloud computing**. Proc. Spie, Qingdao, China, n., p.7752-775204, 2011.

HU, B.; YU, H. **Research of Scheduling Strategy on OpenStack**. International Conference on Cloud Computing and Big Data (CloudCom-Asia), vol., no., pp. 191 - 196, 16-19 Dec. 2013.

IRIMIE, B.; PETCU, D. **Scalable and fault tolerant monitoring of security parameters in the cloud**. 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). vol., no., pp. 289 - 295, 21-24 Sept. 2015.

ISMAIL, U.; M.; ISLAM, S. ; MOURATIDIS, H. **Cloud Security Audit for Migration and Continuous Monitoring**. Trustcom/BigDataSE/ISPA, 2015 IEEE. vol.1, no., pp. 1081 - 1087, 20-22 Aug. 2015.

ISO. **Information technology — Cloud computing — Overview and vocabulary.** International Standard: ISO/IEC17788, vol., no., pp.11-16, 15 Oct. 2014.

ISO. **Information technology — Cloud computing — Reference architecture.** International Standard: ISO/IEC17789, vol., no., pp.31-36, 15 Oct. 2014.

KAUFMAN, L. M. **Data Security in the World of Cloud Computing.** IEEE Security & Privacy, vol.7, no.4, pp. 61 - 64, 10 Aug. 2009.

KIM, W. Cloud computing: today and tomorrow. **Journal of Object Technology.** Zurich: ETH Zurich, v.8, n.1, p. 65-72, Jan-Fev/2009.

LEIMESTER, S.; BOHM, M.; RIEDL, C.; KRCCMAR, H. **The business perspective of cloud computing: Actors, roles and value networks.** Proceedings of 18th European Conference on Information Systems (ECIS 2010). v., n., p. 1-12, Feb 2010.

LIANG, J. DESAI, A. ; JIN, H. ; XUWEN Y. ; GUAN, J. ; KAMBO, R. **Understand and replay cloud platform SDK with vim Shark.** International Conference on Cloud and Service Computing (CSC). v., n., p. 243 - 250, 12-14 Dec. 2011.

LIN, CH. ; CHANG,M ; CHIU,H. ; SHYU, H. **Secure logging framework integrating with cloud database.** Conference on Security Technology (ICCST), 2015 International Carnahan. vol., no., pp. 13 - 17, 21-24 Sept. 2015.

MADHUBALA, P. **Survey on security concerns in Cloud computing.** Conference on Green Computing and Internet of Things (ICGCIoT), 2015 International . vol., no., pp. 1458 - 1462, 8-10 Oct. 2015.

MAJUMDAR,S.; MADI. T. ; WANG, Y. ; JARRAYA, Y.; POURZANDI, M.; WANG,L. ; DEBBABI. M. **Security Compliance Auditing of Identity and Access Management in the Cloud: Application to OpenStack.** 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom). vol., no., pp. 58 - 65, 30 Dec. 2015.

MEERA, G; GEETHAKUMARI, G. **A Provenance Auditing Framework for Cloud Computing Systems.** Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015 IEEE International, vol., no., pp. 1 - 5, 19-21 Feb. 2015.

MDHAFFAR, A.; RIADH, H., B.; JMAIEL ,M. ; FREISLEBEN, B. **A Dynamic Complex Event Processing Architecture for Cloud Monitoring and Analysis.** IEEE International Conference on Cloud Computing Technology and Science, vol.2, no., pp. 270 - 275, 2-5 Dec. 2013.

MYSQL. **Funções Diversas**, Acesso em: <<http://ftp.nchu.edu.tw/MySQL/doc/refman/4.1/pt/miscellaneous-functions.html>>, 24 de junho de 2016.

MYSQL. **MySQL Workbench**, Acesso em: <<https://dev.mysql.com/downloads/workbench/>>, 08 de dezembro de 2015.

NIST. **NIST Programa de Cloud Computing**. Acesso em: <<http://www.nist.gov/itl/cloud/index.cfm>> , 20 de Agosto de 2014.

NIST. **Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations**. Computer Security Division Information Technology Laboratory National Institute of Standards and Technology - NIST 800-137, vol., no., pp.6-8, Sept. 2011.

NURMI, D., WOLSKI, R., GRZEGORCZYK, C., OBERTELLI, G., SOMAN, S., YOUSEFF, L., ZOGORODNOV, D. **The Eucalyptus Open-source Cloud-computing System**. Symposium on Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International. v., n., p. 124 - 131, 18-21 May. 2009.

OPENNEBULA. **Cloud and data center virtual infrastructure management**. Acesso em: <<http://opennebula.org/>>, 02 de Novembro de 2014.

OPENSTACK. **OpenStack Administrator Guide**. Acesso em: <<http://docs.openstack.org/admin-guide/>> , 01 de Setembro de 2014.

PARK, J.; S.; SPETKA, E.; RASHEED, H.; RATAZZI, P.; HAN, K.; J. **Near-Real-Time Cloud Auditing for Rapid Response**. Conference on Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International. vol., no., pp. 1252 - 1257, 26-29 March 2012.

PYTHON. **Python**. Acesso em: <<https://www.python.org/>>, 12 de Novembro de 2015.

QAISAR, J. E.; MARKETS, C. E. **Introduction to Cloud Computing for Developers**. Information Technology Professional Conference (TCF Pro IT), 2012 IEEE TCF. v., n., p. 1 - 6, 9-10 March. 2012.

RACKSPACE. **Rackspace Cloud Monitoring**. Acesso em: <<http://www.rackspace.com/cloud/monitoring/>>, 04 de Julho de 2016.

SABAHI, F. **Cloud computing security threats and responses**. *Communication Software and Networks (ICCSN)*, 2011. IEEE 3rd International Conference on May 2011.

SALEH, O.; GROPPENGIEBER, F.; BETZ, H.; MANDARAWI, W.; SATTLER, K. **Monitoring and Autoscaling IaaS Clouds: A Case for Complex Event Processing on Data Streams**. Conference on Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International. vol., no., pp. 387 - 392, 9-12 Dec. 2013.

SHAIKH, F. B.; HAIDER, S. **Security Threats in Cloud Computing**. International Conference for Internet Technology and Secured Transactions (ICITST). v., n., p. 214 - 219, 11-14 Dec. 2011.

SOUSA, E., SILVA, E., LINS, F., TAVARES, E., MACIEL, P. **Dependability Evaluation of Cloud Infrastructures**. 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC). v., n., p. 1282 - 1287, 5-8 Oct. 2014.

SUCIU, G. ; HALUNGA, S. ; OCHIAN, A. ; SUCIU, V. **Network Management and Monitoring for Cloud Systems**. Conference 6th International on Electronics, Computers and Artificial Intelligence (ECAI) , vol., no., pp. 1 - 4, 23-25 Oct. 2014.

TAURION, C. **Cloud Computing: Computação em Nuvem: Transformando o mundo da tecnologia da informação**. Rio de Janeiro: Brasport, 2009.

VAQUERO, L. M.; R., L.; CACERES, J.; LINDNER, M. **A Break in the Clouds: Towards a Cloud Definition**. Special Interest Group on Data Communication. (SIGCOMM).ACM, v.39, n.1, p. 50 - 55, 05 Jan. 2009.

VERAS, M. **Cloud computing: nova arquitetura da TI**. Rio de Janeiro: Brasport, 2012.

VMWARE. **Server Consolidation**. Acesso em: <<http://www.vmware.com>>, 01 de Setembro de 2014.

WAQAS, A.; YUSOF, Z.; M. ; SHAH, A. **Fault tolerant cloud auditing**. Conference on Information and Communication Technology for the Muslim World (ICT4M), 2013 5th International vol., no., pp. 1 - 5, 26-27 March. 2013.

WEN,X.,GU, G., LI, Q., GAO, Y., ZHANG, X. **Comparison of Open-Source Cloud Management Platforms: OpenStack and OpenNebula**. Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International. v., n., p. 2457 - 2461, 29-31 May. 2012.

XIAOJIANG, L; YANLEI, S. **The Design and Implementation of Resource Monitoring for Cloud Computing Service Platform**. 3rd International Conference on Computer Science and Network Technology, 2013 IEEE International, vol., no., pp. 239 - 243, 12-13 Oct. 2013.

XU,X; ZHU,L; INGO,W; BASS,L; SUN,D. **Crying Wolf and Meaning it:Reducing False Alarms in Monitoring of Sporadic Operations through POD-Monitor**. Complex faUlts and Failures in Large Software Systems (COUFLESS), 2015 IEEE/ACM 1st International Workshop on Complex faUlts and Failures in Large Software Systems, vol., no., pp. 69 - 75, 23-23 May 2015.

XU,X; ZHU,L; INGO,W; BASS,L; SUN,D. **POD-Diagnosis: Error Diagnosis of Sporadic Operations on Cloud Applications**. International Conference on Dependable Systems and Networks, 2014 IEEE International, vol., no., pp. 252 - 263, 23-26 June 2014.

YAQIANG, M.; XUJIAN, G.; YUAN, L. **HVSM: An In-Out-VM security monitoring architecture in IAAS cloud**. Conference on International Information and Network Security, ICINS 2014. vol., no., pp. 185 - 192, 14-16 Nov. 2014.

YU, H. ; VAHDAT, A. **Design and Evaluation of a Conit-based Continuous Consistency model for Replicated Services**. ACM Trans. Comp. Sys, (20)3:239-282, 2002.

ZHANG, J. **A Data Synchronization Method Oriented to Custom Hierarchical Multi-node System**. IEEE International Conference on Computational Intelligence & Communication Technology (CICT). v., n., p. 666 - 669, 13-14 Feb. 2015.

ZHANG, S.; CHEN, X.; ZHANG, S.; HUO, X. **The Comparison Between Cloud Computing and Grid Computing.** International Conference on Computer Application and System Modeling (ICCASM 2010). v., n., p. V11-72 - V11-75, 22-24 Oct. 2010.

ZHANG, X.; CHEN, L. **Fault tolerance study for durable storage on the cloud.** International Conference on Cloud and Service Computing (CSC) . vol., n., pp.360-365, 12-14 Dec. 2011.

ZHU, Q.; TUNG, T. ; XIE, Q. **Automatic Fault Diagnosis in Cloud Infrastructure.** IEEE 5th International Conference on Cloud Computing Technology and Science . vol.1, no., pp. 467 - 474, 2-5 Dec. 2013.

Apêndice A - Instalação do Apache CloudStack

Para instalação do *Apache CloudStack* consiste em dois componentes principais, o servidor de gerenciamento e a infraestrutura de nuvem que ele gerencia.

No laboratório de estudos foram utilizados seis computadores da marca Lenovo ThinkCentre M58 3.0GHz 4GB 500GB, sendo estes, um servidor de gerenciamento o orquestrador, dois computadores com o *hypervisor* que realiza a criação das máquinas virtuais, dois computadores para armazenamento de dados e um computador para realizar o gerenciamento dos *hypervisors*, uma apresentação resumida de cada equipamento neste cluster de estudos, conforme pode ser observado na Quadro 8.

Quadro 9 - Equipamentos utilizados para o ambiente de protótipo da computação em nuvem

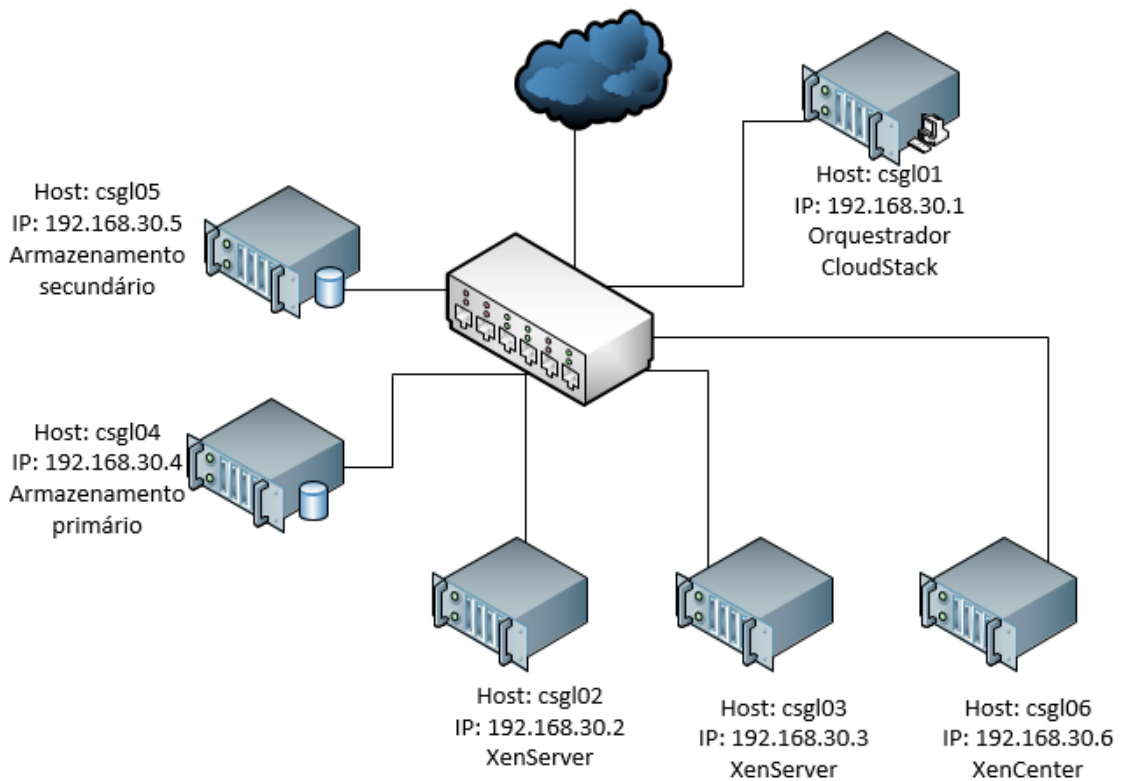
Quantidade de computadores	Definição do equipamento no ambiente de nuvem	Instalação realizada	Algumas funcionalidades para o ambiente
1	Orquestrador	Linux (CentOS 6.6 64 bits)	Interface web de usuário para administrador e de referência para usuários, gerencia a alocação de storage (discos virtuais para hóspedes) e database Mysql para persistência de dados.
2	Hosts	Hypervisor (XenServer 6.2)	Prover os recursos de CPU, memória, armazenamento e de rede necessários para hospedar as máquinas virtuais.
1	Armazenamento de dados primário	Openfiler 2.99 Linux CentOS	Armazenar os volumes de discos para todas as máquinas virtuais sendo executadas em hosts neste cluster.

1	Armazenamento de dados secundário	Openfiler 2.99 Linux CentOS	Armazenar dados como templates, imagens ISO e snapshots de volumes de disco.
1	Gerenciamento dos hosts	XenCenter 6.5	Aplicativo de gerenciamento e monitoração das máquinas virtuais e servidores físicos.

Fonte: Elaborado pelo autor.

Na Figura 35 é apresentado o modelo do layout de instalação do CloudStack, pode-se observar que neste layout estão as definições de cada equipamento, identificação e nome utilizados em rede.

Figura 36 - Layout de instalação do CloudStack



Fonte: Elaborado pelo autor.

Primeiro foi realizada a instalação da plataforma de sistema operacional Linux *CentOS* 6.6 *minimal* de 64bits.

Para instalação do sistema operacional via DVD, realizado conforme passo a passo do seu fabricante, após plataforma instalada foi realizada as seguintes configurações conforme segue abaixo:

Orquestrador *ApacheCloudStack*

Host: csgl01 IP: 192.168.30.1 e máscara de rede: 255.255.255.0

```
yum check-update
```

Configuração das interfaces de rede externa e interna.

```
vi /etc/sysconfig/network-scripts/
```

```
vi ifcfg-eth0
```

```
DEVICE=eth0  
HWADDR=6C:F0:49:F9:29:51  
TYPE=Ethernet  
UUID=85938ecc-dcb8-4adf-adeb-0cfab2fdea60  
ONBOOT=yes  
NM_CONTROLLED=yes  
BOOTPROTO=dhcp  
USERCTL=no  
IPV6INIT=no  
PEERDNS=yes
```

```
vi ifcfg-eth2
```

```
DEVICE=eth2  
BOOTPROTO=static  
TYPE=Ethernet  
ONBOOT=yes  
NM_CONTROLLED=yes  
IPADDR=192.168.30.1  
NETMASK=255.255.255.0  
USERCTL=no
```

Definição do nome da máquina

```
hostname --fqdn  
hostname  
csgl01.localhost
```

Configuração dos nomes das máquinas utilizadas em rede.

```
vi /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1      localhost localhost.localdomain localhost6 localhost6.localdomain6

### IPs públicos
200.145.157.38 csgl01.localhost    csgl01

### Ips privados
192.168.30.1  csgl01.localhost    csgl01
192.168.30.2  csgl02.localhost    csgl02
192.168.30.3  csgl03.localhost    csgl03
192.168.30.4  csgl04.localhost    csgl04
192.168.30.5  csgl05.localhost    csgl05
192.168.30.6  csgl06.localhost    csgl06
```

Configuração do SELinux para que o orquestrador *CloudStack* não venha ter restrições em seu ambiente para atualizações futuras.

```
vi /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Instalação do pacote *ntp*, funciona como *daemon* para ajuste e precisão de horário.

```
yum -y install ntp
```

Configuração do repositório a ser utilizado para instalação do orquestrador *CloudStack*

```
vi /etc/yum.repos.d/cloudstack.repo

[cloudstack]
name=cloudstack
baseurl=http://cloudstack.appt-get.eu/rhel/4.4
enabled=1
gpgcheck=0
```

Instalação do pacote *nfs-utils* que será utilizado para inicialização do armazenamento primário e secundário via NFS e realização da configuração de compartilhamento do *primary storage e secondary storage*.

```
yum -y install nfs-utils

mkdir /primary
mkdir /secondary
```

Configuração do arquivo *nfs*, abaixo estão as linhas que precisam serem habilitadas.

```
vi /etc/sysconfig/nfs

LOCKD_TCPPOINT=32803
LOCKD_UDPPOINT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

Liberação do firewall para que não haja bloqueio de interno e externo.

```
vi /etc/sysconfig/iptables

-A INPUT -p tcp -m tcp --dport 9090 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 8250 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 7080 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 8080 -j ACCEPT
```

Para instalação do orquestrador *CloudStack* a primeira etapa inicia-se com a instalação e configuração do banco de dados Mysql.

```
yum -y install mysql-server
```

Em seu arquivo de configuração *my.cnf* foram acrescentadas estas linhas abaixo.

```
vi /etc/my.cnf  
  
innodb_rollback_on_timeout=1  
innodb_lock_wait_timeout=600  
max_connections=350  
log-bin=mysql-bin  
binlog-format = 'ROW'  
  
service mysqld start  
chkconfig mysqld on
```

Instalação do servidor de gerenciamento do orquestrador.

```
yum -y install cloudstack-management
```

Versão do orquestrador baixada e utilizada para ser instalada .

```
rpm -ivh cloudstack-management-4.4.2-NONOSS_1.el6.x86_64.rpm  
yum localinstall cloudstack-management-4.4.2-NONOSS_1.el6.x86_64.rpm
```

No banco de dados Mysql foi criado um usuário admin, após isso, foi realizado comando para realização do sincronismo da tabela *cloud* que contem este usuário com o orquestrador, a seguir na Figura 36.

Figura 37 - SGBD MySQL do orquestrador CloudStack

```

root@csgl01:~
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cloud |
| cloud_usage |
| cloudbridge |
| mysql |
| test |
+-----+
6 rows in set (0.00 sec)

mysql> select id,username,email from user;
+----+-----+-----+
| id | username | email |
+----+-----+-----+
| 1 | system | NULL |
| 2 | admin | NULL |
+----+-----+-----+
2 rows in set (0.00 sec)

```

Fonte: Elaborado pelo autor.

```
cloudstack-setup-databases cloud:password@localhost --deploy-as=root
```

Instalação do agente *cloudstack* que é responsável pela comunicação do servidor de gerenciamento do orquestrador e controla todas as instâncias no host.

```

yum install cloudstack-agent
service cloudstack-management status
service cloudstack-management stop
service cloudstack-management start

```

Criação do compartilhamento que recebera as informações dos hosts de armazenamento do *primary storage* e *secondary storage*.

```

mount 192.168.30.4:/mnt/csgl_vg/csgl_volume/csgl /primary/
mount 192.168.30.5:/mnt/csgl_vg/csgl_volume/csgl_sec /secondary/

```

Apêndice B - Instalação do *XenServer*

Na instalação do *XenServer* serão utilizados dois computadores, terão a mesma função de gerenciamento de máquinas virtuais.

XenServers

Host: csgl02IP: 192.168.30.2 e máscara de rede: 255.255.255.0

Host: csgl03IP: 192.168.30.3 e máscara de rede: 255.255.255.0

Software baixado para instalação: <http://xenserver.org/open-source-virtualization-download.html>, através de DVD passo a passo, conforme telas apresentadas pelo fabricante.

Realizado uma atualização de correção de erros, solicitado pelo fabricante conforme em: <http://support.citrix.com/article/CTX139788>.

```
wget http://support.citrix.com/article/CTX139788
wget http://downloadns.citrix.com.edgesuite.net/8707/XS62ESP1.zip
unzip XS62ESP1.zip
```

Instalação do pacote ntp, funciona como *daemon* para ajuste e precisão de horário.

```
yum -y install ntp
```

Definição do nome da máquina

```
hostname --fqdn
hostname
csgl02.localhost
```

Configuração dos nomes das máquinas utilizadas em rede

```
vi /etc/hosts
127.0.0.1 csgl02    csgl02.localhost
192.168.30.2  csgl02  csgl02.localhost
192.168.30.1  csgl01  csgl01.localhost
```

Criação do compartilhamento que recebera as informações dos hosts de armazenamento do *primary storage* e *secondary storage*.

```
mount 192.168.30.4:/mnt/csgl_vg/csgl_volume/csgl /primary/
mount 192.168.30.5:/mnt/csgl_vg/csgl_volume/csgl_sec /secondary/
```

Liberação do firewall para que não haja bloqueio de interno e externo.

```
vi /etc/sysconfig/iptables

# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 67 --in-interface xenapi -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 694 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
```

Configuração do pool-list

```
xe pool-list

uuid ( RO)          : 973e7341-888d-7ede-da26-e2cfd8b94d64
name-label ( RW): csgl_pool
  name-description ( RW): csgl_pool
master ( RO): de36ae88-0aa3-4ffd-a7fa-f483645b04e9
default-SR ( RW): <not in database>
```

Configuração dos host-list

```
xe host-list

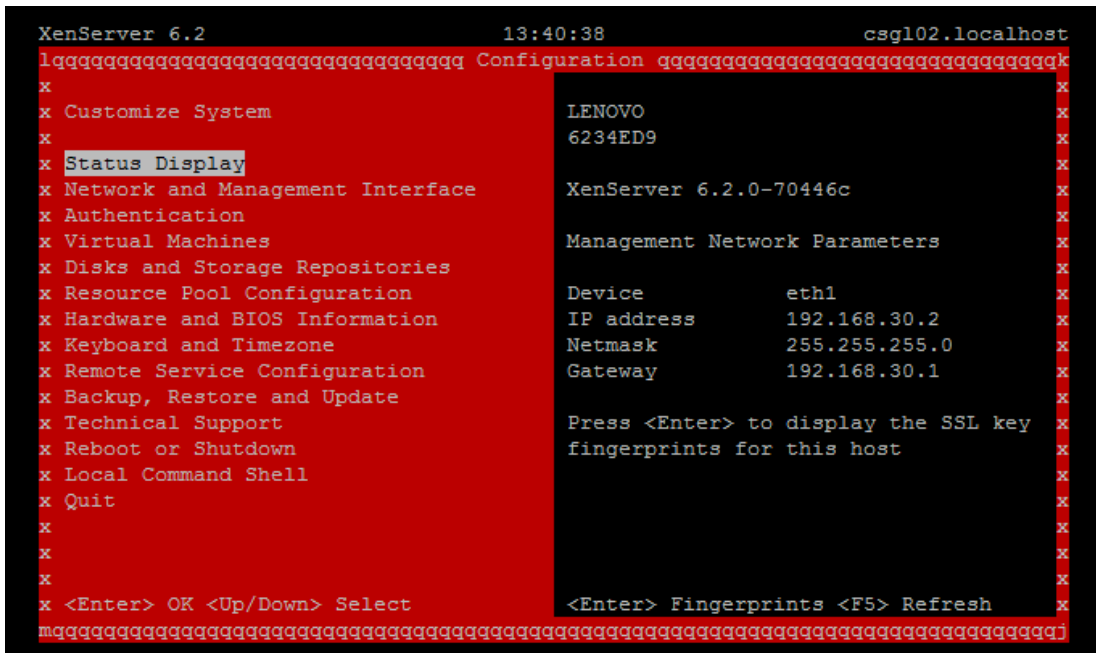
uuid ( RO)           : de36ae88-0aa3-4ffd-a7fa-f483645b04e9
name-label ( RW): csgl02.localhost
name-description ( RW): Default install of XenServer

uuid ( RO)           : 24373c8d-06cb-4cab-a450-1ddfa4030d70
name-label ( RW): csgl03.localhost
name-description ( RW): Default install of XenServer
```

Comando para acesso a tela de console que contem as configurações do XenServer como, rede, autenticação de usuários, máquinas virtuais e overview do ambiente configurado, a seguir na Figura 37.

```
xconsole
```

Figura 38 - Hypervisor XenServer instalado no orquestrador Apache CloudStack



Fonte: Elaborado pelo autor

Apêndice C - Instalação do *OpenFiler*

Na instalação do *OpenFiler* foram utilizados dois computadores sendo que um será o *storage* primário e outro *storage* secundário, ambos rodando versão *Openfiler* versão 2.99 linux CentOS 64 bits as configurações são iguais o que muda é o nome do compartilhamento.

OpenFiler

Storage primário.

Host: csgl04 IP: 192.168.30.4 e máscara de rede: 255.255.255.0

Storage secundário.

Host: csgl05 IP: 192.168.30.5 e máscara de rede: 255.255.255.0

Configuração das interfaces de rede externa e interna.

```
vi /etc/sysconfig/network-scripts/
vi ifcfg-eth0
DEVICE=eth1
BOOTPROTO=static
BROADCAST=192.168.30.255
HWADDR=00:53:44:E3:A8:2A
IPADDR=192.168.30.4
NETMASK=255.255.255.0
NETWORK=192.168.30.0
ONBOOT=yes
```

Definição do nome da máquina

```
hostname --fqdn
hostname
csgl04.localhost
```

Configuração dos nomes das máquinas utilizadas em rede

```
vi /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          csgl04.localhost csgl04 localhost.localdomain localhost csgl04
```

Apêndice D - Instalação do *XenCenter*

Para o gerenciamento e monitoração das máquinas virtuais e servidores físicos criados a partir dos computadores do *XenServers*, foi utilizado o aplicativo *XenCenter 6.5* da Citrix, sendo esta uma versão gratuita, não oferece vários recursos, dentre eles estão, tolerância a falhas e alta disponibilidade.

Utilização da plataforma de sistema operacional Windows Server 2008 64 bits.

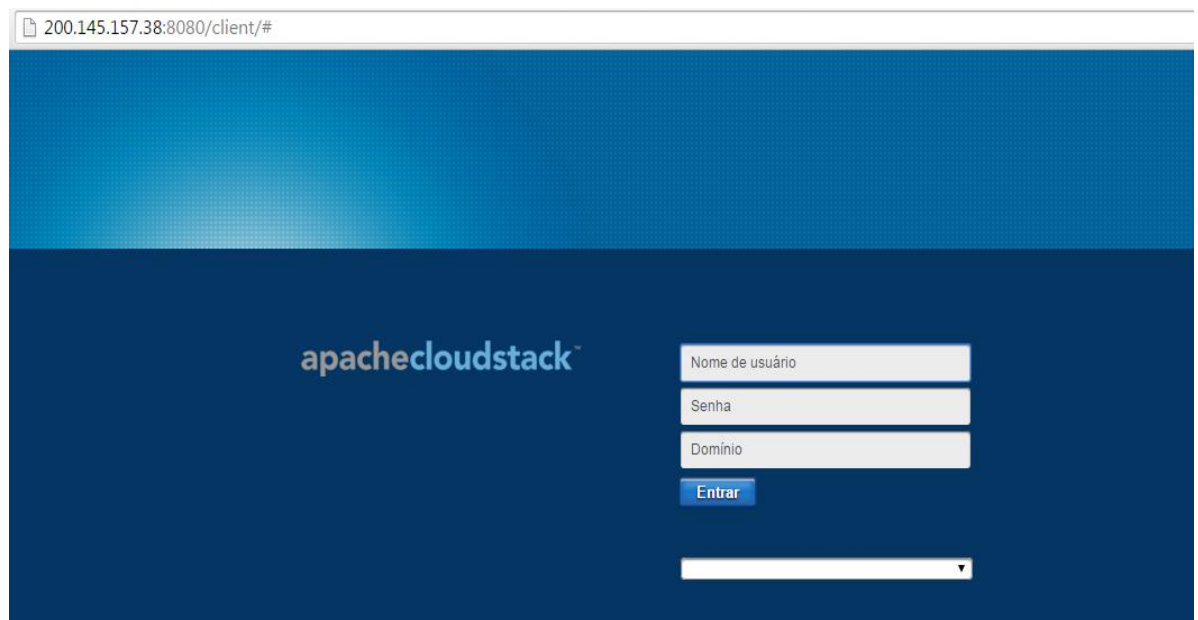
Computadores que foram utilizado no acolhimento ao aplicativo do *XenCenter*.

Host: csgl02IP: 192.168.30.2 e máscara de rede: 255.255.255.0

Host: csgl03IP: 192.168.30.3 e máscara de rede: 255.255.255.0

Após estas etapas concluídas foi realizado o acesso ao ambiente de nuvem do orquestrador *CloudStack*, como segue Figura 38.

Figura 39 - Interface web de gerenciamento do orquestrador Apache CloudStack



Fonte: Elaborado pelo autor.



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

TERMO DE REPRODUÇÃO XEROGRÁFICA

Autorizo a reprodução xerográfica do presente Trabalho de Conclusão, na íntegra ou em partes, para fins de pesquisa.

São José do Rio Preto, 06 / 12 / 2016

Assinatura do autor