

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
LABORATÓRIO DE REDES E GERÊNCIA**

Pedro Henrique Pereira Martins

**PROPOSTA DE UM FRAMEWORK AUTÔNOMO PARA
MONITORAMENTO DE AMBIENTES DE
COMPUTAÇÃO EM NUVEM**

Florianópolis

2015

1 INTRODUÇÃO

Computação em nuvem vem sendo utilizada devido ao seu modelo permitir um melhor aproveitamento dos recursos de hardware e software acarretando em redução de custos, melhor eficiência energética, elasticidade, flexibilidade e disponibilidade de recursos por demanda (ARMBRUST et al., 2010),(SINGH; SOOD; SHARMA, 2011),(FORUM, 2015).

Com o aumento da demanda por de serviços prestados pela nuvem os ambientes crescem constantemente para conseguir atender a demanda (SINGH; SOOD; SHARMA, 2011; WEINGÄRTNER; BRÄSCHER; WESTPHALL, 2015). O crescimento constante dos ambientes da nuvem acaba aumentando muito a complexidade de sua gerência tornando esta inviável de ser realizada por administradores (WEINGÄRTNER; BRÄSCHER; WESTPHALL, 2015),(HALL, 2012),(URGAONKAR; SHENOY; ROSCOE, 2002). Em (KEPHART; CHESS, 2003) é apresentado um modelo de computação autônoma como uma solução para lidar com grandes ambientes computacionais heterogêneos e complexos como ambientes de computação em nuvem.

Kephart (KEPHART; CHESS, 2003) define um elemento autônomo constituinte de 4 partes, Monitoramento, Análise, Planejamento e Execução. A idéia é que o elemento se monitore constantemente para verificar se ele realmente está fazendo aquilo que o foi designado como foi designado. Tudo que for monitorado será armazenado num log que será usado futuramente para possíveis adaptações de comportamento. O elemento também irá monitorar os recursos disponibilizados para o seu funcionamento, para garantir que ele está recebendo de fato o que foi acordado. Garantindo que seu serviço não seja interrompido por falta de recursos.

É necessário realizar o monitoramento constante da nuvem para ter ciência de quanto recurso foi usado por seus usuários e então poder estimar um valor a ser cobrado . O valor a ser cobrado pode ser acordado entre o usuário e o fornecedor de recursos da cloud ou pode já ser predefinido pelo fornecedor. O monitoramento de recursos também ajuda ao usuário ter uma noção de quanto ele está pagando por recurso, podendo então escolher um fornecedor com melhor custo (ACETO et al., 2012).

Além disso o monitoramento também pode ser usado para viabilizar técnicas de otimização e auxiliar numa melhor gerência destes ambientes. Para viabilizar tais otimizações é importante que o monitoramento forneça dados precisos e em tempo hábil sem afetar o fun-

cionamento da nuvem (ACETO et al., 2012) e (VIRATANAPANU et al., 2010)

O monitoramento em tempo real da nuvem é desafiador. Primeiro deve-se levar em conta o grande número de serviços e usuários, que torna a nuvem um ambiente com uma gama grande de informação para ser processada (SHAO et al., 2010), (CLAYMAN; GALIS; MAMATAS, 2010). Além disso tem-se a heterogeneidade dos componentes da nuvem, logo deve-se ter em mente que será necessário implementar meios para monitorar cada um desses componentes (SHAO et al., 2010).

O processo de monitoramento pode vir a sobrecarregar os recursos da nuvem se não for devidamente implantado, sendo este um dos principais desafios para o monitoramento dos recursos em nuvem (SHAO et al., 2010), (CLAYMAN; GALIS; MAMATAS, 2010). Além disso, após implantado o sistema de monitoramento, qualquer modificação necessária deve ser feita sem a necessidade de desligar ou reiniciar o ambiente (CLAYMAN; GALIS; MAMATAS, 2010).

Problemas aparentes de um monitoramento mau sucedido são:

1. A queda de desempenho da nuvem devido a sobrecargas de processamento com monitoramento;
2. Ações de gerência mal sucedidas devido a dados inconsistentes advindos do processo de monitoramento;
3. Perda de dados e/ou obtenção de dados inconsistentes durante o processo de atualização dos componentes de monitoramento.

Perda de dados causadas pelo fato de ter que desligar ou reiniciar a nuvem para poder implantar modificações feitas na ferramenta de monitoramento.

Atualmente as ferramentas de orquestração de computação em nuvem (levantadas durante o desenvolvimento deste trabalho) em sua maioria não realizam ou realizam mas não disponibilizam os dados resultantes do monitoramento dos recursos utilizados por usuários ou pela nuvem. O que estas ferramentas costumam adotar é o monitoramento momentâneo do ambiente, sem a criação de um histórico de uso de recursos, para indicar aos administradores o estado atual do ambiente.

O que existe atualmente para monitoramento são extensões que são adicionadas a essas ferramentas, porém estas extensões são de difícil instalação e não são completas pois não permitem o monitoramento de uma série de recursos físicos e algumas costumam apresentar dados inconsistentes com o ambiente, além de ser mais uma camada de software para gerenciar.

O Zenoss é um exemplo desse tipo de ferramenta de monitoramento que apresentam alguns problemas. No caso do Zenoss, e sua extensão para monitoramento do CloudStack, ele monitora os recursos relativos ao dom0 e não da nuvem como um todo. No caso, dom0 é apenas uma das várias Virtual Machines que podem estar presentes num cluster. Assim o Zenoss acaba fornecendo dados de apenas uma VM específica e não do ambiente inteiro.

Ao atuar nessa área, deve-se usar boas técnicas de manipulação de dados devido a grande gama de dados que podem ser monitorados na nuvem. Ser o menos invasivo possível, ou seja, ser invisível para os usuários da nuvem. Além disso, deve-se tratar a precisão das informações capturadas, os dados fornecidos devem condizer com o ambiente monitorado (ACETO et al., 2012). Quanto mais rápida as informações sobre o estado atual do ambiente forem disponibilizadas, mais eficientemente se conseguirá administrar a nuvem.

2 FUNDAMENTACAO

Neste capítulo será discutido sobre as ferramentas e conceitos em que este trabalho foi embasado.

2.1 COMPUTAÇÃO EM NUVEM

A computação em nuvem é um modelo ubíquo, conveniente e de acesso compartilhado a recursos computacionais como servidores, armazenamento, aplicações e redes (MELL; GRANCE, 2011). Estes recursos compartilhados podem ser rapidamente provisionados e liberados com um esforço mínimo de gestão ou interação com o provedor de serviços (MELL; GRANCE, 2011).

2.1.1 Características

Em (MELL; GRANCE, 2011), são definidas como características de um ambiente de computação em nuvem:

1. Acesso aos recursos sob demanda - o usuário pode escolher quais serviços/componentes ele irá usar dentro do que é disponibilizado pela nuvem sem a necessidade de interação com administradores;
2. Disponibilidade - a nuvem deve estar sempre disponível para o usuário pela internet independente da plataforma de acesso (tablet, smartphone, desktop etc);
3. Recursos compartilhados - os recursos físicos da nuvem devem poder atender diferentes usuários simultaneamente;
4. Elasticidade - conforme um usuário solicita mais recursos, a nuvem deve alocar mais recursos para este usuário evitando a degradação dos serviços prestados, assim como se um usuário deixar de usar recursos, a nuvem pode desalocar recursos excedentes;
5. Monitoramento de serviços - a nuvem deve ser capaz de monitorar o uso de recursos por cada serviço, proporcionando transparência aos fornecedores e usuários do serviço.

2.1.2 Modelos de serviço

(MELL; GRANCE, 2011) descreve os modelos de serviço disponibilizados pela nuvem como sendo:

1. Software como um serviço: a nuvem fornece uma aplicação final via rede para usuários usarem. O usuário não tem controle algum sobre a infraestrutura que hospeda a aplicação ou da plataforma que a aplicação usa.
2. Plataforma como um serviço: a nuvem disponibiliza diferentes plataformas para o usuário desenvolver e hospedar suas próprias aplicações. Neste nível o usuário tem controle total das aplicações, e pode configurar por completo a plataforma.
3. Infraestrutura como um serviço: a nuvem fornece toda a infraestrutura para o usuário, como processador, disco, memória e rede. Neste nível o usuário tem uma máquina virtual pronta para uso, ele define quais plataformas ele irá usar e quais aplicações ele irá rodar em cada plataforma, porém a parte de hardware ele não tem controle, endereços físicos de memória e algumas instruções do processador não serão acessíveis ao usuário.

2.1.3 Tipos de nuvens

Em (MELL; GRANCE, 2011) são definidos quatro tipos de nuvem quanto a disponibilização dos recursos para usuários:

1. Nuvem privada : a infraestrutura da nuvem é disponibilizada apenas para uma única organização utilizar. A nuvem pode ser gerenciada pela própria organização, por um terceiro ou por ambos.
2. Nuvem comunitárias : a infraestrutura da nuvem é disponibilizada para uma comunidade de organizações que tem interesses em comum. A nuvem pode ser gerenciada por alguma(s) dessas organizações, terceiros ou ambos.
3. Nuvem pública : a nuvem é aberta para todo o público. A é gerenciada por alguma empresa, organização governamental/acadêmica ou alguma combinação destas.

4. Nuvem híbrida : a infraestrutura da nuvem é um composto de duas ou mais infraestruturas (privada, comunitária, pública).

2.2 VIRTUALIZAÇÃO

A virtualização é um processo antigo que foi primeiramente incorporado pela IBM com intuito de particionar um *mainframe* com alta capacidade, em *mainframes* virtuais com menor capacidade (SAHOO; MOHAPATRA; LATH, 2010). O processo de virtualização consiste na existência de um *VirtualMachineMonitor*(VMM) que gerencia Sistemas Operacionais convidados e intermedia toda a comunicação entre Hardware e esses sistemas (MENASCÉ, 2005). Estes Sistemas Operacionais convidados rodam encima do VMM como se fossem aplicações em um Sistema Operacional qualquer. Portanto os Sistemas Operacionais convidados, não possuem acesso direto ao Hardware. Com o uso da virtualização, é possível que os mesmos componentes de hardware suportem diferentes sistemas operacionais como sendo diversas unidades lógicas. Estas unidades lógicas em que o hardware é dividido são nomeadas de *VirtualMachine* (VM). Os sistemas operacionais e aplicações enchem esses ambientes virtuais como reais. Para o usuário final, ter uma máquina virtual ou uma máquina física não irá mudar sua experiência de uso do computador (SAHOO; MOHAPATRA; LATH, 2010).

2.2.1 Tipos de virtualização

Nesta seção, serão discutidos os tipos de virtualizações segundo (SAHOO; MOHAPATRA; LATH, 2010), e suas principais características.

1. Virtualização completa: o VMM atua como o gerente das máquinas virtuais e roda encima de um outro Sistema Operacional que irá fazer a comunicação com o hardware. Toda a comunicação que as máquinas virtuais irão fazer com o hardware será intermediada pelo VMM. Todos os endereços de entrada e saída de dados serão mapeados para endereços lógicos disponibilizados pelo VMM . Este tipo de virtualização é complexa e custosa de se fazer devido as camadas de software extras, porém é a mais versátil.
2. Virtualização da camada do Sistema Operacional: este tipo de virtualização permite a criação de várias instâncias do mesmo sistema operacional totalmente independentes rodando em paralelo. Quando comparado com outras técnicas de virtualização,

esta tende a ser mais eficiente que a anterior. Neste tipo de virtualização, os clientes estão presos a criarem VMs exclusivamente do mesmo tipo de sistema operacional do VMM.

3. Virtualização da camada de Hardware: esta abordagem é comumente usado no mercado de servidores devido ao seu elevado isolamento de máquinas virtuais e seu grande desempenho. Neste cenário o hypervisor roda diretamente encima do hardware e controla diretamente o acesso dos sistemas operacionais das VMs aos recursos de hardware.

2.2.2 Vantagens e Desvantagens da virtualização

A virtualização permite um melhor aproveitamento dos recursos de hardware disponíveis. Permite a alteração de peças de hardware sem a necessidade de parar o funcionamento das VMs por longos períodos, basta migrar a VM para outro servidor físico. O custo de ter um máquina virtual é menor do que o custo para ter uma máquina física com a mesma configuração (SAHOO; MOHAPATRA; LATH, 2010). Como problemas temos uma grande sobrecarga de processamento para gerenciar a inúmera quantidade de máquinas virtuais. O desenvolvimento de virtualizadores não é algo simples e é de inúmera importância ter redundância de hardwares pois se um componente de hardware der defeito, todas as VMs vinculadas a tal componente serão penalizadas (POPEK; GOLDBERG, 1974) .

2.2.3 Virtualizadores

Neste capítulo será discutido sobre as ferramentas e conceitos em que este trabalho foi embasado.

XenServer (SYSTEMS, 2015b) : é um virtualizador do tipo 3 de código aberto que da suporte as arquiteturas x86 (WIKIPEDIA, 2015b) x86 64 (WIKIPEDIA, 2015c), IA32 (CORPORATION, 2015), IA64 (DOSHI, 2015) e PowerPC (WIKIPEDIA, 2015a). O Xen foi adquirido pela empresa Citrix (SYSTEMS, 2015a) que abriu o código do Xen, além de fornecer suporte aos usuários.

KVM (LINUX-KVM, 2015) : o KVM assim como o Xen, é um virtualizador do tipo 3. Suportado pela Red Hat (.INC, 2015), o KVM é um virtualizador baseado em linux. O KVM suporta as mesmas arquiteturas do Xen.

ESXi (VMWARE, 2015a) : este virtualizador foi desenvolvido pela VMware (VMWARE, 2015b) Tem suporte apenas para arquiteturas x86 de 32 e 64 bits. É um virtualizador do tipo 1, consequentemente acaba se tornando menos eficiente que o Xen e KVM.

Tabela 1 – Comparação entre virtualizadores analisados

	Empresa Responsável	Arquiteturas Suportadas	Tipo de virtualização	Tipo de Licença
XenServer	Citrix Systems	x86, x86-64, IA 64, PowerPC	3	Pública
KVM	Red Hat	x86, x86-64, IA 64, PowerPC	3	Pública
ESXi	VMware	x86, x86-64	1	Proprietária

Dentre os virtualizadores analisados, será usado neste trabalho o XenServer devido a ser um virtualizador de código aberto e suportado por uma empresa. O fato do Xen ter seu código aberto, possibilita que toda uma comunidade interaja em seu código e possa contribuir com melhorias tanto na documentação quanto em correções de bugs, além é claro do suporte fornecido pela empresa.

2.3 MONITORAMENTO EM CLOUD

Nesta seção, serão abordadas as dificuldades relacionadas ao monitoramento na nuvem, sua importância, as características que um monitoramento deve possuir e algumas ferramentas que realizam monitoramento na nuvem.

2.3.1 Características de ferramentas de monitoramento na nuvem

Segundo (ACETO et al., 2012) existem 10 características que uma ferramenta de monitoramento na nuvem deve possuir para ter um funcionamento correto e sem degradar o funcionamento do ambiente em que ela foi instalada:

1. Escalabilidade: a ferramenta pode ser considerada escalável se ela consegue suportar uma larga escala de objetos a serem monitorados (CLAYMAN; GALIS; MAMATAS, 2010). Na nuvem, tal característica tem importância devida a grande gama de componentes que uma nuvem pode possuir;
2. Elasticidade: para uma ferramenta ser considerada elástica, ela deve conseguir suportar mudanças ocorridas no ambiente moni-

torado, tais como novos objetos para serem monitorados ou objetos deixando de ser monitorados (CLAYMAN; GALIS; MAMATAS, 2010). Isto é de fundamental importância na nuvem, pois o ambiente vive em contínua mudança com a adição e remoção de recursos dinamicamente;

3. Adaptabilidade: uma ferramenta de monitoramento é dita adaptável, se ela consegue se ajustar a mudanças de cargas impostas ao ambiente, sem que este acabe sendo penalizado e mantendo o seu serviço de monitoramento ativo e funcionando corretamente (CLAYMAN; GALIS; MAMATAS, 2010). Para realizar o monitoramento do ambiente, a ferramenta consome recursos de rede, armazenamento e processamento do ambiente. A ferramenta deve se ajustar para consumir menos recursos em ocasiões que o ambiente se encontra sobre carregado com suas próprias atividades;
4. Pontualidade: a ferramenta pode ser considerada pontual, se ela é capaz de relatar eventos que acontecem no ambiente, como perda repentina de um recurso, ou sobrecarga de outro recurso, em tempo hábil para estes serem tratados pelo administrador do ambiente (WANG et al., 2011). Esta característica depende diretamente da boa implantação das outras características da ferramenta;
5. Autonomia: para a ferramenta ser considerada autônoma, ela deve ser capaz de se gerenciar, conseguindo responder a mudanças repentinas do ambiente sem a necessidade de uma outra entidade a supervisionando. Ao mesmo tempo em que torna a complexidade do seu funcionamento discreta para usuários e gerentes do ambiente;
6. Abrangente: uma ferramenta de monitoramento é dita abrangente, se ela consegue monitorar diferentes tipos de recursos e capturar diferentes métricas destes recursos (HASSELMEYER; D'HEUREUSE, 2010) Isto é importante, já que ambientes de nuvem costumam ser heterogêneos;
7. Extensividade: a ferramenta pode ser considerada extensível, se ela suporta a adição de novas métricas para monitorar, sem a necessidade de modificações no código fonte da ferramenta. Não intrusiva: para a ferramenta não ser considerada intrusiva, ela deve poder ser implantada no ambiente sem precisar de modificações significativas deste ambiente (KATSAROS; KUBERT; GALIZO, 2011);

8. Confiabilidade: uma ferramenta de monitoramento é dita confiável, se mesmo com falhas em diversos componentes, ela ainda continua fornecendo seu serviço normalmente com os componentes que ainda lhe restam (LAPRIE, 2008);
9. Disponibilidade: a ferramenta de monitoramento deve estar disponível sempre que os seus serviços forem solicitados. Isto é importante para ambientes da nuvem, pois a nuvem está em constante atividade (SHIREY, 2007);
10. Precisão: para a ferramenta ser considerada precisa, as informações disponibilizadas pela ferramenta, devem ser precisas. Estas informações disponibilizadas pela ferramenta devem conter valores que expressem o mais aproximadamente possível o estado atual em que o objeto monitorado se encontra no momento que a medida foi realizada.

2.3.2 Dificuldades encontradas ao monitorar uma nuvem

Nesta seção, serão discutidos alguns desafios existentes no monitoramento de um ambiente de computação em nuvem definidos em (ACETO et al., 2012).

2.3.2.1 Grande gama de informação

A nuvem possui diversos usuários, assim como componentes e um grande tráfego de rede. Com isso, armazenar informações colhidas na nuvem é uma tarefa nada fácil. Algumas horas de monitoramento contínuo, podem gerar Tera Bytes de informações para serem armazenadas.

2.3.2.2 Mudança constante do ambiente

Devida a elasticidade de um ambiente de computação em nuvem, as métricas a serem monitoradas podem mudar a qualquer momento, pode haver a necessidade de monitorar um novo elemento que não estava previsto.

2.3.2.3 Sensibilidade de ambientes carregados

Em ambientes de computação em nuvem muito carregados, a presença do monitoramento pode acarretar em sobre carga de processamento, e acabar comprometendo o funcionamento de todo o ambiente.

2.3.2.4 Garantir autonomia da ferramenta

Esta tarefa não é nada trivial, ela demanda a possível existência de um laço que irá receber informações do ambiente de tempos em tempos (a própria ferramenta que deve obter essas informações), processar essas informações e repassar as ações que cada nodo deve executar (no caso de uma ferramenta distribuída). Realizar este processo tentando gerar o menor impacto possível no ambiente é o grande desafio.

2.3.2.5 Heterogeneidade do ambiente

A nuvem é um ambiente que possui inúmeros dispositivos com diferentes arquiteturas e tecnologias. Conseguir prover suporte de monitoramento para cada um desses dispositivos e ainda manter um bom isolamento entre eles, não é algo fácil.

2.3.2.6 Rastreamento de elementos da nuvem

A nuvem permite que seus recursos sejam migrados de um dispositivo físico para outro. Garantir que a ferramenta consiga detectar a nova localização de um recurso migrado e continuar monitorando este recurso em sua nova localização, em vez de apenas acusar uma falta do recurso migrado e a detecção de num novo recurso, é algo complicado.

2.3.3 Importância do monitoramento para nuvem

O monitoramento na nuvem é de grande importância tanto para os clientes quanto para os fornecedores de serviços na nuvem. O monitoramento é um meio para se garantir um melhor controle dos dispositivos de hardware e das aplicações que estão rodando na nuvem (ACETO et al., 2012). O constante monitoramento do ambiente, auxi-

lia na garantia da qualidade do serviço prestado para os usuários do ambiente, como por exemplo: disponibilidade e tempo de resposta das aplicações hospedadas no ambiente. Para os fornecedores de serviços, o monitoramento é um jeito de mensurar os recursos utilizados pelo usuário e então poder ser cobrada uma taxa referente ao uso dos mesmos (ACETO et al., 2012). Para ambos os casos, o monitoramento auxilia na prevenção e até mesmo correção de desvios no termo de prestação de serviços acordado por usuário e provedor (ACETO et al., 2012).

2.3.4 Ferramentas de monitoramento

Existem uma série de ferramentas de monitoramento para ambientes de computação em nuvem, nesta parte serão mostradas algumas destas ferramentas:

- Nagios: o Nagios (NAGIOS, 2015) suporta o monitoramento de máquinas virtuais e serviços de armazenamento (CARON et al., 2012). O Nagios é uma ferramenta de código aberto que possui inúmeras extensões. O que permite que o Nagios consiga monitorar sistemas do tipo Eucalyptus (Elastic Utility Computing Architecture for Linking Your Programs To Useful System) (NAGIOS, 2015). Nagios é uma das ferramentas usadas para prover o monitoramento do OpenStack (OPENSTACK, 2015), que é uma ferramenta de orquestração de infra-estruturas de computação em nuvem. A principal proposta do Nagios é a extensibilidade.
- CloudStack ZenPack: o CloudStack (APACHE, 2015b), é uma ferramenta de orquestração de infra-estrutura de ambientes de computação em nuvem. É uma ferramenta opensource escrita em java com suporte a diversos virtualizadores, como por exemplo, KVM, XenServer e VMware. Usuários do CloudStack, tem como opção para o monitoramento do ambiente, uma extensão (ZENPACK, 2015), disponível no ZenPack. Esta ferramenta possui algumas inconsistências nos valores de memória disponibilizados. Os valores dão a entender que são do ambiente como um todo, quando na verdade, são relativos a apenas uma máquina virtual (Dom0). Como principal característica dessa ferramenta, se encontra a pontualidade
- Nimbus: a Nimbus (PROJECT, 2015) é uma plataforma de código aberto que da suporte a diferentes infra-estruturas de nuvens.

Esta ferramenta é voltada especialmente para projetos relacionados a comunidade científica. A Nimbus, visa autonomia no monitoramento tanto de aplicações da nuvem quanto na infraestrutura como um todo, no caso, ela não suporta monitoramento de clusters em específico, apesar de monitorar aplicações rodando em VMs.

2.4 ORQUESTRAÇÃO DA NUVEM

Computação em nuvem é algo complexo devido aos seus inúmeros componentes heterogêneos. Estes componentes devem ser gerenciados de forma a conseguir trabalhar em conjunto mesmo com as suas diferentes arquiteturas. Realizar isso num ambiente vasto como a nuvem com uma grande quantidade destes componentes, acaba se tornando algo complexo para um administrador humano (WEINGÄRTNER; BRÄSCHER; WESTPHALL, 2015).

As ferramentas de orquestração na nuvem servem para abstrair do administrador toda a parte de interação com os virtualizadores, serviços de armazenamento, rede e outros. Permitindo que o administrador não precise de amplos conhecimentos nas tecnologias de virtualização, rede, armazenamento e outras utilizadas.

Essas abstrações proporcionadas pelas ferramentas de orquestração tornam possível que o administrador da nuvem possa delimitar recursos físicos para cada usuário facilmente. Assim como também possibilita que usuários sem conhecimento nenhum de computação possam criar suas próprias máquinas virtuais facilmente.

2.4.1 Ferramentas de orquestração

Foram pesquisadas algumas ferramentas de orquestração em nuvem, tais como:

- CloudStack: o cloudStack (APACHE, 2015b) é uma ferramenta de orquestração de código aberto, suportada pela fundação Apache (APACHE, 2015a) e desenvolvida em java. O cloudStack dá suporte para virtualizadores como KVM (LINUX-KVM, 2015), VMware (VMWARE, 2015b), XenServer (SYSTEMS, 2015b) e Hyper-V (MICROSOFT, 2015). Esta ferramenta dá suporte aos sistemas de arquivos iSCSI (INC, 2015) e NFS (WIKIPEDIA, 2015).
- Eucalyptus: o Eucalyptus (HP, 2015a), assim como o CloudStack,

é uma ferramenta de orquestração de ambientes de computação em nuvem de código aberto. Suportada pela HP (HP, 2015b) e desenvolvida em java e C. O Eucalyptus, usa o Walrus como sistema de arquivo e dá suporte aos virtualizadores KVM, XenServer e VMware.

- OpenStack: O OpenStack (OPENSTACK, 2015), dá suporte aos mesmos virtualizadores que o CloudStack, é uma ferramenta de código aberto desenvolvida em python. Suporta os sistemas de arquivo Swift e Cinder.

Tabela 2 – Comparação entre ferramentas de orquestração analisadas

	Linguagem Suportada	Sistema de Arquivos	Tipo de Licença
CloudStack	Java	iSCSI , NFS	Código aberto
OpenStack	Python	Swift , Cinder	Código aberto
Eucalyptus	Java , C	Walrus	Código aberto

Dentre essas ferramentas de orquestração em nuvem, foi optado o uso do CloudStack devida a uma série de motivos, tais como:

- Suportado por uma fundação e uma comunidade amplamente ativa.
- Possui uma documentação bem descrita.
- É codificada em java, que é uma linguagem de programação bem conhecida para futuras extensões.
- Dá suporte para sistemas de arquivos conhecidos, o que diminui o tempo que seria gasto com familiarizações com sistemas de arquivos.
- É uma ferramenta de código aberto, o que possibilita um melhor conhecimento da ferramenta de implantação de melhorias.

REFERÊNCIAS

ACETO, G. et al. Cloud monitoring: Definitions, issues and future directions. **CLOUDNET**, v. 12, p. 63–67, 2012.

APACHE, F. **Apache**. 2015. Disponível em: <<http://apache.org/>>.

APACHE, f. **CloudStack**. 2015. Disponível em: <<http://cloudstack.apache.org/>>.

ARMBRUST, M. et al. A view of cloud computing. **Communications of the ACM**, ACM, v. 53, n. 4, p. 50–58, 2010.

CARON, E. et al. Auto-scaling, load balancing and monitoring in commercial and open-source clouds. 2012.

CLAYMAN, S.; GALIS, A.; MAMATAS, L. Monitoring virtual networks with lattice. In: IEEE. **Network Operations and Management Symposium Workshops (NOMS Wksp), 2010 IEEE/IFIP**. [S.l.], 2010. p. 239–246.

CORPORATION, I. I. **IA 32**. 2015. Disponível em: <<http://www.intel.com/content/dam/www/public/us/en/documents/manuals/ia-32-architectures-software-developer-manual-325462.pdf>>.

DOSHI, G. **IA 64**. 2015. Disponível em: <<http://www.csee.umbc.edu/portal/help/architecture/idfisa.pdf>>.

FORUM, C. I. **UK Cloud Adoption Trends for 2015 InsightBrief**. 2015. Disponível em: <<http://cloudindustryforum.org/downloads/whitepapers/UK-Cloud-Adoption-Trends-for-2015-InsightBrief.pdf>>.

HALL, P. Opportunities for csps in enterprise-grade public cloud computing. **OVUM**, May, 2012.

HASSELMEYER, P.; D'HEUREUSE, N. Towards holistic multi-tenant monitoring for virtual data centers. In: IEEE. **Network Operations and Management Symposium Workshops (NOMS Wksp), 2010 IEEE/IFIP**. [S.l.], 2010. p. 350–356.

HP, I. **Eucalyptus**. 2015. Disponível em: <<http://www8.hp.com/us/en/cloud/helion-eucalyptus-overview.html>>.

HP, I. **HP**. 2015. Disponível em: <<http://www.hp.com/>>.

INC, M. **iSCSI**. 2015. Disponível em:
<<http://windows.microsoft.com/pt-br/windows-vista/what-is-internet-small-computer-system-interface-iscsi>>.

.INC, R. **RedHatVirtualization**. 2015. Disponível em:
<<http://www.redhat.com/en/technologies/virtualization/enterprise-virtualization>>.

KATSAROS, G.; KUBERT, R.; GALLIZO, G. Building a service-oriented monitoring framework with rest and nagios. In: **IEEE. Services Computing (SCC), 2011 IEEE International Conference on**. [S.l.], 2011. p. 426–431.

KEPHART, J. O.; CHESS, D. M. The vision of autonomic computing. **Computer**, IEEE, v. 36, n. 1, p. 41–50, 2003.

LAPRIE, J.-C. From dependability to resilience. In: CITESEER. **38th IEEE/IFIP Int. Conf. On Dependable Systems and Networks**. [S.l.], 2008. p. G8–G9.

LINUX-KVM. **KVM**. 2015. Disponível em:
<http://www.linux-kvm.org/page/Main_Page>.

MELL, P.; GRANCE, T. The nist definition of cloud computing. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg, 2011.

MENASCÉ, D. A. Virtualization: Concepts, applications, and performance modeling. In: **Int. CMG Conference**. [S.l.: s.n.], 2005. p. 407–414.

MICROSOFT, I. **Hyper-V**. 2015. Disponível em:
<<http://hyperv.veeam.com/what-is-hyper-v-technology/>>.

NAGIOS. **Nagios**. 2015. Disponível em:
<<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore-3-en.pdf>>.

OPENSTACK, I. **OpenStack**. 2015. Disponível em:
<<http://docs.openstack.org/index.html>>.

POPEK, G. J.; GOLDBERG, R. P. Formal requirements for virtualizable third generation architectures. **Communications of the ACM**, ACM, v. 17, n. 7, p. 412–421, 1974.

PROJECT, N. **Nimbus**. 2015. Disponível em:
<<http://www.nimbusproject.org/>>.

SAHOO, J.; MOHAPATRA, S.; LATH, R. Virtualization: A survey on concepts, taxonomy and associated security issues. In: IEEE. **Computer and Network Technology (ICCNT), 2010 Second International Conference on**. [S.l.], 2010. p. 222–226.

SHAO, J. et al. A runtime model based monitoring approach for cloud. In: IEEE. **Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on**. [S.l.], 2010. p. 313–320.

SHIREY, R. W. Internet security glossary, version 2. 2007.

SINGH, G.; SOOD, S.; SHARMA, A. Cm-measurement facets for cloud performance. **International Journal of Computer Applications**, International Journal of Computer Applications, 244 5th Avenue, # 1526, New York, NY 10001, USA India, v. 23, n. 3, p. 37–42, 2011.

SYSTEMS, I. C. **Citrix**. 2015. Disponível em:
<<http://www.citrix.com/>>.

SYSTEMS, I. C. **Xen Project Software Overview**. 2015.
Disponível em: <http://wiki.xenproject.org/wiki/Xen_Overview>.

URGAONKAR, B.; SHENOY, P.; ROSCOE, T. Resource overbooking and application profiling in shared hosting platforms. **ACM SIGOPS Operating Systems Review**, ACM, v. 36, n. SI, p. 239–254, 2002.

VIRATANAPANU, A. et al. On demand fine grain resource monitoring system for server consolidation. In: IEEE. **Kaleidoscope: Beyond the Internet?-Innovations for Future Networks and Services, 2010 ITU-T**. [S.l.], 2010. p. 1–8.

VMWARE. **ESXI**. 2015. Disponível em:
<<http://www.vmware.com/products/esxi-and-esx/>>.

VMWARE. **VMware**. 2015. Disponível em:
<<http://www.vmware.com>>.

WANG, C. et al. A flexible architecture integrating monitoring and analytics for managing large-scale data centers. In: ACM. **Proceedings of the 8th ACM international conference on Autonomic computing**. [S.l.], 2011. p. 141–150.

WEINGÄRTNER, R.; BRÄSCHER, G. B.; WESTPHALL, C. B. Cloud resource management: A survey on forecasting and profiling models. **Journal of Network and Computer Applications**, Elsevier, v. 47, p. 99–106, 2015.

WIKIPEDIA. **NFS**. 2015. Disponível em:
<https://en.wikipedia.org/wiki/Network_File_System>.

WIKIPEDIA, I. **Power PC**. 2015. Disponível em:
<<https://en.wikipedia.org/wiki/X86-64>>.

WIKIPEDIA, I. **x86**. 2015. Disponível em:
<<https://en.wikipedia.org/wiki/X86>>.

WIKIPEDIA, I. **x86_64**. 2015. Disponível em:
<<https://en.wikipedia.org/wiki/X8664>>.

ZENPACK. **Zenoss**. 2015. Disponível em:
<<https://github.com/zenoss/ZenPacks.zenoss.CloudStack>>.