

Eucalyptus: uma plataforma de *Cloud Computing* para qualquer tipo de usuário

Gustavo Henrique Rodrigues Pinto Tomas

*Universidade Federal de São Carlos, *Campus* Sorocaba

Sorocaba, São Paulo

Email: gustavoh_rodrigues@hotmail.com

Resumo—A *Cloud Computing* esta se tornando um dos principais paradigmas da Computação e, cada vez mais, é utilizado em organizações que necessitam de uma grande infraestrutura de Tecnologia da Informação. Neste cenário, surge uma lacuna deixada pelas organizações que possuem informações sigilosas e necessitam implementar suas próprias políticas de gerenciamento de recursos.

Visando preencher essa lacuna e algumas outras metas da computação em distribuída em larga-escala, a plataforma Eucalyptus foi desenvolvida. O Eucalyptus possui duas versões: *Eucalyptus Enterprise Edition* e *Eucalyptus Open-Source*. Na versão *Enterprise*, as organizações tem a possibilidade de implementar Nuvens privadas com suas próprias políticas, de forma bastante intuitiva utilizando todo o aparato desenvolvido pela plataforma Eucalyptus. Já na versão *Open-Source*, que também conta com todo aparato da plataforma, qualquer configuração pode ser visualizada por toda comunidade, além de uma série de restrições que são impostas para garantir a disponibilidade da plataforma.

Index Terms—*Cloud Computing*, Nuvem privada, Extensibilidade

I. INTRODUÇÃO

A tecnologia esta em constante evolução, sempre lançando novos paradigmas e trazendo diversos novos benefícios, como por exemplo, a *Cloud Computing* (Computação em Nuvem). Segundo Taurion, o termo computação em nuvem surgiu em 2006 em uma palestra de Eric Schmidt, da Google, sobre como sua empresa gerenciava seus *data centers* (local onde são concentrados os computadores e sistemas responsáveis pelo processamento de dados de uma empresa ou organização). Hoje, *Cloud Computing* se apresenta como o cerne de um movimento de profundas transformações do mundo da tecnologia [1].

A *Cloud Computing* está se tornando um paradigma chave da indústria de Tecnologia da Informação. A nuvem é uma representação para a Internet e sua infraestrutura de comunicação entre os componentes arquiteturais. Cada parte desta infraestrutura é provida como um serviço e estes são normalmente alocados em *data centers*, utilizando *hardware* compartilhado para computação e armazenamento [2].

A computação em nuvem é um novo modelo de computação emergente que move todos os dados e as aplicações dos usuários para grandes centros de armazenamento. Com isso, as aplicações e os sistemas de *hardware* são distribuídos na forma de serviços baseados na Internet. Fundamentada em conceitos já estabelecidos previamente, como a virtualização e o modelo *pay-per-use* (modelo de pagamento baseado no

uso, semelhante aos serviços de telefonia e energia elétrica), a computação em nuvem possui uma série de vantagens, como a possibilidade de ampliar os recursos utilizados sempre que necessário [3]. Logo, pode-se dizer que a computação em nuvem é uma evolução dos serviços e produtos de tecnologia da informação sob demanda, também chamada de *Utility Computing*.

De acordo com [4] e [3], existem alguns modelos de implementação de *Cloud Computing* que devem ser escolhidos de acordo com a necessidade das aplicações que serão implementadas: Privado, Público, Comunidade e Híbrido. A restrição ou abertura de acesso destes modelos depende do processo de negócios, do tipo de informação e do nível de visão desejado.

- **Privado:** As nuvens privadas são aquelas construídas exclusivamente para um único usuário (uma empresa, por exemplo), na qual ele possui total controle sobre como as aplicações são implementadas na nuvem. Uma nuvem privada é, em geral, construída sobre um *data center* privado [3].
- **Público:** As nuvens públicas são aquelas que são executadas por terceiros. As aplicações de diversos usuários ficam misturadas nos sistemas de armazenamento, o que pode parecer ineficiente a princípio. Porém, se a implementação de uma nuvem pública considera questões fundamentais, como desempenho e segurança, a existência de outras aplicações sendo executadas na mesma nuvem permanece transparente tanto para os prestadores de serviços como para os usuários [3].
- **Comunidade:** A infraestrutura de nuvem é compartilhada por diversas organizações e suporta uma comunidade específica que partilha as preocupações (por exemplo, a missão, os requisitos de segurança, política e considerações sobre o cumprimento). Pode ser administrado por organizações ou por um terceiro e pode existir localmente ou remotamente [4].
- **Híbrido:** Nas nuvens híbridas temos uma composição dos modelos de nuvens públicas e privadas. Elas permitem que uma nuvem privada possa ter seus recursos ampliados a partir de uma reserva de recursos em uma nuvem pública. Essa característica possui a vantagem de manter os níveis de serviço mesmo que haja flutuações rápidas na necessidade dos recursos. A conexão entre as

nuvens pública e privada pode ser usada até mesmo em tarefas periódicas que são mais facilmente implementadas nas nuvens públicas, por exemplo. O termo “computação em ondas” é, em geral, utilizado quando se refere às nuvens híbridas [3].

Neste cenário, a plataforma Eucalyptus implementa a *cloud computing* baseado num modelo privado e público. Além disso, o Eucalyptus implementa o modelo de serviço de *Infrastructure as a Service*(IaaS) que inclui os serviços oferecidos na camada de infraestrutura (servidores, roteadores, sistemas de armazenamento e outros recursos de computação). Também é responsável por prover toda a infraestrutura necessária para a *Software as a Service*(SaaS) e o *Platform as a Service*(PaaS).

Este artigo tem como objetivo explorar algumas características primordiais da plataforma Eucalyptus que vem se destacando no ambiente *cloud computing* devido, principalmente, a grande demanda dos governos e organizações por uma versão de *cloud on-premise* em um *data center* privado [5], pois estes necessitam implementar suas próprias políticas locais de gerenciamento de recursos [6]. Na próxima seção é realizado um *overview* da plataforma Eucalyptus, incluindo sua história e suas metas principais. Em seguida, é descrita a arquitetura da plataforma Eucalyptus e todos os seus principais componentes. Na seção seguinte são descritos alguns experimentos que estabelecem um quadro comparativo da plataforma Eucalyptus com *Amazon AWS*. Por fim são demonstrados alguns casos de instituições que utilizam a plataforma Eucalyptus.

II. EUCALYPTUS

O Projeto Eucalyptus (*Elastic Utility Computing Architecture Linkin Your Programs To Useful Systems*) começou na *Computer Science Department at the University of California, Santa Barbara* com o pesquisador Rich Wolski e seu time, com o propósito de investigar a área de *High Performance Computing* (HPC), mais especificamente a partir do projeto *Virtual Grid Application Development Software Project* (VGrADS) financiado pela *National Science Foundation*. Para concluir os testes finais do VGrADS em supercomputadores, foi escolhida a nuvem pública da Amazon. No entanto, VGrADS sempre foi um projeto conjunto entre Universidade e Laboratórios de Pesquisa com dados privados e era necessária uma investigação local mais detalhada sobre os comportamentos de diferentes aplicações. A partir de então, no início de fevereiro de 2008, iniciou-se o desenvolvimento da plataforma *open-source* Eucalyptus. A primeira versão foi lançada em 29 de maio de 2008 com suporte apenas a EC2 e em dezembro de 2008 foi incluída a interface com o serviço S3. Em 2009, a equipe Eucalyptus criou a companhia *Eucalyptus Systems Inc.* para comercializar o *Eucalyptus Enterprise*.

O Eucalyptus é uma infraestrutura de software de código aberto que utiliza diversos recursos computacionais disponível aos pesquisadores, como *clusters* e *workstation farms*. A fim de promover a exploração da comunidade científica de sistemas de computação em nuvem, o Eucalyptus foi desenvolvido totalmente modularizado, permitindo aos

pesquisadores realizarem experimentos de segurança, escalabilidade, confiabilidade e implementação de interfaces. [7].

O desenvolvimento do Eucalyptus foi guiado para atingir duas metas principais:

- Ser capaz de ser implantado e executado em ambientes de *hardware* e *software* que não estão sob o controle de seus criadores;
- Extensibilidade: deve ser modularizado, para permitir a fácil manutenção de um componente ou até mesmo sua substituição.

A extensibilidade é garantida ao construir uma arquitetura utilizando *Web Services*, na qual as interfaces são bem definidas utilizando documentos WSDL e a autenticação é feita via mecanismos *WS-Security*. Ao mesmo tempo, o Eucalyptus suporta uma interface com a nuvem popular da Amazon AWS. O Eucalyptus é compatível com várias distribuições de Linux, incluindo Ubuntu, *Red Hat Enterprise Linux* (RHEL), CentOS, *SUSE Linux Enterprise Server* (SLES), openSUSE, Debian e Fedora e com uma variedade de tecnologias de virtualização, como VMware, Xen e *KVM hypervisors*.

Além disso, no Eucalyptus foi desenvolvido soluções de tecnologia de nível empresarial que rodam em cima do núcleo do software de código aberto. Desta forma é possível fornecer recursos como personalização para o usuário final, suporte a aplicativos legados, *Service-level agreements*(SLA) personalizadas, monitoramento de nuvens e suporte ao dimensionamento automático.

Como o Eucalyptus implementa o modelo de serviço de IaaS, então, indubitavelmente, ele compartilha algumas características relacionadas à infraestrutura, como uma interface única para administração, uma *Application Programming Interface*(API) para interação com outros *hosts*, *switches*, roteadores e o suporte para adicionar novos equipamentos de forma simples e transparente.

O Eucalyptus possui duas versões: *Eucalyptus Enterprise Edition* e *Eucalyptus Open-Source*. Na versão *Enterprise* as organizações tem a possibilidade de implementar nuvens privadas com suas próprias políticas, de forma bastante intuitiva utilizando todo o aparato desenvolvido pela plataforma Eucalyptus. Já na versão *Open-Source*, que também conta com todo aparato da plataforma, qualquer configuração pode ser visualizada por toda comunidade, além das seguintes restrições que são impostas, visando, de acordo com a própria Equipe Eucalyptus: “maximizar o número de membros da comunidade que se beneficiam do sistema e para minimizar os efeitos negativos dos experimentos que deram errados”.

- A duração máxima de uma instância é de 6 horas;
- No máximo 4 instâncias podem ser executadas simultaneamente por usuário (quando houver mais de quatro ocorrências, as instâncias serão aleatoriamente encerradas até que apenas quatro permaneçam);
- A duração máxima da alocação de um IP estático é de 6 horas;
- A duração máxima de um “*bucket*” *Walrus* é de 3 semanas;

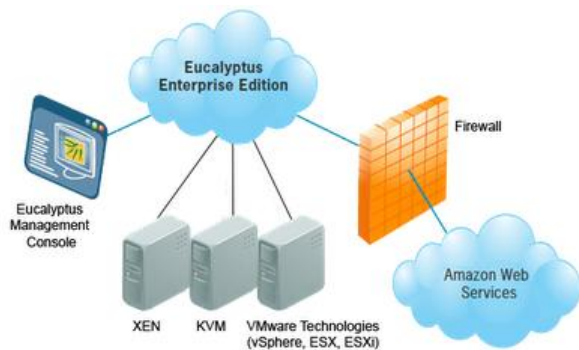


Figura 1. Overview da arquitetura do Eucalyptus e suas interfaces de comunicações

- O tamanho máximo de um “bucket” Walrus é de 5GB;
- O tempo de vida máximo de uma instância EBS é de 3 semanas;
- O tamanho máximo de uma instância EBS é de 5GB.

Segundo a Equipe Eucalyptus, caso algum projeto *open-source* esteja com a produtividade prejudicada devido a estas restrições, basta enviar um e-mail para “ecc-administrator@eucalyptus.com” que eles fornecem uma infraestrutura adequada.

A Figura 1 ilustra um *overview* da arquitetura do Eucalyptus e suas interfaces de comunicações.

III. ARQUITETURA

Os usuários do Eucalyptus tem acesso a diversos recursos, como pequenos *clusters*, conjunto de *workstations* e várias máquinas servidoras ou desktop. Como os endereços de IP são escassos e o acesso público a toda ramificação dos hosts a nuvem interna pode ser perigosa, geralmente os administradores de rede elegem um nó (conhecido como nó *front-end*) responsável por fazer um roteamento das informações públicas para os hosts internos. Estes hosts internos somente podem se comunicar com seu respectivo nó *front-end* ou com os demais hosts internos através de uma rede privada [8].

Os componentes que compõe a arquitetura do Eucalyptus são: *Node Controller*(NC), *Cluster Controller*(CC), *Cloud Controller*(CLC), *Storage Controller* e *Walrus*. A Figura 2 ilustra esta arquitetura e seus componentes.

A. Node Controller

O *Node Controller* 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.

Apesar desse cenário ser bastante propício para gerar um gargalo, na arquitetura do Eucalyptus, esta hipótese é praticamente descartada devido ao monitoramento realizado pelo *Cluster Controller*.

Em cada *Node Controller* há um arquivo WSDL que define a interface para realizar diversas operações, como por exemplo,

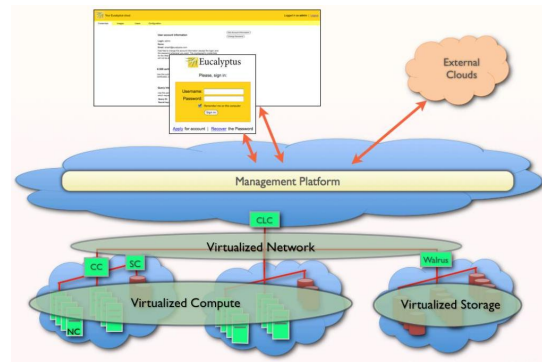


Figura 2. Arquitetura hierárquica de recursos utilizada pelo Eucalyptus

runInstance e *describeInstance* que são responsáveis, respectivamente, por iniciar uma instância e obter um relatório dos recursos físicos de uma instância (quantidade de núcleos de processamento, memória e capacidade do disco).

B. Cluster Controller

O *Cluster Controller* é o componente responsável por realizar o roteamento de pacotes entre as redes virtualizadas externa(pública) com a interna(privada). Desta forma, todas as manipulações realizadas nos recursos físicos alocados nos *Node Controllers* devem ser, necessariamente, gerenciadas pelo *Cluster Controller*.

Para tal, o *Cluster Controller* armazena um arquivo WSDL com todas as definições das interfaces para realizar as devidas operações. Este arquivo WSDL é bastante similar ao do *Node Controller*, porém este possibilita realizar as operações tanto para um único *Node Controller* quanto para um conjunto de *Nodes Controllers*.

Quando um *Cluster Controller* recebe uma requisição de *runInstance*, ele avalia qual o melhor *Node Controller* para executar esta nova instância a partir de critérios como a quantidade de recursos disponíveis e o consumo de energia da máquina física associada ao *Node Controller*.

C. Cloud Controller

O *Cloud Controller* é a porta de entrada na Nuvem para os 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* [8].

O *Cloud Controller*, conforme ilustrado na Figura 3, é composto por uma coleção de serviços que manipulam as requisições dos usuários, autenticação, sistema de persistência e metadados do usuário (por exemplo, imagens das máquinas virtuais e pares de chave ssh). Esses serviços são configurados e gerenciados por um *Enterprise Service Bus*(ESB) [9].

Para garantir a extensibilidade nessa nível de granularidade, os componentes que compõem (incluindo máquinas virtuais, SLA e as interfaces de usuário e administrador) são mutuamente isoladas e possuem interfaces de comunicação interna bem definidas, na qual toda essa orquestração é realizada pelo

ESB. Com isso, a implementação do *Cloud Controller* permite trabalhar com a Amazon EC2 usando *Web-Services* ou *Query Interfaces*

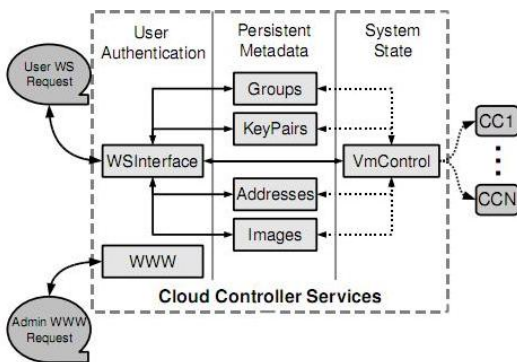


Figura 3. Serviços que compõe o *Cloud Controller*. As linhas indicam o fluxo de mensagens, na qual as linhas tracejadas indicam as mensagens do sistema interno

D. Storage Controller

O *Storage Controller* 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, iSCSI, etc. 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.

Um volume EBS não pode ser compartilhado entre instâncias, mas permitir que uma *snapshot* possa ser criado e armazenado em um sistema central de armazenamento, tal como acontece no Walrus.

E. Walrus

O *Walrus* permite aos usuários armazenar dados e organizá-los em *buckets*. Além disso, no *Walrus* é possível inserir políticas de acesso aos *buckets* e aos dados armazenados.

Cada *bucket* é compatível com *Amazon S3* e *Amazon Machine Image*(AMI), que proporciona um mecanismo de acesso aos dados do usuário pelas diferentes instâncias das máquinas virtuais.

IV. EXPERIMENTOS

Visando avaliar os principais princípios que norteiam as metas do Eucalyptus, foi realizada uma série de experimentos. Dentre eles, destacam-se os experimentos de performance, segurança, confiabilidade e da funcionalidade de ser simples de operar por usuários sem muito conhecimento em *cloud computing*. Em [7], [10], [11], [12] são descritos alguns destes experimentos.

Em [7] foi instalado o Eucalyptus em um pequeno *cluster* com Linux, composto de 7 computadores rodando NC, um único CLC *front-end* e um CC. Nesta configuração, uma requisição do usuário acessa a *Eucalyptus Public Cloud*(OPC) requisitando autorização ao CLC através de uma página web. Em seguida, a requisição de alocação é limitada em quatro instâncias, automaticamente.

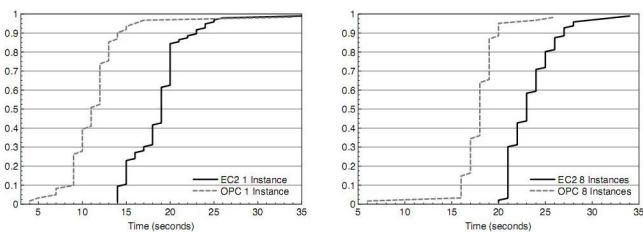


Figura 4. Comparação empírica da quantidade de segundos para iniciar uma e oito instâncias virtuais com Eucalyptus e EC2

A. Throughput em instâncias

Este experimento, descrito em [7], teve como principal objeto mensurar a performance das operações (principalmente a operação de criação) sobre uma instância virtual, comparando Eucalyptus com *Amazon EC2*. Para tal, foram disparadas diversas consultas do tipo “*ec2-describe-instances*” que simplesmente retorna as informações a respeito de uma instância e exibe no terminal. Para cada tentativa, o primeiro *timestamp* foi disparado exatamente antes da consulta e o segundo após todas as informações serem impressas corretamente na tela. O valor da diferença entre estes dois *timestamp* representa os segundos entre a criação da instância e esta ficar disponível para uso. Cada valor é caracterizado por quatro variáveis: a primeira representa o tipo da VM requisitada, composto pela quantidade de núcleos, memória RAM e espaço de disco; a segunda variável representa a imagem da instância, que, neste experimento, tanto no Eucalyptus quanto no EC2 foram utilizados uma distribuição compacta do Linux, conhecida como “*tylinux*” [13]; a terceira variável de interesse é o número de instâncias simuladas, que foi variado de uma até oito instâncias; por fim, a última variável é o tipo do sistema utilizado, Eucalyptus ou EC2.

A Figura 4 demonstra os resultados deste experimento. A figura mostra duas funções de distribuição acumulativa que permitem avaliar a variância do tempo necessário para criar instâncias virtuais usando Eucalyptus e EC2. O eixo Y representa a porcentagem do total de criação das instâncias enquanto o eixo X os segundos.

Em ambos os casos, é evidente que as instâncias baseadas no Eucalyptus EPC foram criadas antes do EC2. Por exemplo, 98% das oito instâncias já haviam sido criadas com menos de 24 segundos, enquanto com EC2 apenas 75% das instâncias foram criadas. No caso de uma única instância, esta diferença é ainda maior: 98% das instâncias já haviam sido criadas com menos de 17 segundos, enquanto que no EC2, com os mesmos 17 segundos, apenas 32% haviam completados.

Este resultado pode indicar que a implementação do Eucalyptus é relativamente mais eficiente que EC2, porém não possível afirmar tal hipótese, devido a dois fatores principais: o primeiro é a quantidade de recursos disponíveis no Eucalyptus e o segundo é que o EC2 não foi projetado para obter bons resultados em ambientes simulados com pequenos *clusters*.

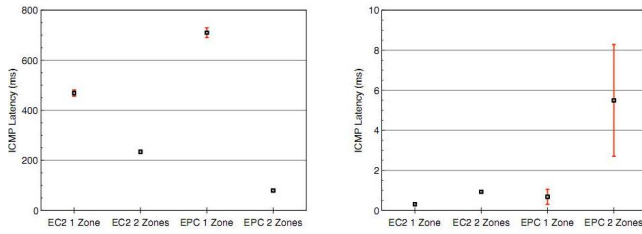


Figura 5. Medidas *throughput* TCP (esquerda) e latência *roundtrip*(direita) entre instâncias alocadas com Eucalyptus e EC2

B. Performance de rede

O segundo experimento, descrito em [7], tem como objetivo avaliar as características de uma conexão em rede do Eucalyptus com a abordagem EC2. Como a configuração de rede do EC2 não é conhecida, não é possível comparar os dois sistemas em termos de detalhes funcionais. Qualquer discrepância pode indicar uma diferença positiva significativa na abordagem ou um “bug” na implementação do Eucalyptus. A implementação do Eucalyptus utilizada foi a *Virtual Distributed Ethernet*(VDE) [14].

O experimento foi conduzido entre dois ambientes simulados: cliente e servidor. Em ambos, foi utilizada a distribuição Linux Debian e instalado a ferramenta “iperf” [15] para monitorar o tráfego de rede, tanto TCP quanto UDP. Além disso, foi utilizado “ping” para mensurar a latência de um *echo* ICMP.

Para avaliar como a distância física afeta a performance da rede, foi realizado um teste com cliente e servidor na mesma zona e outro em zonas diferentes.

A Figura 5 ilustra os resultados encontrados no experimento de rede. Além da média aritmética das 32 medições individuais (os quadrados), também é exibido o intervalo de confiança, estipulado em 95%, quando o intervalo é grande o suficiente para ser considerado.

Com o EC2 foi observado que a largura de banda numa mesma zona supera em, aproximadamente, fator de 2 em relação a zonas diferentes. A possível justificativa para tal é a adição dos mecanismos de segurança na rede privada VDE que não existem na EC2. Outro fato constatado esta relacionado à latência dos resultados, mostrando uma variância muito maior no RTT para os pacotes ICMP ao percorrer a rede VDE, os mecanismos de segurança impostos na rede VDE novamente são justificativa para esse segundo fato.

Este experimento também demonstrou o impacto de desempenho associado com a abordagem de sobreposição implementada pelo Eucalyptus. Especificamente numa rede VDE que é necessário implementar um mecanismo de segurança e uma sobreposição no espaço do usuário na camada dois para cada *cluster cloud* alocado separadamente numa rede privada.

V. CONCLUSÃO

Este artigo demonstrou um breve histórico do desenvolvimento do Eucalyptus que, inicialmente, era um simples projeto de Universidade sem fins mercadológicos e hoje é referência

quando se trata de *cloud computing* privada, com adoção por diversos governos e organizações. Também foram debatidas as metas que nortearam o desenvolvimento da plataforma Eucalyptus, baseada na extensibilidade e na opção de criação de redes privadas. A extensibilidade é garantida a partir de uma arquitetura totalmente modularizada, na qual cada interface é bem definida em arquivos WSDL. Já as redes privadas são executadas sobre a rede *open-source* e conta com um mecanismo de autenticação baseado em *WS-Security*. Nestas redes privadas, os usuários podem realizar a implementação de qualquer política local, gerenciando todos os recursos e serviços disponibilizados pelo Eucalyptus.

Além disso, foram debatidos os cinco principais componentes que compõem a arquitetura do Eucalyptus e suas respectivas responsabilidades para garantir segurança, confiabilidade e a disponibilidade dos recursos. Destacam-se os seguintes componentes: *Node Controller*, pela capacidade de ser executado em apenas uma máquina física; *Cluster Controller* por realizar o roteamento da rede pública para a rede interna; e o *Cloud Controller* por gerenciar a interface do usuário e seus recursos alocados e com o restante das funcionalidades do Eucalyptus.

Devido à capacidade de implementar Nuvens privadas, várias organizações aderiram ao Eucalyptus. Dentre elas, destaca-se a *National Aeronautics and Space Administration*(NASA) que é uma agência do Governo dos Estados Unidos da América, responsável pela pesquisa e desenvolvimento de tecnologias e programas de exploração espacial. Certamente a NASA contém uma série de dados sigilosos como informações de satélites, missões, sondas de pesquisa espacial e etc, logo ela optou por criar a Nuvem NASA NEBULA. Para a implementação desta Nuvem, foi empregado a plataforma Eucalyptus para gerenciar todas as máquinas virtuais. Além disso, a partir do Eucalyptus, a NASA NEBULA implementa um SaaS, PaaS e IaaS [16].

Outra organização que utiliza o Eucalyptus é a *Ubuntu Enterprise* para a construção da *Ubuntu Enterprise Cloud*(UEC). Esta solução foi desenvolvida para construir e gerenciar uma Nuvem interna de qualquer empresa, independente do seu tamanho, possibilitando as pequenas empresas a criar sua própria infraestrutura *cloud* [17].

Por fim, a multinacional farmacêutica *Lilly* também construiu uma Nuvem privada a partir da plataforma Eucalyptus para gerenciar seus recursos de *hardware* com uma interface com a *Amazon AWS*.

Na opinião do autor deste artigo, o Eucalyptus complementa uma lacuna que poderia afastar o governo e as grandes corporações da plataforma *cloud computing*: a criação de uma Nuvem privada, sem diminuir as vantagens da utilização de uma Nuvem pública. Como o Eucalyptus é uma plataforma *open-source*, torna-se um excelente material didático para a investigação pelos pesquisadores e desenvolvedores Nuvem, além de incentivar a participação da comunidade no aprimoramento da plataforma.

Além disso, o autor acredita que com uma arquitetura totalmente modularizada, o Eucalyptus está devidamente preparado

para as constantes mudanças no paradigma *cloud computing*.

Outro fator constatado é a dificuldade em realizar experimentos empíricos para avaliar o Eucalyptus, devido a uma série de fatores, dentre eles: dificuldade de quantificar algumas variáveis, subjetividade em algumas informações ou até mesmo a falta de informação a respeito de um determinado conceito técnico. Além disso, não há, até o momento, outra plataforma semelhante ao Eucalyptus torna-se praticamente impossível de fazer conclusões a partir de comparações.

O autor também acredita que o Eucalyptus ainda não atingiu todo o seu potencial em termos financeiros, e, futuramente, várias organizações irão cada vez mais utilizar a plataforma. Logo, cabe a Equipe Eucalyptus continuar seu aprimoramento, incluindo outros *hypervisors* e aumentar a abstração da nuvem e seus componentes.

REFERÊNCIAS

- [1] C. Taurion, *Transformando o mundo da tecnologia da informação*. Associação Catarinense de Bibliotecários (ACB), 2009, vol. 15.
- [2] F. R. C. Sousa, L. O. Moreira, and J. C. Machado, "Computação em nuvem: Conceitos, tecnologias, aplicações e desafios," *ERCEMAPI 2009*, 2010. [Online]. Available: http://www.es.ufc.br/~flavio/files/Computacao_Nuvm.pdf
- [3] F. S. Chirigati, L. H. Costa, and O. Duarte, "Computação em nuvem," 2010. [Online]. Available: http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2009_2/seabra/
- [4] P. Mell and T. Grance, "The nist definition of cloud computing," 2009. [Online]. Available: <http://csrc.nist.gov/groups/SNS/cloudcomputing/cloud-def-v15.doc>
- [5] G. C. Drive, "Cloud computing and open source: It climatology is born," *Eucalyptus Systems, Inc.*, 2010.
- [6] E. G. C. Drive, "Five steps to enterprise cloud computing," *Eucalyptus Systems, Inc.*, 2010.
- [7] D. Nurmi, R. W. C. Grzegorzczuk, G. Obertelli, S. Soman, L. Youselfff, and D. Zagorodnov, "Eucalyptus: A technical report on an elastic utility computing architecture linking your programs to useful systems," *UCSB Computer Science Technical*, 2008-2010.
- [8] G. C. Drive, "Eucalyptus open-source cloud computing infrastructure - an overview," *Eucalyptus Systems, Inc.*, 2009.
- [9] M. Schmidt, B. Hutchison, P. Lambros, and R. Phippen, "The enterprise service bus: Making service-oriented architecture real," *IBM Systems Journal*, pp. 781-797, 2005.
- [10] G. C. Drive, "Turning months into minutes with eucalyptus," 2010.
- [11] U. E. Cloud, C. Ltd., and I. Eucalyptus Systems, "Intel cloud builder guide to cloud design and deployment on intel platforms - ubuntu enterprise cloud," 2010.
- [12] L. M. Vaquero, L. Roderio-Merino, and R. Buyya, "Dynamically scaling applications in the cloud," in *ACM SIGCOMM Computer Communication Review*, vol. 41, 2011, pp. 45-52.
- [13] *Ttylinux Home Page*. [Online]. Available: <http://www.minimalinux.org/ttylinux/>
- [14] *Virtual Distributed Ethernet(VDE) Home Page*. [Online]. Available: <http://vde.sourceforge.net/>
- [15] *Iperf Home Page*. [Online]. Available: <http://iperf.sourceforge.net/>
- [16] *NASA NEBULA Home Page*. [Online]. Available: <http://nebula.nasa.gov/services/>
- [17] *Ubuntu Enterprise Cloud(UEC) Home Page*. [Online]. Available: <http://www.ubuntu.com/business/cloud/overview>