

UNIVERSIDADE DE CAXIAS DO SUL  
CENTRO DE COMPUTAÇÃO E TECNOLOGIA DA INFORMAÇÃO  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MARCELO CRESTANI BRAZIL

**Desenvolvimento de um Ambiente  
Computacional Baseado em Cloud  
Computing**

Prof. André L. Martinotto  
Orientador

Caxias do Sul, Dezembro de 2009

*“Nada é impossível  
para aquele que persiste.”*

— ALEXANDRE, O GRANDE

*“Only two things are infinite,  
the universe and human stupidity,  
and I’m not sure about the former.”*

— ALBERT EINSTEIN

## AGRADECIMENTOS

Agradeço, em primeiro lugar, aos meus pais, Hélio e Angela, que me deram todo o apoio necessário durante essa longa jornada, me incentivaram desde o início, me deram força quando eu mais necessitava e sempre batalharam para que esse sonho pudesse ser realizado.

Agradeço também à minha irmã Danielle, pela companhia nos momentos mais difíceis, pela amizade e também por sempre ter me auxiliado da maneira que podia.

Ao meu orientador Prof. André Martinotto por ter sido um verdadeiro mestre. Também pelos seus incentivos, pela sua paciência e dedicação, pelos seus sábios conselhos, pelo seu companheirismo e, principalmente, pela sua amizade. Hoje eu sei que tenho nele um amigo.

Ao Guilherme, pela companhia nessa reta final e pelo auxílio nas dificuldades pelas quais passamos neste trabalho.

Aos professores que, durante esses, anos sempre estiveram dispostos a compartilhar comigo seu conhecimento.

A todos os colegas que, de uma maneira ou outra, me auxiliaram ao longo da minha vida acadêmica. Em especial, ao Alex, ao Chaveiro, ao Davi, ao José, ao Mário, ao Rodrigo, ao Sehba e ao Teco, que foram essenciais e me acompanharam por quase todo esse tempo. Agradeço também ao Saimon que, apesar de tudo, também estava sempre junto nesses momentos. E ao Irmão. Amigo e companheiro de todas as horas. Todas as horas, sejam elas boas ou ruins.

Aos amigos. Os amigos de ontem, aos amigos de hoje, mas, principalmente, aos amigos de sempre.

Aos meus ex-colegas e atuais colegas de trabalho, que sempre me incentivaram e souberam compreender o que o cansaço pode fazer com uma pessoa.

A todos aqueles que duvidaram de mim. Vocês podem ter certeza de que o maior incentivo que tive foi de vocês.

A todas essas pessoas, o meu mais sincero agradecimento.

*Gracias Totales!*

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS</b> . . . . .	6
<b>LISTA DE FIGURAS</b> . . . . .	7
<b>LISTA DE TABELAS</b> . . . . .	9
<b>RESUMO</b> . . . . .	10
<b>ABSTRACT</b> . . . . .	11
<b>1 INTRODUÇÃO</b> . . . . .	12
1.1 <b>Objetivos</b> . . . . .	13
1.2 <b>Estrutura do Trabalho</b> . . . . .	13
<b>2 COMPUTAÇÃO NAS NUVENS</b> . . . . .	15
2.1 <b>Computação nas Nuvens</b> . . . . .	16
2.1.1 <b>Infra-Estrutura de Nuvens Computacionais</b> . . . . .	17
2.1.2 <b>Vantagens de Computação nas Nuvens</b> . . . . .	19
2.1.3 <b>Desvantagens de Computação nas Nuvens</b> . . . . .	20
2.2 <b>Serviços Baseados em Computação Nas Nuvens</b> . . . . .	21
2.2.1 <i>Google Apps</i> . . . . .	21
2.2.2 <i>Amazon Web Services</i> . . . . .	22
2.2.3 <i>openQRM</i> . . . . .	24
2.3 <b>Considerações Finais</b> . . . . .	25
<b>3 GERENCIAMENTO DE UMA NUVEM COMPUTACIONAL</b> . . . . .	26
3.1 <b>Infra-Estrutura Física</b> . . . . .	26
3.2 <b>Gerenciador de Uma Nuvem Computacional</b> . . . . .	28
3.2.1 <i>Eucalyptus</i> . . . . .	29
3.2.2 <i>OpenNebula</i> . . . . .	30
3.2.3 <i>Citrix Cloud Center</i> . . . . .	31

3.2.4	VMware vSphere . . . . .	32
<b>3.3</b>	<b>Considerações Finais . . . . .</b>	<b>33</b>
<b>4</b>	<b>VIRTUALIZAÇÃO . . . . .</b>	<b>35</b>
<b>4.1</b>	<b>Máquinas Virtuais . . . . .</b>	<b>35</b>
4.1.1	Técnicas Para Desenvolvimento de Máquinas Virtuais . . . . .	36
4.1.2	Vantagens e Desvantagens do Uso de Virtualização . . . . .	39
<b>4.2</b>	<b>Monitores de Máquinas Virtuais . . . . .</b>	<b>40</b>
4.2.1	VMware ESX . . . . .	40
4.2.2	Xen . . . . .	41
4.2.3	KVM . . . . .	41
<b>4.3</b>	<b>Considerações Finais . . . . .</b>	<b>42</b>
<b>5</b>	<b>PROJETO DE UM AMBIENTE COMPUTACIONAL BASEADO EM CLOUD COMPUTING . . . . .</b>	<b>43</b>
<b>5.1</b>	<b>Arquitetura da Nuvem Computacional . . . . .</b>	<b>43</b>
<b>5.2</b>	<b>Instalação da Nuvem Computacional . . . . .</b>	<b>45</b>
<b>5.3</b>	<b>Testes da Nuvem Computacional . . . . .</b>	<b>48</b>
5.3.1	Inclusão de Recursos na Nuvem Computacional . . . . .	48
5.3.2	Manutenção da Nuvem Computacional . . . . .	51
5.3.3	Tolerância a Falhas . . . . .	53
5.3.4	Ambientes Padronizados Sob Demanda . . . . .	54
5.3.5	Atualização de Um Ambiente na Nuvem Computacional . . . . .	56
<b>5.4</b>	<b>AppScale . . . . .</b>	<b>57</b>
<b>5.5</b>	<b>Considerações Finais . . . . .</b>	<b>60</b>
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>61</b>
6.1	Trabalhos Futuros . . . . .	62
	<b>REFERÊNCIAS . . . . .</b>	<b>64</b>
<b>7</b>	<b>ANEXOS . . . . .</b>	<b>68</b>
7.1	Instalação Da Nuvem Computacional . . . . .	68
7.2	Instalação Ao AppScale . . . . .	75

## LISTA DE ABREVIATURAS E SIGLAS

API	<i>Aplication Programming Interface</i>
CAD	<i>Computer-Aided Design</i>
CD	<i>Compact Disc</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DVD	<i>Digital Versatile Disc</i>
ERP	<i>Enterprise Resource Planning</i>
IBM	<i>International Business Machine</i>
Gb	<i>Gigabyte</i>
GPL	<i>General Public License</i>
Ghz	<i>Gigahertz</i>
IP	<i>Internet Protocol</i>
IRQ	<i>Interrupt Request</i>
Mb	<i>Megabyte</i>
NTP	<i>Network Time Protocol</i>
PC	<i>Personal Computer</i>
PDA	<i>Personal Digital Assistants</i>
RAM	<i>Random Access Memory</i>
SLA	<i>Service Level Agreement</i>
Tb	<i>Terabyte</i>
USB	<i>Universal Serial Bus</i>

## LISTA DE FIGURAS

Figura 2.1: Infra-estrutura da Nuvem Computacional . . . . .	18
Figura 3.1: Infra-Estrutura de uma Nuvem Computacional . . . . .	27
Figura 4.1: Tipos de Máquinas Virtuais . . . . .	37
Figura 4.2: Virtualização Total . . . . .	38
Figura 4.3: Paravirtualização . . . . .	38
Figura 5.1: Infra-estrutura da Nuvem Computacional Desenvolvida . . . . .	44
Figura 5.2: <i>Website</i> de Configuração do <i>Eucalyptus</i> . . . . .	46
Figura 5.3: Cenário de Teste de Inclusão de Recursos na Nuvem Computacional	49
Figura 5.4: <i>Log</i> do Controlador da Nuvem No Teste de Inclusão de Recursos .	49
Figura 5.5: <i>Log</i> do Controlador Para Solicitação de Novas Instâncias no Teste de Inclusão de Recursos . . . . .	50
Figura 5.6: Cenário de Teste de Manutenção na Nuvem Computacional . . .	51
Figura 5.7: <i>Log</i> do Controlador da Nuvem No Cenário de Teste Para a Manutenção da Nuvem Computacional . . . . .	52
Figura 5.8: <i>Log</i> do Nodo Desabilitado No Cenário de Teste Para a Manutenção da Nuvem Computacional . . . . .	52
Figura 5.9: Cenário de Teste de Tolerância a Falhas . . . . .	53
Figura 5.10: <i>Log</i> do Controlador da Nuvem No Cenário de Teste de Tolerância a Falhas . . . . .	54
Figura 5.11: <i>Log</i> do Nodo No Cenário de Teste de Tolerância a Falhas . . . . .	54
Figura 5.12: Configuração das Classes de Ambientes da Nuvem Computacional	55
Figura 5.13: Máquinas Virtuais Ativas Na Alocação Inicial . . . . .	56
Figura 5.14: Máquinas Virtuais Ativas Após a Solicitação de Nova Instância .	56
Figura 5.15: Tela da Aplicação <i>Guestbook</i> . . . . .	58
Figura 5.16: <i>Website</i> do <i>AppScale</i> Com o Status da Aplicação . . . . .	59
Figura 5.17: Execução do Comando <i>appscale-describe-instances</i> . . . . .	60
Figura 7.1: Página Inicial do <i>Eucalyptus</i> . . . . .	73

Figura 7.2: Página de Credenciais do *Eucalyptus* . . . . . 74



## LISTA DE TABELAS

Tabela 2.1: <i>Valor hora de uma máquina virtual na Amazon EC2</i> . . . . .	24
--	----

## RESUMO

A computação nas nuvens pode ser descrita como um novo paradigma para infraestrutura que tem como objetivo prover um serviço sob demanda utilizando-se do potencial disponível em computadores distribuídos e baseando-se na utilização de virtualização.

Para a criação de uma nuvem computacional, são necessários de alguns computadores interconectados em uma rede. Além disso, para o gerenciamento desses computadores e da nuvem computacional, é necessário um *software* gerenciador que, além de fornecer mecanismos que permitam o gerenciamento da nuvem computacional, será responsável por prover o serviço que a nuvem computacional oferece.

Este trabalho teve como objetivo realizar um estudo sobre computação nas nuvens e o desenvolvimento de uma infra-estrutura de nuvem computacional. Essa foi desenvolvida utilizando-se do gerenciador *Eucalyptus* e possui como principal objetivo fornecer poder de processamento aos seus usuários.

**Palavras-chave:** Computação nas nuvens, virtualização.

## Developing an Computing Enviroment Based On Cloud Computing

### ABSTRACT

Cloud computing can be described as a new paradigm for infrastructure that aims to provide a service on demand using all resources available on distributed computers through of the using of the virtualization.

To create a computational cloud are necessary some computers connected on a network. Yet, its necessary, to manage these computers and the cloud, a software that, in adittion to provide tools to manage the cloud, also will be responsible to provide the service that the cloud will offer.

This work has the goal of study cloud computing and build an computational cloud. This cloud was built using Eucalyptus as his manager and it has as his main goal to provide processing power to his users.

**Keywords:** Cloud Computing, Virtualization.

# 1 INTRODUÇÃO

Os primeiros sistemas computacionais surgiram com o objetivo de auxiliar e automatizar tarefas que, anteriormente, eram realizadas de forma manual e, conseqüentemente, demandavam muito tempo para a sua execução. Além disso, por serem realizados manualmente, estavam muito suscetíveis a erros. Essa automatização proporcionou às empresas uma diminuição significativa dos seus custos.

Essa contribuição que os sistemas computacionais proporcionam acaba justificando os grandes esforços para a constante evolução desses sistemas, uma vez que a necessidade de poder de processamento é cada vez maior. Entretanto essa constante evolução acaba proporcionando um grande problema, pois os computadores tornam-se desatualizados em um intervalo cada vez menor de tempo. Portanto, torna-se altamente dispendioso para as empresas manter esses sistemas atualizados. Além disso, essas empresas apresentam o desafio de extrair o máximo possível desses recursos de forma a aproveitar melhor os investimentos efetivados.

Nas últimas décadas, observou-se o surgimento de diversos paradigmas que tinham como objetivo a resolução desse problema da constante necessidade de poder de processamento. Um exemplo desses é a computação nas nuvens (*cloud computing*). Esse paradigma é uma proposta de infra-estrutura com o objetivo de oferecer serviços sob demanda através de uma rede, aproveitando-se do poder de processamento disponível em computadores distribuídos. Essa característica acaba proporcionando algumas facilidades para o gerenciamento, manutenção e atualização dessa infra-estrutura, uma vez que essa estará centralizada em algumas poucas máquinas físicas.

A computação nas nuvens tem atraído a atenção e os investimentos de grandes empresas do ramo da informática, como, por exemplo, *Google*, *IBM* e *Sun*, bem como de outras empresas que não tinham na informática o seu principal ramo de atuação, como, por exemplo, a *Amazon.com*.

Dentro desse contexto, neste trabalho foi desenvolvida uma infra-estrutura baseada no conceito de computação nas nuvens. Essa infra-estrutura tem como

objetivo disponibilizar aos usuários um ambiente onde ele possa desfrutar do poder de processamento de uma nuvem computacional. Além disso, foi realizada uma avaliação das tecnologias e das ferramentas disponíveis para a criação de uma nuvem computacional. Por fim, foi escolhida uma aplicação para ser utilizada para testes de desempenho da infra-estrutura que foi desenvolvida.

## 1.1 Objetivos

Este trabalho possui os seguintes objetivos:

- Projeto e desenvolvimento de uma arquitetura de um ambiente computacional baseado em computação nas nuvens;
- Avaliação da ferramenta utilizada para o gerenciamento da nuvem computacional;
- Escolha ou desenvolvimento de uma aplicação que permita avaliar a infra-estrutura de nuvem desenvolvida.

## 1.2 Estrutura do Trabalho

O presente trabalho está organizado da seguinte forma:

- No Capítulo 2 é apresentada uma definição de computação nas nuvens. Nesse, também, é apresentada a infra-estrutura de uma nuvem computacional, bem como vantagens e desvantagens da utilização de computação nas nuvens. Por fim, são apresentados exemplos de serviços baseados em computação nas nuvens;
- No Capítulo 3 são apresentados alguns aspectos sobre a infra-estrutura física necessária para a criação de uma nuvem computacional. Além disso, são descritas algumas características dos *softwares* gerenciadores de nuvem computacional. Por fim, são apresentados alguns *softwares* gerenciadores utilizados em nuvens computacionais;
- No Capítulo 4 é descrito de uma forma um pouco mais detalhada a técnica de virtualização, apresentando-se as técnicas que podem ser utilizadas no desenvolvimento de máquinas virtuais. Além disso, são apresentadas vantagens e desvantagens da utilização de virtualização em nuvens computacionais. Após, são apresentados alguns *softwares* monitores de máquinas virtuais;

- No Capítulo 5 é apresentada a nuvem computacional desenvolvida nesse trabalho. Nele são apresentados os passos para a implementação desse ambiente e alguns testes realizados;
- No Capítulo 6 das descritas as conclusões do trabalho juntamente com sugestões para possíveis trabalhos futuros.

## 2 COMPUTAÇÃO NAS NUVENS

A informática é conhecida por ser uma área extremamente dinâmica. O que hoje é uma novidade amanhã poderá ser considerado ultrapassado. As quebras de paradigmas acontecem com uma velocidade tão grande que torna-se difícil acompanhá-las. Dentro deste ambiente de constantes mudanças sempre existiu uma preocupação das empresas em obter o máximo de desempenho possível utilizando a menor quantidade de recursos computacionais.

No início da era da informática, os computadores possuíam grandes dimensões e eram extremamente caros. Desta forma, o uso desses era limitado a grandes corporações, universidades ou entidades governamentais. Entretanto, no início da década de 1980, com a evolução da tecnologia, com a criação e a popularização dos PCs (computadores pessoais), os grandes computadores tornaram-se obsoletos, uma vez que a maioria do processamento poderia ser realizado utilizando-se de PCs (HAYES, 2008).

Com a popularização dos sistemas computacionais, esses passaram a ser encarados pelas empresas como uma forma de redução dos custos operacionais, uma vez que esses auxiliam a diminuir o tempo necessário para a execução de quase todas as tarefas. Desta forma, os investimentos das empresas nos setores de tecnologia da informação tem aumentado com o passar dos anos (HAYES, 2008).

Porém, com o desenvolvimento da indústria tecnológica, os recursos computacionais tornam-se obsoletos em um período cada vez menor. Essa questão, em particular, é crucial no momento em que as empresas decidem investir em melhorias na sua infra-estrutura computacional. Hoje, uma empresa que deseja manter-se atualizada tecnologicamente necessita dispor de muitos recursos financeiros. Além disso, essas empresas possuem receio em investir em uma infra-estrutura que torna-se obsoleta de forma tão rápida. Uma proposta para a solução deste dilema é o uso do conceito de computação nas nuvens (*cloud computing*) (STARK; STERLING, 2009).

Esse novo paradigma, apesar de ser amplamente discutido, ainda não possui uma definição que seja de consenso no mundo da informática. Na próxima seção será

apresentada uma definição de computação nas nuvens. Além disso, na Seção 2.2 será feita uma breve descrição sobre a infra-estrutura de uma nuvem computacional. Já nas Seções 2.3 e 2.4 serão apresentadas, respectivamente, vantagens e desvantagens da utilização de computação nas nuvens. Por fim, na Seção 2.5 serão apresentados exemplos de serviços baseados em computação nas nuvens.

## 2.1 Computação nas Nuvens

Como já mencionado, o paradigma de computação nas nuvens ainda não possui uma definição que seja de consenso na informática. Muitos autores propõem diferentes definições para uma nuvem computacional, sendo que essas possuem algumas características em comum que são chaves para o funcionamento de uma nuvem. Algumas dessas definições são apresentadas abaixo (GEELAN, 2009):

- R. Buya (2008): "Uma nuvem computacional é um sistema distribuído e virtualizado, onde os recursos são dinamicamente provisionados de acordo com contratos de serviços estabelecidos através de negociações entre os provedores de serviço e os consumidores";
- R. Cohen (2008): "Computação nas nuvens é um daqueles jogos de palavras que tenta abranger uma variedade de aspectos, desde a criação, balanceamento de carga, provisionamento e arquitetura. A explicação mais simples é que ela pode ser descrita como uma centralização de *softwares* numa rede";
- J. Kaplan (2008): "Uma rede de serviços que tem como objetivo permitir aos usuários obter uma vasta quantidade de funcionalidades baseado no método *pay-per-use* (pague pelo que você utiliza)";
- B. Kepes (2008): "Computação nas nuvens é a mudança de paradigma de infra-estrutura que permite a ascensão do *software* como um serviço";
- K. Sheynkman (2008): "Nuvens computacionais são focadas em transformar a camada de *hardware* de maneira a permitir a utilização de recursos conforme a demanda. Este é um primeiro passo importante, porém, para que as empresas possam utilizar todo o poder de uma nuvem computacional, a infra-estrutura dessa deve ser de fácil configuração e ser escalável nesses ambientes virtualizados";
- J. Pritzker (2008): "Nuvens computacionais são conjuntos de recursos virtualizados e com alocações conforme a demanda";



- P. McFedries (2008): "Com computação nas nuvens, além dos nossos dados residirem dentro da nuvem, nós podemos acessá-los não só através de PCs, mas também, de outros dispositivos, como, por exemplo, *smart phones*".

Assim, a computação nas nuvens pode ser descrita como um novo paradigma para infra-estrutura que tem como objetivo prover um serviço sob demanda, através de uma rede, utilizando-se do potencial disponível em computadores distribuídos. Essa baseia-se na utilização de virtualização, de forma a aproveitar todo o potencial disponível nesses computadores, diminuindo a ociosidade e, conseqüentemente, obtendo-se o máximo de capacidade de processamento possível. Como a virtualização é uma das principais técnicas utilizadas na criação da infra-estrutura de uma nuvem computacional, ela será abordada com mais detalhes no Capítulo 4.

Além disso, muitos autores comparam a computação nas nuvens com computação em *grids*. Doerksen (GEELAN, 2009) afirma que "computação nas nuvens é a versão amigável de computação em *grid*". O principal objetivo de um *grid* computacional é utilizar todo o potencial disponível em computadores distribuídos para realizar o processamento ou o armazenamento de dados, de forma que este processo seja transparente para quem utiliza os recursos do *grid* (KESSELMAN; FOSTER, 1998). A principal diferença entre uma nuvem computacional e um *grid* computacional consiste, principalmente, no fato de que a primeira delas utiliza-se de virtualização para obter o potencial máximo disponível de um computador.

### 2.1.1 Infra-Estrutura de Nuvens Computacionais

A Figura 2.1 demonstra, de forma abstrata, a infra-estrutura de uma nuvem computacional. Essa estrutura é bastante simples e consiste, fisicamente, em computadores interconectados por uma rede, que serão responsáveis por fornecer os recursos que serão utilizados pela nuvem. Para o usuário acessar a infra-estrutura da nuvem, ele necessitará de um dispositivo, como por exemplo, um computador, um PDA ou um telefone celular, com o qual ele consiga conectar-se a nuvem. A partir do momento que a conexão for estabelecida, o usuário poderá usufruir do serviço da nuvem (MILLER, 2009).

Para o controle dessa infra-estrutura, é necessário um *software* gerenciador. Esse será responsável pelo gerenciamento da infra-estrutura da nuvem, frequentemente de forma centralizada, e deverá ser capaz de, por exemplo: alocar os recursos, controlar o acesso dos usuários aos recursos e aos serviços, permitir a inclusão de novos nodos<sup>1</sup>, prover os mecanismos de tolerância a falhas, controlar todos os aspectos relacionados a segurança da nuvem, etc. Alguns exemplos de *softwares* gerenciadores são:

---

<sup>1</sup>Nodo é cada um dos computadores que pertencem a infra-estrutura da nuvem computacional.

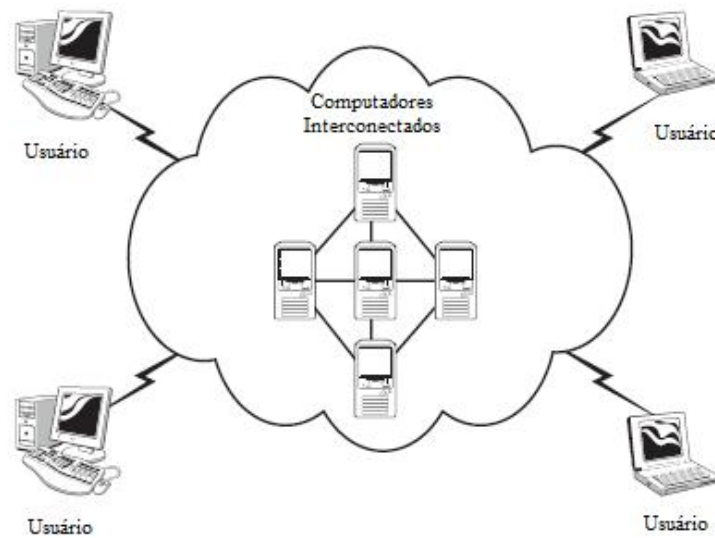


Figura 2.1: Infra-estrutura da Nuvem Computacional

o *OpenNebula* (OPENNEBULA.ORG, 2009), o *Eucalyptus* (Eucalyptus, 2009a), *VMware vSphere* (VMWARE, 2009a) e o *Citrix Cloud Center* (CITRIX, 2009a). Esses *softwares* serão descritos com maiores detalhes no Capítulo 3.

Um outro componente importante dessa infra-estrutura consiste no serviço a ser provido por ela. De acordo com o serviço, as nuvens computacionais podem ser classificadas em (LEAVITT, 2009):

- **Infra-Estrutura:** a nuvem prove toda a infra-estrutura que um computador convencional normalmente fornece, ou seja, um usuário ao utilizar a infra-estrutura da nuvem, irá acessar os recursos como se estivesse utilizando o seu PC, sem necessariamente ter conhecimento da infra-estrutura da nuvem. Como exemplo de uma nuvem computacional que se enquadra nesta categoria pode-se citar a *Amazon EC2* (*Amazon Elastic Compute Cloud*), que é descrita na Seção 2.2.2;
- **Plataforma:** a nuvem prove toda uma infra-estrutura necessária para o desenvolvimento de aplicações. Desta forma, os usuários podem utilizar essa estrutura para o desenvolvimento de *software* de forma *online* e concorrentemente com os demais usuários. Como exemplo de uma nuvem computacional que se enquadra nesta categoria pode-se citar o *Google App Engine*, que será descrita na Seção 2.2.3;
- **Software:** a nuvem prove uma série de *softwares* que podem ser utilizadas através da rede pelos usuários. Estes *softwares* podem variar, desde simples editores de texto até aplicações mais complexas, como um ERP (*Enterprise Resource Planning*). Como exemplo de uma nuvem computacional que se

enquadra nesta categoria pode-se citar o *Photoshop.com*, que é uma versão *online* do software *Photoshop* (*software* para edição de imagens) da empresa Adobe. O *Photoshop.com* disponibiliza, ainda, ao usuário uma área com capacidade de armazenamento de 2 Gb para o armazenamento de suas imagens.

### 2.1.2 Vantagens de Computação nas Nuvens

Em uma nuvem computacional, a utilização de um gerenciamento centralizado traz diversas vantagens. Dentre essas destaca-se a possibilidade de uma fácil alteração dos recursos que essa utiliza. Por exemplo, para a adição de novos recursos é necessário somente inicializar novas máquinas virtuais e disponibilizá-las na nuvem. A inclusão de novas máquinas físicas (nodos) na infra-estrutura da nuvem também torna-se mais fácil, pois elas somente passarão a fazer parte da infra-estrutura no momento em que o administrador desejar. Assim como os recursos podem ser adicionados a nuvem, eles também podem ser desalocados facilmente.

O gerenciamento centralizado facilita ainda a manutenção de qualquer nodo da nuvem. Para isso, basta desabilitar o nodo e habilitá-lo novamente após o término da manutenção. Além disso, com uma maior facilidade para a inclusão de novas máquinas físicas, pode-se também aumentar o número de usuários da nuvem sem comprometer o desempenho da mesma (LEAVITT, 2009).

Essa possibilidade de um gerenciamento centralizado também proporciona ganhos relacionados a segurança de sua estrutura. Uma vez que todos os recursos da infra-estrutura são controlados de um único ponto, torna-se mais fácil controlar o acesso e a utilização dos recursos. Além disso, pelo fato de todas as informações estarem armazenadas em poucas máquinas, torna-se mais simples a recuperação de informações perdidas. Isso deve-se, principalmente, a uma maior facilidade na realização de *backups*, uma vez que é mais simples efetuar *backups* de informações contidas em poucos servidores do que em várias máquinas de usuários.

Outra vantagem da utilização de computação nas nuvens consiste no fato de que, ao prover ao usuário um ambiente de trabalho semelhante ao ambiente que ele utiliza numa máquina física, o usuário não sofrerá um impacto tão grande ao passar a utilizar a nuvem. Em algumas situações, como a nuvem possui uma maior capacidade de processamento e um maior espaço de armazenamento, será melhor para o próprio usuário a utilização da mesma.

Além disso, através de uma nuvem computacional, é possível oferecer aos usuários um ambiente padrão. Dessa forma, pode ser criado um ambiente configurado conforme o perfil de cada usuário da nuvem. Esse ambiente padrão poderá, por exemplo, conter uma determinada configuração de *hardware*, especificando, por exemplo, velocidade do processador, quantidade de memória RAM e quantidade

de disco rígido para cada usuário, além de alguns *softwares* instalados.

Finalmente, o gerenciador da nuvem também pode definir a maneira pela qual cada usuário irá utilizar a nuvem. Essa definição poderá ser realizada de acordo com as necessidades de cada usuário. Por exemplo, um usuário que deseja utilizar uma planilha eletrônica não necessita de uma capacidade de processamento similar a um usuário que deseja utilizar um aplicativo de CAD. A atualização ou a instalação de novos aplicativos também é facilitada. Para isso, basta instalar o aplicativo somente nos ambientes onde ele seja necessário. Após a instalação, esse aplicativo será disponibilizado e a atualização será visível a todos os usuários que utilizem esse ambiente (VAQUERO et al., 2009).

### 2.1.3 Desvantagens de Computação nas Nuvens

Entretanto, a utilização dessa abordagem centralizada acaba criando algumas desvantagens. Justamente por ser centralizado em poucas máquinas físicas, qualquer problema que ocorra em um dos nodos da infra-estrutura da nuvem (caso não seja criado um suporte de tolerância a falhas) pode acarretar em problemas em toda a nuvem e, conseqüentemente, a indisponibilidade dos serviços oferecidos pela mesma (VAQUERO et al., 2009).

Além disso, é importante ressaltar o fato de não ser possível garantir a disponibilidade em 100% do tempo, pois, além de falhas na infra-estrutura da nuvem, podem ocorrer falhas que não sejam de responsabilidade da nuvem, como, por exemplo, uma falha na empresa de telefonia que atenda o usuário, impedindo que esse se comunique com a nuvem e, conseqüentemente, acesse a mesma.

Outro aspecto importante consiste na qualidade de conexão utilizada pelo usuário, uma vez que todo o acesso à uma nuvem computacional é realizado através dessa. Se o usuário utilizar uma conexão de baixa qualidade, o serviço oferecido também terá a sua performance prejudicada, ou até mesmo, se tornará indisponível (MILLER, 2009).

Por fim, com o gerenciamento centralizado, a segurança torna-se um dos principais aspectos em uma nuvem computacional. Se uma pessoa não autorizada obtiver acesso à nuvem, ela poderá também ter acesso à informações de outros usuários. Desta forma, uma nuvem computacional deve ser provida de diferentes mecanismos de forma a garantir a segurança da mesma (LEAVITT, 2009).

Apesar dessas desvantagens, a computação nas nuvens tem sido amplamente utilizada pois permite um melhor aproveitamento dos recursos computacionais e, conseqüentemente, uma diminuição dos custos das empresas. Na próxima seção, serão descritos os serviços mais conhecidos e que foram desenvolvidos seguindo o paradigma de computação nas nuvens.

## 2.2 Serviços Baseados em Computação Nas Nuvens

O conceito de computação em nuvem despertou o interesse de muitas organizações que acabaram fazendo altos investimentos no desenvolvimento da sua própria nuvem. Entretanto, diferentemente do que ocorreu com outros paradigmas, algumas empresas tiveram a visão de que poderiam obter lucros através do uso de nuvens computacionais. O fato de poder disponibilizar serviços, através da *internet*, com performance similar a um *software* executado em um PC, foi o ponto chave para que algumas empresas alterassem o foco de seus produtos e disponibilizassem novos serviços através desse paradigma. Dentro desse contexto, surgiram diversos serviços baseados em computação nas nuvens.

Diversos desses serviços são oferecidos de forma gratuita, entretanto outras empresas, além de oferecerem esses serviços de forma gratuita, possuem planos em que, através do pagamento de uma taxa, o usuário tem acesso aos mesmos serviço, porém, contando com mais recursos para a sua utilização e, principalmente, uma garantia de disponibilidade desse serviço.

Com a contratação de um serviço, tem-se, por parte do usuário, uma maior exigência com relação a disponibilidade e a qualidade desse. Quando o usuário acessa um serviço gratuito esse possui uma certa tolerância a problemas, como, por exemplo, a indisponibilidade. Porém, ao utilizar um serviço pago, o usuário deseja uma garantia de disponibilidade, e até mesmo, se o serviço que foi contratado está realmente sendo oferecido por completo.

A fim de garantir ao cliente esses aspectos são feitos SLAs (acordos de níveis de serviços), onde a empresa prestadora inclui a garantia ter o serviço disponível por, no mínimo, um determinado tempo. Mesmo com essa garantia contratual, ainda devem existir mecanismos de QoS (qualidade de serviço) e um monitoramento contínuo da nuvem. Cabe ao administrador ajustar os recursos da nuvem para que sejam cumpridos os SLAs (BUYYA et al., 2008).

Outro fator muito importante a ser observado é o preço a ser cobrado pelo serviço. A empresa fornecedora de serviço deverá encontrar um valor que seja condizente com os seus custos e atrativos para as empresas contratantes (WALKER, 2009). Nas próximas seções serão descritas algumas das mais populares nuvens computacionais existentes.

### 2.2.1 *Google Apps*

O *Google Apps* (Google.com, 2009a) talvez seja o exemplo de nuvem computacional mais utilizada no mundo. Esse foi desenvolvido pela empresa *Google* e foi lançado em 2006, consistindo-se em um pacote de aplicativos de automação de escritórios muito similar ao *Microsoft Office*.

Inicialmente, o *Google Apps* contava com três aplicativos: uma agenda, um programa de mensagens instantâneas e um programa para criação de *websites*. Posteriormente, foram adicionados outros serviços, como, por exemplo, um editor de texto e uma planilha eletrônica. Atualmente, esses aplicativos encontram-se integrados ao *webmail* da *Google* (*Gmail*), permitindo assim a criação de documentos *online* e a possibilidade de compartilhamento de documentos entre os usuários.

Já em 2007, foi lançada a versão *Premier* do *Google Apps*, onde, além de oferecer os serviços da versão básica, o *Google* firma um SLA que garante a disponibilidade do serviço em 99.9% do tempo e suporte em tempo integral. Esta versão, atualmente, possui um custo anual de 50 dólares (*Google.com*, 2009a).

Finalmente, em 2008 a *Google* disponibilizou, dentro do *Google Apps*, um novo tipo de serviço, chamado *Google App Engine* (*Google.com*, 2009b). Este serviço disponibiliza aos usuários uma plataforma para o desenvolvimento de aplicações. A partir dessa plataforma o usuário pode desenvolver a sua própria aplicação e utilizar todo o potencial da nuvem computacional da *Google* para a execução da mesma. Além da plataforma, o serviço disponibiliza, ainda, diversas APIs de integração, não só com a própria plataforma, mas, também, com outros serviços disponibilizados pelo *Google Apps*, como por exemplo o *Gmail*. Atualmente, o *Google App Engine* permite hospedar aplicações desenvolvidas nas linguagens *Python* e *Java*. De maneira similar ao *Google Apps*, o *Google App Engine* apresenta uma versão gratuita e outra paga<sup>2</sup>, sendo que a versão gratuita possui restrições quanto ao número de aplicações e o tamanho dessas.

### 2.2.2 *Amazon Web Services*

A *Amazon.com* consiste em uma livraria virtual criada em 1994 por Jeff Bezos e a sua proposta inicial era que as pessoas, ao invés de irem até uma livraria para adquirir um livro, poderiam fazê-lo através da *internet*. Essa foi pioneira numa área onde hoje, existem cada vez mais empresas (*Amazon.com*, 2009). Entretanto, a proposta de ser uma simples biblioteca virtual não durou muito tempo e o catálogo de produtos oferecidos pela *Amazon.com* passou a conter diversos artigos, desde CDs e DVDs até produtos alimentícios e produtos automotivos.

Já no ano de 2006, a *Amazon.com* propôs mais uma inovação, ao criar a sua própria nuvem computacional, a AWS (*Amazon Web Services*) (*Amazon Web Services*, 2009), onde o usuário paga uma taxa correspondente ao serviço utilizado, ou seja, o pagamento é realizado sob demanda. Alguns dos tipos de serviço que a AWS oferece são: *Amazon SimpleDB*, que é um serviço que disponibiliza um banco de dados a partir de uma nuvem computacional; *Amazon S3* (*Amazon Simple*

---

<sup>2</sup>Uma tabela contendo os valores da versão paga do *Google App Engine* pode ser encontrada em <<http://code.google.com/intl/pt-BR/appengine/docs/billing.html>>.

*Storage Service*), que consiste em um serviço de armazenamento de arquivos em uma nuvem computacional; e o *Amazon SQS* (*Amazon Simple Queue Service*), que é um serviço para o armazenamento e a organização de mensagens e que é utilizado para a integração de aplicações.

Além desses serviços, a AWS disponibiliza ainda a *Amazon EC2* (*Amazon Elastic Compute Cloud*) (Amazon Elastic Compute Cloud, 2009). Através dele, o usuário tem acesso aos recursos computacionais disponíveis na nuvem computacional da *Amazon.com*. A partir da *Amazon EC2*, tem-se acesso a um servidor virtualizado dentro da nuvem computacional, sendo que atualmente estão disponíveis 5 configurações de servidores, que são:

- *Small Instance*: plataforma de 32 bits com processamento equivalente à um processador virtual com um único núcleo<sup>3</sup>. Essa plataforma possui 1.7 Gb de memória RAM e 160 Gb de disco rígido;
- *Large Instance*: plataforma de 64 bits com processamento equivalente à dois processadores virtuais, sendo que cada um deles possui dois núcleos. Além disso, essa plataforma apresenta 7.5 Gb de memória RAM e 850 Gb de disco rígido;
- *Extra Large Instance*: plataforma de 64 bits com processamento equivalente à dois processadores virtuais, sendo cada um deles com 4 núcleos. Essa possui, ainda, 15 Gb de memória RAM e 1.65 Tb de disco rígido;
- *High-CPU Medium Instance*: plataforma de 32 bits com poder de processamento equivalente à 2.5 processadores virtuais, com dois núcleos cada. A plataforma possui ainda 1.7 Gb de memória RAM e 350 Gb de disco rígido;
- *High-CPU Extra Large Instance*: plataforma de 64 bits com processamento equivalente à 2.5 processadores virtuais, sendo cada um deles possui oito núcleos. Além disso, essa plataforma possui 7 Gb de memória RAM e 1.65 Tb de disco rígido.

Além das configurações básicas oferecidas pela *Amazon EC2*, o usuário pode contratar serviços extras. Porém, esses serão cobrados a parte conforme o serviço contratado. Por exemplo, um usuário poderá contratar espaço extra de armazenamento. Nesse caso, será cobrado conforme o espaço de armazenamento necessário e o volume de transferência de dados.

Independente da configuração escolhida, o usuário poderá optar pelo sistema operacional a ser utilizado, sendo que, atualmente, estão disponíveis os seguintes

---

<sup>3</sup>Cada núcleo de processador virtual equivale a um processador *Opteron* com *clock* de 1 Ghz.

sistemas operacionais: *Windows Server 2003*, *OpenSolaris* e diversas distribuições de *Linux*, como, por exemplo, *Ubuntu*, *Red Hat* e *Fedora* (Amazon Elastic Compute Cloud, 2009).

Da mesma forma que os demais serviços da AWS, o pagamento pelo serviço do *Amazon EC2* é realizado através do princípio de *pay-per-use*, ou seja, pague somente pelo que for consumido. Nesse caso, a cobrança é feita através do tempo de processamento utilizado. Outros fatores que influenciam no preço do serviço contratado são o tipo de sistema operacional escolhido pelo usuário (Windows ou Unix/Linux), e se a máquina virtual ficará hospedada (fisicamente) na Europa ou nos Estados Unidos. Na Tabela 2.1, são listados os valores hora, de acordo com a configuração e o sistema operacional para máquinas virtuais localizadas nos Estados Unidos<sup>4</sup>.

Tabela 2.1: Valor hora de uma máquina virtual na Amazon EC2

Configuração	Linux/Unix	Windows
Small Instance	U\$ 0.10	U\$ 0.125
Large Instance	U\$ 0.40	U\$ 0.50
Extra Large Instance	U\$ 0.80	U\$ 1.00
High-CPU Medium Instance	U\$ 0.20	U\$ 0.30
High-CPU Extra Large Instance	U\$ 0.80	U\$ 1.20

Além do sistema operacional utilizado, o usuário pode configurar a sua máquina virtual com aplicativos de acordo com a sua necessidade. Estão disponíveis, por exemplo: os bancos de dados IBM DB2, *Oracle 11G* e *Microsoft SQL Server 2005*; os servidores de aplicação IBM *WebSphere* e *Oracle WebLogic*; e as plataformas de aplicação *JBoss* e *Ruby on Rails*.

Como exemplo de empresas que utilizam-se dos serviços da *Amazon EC2* pode-se citar-se: *Harvard Medical School*, que utiliza a *Amazon EC2* para a execução de simulações com modelos genéticos; *Washington Post*, que utiliza a *Amazon EC2* para hospedar uma ferramenta de pesquisa de notícias e documentos correspondentes a edições antigas do jornal; e a *Virgin Atlantic Airways*, que possui, hospedado na *Amazon EC2*, um site comemorativo aos 25 anos da empresa.

### 2.2.3 openQRM

O openQRM foi lançado pela empresa *Qlusters* em 2006, como sendo um *software* para gerenciamento de *datacenters*. Atualmente, ele é o um *software* de código aberto sob a licença *GPL* (openQRM, 2009). O openQRM possui, além de ferramentas para o gerenciamento de *datacenters*, funcionalidades que facilitam o

<sup>4</sup>Os preços dessas configurações máquinas virtuais localizadas na Europa pode ser encontrado em <http://aws.amazon.com/ec2/>.



desenvolvimento de uma nuvem computacional.

O projeto openQRM disponibiliza ainda, gratuitamente, através do seu *website*, a possibilidade da criação de máquinas virtuais para avaliação dentro de sua própria nuvem computacional. Nesse caso, as máquinas virtuais ficam ativas por um período de tempo determinado no momento da sua criação, sendo que o período máximo é de trinta dias. Além disso, caso essas não sejam utilizadas, os recursos são automaticamente desalocados pela nuvem. Como opções de sistema operacionais, o openQRM, possibilita que as máquinas virtuais criadas utilizem algumas distribuições de *Linux*, como, por exemplo: *Ubuntu*, *Debian* e *Fedora*.

Como é uma aplicação somente para avaliação, os recursos computacionais disponíveis são limitados. Por exemplo, as máquinas virtuais podem ser criadas com somente um processador, 250 Mb de memória e 5 Gb de disco rígido. Além disso, o acesso à nuvem é bastante restrito, sendo disponibilizados um pequeno número de contas de acesso que são criadas somente após uma avaliação da solicitação. Atualmente não é disponibilizada nenhuma versão paga deste serviço.

## 2.3 Considerações Finais

Neste capítulo, foi feita uma breve descrição do paradigma de computação nas nuvens. Inicialmente, foram vistos alguns conceitos sobre nuvens computacionais e, após, foi realizada uma descrição básica da infra-estrutura de uma nuvem computacional. Além disso, foram apresentadas vantagens e desvantagens provenientes da utilização de nuvem computacional. Por fim, foram apresentados alguns serviços baseados em computação nas nuvens existentes na atualidade.

Atualmente, observa-se um aumento no número de ofertas baseadas no modelo de computação nas nuvens. Isso deve-se a possibilidade da utilização desse modelo como uma ferramenta para a diminuição de custos nas organizações. Além disso, esse modelo tem atraído atenção de grandes empresas com o intuito de expandir os seus negócios através da criação da sua própria nuvem computacional e o fornecimento de serviços a partir dessa. Para maiores informações sobre computação nas nuvens, sugere-se (BUYA et al., 2008), (VAQUERO et al., 2009) e (MILLER, 2009).

No próximo capítulo será feito um detalhamento sobre o gerenciamento da infra-estrutura de uma nuvem computacional, apresentando ferramentas que podem ser utilizadas para o desenvolvimento de uma infra-estrutura de nuvem. Esse capítulo tem como objetivo familiarizar o leitor de forma a facilitar a compreensão do ambiente criado nesse trabalho.

## 3 GERENCIAMENTO DE UMA NUVEM COMPUTACIONAL

Para o desenvolvimento de uma infra-estrutura de uma nuvem computacional são necessários três componentes. Inicialmente, são necessários computadores interconectados, que serão responsáveis por fornecer os recursos que serão utilizados pela nuvem. Para gerenciar esses computadores, será necessária a utilização de um *software* gerenciador. Por fim, será necessário um serviço, para que a nuvem computacional possa provê-lo.

Na próxima seção, serão descritos alguns aspectos sobre a infra-estrutura física necessária para a criação da nuvem computacional. Além disso, na Seção 3.2 serão apresentadas algumas características de *softwares* gerenciadores para nuvens computacionais. Por fim, nas Seção 3.2 são apresentados alguns gerenciadores disponíveis.

### 3.1 Infra-Estrutura Física

A infra-estrutura física de uma nuvem computacional é desenvolvida a partir da interconexão, através de uma rede de computadores, de diversos nodos. Cada um dos nodos será responsável por prover os recursos computacionais que serão utilizados pela nuvem. Esses nodos serão responsáveis por hospedar as máquinas virtuais (MV), que irão fornecer os recursos computacionais necessários para o funcionamento da nuvem computacional (BUY YA; YEO; VENUGOPAL, 2008).

Frequentemente, esses nodos são organizados em forma de *clusters*, sendo necessário a utilização de um desses nodos para o controle do *cluster*. Esse nodo recebe o nome de *Cluster Controller*, ou, Controlador de *Cluster* (CC), e é responsável por reunir informações sobre os nodos do *cluster* e gerenciá-los. Também fica a cargo desse verificar a disponibilidade dos nodos que formam o *cluster*, ou seja, esse deverá verificar a possibilidade de algum nodo atender uma determinada requisição. Em uma nuvem computacional, podem existir vários *clusters*, sendo que cada nodo só poderá pertencer à um *cluster* (NURMI et al., 2009a).

Para gerenciar os *clusters*, bem como toda a infra-estrutura da nuvem, será necessário um nodo, chamado de *Cloud Controller*, ou, Controlador da Nuvem (CLC). Esse deverá manter as informações sobre cada um dos nodos da nuvem computacional. É através do CLC que a nuvem computacional é acessada pelos usuários. Assim, esse é responsável por atender todas as requisições da nuvem e encaminhar as solicitações dos usuários para os CCs. Em uma nuvem computacional só existirá um único nodo controlador da nuvem. A Figura 3.1 demonstra a infra-estrutura descrita, bem como as interligações entre os seus componentes.

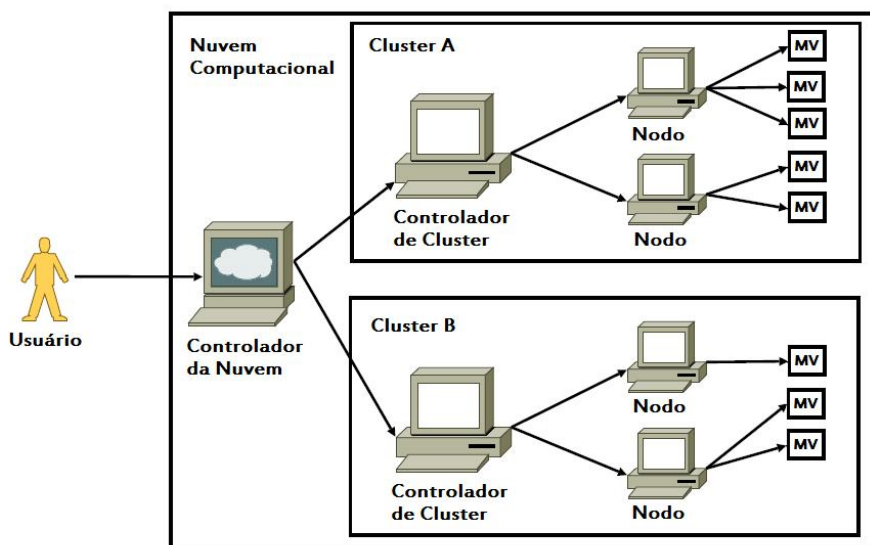


Figura 3.1: Infra-Estrutura de uma Nuvem Computacional

Apesar de estar representada de forma distribuída na Figura 3.1, essa infra-estrutura pode ser simplificada. Uma mesma máquina pode, por exemplo, desempenhar as funções de controlador da nuvem, controlador do *cluster* e de um nodo que prove recursos a nuvem. Entretanto, essa abordagem não é recomendada, pois ela cria uma sobrecarga de funções em uma mesma máquina, o que pode acarretar em problemas de performance da nuvem. Uma estratégia que pode ser adotada, dependendo do tamanho da nuvem, é de unir as funções de controlador da nuvem e do *cluster* num único nodo (SOTOMAYOR et al., 2009).

A utilização de uma infra-estrutura distribuída traz um benefício para a segurança da nuvem. Como os controladores de *cluster* e os nodos não são visíveis no exterior da nuvem, eles são menos suscetíveis à ataques que o controlador da nuvem. Essa infra-estrutura distribuída permite, ainda, que, fisicamente, os componentes da nuvem estejam distribuídos em localizações geográficas diferentes como ocorre, por exemplo, com a *Amazon EC2*, que, atualmente, possui recursos localizados nos Estados Unidos e na Europa.

Além disso, da mesma forma que podem ser adicionados nodos que não estejam

localizados no mesmo local que o controlador, uma nuvem pode ser combinada com outra e, assim, prover recursos para outra nuvem. Por exemplo, uma máquina virtual de uma determinada nuvem computacional ser utilizada como nodo de outra.

Por fim, separar cada controlador em computadores diferentes facilita, ainda, a manutenção em qualquer ponto da nuvem, uma vez que não será necessário tornar indisponível toda a nuvem, mas sim, somente, um computador que necessite de manutenção.

## 3.2 Gerenciador de Uma Nuvem Computacional

Para o gerenciamento de uma nuvem computacional, frequentemente, utiliza-se de um *software* gerenciador, que, em geral, é dividido em três pontos: gerenciador dos nodos, gerenciador do *cluster* e gerenciador da nuvem computacional.

O *software* gerenciador dos nodos da nuvem deverá ser capaz de criar e desalocar as instâncias de máquinas virtuais existentes. Além disso, esse deve ser capaz de inspecioná-las, de forma a obter informações a respeito das instâncias que estão ativas em um determinado momento. Entretanto, esse controlador de nodo somente irá realizar essas operações mediante a uma requisição proveniente do controlador de *cluster*.

Já o *software* controlador de *cluster* deverá gerenciar os *clusters* formados através da união dos nodos. Esse deverá ser capaz de obter informações sobre os estados de cada um dos nodos, podendo enviar-lhes comandos para a criação e desalocação de instâncias de máquinas virtuais. Da mesma forma que o controlador de nodo, o controlador de *cluster* também irá receber comandos do controlador da nuvem. Entretanto, ficará a cargo do controlador de *cluster* decidir, se for o caso, qual nodo que será alocado para o atendimento de uma determinada requisição. Por exemplo, no caso de uma requisição para a criação de uma instância de máquina virtual, o controlador de *cluster* deverá verificar qual nodo possui recursos disponíveis para a criação dessa instância.

Por fim, fica a cargo do *software* controlador da nuvem o trabalho de receber as requisições dos usuários da nuvem e atendê-las. Esse será responsável, ainda, por realizar a autenticação do usuário e verificar as permissões que esse usuário possui, além de executar rotinas administrativas na nuvem. Outra funcionalidade que o *software* gerenciador da nuvem deverá prover é um suporte a tolerância a falhas. Caso ocorra um problema em algum ponto da infra-estrutura da nuvem, esse deverá ser capaz de contornar esse problema de forma que a nuvem não fique comprometida e o defeito possa ser contornado de forma transparente aos usuários. Fica, ainda, sob responsabilidade do *software* o controle do QoS e dos SLAs dos usuários. De posse dessas informações, o *software* deverá tentar ajustar os recursos de forma que

o SLA de cada usuário seja cumprido. Por fim, caso os recursos providos pela nuvem sejam comercializados, o gerenciador deverá ainda incluir mecanismos que permitam a medição dos recursos utilizados. Essas medições serão utilizadas como base para a cobrança pelos serviços fornecidos pela nuvem (BUY YA; YEO; VENUGOPAL, 2008).

Atualmente, não existem muitos *softwares* gerenciadores de nuvem computacional disponíveis. Nas próximas seções, serão descritos os *softwares* gerenciadores mais conhecidos e utilizados no desenvolvimento de nuvens computacionais.

### 3.2.1 Eucalyptus

O *Eucalyptus* teve o seu desenvolvimento iniciado em fevereiro de 2008, nos laboratórios do departamento de Ciências da Computação, da Universidade da Califórnia. Esse foi o resultado de uma pesquisa realizado pelo projeto *Virtual Grid Application Development Software Project* (vGrADS), da Fundação Nacional de Ciências dos Estados Unidos. O propósito desse projeto consistia em investigar técnicas para o desenvolvimento e execução de rotinas em *grids* computacionais. Ao final, o resultado desse projeto foi um ambiente com uma plataforma virtualizada para a execução dessas rotinas. Nesse, cada tarefa era executada em um ambiente virtualizado e dedicado. Esse projeto procurou, ainda, realizar uma integração entre os computadores da Fundação Nacional de Ciências com a nuvem computacional disponibilizada pela *Amazon* (AWS) (Eucalyptus, 2009b).

A primeira versão do *Eucalyptus* foi lançada em maio de 2008, sob a licença GPL e permitia a criação e o gerenciamento de uma nuvem computacional, desenvolvida utilizando-se de Linux. Para isso, essa ferramenta utiliza-se exclusivamente de softwares de código aberto. Inicialmente, esse continha algumas restrições, como por exemplo, não permitia a criação de modelos de ambiente, ou seja, esse disponibilizava somente um ambiente e não permitia a sua alteração. Além disso, esse permitia somente a utilização do *Xen* para o gerenciamento das máquinas virtuais dos nodos.

Uma vez que o *Eucalyptus* deveria ser capaz de interagir com a nuvem da AWS, o seu funcionamento também é similar ao da AWS. Essa similaridade permite que os usuários que utilizam a AWS passem a utilizar o *Eucalyptus* sem maiores dificuldades. Atualmente, todas as melhorias que estão sendo feitas no *Eucalyptus* tem como objetivo implementar a maior parte das funcionalidades existentes na AWS. O *Eucalyptus* possui, ainda, diversas APIs que podem ser utilizadas para a integração de uma nuvem computacional com a AWS, permitindo, por exemplo, que máquinas virtuais da AWS possam ser utilizadas como nodos de uma nuvem criada a partir do *Eucalyptus*.

Logo após o lançamento do *Eucalyptus*, houve, também, o lançamento da

*Eucalyptus Public Cloud*<sup>1</sup>, que consiste em uma versão pública da *Eucalyptus*, onde o usuário tem a disposição um ambiente *web* que permite a avaliação do *Eucalyptus*. Como essa nuvem é somente utilizada para avaliação, ela possui algumas limitações, como, por exemplo, só permite a criação de 4 instâncias de máquinas virtuais simultaneamente, sendo que cada uma delas ficará ativa por, no máximo, 6 horas (Eucalyptus, 2009c).

Atualmente, o *Eucalyptus* permite também a utilização do hipervisor KVM para a criação das instâncias das máquinas virtuais nos nodos, além da utilização do *Xen*. O *Eucalyptus* também foi incorporado ao *Ubuntu*, a partir da versão 9.10, permitindo que o *Ubuntu* seja utilizado na criação de um ambiente de uma nuvem computacional. Além disso, ele também está disponível para outras distribuições do Linux, como, por exemplo: *Suse* e *Debian*.

Além da versão gratuita e de código aberto, existe uma versão paga do *Eucalyptus* chamada *Eucalyptus Enterprise Edition*<sup>2</sup>. Além de todas as funcionalidades existentes na versão gratuita, a versão paga permite ao usuário a possibilidade de utilizar o *VMware* para a criação de máquinas virtuais nos nodos, bem como migrar todas as máquinas virtuais desenvolvidas por esse para uma nuvem computacional. Além dessas funcionalidades, os usuários dessa versão paga podem utilizar-se de um suporte prestado pela empresa (Eucalyptus, 2009d)<sup>3</sup>.

### 3.2.2 OpenNebula

O *OpenNebula* foi desenvolvido pelo *Distributed Systems Architecture Research Group* (DSA) da *Universidad Complutense de Madrid*, como parte do projeto *Resources and Services Virtualization Without Barriers* (RESERVOIR), da União Europeia. Esse projeto teve, como principal objetivo, desenvolver uma plataforma que disponibilizasse serviços através de uma rede, utilizando-se para isso de virtualização, provisionando e alocando serviços sob demanda (RESERVOIR, 2009). Atualmente, esse projeto é financiado por grandes empresas do ramo da informática e das telecomunicações, como, por exemplo, IBM, Sun, Telefonica e SAP.

A primeira versão do *OpenNebula* foi lançada em julho de 2008, sob a licença *Apache v2.0*, para as plataformas *Linux* e *Mac OS X*. Esse permitia a alocação dinâmica de máquinas virtuais num *cluster* de máquinas físicas provendo recursos para uma nuvem computacional. O *OpenNebula* foi desenvolvido de forma que suas funcionalidades fossem separadas em módulos. Dessa forma, facilitando ao

---

<sup>1</sup>A *Eucalyptus Public Cloud* pode ser acessada através do endereço <<https://mayhem9.cs.ucsb.edu:8443/>>

<sup>2</sup>No *website* do *Eucalyptus* não são citados valores para essa versão paga. Entretanto, é oferecida a possibilidade da realização de uma prova de conceito do *Eucalyptus*, ao custo de 499 dólares

<sup>3</sup>Maiores informações sobre o *Eucalyptus* podem ser encontrados em <<http://www.eucalyptus.com/>>

administrador da nuvem a incorporação ou remoção de um determinado componente ou funcionalidade.

Os módulos do *OpenNebula* foram criados a partir da utilização de diversos *softwares* de código aberto. Um exemplo dos softwares que auxiliam no gerenciamento da nuvem é o *Haizea*, que permite gerenciar um *cluster* de computadores, realizando o provisionamento e a alocação dos recursos de acordo com a demanda, utilizando-se, para isso, de máquinas virtuais. O *Haizea* permite a utilização dos seguintes hipervisores: *Xen*, *VMware* e KVM (Haizea, 2009).

Outra funcionalidade importante que existe no *OpenNebula* é a possibilidade de integração da nuvem computacional com outras nuvens, como, por exemplo, a AWS. Esse permite, por exemplo, que máquinas virtuais da AWS sejam utilizadas como nodos de uma nuvem.

Desde o seu lançamento, não foram adicionadas muitas novas funcionalidades no *OpenNebula*, sendo que somente foram feitas algumas melhorias em funcionalidades já existentes, como, por exemplo, o gerenciamento da rede formada pelos *clusters* (OPENNEBULA.ORG, 2009)<sup>4</sup>.

### 3.2.3 Citrix Cloud Center

O *Citrix Cloud Center* (C3) é o *software* gerenciador de nuvem computacional desenvolvido pela empresa *Citrix Systems*<sup>5</sup>. Essa é uma empresa norte-americana fundada em 1989, cujo os primeiros *softwares* eram voltados para o sistema operacional OS/2 da IBM. Mas foi somente em 1993 que essa lançou o seu primeiro *software* de sucesso comercial. Esse era chamado de *WinView* e provia acesso remoto a computadores que utilizassem o sistema operacional MS-DOS e o aplicativo *Windows 3.1* da *Microsoft* (HARWOOD, 2004). O sucesso obtido pela *Citrix* com o *WinView* fez com que a *Microsoft* e a *Citrix* selassem um acordo, no qual a *Citrix* licenciou o *WinView* para a *Microsoft*, de forma que esse fosse agregado ao *Windows* com o nome de *Windows Terminal Server*. Em contrapartida, a *Citrix* obteve o direito de vender *softwares* complementares ao *Windows*, lançados sob o nome de *Metaframe*.

O envolvimento da *Citrix* com virtualização teve início em 2007, quando essa adquiriu a *XenSource*, empresa que vendia o hipervisor *Xen*,. Esse foi desenvolvido na Universidade de Cambridge e lançado em 2003. Posteriormente, a *Citrix* ainda desenvolveu alguns produtos voltados a virtualização e ao *Xen* (XEN, 2009).

Já em 2008, a *Citrix* lançou o C3, que consiste em um *software* proprietário com a proposta de gerenciar a infra-estrutura de uma nuvem computacional (CITRIX,

---

<sup>4</sup>Maiores informações sobre o *OpenNebula* podem ser encontrados em <<http://www.opennebula.org/>>

<sup>5</sup>No website do C3, não são citados quais são os valores para a aquisição do *software*.

2009a). Esse foi desenvolvido baseado no hipervisor *Xen* e traz ferramentas desenvolvidas pela Citrix que facilitam a criação e o gerenciamento de uma nuvem computacional. Como exemplo, pode-se citar o *Citrix Xen Server Cloud Edition*, que prove o serviço de infra-estrutura para a nuvem computacional.

Outra funcionalidade que o C3 provê é uma facilidade para a migração de máquinas virtuais que utilizem o *XenServer*. Da mesma forma, esse provê compatibilidade com o *Windows Hiper-V*, de forma que uma máquina virtual criada a partir do Hiper-V possa ser transportada facilmente para uma nuvem computacional desenvolvida utilizando-se do C3.

Como desvantagem do C3, em relação ao *Eucalyptus* e ao *OpenNebula*, pode-se citar o fato de que esse não possui funcionalidades para a integração da nuvem desenvolvida à outras já existentes.

### 3.2.4 VMware vSphere

A empresa *VMware* foi criada em 1998 e tornou-se rapidamente a líder mundial em *softwares* para virtualização. O seu primeiro produto, o *VMware Workstation*, lançado em 1999, foi o responsável pela popularização da empresa. Esse tinha como objetivo prover um ambiente computacional através da utilização de virtualização (VMWARE, 2009b).

Com a popularização da virtualização, a *VMware* lançou, em 2001, o *VMware Server*, um *software* para virtualização de servidores. Posteriormente, a empresa lançou, também, outros produtos voltados à virtualização e alguns complementos aos produtos já existentes. Em 2003, foi lançado o *VMware Infrastructure*, um software que provê soluções para virtualização de computadores pessoais e de servidores, integrando o *VMware Workstation* e o *VMware Server*, além de apresentar melhorias nesses dois *softwares*.

Já em 2009, a *VMware* lançou uma nova versão do *VMware Infrastructure*, o *VMware vSphere* (VMWARE, 2009a). Esse, além de ser uma nova versão do *VMware Infrastructure*, possui recursos que permitem o gerenciamento de uma nuvem computacional. Esse é um *software* proprietário e que foi desenvolvido de forma modularizada, com o objetivo de disponibilizar versões diferentes do *VMware vSphere*, agregando ou não os módulos, conforme a versão adquirida. As diferentes versões existentes são<sup>6</sup>:

- *Standard*: versão disponível para utilização em servidores com processadores contendo até 6 núcleos e no máximo 256 Gb de memória RAM. Essa versão possui as funcionalidades básicas de gerenciamento da nuvem computacional, como, por exemplo, gerenciamento centralizado, controle de disponibilidade e

---

<sup>6</sup>Mais informações sobre as versões do *VMware vSphere* podem ser encontradas em <http://www.vmware.com/vmwarestore/vsphere-purchaseoptions.html>.



provisionamento de recursos. Essa versão é comercializada por US\$ 1,118.00 por processador;

- *Advanced*: versão disponível para utilização em servidores com processadores contendo até 12 núcleos e no máximo 256 Gb de memória RAM. Essa versão possui as mesmas funcionalidades presentes na versão *Standard*, porém, agregando, também, as funcionalidades de tolerância a falhas e de migração de máquinas virtuais dentro da nuvem. Essa versão é comercializada por US\$ 2,806.00 por processador;
- *Enterprise*: versão disponível para a utilização em servidores com processadores contendo até 6 núcleos e no máximo 256 Gb de memória RAM. Essa versão possui as funcionalidades presentes na versão *Advanced*, agregando, também, as funcionalidades de migração de áreas de dados e controle de recursos e de energia. Essa versão é comercializada por US\$ 3,594.00 por processador;
- *Enterprise Plus*: versão disponível para utilização em servidores com processadores contendo até 12 núcleos e no máximo 256 Gb de memória RAM. Essa versão possui, além das funcionalidades presentes na versão *Enterprise*, suporte a criação de perfis para usuários e de gerenciamento de redes. Atualmente, essa versão é comercializada por US\$ 4,369.00 por processador.

Da mesma forma o que o C3, o *VMware vSphere* não provê funcionalidades para a integração da sua nuvem computacional com outras nuvens computacionais. Entretanto, como o *VMware vSphere* é uma evolução do *VMware Infrastructure*, ele apresenta facilidades para a migração de máquinas virtuais criadas pelo *VMware Infrastructure* para a nuvem gerenciada pelo *VMware vSphere*.

### 3.3 Considerações Finais

Neste capítulo, foi feita uma breve descrição sobre um ambiente computacional baseado em *cloud computing*. Inicialmente, foi feita uma definição da infraestrutura necessária para a criação de uma nuvem computacional. Depois, foi feita uma descrição sobre o *software* que irá gerenciar a infra-estrutura da nuvem computacional. Por fim, foram apresentados *softwares* utilizados para a criação a gerenciamento de nuvem computacional. Para maiores informações, sugere-se (BUYYA; YEO; VENUGOPAL, 2008) e (NURMI et al., 2009b)

A complexidade da infra-estrutura de uma nuvem computacional será proporcional ao tamanho e a capacidade de processamento que a nuvem deverá

prover. Independente dessa complexidade, o *software* gerenciador deverá ser capaz de gerenciar essa infra-estrutura de maneira eficaz. Um gerenciamento mal feito pode, por exemplo, provocar uma subutilização dos recursos da nuvem.

Como citado anteriormente, a base para a criação de uma nuvem computacional computacional consiste na utilização de uma técnica chamada virtualização. Dessa forma, no Capítulo 4 é feito um detalhamento dessa técnica, para que, posteriormente, seja descrita a infra-estrutura desenvolvida.

## 4 VIRTUALIZAÇÃO

Como visto nos capítulos anteriores, virtualização é a principal técnica utilizada na construção de uma infra-estrutura de nuvem computacional. Dessa forma, optou-se, nesse capítulo, por descrever de forma um pouco mais detalhada essa técnica. Na próxima seção, será apresentado o conceito de virtualização. Na Seção 4.1.1 serão apresentadas as diferentes técnicas que podem ser utilizadas no desenvolvimento de máquinas virtuais. Além disso, nas Seção 4.1.2 serão apresentadas vantagens e desvantagens da utilização de virtualização em uma nuvem computacional. Já na Seção 4.2 serão apresentados alguns monitores de máquinas virtuais existentes.

### 4.1 Máquinas Virtuais

Apesar da utilização da virtualização ter se tornado popular há poucos anos, ela já é utilizada desde a década de 60. As primeiras aplicações virtualizadas foram desenvolvidas pela IBM com a criação de um sistema operacional chamado M44/44X. O principal objetivo do M44/44X era fornecer, a cada usuário, um ambiente independente dos demais usuários, permitindo, desta forma, a instalação nesse ambiente do seu próprio sistema operacional e seus próprios aplicativos (LAUREANO; MAZIERO, 2008).

Os altos custos relacionados aos primeiros sistemas virtualizados e a popularização dos PCs, nas décadas seguintes, acabou provocando uma diminuição na utilização de virtualização. Porém, com as constantes evoluções, os computadores atuais passaram a ter um poder computacional que muitas vezes é subutilizado. Esses são providos de processadores *multi core*, que passam a maior parte do tempo ociosos, em geral, alcançando picos de utilização somente em alguns períodos do dia. Assim, é possível aumentar o uso desses computadores através da utilização técnicas de virtualização, uma vez que essa permite dividir os recursos desses computadores entre diversos ambientes (SEO, 2009).

Segundo Carrisimi (CARISSIMI, 2009), para que uma máquina virtual funcione corretamente, independentemente da técnica utilizada para a implementação da

mesma, ela deve possuir as seguintes propriedades:

- **Equivalência:** a máquina virtual deverá ser capaz de criar um ambiente virtual com o mesmo comportamento e as mesmas características de uma máquina real, ou seja, o sistema operacional que for instalado na máquina virtual hóspede<sup>1</sup> deverá funcionar da mesma maneira que se estivesse instalado na máquina real;
- **Controle de Recursos:** somente a máquina virtual deverá possuir o controle dos recursos da máquina real. O sistema operacional hóspede só poderá acessar os recursos que a máquina virtual disponibilizar à ele;
- **Eficiência:** a maior parte das instruções deverá ser executada diretamente pelo processador da máquina real. Quando isto não for possível, as instruções deverão ser convertidas pela máquina virtual de forma que o processador da máquina real seja capaz de executar essas instruções;

Nas próximas seções serão apresentadas as técnicas existentes para o desenvolvimento de máquinas virtuais, bem como suas respectivas vantagens e desvantagens. Por fim, serão apresentadas vantagens e desvantagens da utilização de virtualização em computação nas nuvens.

#### 4.1.1 Técnicas Para Desenvolvimento de Máquinas Virtuais

Duas técnicas podem ser utilizadas para o desenvolvimento de máquinas virtuais. A primeira delas, chamada de máquina virtual de processo, consiste em desenvolver um programa que permita realizar chamadas de sistemas do sistema operacional nativo ou utilizar uma biblioteca de funcionalidades da própria máquina virtual para realizar essas chamadas (CARISSIMI, 2007). Com esta técnica, por exemplo, quando uma aplicação necessita criar um novo arquivo, essa irá realizar uma chamada de um procedimento de criação de um arquivo à máquina virtual. Então, ficará sob responsabilidade da máquina virtual criar esse arquivo. Assim, uma aplicação pode ser construída de forma que ela não possua nenhum vínculo com o sistema operacional nativo. Como consequência disso, essa aplicação poderá ser executada em qualquer ambiente, bastando, para isso, possuir uma máquina virtual instalada. Um exemplo de máquina virtual que trabalha com esta técnica é a Máquina Virtual Java (*Java Virtual Machine* - JVM).

Outra técnica que pode ser utilizada para a criação de máquinas virtuais é chamada de máquina virtual de sistema e consiste em criar uma camada

---

<sup>1</sup>Sistema Operacional Hóspede será qualquer sistema operacional que seja executada em uma máquina virtual hospedeira.

de virtualização entre o *hardware* e o sistema operacional hospedeiro<sup>2</sup>. Nessa técnica, a camada de virtualização será a responsável por receber as instruções do sistema operacional da máquina virtual hóspede e interpretá-las de forma que essas possam ser executadas pelo *hardware* hospedeiro. Essa camada de virtualização também é conhecida como monitor de máquina virtual ou, ainda, como hipervisor (CARISSIMI, 2007). Na Figura 4.1 tem-se ilustradas as diferenças entre uma máquina virtual de processo e uma máquina virtual de sistema.

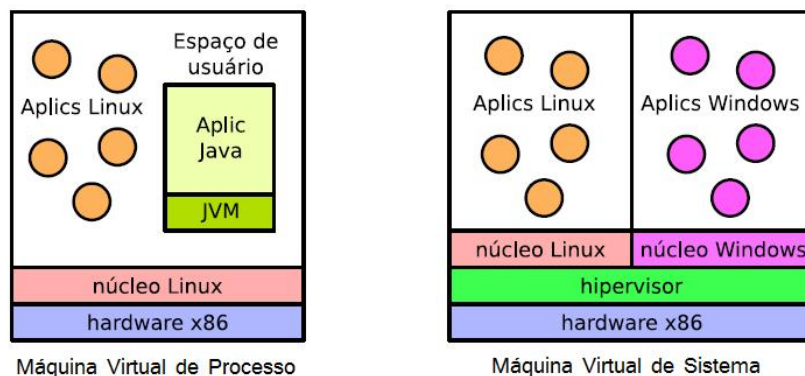


Figura 4.1: Tipos de Máquinas Virtuais

As máquinas virtuais de sistema podem ser implementadas utilizando duas estratégias distintas. Na primeira delas, chamada de virtualização total, a virtualização pode ser obtida através de um hipervisor ou, ainda, uma funcionalidade no próprio hardware do computador (CARISSIMI, 2009). Nessa, quando o sistema operacional hóspede necessita fazer uma chamada de sistema para um determinado dispositivo, essa será encaminhada para o hipervisor, que ficará responsável por "traduzir" essa chamada para o *hardware* da máquina virtual. No caso da virtualização total, todas as chamadas de sistemas realizadas são traduzidas pelo hipervisor.

Uma vantagem na utilização dessa estratégia consiste no fato de que qualquer sistema operacional pode ser utilizado como hospedeiro. Entretanto, isso possui algumas desvantagens, pelo fato de que o hipervisor deverá tratar todas as chamadas de sistemas que serão realizadas pelo hospedeiro, acarretando, dessa forma, em uma perda de desempenho. Outra desvantagem dessa abordagem consiste no fato de que o hipervisor só fornecerá funcionalidades genéricas para cada dispositivo, limitando, assim, as funcionalidades e o desempenho dos dispositivos. Alguns exemplos de hipervisores que utilizem essa técnica são: o *VMware* (VMWARE, 2009b), o *Microsoft Virtual PC* (MICROSOFT, 2009) e o *VirtualBox* (SUN, 2009). Na Figura 4.2 tem-se uma representação dessa estratégia.

<sup>2</sup>Sistema Operacional Hospedeiro será o sistema operacional onde será executado o *software* gerenciador da máquina virtual, utilizado para criar máquinas virtuais.

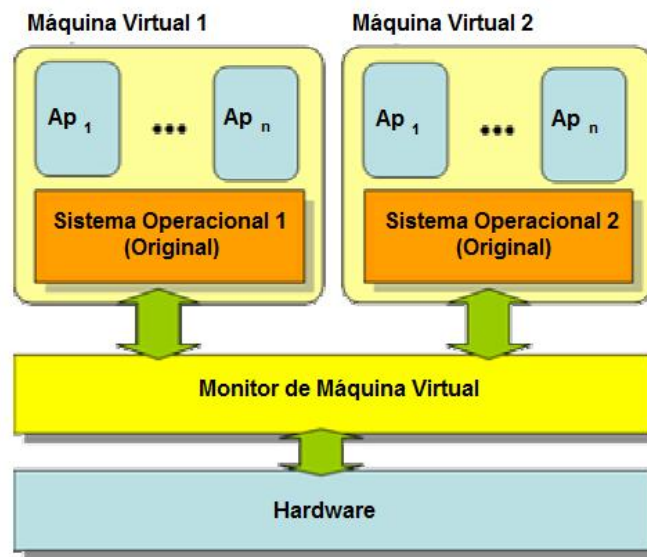


Figura 4.2: Virtualização Total

Outra estratégia que pode ser utilizada para a implementação de máquinas virtuais de sistema é chamada de paravirtualização. Essa estratégia diferencia-se da virtualização total pelo fato de que o hipervisor será responsável somente por realizar o tratamento de eventuais chamadas de sistemas que possam causar conflitos ou *traps*<sup>3</sup> (CARISSIMI, 2009).

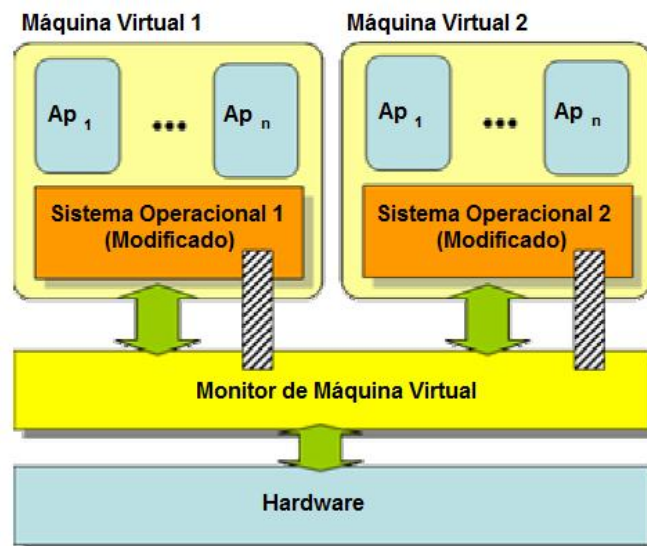


Figura 4.3: Paravirtualização

Para que uma máquina virtual seja criada utilizando a técnica de paravirtualização, o sistema operacional hóspede deve ser modificado para que o hipervisor seja responsável somente para tratar as chamadas de sistemas que possam

<sup>3</sup>*Traps* são interrupções que ocorrem na execução do sistema operacional causadas por alguma situação excepcional e que demandam um tratamento especial pelo sistema operacional

causar conflitos ou *traps*. As demais chamadas de sistemas, que não necessitem de tratamento, poderão ser feitas diretamente pelo sistema operacional hóspede.

Essa alteração que é feita no sistema operacional hospedeiro tem como vantagem um melhor desempenho se comparado a virtualização total. Isso porque nessa não é necessário que o hipervisor faça qualquer tipo de tratamento ou tradução das chamadas de sistemas, mas sim, somente gerenciar conflitos ou exceções que possam ocorrer, reduzindo, assim, o processamento necessário para a realização da chamada de sistema (CARISSIMI, 2009).

Uma desvantagem na utilização dessa estratégia consiste no fato de que o sistema hospedeiro deve ser alterado, de forma que as chamadas sejam feitas diretamente para o hipervisor. Assim, nem todos os sistemas operacionais poderão ser utilizados na criação das máquinas virtuais. Um exemplo de hipervisor que é implementado através dessa estratégia é o *Xen* (XEN, 2009). A Figura 4.3 demonstra esta estratégia.

#### 4.1.2 Vantagens e Desvantagens do Uso de Virtualização

Uma vantagem que a utilização de máquina virtual proporciona é que, por possuir todas as informações relativas ao seu estado atual, a máquina virtual pode salvar estas informações, permitindo, no futuro, que ela retorne a um estado anterior, no caso de, por exemplo, ocorrer uma falha, ou ainda realizar a migração desta máquina virtual de uma máquina hospedeira para outra. Estes pontos de salvamento são conhecidos como *checkpoints* (LAUREANO; MAZIERO, 2008). Através dessa funcionalidade, o gerenciador da nuvem computacional pode migrar as máquinas virtuais de um nodo para outro, facilitando, assim, qualquer manutenção que seja necessária na nuvem computacional, ou ainda, objetivando um melhor balanceamento de carga.

Além disso, a utilização de máquinas virtuais em uma nuvem computacional proporciona a possibilidade de, com a utilização de somente uma máquina física, oferecer vários ambientes heterôgeneos, pois, as máquinas virtuais não precisam, necessariamente, contar com as mesmas configurações (de *hardware*) e nem com o mesmo sistema operacional das outras máquinas virtuais ou, ainda, que a máquina hóspede (CARISSIMI, 2009). Dessa forma, a nuvem computacional pode prover, de forma simultânea, serviços baseados em diferentes sistemas operacionais, ou ainda, ambientes de diferentes configurações com um menor número de computadores, reduzindo, dessa forma, os custos, o espaço físico necessário, consumo de energia, etc.

Por fim, como as máquinas virtuais são executadas como se fossem processos em uma máquina hospedeira, caso haja alguma instabilidade com uma das máquinas virtuais hóspedes na nuvem computacional que esteja provocando instabilidades no

nodo e, por consequência, nas outras máquinas virtuais desse, a máquina instável pode ser desalocada, de forma a resolver o problema.

Como desvantagem do uso de virtualização em uma nuvem computacional consiste no fato de que, apesar de as máquinas virtuais trabalharem de maneira independente, elas compartilham os recursos de um nodo. Desta forma, qualquer instabilidade que ocorra nesse nodo pode acarretar em problemas nas máquinas virtuais da nuvem computacional. É importante salientar ainda que esse compartilhamento de recursos, com o aumento de máquinas virtuais, provoca um aumento na concorrência pelos recursos do nodo, ocasionando uma perda de desempenho (CARISSIMI, 2007).

## 4.2 Monitores de Máquinas Virtuais

Com a popularização da técnica de virtualização, surgiram diversos *softwares* que implementam essa técnica. Nas próximas seções serão descritos os *softwares* monitores de máquinas virtuais mais conhecidos e utilizados.

### 4.2.1 VMware ESX

O *VMware ESX* é o *software* monitor de máquina virtual desenvolvido pela empresa *VMware*. Esse trabalha de maneira diferente da maioria dos *softwares* monitores de máquina virtual, pois ele é executado a partir do seu próprio sistema operacional, que foi desenvolvido baseado no *Linux*<sup>4</sup>. Dessa forma, ao ligar o computador, o *VMware ESX* é executado logo após a inicialização do sistema operacional hospedeiro, sendo possível, então, gerenciar todas as máquinas virtuais (VMWARE, 2009c).

Por possuir um *kernel* próprio, chamado de *VMkernel*, o *VMware ESX* possui suporte as técnicas de virtualização total e de paravirtualização<sup>5</sup>. Além disso, o *VMware ESX* pode ser utilizado em conjunto outras ferramentas da *VMware*, como, por exemplo, o *VMware vSphere*.

O *VMware ESX* é um software proprietário da *VMware*<sup>6</sup>. Entretanto, existe uma versão gratuita dele, chamada *VMware ESXi*. O *VMware ESXi* possui as mesmas funcionalidades que o *VMware ESX*, exceto um console de gerenciamento gráfico. Dessa forma, todo o gerenciamento das máquinas virtuais deve ser feito através da linha de comando do sistema operacional (VMWARE, 2009d).

---

<sup>4</sup>O *XenServer*, da empresa *Citrix*, tem um funcionamento semelhante ao do *VMware ESX*.

<sup>5</sup>Como já citado anteriormente, a paravirtualização só pode ser utilizada quando o sistema operacional hospedeiro for alterado de forma a prover suporte à essa funcionalidade.

<sup>6</sup>No website do *VMware ESX* não são citados quais são os valores para a aquisição do *software*.



### 4.2.2 Xen

Como citado anteriormente, o *Xen* é o *software* monitor de máquina virtual da empresa *Citrix*. Esse foi desenvolvido sob a licença GPL para a plataforma *Linux*. No momento que o *Xen* é instalado em um sistema operacional hospedeiro, o *kernel* desse é atualizado para uma versão específica que contenha o hipervisor *Xen*. Dessa forma, o *Xen* é inicializado junto com o sistema operacional hospedeiro (XEN, 2009).

O *Xen* foi implementado com a técnica de paravirtualização, portanto, o sistema operacional que será utilizado nas máquinas virtuais hóspedes deverá ser alterado para que exista suporte a essa técnica. Além disso, o sistema operacional hospedeiro também deverá implementado de forma a ter suporte para essa técnica<sup>7</sup>.

Além do *Xen*, a *Citrix* possui outro software que pode ser utilizado para o desenvolvimento de máquinas virtuais chamado *XenServer*. Esse trabalha de forma similar ao *VMware ESX*, ou seja, ele é executado a partir do seu próprio sistema operacional, desenvolvido baseado no *Linux*, utilizando-se do *Xen* para o gerenciamento das máquinas virtuais(CITRIX, 2009b).

O *XenServer* é disponibilizado de forma gratuita. Além disso, a *Citrix* comercializa uma ferramenta chamada *Citrix Essentials* que adiciona funcionalidades ao *XenServer*, como, por exemplo, balanceamento de carga e provisionamento sob demanda.

### 4.2.3 KVM

O KVM (*Kernel-based Virtual Machine*) é um *software* monitor de máquina virtual desenvolvido sob licença GPL para a plataforma *Linux*. Esse é executado no sistema operacional como um processo e utiliza-se de outros *softwares* para o desenvolvimento das máquinas virtuais, como, por exemplo, o *QEMU* que é responsável pela emulação de um processador (KVM, 2009).

O KVM foi implementado com a técnica de virtualização total, portanto, qualquer sistema operacional pode ser utilizado nas máquinas virtuais hóspedes. Entretanto, o KVM somente pode ser utilizado em computadores que possuam processadores que tenham suporte a virtualização. Como os processadores com suporte a virtualização começaram a ser comercializados recentemente, essa restrição acaba, por muitas vezes, impedindo a popularização do KVM.

Futuramente, o KVM também irá contar com suporte a paravirtualização, bem como poderá ser utilizado para a criação de novas máquinas virtuais em sistemas operacionais que estejam sendo executados em máquinas virtuais hóspedes.

---

<sup>7</sup>Em <<http://wiki.xensource.com/xenwiki/DistributionSupport>> pode ser encontrada uma lista das distribuições *Linux* que possuem suporte ao *Xen*.

### 4.3 Considerações Finais

Neste capítulo, foi feita uma breve descrição sobre virtualização, uma vez que essa é principal técnica utilizada na construção de uma infra-estrutura de uma nuvem computacional e será a base para o desenvolvimento do ambiente proposto neste trabalho. Inicialmente, foi realizada uma definição de virtualização. Depois, foram demonstrados os tipos existentes de máquinas virtuais. Além disso, foram apresentadas algumas vantagens e desvantagens da utilização de virtualização em ambientes de computação nas nuvens. Por fim, foram apresentados alguns *softwares* monitores de máquinas virtuais. Para maiores informações sobre virtualização sugere-se (CARISSIMI, 2009) e (SILVA, 2009).

A utilização de virtualização com computação nas nuvens deve-se ao fato de que, a partir dessa, pode-se obter um melhor aproveitamento dos recursos de *hardware*, diminuindo dessa forma a ociosidade. Além disso, a virtualização permite que a infra-estrutura da nuvem ofereça diferentes ambientes utilizando a mesma estrutura física. No próximo capítulo será descrita a plataforma de computação nas nuvens desenvolvido nesse trabalho. Nele serão descritas as características e detalhes da implementação.

## 5 PROJETO DE UM AMBIENTE COMPUTACIONAL BASEADO EM CLOUD COMPUTING

Nos capítulos anteriores, foram descritas algumas características necessárias para o desenvolvimento de uma nuvem computacional. Nesse capítulo é descrita a infraestrutura desenvolvida. Na Seção 5.2 será descrita a implementação desse ambiente. Além disso, na Seção 5.3 serão apresentados os testes realizados nesse ambiente. Por fim, na Seção 5.4 será apresentado o *software AppScale*, que foi desenvolvido pelo laboratório *RACELab*, da Universidade de Santa Barbara e possui o objetivo de aproveitar o potencial disponível em uma nuvem computacional.

### 5.1 Arquitetura da Nuvem Computacional

A nuvem computacional que foi desenvolvida nesse trabalho possui como objetivo fornecer poder de processamento aos seus usuários, semelhante ao serviço EC2 provido pela AWS. Entretanto, como a nuvem computacional que foi desenvolvida tem como principal objetivo a avaliação, tanto do *software* gerenciador quanto da própria nuvem computacional, alguns pontos importantes foram desconsiderados, como, por exemplo, preocupação com aspectos de segurança e com mecanismo de tolerância a falhas.

Como foi citado nos Capítulo 3, são necessários alguns computadores interconectados para serem utilizados como nodos para proverem recursos para a nuvem computacional. Para tal, serão utilizados três computadores disponíveis nos laboratórios de informática da Universidade de Caxias do Sul (UCS), com as seguintes configurações: *Pentium* IV 2.8 Ghz, 1024 Mb de memória RAM e 40 Gb de disco rígido. Esses possuem, ainda, uma placa de rede 10/100/1000, sendo que, o controlador possui ainda uma placa de rede *onboard* 10/100 que é utilizada para o acesso externo a nuvem.

Um desses computadores foi utilizado para as funções de controlador da nuvem e de controlador de *cluster*. Como a infra-estrutura da nuvem computacional

é simples, é possível unir essas duas funcionalidades em um único nó, sem a preocupação de perda de performance. Os outros dois computadores foram utilizados como nós da nuvem computacional, sendo responsáveis por hospedar as máquinas virtuais e prover os recursos computacionais necessários para o funcionamento da nuvem.

Na infra-estrutura desenvolvida, somente o controlador pode ser acessado por computadores que não pertençam a nuvem computacional. Dessa forma, a rede da nuvem é organizada de forma a ter-se uma rede privada, conectando os nós ao controlador. Portanto, qualquer acesso dos nós a um ponto externo da nuvem deve ser realizado através do controlador. A infra-estrutura desenvolvida está representada na Figura 5.1.

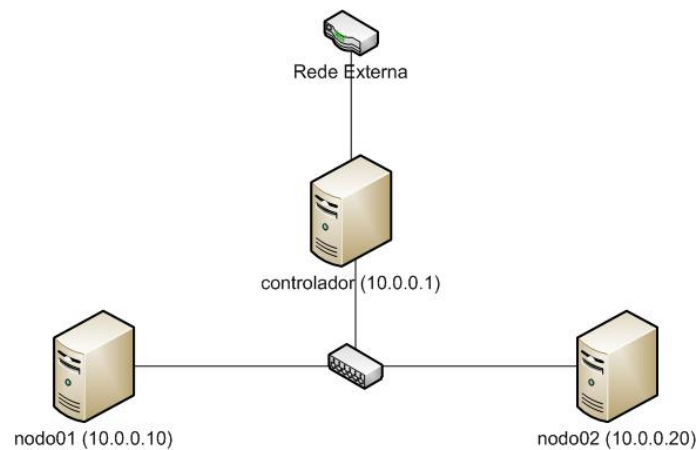


Figura 5.1: Infra-estrutura da Nuvem Computacional Desenvolvida

Para gerenciar a infra-estrutura da nuvem computacional foi utilizado o *software* gerenciador *Eucalyptus*, pois esse apresentou, dentre os *softwares* de código livre estudados, algumas facilidades de utilização e configuração, além do fato de que por ser amplamente utilizado, apresenta uma melhor documentação.

Como já citado anteriormente, o *Eucalyptus* possui duas versões, uma de código aberto e outra proprietária. Para o gerenciamento da infra-estrutura optou-se pela utilização da versão de código aberto, que permite a utilização dos hipervisores *Xen* e *KVM*. Devido ao fato do *KVM* necessitar de processadores com suporte a virtualização e os computadores descritos anteriormente não possuírem tal característica, optou-se pela utilização do *Xen*.

O *Eucalyptus* somente é disponibilizado para o sistema operacional *Linux* nas distribuições *CentOS*, *Ubuntu*, *openSUSE* e *Debian*. Uma vez que optou-se pela utilização do hipervisor *Xen*, torna-se necessária a utilização de uma distribuição que possua suporte nativo ao *Xen*, pois como já foi citado no Capítulo 4, o sistema operacional deve ser alterado de forma a possuir suporte a esse. Assim, optou-se nesse pela utilização da distribuição *CentOS*, que possui suporte nativo ao *Xen*.

Para a utilização das demais distribuições torna-se necessário aplicar um *patch* ao *kernel* do *Linux* e recompilá-lo.

## 5.2 Instalação da Nuvem Computacional

Nesta seção é feita uma breve descrição do processo de instalação da nuvem computacional. Esse processo é descrito com mais detalhes no Anexo *Instalação da Nuvem Computacional*. Nesse são listados os pacotes instalados, bem como os passos que devem ser realizados para a instalação dos mesmos.

Para a criação do ambiente da nuvem, os computadores utilizados foram nomeados conforme a função que vão exercer na nuvem computacional: *controlador*, para o computador que será o controlador da nuvem e do cluster; e *nodo01* e *nodo02* para os computadores que serão utilizados como nodos da nuvem.

A instalação do sistema operacional foi feita da mesma forma nos três computadores. Uma vez que o *Eucalyptus* encontra-se integrado na versão 9.10 do *Ubuntu*, primeiramente realizou-se testes utilizando-se essa versão. Após a instalação do *Ubuntu*, foi necessária a aplicação de um *patch* no *kernel*, pois esse deveria permitir a utilização do hipervisor *Xen*. Porém, devido ao fato de terem ocorrido problemas e instabilidades no *Ubuntu* após a aplicação desse *patch*, optou-se, então, pela utilização da distribuição *CentOS*, na versão 5.4.

No controlador, foram necessários instalar alguns pacotes à que o *Eucalyptus* possui dependência e que não estavam disponíveis depois na instalação padrão do *CentOS*. Posteriormente, foram instalados os pacotes necessários para o funcionamento do CLC e do CC. Instalou-se ainda no controlador uma ferramenta auxiliar ao *Eucalyptus*, chamada *Euca2ools* (EUCA2TOOLS, 2009). Essa ferramenta, que pode ser utilizada também com a AWS, é utilizada para acesso e gerenciamento da nuvem e configuração do *Eucalyptus*. Essa ferramenta pode ser usada, por exemplo, para registrar os nodos e os *clusters*. É através dessa ferramenta, também, que serão gerenciadas as instâncias de máquinas virtuais da nuvem computacional.

Nos nodos, inicialmente, foi instalado o *Xen*. Nessa instalação, o atualizador de pacotes do *CentOS* instalou não só o *Xen*, mas também alguns outros utilitários necessários para que o esse possa ser executado, bem como a versão específica do *Kernel* com suporte ao mesmo. Após a instalação do *Xen*, foram instalados os pacotes necessários para a configuração do *Eucalyptus*, bem como os pacotes com as funcionalidades de controlador do nodo.

Nesse ponto da instalação da nuvem computacional, criou-se uma rede privada, entre os nodos e o controlador. A rede foi organizada de forma que os nodos não possuam acesso à rede externa a nuvem, ou seja, esses possuem acesso somente aos outros nodos. Como já foi citado anteriormente, o controlador possui duas placas de

rede. Uma dessas foi utilizada para a comunicação do controlador à rede da externa, enquanto que, a outra, foi utilizada para conectar o controlador à rede privada da nuvem. Assim, a partir da rede externa, não é possível acessar os nodos da nuvem, somente o controlador.

Após a configuração da rede, foi necessário a a instalação, no controlador, de um servidor NTP (*Network Time Protocol*), bem como a configuração nos nodos para que esses atualizem o seu relógio conforme o horário do controlador. O NTP é necessário pois o *Eucalyptus* exige que os computadores da nuvem possuam os seus relógios sincronizados. Também foi instalado e configurado, no controlador, um servidor DHCP, que será utilizado pelo *Eucalyptus* para atribuir endereços IPs às instâncias das máquinas virtuais que serão criadas por ele.

Após a instalação dos pacotes necessários e a configuração da rede, foram feitos alguns testes de comunicação entre os nodos e o controlador, e do acesso aos componentes do *Eucalyptus* instalados, tanto nos nodos quanto nos controladores. O *Eucalyptus* disponibiliza um *website* através do qual o administrador da nuvem computacional pode consultar as imagens da nuvem computacional, bem como as configurações da mesma. Na Figura 5.2 tem-se uma imagem desse *website*, onde podem ser visualizadas as configurações da nuvem computacional.

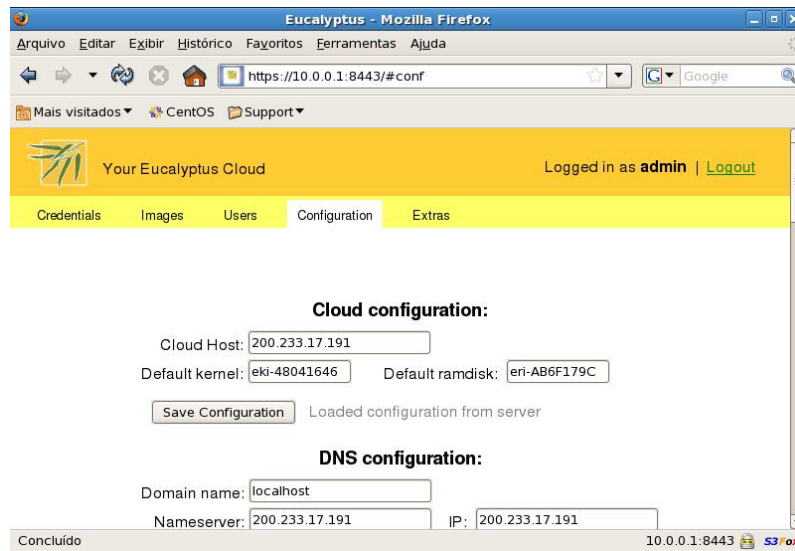


Figura 5.2: *Website* de Configuração do *Eucalyptus*

A partir desse *website*, é necessário executar um *script* responsável pela configuração das variáveis de ambiente do *Eucalyptus*. A partir desse é possível também a criação de chaves que serão utilizadas para comunicação entre os componentes da nuvem. É importante salientar que, em sua instalação padrão, o *Eucalyptus* só permite, nos nodos, a criação de uma máquina virtual por núcleo de processador. Dessa forma, tornou-se necessário alterar no arquivo de configuração

do *Eucalyptus*<sup>1</sup> para que esse permitisse a criação de um número maior de instâncias de máquinas virtuais. Para testes, optou-se pela utilização de 3 máquinas virtuais por computador.

Ao final da instalação, para que a nuvem possa ser utilizada, deve-se, inicialmente, registrar os componentes da nuvem. Isso pode ser realizado através da utilização da ferramenta *Euca2ools*. Essa permite cadastrar o endereço do *Walrus*, que é o serviço responsável por armazenar os arquivos do usuário na nuvem computacional, bem como os dados das máquinas virtuais. Além disso, é necessário registrar cada um dos *clusters* na nuvem computacional, atribuindo-lhes um nome e informando qual será o seu *storage controller* (serviço auxiliar ao *Walrus*). Após o registro dos componentes do *cluster* à nuvem computacional, é necessário registrar os nodos que compõem cada um dos *clusters* da nuvem.

Para a utilização da nuvem é necessário disponibilizar aos usuários os ambientes de trabalho. Devido a simplicidade, optou-se por disponibilizar somente um ambiente com o sistema operacional *Ubuntu* 9.04. O processo para disponibilizar os ambientes na nuvem computacional consiste em montar uma imagem de máquina virtual com o sistema operacional e os *softwares* necessários. Estão disponíveis, no próprio site do *Eucalyptus*, algumas imagens de máquina virtual prontas para a instalação<sup>2</sup>. Além da imagem da máquina virtual, é necessário, também, os arquivos de *kernel* e *ramdisk* da imagem que será utilizada. É importante salientar o fato de que, como está sendo utilizado o hipervisor *Xen*, o *kernel* da imagem deve ser modificado de forma que esse possa ser utilizado com o *Xen*.

Após, deve-se empacotar, submeter e registrar cada um desses arquivos na nuvem computacional. É necessário que, no momento do empacotamento dos arquivos de *kernel* e *ramdisk*, seja informado qual desses arquivos correspondem aos arquivos de *kernel* e *ramdisk*. Também é importante que esses arquivos sejam submetidos à nuvem computacional antes do arquivo de imagem, pois ao empacotar o arquivo de imagem, deve ser informado qual será o *kernel* e o *ramdisk* dessa.

Ao final desses passos, com a imagem disponível para utilização, a nuvem está pronta para aceitar requisições de criação de instâncias de máquinas virtuais e disponibilizá-las para os usuários.

---

<sup>1</sup>Todas as configurações do *Eucalyptus* podem ser visualizadas dentro do arquivo *eucalyptus.conf*, disponível, dentro dos computadores que compõem a nuvem computacional, no diretório */etc/eucalyptus/*

<sup>2</sup>As imagens que estão disponíveis no site do *Eucalyptus* podem ser encontradas em [http://open.eucalyptus.com/wiki/EucalyptusUserImageCreatorGuide\\_v1.6](http://open.eucalyptus.com/wiki/EucalyptusUserImageCreatorGuide_v1.6).

### 5.3 Testes da Nuvem Computacional

Para a realização dos testes de funcionamento e performance na nuvem computacional, foram definidos alguns cenários. Esses foram baseados nas vantagens, proporcionadas pelo uso de computação nas nuvens e que foram descritas na Seção 2.1.2. que a utilização de nuvem computacional proporcionam. Os testes realizados objetivaram verificar as seguintes vantagens:

- Facilidade para inclusão de novos recursos;
- Manutenção facilitada;
- Tolerância a falhas;
- Disponibilização de diferentes ambientes;
- Facilidade para atualização de *softwares*.

Nas próximas seções serão descritos os cenários criados para os testes e os resultados obtidos. Além disso, na Seção 5.4 serão apresentados os testes realizados com o *AppScale*, que é uma aplicação desenvolvida especificamente para ser executada em uma infra-estrutura de nuvem computacional. Esses testes tiveram como objetivo avaliar a execução de uma aplicação na nuvem computacional desenvolvida.

#### 5.3.1 Inclusão de Recursos na Nuvem Computacional

A partir deste teste, pretende-se verificar o comportamento da nuvem computacional a partir da inclusão de novos recursos à nuvem computacional. A Figura 5.3 demonstra os passos descritos neste teste.

Primeiramente, a nuvem computacional foi configurada de forma a possuir um único nodo, o computador *nodo01*. Ao final da inicialização do sistema operacional do *nodo01*, esse nodo dispunha de 794 Mb de memória RAM para a alocação de máquinas virtuais. Foram solicitadas, então 2 instâncias de máquinas virtuais para a nuvem computacional, cada uma delas com 128 Mb de memória RAM. A nuvem computacional, então, instanciou as duas máquinas virtuais solicitadas no *nodo01*. Ao final desse processo, o *nodo01* possuía 538 Mb de memória RAM livre para a alocação de novas máquinas virtuais.

Após, foi incluído na nuvem computacional o segundo nodo, o *nodo02*. Esse, ao final de sua inicialização, também possuía 794 Mb de memória RAM para a alocação de máquinas virtuais. Na Figura 5.4 está representado o arquivo de *log* do controlador da nuvem, onde, na linha 6 é mostrado o momento em que a nuvem



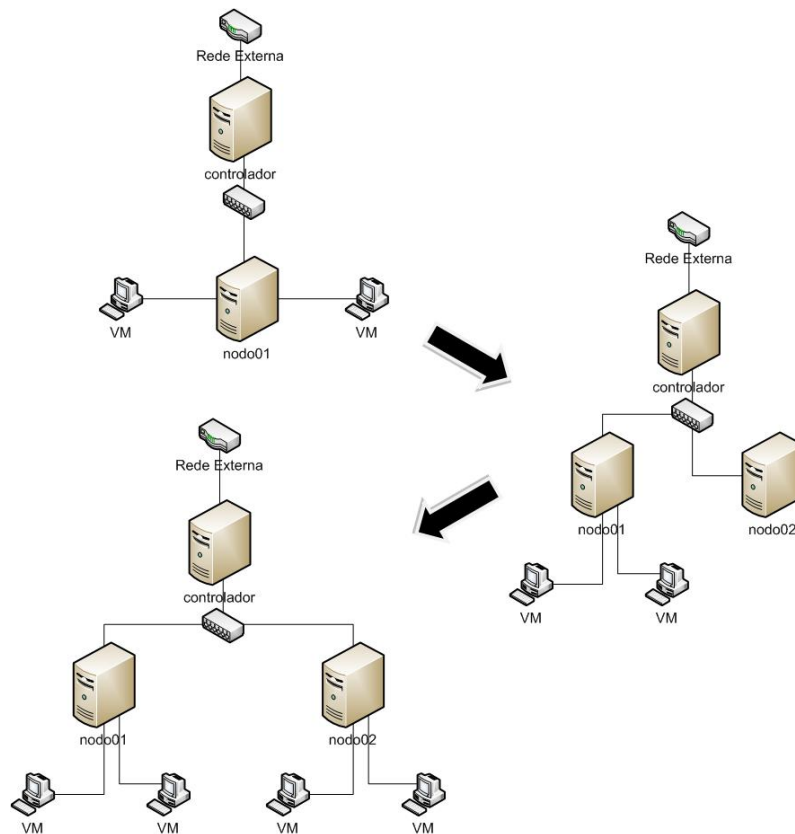


Figura 5.3: Cenário de Teste de Inclusão de Recursos na Nuvem Computacional

```

1. node=10.0.0.10 mem=794/538 disk=4385/3871 cores=3/1
2. refresh_resources(): done
3. timeout(10/10)
4. DescribeInstances(): done
5. DescribeResources(): done
6. config file has been modified, refreshing node list
7. refresh_resources(): called
8. calling http://10.0.0.10:8775/axis2/services/EucalyptusNC
9. time left for next op: 60
10. node=10.0.0.10 mem=794/538 disk=4385/3871 cores=3/1
11. discovered MAC '00:04:75:9A:97:58' for host 10.0.0.10(10.0.0.10)
12. calling http://10.0.0.20:8775/axis2/services/EucalyptusNC
13. time left for next op: 60
14. node=10.0.0.20 mem=794/794 disk=2044/2044 cores=3/3
15. discovered MAC '00:00:F0:7E:01:5B' for host 10.0.0.20(10.0.0.20)

```

Figura 5.4: Log do Controlador da Nuvem No Teste de Inclusão de Recursos

```

1. running instance i-41EC0797 with emiId emi-5B6112C2...
2. assigning MAC/IP: d0:0d:41:EC:07:97/0.0.0.0/0.0.0.0/-1
3. scheduler using ROUNDROBIN policy to find next resource
4. scheduler state starting at resource 1
5. scheduler state finishing at resource 0
6. scheduler decided run instance 'i-41EC0797' on
   resource 'http://10.0.0.20:8775/axis2/services/EucalyptusNC'
   .
   .
   .
7. running instance i-31540708 with emiId emi-5B6112C2...
8. assigning MAC/IP: d0:0d:31:54:07:08/0.0.0.0/0.0.0.0/-1
9. scheduler using ROUNDROBIN policy to find next resource
10. scheduler state starting at resource 1
11. scheduler state finishing at resource 0
12. scheduler decided run instance 'i-31540708' on
    resource 'http://10.0.0.20:8775/axis2/services/EucalyptusNC'

```

Figura 5.5: *Log* do Controlador Para Solicitação de Novas Instâncias no Teste de Inclusão de Recursos

computacional detectou o novo nodo e a partir da linha 10 é mostrado o momento em que a nuvem computacional recebeu os recursos disponíveis nos seus nodos.

Foram solicitadas, novamente, 2 instâncias de máquinas virtuais para a nuvem computacional, cada uma delas com 128 Mb de memória RAM. A nuvem computacional, então, após verificar a disponibilidade dos recursos criou as duas instâncias de máquinas virtuais no *nodo02*, uma vez que esse possuía mais recursos livres. Na Figura 5.5 está representado o arquivo de *log* do controlador da nuvem, onde, a nuvem verifica quais são os recursos disponíveis em cada um dos nodos (linhas 3 e 9, respectivamente para cada uma das solicitações de instâncias de máquinas virtuais). Após, as máquinas virtuais são instanciadas no *nodo02* (linhas 6 e 12).

Dessa forma, pode-se constatar que o *Eucalyptus* provê uma facilidade para adicionar novos nodos à nuvem computacional, aumentando, dessa forma, os recursos da nuvem computacional. Além disso, pode-se constatar que, com a adição de novos recursos, a nuvem computacional, no momento da alocação de novas instâncias, foi capaz de realizar um balanceamento de carga, ou seja, não sobrecarregando um único nodo, mas sim, distribuindo as requisições através dos nodos da nuvem. Entretanto, o *Eucalyptus* não foi capaz de balancear os recursos que já estavam alocados no momento em que foi incluído o novo nodo à nuvem computacional. Esse balanceamento só foi realizado para a alocação de novas instâncias de máquinas virtuais.

### 5.3.2 Manutenção da Nuvem Computacional

A partir deste teste, pretende-se verificar uma maior facilidade de manutenção, Caso seja necessário, um nodo da nuvem computacional pode ser desabilitado para a realização de manutenção, porém, a desabilitação desse nodo, não deve acarretar em uma paralisação nos serviços oferecidos pela nuvem computacional. A Figura 5.6 demonstra os passos deste teste. Neste teste, o *nodo02* foi desativado de forma explícita, ou seja, esse foi desativado utilizando as ferramentas de gerenciamento disponíveis no *Eucalyptus*

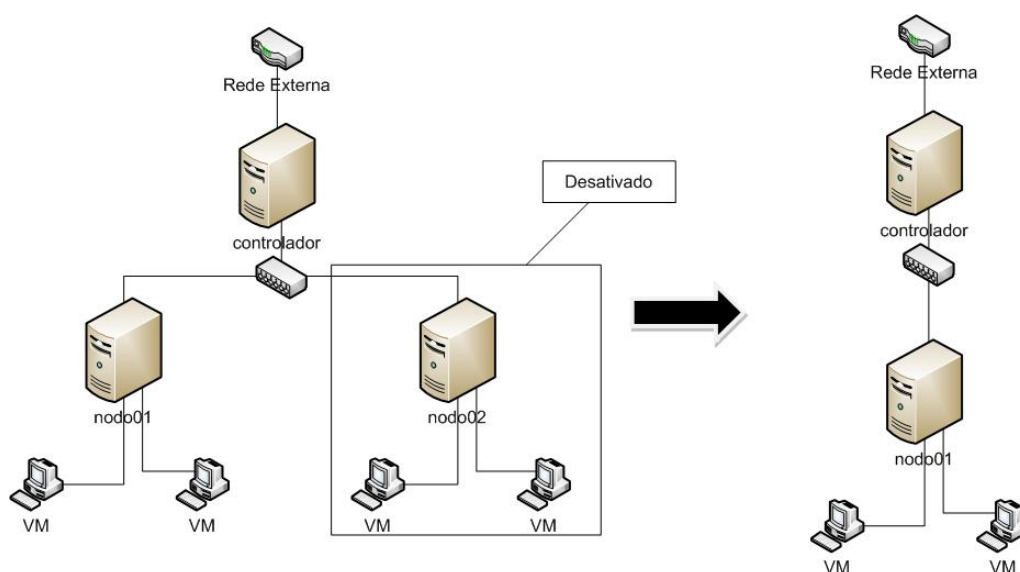


Figura 5.6: Cenário de Teste de Manutenção na Nuvem Computacional

Primeiramente, a nuvem computacional foi configurada de forma que essa contasse com dois nodos: *nodo01* e *nodo02*. Na nuvem computacional, estavam ativas quatro instâncias de máquinas virtuais, sendo que cada um dos nodos era responsável por hospedar duas dessas máquinas virtuais. Nesse momento, na máquina controladora da nuvem, foi desabilitado o *nodo02*, a fim de simular uma parada para a manutenção desse. Na Figura 5.7 está representado o arquivo de *log* do controlador da nuvem. Analisando-se esse *log*, pode-se visualizar que, o controlador, ao detectar essa alteração na configuração dos nodos (conforme demonstrado na linha 9), passou a não verificar mais os recursos do nodo desabilitado.

O nodo, então, ao verificar que não havia mais comunicação com o controlador da nuvem (depois de um determinado intervalo de tempo) eliminou todas as máquinas virtuais que estavam sendo executadas, conforme é demonstrado no arquivo de *log* do nodo, que está representado na Figura 5.8. Porém, essas instâncias foram simplesmente eliminadas sem que a nuvem tentasse transferir essas instâncias para o outro nodo ativo.

Da mesma forma que o *Eucalyptus* provê uma facilidade para adicionar novos

```
1. calling http://10.0.0.10:8775/axis2/services/EucalyptusNC
2. time left for next op: 60
3. node=10.0.0.10 mem=794/666 disk=4385/3871 cores=3/2
4. calling http://10.0.0.20:8775/axis2/services/EucalyptusNC
5. time left for next op: 60
6. node=10.0.0.20 mem=794/538 disk=2044/1016 cores=3/1
7. refresh_resources(): done
8. DescribeResources(): done
9. config file has been modified, refreshing node list
10. DescribeNetworks(): called
11. DescribeResources(): called 5
12. refresh_resources(): called
13. calling http://10.0.0.10:8775/axis2/services/EucalyptusNC
14. time left for next op: 60
15. DescribeNetworks(): done
```

Figura 5.7: *Log* do Controlador da Nuvem No Cenário de Teste Para a Manutenção da Nuvem Computacional

```
1. doDescribeInstances() invoked
2. doDescribeResource() invoked
3. doTerminateInstance() invoked (id=i-31540708)
4. destroyed domain for instance i-31540708
5. doTerminateInstance() invoked (id=i-41EC0797)
6. destroyed domain for instance i-41EC0797
```

Figura 5.8: *Log* do Nodo Desabilitado No Cenário de Teste Para a Manutenção da Nuvem Computacional

nodos à nuvem computacional, também é provida uma facilidade para desabilitar esses nodos da nuvem computacional. Entretanto, pode-se constatar que o *Eucalyptus* não realiza nenhuma ação com o objetivo de transferir as instâncias que estavam ativas em um nodo desabilitado para outro nodo ativo, esse simplesmente finaliza essas instâncias. Dessa forma, se algum usuário estivesse utilizando alguma instância em um nodo desabilitado, seu ambiente seria finalizado e o seu trabalho seria perdido.

### 5.3.3 Tolerância a Falhas

A partir deste teste, pretende-se verificar qual o comportamento do *Eucalyptus* no momento de uma falha em um dos nodos da nuvem computacional. A Figura 5.9 demonstra os passos descritos neste teste. Inicialmente, a nuvem computacional foi configurada de forma que ela contasse dois nodos: *nodo01* e *nodo02*. Na nuvem computacional, estavam ativas quatro instâncias de máquinas virtuais, sendo que cada um dos nodos era responsável por hospedar duas dessas máquinas virtuais. Nesse momento, foi desconectado o cabo de rede utilizado para a conexão do *nodo02* à nuvem computacional, simulando uma situação de queda abrupta de um dos nodos da nuvem.

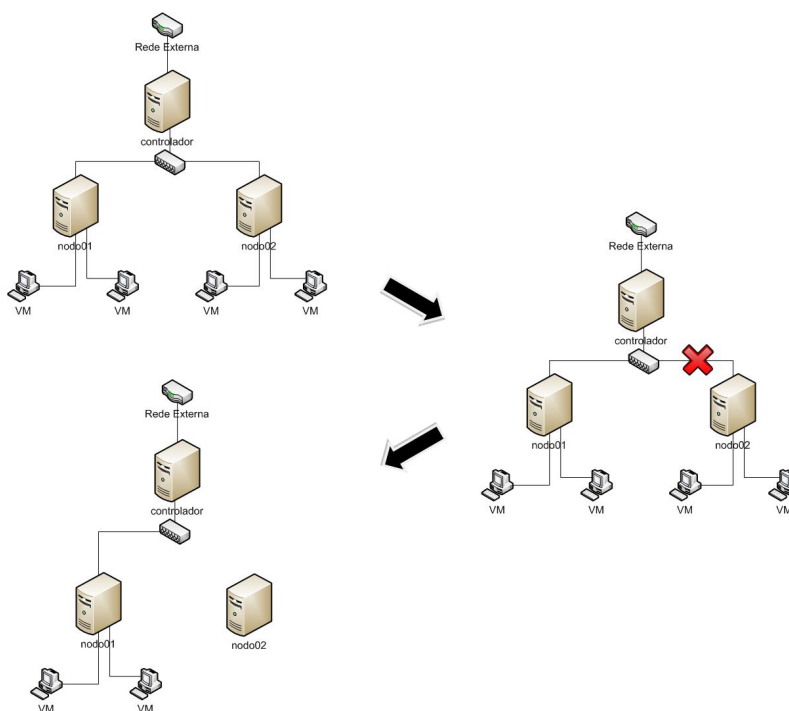


Figura 5.9: Cenário de Teste de Tolerância a Falhas

Após a queda do nodo, o controlador da nuvem continuou tentando comunicar-se com esse nodo, conforme pode ser observado no *log* do controlador que está representado na Figura 5.10. Nas linhas 2 e 3 do arquivo de *log* é demonstrado que o controlador não conseguiu comunicar-se com o *nodo01* e nas linhas 4, 5 e 6 é

```

1. DescribeInstances(): called
2. ERROR: DescribeResource() could not be invoked
   (check NC host, port, and credentials)
3. bad return from ncDescribeResource(10.0.0.10) (18608/1)
4. calling http://10.0.0.20:8775/axis2/services/EucalyptusNC
5. time left for next op: 57
6. node=10.0.0.20 mem=794/538 disk=2044/1016 cores=3/1

```

Figura 5.10: *Log* do Controlador da Nuvem No Cenário de Teste de Tolerância a Falhas

```

1. libvirt: Domain not found:
   xenUnifiedDomainLookupByName (code=42)
2. warning: hypervisor failed to find domain i-4F7108F9,
   will retry 0 more times
3. libvirt: Domain not found:
   xenUnifiedDomainLookupByName (code=42)
4. warning: hypervisor failed to find domain i-4F7108F9,
   assuming it was shut off

```

Figura 5.11: *Log* do Nodo No Cenário de Teste de Tolerância a Falhas

demonstrado o retorno fornecido pelo *nodo02*. O controlador, como não conseguia obter resposta do nodo, concluiu que as instâncias que estavam executando nesse não pertenciam mais a nuvem computacional e que poderiam ser finalizadas. Porém, as instâncias das máquinas virtuais permaneciam ativas nos nodos e o controlador do nodo, após algumas tentativas de comunicação com o controlador da nuvem, concluiu que essas instâncias poderiam ser eliminadas, conforme é representado na Figura 5.11, onde é demonstrado o *log* do *Eucalyptus* no nodo, onde, na linha 3 o nodo tenta comunicar-se com o controlador da nuvem e na linha 4 ele conclui que a instância pode ser eliminada. Dessa forma, as instâncias foram encerradas sem que a nuvem computacional reexecutasse essas em um outro nodo ativo.

### 5.3.4 Ambientes Padronizados Sob Demanda

A partir deste teste, pretende-se verificar as facilidades de disponibilização de um ambiente padronizado para os usuários da nuvem, bem como a possibilidade de disponibilização de ambientes com capacidade de processamento conforme a necessidade de cada um dos usuários.

O *Eucalyptus* permite a parametrização de cinco diferentes classes de instâncias das máquinas virtuais. É possível parametrizar, para cada uma dessas classes, quantos processadores que ela irá conter, bem como a quantidade de memória RAM (em *megabytes*) e a quantidade de disco rígido (em *gigabytes*). É importante salientar

que essa parametrização não possui nenhum tipo de restrição, ou seja, é possível informar qualquer valor a esses atributos, mesmo que a nuvem não consiga atender essa solicitação. Também é importante salientar que, apesar da possibilidade de instanciação de máquinas virtuais com diferentes recursos, essas podem ser criadas utilizando-se da mesma imagem de máquina virtual, ou seja, oferecendo um ambiente com os mesmos *softwares* disponíveis.

Inicialmente, foi parametrizado, na nuvem computacional, cinco diferentes classes. Essa parametrização pode ser visualizada na Figura 5.12. Neste teste, optou-se pelo monitoramento do atributo memória RAM para verificar a alocação de instâncias com diferentes configurações.

**VM Types:**

Name	CPUs	Memory (MB)	Disk (GB)
m1.small	<input type="text" value="1"/>	<input type="text" value="128"/>	<input type="text" value="2"/>
c1.medium	<input type="text" value="1"/>	<input type="text" value="256"/>	<input type="text" value="5"/>
m1.large	<input type="text" value="1"/>	<input type="text" value="512"/>	<input type="text" value="10"/>
m1.xlarge	<input type="text" value="1"/>	<input type="text" value="1024"/>	<input type="text" value="10"/>
c1.xlarge	<input type="text" value="1"/>	<input type="text" value="1024"/>	<input type="text" value="20"/>

Figura 5.12: Configuração das Classes de Ambientes da Nuvem Computacional

Para esse teste, foram utilizadas as duas primeiras classes de máquinas virtuais disponíveis: a classe *t1.small*, que foi parametrizada com 128 Mb de memória RAM; e a classe *t1.medium*, que foi parametrizada com 256 Mb de memória RAM. Após, foram solicitadas três instâncias de máquina virtual da classe *t1.small*. Essas foram alocadas da seguinte forma: o *nodo01* foi responsável por alocar duas instâncias (i-4DA508E4 e i-2FF10696), enquanto que, o *nodo02* foi responsável por alocar uma única instância (i-54B40A75). Na Figura 5.13 está demonstrado, respectivamente, no *nodo01* e no *nodo02*, quais instâncias de máquinas virtuais estavam ativas, bem como a quantidade de memória que cada uma estava utilizando. Essa informações foram obtidas através do comando *xm list*, que lista as instâncias de máquinas virtuais que estão ativas em um computador e que foram criadas pelo *Xen*.

Em seguida, foi solicitada uma nova instância de máquina virtual da classe *t1.medium*. Essa foi alocada no *nodo02* (i-3FD2071F), uma vez que esse possuía uma maior quantidade de recursos. Na Figura 5.14 está demonstrado, respectivamente, no *nodo01* e no *nodo02*, quais instâncias de máquinas virtuais estavam ativas, bem como a quantidade de memória que cada uma estava utilizando, onde pode-se verificar que o *nodo02* possuía a nova instância com 256 Mb de memória RAM

```

root@nodo1 # xm list
Name                ID Mem(MiB) VCPUs State  Time(s)
Domain-0            0    746      1 r----- 239.0
i-2FF10696          2    127      1 -b---- 10.8
i-4DA508E4          1    127      1 -b---- 10.7

root@nodo2 # xm list
Name                ID Mem(MiB) VCPUs State  Time(s)
Domain-0            0    746      1 r----- 43624.1
i-54B40A75          5    127      1 -b---- 6.4

```

Figura 5.13: Máquinas Virtuais Ativas Na Alocação Inicial

```

root@nodo1 # xm list
Name                ID Mem(MiB) VCPUs State  Time(s)
Domain-0            0    746      1 r----- 287.5
i-2FF10696          2    127      1 -b---- 11.5
i-4DA508E4          1    127      1 -b---- 11.4

root@nodo2 # xm list
Name                ID Mem(MiB) VCPUs State  Time(s)
Domain-0            0    746      1 r----- 43668.1
i-3FD2071F          6    255      1 -b---- 6.3
i-54B40A75          5    127      1 -b---- 7.3

```

Figura 5.14: Máquinas Virtuais Ativas Após a Solicitação de Nova Instância

ativa.

Assim, a partir desses testes, pode-se concluir que o *Eucalyptus* permite a utilização de diferentes classes de máquinas virtuais. Neste teste, foram utilizadas somente duas classes, porém, a utilização de qualquer outra classe poderia ser realizada da mesma forma provendo, assim, ambientes com diferentes configurações conforme a demanda dos usuários. Além disso, pode-se observar que o *Eucalyptus* realizou o escalonamento das instâncias das máquinas virtuais, sempre alocando as novas instâncias no nodo que apresentava mais recursos disponíveis, no caso desses testes, a disponibilidade de memória.

### 5.3.5 Atualização de Um Ambiente na Nuvem Computacional

A partir deste teste, pretende-se verificar a facilidade de atualização de um ambiente disponível aos usuários de um ambiente. Inicialmente, a nuvem computacional possui um arquivo de imagem, representando um ambiente disponibilizado para os usuários. Essa imagem possui a instalação básica do sistema operacional *Ubuntu* 9.04. Houve, então, a necessidade de instalação do *software*



*traceroute*, que é utilizado para a visualização das rotas utilizadas por uma mensagem em uma rede de computadores.

Para a instalação, foi instanciada uma máquina virtual, através do *Xen*, utilizando-se a imagem original, ou seja, sem o aplicativo *traceroute*. Após o processo de inicialização do sistema operacional, foi instalado o *traceroute* nessa imagem e a instância dessa máquina virtual foi finalizada. Ao final, a imagem modificada foi empacotada, submetida e registrada na nuvem computacional, através do *Euca2ools*. Em seguida, foi solicitada uma instância de máquina virtual dessa nova imagem. Quando a nuvem disponibilizou essa instância, ao acessá-la foi possível verificar que o *traceroute* estava instalado e disponível para utilização.

Dessa forma, pode-se constatar que, após a instalação de um novo *software* e a atualização dessa imagem na nuvem computacional, essa passou a oferecer aos usuários o novo *software*, sem a necessidade de uma intervenção do usuário. Além disso, o *Eucalyptus* permite que novas imagens sejam disponibilizadas facilmente, conforme a necessidade dos usuários da nuvem. Entretanto, deve-se salientar o fato de que as imagens nunca são sobrescritas, ou seja, um ambiente nunca é, de fato, atualizado na nuvem computacional, mas sim, sempre são criados novos ambientes. Esse fato pode acarretar em problemas para o gerenciamento dessas imagens, uma vez que, para cada *software* necessário também será necessária a criação de uma nova imagem.

## 5.4 AppScale

Como aplicação utilizada para a avaliação do funcionamento da nuvem computacional optou-se pelo *AppScale*, que é um *software* que foi desenvolvido pelo laboratório *RACELab* da Universidade de Santa Barbara e mantido por algumas empresas, entre elas, o *Google* e a *IBM*. Esse *software* foi desenvolvido tendo como um dos seus objetivos simular o ambiente existente no *Google App Engine* (APPSCALE, 2009). Além disso, esse tem como objetivo disponibilizar na nuvem um serviço que permita a outras aplicações utilizem de forma mais eficiente os recursos disponíveis na nuvem. Esse *software* pode ser utilizado tanto na *Amazon EC2* quanto em uma nuvem computacional desenvolvida com a utilização do *Eucalyptus*.

Os *softwares* desenvolvidos de forma a utilizarem o *AppScale* podem usufruir das vantagens que uma nuvem computacional prove aos seus usuários. Por exemplo, através do *AppScale*, uma aplicação poderá realizar o balanceamento de carga entre as instâncias de máquinas virtuais existentes e desta forma evitar uma sobrecarga em uma única instância. Isso pode ser feito através da utilização de APIs de integração entre o *software* desenvolvido e o *AppScale*. Além disso, o *AppScale* permite a

utilização dos gerenciadores de banco de dados *Hypertable*, *HBase*, *Cassandra*, *Voldemort* e *MySQL*.

Para a disponibilização do *AppScale* na nuvem computacional, é necessária a sua instalação em uma imagem de máquina virtual que será disponibilizada na nuvem computacional. Esse processo é descrito com mais detalhes no Anexo *Instalação do AppScale*. Ao final da instalação, essa imagem modificada deve ser empacotada, submetida e registrada na nuvem computacional, através do *Euca2ools*, criando-se, assim uma nova imagem para os usuários da nuvem.

Entretanto, a utilização do *AppScale* na nuvem computacional difere um pouco da maneira usual de utilização da nuvem, descrita na seção anterior. Ao invés de utilizar o *Euca2ools*, deve ser utilizado um outro *software* específico para o *AppScale*, chamado *AppScale Tools*. É através do *AppScale Tools* que são solicitadas as instâncias de máquinas virtuais para a nuvem computacional (KRINTZ et al., 2009). Na criação dessas instâncias é necessário informar um identificador para as mesmas. Dessa forma, o *AppScale* irá agrupar essas instâncias de forma a executar a aplicação somente nelas. Para disponibilizar uma aplicação é necessário submetê-la a nuvem computacional, informando o identificador do grupo de máquinas que irá executar essa aplicação.

Junto com o *AppScale Tools*, são disponibilizadas algumas aplicações que podem ser utilizadas para testes. Para o teste da nuvem computacional desenvolvida, foi escolhida uma aplicação chamada *Guestbook*, que consiste em um mural de recados que permite, aos usuários, postarem recados que podem ser visualizados pelos outros usuários. Na Figura 5.15 tem-se uma imagem dessa aplicação.

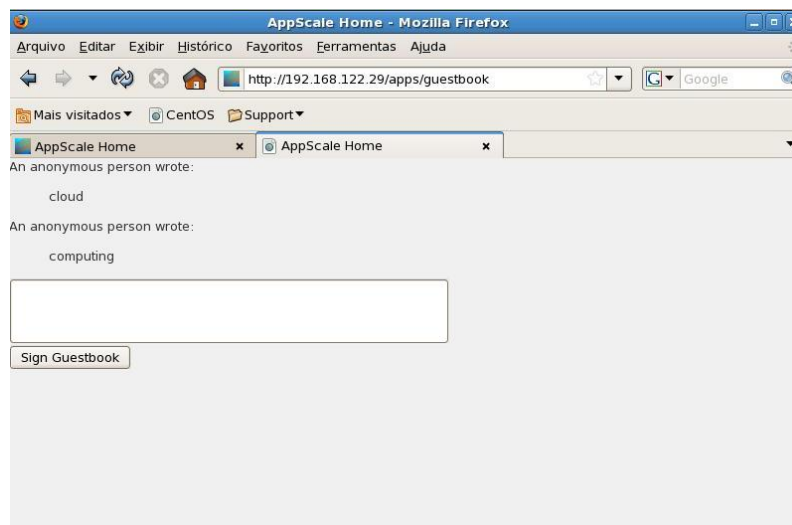


Figura 5.15: Tela da Aplicação *Guestbook*

Para o teste do *AppScale* na nuvem desenvolvida, utilizou-se de duas máquinas virtuais. Como o *AppScale* é um *software* que necessita de bastante recursos,

cada uma das máquinas virtuais foi inicializada com 512 Mb de memória RAM. O *AppScale*, então inicializou essas duas máquinas virtuais, uma com o endereço IP 192.168.122.29 e outra com o endereço IP 192.168.122.220. Esses endereços IPs foram fornecidos a essas instâncias pelo servidor de DHCP da nuvem computacional. O *AppScale* designou, então, a máquina com endereço IP 192.168.122.29 como controladora desse grupo de máquinas. A máquina controladora, além de hospedar a aplicação, possui a função de centralizar todas as requisições para a aplicação. Portanto, todo o acesso à aplicação será feito através dessa controladora.

A consulta dos recursos da nuvem que são utilizadas pelo *AppScale* pode ser realizadas de duas formas. A primeira delas demonstra como está a alocação de todos recursos da nuvem, e é realizado acessando-se através de um navegador a máquina controladora. Nesse *website* é exibida a alocação dos recursos da nuvem para a aplicação. Na Figura 5.16 é demonstrado esse *website* com a alocação dos recursos no teste realizado.

## [AppScale](#)

[The open source framework for running Google App Engine applications.](#)

- [Home](#)
- [Create Account](#)
- [Status](#)
- [Login](#)

### CPU / Memory Usage:

IP Address	CPU % Usage	Mem % Usage
192.168.122.29	7.3	73.44

### Database Information:

- Powered by voldemort
- Data is replicated 1 times

### Applications Hosted in This Cloud:

- [guestbook](#)

Figura 5.16: *Website* do *AppScale* Com o Status da Aplicação

A segunda forma consiste através do comando *appscale-describe-instances*, do *AppScale Tools*, solicitar por todas as instância de um determinado grupo. Dessa forma, são exibidas as informações detalhadas sobre cada um dos recursos das máquinas virtuais, como, por exemplo, qual o percentual de processamento e o percentual de memória que o *AppScale* está utilizando, bem como o estado de cada uma das instâncias. Na Figura 5.17 é demonstrado o resultado da execução desse comando para esse teste realizado. A partir desses testes verificou-se que a nuvem encontra-se apta para a execução de aplicações desenvolvidas com o objetivo específico de execução em uma infra-estrutura de uma nuvem computacional.

```
[root@controlador]# appscale-describe-instances
Status of node at 192.168.122.29:
  Currently using 8.3 Percent CPU and 80.39 Percent Memory
  Database is at 192.168.122.29
  Hosting the following apps: guestbook
  Current State: Done starting up AppScale, now in heartbeat mode
Status of node at 192.168.122.220:
  Currently using 4.5 Percent CPU and 70.44 Percent Memory
  Database is at 192.168.122.29
  Hosting the following apps: guestbook
  Current State: Preparing to run AppEngine apps if needed
```

Figura 5.17: Execução do Comando *appscale-describe-instances*

## 5.5 Considerações Finais

Neste capítulo foi descrito o projeto e a implementação de um ambiente computacional baseado em computação nas nuvens. Inicialmente, foi desenvolvido o projeto de uma nuvem computacional. Depois, foi descrita a implementação dessa nuvem computacional. Além disso, foram feitos alguns testes baseados em cenários que foram definidos a partir das vantagens da utilização de computação nas nuvens.

Por fim, foi apresentado o *AppScale*, que é uma aplicação que utiliza-se do potencial disponível na nuvem computacional e foram feitos alguns testes com essa aplicação. Esses testes tiveram como objetivo avaliar a execução de uma aplicação desenvolvida especificamente para uma nuvem. Para maiores informações sobre o *AppScale* sugere-se (APPSCALE, 2009) e (KRINTZ et al., 2009).

No próximo capítulo serão apresentadas as considerações finais deste trabalho, bem como as sugestões para possíveis trabalhos futuros.

## 6 CONCLUSÃO

Esse trabalho teve como objetivo o projeto e desenvolvimento de um ambiente computacional baseado em computação nas nuvens. Além disso, foi escolhida uma aplicação que permitia avaliar a infra-estrutura da nuvem desenvolvida. O estudo preliminar realizado sobre computação nas nuvens demonstrou que, apesar de ser um paradigma amplamente discutido, não existe um consenso dos autores lidos sobre uma definição de computação. Durante o desenvolvimento desse estudo, houveram dificuldades com relação ao levantamento bibliográfico devido a pouca disponibilidade de referências sobre o funcionamento de uma nuvem computacional e sobre os *softwares* de gerenciamento disponíveis.

Neste trabalho foi estudado, também, a infra-estrutura necessária para a criação de uma nuvem computacional. A infra-estrutura física de uma nuvem computacional é bastante simples, pois, são necessários somente alguns computadores interconectados em uma rede. Entretanto, o *software* utilizado para gerenciar toda essa infra-estrutura demonstra-se bastante complexo, uma vez que esse deve ser capaz de lidar com vários computadores e também deve prever tratamento para situações inesperadas que ocorram no gerenciamento da nuvem computacional.

No estudo dos *softwares* gerenciadores foram encontradas dificuldades para a realização do levantamento bibliográfico, uma vez que existem poucos autores que abordam especificamente as funcionalidades desses *softwares*. Entre os *softwares* gerenciadores estudados, os *softwares* de código aberto foram mais detalhados pois esses apresentavam mais material bibliográfico disponível do que os *softwares* proprietários. Por apresentar maiores facilidades e uma maior quantidade de documentação, optou-se pela utilização do *Eucalyptus* para o desenvolvimento da nuvem computacional.

No desenvolvimento da nuvem computacional e nos testes propostos, o *Eucalyptus* apresentou alguns problemas de utilização e mostrou algumas falhas em alguns pontos críticos que devem ser observados quando for criada uma nuvem computacional com o objetivo de disponibilizá-las aos usuários. Entre esses pontos

pode-se citar: deficiências nos mecanismos de tolerância a falhas e ineficiência na realização do balanceamento de carga das instâncias de máquinas virtuais ativas. Entretanto, o *Eucalyptus* mostrou-se bastante eficiente em outros aspectos importantes da nuvem computacional, como, por exemplo, a distribuição da alocação de novas instâncias entre os nodos da nuvem e a possibilidade de oferecer ambientes sob demandas aos usuários. Por fim, os resultados dos testes realizados indicaram alguns pontos em que a utilização da nuvem computacional desenvolvida trouxe vantagens e indicou alguns outros pontos em que o *software* gerenciador deve ser melhorado para que ele possa ser amplamente utilizado.

Apesar de todas as dificuldades encontradas no desenvolvimento deste trabalho, como pouca bibliografia e o fato de os *softwares* gerenciadores de código aberto terem sido lançados há pouco tempo e possuírem alguns problemas que devem ser solucionados, o paradigma de computação nas nuvens é uma solução que mostra-se promissora para ser adotada nas empresas como solução para a redução dos seus custos. Apesar de, neste trabalho, terem sido descritas de maneira mais detalhada as ferramentas de código aberto, as ferramentas de código proprietário afirmam que solucionar alguns dos problemas encontrados durante o teste com o *Eucalyptus*, como, por exemplo, a questão da tolerância a falhas. A própria versão paga do *Eucalyptus* já provê tratamento para essa questão.

Além disso, durante o desenvolvimento deste trabalho, observou-se o surgimento da oferta de novos serviços baseados em computação nas nuvens, bem como divulgação nas grandes mídias desse paradigma. Por fim, os investimentos realizados por grandes empresas do ramo tecnológico, como, por exemplo, *Google*, *IBM*, *Microsoft*, *Sun*, entre outras, irão funcionar como um grande impulso para as pesquisas e melhorias dos *softwares* baseados nesse paradigma.

## 6.1 Trabalhos Futuros

Durante o desenvolvimento desse trabalho observou-se alguns pontos em que poderia ser feito um estudo mais detalhado. A seguir são listadas algumas sugestões de trabalhos futuros:

- Desenvolver um *software* para ser oferecido como serviço em uma nuvem computacional. Dentro desta ideia, o *software* deve ser capaz de utilizar todo o potencial disponível por uma nuvem computacional, como, por exemplo, utilizar múltiplas instâncias de máquinas virtuais, prover balanceamento de carga e mecanismos de tolerância a falhas;
- Utilizar a nuvem computacional para a realização de processamento distribuído, desta forma utilizando-se das instâncias disponíveis para a

execução de aplicações paralelas;

- Estudar mecanismos de QoS, SLAs e métodos para realizar medições de recursos consumidos, com o objetivo de implementar uma nuvem computacional com propósito comercial;
- Avaliação de outros *softwares* gerenciadores de nuvem computacional disponíveis;
- Desenvolver um estudo mais aprofundado visando a implementação de técnicas que garantam uma maior segurança na nuvem computacional.

## REFERÊNCIAS

Amazon Elastic Compute Cloud. Amazon elastic compute cloud. , 2009. Disponível em: <<http://aws.amazon.com/ec2/>>. Acesso em: julho 2009.

Amazon Web Services. Amazon web services. , 2009. Disponível em: <<http://aws.amazon.com>>. Acesso em: julho 2009.

Amazon.com. Amazon. , 2009. Disponível em: <<http://www.amazon.com/>>. Acesso em: setembro 2009.

APPSCALE. Appscale. , 2009. Disponível em: <<http://code.google.com/p/appscale/>>. Acesso em: novembro 2009.

BUYYA, R.; YEO, C. S.; VENUGOPAL, S. Market-oriented cloud computing: vision, hype, and reality for delivering it services as computing utilities. , 2008.

BUYYA, R.; YEO, C. S.; VENUGOPAL, S.; BROBERG, J.; BRANDIC, I. Cloud Computing and Emerging IT Plataforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. , 2008.

CARISSIMI, A. Virtualização: da Teoria a Soluções. , 2007.

CARISSIMI, A. Virtualização: princípios básicos e aplicações. **ERAD 2009**, Caxias do Sul, 2009.

CITRIX. Citrix cloud center. , 2009. Disponível em: <[http://www.citrix.com.br/products/cloud\\_center.php](http://www.citrix.com.br/products/cloud_center.php)>. Acesso em: outubro 2009.

CITRIX. Citrix xenserver. , 2009. Disponível em: <<http://www.citrix.com.br/products/xenserver.php>>. Acesso em: outubro 2009.

EUCA2TOOLS. Euca2ools. , 2009. Disponível em: <<http://open.eucalyptus.com/wiki/Euca2oolsGuide>>. Acesso em: dezembro 2009.



Eucalyptus. Eucalyptus. , 2009. Disponível em: <<http://open.eucalyptus.com/>>. Acesso em: outubro 2009.

Eucalyptus. Eucalyptus. , 2009. Disponível em: <<http://www.eucalyptus.com>>. Acesso em: outubro 2009.

Eucalyptus. Eucalyptus public cloud. , 2009. Disponível em: <<https://eucalyptus.rightscale.com/>>. Acesso em: agosto 2009.

Eucalyptus. Eucalyptus enterprise edition. , 2009. Disponível em: <<http://www.eucalyptus.com/eee>>. Acesso em: outubro 2009.

GEELAN, J. Twenty-One Experts Define Cloud Computing. **Cloud Computing Journal**, 2009.

Google.com. Google apps. , 2009. Disponível em: <<http://www.google.com/apps/intl/pt-BR/business/index.html>>. Acesso em: setembro 2009.

Google.com. Google app engine. , 2009. Disponível em: <<http://code.google.com/intl/pt-BR/appengine/>>. Acesso em: setembro 2009.

Haizea. Haizea. , 2009. Disponível em: <<http://haizea.cs.uchicago.edu/>>. Acesso em: outubro 2009.

HARWOOD, T. **Inside Citrix MetaFrame XP**. [S.l.: s.n.], 2004.

HAYES, B. Cloud Computing. , Durham,EUA, 2008.

KESSELMAN, C.; FOSTER, I. **The grid**: blueprint for a new computing infrastructure. [S.l.]: Morgan Kaufmann Publishers, 1998.

KRINTZ, C.; BUNCH, C.; MOSTAFA, N.; WOLSKI, R.; SOMAN, S.; PANG, S. AppScale Desing and Implementation. , 2009.

KVM. Kvm. , 2009. Disponível em: <<http://www.linux-kvm.org>>. Acesso em: outubro 2009.

LAUREANO, M. A. P.; MAZIERO, C. A. Virtualização: Conceitos e Aplicações em Segurança. , 2008.

LEAVITT, N. Is Cloud Computing Really Ready for Prime Time? , 2009.

MICROSOFT. Microsoft Virtual PC. , 2009. Disponível em: <<http://www.microsoft.com/windows/virtual-pc/>>. Acesso em: maio 2009.

MILLER, M. **Cloud computing - web-based applications that changes the way you work and collaborate online**. [S.l.: s.n.], 2009.

NURMI, D.; WOLSKI, R.; GRZEGORCZYK, C.; OBERTELLI, G.; SOMAN, S.; YOUSEFF, L.; ZAGORODNOV, D. Eucalyptus: an open-source cloud computing infrastructure. , 2009.

NURMI, D.; WOLSKI, R.; GRZEGORCZYK, C.; OBERTELLI, G.; SOMAN, S.; YOUSEFF, L.; ZAGORODNOV, D. Eucalyptus: a technical report on an elastic utility computing arquitetura linking your programas to useful systems. , 2009.

OPENNEBULA.ORG. Opennebula.org. , 2009. Disponível em: <<http://www.opennebula.org/>>. Acesso em: outubro 2009.

openQRM. openqrm. , 2009. Disponível em: <<http://www.openqrm.com/>>. Acesso em: setembro 2009.

RESERVOIR. Reservoir. , 2009. Disponível em: <<http://www.reservoir-fp7.eu/>>. Acesso em: outubro 2009.

SEO, C. E. Virtualização - Probelmas e Desafios. , Hortolândia, SP, Brasil, 2009.

SILVA, Y. F. da. Uma avaliação sobre a viabilidade do uso de técnicas de virtualização em ambientes de alto desempenho. , 2009.

SOTOMAYOR, B.; MONTERO, R.; LLORENTE, I.; FOSTER, I. An open source solution for virtual infrastructure management in private and hybrid clouds. , 2009.

STARK, D.; STERLING, T. A high-performance computing forecast: partly cloud. , 2009.

SUN. VirtualBox. , 2009. Disponível em: <<http://www.virtualbox.org/>>. Acesso em: agosto 2009.

VAQUERO, L.; RODERO-MERINO, L.; CACERES, J.; LINDNER, M. A Break in the Clouds: Towards a Cloud Definition. , 2009.

VMWARE. VMware vSphere. , 2009. Disponível em: <<http://www.vmware.com/products/vsphere/>>. Acesso em: outubro 2009.

VMWARE. VMware. , 2009. Disponível em: <<http://www.vmware.com>>. Acesso em: abril 2009.

VMWARE. VMware ESX. , 2009. Disponível em: <<http://www.vmware.com/products/esx/>>. Acesso em: novembro 2009.

VMWARE. VMware ESXi. , 2009. Disponível em: <<http://www.vmware.com/products/esxi/>>. Acesso em: novembro 2009.

WALKER, E. The Real Cost of a CPU Hour. , Austin,EUA, 2009.

XEN. Xen. , 2009. Disponível em: <<http://www.xen.org/>>. Acesso em: setembro 2009.

## 7 ANEXOS

### 7.1 Instalação Da Nuvem Computacional

Para a instalação da nuvem computacional, foi necessário, primeiramente, instalar o sistema operacional *CentOS* nos computadores, através de um DVD de instalação.

Em seguida, foi necessário fazer o *download* do arquivo *eucalyptus-1.6.1-centos-i386.tar.gz*, que contém o instalador do *Eucalyptus*, através do endereço `<http://open.eucalyptus.com/downloads/131>`, desempacotar o instalador e acessar o diretório para onde esse foi extraído, através dos seguintes comandos:

```
tar zxvf eucalyptus-1.6.1-*.tar.gz
cd eucalyptus-1.6.1-*
```

Foi necessário, também, a instalação de um serviço de NTP, pois o *Eucalyptus* necessita que os computadores possuam os seus horários sincronizados. Esse foi instalado utilizando-se a ferramenta *yum*, disponível no *CentOS* e que é utilizada para a instalação de novos programas, através do seguinte comando:

```
yum install -y ntp
```

Após, fora, necessárias algumas configurações específicas para os computadores conforme o seu papel na nuvem computacional:

- Instalação Específica dos Nodos:

Inicialmente, foi instalado o hipervisor *Xen*, através dos seguintes comandos:

```
yum install -y xen
sed --in-place \
    's/\#(xend-http-server no)/(xend-http-server yes)/' \
    /etc/xen/xend-config.sxp
sed --in-place \
```

```
's/\#(xend-address localhost)/(xend-address localhost)/' \
/etc/xen/xend-config.sxp
/etc/init.d/xend restart
```

Nesse ponto, deve ser verificado para que, no arquivo */etc/xen/xend-config.sxp*, as seguintes opções sejam configuradas com os seguintes valores:

```
(xend-http-server yes)
(xend-unix-server yes)
(xend-unix-path /var/lib/xend/xend-socket)
(xend-address localhost)
(network-script network-bridge)
(vif-script vif-bridge)
(dom0-min-mem 196)
(dom0-cpus 0)
(vncpasswd '')
```

Após a instalação do hipervisor, deve ser instalado o *Eucalyptus*. Antes, porém, devem ser instalados os pacotes cujo o *Eucalyptus* possui dependência. Esses pacotes são: *aoetools*, *euca-axis2c* e *euca-rampartc*. Esses foram instalados através do seguinte comando:

```
cd eucalyptus-1.6.1-rpm-deps-i386
rpm -Uvh aoetools-21-1.el4.i386.rpm \
        euca-axis2c-1.6.0-1.i386.rpm \
        euca-rampartc-1.3.0-1.i386.rpm
cd ..
```

Após a instalação dos pacotes de dependência do *Eucalyptus*, deve ser instalado o serviço de *Node Controller* do *Eucalyptus* e inicializá-los, através dos seguintes comandos:

```
rpm -Uvh eucalyptus-1.6.1-1.i386.rpm \
        eucalyptus-gl-1.6.1-1.i386.rpm \
        eucalyptus-nc-1.6.1-1.i386.rpm
```

```
/etc/init.d/eucalyptus-nc start
```

- Instalação Específica do Controlador:

No computador controlador, devem ser instalados, primeiramente, alguns *softwares* que são necessários para o funcionamento do serviço de *Cloud Controller*. Os *softwares* necessários são: *java*, *ant*, *ant-nodeps*, *dhcp* e *bridge-utils*. Esses podem ser instalados utilizando-se do utilitário *yum* através do seguinte comando:

```
yum install -y java-1.6.0-openjdk ant ant-nodeps \
    dhcp bridge-utils
```

Após a instalação desses *softwares*, devem ser instalados os pacotes de dependência dos serviços de CLC e CC, através dos seguintes comandos:

```
cd eucalyptus-1.6.1-rpm-deps-i386
rpm -Uvh aoetools-21-1.el4.i386.rpm \
    euca-axis2c-1.6.0-1.i386.rpm \
    euca-rampartc-1.3.0-1.i386.rpm \
    vblade-14-1mdv2008.1.i386.rpm \
    groovy-1.6.5-1.noarch.rpm \
    vtun-3.0.2-1.el5.rf.i386.rpm \
    lzo2-2.02-3.el5.rf.i386.rpm
cd ..
```

Depois de instalados os pacotes de dependência do *Eucalyptus*, deve ser instalado o serviço de CLC e CC, além de inicializá-los. Isso pode ser realizado através dos seguintes comandos:

```
rpm -Uvh eucalyptus-1.6.1-1.i386.rpm \
    eucalyptus-common-java-1.6.1-1.i386.rpm \
    eucalyptus-cloud-1.6.1-1.i386.rpm \
    eucalyptus-walrus-1.6.1-1.i386.rpm \
    eucalyptus-sc-1.6.1-1.i386.rpm \
    eucalyptus-cc-1.6.1-1.i386.rpm \
```

```
eucalyptus-gl-1.6.1-1.i386.rpm
```

```
/etc/init.d/eucalyptus-cloud start
/etc/init.d/eucalyptus-cc start
```

Nesse ponto, todos os *softwares* necessários para o funcionamento da nuvem computacional estão instalados. Entretanto, ainda é necessário configurar alguns parâmetros para a utilização da nuvem computacional. Inicialmente, torna-se necessário registrar os componentes da nuvem computacional ao CLC. A seguir, são demonstrados, respectivamente os comandos para registrar o serviço do *Walrus*, o CC, o *Storage Controller* e os nodos<sup>1</sup>.

```
euca_conf --register-walrus 10.0.0.1
euca_conf --register-cluster cluster 10.0.0.1
euca_conf --register-sc cluster 10.0.0.1
euca_conf --register-nodes "10.0.0.10 10.0.0.20"
```

Deve ser configurado, também, no controlador um servidor de NTP, de forma que os nodos sincronizem os seus horários conforme o horário do controlador. Para isso, deve ser editado o arquivo */etc/ntp.conf*. Abaixo tem-se um exemplo desse arquivo.

```
# Arquivo /etc/ntp.conf
# Servidores de horário públicos
server 200.160.0.8
server 200.189.40.8
server 200.192.232.8
restrict 200.160.0.8 mask 255.255.255.255 nomodify notrap noquery
restrict 200.189.40.8 mask 255.255.255.255 nomodify notrap noquery
restrict 200.192.232.8 mask 255.255.255.255 nomodify notrap noquery
```

Além de configurar o servidor NTP no controlador, os nodos devem ser configurados de forma que esses atualizem os seus horários conforme o controlador. Isso pode ser feito alterando o arquivo */etc/ntp.conf* nos nodos para que ele fique de forma semelhante a como é listado a seguir:

---

<sup>1</sup>Nos comandos, são informados os endereços IP conforme a instalação realizada a nuvem computacional desenvolvida neste trabalho. O endereço 10.0.0.1 corresponde ao endereço interno do computador controlador da nuvem computacional, enquanto que os endereços 10.0.0.10 e 10.0.0.20 correspondem aos endereços IP dos nodos. Ao registrar-se o cluster, também é informado o nome do *cluster* utilizado na nuvem computacional desenvolvida nesse trabalho, cujo nome é *cluster*.

```
# Arquivo /etc/ntp.conf
restrict 0.centos.pool.ntp.org mask 255.255.255.255
    nomodify notrap noquery
restrict 1.centos.pool.ntp.org mask 255.255.255.255
    nomodify notrap noquery
restrict 2.centos.pool.ntp.org mask 255.255.255.255
    nomodify notrap noquery
server 10.0.0.1
restrict 10.0.0.1 mask 255.255.255.255 nomodify notrap noquery
```

Após configuração do serviço de NTP, deve ser configurado, no controlador, o servidor de DHCP. Esse serviço é utilizado pela nuvem computacional para atribuir endereços IPs às máquinas virtuais instanciadas. Essa configuração deve ser feita alterando o arquivo */etc/dhcp.conf* para que ele fique de forma semelhante à como é listado abaixo<sup>2</sup>.

```
# REDE A QUAL SE APLICA
subnet 10.0.0.0 netmask 255.0.0.0 {

# DEFAULT GATEWAY
option routers          10.0.0.1;
option subnet-mask     255.0.0.0;

# DOMINIO E SERVIDORES DNS
option domain-name     "localhost";
option domain-name-servers 200.233.31.210,200.233.31.211;

# HORARIO (BR)
option time-offset     -10800;

# RANGE DE IP (10.0.0.50 ao 10.0.0.70)
range dynamic-bootp 10.0.0.50 10.0.0.70;
default-lease-time 21600;
max-lease-time 43200;
}
```

---

<sup>2</sup>Nos arquivo, são informados os endereços IPs que foram utilizados no desenvolvimento da nuvem computacional desenvolvida neste trabalho. Foram utilizados como endereço da sub-rede o endereço 10.0.0.0, como rota o endereço 10.0.0.1, que corresponde ao endereço IP do controlador da nuvem e, também, foi parametrizado para que os endereços IPs que serão atribuídos estema no intervalo de 10.0.0.50 até 10.0.0.70. Esses valores podem ser alterados conforme a conveniência



Ao final desses passos, a nuvem computacional está configurada e pronta para utilização. Se a instalação foi finalizada com sucesso, o *website* onde podem ser visualizadas as configurações da nuvem pode ser acessada através do seguinte endereço:  $\langle \text{https://}\langle \text{Endereco\_Controlador} \rangle:8443 \rangle$ . A Figura 7.1 demonstra a página inicial desse *website*, onde o usuário informa o seu usuário e senha para que se possa acessar o *Eucalyptus*.

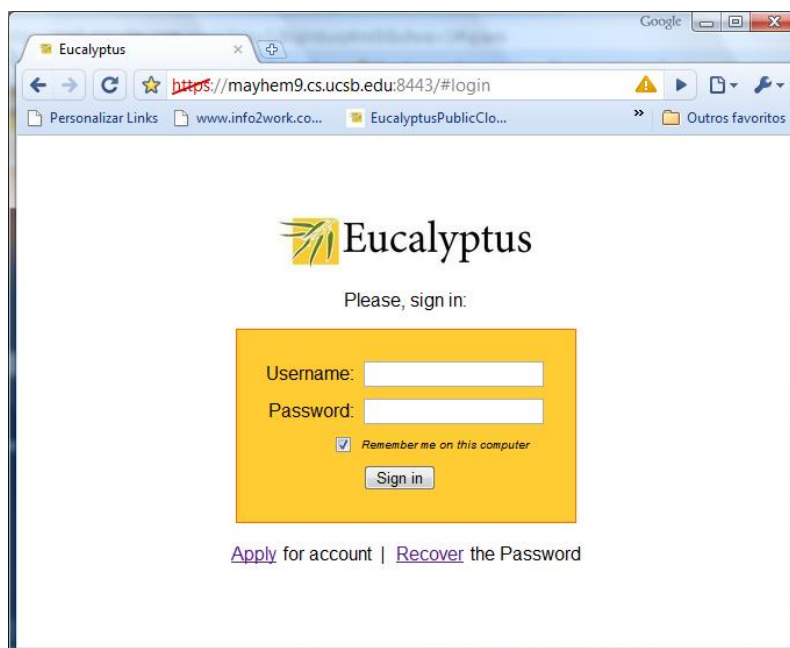


Figura 7.1: Página Inicial do *Eucalyptus*

Entretanto, para que a nuvem computacional possa instanciar as máquinas virtuais, é necessário que elas sejam disponibilizadas na nuvem computacional. Para essa tarefa, existe uma ferramenta, auxiliar ao *Eucalyptus*, que pode ser utilizada para empacotar, submeter e registrar, na nuvem computacional, os seus componentes. Essa ferramenta chama-se *Euca2ools*. Essa está disponível no endereço  $\langle \text{http://open.eucalyptus.com/downloads/137} \rangle$  e a sua instalação deve ser feita através dos seguintes comandos:

```
tar zxvf euca2ools-1.1-*.tar.gz
cd euca2ools-1.1-*
sudo -s
yum install -y swig
rpm -Uvh python25-2.5.1-bashton1.i386.rpm \
    python25-libs-2.5.1-bashton1.i386.rpm \
    euca2ools-1.1-1.i386.rpm
```

Após a instalação do *Euca2ools*, é necessário obter, através do *website* da nuvem computacional, um arquivo de credenciais, que contém chaves que são requeridas

pelo *Euca2ools*. Esse arquivo é necessário para a comunicação do *euca2ools* com a nuvem computacional. O arquivo com as credenciais pode ser obtido conforme indicado na Figura 7.2, acessando-se a aba *Credentials* e clicando-se no botão *Download Certificate*. De posse desse arquivo, ele deve ser descompactado e executado, para que essas credenciais possam ser utilizadas. Isso pode ser feito através dos seguintes comandos:

```
mkdir $HOME/.euca
unzip euca2-admin-x509.zip -d $HOME/.euca
. $HOME/.euca/eucarc
```

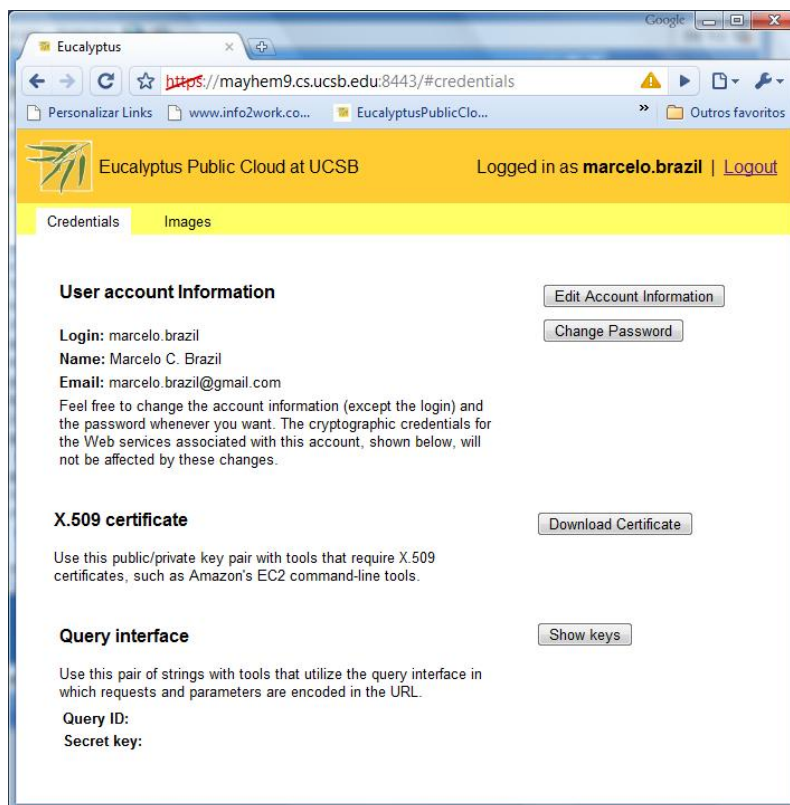


Figura 7.2: Página de Credenciais do *Eucalyptus*

Nesse momento, então, as imagens podem ser submetidas ao *Eucalyptus*. Para esse processo, é necessário, primeiramente, empacotar, submeter e registrar, no *Eucalyptus*, os arquivos com o *kernel* e o *ramdisk* que serão utilizados na criação da instância de máquina virtual. Isso deve ser feito, respectivamente através dos seguintes comandos:

```
euca-bundle-image -i \
euca-ubuntu-9.04-i386/xen-kernel/vmlinuz-2.6.28-11-generic \
--kernel true
```

```

euca-upload-bundle -b ubuntu-kernel-bucket \
  -m /tmp/vmlinuz-2.6.28-11-generic.manifest.xml
euca-register \
  ubuntu-kernel-bucket/vmlinuz-2.6.28-11-generic.manifest.xml

euca-bundle-image -i \
  euca-ubuntu-9.04-i386/xen-kernel/initrd.img-2.6.28-11-generic \
  --ramdisk true
euca-upload-bundle -b ubuntu-ramdisk-bucket \
  -m /tmp/initrd.img-2.6.28-11-generic.manifest.xml
euca-register \
  ubuntu-ramdisk-bucket/initrd.img-2.6.28-11-generic.manifest.xml

```

Após a execução dos comandos *euca-register*, será impresso na tela um número para a identificação do *kernel* (*eki*) e um para a identificação do *ramdisk* (*eri*). Esses números devem ser armazenados, pois, ao submeter a imagem da máquina virtual que será utilizada, esses números deverão ser informados, conforme os comandos a seguir<sup>3</sup>:

```

euca-bundle-image -i \
  euca-ubuntu-9.04-i386/ubuntu.9-04.x86-64.img \
  --kernel $EKI --ramdisk $ERI
euca-upload-bundle -b ubuntu-image-bucket \
  -m /tmp/ubuntu.9-04.x86-64.img.manifest.xml
euca-register \
  ubuntu-image-bucket/ubuntu.9-04.i386.img.manifest.xml

```

Ao final desse processo, a nuvem computacional está pronta para receber requisições de criação de instâncias de máquinas virtuais.

## 7.2 Instalação Ao AppScale

Para a instalação do *AppScale* foi necessário, primeiramente, fazer o *download* dos seus componentes através do endereço `<http://appscale.googlecode.com/files/appscale-1.2.tar.gz>`. Após obter o arquivo, foi necessário instanciar uma máquina virtual com a imagem da máquina virtual onde será instalado o *AppScale*, através do *Xen* e copiar o arquivo com o *AppScale*

---

<sup>3</sup>Onde consta \$EKI e \$ERI, esses valores podem ser substituídos pelos valores impressos na tela. Caso não sejam, essas variáveis de ambiente contém, respectivamente, os valores dos últimos kernel e ramdisk que foram submetidos à nuvem computacional.

para essa máquina virtual. Para a instalação, optou-se pela utilização de uma imagem de máquina virtual contendo o *Ubuntu* 9.04. Após, foi necessário acessar a instância dessa máquina virtual e extrair os arquivos com o *AppScale* do arquivo compactado. Isso pode ser feito através dos seguintes comandos:

```
tar xzvf appscale-1.2.tgz
mv appscale-1.2 appscale
```

Após isso, foi executado o comando que inicializa a instalação do *AppScale* nessa imagem. Isso foi feito através do seguinte comando:

```
./appscale_scratch_install.sh
```

Durante o processo de instalação, além da instalação do *AppScale*, são instalados vários pacotes de dependência do *AppScale* depende, como, por exemplo, *Ruby* e *Phyton*. Além disso, são instalados os bancos de dados que podem ser utilizados com o *AppScale*: *Hypertable*, *HBase*, *Cassandra*, *Voldemort* e *MySQL*. A única intervenção que o usuário faz durante o processo de instalação é quando é solicitado ao usuário se deseja utilizar o interpretador *dash* ou o *bash* no processo de instalação. Os desenvolvedores do *AppScale* recomendam a utilização do *bash*.

Quando foi finalizado o processo de instalação nessa imagem, a máquina virtual foi finalizada e a nova imagem alterada empacotada, submetida e registrada no *Eucalyptus*. Esse registro pode ser realizado através da utilização do *software Euca2ools*.

Ao final desse processo, na máquina de onde serão solicitadas as instâncias de máquinas virtuais, deve ser instalado um *software* chamado *AppScale Tools*, que pode ser obtido através do endereço <<http://appscale.googlecode.com/files/appscale-tools-1.2.tar.gz>>. Após, esse arquivo deve ser desempacotado e o *AppScale Tools* instalado. Isso pode ser feito através dos seguintes comandos:

```
tar xzvf appscale-tools-1.2.tgz
cd appscale-tools-1.2
rm -rf /usr/local/appscale-tools-1.2
mkdir -p /usr/local/appscale-tools-1.2
cp -r * /usr/local/appscale-tools-1.2
```

Além da instalação do *AppScale Tools*, foi necessária a instalação do *Ruby*, uma vez que o *AppScale Tools* foi desenvolvido em *Ruby* e necessita desse para ser executado. Esse pode ser instalado através do utilitário *yum*, através do seguinte comando:

```
yum install ruby curl
```

Também é necessário adicionar à variável de ambiente *PATH* o diretório onde foi instalado o *AppScale Tools*. Isso pode ser feito através do seguinte comando:

```
export PATH=${PATH}:/usr/local/appscale-tools-1.2/bin
```

Ao final desse processo, o *AppScale* está instalado e disponível para utilização na nuvem computacional. A utilização desse é idêntica ao *software* que é utilizado para o gerenciamento das instâncias das máquinas virtuais do *Eucalyptus*.