

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

GUILHERME TELES PAZ

**Desenvolvimento de um Ambiente
para o Armazenamento de Arquivos
baseado em Computação nas Nuvens**

Prof. André Luis Martinotto
Orientador

Caxias do Sul, dezembro de 2009

"If you only do what you know you can do- you never do very much."

— TOM KRAUSE

"If you're not failing every now and again, it's a sign you're not doing anything very innovative."

— WOODY ALLEN

AGRADECIMENTOS

Gostaria de agradecer a toda minha família pela educação que tive. Sou o reflexo das ações e ensinamentos de vocês. Obrigado por tudo que fizeram por mim. Não existem palavras que possam descrever de forma sincera o quanto sou grato. Muito obrigado!

Aos amigos que fiz durante todos esses anos, que compartilharam tantos momentos bons e que perderam horas de sono em prol de uma boa festa. Vocês estarão sempre na minha memória.

Ao meu orientador André Luis Martinotto, por todo o seu apoio e dedicação, um exemplo não apenas de um profissional, mas também de um amigo. Você não só possui o conhecimento, mas também sabe transmiti-lo.

Enfim, agradeço a todas as pessoas com quem tive o prazer de conviver durante a minha vida. De alguma forma, carrego um pouco de cada um de vocês.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	9
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 Objetivos	14
1.2 Estrutura do Trabalho	14
2 COMPUTAÇÃO NAS NUVENS	16
2.1 Definição de Computação nas Nuvens	17
2.2 Arquitetura de uma nuvem computacional	18
2.3 Desafios e Oportunidades na Utilização de Nuvens Computacionais	19
2.4 Vantagens e Desvantagens da Computação nas Nuvens	21
2.5 Serviços Baseados em Computação nas Nuvens	22
2.5.1 Amazon Web Services (AWS)	23
2.5.2 Google	25
2.5.3 Azure	26
2.5.4 Salesforce	28
2.5.5 Serviços de Armazenamento baseados em Nuvem Computacional	29
2.6 Considerações Finais	30
3 INFRA-ESTRUTURA E GERENCIAMENTO DE UMA NUVEM COMPUTACIONAL	31
3.1 Arquitetura de uma Nuvem Computacional	31
3.1.1 Controlador da Nuvem	32
3.1.2 Controlador de Cluster	33

3.1.3	Controlador de Nodo	33
3.2	Gerenciadores de Nuvem Disponíveis	34
3.2.1	Eucalyptus	34
3.2.2	Open Nebula	36
3.2.3	VMware vSphere	39
3.2.4	Citrix Cloud Center (C3)	40
3.2.5	Outras ferramentas gerenciadoras disponíveis	42
3.3	Considerações Finais	43
4	CONSTRUÇÃO DO AMBIENTE	44
4.1	Arquitetura Física da Nuvem Computacional	45
4.2	Softwares utilizados na Nuvem Computacional	46
4.3	Acessando a Nuvem	48
4.4	Testes Realizados	49
4.4.1	Remoção ou inclusão de Recursos	50
4.4.2	Realização de cópias de segurança	52
4.4.3	Capacidade de Armazenamento	53
4.4.4	Padronização de API	53
4.4.5	Confidenciabilidade dos dados	54
4.4.6	Cache de Imagens	55
4.4.7	Transferência de arquivos entre cliente e <i>Walrus</i>	56
4.5	Considerações Finais	58
5	CONCLUSÃO	59
5.1	Trabalhos Futuros	60
	REFERÊNCIAS	61
6	ANEXOS	64
6.1	Instalação do <i>Eucalyptus</i>	64
6.1.1	Instalação Específica nos Nodos	64
6.1.2	Instalação Específica no Controlador da Nuvem	65
6.1.3	Registrar os Componentes da Nuvem ao Controlador	66
6.1.4	Alterar o arquivo de configurações do <i>Eucalyptus</i> nos nodos	67
6.2	Configuração do Serviço de <i>NTP</i>	67
6.3	Configuração do Servidor de <i>DHCP</i>	67
6.4	Configurações Adicionais no <i>CentoOS</i>	68
6.5	Arquivo de Configuração do <i>s3cmd</i>	68
6.6	Utilização do <i>euca2ools</i>	69
6.6.1	Instalação e obtenção das credenciais	69

6.7	<i>Backup</i> dos arquivos da nuvem computacional	70
-----	---	----

LISTA DE ABREVIATURAS E SIGLAS

C3	<i>Citrix Cloud Center</i>
CC	Controlador do <i>Cluster</i>
CDN	<i>Content Delivery Network</i>
CLC	Controlador da Nuvem
CN	Controlador do Nodo
CPU	<i>Central Processing Unit</i>
CRM	<i>Customer Relationship Management</i>
CTP	<i>Community Technology Preview</i>
EB	<i>Exabyte 10¹⁸ bytes</i>
EBS	<i>Elastic Block Storage</i>
EC2	<i>Elastic Compute Cloud</i>
ECP	<i>Elastic Computing Platform</i>
ERP	<i>Enterprise Resource Planning</i>
GAE	<i>Google App Engine</i>
GB	<i>Gigabyte 10⁹ bytes</i>
GPL	<i>General Public License</i>
GMV	<i>Gerenciador de Máquinas Virtual</i>
HT	<i>Hyper Threading</i>
IP	<i>Internet Protocol</i>
QOS	<i>Quality of Service</i>
PB	<i>Petabyte 10¹⁵ bytes</i>
RAM	<i>Random Access Memory</i>

REST	<i>Representational State Transfer</i>
S3	<i>Simple Storage Service</i>
SLA	<i>Service Level Agreement</i>
SOAP	<i>Simple Object Access Protocol</i>
TB	<i>Terabyte 10¹² bytes</i>
TI	Tecnologia da Informação
XML	<i>eXtensible Markup Language</i>

LISTA DE FIGURAS

Figura 2.1: Arquitetura de uma Nuvem Computacional	19
Figura 3.1: Componentes da infra-estrutura de uma nuvem computacional . .	32
Figura 3.2: Arquitetura de uma Nuvem Computacional construída com Eu- calytpus	35
Figura 3.3: Arquitetura de uma Nuvem Computacional construída com o OpenNebula	37
Figura 3.4: Estrutura interna do OpenNebula	38
Figura 4.1: Arquitetura da nuvem computacional construída	45
Figura 4.2: Tela inicial do <i>Eucalyptus</i>	49
Figura 4.3: <i>Logs</i> do controlador do <i>cluster</i> quando o <i>nodo2</i> foi removido da nuvem	51
Figura 4.4: <i>Logs</i> do <i>nodo2</i> quando esse foi removido da nuvem	51
Figura 4.5: <i>Logs</i> do controlador do <i>cluster</i> , onde o <i>nodo2</i> volta a receber instâncias	51
Figura 4.6: <i>Logs</i> do controlador do <i>cluster</i> quando a comunicação foi inter- rompida e logo após reestabelecida	52
Figura 4.7: <i>Logs</i> gerados pela utilização do comando <i>s3cmd</i> para a criação de um <i>bucket</i>	54
Figura 4.8: Acesso a um arquivo público	55
Figura 4.9: Acesso a um arquivo privado	56
Figura 4.10: <i>Logs</i> do nodo da nuvem descrevendo o <i>cache</i> de imagens	56

RESUMO

O “universo digital” encontra-se em constante crescimento, e conseqüentemente, a quantidade de dados gerados por esse também cresce. Além disso, a tecnologia inserida na vida das pessoas através de equipamentos eletrônicos como filmadoras, câmeras digitais e celulares, provocou um grande crescimento na quantidade de dados gerados e compartilhados.

Esse cenário apresenta como vantagem uma grande quantidade de dados disponíveis, porém encontra-se uma dificuldade cada vez maior para o armazenamento, o gerenciamento e a análise desses dados. Esse grande conjunto de dados precisa ser devidamente armazenado e disponibilizado aos usuários, de uma forma simples e eficiente, a fim de facilitar uma posterior análise ou recuperação dos mesmos.

Com o advento das redes de computadores de alta velocidade e as garantias de acesso proporcionadas pela *internet*, uma possível solução encontrada para o armazenamento de arquivos é o armazenamento de arquivos baseado em computação nas nuvens, onde os arquivos podem estar armazenados em qualquer lugar e acessíveis remotamente através da *internet*.

Assim, esse trabalho tem com objetivo desenvolver uma ambiente de armazenamento de arquivos baseado em computação nas nuvens, semelhante ao serviço *S3* da *Amazon*. A partir do desenvolvimento desse ambiente pretende-se avaliar a utilização do paradigma de computação nas nuvens para o armazenamento de arquivos, bem como as ferramentas disponíveis para o gerenciamento de nuvens no que diz respeito ao armazenamento.

Palavras-chave: Computação nas nuvens, *Eucalyptus*, *S3*.

Development of a file storage environment using cloud computing

ABSTRACT

The “digital universe” is in constant growth, thus, the amount of data generated by that also increases. Moreover, the technology embedded in people’s lives through electronic devices such as camcorders, digital cameras and mobile phones, caused a large increase in the amount of data generated and shared.

This scenario presents as advantage a large amount of data available to users, however there is an increasing difficult for the storage, management and analysis of those data. This large set of data needs to be properly stored and available, in a simple and efficient way, in order to facilitate later analysis or recovery.

With the advent of computer networks, high speed and guarantees of access by *internet*, one possible solution for storing files is to store files based on cloud computing, where the files can be stored anywhere and accessible remotely via *internet*.

Thus, this work is intended to develop an environment-based file storage in the clouds, similar to *Amazon’s S3* service. From the development of this environment intended to evaluate the use of the paradigm of cloud computing for storage of files, as well as the tools available for managing the clouds with respect to storage.

Keywords: cloud computing, S3, Eucalyptus.

1 INTRODUÇÃO

Atualmente a humanidade encontra-se na era dos dados digitais. Não é fácil mensurar o volume total de dados que estão armazenados eletronicamente, mas o IDC (*International Data Group*) estima que o tamanho do “universo digital” em 2006 era de 1.8 EB, e espera um crescimento de aproximadamente 10 vezes para o ano de 2011 (GANTZ, 2008).

O aumento dos dados armazenados em meio eletrônico tem várias origens. De acordo com (WHITE, 2009), pode-se citar como exemplo: o *Facebook* que armazena aproximadamente 10 bilhões de fotos, o que corresponde a aproximadamente 1 PB de dados; e a bolsa de valores de Nova Iorque que gera aproximadamente 1 TB de novos dados por dia. Além disso, a tecnologia inserida na vida das pessoas através de equipamentos como filmadoras, câmeras digitais e celulares também provocou um grande crescimento na quantidade de dados gerados e compartilhados. Esse cenário apresenta como vantagem uma grande quantidade de dados disponíveis aos usuários, porém encontra-se uma dificuldade cada vez maior para o armazenamento, o gerenciamento e a análise desses dados (WHITE, 2009).

Esse conjunto de dados precisa ser devidamente armazenado e disponibilizado aos usuários, de uma forma simples e eficiente, a fim de facilitar uma posterior análise ou recuperação dos mesmos. Para esse fim, a alternativa mais simples é a utilização de um servidor de arquivos centralizado, que armazena todos os arquivos de um grupo de usuários, como por exemplo, uma grande empresa. Essa alternativa possui um desafio a ser vencido, que é o contínuo crescimento da quantidade de dados armazenada. Além disso, de acordo com (TANEMBAUM, 1995), o servidor pode sofrer uma sobrecarga, devido ao aumento no número de usuários e de requisições. Por fim, em caso de falhas esse sistema poderá (caso não apresente suporte a falhas) ficar indisponível, ocasionando a interrupção do serviço.

Os cenários atuais e as projeções para o futuro mostram que o modelo centralizado para o armazenamento de arquivos é limitado, e que se faz necessário a busca por uma alternativa para esse modelo de armazenamento. A principal alternativa para esse modelo consiste na utilização de um modelo distribuído. Nesse modelo, os

arquivos ficam armazenados em um conjunto de servidores, que além de armazenar os arquivos também são responsáveis por processar as requisições dos usuários, garantindo assim um melhor desempenho. Esse modelo apresenta, ainda, uma maior tolerância a falhas, pois um problema em um servidor não afetará a disponibilidade dos demais servidores. Por fim, o modelo distribuído apresenta também a flexibilidade de poder-se aumentar ou diminuir a capacidade de armazenamento através da simples inclusão ou remoção de servidores (TANEMBAUM, 1995).

Uma possível solução baseada no modelo distribuído é o armazenamento de arquivos utilizando o conceito de computação nas nuvens (AMAZON, 2009a). A computação nas nuvens pode ser definida como um modelo no qual a computação (processamento, armazenamento e *softwares*) está em algum lugar da rede e é acessada remotamente, via *internet* (VAQUERO, 2009). Esta solução utiliza-se da possibilidade dos usuários recuperarem os dados armazenados a qualquer momento e em qualquer lugar na *internet*, sem precisar, necessariamente, conhecer detalhadamente a infra-estrutura de armazenamento, ou seja, o usuário não precisa conhecer a localização física e a estrutura utilizada para o armazenamento dos dados. Desta forma, o armazenamento de arquivos em uma nuvem computacional torna mais simples a tarefa de acesso aos dados por aplicações que estão inseridas na *web*. A necessidade de compartilhamento dos dados entre as pessoas, a possibilidade de acesso em qualquer lugar, e de um ponto de acesso único para a integração dos dados são características que tornam a utilização da computação nas nuvens atrativa.

Com o advento das redes de computadores de alta velocidade e as garantias de acesso proporcionadas pela *internet* esse modelo de armazenamento começou a ter destaque e passou a ser largamente utilizado. Dentre as empresas que possuem projetos e soluções voltadas para computação nas nuvens, incluindo o armazenamento de dados, cita-se grandes empresas como a *Microsoft* e a *Google*. Entretanto, de todas as empresas que possuem ambientes baseados em computação nas nuvens, a com maior destaque é a *Amazon.com*, que dentro de um pacote de serviços chamado *Amazon Web Services*, disponibiliza o *Amazon Elastic Simple Storage Service (Amazon S3)* (AMAZON, 2009a). Através do *Amazon S3*, a *Amazon.com* disponibiliza aos assinantes o acesso aos servidores para o armazenamento *online* de arquivos. O principal objetivo do *Amazon S3* é prover aos seus clientes um *web service* que permite o acesso aos dados armazenados em seus servidores. Dentre os clientes desse serviço, destaca-se o jornal *The New York Times* que utiliza esse serviço para o armazenamento de imagens das edições mais antigas do jornal.

Dentro desse contexto, este trabalho tem como objetivo realizar um estudo sobre soluções de armazenamento já existentes baseadas em computação nas nuvens. A partir desse estudo será proposto e desenvolvido um ambiente para armazenamento de arquivos baseado em computação nas nuvens. A partir do desenvolvi-

mento dessa infra-estrutura pretende-se avaliar o paradigma de computação nas nuvens em relação a utilização para o armazenamento de arquivos. Além disso, a partir dessa infra-estrutura, pretende-se avaliar as ferramentas de gerenciamento de nuvens computacionais em relação as facilidades disponíveis para o armazenamento de arquivos.

1.1 Objetivos

O presente trabalho tem como principal objetivo o desenvolvimento de um ambiente para o armazenamento de arquivos baseado no conceito de computação nas nuvens. Nesse deverá ser abstraído, do usuário, qualquer indício sobre a localização dos arquivos, que devem ser distribuídos entre os servidores que integram a nuvem de acordo com a capacidade de armazenamento deles.

Existem diversos ambientes que utilizam o conceito de computação nas nuvens para o armazenamento de arquivos, utilizando para isso ferramentas e serviços que realizam o papel de prover o acesso unificado e a distribuição dos arquivos na nuvem. Assim destaca-se que o presente trabalho tem como principal objetivo prover esses recursos utilizando ferramentas disponíveis para a construção de um ambiente de armazenamento baseado em computação nas nuvens e não está focado no desenvolvimento de ferramentas que permitam auxiliar o desenvolvimento da infra-estrutura almejada.

1.2 Estrutura do Trabalho

O presente trabalho é composto da seguinte forma:

- No capítulo 2 é apresentada uma introdução ao paradigma de nuvens computacionais. Nesse são descritas as principais vantagens e desafios apresentados no desenvolvimento de uma infra-estrutura baseada nesse conceito. Além disso, são apresentados também exemplos de aplicações que suportam o armazenamento de arquivos na nuvem.
- No capítulo 3 são descritos os componentes de uma arquitetura de nuvem computacional. Além disso, são apresentadas as ferramentas mais utilizadas para o desenvolvimento desse tipo de arquitetura.
- No capítulo 4 é descrita a infra-estrutura de computação nas nuvens desenvolvida nesse trabalho e os *softwares* utilizados para o desenvolvimento da nuvem computacional. Além disso, nesse capítulo, são descritos os cenários que foram utilizados para os testes da infra-estrutura desenvolvida.

- No capítulo 5 desse trabalho, são apresentadas as conclusões e propostas de trabalhos futuros.

2 COMPUTAÇÃO NAS NUVENS

Com o avanço da sociedade os serviços essenciais são providos para as pessoas de modo que elas tenham um acesso facilitado a esses. Serviços como, por exemplo, o fornecimento de gás, energia elétrica, água e telefonia tornaram-se indispensáveis para a maior parte da população. Esses são utilizados tão frequentemente e devem estar disponíveis a qualquer momento para que os usuários possam usufruir dos mesmos, pagando aos provedores apenas a quantidade utilizada do recurso. Da mesma forma, os recursos computacionais estão tornando-se um utilitário na vida das pessoas. Assim, as pessoas desejam adquirir estes recursos da mesma maneira que consomem outros tipos de serviços (BUYA, 2008). A fim de atender essa necessidade surgiu o conceito de computação nas nuvens.

A computação nas nuvens pode ser definida como uma forma de computação onde as aplicações, serviços e arquivos são hospedados em uma “nuvem” que é formada por milhares de computadores e servidores, agrupados e acessíveis pela *internet*. Desta forma, todos os programas e documentos são acessíveis através de qualquer computador, desde que o mesmo esteja conectado na *internet* (MILLER, 2009). Assim, os documentos e serviços irão migrar do computador do usuário para a nuvem (MILLER, 2009). Este novo paradigma promete serviços confiáveis prestados através de uma nova geração de *datacenters* que são construídos baseados em tecnologias de virtualização e armazenamento (BUYA, 2008).

Nesse capítulo é apresentada uma introdução sobre o paradigma de computação nas nuvens. Na Seção 2.1 será apresentada uma definição de computação nas nuvens e na Seção 2.2 será detalhada a infra-estrutura de uma nuvem computacional em alto-nível. Já na Seção 2.3 serão apresentados os desafios e oportunidades gerados com a utilização dessa nova tecnologia. Além disso, na Seção 2.4, são apresentadas as vantagens e desvantagens de uma nuvem computacional. Por fim, na Seção 2.5 serão apresentados exemplos de serviços baseados em computação nas nuvens, descrevendo as suas principais características e objetivos.

2.1 Definição de Computação nas Nuvens

A computação nas nuvens ainda não apresenta uma definição que seja um consenso na área da informática. Essa ainda é confundida com a computação através da rede (MILLER, 2009), onde as aplicações estão hospedadas em apenas um servidor e são acessadas pela rede de uma organização. Porém, a computação nas nuvens é mais que isso, uma vez que, nessa, os serviços e dados armazenados estão acessíveis em qualquer lugar do mundo através de uma conexão com a *internet*. O termo terceirização da computação também não se aplica a computação nas nuvens. Isto porque, o fato de uma corporação terceirizar um *datacenter* não torna os serviços disponíveis por esse acessíveis em qualquer parte do mundo através da *internet*.

Em seu artigo (BUYA, 2008) procura definir a essência de nuvem computacional definindo essa como um sistema paralelo e distribuído que consiste em um grupo de computadores virtualizados e interconectados. Esses são provisionados de forma dinâmica e são representados por um ou mais recursos, dependendo do SLA (*Service Level Agreement*)¹ estabelecido entre o provedor de serviços e o usuário. A nuvem computacional possui características comuns de *cluster*² e *grids computacionais*³, porém com atributos próprios, como por exemplo um forte suporte a virtualização. Além disso, as nuvens devem prover serviços para os usuários sem referenciar a infra-estrutura na qual esses serviços estão executando.

A fim de melhor conceituar a computação nas nuvens, apresenta-se uma perspectiva do *Google*, que é uma das empresas pioneiras da computação nas nuvens. Nessa perspectiva, esse tipo de computação possui 6 propriedades bem definidas, que são (MILLER, 2009):

- É centrada no usuário: uma vez um usuário esteja autenticado e conectado na nuvem, o que estiver armazenado lá (documentos, imagens, mensagens, aplicativos) torna-se de sua propriedade. Além disso, esses dados podem ser compartilhados com outros usuários;
- É baseada em tarefas: o foco consiste na tarefa a ser executada pelo cliente e como a aplicação executará essa tarefa;
- É poderosa: conectando centenas de milhares de computadores na nuvem obtêm-se um poder computacional que dificilmente poderia de ser alcançado por um único computador;

¹Um acordo onde o nível de prestação do serviço é definido formalmente.

²*Cluster* é um tipo de sistema distribuído e paralelo, que consiste em um grupo de computadores interconectados trabalhando em conjunto como um único recurso computacional (BUYA, 2008).

³*Grid computacional* é um sistema distribuído e paralelo que possibilita o compartilhamento dinâmico de recursos computacionais espalhados geograficamente, dependendo da disponibilidade, capacidade, performance e custo (BUYA, 2008).

- É acessível: os usuários não estão limitados apenas a uma origem para recuperar os dados como acontece com o armazenamento de arquivos centralizado. Isso deve-se, principalmente, ao fato dos dados estarem armazenados em *datacenters* acessíveis através da *internet*;
- É heterogênea: os dados armazenados na nuvem computacional são heterogêneos e necessitam de mineração e análise para serem acessíveis de uma forma eficiente pelas aplicações que fazem uso desses;
- É programável: muitas das tarefas necessárias para a computação nas nuvens devem ser automatizadas. Por exemplo, se um computador que compõe a nuvem não estiver funcionando adequadamente a programação existente nessa deve redistribuir os dados desse computador para um outro computador da nuvem.

2.2 Arquitetura de uma nuvem computacional

A chave para entender a arquitetura é a nuvem, que consiste em uma imensa rede de servidores, ou até mesmo de computadores pessoais interconectados em um *grid* (MILLER, 2009). Esses computadores podem trabalhar em paralelo, e assim combinar todos os seus recursos de forma a obter um maior poder computacional. Esses computadores normalmente pertencem a empresas terceirizadas que “alugam” o poder computacional gerado pela combinação dos seus recursos.

Na Figura 2.1 tem-se uma imagem em alto nível de uma infra-estrutura de nuvem. Como pode ser visto nessa, os usuários conectam-se na nuvem através de seus computadores pessoais ou dispositivos portáteis conectados a *internet*. Para estes usuários, a nuvem é vista como aplicações, dispositivos ou documentos. O *hardware* da nuvem, bem como, o sistema operacional e os demais *softwares* que controlam toda a arquitetura permanecem invisíveis para o usuário, ou seja, fica abstraído do usuário as configurações de recursos e a localização dos dados acessados.

De acordo com o tipo de serviço provido por uma nuvem essa pode ser classificada em (LEAVITT, 2009):

- *Software-as-a-service (SaaS)*: software como um serviço. A nuvem prove aos usuários uma série de *softwares* que podem ser utilizados através da rede. Esses *softwares* podem variar desde simples editores de texto até aplicações mais complexas, como por exemplo, um ERP (*Enterprise Resource Planning*). Como exemplo de uma nuvem computacional que se enquadra nesta categoria pode-se citar o *Force.com* que disponibiliza um software de CRM (*Customer Relationship Management*) que pode ser utilizado pelos usuários como um serviço. A *Force.com* é melhor descrita na Seção 2.5.4;

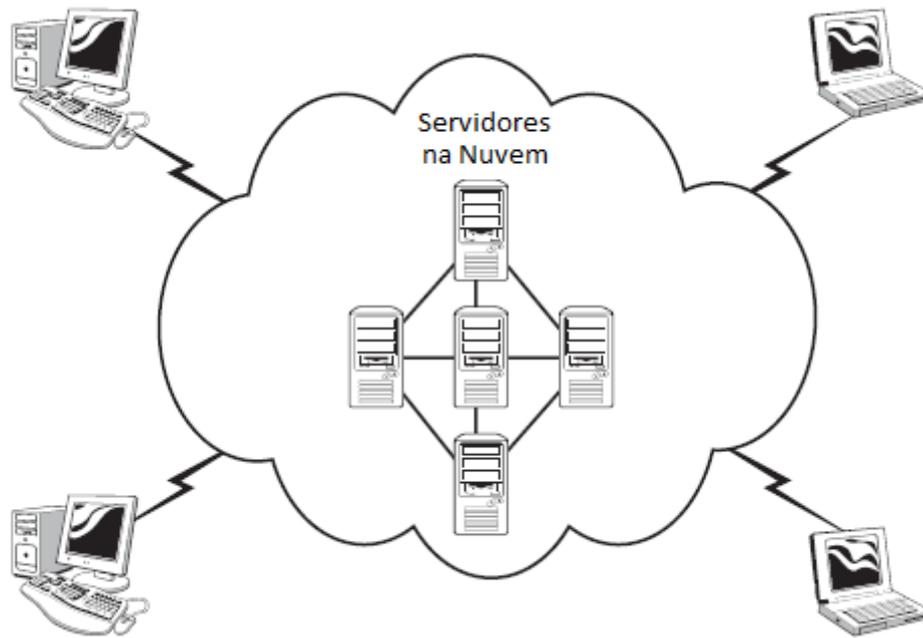


Figura 2.1: Arquitetura de uma Nuvem Computacional

- *Platform-as-a-service (PaaS)*: plataforma como um serviço. A nuvem provê toda a estrutura necessária para o desenvolvimento de aplicações. Dessa forma, os usuários podem utilizar essa estrutura de forma *online* e em colaboração com outros usuários. Como exemplo de uma nuvem computacional que se enquadra nesta categoria pode-se citar o *Google App Engine*, que é melhor descrito na Seção 2.5.2;
- *Infrastructure-as-a-service (IaaS)*: infra-estrutura como um serviço. a nuvem provê uma infra-estrutura via *internet* para a execução de aplicações e o armazenamento de dados. Como exemplo de uma nuvem computacional que se enquadra nesta categoria pode-se citar a *Amazon* e seus serviços *EC2* e *S3*, que são melhor descritos na Seção 2.5.1.

2.3 Desafios e Oportunidades na Utilização de Nuvens Computacionais

A computação na nuvem, como todo novo paradigma, possui alguns obstáculos a serem vencidos. De acordo com (ARMBRUST, 2009), alguns dos obstáculos que dificultam a utilização desse tipo de serviço são:

- Disponibilidade do serviço: quando uma empresa contrata um provedor de forma a disponibilizar a sua aplicação na nuvem, ela deseja que suas aplicações e dados possuam alta disponibilidade. A fim de solucionar este problema os

provedores estão adequando réplicas dos seus *datacenters*, em diferentes regiões geográficas, provendo uma maior disponibilidade dos serviços aos clientes que desejam migrar para a nuvem. Além disso, ao disponibilizar aplicações e dados na nuvem, o contratante pode optar por utilizar mais de um provedor, garantindo uma maior disponibilidade dos seus recursos na nuvem. Outro desafio que pode impedir a disponibilidade de um serviço é um ataque de *DDoS* (*Distributed Denial of Service*⁴), onde *hackers* tentam interromper o serviço disponível na nuvem através de um ataque de negação de serviço. Mas devido a dinamicidade dos serviços na nuvem, os ataques ao invés de serem sofridos pela instância virtual, irão cair no poderio computacional, que é provido de forma dinâmica, dificultando a indisponibilidade dos serviços por meio deste tipo de ataque;

- Padronização de API: uma vez que as APIs (*Application Programming Interface*) de programação não são padronizadas existe a possibilidade de serem diferentes entre os provedores de serviços. Desta forma, os usuários podem manter-se presos a determinados provedores ou até mesmo terem dificultada a sua migração para a nuvem computacional. Uma solução seria o desenvolvimento de uma API padronizada de forma a facilitar o desenvolvimento e o suporte de aplicações baseadas em computação nas nuvens;
- Confidenciabilidade dos dados: baseia-se na dificuldade de garantir a segurança dos dados na nuvem. Essa falta de segurança dos dados, faz com que os usuários evitem esse tipo de arquitetura para o armazenamento de dados sigilosos. Uma forma de minimizar esse problema consiste em criptografar os dados antes de disponibilizá-los na nuvem;
- Limitações na transferências dos dados: essa transferência para a nuvem é um limitante e é altamente dependente dos meios de comunicações envolvidos. A tendência é que esse obstáculo seja diminuído com o aumento na velocidade dos *links* de comunicação;
- Problemas em sistemas distribuídos de larga escala: em grandes sistemas distribuídos, é difícil verificar a existência de falhas em pequenas simulações. Desta forma, frequentemente, os erros no sistema são encontrados apenas quando os sistemas estão em ambiente de produção. A utilização de sistemas virtualizados pode ser uma solução para este desafio, pois através do uso de virtualização é possível capturar informações que auxiliem no tratamento

⁴Um ataque de negação de serviço, é uma tentativa em tornar os recursos de um sistema indisponíveis para seus utilizadores. Desta forma não se trata de uma invasão do sistema, mas sim da sua invalidação por sobrecarga.

das falhas. Essas informações seriam difíceis de serem obtidas em ambientes não virtualizados;

- Licenciamento de *software*: os *softwares* utilizados na nuvem computacional devem ter o seu licenciamento baseado em tempo de uso. Esse tipo de licenciamento permite minimizar os custos de manutenção e aquisição das licenças junto as empresas de *software*;

2.4 Vantagens e Desvantagens da Computação nas Nuvens

A utilização do conceito de nuvem computacional traz diversas vantagens. Entre essas destaca-se a possibilidade de uma fácil alteração dos recursos que essa utiliza, assim como a dinamicidade que os serviços disponíveis na forma de nuvem computacional proporcionam. Segundo (MILLER, 2009), as principais vantagens do uso da computação nas nuvens são:

- Menor custo em recursos computacionais: o usuário não precisa investir em computadores ou servidores com alto poder computacional, uma vez que o poder computacional será disponibilizado pela nuvem;
- Melhor performance: em geral, os computadores pertencentes à nuvem possuem poucos programas em memória, deixando mais recursos disponíveis para a execução das aplicações;
- Menor custo de infra-estrutura: adotando o paradigma de computação nas nuvens, o custo em infra-estrutura de TI pode ser reduzido. Ao invés de comprar novos servidores e investir em maior poder computacional, as empresas podem usar o poder computacional da nuvem de forma a atender picos de processamento;
- Menor custo com *software*: usando uma infra-estrutura de nuvem, as empresas podem diminuir seu investimento na aquisição de licenças de *software*;
- Atualizações instantâneas dos *softwares*: quando a aplicação é baseada na *web*, as atualizações acontecem automaticamente e estarão disponíveis na próxima vez que o usuário utilizar a aplicação na nuvem;
- Maior poder computacional: quando se está utilizando um sistema disponível em uma nuvem computacional tem-se todo o poder de processamento dessa nuvem para a execução das tarefas;
- Capacidade de armazenamento: a nuvem pode conter centenas de PB para o armazenamento de dados;

- Melhor compatibilidade entre sistemas operacionais: para o usuário, não importa o sistema operacional utilizado pelos computadores que hospedam as aplicações, e nem como são mantidos os dados nessa nuvem computacional. Para os usuários o importante são os tipos dos dados e a forma de acesso utilizada;
- Acesso universal aos documentos: esses documentos podem ser acessados em qualquer parte do mundo, bastando para isso possuir uma conexão com a *internet*;
- Possibilidade de manter os dados salvos: em uma nuvem tem-se uma maior segurança em relação a perda de informação, uma vez que os dados na nuvem possuem cópias de segurança e tarefas de replicação. Desta forma, é possível manter na nuvem dados importantes a salvo de problemas de infra-estrutura.

Apesar dessas vantagens, a computação nas nuvens possui algumas desvantagens. De acordo com (MILLER, 2009), as principais são:

- Requer constante conexão com a *internet*: sem uma conexão com *internet* é impossível acessar a nuvem;
- Não apresenta boa performance com uma conexão lenta: acessar a nuvem através de uma conexão de baixa velocidade torna a experiência com a nuvem abaixo da expectativa. Assim, a experiência com a nuvem depende muito da conexão utilizada pelo usuário;
- As configurações podem ser limitadas: frequentemente, as aplicações na *web* possuem um menor número de configurações se comparadas as inúmeras configurações disponíveis em aplicações *desktop*;
- Os dados podem não estar seguros: uma vez que os dados encontram-se em algum lugar na *internet* o usuário não tem uma garantia de segurança dos mesmos;
- A nuvem pode perder os dados: apesar de replicações e cópias de segurança, a nuvem não está a salvo de sofrer problemas e, conseqüentemente, alguns dados podem ser perdidos.

2.5 Serviços Baseados em Computação nas Nuvens

Segundo (STANLEY, 2008), a computação nas nuvens é uma das tendências mais promissoras e, em decorrência disso, tem-se observado um aumento no número de plataformas que utilizam este paradigma. Nesta seção serão apresentadas as mais

conhecidas e representativas soluções de infra-estrutura, plataformas e aplicações baseadas em computação nas nuvens.

2.5.1 Amazon Web Services (AWS)

A *Amazon*, uma das maiores empresas de vendas de produtos na *internet*, foi uma das pioneiras na disponibilização de serviços baseados em computação nas nuvens. A partir do ano de 2006, a *Amazon* passou a fornecer para os usuários uma plataforma de *web services* baseada nesse tipo de infra-estrutura. Através dos serviços da AWS é possível solicitar serviços que disponibilizam, entre outras coisas, capacidade de processamento e armazenamento de forma dinâmica (AMAZON, 2009a).

A *Amazon* conta com vários serviços na nuvem, dentre os quais destaca-se o *Amazon Elastic Compute Cloud (EC2)*, que provê aos usuários a possibilidade de alugarem computadores para a execução de suas próprias aplicações. Para isso o EC2 provê *web services*, através dos quais, os usuários podem criar máquinas virtuais, como por exemplo, instâncias de servidores, onde irão instalar e executar *softwares* de sua escolha. Nesse caso, o usuário pagará somente pelas horas em que as máquinas virtuais permanecerem ativas. Devido a essa característica a *Amazon* atribui o termo *Elastic* para esse serviço.

O *EC2* possibilita aos usuários escolherem instâncias de máquinas virtuais de 2 famílias, que são descritas abaixo (AMAZON, 2009b). Para uma referência completa dos valores dos serviços do *EC2* e as políticas de SLA, sugere-se (AMAZON, 2009b).

1. Instâncias padrão: as instâncias de máquinas virtuais dessa família são utilizadas pela maioria das aplicações submetidas no ambiente *EC2*. Essas instâncias caracterizam-se por aplicações que não necessitam de um grande poder de processamento, mas de recursos de memória RAM e armazenamento. As configurações disponíveis são:
 - (a) *Small* : instância padrão comercializada pela *Amazon* com plataforma de 32 *bits* e processamento equivalente a 1 CPU *EC2*⁵ (1 núcleo virtual com 1 CPU *EC2*). Essa plataforma possui 1.7 GB de memória RAM e 160 GB para armazenamento;
 - (b) *Large*: essas instâncias possuem plataforma de 64 *bits* com 4 CPUs *EC2* (2 núcleos virtuais com 2 CPUs *EC2* cada). Essa plataforma apresenta 7.5 GB de memória RAM, e 850 GB para armazenamento;
 - (c) *Extra Large*: essas instâncias apresentam plataforma de 64 *bits* com 8 CPUs *EC2* (4 núcleos virtuais com 2 CPUs *EC2*). A plataforma conta ainda com 15 GB de memória RAM e 1.65 TB para armazenamento;

⁵Uma CPU *EC2* (ECU) provê capacidade de processamento equivalente a um *Opteron 1.2 Gigahertz* ou processador *Xeon 2007*.

2. Instâncias para alto processamento: as instâncias de máquinas virtuais dessa família proporcionam uma maior capacidade de processamento, porém com uma menor capacidade de memória. Essas são recomendadas para aplicações que requisitam como recurso principal um maior poder de processamento. As configurações disponíveis são:

- (a) *High-CPU Medium Instance*: essas instâncias possuem plataforma de 32 *bits* com 5 CPUs *EC2* (2 núcleos virtuais com 2.5 CPUs *EC2* cada). A plataforma possui ainda 1.7 GB de memória RAM e 350 GB de armazenamento;
- (b) *High-CPU Extra Large Instance*: essas instâncias apresentam plataforma de 64 *bits* com 20 CPUs *EC2* (8 núcleos virtuais com 2.5 CPUs *EC2* cada). A plataforma possui ainda 7 GB de memória RAM e 1.65 TB de armazenamento.

Além do *EC2*, a *Amazon* conta ainda com um serviço chamado de *Amazon Simple Storage Service (S3)*, que é um um serviço de armazenamento através do qual é possível armazenar e recuperar dados estando em qualquer lugar na *internet*. O *S3* permite que arquivos de 1 *byte* até 5 GB sejam escritos, lidos e apagados através de uma *interface* de *web services*. Atualmente os arquivos são armazenados em *datacenters* localizados nos Estados Unidos e Europa, a escolha do cliente. O preço do serviço é calculado mensalmente com base na capacidade ocupada e a taxa de dados transferida nesse período (AMAZON, 2009c). Para uma referência completa dos valores praticados e políticas de *SLA*, sugere-se (AMAZON, 2009c).

Os dados no *S3* são armazenados em *buckets*, que são uma espécie de *container* para os arquivos. Esses provêm um *namespace* único para o gerenciamento dos *buckets*, isto é, o nome que identificará o *bucket* deve ser único na arquitetura do *S3*. Já os arquivos são identificados através do nome do *bucket* onde estão armazenados e um identificador único dentro do *bucket*. O *S3* apresenta um limite de 100 *buckets* que podem ser criados e armazenados por usuário.

Por fim, a *Amazon* disponibiliza APIs que permitem ao usuário desenvolver suas próprias aplicações. Atualmente, linguagens de programação largamente utilizadas, como por exemplo, Java, PHP, Perl, C# e *Visual Basic* já possuem bibliotecas para o desenvolvimento de aplicações a serem executadas na nuvem. A comunicação com os *web services* da *Amazon* é realizada através do padrão *SOAP*. Através dessas APIs, os usuários que utilizam esses serviços podem integrar suas aplicações aos serviços da *Amazon*. Devido a larga utilização dos seus serviços de computação elástica e de armazenamento, as APIs da *Amazon* tem sido utilizadas como padrões em diversas ferramentas de computação nas nuvens.

Como exemplo de empresas que utilizam os serviços da *Amazon* pode-se citar: *Harvard Medical School*, que utiliza o *Amazon EC2* para a execução de simulações com modelos genéticos; e o *The New York Times* que utiliza o *S3* para o armazenamento de imagens das edições mais antigas do jornal.

2.5.2 Google

A *Google* é a empresa líder em aplicações baseadas na *web*, e não é surpresa que seja uma das pioneiras na disponibilização de serviços baseados em computação nas nuvens. O *Google Docs* é hoje o principal conjunto de aplicações baseada na *web*. Esse é formado por 3 componentes que são: o *Google Docs*, que é um processador de textos; *Google Spreadsheets* para confecção de planilhas; e o *Google Presentations* que consiste em um *software* para a criação de apresentações eletrônicas. O *Google Docs* pode ser acessado utilizando um navegador em qualquer computador que possua conexão com a *internet*. Esse serviço é gratuito, porém existem planos corporativos que garantem um *SLA* com uma disponibilidade de 99.99% (GOOGLE, 2009). 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.

Além disso, a *Google* oferece serviços de computação nas nuvens voltados para desenvolvedores. Estes serviços são disponibilizados a partir do *Google App Engine*, (*GAE*) (GOOGLE, 2009), que permite aos usuários construir as suas aplicações de forma que essas sejam executadas na infra-estrutura da *Google*. Esse possibilita ao usuário executar aplicações *web* escritas utilizando as linguagens Python e Java. Além disso, o *GAE* disponibiliza APIs para serviços oferecidos pela própria *Google*, como por exemplo, serviços de *email*, manipulação de imagens e armazenamento de dados. O serviço provê ainda um console de administração para que o usuário possa gerenciar as suas aplicações. A infra-estrutura do *GAE* gerencia toda a distribuição, replicação e balanceamento de carga de dados através dessa API.

Além disso, o *App Engine* oferece um serviço distribuído de armazenamento de dados, o *datastore*, que conta com um mecanismo de consulta e transações. Segundo a *Google*, o *datastore* foi desenvolvido pensando em aplicações que executam na *internet*, ou seja, aplicações que necessitam de uma maior performance em operações de leitura e consulta de dados. Esse serviço possui a limitação de armazenamento, no modo gratuito, de 500 MB e o poder de processamento e a transferência de dados é suficiente para 5 milhões de visualizações mensais. Para usuários com necessidades maiores é oferecido pela empresa um pacote pago onde não há limites de armazenamento e processamento⁶. Para maiores informações sobre o ambiente de

⁶Maiores informações sobre valores praticados pela *Google* e políticas de cobrança sugere-se <http://code.google.com/intl/pt-BR/appengine/docs/billing.html>

desenvolvimento, serviços e funcionalidades oferecidas, sugere-se (GOOGLE, 2009).

2.5.3 Azure

A *Microsoft* é uma gigante da informática dominando diversos mercados através de produtos e *softwares* que vão desde sistemas operacionais até *vídeo games* e *players* de áudio. Como uma empresa pioneira na informática, a *Microsoft* também investiu na computação nas nuvens criando a plataforma *Azure* (CHAPPELL, 2008).

O *Azure* é o projeto onde a *Microsoft* planeja prover um ambiente integrado para o desenvolvimento e gerenciamento de uma nuvem computacional (MICROSOFT, 2009). Esse possui o intuito de auxiliar os desenvolvedores de *software* a criar, disponibilizar e gerenciar aplicações *desktop* e *web* através de seu *datacenter*.

O *Azure* conta com suporte a diversas ferramentas de desenvolvimento e protocolos *Microsoft*, como por exemplo, o *SQL Services*, *Sharepoint* e *softwares* de CRM. Além disso, permite aos desenvolvedores a utilização dos protocolos *SOAP* e *REST* para a comunicação entre seus aplicativos e outros serviços.

O *Azure* é um conjunto de serviços de computação nas nuvens hospedado nos *datacenters* da *Microsoft* que provê um sistema operacional e uma série de serviços que podem ser utilizados individualmente ou em conjunto. Os serviços disponíveis pelo *Azure* são:

- *Windows Azure*: é uma plataforma para a execução de aplicações desenvolvidas para o sistema operacional *Windows* e para o armazenamento dos dados utilizados por essas aplicações. Para isso são utilizados os próprios servidores da *Microsoft*. Esse serviço funciona de forma semelhante ao *EC2* da *Amazon* provendo recursos computacionais sob demanda para os usuários. Atualmente o *Windows Azure* encontra-se em um status que a *Microsoft* chama de CTP (*Community Technology Preview*⁷). Durante este período, a *Microsoft* permite que os usuários utilizem os serviços com algumas limitações, como por exemplo, 2000 horas de processamento, capacidade de armazenamento de 50 GB e uma transferência de 20 GB por dia. Após este período a *Microsoft* cobrará pelo seu serviço. Essa cobrança será realizada de acordo com as horas de processamento utilizadas, capacidade de armazenamento e transferência de dados realizada. Para uma referência completa dos valores desse serviço e políticas de SLA disponíveis, sugere-se (MICROSOFT, 2009);
- *Microsoft .NET Services*: são serviços através dos quais pode-se criar aplicações na nuvem e conectá-las aos sistemas corporativos. O serviço conta com 3 com-

⁷Forma de disponibilização de ferramentas, *software* e tecnologias onde versões de avaliação são distribuídas para pessoa interessadas.

ponentes:

- *Access control*: componentes que provêm meios para o controle de segurança, autenticação das aplicações e serviços disponíveis na nuvem;
- *Service bus*: serviço que visa facilitar a comunicação entre as aplicações através da *internet*. Os serviços registrados no *Service Bus* podem ser descobertos e acessados através de qualquer computador conectado a *internet*;
- *Workflow*: componente que permite a execução de vários aplicativos na nuvem, da forma que estes sejam executados como um *workflow*⁸. No caso do *Microsoft .NET Services*, a cobrança será baseada na quantidade de mensagens utilizadas para a comunicação entre as aplicações da nuvem e as corporativas;
- *Live Services*: oferece um *framework* que possibilita acesso aos dados e a sincronização desses com aplicações *Live*. Essas aplicações consistem em serviços da *Microsoft*, como por exemplo, *email*, serviços de busca e mapas que podem ser acessados através de computadores e uma série de dispositivos. Os desenvolvedores podem criar as suas aplicações de forma que as mesmas possam utilizar esses serviços da *Microsoft* através de um componente, chamado *Live Operating Environment*, que executa na nuvem;
- *Microsoft SQL Services*: serviço de armazenamento oferecido pela plataforma *Azure* que inclui facilidades para a manipulação de dados armazenados de forma relacional. O serviço, como a maioria dos serviços baseados em computação nas nuvens, é pago de acordo com o tamanho do banco de dados utilizado e a quantidade de dados transferida. Esse serviço pode ser acessado através da linguagem C# utilizando a sintaxe LINQ (*Language Integrated Query*), definida pela *Microsoft*, ou através de acesso direto RESTFUL⁹(*Representational State Transfer*), que é uma espécie de *interface web* para a comunicação.

Maiores informações sobre os serviços da plataforma *Azure* da *Microsoft* podem ser encontradas em <http://www.microsoft.com/azure/default.aspx>.

⁸Um *workflow* define um fluxo de um processo ou uma combinação de tarefas que produz um determinado resultado

⁹Transferência do Estado Representacional: ignora os detalhes da implementação de um componente ou a sintaxe de protocolo para utilizar as regras definidas para cada componente e as ligações com outros componentes.

2.5.4 Salesforce

A *Salesforce.com* é conhecida pelos *softwares* de CRM e gerenciamento de força de vendas. Essa é também uma das empresas líderes no desenvolvimento de aplicações baseadas em nuvem computacional, sendo que a sua arquitetura de nuvem é chamada de *Force.com*. Essa plataforma executa de forma provisionada, e permite ao usuário a transferência de aplicações do *datacenter* da empresa para a infraestrutura de computação nas nuvens da *SalesForce* (SALESFORCE, 2009). Este serviço é fornecido através de 3 planos:

- *Force.com Free*: serviço gratuito que possibilita ao usuário disponibilizar apenas uma aplicação, permitindo o acesso de até 100 usuários e um armazenamento limitado a 1 GB;
- *Force.com Enterprise*: permite a disponibilização de até 10 aplicações e possibilidade da disponibilização dessas para mais de 100 usuários. Além disso, esse serviço permite o armazenamento de 200 objetos em um banco de dados. Para a utilização desse serviço a *SalesForce* cobra uma taxa mensal por usuário. Para uma referência completa dos valores praticados e políticas de SLA, sugere-se (SALESFORCE, 2009);
- *Force.com Ilimitado*: serviço que não impõe limites para o número de aplicações, número de usuário ou armazenamento. O único limite nesse serviço consiste na quantidade de objetos armazenados no banco de dados que é de 2000 objetos. Além disso, a *Force.com* disponibiliza um suporte integral e um administrador dedicado. Para a utilização desse serviço a *SalesForce* cobra uma taxa mensal por usuário. Os valores desse serviço, bem como, as políticas de SLA podem ser encontradas em (SALESFORCE, 2009).

Para o armazenamento a plataforma *Force.com* disponibiliza um banco de dados, onde são armazenados objetos contendo inúmeros campos. Esses objetos possuem um *metadata* com uma descrição dos campos e os registros com os dados propriamente ditos. Para a consulta desses dados deve ser utilizada uma linguagem própria do *Force.com*, o *Salesforce.com Object Query Language (SOQL)* ou o *Salesforce.com Object Search Language (SOSL)*. Desta forma, as aplicações desenvolvidas ficam restritas a plataforma *Force.com*.

A *Force.com* conta ainda com uma linguagem de programação própria na sua nuvem, a *Apex*. Esse serviço é semelhante aos fornecidos pela *Google*, *Amazon* e *Microsoft*. A linguagem proporciona os desenvolvedores uma série de características que facilitam o desenvolvimento de aplicações na nuvem da *Salesforce.com*, como por exemplo, o controle dos dados e rotinas de negócio na nuvem. Porém, com a utilização da *Apex* a aplicação fica restrita a executar na infra-estrutura da *Force.com*.

2.5.5 Serviços de Armazenamento baseados em Nuvem Computacional

Uma vez que esse trabalho está voltado para o desenvolvimento de uma infraestrutura para armazenamento na nuvem, nessa Seção serão descritos serviços menos conhecidos, porém importantes devido ao fato de serem dedicados a armazenamento. Nessa seção serão descritos brevemente os serviços oferecidos pela empresa *Rackspace* e a *Nirvanix*.

- *Rackspace*: essa infra-estrutura conta com 3 serviços distintos, que são:
 1. *Cloud servers*: consiste em um serviço semelhante ao *EC2* da *Amazon*. Nesse o usuário pode configurar um servidor, de acordo com os recursos necessários, e a *Rackspace* provê uma instância virtualizada contemplando esses recursos. Nesse serviço os valores são cobrados conforme o tempo de utilização;
 2. *Cloud sites*: esse serviço provê a possibilidade de hospedar um número ilimitado de *websites* que são auto-escaláveis através de vários servidores da nuvem. Para a utilização desse serviço é cobrado um valor mensal para uma configuração básica oferecida, porém são oferecidos incrementos na configuração, dependendo das características do *website* a ser hospedado;
 3. *Cloud Files*: esse serviço é semelhante ao serviço *S3* da *Amazon*. Esse permite o armazenamento ilimitado de dados, porém com limite no tamanho dos arquivos de 5GB. A utilização mais comum do serviço de armazenamento da *Rackspace* consiste no armazenamento de cópias de segurança dos dados, armazenamento de arquivos para serem utilizados por aplicações e a distribuição de conteúdo na forma CDN¹⁰ (RACKSPACE, 2009). Para acesso aos dados a *Rackspace* provê ferramentas de acesso *web*, *desktop* ou através do uso de APIs específicas para a plataforma .NET. Essas APIs estão disponíveis também para as linguagens Ruby, PHP, Python ou Java. A cobrança desse serviço é realizada a partir da capacidade de armazenamento utilizada.

Uma referência completa dos valores e políticas de SLA dos serviços *Cloud Servers*, *Cloud Sites* e *Cloud Files* podem ser encontradas em (RACKSPACE, 2009).

¹⁰*Content Delivery Network*: sistema de computadores contendo uma cópia dos dados, localizados em vários pontos de uma rede com o objetivo de maximizar a banda para o acesso aos dados pelos clientes através da rede. Um cliente acessa uma cópia dos dados localizados no servidores mais próximos ao invés de todos os clientes acessarem ao mesmo servidor central causando limitação no acesso aos dados (RACKSPACE, 2009)

- *Nirvanix*: o *Nirvanix Storage Delivery Network (SDN)* surgiu em setembro de 2007, oferecendo uma alternativa ao serviço *S3* da *Amazon* para o armazenamento de arquivos na nuvem. Esse serviço permite o armazenamento de arquivos com tamanho individual de até 256GB. A Empresa conta com *storages* localizados nos Estados Unidos, Europa e Ásia que são acessíveis através de um *namespace* único provido pelo sistema de arquivo IFS (*Internet File System*). Esse serviço conta ainda com uma API que facilita o desenvolvimento e a migração de aplicações para a nuvem. Essa API possui suporte as linguagens Java, PHP, Zend, Python e C#, e aos protocolos de comunicação SOAP e REST. Para uma referência completa dos valores praticados e políticas de SLA, sugere-se (NIRVANIX, 2009).

2.6 Considerações Finais

Nesse capítulo foi feita uma breve descrição do paradigma de computação nas nuvens. Após, foram apresentadas as características básicas de uma nuvem computacional, e foi realizada uma breve descrição da infra-estrutura de uma nuvem. Em seguida foram apresentados vários desafios e oportunidades existentes na utilização de uma nuvem computacional. Foram destacadas ainda, vantagens e desvantagens desse modelo. Por fim, foram apresentados os serviços mais representativos baseados em computação nas nuvens, com suas características e funcionalidades. Para maiores informações sobre computação nas nuvens, sugere-se (MILLER, 2009)

No próximo capítulo serão descritos os componentes presentes em uma nuvem computacional, bem como, ferramentas disponíveis para a construção e o gerenciamento de uma infra-estrutura de nuvem.

3 INFRA-ESTRUTURA E GERENCIAMENTO DE UMA NUVEM COMPUTACIONAL

Para o desenvolvimento de uma nuvem, de forma geral, são necessários 3 componentes: computadores interconectados, um gerenciador e os serviços. Os computadores interconectados são responsáveis por fornecer os recursos para a infra-estrutura. Para gerenciar esses recursos, frequentemente, são utilizados *softwares* que facilitam essa tarefa. Por fim, os serviços consistem nas aplicações que são providas de forma dinâmica e provisionada. Nesta seção, serão descritos alguns aspectos sobre a infra-estrutura necessária para a criação de uma nuvem computacional e, além disso, serão apresentadas ferramentas que atuam como gerenciadoras da nuvem e que podem ser utilizadas para facilitar o desenvolvimento dessa.

3.1 Arquitetura de uma Nuvem Computacional

Uma infra-estrutura de uma nuvem computacional, como pode ser observado na Figura 3.1, consiste em 3 componentes principais: o controlador da Nuvem, o controlador de *cluster* e o controlador de nodo (WARDLEY; GOYER; BARCETS, 2009). Frequentemente, esses componentes são implementados separadamente, na forma de *web services*. Apesar de ser representada de forma distribuída, essa infra-estrutura pode ser simplificada. Por exemplo, uma mesma máquina pode, desempenhar as funções de controlador da nuvem, controlador de *cluster* e controlador de nodo, entretanto, essa abordagem não é recomendada, pois ela pode criar uma sobrecarga de funções provocando uma perda de performance da nuvem.

Além disso, a utilização de uma infra-estrutura distribuída permite que os componentes da nuvem estejam distribuídos em diferentes localizações geográficas. Isso ocorre, por exemplo, com a *Amazon EC2*, onde, atualmente, existem servidores localizados nos Estados Unidos e na Europa. A utilização dessa infra-estrutura ainda traz benefícios de segurança para a nuvem, pois os controladores de nodo e os controladores de *cluster* não são visíveis externamente à nuvem e, desta forma, não são suscetíveis a ataques como o componente que controla a nuvem (CLC).

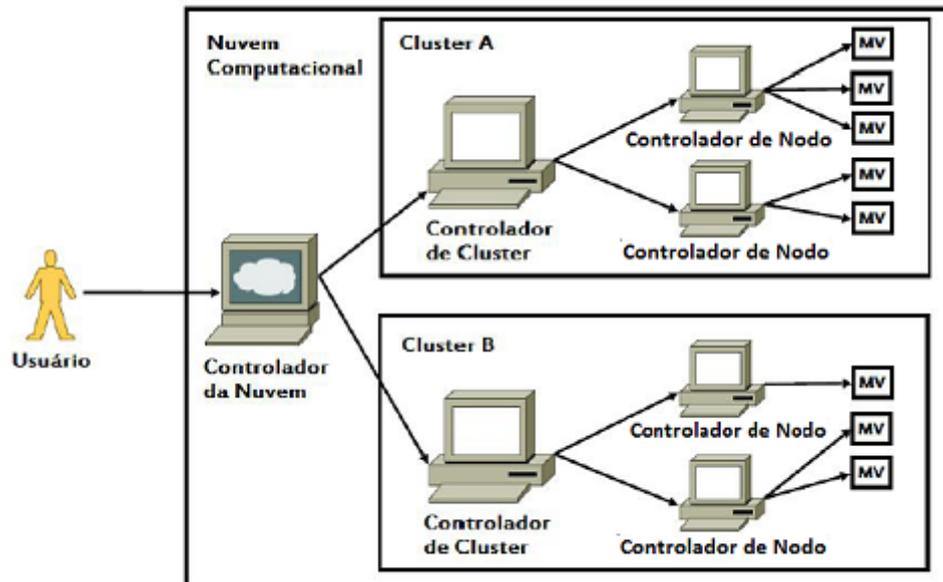


Figura 3.1: Componentes da infra-estrutura de uma nuvem computacional

A comunicação entre os componentes, representados na Figura 3.1, frequentemente é, realizada utilizando-se *web services*, que são componentes que permitem às aplicações enviar e receber dados em formato XML. Assim, cada componente pode ter a sua própria “linguagem”, que é traduzida para o formato XML. Os componentes utilizam *web services* de forma a facilitar a adição de novos recursos à nuvem, uma vez que para isso basta se desenvolver novos *web services*. Além disso, a utilização desses, facilita a integração com outras nuvens computacionais.

Por fim, a arquitetura distribuída facilita a manutenção da nuvem, uma vez que para realizar uma manutenção em um componente da nuvem não será necessário indisponibilizar toda a nuvem, e sim apenas o computador que estiver apresentando problemas. Nas seções seguintes, serão descritos os componentes da infraestrutura de uma nuvem computacional, suas características e funcionalidades.

3.1.1 Controlador da Nuvem

O controlador da nuvem (CLC) é o componente da arquitetura da nuvem mais visível na infra-estrutura, uma vez que esse provê a *interface* com a qual os usuários interagem com a nuvem. É através do CLC que os usuários fazem requisições de serviços a nuvem. Esse componente possui também a responsabilidade de realizar a autenticação do usuário e o controle de permissões. Além disso, deverá possibilitar ao usuário administrador executar rotinas administrativas na infra-estrutura da nuvem computacional.

Outra função do controlador da nuvem, é o controle de qualidade de serviço (QoS) e dos SLAs dos usuários, que devem ser ajustados de forma a atender as condições definidas no contrato. Por fim, o controlador da nuvem deve prover meios

que permitam medir a quantidade de recursos utilizados pelo usuário e, desta forma, possibilitar a cobrança pela utilização desses recursos.

O controlador da nuvem interage, ainda, com o controlador do nodo e mantém disponíveis informações sobre as máquinas virtuais que estão em execução na nuvem, bem como outras informações sobre o sistema de nuvem computacional. Assim, o CLC mantém o estado geral de uso do sistema. Além disso, esse é responsável por realizar operações de escalonamento em alto nível, de forma que, com posse de informações sobre os recursos disponíveis em cada controlador de *cluster*, ele poderá definir para qual controlador ele enviará a requisição de um usuário.

3.1.2 Controlador de Cluster

O componente da infra-estrutura chamado de controlador de *cluster* (CC) é um *software* que opera entre o controlador de nodo (CN) e o controlador da nuvem (CLC), desta forma, precisando ter acesso a rede em que pertencem ambos. A função desse componente é receber requisições para a alocação de máquinas virtuais vindas do controlador da nuvem e decidir em qual controlador de nodo, disponível em seu *cluster*, ele irá instanciar essa máquina virtual. Essa decisão é baseada em informações que o controlador do *cluster* recebe de cada controlador de nodo. Essas informações são obtidas através de inspeções periódicas a cada controlador de nodo.

Além de receber requisições provenientes do controlador da nuvem, o controlador de *cluster* interage com o controlador da nuvem informando para esse a capacidade disponível em seu *cluster*. É a partir dessa informação que o controlador da nuvem irá decidir em qual controlador de *cluster* (se a infraestrutura possuir mais de um) ele irá executar as novas instâncias.

Por fim, o controlador de *cluster* também é responsável por gerenciar as redes virtuais onde as máquinas virtuais executam. Esse efetua o roteamento das mensagens enviadas às máquinas virtuais e o tráfego de mensagens provenientes das máquinas virtuais. Este papel desempenhado pelo controlador de *cluster* depende do modelo de rede escolhido para executar as instâncias.

3.1.3 Controlador de Nodo

O controlador de nodo é o componente que é executado nos computadores que hospedam as instâncias das máquinas virtuais. Esse é responsável por controlar ações pertinentes a estas máquinas, como por exemplo, a inicialização e a desalocação dessa instância. Normalmente, um conjunto de controladores de nodo compõem uma arquitetura de uma nuvem computacional, mas é necessário apenas um controlador por máquina física, uma vez que um único controlador de nodo poderá gerenciar todas as instâncias virtuais que executam naquele computador.

Ao controlador de nodo cabe a função de gerenciar os recursos físicos do com-

putador em que está instalado, utilizando para isto de um *software* de virtualização conhecido como *hipervisor*¹. O *software* do controlador de nodo irá interagir com o *hipervisor* que está executando no nodo de forma a atender a uma requisição do controlador do *cluster*.

A comunicação do controlador de nodo com os outros componentes da infraestrutura é feita de forma que o controlador de nodo, quando for solicitado pelo controlador de *cluster*, obterá informações do ambiente em que está sendo executado, em termos de recursos disponíveis (tipo e número de processadores virtuais, memória e dispositivos de armazenamento), e irá encaminhá-las para o controlador de *cluster*.

3.2 Gerenciadores de Nuvem Disponíveis

No mercado existem diversos *softwares* para a criação de um ambiente de computação nas nuvens. Nessa seção serão descritas as ferramentas mais utilizadas e conhecidas.

3.2.1 Eucalyptus

O *Eucalyptus* é uma ferramenta de código aberto para o desenvolvimento de uma estrutura de computação nas nuvens do tipo *IaaS*. A *interface* atual do *Eucalyptus* é compatível com os serviços da *Amazon EC2* e *S3*, mas a infra-estrutura desse é desenvolvida de forma que, futuramente, esse possa suportar *interfaces* com outros serviços de nuvem.

O *Eucalyptus* originou-se no Departamento de Ciência da Computação da Universidade de Santa Bárbara, no estado americano da Califórnia, como um projeto de pesquisa e hoje é mantido pela companhia *Eucalyptus System*. Essa foi fundada pelos coordenadores do projeto com intuito de dar continuidade ao *software* criado. O *Eucalyptus*, atualmente, encontra-se disponível na versão 1.5.2 (EUCALYPTUS, 2009).

O projeto *Eucalyptus* tem atraído muita atenção, tanto que esse foi incluído na versão 9.10 do sistema operacional Linux *Ubuntu* (chamado *Karmic Koala*) (TAN; KIDDLE, 2009). Esse é um dos *softwares* mais reconhecidos para a construção de uma arquitetura de nuvem computacional e provê, a possibilidade de controlar e executar instâncias de máquinas virtuais hospedadas em diferentes computadores.

O *Eucalyptus* é distribuído através das licenças de *software* de código aberto *BSD License* e *Gnu Public License 3* (GPL3), porém, uma versão do *software* chamada de *Eucalyptus Enterprise Edition* é comercializada. Esta versão conta com a base do sistema *Eucalytpus* disponível como código aberto, porém conta com suporte a

¹*Software* de plataforma computacional de virtualização de *software* ou *hardware* que permite que vários sistemas operacionais possam executar em um único computador simultaneamente.

máquinas virtuais que utilizam o *hipervisor* da *VMware*. Já as versões de código aberto apresentam suporte apenas aos *hipervisores* *KVM* e *Xen*.

Como pode ser observado na Figura 3.2, a infra-estrutura de um ambiente desenvolvido com a ferramenta *Eucalyptus* conta com diversos componentes que interagem entre si. Nessa o CLC corresponde ao controlador da nuvem que provê a *interface* com a qual os usuários interagem com a nuvem. Os controladores de cluster (CCs) formam um *front-end* para cada *cluster* definido na nuvem. Abaixo dos CCs encontram-se os controladores de nodo (CNs) que são os computadores onde as instâncias das máquinas virtuais irão executar.

Pode ser observado ainda na Figura 3.2 dois outros componentes que são: o controlador de armazenamento (CA) e *Walrus*. O CA irá prover os dispositivos de armazenamento utilizados pelas máquinas virtuais individualmente, enquanto o serviço de armazenamento *Walrus* expande-se através de toda a nuvem e tem as funções similares ao *S3* da *Amazon*.

Já, a plataforma de gerenciamento, provê uma *console* para a administração, configuração e gerenciamento da nuvem, contando com níveis de acesso e privilégios customizados (EUCALYPTUS, 2009). Por fim, como pode ser visto na Figura 3.2, a nuvem pode ser integrada com outras nuvens, sendo que essa possui suporte nativo para a integração à nuvem da *Amazon*.

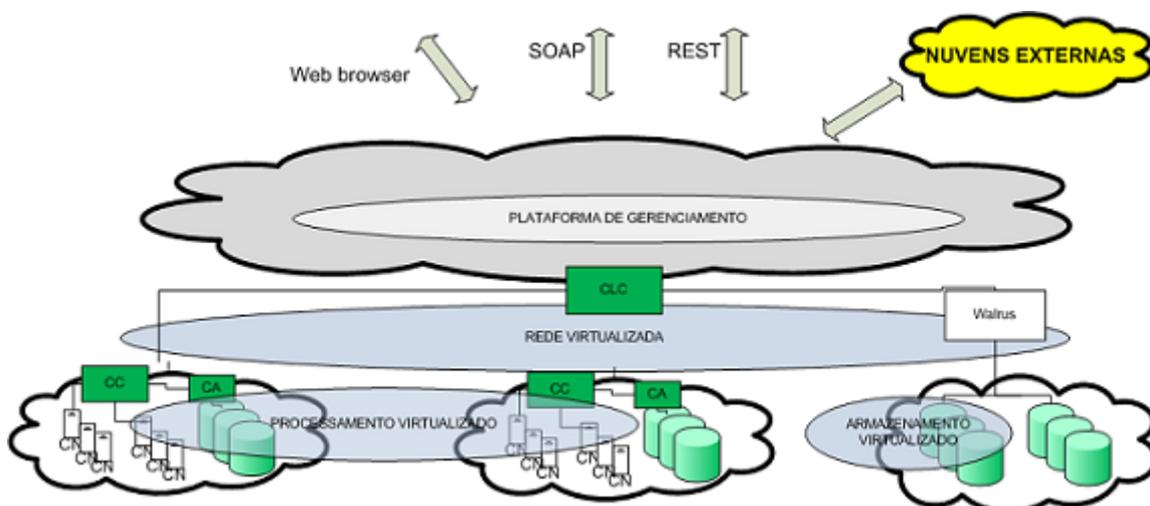


Figura 3.2: Arquitetura de uma Nuvem Computacional construída com Eucalyptus

No *Eucalyptus*, o serviço que provê o armazenamento de dados persistentes é o *Walrus Storage Controller (WS3)*. O *Eucalyptus* utiliza esse serviço para armazenar os dados em *buckets* acessados através do nome do *bucket* e uma chave única dentro do *bucket*. O serviço provê operações de armazenamento, exclusão e recuperação dos dados e permite realizar configurações de políticas de acesso (WARDLEY; GOYER; BARCETS, 2009). É importante ressaltar que a sua *interface* é compatível com o *S3* da *Amazon*, implementando os protocolos *REST* (*Representational State Transfer*)

e *SOAP* (Simple Object Access). O *Walrus* possui, ainda APIs de comunicação compatíveis com o protocolo do S3, além de possuir suporte aos arquivos do tipo AMI (*Amazon Machine Image*). O *Walrus* deve ser considerado como um sistema de armazenamento a nível de arquivos, não possuindo suporte de bloqueio a arquivos compartilhados (WARDLEY; GOYER; BARCETS, 2009).

O *Walrus* é um serviço que provê as necessidades mais simples de armazenamento mas não possui facilidades para o processamento de uma grande quantidade de dados ou dados que possuem relacionamentos entre si. Imaginando, por exemplo, o cenário de um setor de teste de qualidade e uma linha de produção de uma empresa. Uma grande quantidade de dados devem ser processados através de sistemas que definem relacionamentos entre esses dados. As relações são complexas e não são triviais de serem modeladas para acessarem um serviço de armazenamento baseado em chaves e valores da forma que o *Walrus* é implementado (TAN; KIDDLE, 2009). Neste caso, pode-se utilizar o *Walrus* como um serviço de persistência, em conjunto com ferramentas que permitam a construção de consultas mais complexas. Como um exemplo desse tipo de ferramenta pode-se citar o *Hadoop*, que é uma ferramenta de código aberto mantida pelo grupo Apache, que tem como objetivo possibilitar as aplicações processarem PB de dados em milhares de nodos (WHITE, 2009).

Na versão atual, que é a versão 1.5.2, o serviço *Walrus* deve ser executado na mesma máquina que o controlador da nuvem. Essa limitação estará sendo removida na versão 1.6, que já se encontra disponível para testes e avaliação. Para maiores informações sobre o *Eucalyptus*, sua estrutura e documentações, sugere-se (EUCALYPTUS, 2009).

3.2.2 Open Nebula

O *software OpenNebula* é uma ferramenta de código aberto que procura prover recursos para a construção de uma infra-estrutura de nuvem computacional. Essa ferramenta utiliza-se de diversas tecnologias de armazenamento, rede e virtualização para prover serviços de forma dinâmica. Além disso, o *OpenNebula* permite combinar os recursos de *datacenter* local com nuvens computacionais existentes, como por exemplo, nuvens computacionais que possuam *interfaces* compatíveis com o serviço *EC2* da *Amazon*.

O *OpenNebula* foi desenvolvido pelo grupo de sistemas distribuídos da Universidade Complutense de Madrid (OPENNEBULA, 2009). Esse foi desenvolvido como parte do projeto *Resources and Services Virtualization Without Barriers (RESERVOIR)*, que tinha como objetivo permitir a instalação de serviços de computação em larga escala e o gerenciamento de serviços de tecnologia da informação entre diferentes domínios administrativos, plataformas e regiões geográficas. A partir do *OpenNebula*, poder-se-ia prover serviços onde, através da utilização de tecnologias

de virtualização, os recursos computacionais e os serviços eram provisionados de forma transparente e gerenciados sob demanda (RESERVOIR, 2009). Atualmente o projeto *Reservoir* é financiado por grandes empresas da área da informática, como por exemplo, IBM e SUN, assim como, empresas de telecomunicações como por exemplo a Telefônica.

Atualmente, o *OpenNebula* é distribuído através da licença *Apache 2.0*, que permite o uso e distribuição do código fonte, tanto em *software* de código aberto como no *software* proprietário. A utilização da licença *Apache*, permite que a essa seja incluída por referência e, além disso, permite a incorporação do *software* em *softwares* proprietários, sob a condição que essa licença seja incorporada.

Em alto nível, como pode ser observado na Figura 3.3, a infra-estrutura do *OpenNebula* pode ser representada através de um serviço responsável por gerenciar a infra-estrutura virtual, e por receber as requisições de usuários internos e externos. Além disso, o *OpenNebula*, permite a integração com nuvens públicas compatíveis, como por exemplo, a *Amazon EC2*.

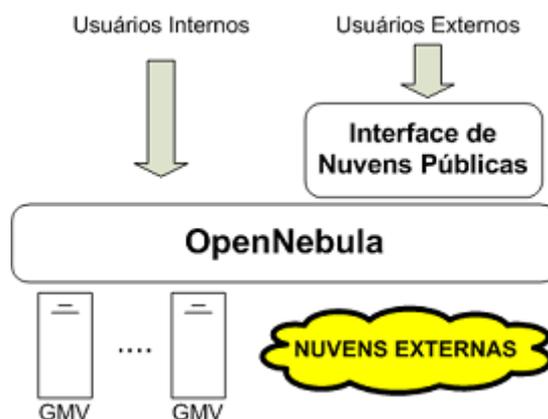


Figura 3.3: Arquitetura de uma Nuvem Computacional construída com o OpenNebula

Basicamente, a arquitetura interna do *OpenNebula* pode ser dividida em 3 camadas (Figura 3.4), que são:

- Núcleo: contém os principais componentes da arquitetura do *OpenNebula*, que são: gerenciador de rede virtual, que tem a função de disponibilizar endereços de rede para as máquinas virtuais; o gerenciador de máquinas virtuais, que tem a função de gerenciar o ciclo de vida das máquinas virtuais; o gerenciador de nodo, que tem como função armazenar informações referentes aos nodos da nuvem e as informações necessárias para a comunicação e a interação com eles; e o componente XML-RPC, que tem como função prover uma *interface* única para o gerenciamento e obtenção das informações das entidades do *OpenNebula*;

- **Ferramentas:** compostas por ferramentas de gerenciamento que utilizam-se das *interfaces* providas pelo núcleo do *OpenNebula*. Dentre as ferramentas, pode-se citar o escalonador e ferramentas de linhas de comando. O escalonador tem como função efetuar a busca por nodos onde possam ser instanciadas as máquinas virtuais. Já as ferramentas de linha de comando são utilizadas para o gerenciamento das entidades do *OpenNebula* (nodos, redes e máquinas virtuais);
- **Drivers:** componentes que permitem a utilização de diferentes plataformas de virtualização, armazenamento, monitoramento e serviços de nuvem. Existem 3 tipos de *drivers* que são: *drivers* de informação, que possuem a função de obter informações sobre os recursos dos nodos; os *drivers* de máquinas virtuais, que gerenciam o ciclo de vida das máquinas virtuais criadas; e os *drivers* de transferência, que possuem a função de realizar operações administrativas sobre as máquinas virtuais, como por exemplo, clonagem e exclusão.

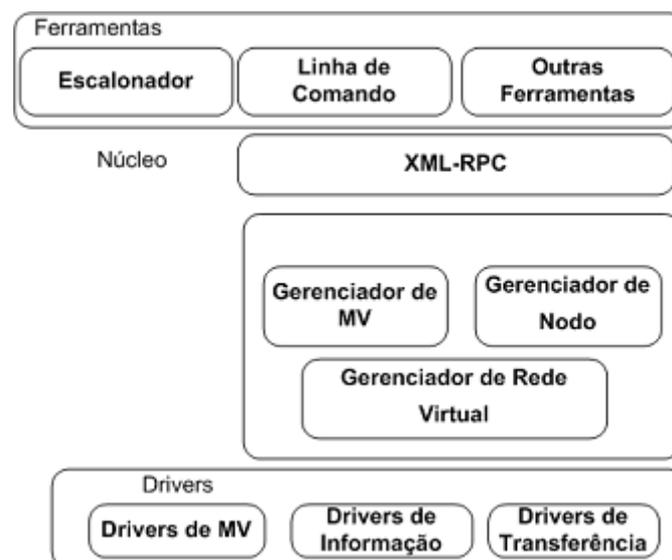


Figura 3.4: Estrutura interna do OpenNebula

Para gerenciamento do provisionamento das máquinas virtuais o *OpenNebula* utiliza-se do *Haizea* (HAIZEA, 2009). Essa ferramenta possibilita ao provedor disponibilizar os recursos computacionais contratados pelos consumidores. Através da utilização do *Haizea*, é possível definir os termos do contrato de provisionamento a nível de recursos de *hardware*, ambientes de *software* e o período em que esses recursos deverão estar disponíveis (SOTOMAYOR, 2008).

Com relação ao armazenamento, o *OpenNebula* suporta diferentes configurações para o armazenamento de máquinas virtuais e dados do usuário. As configurações suportadas, na versão atual, do *OpenNebula* são:

- Repositório de imagens: define os dispositivos locais e remotos onde as imagens das máquinas virtuais serão armazenadas. Como exemplo de armazenamento remoto pode-se citar um servidor de arquivos remoto ou uma *URL* remota. É importante destacar que o repositório de imagens deve estar acessível ao controlador da nuvem, pois a transferência é realizada através desse;
- Diretório para máquinas virtuais: é o diretório no controlador de nodo onde a máquina virtual é executada. Todos os arquivos necessários para a máquina virtual, como por exemplo, imagens, arquivos de inicialização e estados da máquina virtual salvos são armazenados nesse diretório. Esse diretório deve estar acessível por todos os *hipervisores*, de forma a permitir operações de migração das máquinas virtuais entre nodos (*live migrations*).

Para maiores informações sobre o *OpenNebula*, sua estrutura e documentações, sugere-se (OPENNEBULA, 2009).

3.2.3 VMware vSphere

A *VMware* é atualmente a empresa líder em *softwares* para virtualização. Essa foi fundada em 1998, e um ano mais tarde desenvolveu um produto chamado *VMware Workstation*, que consistia em um ambiente computacional baseado em virtualização. Com o avanço da virtualização, a *VMWare* lançou outros produtos, destacando-se entre esses: o *VMware Server*, que é um *software* para virtualização de servidores; e o *VMWare Workstation*, que é um *software* para a execução de vários “computadores virtuais” dentro de um único sistema operacional.

Em 2003, foi lançado o *VMware Infrastructure*, que é um *software* que provê soluções para a virtualização e a agregação de servidores, dispositivos de armazenamento e de rede em um conjunto de recursos compartilhados (VMWARE, 2009a). Esse basicamente é uma integração entre o *VMware Workstation* e o *VMware Server*, além de apresentar melhorias nesses dois *softwares*.

Já no ano de 2009, a *VMware* lançou uma nova versão do *VMware Infrastructure*, o *VMware vSphere*, que tem como principal objetivo facilitar o desenvolvimento de uma nuvem computacional a partir de um *datacenter* (VMWARE, 2009b). O *software* atua ainda como o gerenciador de nuvem computacional da *VMware*.

O *VMware vSphere* foi estrategicamente desenvolvido de forma modularizada, com o objetivo de facilitar a comercialização de diferentes versões. Essas versões diferenciam-se por agregarem ou não as funcionalidades dos módulos. Em (VMWARE, 2009a), podem ser encontradas as versões disponíveis, os valores dessas e as funcionalidades incorporadas em cada versão do *vSphere*. De forma resumida, as versões disponíveis são:

- *Standard*: provê uma solução básica para a consolidação de aplicações. O valor da licença com suporte pelo período de 1 ano é de U\$\$ 1.118,00;
- *Advanced*: além de prover os recursos de virtualização da versão *Standard*, essa versão possibilita a migração de máquinas virtuais, possui suporte a tolerância a falhas e cópias de segurança das máquinas virtuais que executam no servidor. O valor da licença é de U\$\$ 2.806,00;
- *Enterprise*: esse possui todas as funcionalidades das versões anteriores e, além disso, provê meios para minimizar o tempo inatividade, proteção dos dados e também automatização do gerenciamento dos recursos. A *VMWare* comercializa essa versão pelo valor de U\$\$ 3.594,00;
- *Enterprise Plus*: essa versão possui todas as configurações fornecidas pela versão *Enterprise*. Além disso, possibilita a cópia da configuração de servidores com objetivo de replicar essa configuração em outras máquinas. Todas as configurações disponíveis oferecidas pelo produto *vSphere* da *VMware* estão presentes nesse produto, e o valor da sua licença é de U\$\$ 4.369,00.

Com relação ao serviço de armazenamento, o *Vsphere* provê todas as características de armazenamento das versões anteriores do *VMWare Server* e *Workstation*, como por exemplo, o armazenamento das máquinas virtuais, em discos locais ou discos remotos, gerenciados através do *VMware vStorage VMFS*. Além disso, esse permite o acesso compartilhado a esses dispositivos por diferentes máquinas virtuais, gerenciando o acesso através de bloqueios no sistema de arquivos. A implementação atual do *VStorage*, que está incluída na versão 4.0 do *VSphere*, possui características adicionais, como por exemplo, o provisionamento da capacidade de armazenamento através do *Storage Vmotion*, melhorias no suporte de dispositivos de armazenamento que utilizam comunicação *iSCSI* e a possibilidade de expansão de volumes *VMFS* sem a necessidade de interrupção das máquinas virtuais em execução.

Diferentemente dos *softwares* gerenciadores de nuvem de código aberto, como por exemplo, o *Eucalyptus* e o *OpenNebula*, o *VMware vSphere* não possui recursos que permitem integrar a sua nuvem computacional a outras nuvens computacionais, como por exemplo, o *EC2* da *Amazon*.

A ferramenta *VMware Vsphere* possui licença proprietária, e para obter-se maiores informações sobre o produto e valores praticados para licenciamento, deve-se entrar em contato com a empresa através do seu website (VMWARE, 2009a).

3.2.4 Citrix Cloud Center (C3)

A *Citrix Systems* é uma empresa norte-americana fundada em 1989 que inicialmente desenvolvia *softwares* para o sistema operacional da IBM, o OS/2. Porém, a

empresa somente ficou conhecida através do *software WinView* que era fornecido à *Microsoft*. Mais tarde esse *software* foi licenciado como *Windows Terminal Server* e é um dos *softwares* mais utilizados para prover acesso remoto a computadores (HARWOOD, 2004). Atualmente, a empresa conta com um produto de acesso remoto chamado *Metaframe*, através do qual, torna-se possível gerenciar aplicações corporativas e disponibilizá-las para acesso remoto.

A empresa *Citrix* passou a trabalhar com virtualização em 2007, quando adquiriu a empresa *XenSource* que distribuía o *hipervisor Xen*. Esse foi desenvolvido na Universidade de Cambridge em 2003. Atualmente, a *Citrix* dá suporte no desenvolvimento do projeto de código aberto do *Xen* e comercializa também uma versão proprietária do *software*.

Já, o *Citrix Cloud Center*, conhecido como *C3*, foi lançado no ano de 2008 e consiste em uma ferramenta para o gerenciamento de uma infra-estrutura de nuvem computacional (CITRIX, 2009). O *C3* foi projetado para dar aos provedores de serviços baseados em nuvem, um conjunto completo de serviços para a hospedagem, gerenciamento e disponibilização de serviços. Basicamente a arquitetura do serviço *C3* apresenta 5 características, que são:

- **Plataforma:** o *C3* conta com o *Citrix XenServer*, que é um conjunto de ferramentas de virtualização que permite os provedores manter e gerenciar servidores virtualizados com os ambientes *Linux* e *Windows*;
- **Serviços e aplicações *desktop*:** para o desenvolvimento de uma infra-estrutura de *IaaS*, o *C3* conta com o *Citrix XenApp*, que é um sistema para a distribuição de aplicações *Windows*, de forma elástica no modelo *SaaS*. Além disso, o *C3* conta com o *Citrix XenDesktop*, que é uma ferramenta para a virtualização de *desktops*;
- **Distribuição:** o *C3* possui uma ferramenta para a distribuição de recursos da nuvem através da *web* chamada *Citrix NetScaler*. Essa permite aos provedores escalonar o processamento através dos ambientes de nuvem computacional e, de forma transparente, realizar operações de tolerância a falhas, redirecionando o tráfego e o processamento entre os servidores disponíveis;
- **Bridge:** para o tráfego de dados entre a infra-estrutura local de uma empresa e uma nuvem computacional, é necessário a construção de uma conexão entre a nuvem e o *datacenter* da empresa. O *C3* fornece um conjunto de *interfaces* que permitem realizar a comunicação entre as duas infra-estruturas e, além disso, essa disponibiliza mecanismos para a otimização dessa comunicação entre as duas pontas da conexão;

- Automação: os produtos que integram o C3 podem ser gerenciados de forma dinâmica e automatizada. Além disso, esses podem ser integrados a regras de negócio e políticas do departamento de TI. Para isso, o C3 disponibiliza uma ferramenta chamada *Citrix Workflow Studio*. Através dessa é possível escalar os recursos sob demanda, ou seja, utilizar somente os recursos de *hardware* necessários, sendo que os demais recursos são desligados automaticamente.

Diferentemente do *OpenNebula* e *Eucalyptus*, o *C3* da *Citrix* não apresenta funcionalidades visando a integração da nuvem computacional com outras já existentes. Entretanto, no segundo semestre de 2009, a *Citrix Systems* anunciou uma nova solução para nuvens chamada de *Citrix C3 Lab* que inclui o *Amazon Web Services (AWS)*. O *Citrix C3 Lab* foi desenvolvido de forma a oferecer um ambiente pré-configurado para profissionais de TI que queiram testar soluções usando aplicativos já em produção e inclui os componentes necessários para a criação de um centro de distribuição de serviços na nuvem. Esse é baseado nas tecnologias *Amazon Elastic Compute Cloud (EC2)*, *Amazon Simple Storage Service (S3)* e *Citrix Cloud Center (C3)* (CITRIX, 2009).

A ferramenta da *Citrix* possui licença proprietária, e para obter-se maiores informações sobre o produto e valores praticados para licenciamento, deve-se entrar em contato com a empresa através do seu *website* (CITRIX, 2009).

3.2.5 Outras ferramentas gerenciadoras disponíveis

Além das ferramentas gerenciadoras de nuvem descritas nas seções anteriores, existem outras ferramentas que são brevemente descritas nessa seção. São elas:

- *Enomalism*: fundada em 2004 a *Enomaly* é uma das empresas líderes no mercado de computação nas nuvens. A *Elastic Computing Platform (ECP)* da *Enomaly* é uma infra-estrutura de nuvem virtual programável para pequenas, médias e grandes empresas. O ECP possui recursos de forma a facilitar o projeto, distribuição e gerenciamento de aplicações virtuais em nuvem. O ECP teve a sua primeira versão lançada em 2005 e, atualmente, está disponível em 3 versões: *Community* que é a versão gratuita do *software*; *Service Provider* que é uma versão oferecida para provedores; e a *Datacenter Edition* que permite a criação de nuvens computacionais públicas, privadas e híbridas. Para maiores informações sobre as versões disponíveis, licenciamento, e documentações do ECP, sugere-se (ENOMALISM, 2009);
- *Nimbus*: o *Nimbus* é um conjunto de ferramentas de código aberto que provê uma solução de infraestrutura do tipo *IaaS*. A primeira versão do *Nimbus* foi concebida em 2005, como objetivo de facilitar a transformação de um *cluster*

em um ambiente de computação em nuvem. O foco principal do *Nimbus* é a construção de infra-estrutura que permite a execução de aplicações científicas, mas esse pode ser utilizado também em soluções corporativas. O *Nimbus*, assim como o *OpenNebula*, é distribuído através da licença *Apache 2.0*. Essa licença permite o uso e distribuição do código fonte tanto no *software* de código aberto como no proprietário. Para maiores informações, sugere-se (NIMBUS, 2009).

3.3 Considerações Finais

Nesse capítulo, foi feita uma descrição de uma infra-estrutura baseada em computação nas nuvens, bem como de seus principais componentes. Após, foram apresentados alguns *softwares* que executam as funções de gerenciamento de uma nuvem computacional.

A complexidade da infra-estrutura de uma nuvem é proporcional aos serviços e recursos que a mesma deseja prover. Um gerenciador de nuvem computacional eficaz, deve prover um gerenciamento eficiente da infra-estrutura do ambiente, desde o gerenciamento dos recursos das máquinas virtuais que executam nos nodos, até a integração com outros serviços que executam fora dela. Um gerenciamento não eficaz, pode acarretar em problemas de indisponibilidade da nuvem, ou ainda em uma má distribuição dos recursos de processamento e armazenamento.

Após ter sido feito uma apresentação da infra-estrutura de uma nuvem computacional e um estudo sobre as ferramentas que permitem a construção e o gerenciamento dessa infraestrutura, será realizado, no próximo capítulo, um projeto de uma ambiente de armazenamento de arquivos baseado em computação nas nuvens.

4 CONSTRUÇÃO DO AMBIENTE

O objetivo do presente trabalho é o desenvolvimento de uma infra-estrutura para o armazenamento de arquivos baseada em computação nas nuvens. A infra-estrutura será semelhante ao serviço *Simple Storage Service (S3)* da *Amazon* e o *Cloud Files* da *Rackspace*.

Através da infra-estrutura desenvolvida, os arquivos devem ser armazenados em *containers* chamados *buckets* que permitam o acesso e a manipulação dos mesmos através de comandos simples de escrita, leitura e exclusão. Os arquivos armazenados nessa infra-estrutura devem possuir controle de acesso e devem ser acessíveis de qualquer computador conectado na *internet*. Além disso, deve ser abstraído do usuário a configuração dos recursos de *hardware*, bem como a localização dos dados acessados. O acesso dessa infra-estrutura poderá ser realizado através de um cliente *desktop*, através de *web services* ou, ainda através de uma API.

Com relação às APIs de acesso, a infra-estrutura deve ter uma API de acesso compatível com a utilizada pelo serviço *S3*, uma vez que o serviço de armazenamento da *Amazon* é amplamente utilizado. A utilização de uma API compatível proporcionaria a integração da nuvem privada desenvolvida com serviços da *Amazon*.

Além desses objetivos principais de armazenar arquivos em uma infra-estrutura de nuvem computacional, o desenvolvimento desse trabalho também possui alguns objetivos secundários. Um desses objetivos secundários é que a infra-estrutura deve ser escalável, ou seja, deverá permitir que outros servidores sejam adicionados ao sistema de forma a aumentar a capacidade de armazenamento que é provida ao usuário sob demanda. Outro objetivo é permitir que o sistema possa armazenar arquivos de qualquer tipo e de diferentes tamanhos. Além disso, a infra-estrutura deve prover meios para realização de cópias de segurança. Não serão abordados nesse trabalho assuntos como tolerância a falhas ou replicação dos dados. No entanto, tais assuntos poderão ser retomados na ocasião de trabalhos futuros.

Nesse capítulo será apresentada a nuvem computacional desenvolvida. Na Seção 4.1, será abordado detalhadamente a arquitetura física da nuvem computacional construída, enquanto na Seção 4.2 serão abordados os detalhes dos *softwares* uti-

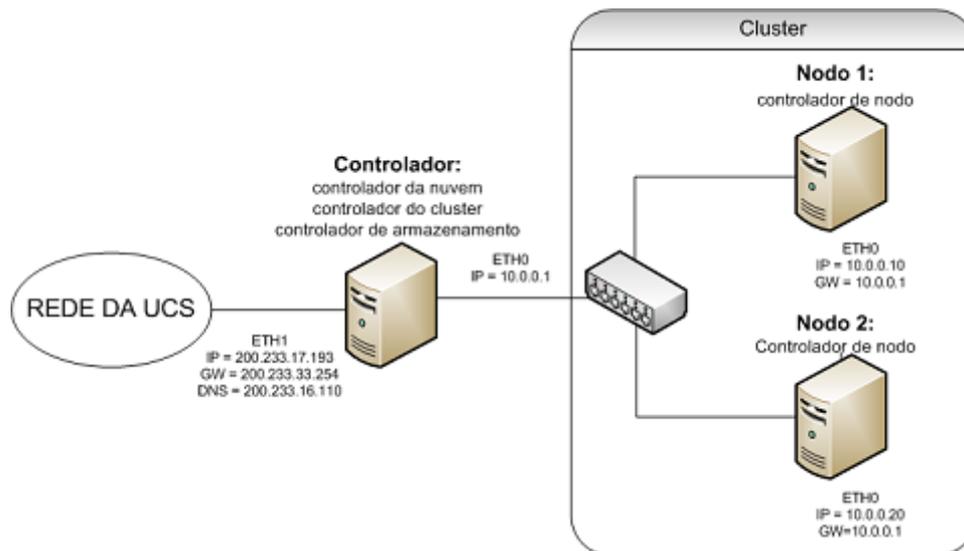


Figura 4.1: Arquitetura da nuvem computacional construída

lizados. Por fim, na Seção 4.3, serão apresentados os resultados obtidos com a construção da infra-estrutura proposta.

4.1 Arquitetura Física da Nuvem Computacional

Para o desenvolvimento da infra-estrutura de nuvem computacional, foram utilizados 3 computadores, sendo que 2 desses foram utilizados como nodo da nuvem, recebendo as instâncias de máquinas virtuais que serão executadas na infra-estrutura. Já, o terceiro computador é responsável por executar as funções de controlador da nuvem, armazenamento de dados e controlador do *cluster*. Nesse caso, o *cluster* foi formado utilizando-se os 2 nodos da nuvem. As máquinas que compõem a nuvem computacional possuem a seguinte configuração de *hardware*: processador Intel Pentium IV com tecnologia HT¹ e plataforma de 32 *bits*, 1 GB de memória RAM, 40 GB de capacidade de disco rígido e 2 placas de rede, sendo 1 de 100Mb *onboard* e uma *offboard* de 1Gb.

Na Figura 4.1, é apresentada a arquitetura da nuvem computacional construída, onde é possível observar as máquinas utilizadas, as funções delegadas a cada máquina, e as conexões entre elas. As máquinas com função de nodo comunicam-se apenas com o controlador do *cluster*, formando uma rede interna entre as máquinas que compõem a nuvem. A rede configurada é uma rede classe A com endereço $10.0.0.0/8$, onde a partir dessa configuração é possível endereçar aproximadamente 16 milhões de máquinas. Ainda é possível observar na Figura 4.1, os endereços atribuídos a cada máquina: o *nodo1* está configurado com o endereço $10.0.0.10$ e o *nodo2* está confi-

¹*Hyper threading*: tecnologia usada em processadores que o faz simular dois processadores tornando o sistema mais rápido quando se usa vários programas ao mesmo tempo.

gurado com o endereço *10.0.0.20*. A máquina controladora da nuvem comunica-se com os nodos através da *interface eth0* que corresponde à placa de rede de 1 GB, e que está configurado com o endereço IP *10.0.0.1*.

Ainda na Figura 4.1, pode ser observado que a máquina controladora da nuvem, nomeada de controlador, comunica-se através da sua *interface* de rede *eth1* de *10/100 Mb* com a rede da UCS. Dessa forma, é possível permitir a conexão de qualquer máquina conectada a rede da UCS ao controlador da nuvem, e assim provendo aos usuários dessas máquinas a possibilidade de interagirem com a nuvem computacional criada. Porém, não é possível o acesso direto aos nodos da nuvem.

4.2 Softwares utilizados na Nuvem Computacional

Para a construção da nuvem computacional proposta, foi utilizado o *Eucalyptus*, que foi apresentado na Seção 3.2.1. A escolha do *Eucalyptus* como ferramenta gerenciadora da nuvem computacional baseou-se nas seguintes premissas:

- Ser um *software* de código aberto;
- Possuir *interfaces* semelhantes aos serviços de computação da *Amazon*, *EC2* e *S3*, que são amplamente utilizados por usuários de nuvens públicas;
- Forte utilização pela comunidade de *software* livre;
- Documentação de fácil acesso, além de possuir meios de contato facilitados com os desenvolvedores da ferramenta.

Inicialmente a proposta era utilizar o sistema operacional *Ubuntu* que na sua versão atual 9.10, conhecida como *Karmic* e que foi lançada no início de novembro de 2009, já possui integrado o *software* do *Eucalyptus*. Porém, com a utilização do *Ubuntu*, ocorreram problemas, isso por que, as máquinas utilizadas não possuíam em seus processadores a disponibilidade da tecnologia de virtualização (VT), necessária para a utilização do *KVM*, que é o *software* de virtualização utilizado pela UEC (*Ubuntu Enterprise Cloud*). Devido a impossibilidade da utilização do *KVM* nesses computadores, decidiu-se por utilizar o *software* de virtualização *Xen*, que também é suportado pelo *Eucalyptus*. Entretanto, a única versão do *Ubuntu* onde conseguiu-se realizar a configuração do *Xen* foi a versão 8.04, que foi lançada em abril de 2008. Uma vez que a versão do sistema operacional *Ubuntu* que suportou a configuração inicial era antiga, optou-se pela utilização da distribuição Linux *CentOS*, que é uma outra distribuição Linux que possui compatibilidade com o *Eucalyptus*.

O sistema operacional CentOS (*Community Enterprise Operating System*) é um sistema operacional para servidores empresariais gratuito mantido pela *RedHat*, uma

das empresas mais significativas no mercado de sistemas operacionais empresariais Linux. A versão utilizada do *CentOS* foi a versão 5.4, que foi lançada em outubro de 2009.

Em cada nodo da nuvem computacional foi instalado o sistema operacional *CentOS* e instalados os pacotes necessários para a utilização do computador como um nodo da nuvem. O procedimento de instalação dos pacotes necessários pode ser encontrado nos anexos. Primeiramente foi instalado o serviço do *Xen*, que necessita de um *kernel* compilado com algumas configurações específicas. Esse serviço, que executa em cada nodo, tem a função de atuar como o *hipervisor*, ou seja, esse controla as máquinas virtuais em execução em cada nodo da nuvem.

Além do *Xen*, nos computadores utilizados como nodo, fez-se necessário a configuração de um cliente de NTP (*Network Time Protocol*), para a sincronização dos relógios dos nodos com o controlador da nuvem. O serviço de NTP é um protocolo para a sincronização dos relógios de computadores, ou seja, ele define a forma como um grupo de computadores possa conversar entre si, de modo a sincronizar seus relógios, baseados em uma fonte confiável de tempo, como por exemplo, um servidor de tempo disponível através da *internet*.

Nos computadores utilizados como nodos, efetuou-se também a desativação do *software* de *firewall* do Linux, permitindo assim a conexão com a porta 8775, necessária para a comunicação com o controlador do *cluster*. Além disso, em cada nodo foi instalado o módulo de controlador de nodo do *Eucalyptus*, que é o *software* que será responsável pela inicialização das máquinas virtuais e monitoramento de estado das instâncias que estão executando, de acordo com as requisições enviadas pelo controlador da nuvem.

Para o computador que tem as funções de controlador da nuvem, foi instalado o sistema operacional *CentOS* com o *kernel* sem customizações, e os requisitos necessários para a instalação do *Eucalyptus*, que são:

- Servidor DHCP: serviço que executa na máquina controladora que disponibiliza endereços de rede para as máquinas virtuais criadas. Quando uma nova máquina virtual é instanciada, essa recebe um endereço de IP do servidor de DHCP;
- Servidor NTP: serviço que provê o sincronismo dos relógios das máquinas da nuvem computacional com o controlador da nuvem. Dessa forma, as máquinas que compõem a nuvem computacional terão os seus relógios sincronizados;
- Compilador java: necessário para compilar e executar os módulos do *Eucalyptus* que executam em um *container web*;

- Utilitários de *bridge*: responsável por efetuar o roteamento das mensagens entre as máquinas virtuais criadas e o controlador da nuvem computacional;
- *Eucalyptus cloud*: módulo do *Eucalyptus* responsável por executar as funções de controlador da nuvem;
- *Eucalyptus walrus*: módulo do *Eucalyptus* responsável por gerenciar e controlar o armazenamento de dados e as imagens das máquinas virtuais registradas na nuvem;
- *Eucalyptus cluster*: módulo do *Eucalyptus* responsável por executar as funções de controlador de *cluster* da nuvem computacional.

Além das configurações acima citadas, no controlador da nuvem, as portas 8443, 8773 e 8774 devem estar liberadas, de forma a permitir a comunicação do *Eucalyptus* com os demais componentes da nuvem. Por fim, o serviço de armazenamento *Walrus* foi configurado, de forma que todos os parâmetros sejam compatíveis com o poder de armazenamento que está sendo fornecido pela nuvem, que são:

- Número máximo de *buckets* por usuário: optou-se por configurar 5 *buckets* por usuário;
- Capacidade máxima de armazenamento de cada *bucket*: foi configurado de forma que cada *bucket* possua um limite de 5 GB;
- Capacidade reservada para o *cache* de imagens: é responsável pelo armazenamento das imagens que serão utilizadas para a instanciação das máquinas virtuais, e foi configurado com um limite de 10 GB;
- Capacidade reservada para os *snapshots*: a capacidade em disco, reservada para os *snapshots* das máquinas virtuais que executam nos nodos, foi de 20 GB.

4.3 Acessando a Nuvem

O acesso à nuvem pode ser realizado através de um navegador *web*, conforme apresentado na Figura 4.2. Inicialmente as credenciais para acessar o *Eucalyptus* são usuário *admin* e senha *admin*, e essas deverão ser trocadas após a realização do primeiro acesso. Após o administrador da nuvem tem acesso às telas de ajustes dos parâmetros de configurações de *cluster*, armazenamento do controlador da nuvem, etc.

Quando um novo usuário deseja obter credenciais para acessar a *interface* do *Eucalyptus*, ele acessa o *website* da nuvem, da mesma forma que o administrador, e

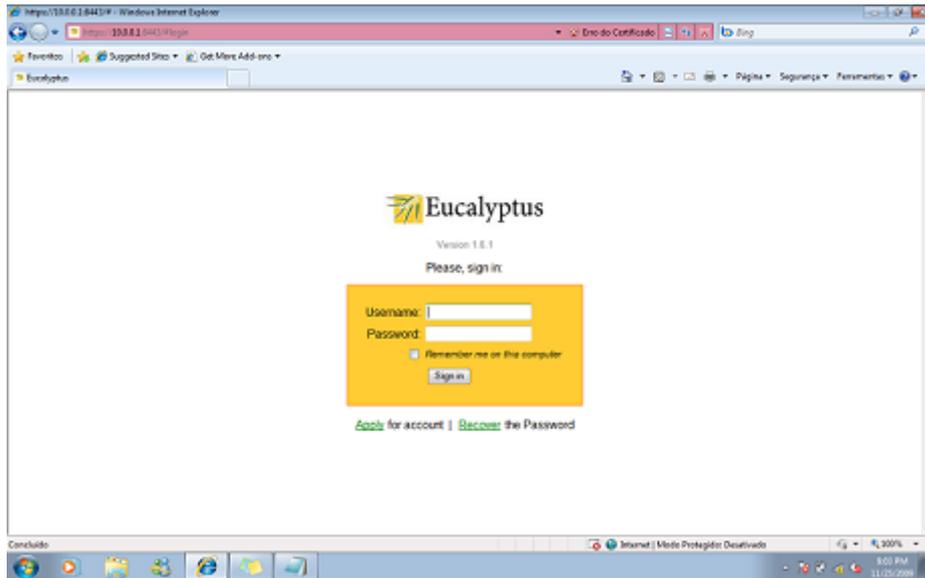


Figura 4.2: Tela inicial do *Eucalyptus*

solicita a criação de um novo usuário, que será liberado pelo administrador da nuvem. As credenciais de acesso são obtidas de duas formas: efetuando o *download* de um arquivo zipado, disponível na página do controlador da nuvem, que será utilizado pela ferramenta *euca2ools* (detalhes da instalação da ferramenta estão inseridos no Anexo deste trabalho); ou através da utilização da ID de acesso e da chave secreta também obtidas a partir do controlador da nuvem. O ID e a chave secreta serão utilizados também para o acesso através de outras ferramentas e API's.

4.4 Testes Realizados

Para testar se os objetivos propostos pela infra-estrutura desenvolvida foram alcançados, foram criados alguns cenários de teste. Esses cenários foram definidos, de forma a verificar se a nuvem desenvolvida atende aos objetivos definidos no início desse capítulo. Os testes realizados foram:

- Remoção e inclusão de recursos: a nuvem computacional deve prover meios à inclusão e remoção de recursos sob demanda. Esse teste tem como objetivo verificar o comportamento da nuvem computacional quando são retirados ou incluídos nodos na infra-estrutura da nuvem;
- Realização de cópias de segurança: um dos objetivos desse trabalho é a possibilidade de realizar cópias de segurança dos dados que estão na nuvem. Esse teste tem como objetivo verificar a possibilidade realizar *backups* dos arquivos da nuvem;
- Aumento da capacidade de armazenamento: o aumento da capacidade dis-

ponível de armazenamento, é um dos objetivos da infra-estrutura proposta. Esse teste foi realizado para verificar se o ambiente de armazenamento provê essa característica;

- Padronização de API: o ambiente de arquivos desenvolvido deve prover uma interface compatível com o serviço S3 da *Amazon*. Esse teste tem como objetivo verificar a possibilidade de acesso aos arquivos da nuvem, através da utilização de ferramentas que foram desenvolvidas para a utilização dos serviços de armazenamento da *Amazon*;
- Confidenciabilidade dos dados: esse teste tem como objetivo verificar a possibilidade de atribuição de permissões de acesso simples aos arquivos que são armazenados na nuvem, de forma semelhante ao serviço de armazenamento de arquivos da *Amazon*;
- Utilização do cache de imagens: para executar uma instância de máquina virtual em um nodo da nuvem, é necessário copiar a imagem da máquina virtual para o nodo. Esse teste tem como objetivo verificar se, de alguma forma, é feito cache dessas imagens para os nodos da nuvem, de forma a melhorar o desempenho da nuvem;
- Transferência de arquivos entre cliente e *Walrus*: uma das ideias do armazenamento de arquivos na nuvem é melhorar o atendimento às requisições. Para testar se esse objetivo foi alcançado, foram realizados alguns testes de envio de arquivos para a nuvem computacional.

4.4.1 Remoção ou inclusão de Recursos

Esse teste tem com objetivo verificar o comportamento da nuvem quando um nodo fosse retirado, seja para manutenção ou por algum outro problema. Para esse teste, foram solicitadas 4 instâncias de máquinas virtuais. O controlador do *cluster* analisou os recursos dos nodos disponíveis e alocou 2 instâncias de máquinas virtuais em cada um dos nodos.

Em seguida, como pode ser observado na linha 1 do trecho de *log* da Figura 4.3, foi retirado o *nodo2* da nuvem, alterando o arquivo de configuração do *Eucalyptus* que possui a lista de nodos ativos na nuvem. O controlador do *cluster* automaticamente atualizou as informações sobre os nodos que formam o *cluster*, como é apresentado na linha 2 da Figura 4.3, onde pode ser observado que esse *cluster* agora é formado apenas pelo *nodo1*. Após, o controlador do *cluster* encerrou as instâncias de máquinas virtuais, como mostra a linha 5 da Figura 4.3.

Já, na Figura 4.4 tem-se o arquivo de log do *nodo2*, que foi removido da configuração da nuvem computacional e, conseqüentemente não teve mais troca de men-

```

1-[19:44:59] [EUCAINFO] config file has been modified,
refreshing node list
2-[19:44:59] [EUCADEBUG] DescribeNetworks(): called
3-[19:44:59] [EUCADEBUG] calling
http://10.0.0.10:8775/axis2/services/EucalyptusNC
4-[19:44:59] [EUCADEBUG ]DescribeNetworks(): done
5-[19:47:00 2009] [EUCAWARN] failed to terminate 'i-31540708':
instance may not exist any longer

```

Figura 4.3: *Logs* do controlador do *cluster* quando o *nodo2* foi removido da nuvem

sagens como o controlador do *cluster*. Assim, o nodo após um determinado tempo, encerrou as máquinas virtuais que estavam executando no *hipervisor* controlado por ele. As linhas 2 e 3 do arquivo de *log* do controlador do nodo, mostrado na figura 4.4, descrevem o momento em que o controlador do *nodo2* encerra as máquinas virtuais que ainda estavam em execução.

```

1-[19:45:22] [EUCADEBUG] doDescribeResource() invoked
2-[19:47:27] [EUCAINFO] doTerminateInstance() invoked (id=i-31540708)
3-[19:47:29] [EUCAINFO] destroyed domain for instance i-31540708

```

Figura 4.4: *Logs* do *nodo2* quando esse foi removido da nuvem

Após, o arquivo de configuração do *Eucalyptus* foi alterado e o *nodo2* foi inserido novamente. A nuvem automaticamente reconheceu que o *nodo2* estava ativo e passou a consultá-lo de forma a alocar novas instâncias. Foi solicitada então uma nova instância de máquina virtual, de forma que o nodo que foi reativado na nuvem, recebesse pelo menos uma dessas instâncias. Na Figura 4.5, nas linha 1 e 2 do *log*, o controlador do *cluster* utiliza o algoritmo de *ROUND ROBIN* para definir o nodo da nuvem que possui mais recursos para receber a instância, ou seja, o *nodo2*. Ainda na Figura 4.5, na linha 3 do *log*, é apresentado os recursos de hardware que a instância solicitada vai consumir.

```

1-[19:49:30] [EUCAINFO] scheduler using ROUNDROBIN policy
2-[19:49:30] [EUCAINFO] scheduler decided to run instance 'i-5260091E'
on resource 'http://10.0.0.20:8775/axis2/services/EucalyptusNC'
3-[19:49:30] [EUCAINFO] asking for virtual hardware
(mem/disk/cores): 128/2/1

```

Figura 4.5: *Logs* do controlador do *cluster*, onde o *nodo2* volta a receber instâncias

Um outro teste realizado dentro do cenário apresentado, foi interromper a comunicação do *nodo1* da nuvem com o controlador do *cluster*, simulando uma possível falha no nodo. O controlador do *cluster* acusou erros de comunicação com o *nodo1*, e logo após o restabelecimento da comunicação começou novamente a operar com o

nodo1. Na Figura 4.6, a linha 1 do *log* apresenta o momento em que a comunicação do *cluster* com o *nodo1* foi interrompida, enquanto a linha 2 mostra o momento em que a comunicação com o *cluster* foi restabelecida.

```
1-[19:56:51] [EUCAERROR] bad return from ncDescribeResource(10.0.0.10)
2-[19:59:36] [EUCAINFO] node=10.0.0.10 mem=794/666 disk=4385/3871
cores=3/2
```

Figura 4.6: *Logs* do controlador do *cluster* quando a comunicação foi interrompida e logo após reestabelecida

Observa-se, que a partir dos testes realizados, as tarefas em execução no *nodo2* simplesmente foram encerradas e em nenhum momento foram reexecutadas em um outro *nodo*. Isso porque, o *Eucalyptus* não realiza a recuperação das instâncias que estavam em execução nos *nodos*. Nos dois testes que foram realizados, as instâncias que estavam em execução nos *nodos* que foram desabilitados simplesmente foram encerradas.

O *Eucalyptus* poderia, em uma implementação futura, utilizar-se da propriedade de migração de máquinas virtuais que o *hipervisor Xen* provê e, possibilitar a migração das máquinas virtuais antes da retirada de um *nodo* da nuvem, evitando assim a perda das instâncias que estavam em execução no *nodo* no momento da remoção do mesmo da nuvem.

4.4.2 Realização de cópias de segurança

Segundo (EUCALYPTUS, 2009), os dados da nuvem computacional podem ser salvos através de rotinas de *backup* definidas pelo administrador da nuvem. Para a nuvem desenvolvida nesse trabalho, os dados a serem salvos, de forma a possuir uma cópia de segurança consistente, que permitisse retornar a nuvem a um estado anterior são o seguintes:

- Arquivos de configuração;
- Arquivos de banco de dados;
- Arquivos de chaves criptografadas;
- *Buckets* do *Walrus*;
- Arquivos de volume.

O *Eucalyptus* não possui mecanismos que facilitem a realização de *backups*. As cópias foram realizadas manualmente e podem ser automatizadas por *scripts* desenvolvidos pelo administrador da nuvem. Entretanto, o *backup* torna-se mais fácil, uma vez que todos os dados estão centralizados no controlador da nuvem.

Para efetuar o *backup* dos dados do *Eucalyptus*, de forma a manter a consistência nos dados, todos os serviços do *Eucalyptus* foram finalizados. Após, foram copiados os arquivos necessários, garantindo que todas as permissões também fossem copiadas juntamente com os arquivos. Após realizar a cópia dos arquivos, obteve-se como resultado um arquivo compactado no formato *.tar*, contendo uma cópia de segurança dos arquivos da nuvem computacional. Os comandos utilizados para a realização do teste desse cenário constam nos Anexos desse trabalho.

4.4.3 Capacidade de Armazenamento

Um dos principais objetivos desse trabalho, era a possibilidade de aumentar a capacidade de armazenamento da nuvem sob demanda, entretanto, o *Eucalyptus* por si só não possibilitou que esse objetivo fosse alcançado.

Conforme *Neil Soman* (através de comunicação por email), um dos desenvolvedores da ferramenta *Eucalyptus*, é possível utilizar um sistema de arquivos distribuído, como por exemplo o NFS, de forma a configurar a máquina em que o serviço *Walrus* está executando, e gravar os dados dos *buckets* nesse sistema de arquivos. Assim, o aumento da capacidade de armazenamento é efetuado a partir da adição de novos servidores ao sistema de arquivos distribuído. Deste modo, o aumento da capacidade de armazenamento é garantido pelo sistema de arquivos distribuído e não pelo serviço *Walrus*.

Outra solução, consiste em utilizar um sistema de *storage* para o armazenamento dos arquivos, entretanto, da mesma forma que o sistema de arquivos distribuído, o aumento da capacidade de armazenamento é garantido pelo *storage* e não pelo *Walrus*.

4.4.4 Padronização de API

Um dos objetivos desse trabalho é manter a possibilidade de integração com a nuvem da *Amazon*. A nuvem desenvolvida deve prover interface de comunicação semelhante à utilizada pela *Amazon* em seus serviços.

Para esse teste foi utilizada uma ferramenta de acesso ao controlador de armazenamento *Walrus* chamada *s3cmd*, e que possui compatibilidade de acesso a nuvem da *Amazon*. O *s3cmd* é uma ferramenta de código aberto, desenvolvida na linguagem de programação Python, que utiliza da API da *Amazon* para acesso aos dados. Essa ferramenta realiza operações básicas no servidor de armazenamento, como por exemplo, a criação de *buckets*, armazenamento e recuperação de arquivos, e definição de permissões. Para a utilização da ferramenta é necessário criar um arquivo de configuração. A configuração utilizada nos testes com a ferramenta encontra-se disponível na seção de Anexos. Para a ferramenta *s3cmd* comunicar-se com o *Walrus*, foi preciso acrescentar a localização do servidor do *Walrus* e do

bucket raiz.

Na Figura 4.7, tem-se os *logs* gerados pela utilização da ferramenta. Através dessa, foi criado um *bucket* chamado “teste”. Na linha 1 do arquivo de log, é apresentada a mensagem HTTP enviada para o *Walrus*. Já, na linha 2, é apresentada a mensagem retornada pela execução do *Response* da linha 1, informando que o *bucket* foi criado.

```
1- DEBUG: Response: {'status': 200, 'headers': {'content-length': '222',
'content-type': 'application/xml'}, 'reason': 'OK', 'data':
'<CreateBucketResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<CreateBucketResponse>
<BucketName>teste</BucketName>
</CreateBucketResponse> </CreateBucketResponse>'}
2- Bucket 'teste' created
```

Figura 4.7: *Logs* gerados pela utilização do comando *s3cmd* para a criação de um *bucket*

O *s3cmd* funcionou perfeitamente e realizou todas as operações com arquivos utilizadas nos testes, como por exemplo, a criação de *buckets*, checagem de consistência, entre outras.

4.4.5 Confidenciabilidade dos dados

Um dos objetivos desse trabalho, é a possibilidade de garantir a segurança dos dados na nuvem. Uma das formas de minimizar esse problema consiste em criptografar os dados antes de disponibilizá-los na nuvem ou definir permissões diferenciadas para os arquivos. Utilizando o *Walrus*, é possível, da mesma forma que o S3 da *Amazon*, garantir algumas permissões nos arquivos enviados a nuvem.

Os arquivos inseridos na nuvem através da ferramenta *s3cmd*, por padrão (definido no arquivo de configuração) são inseridos de forma privada, ou seja, apenas um administrador da nuvem e o usuário que disponibilizou o arquivo na nuvem podem ter acesso a esse arquivo. Porém, se o usuário desejar disponibilizar um arquivo para acesso público, basta configurar as permissões do mesmo para pública. Para testar as permissões dos arquivos na nuvem, foram enviados 2 arquivos para serem armazenados na nuvem. O primeiro arquivo foi enviado ao *bucket* de forma pública, enquanto o segundo arquivo foi enviado para o *bucket* de forma privada.

Os resultados das tentativas de acesso aos arquivos são apresentados abaixo. Na Figura 4.8, é apresentado o resultado da tentativa de acesso do arquivo armazenado com permissões públicas, através de um navegador. Já, na Figura 4.9 é apresentado o resultado do acesso ao arquivo armazenado de forma privada. Assim, o *Eucalypt-*

tus possui mecanismos que facilitam a utilização de permissões nos arquivos armazenados, semelhante ao serviço de armazenamento da *Amazon*.

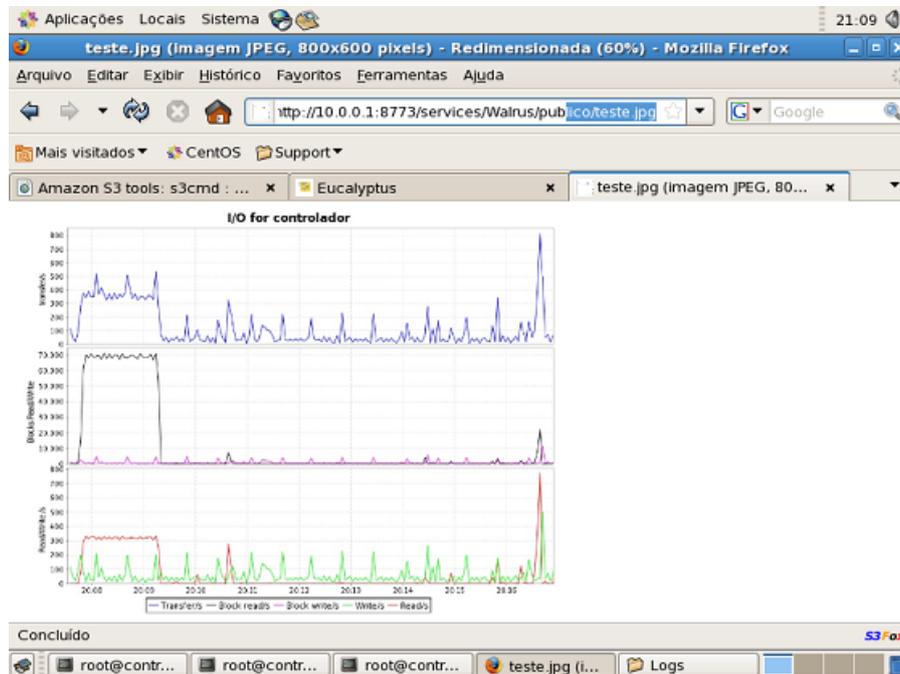


Figura 4.8: Acesso a um arquivo público

4.4.6 Cache de Imagens

O *Eucalyptus* realiza *cache* dos arquivos de máquinas virtuais em cada nodo. Quando o controlador de *cluster* envia a requisição para um nodo instanciar uma máquina virtual, ele verifica se a máquina virtual já está no *cache* do nodo. O *cache* das máquinas virtuais nos discos rígidos locais dos nodos, reduzem o tempo para uma máquina virtual ser executada, pois não é necessária realizar a cópia de toda a imagem da máquina virtual através da rede.

Na Figura 4.10, é apresentado os *logs* de um dos nodos quando é solicitada uma instância de máquina virtual e a imagem da máquina virtual solicitada não está em cache. Nas linhas 1 e 2 dos *logs*, o nodo realiza uma requisição para o *Walrus* solicitando a imagem de máquina virtual e copiando para o seu disco local.

Para esse cenário, foi requisitada uma instância de máquina virtual que não estava no cache de nenhum dos nodos e, foram analisados o processamento da máquina controladora do *cluster* no período e a quantidade de pacotes que trafegaram pela rede. Após, foi solicitado uma nova instância dessa mesma máquina virtual, sendo que nesse caso ela já estava em *cache*. A transferência total do arquivo de uma máquina virtual da distribuição Linux *Ubuntu*, com tamanho total de 1 GB, para um nodo, foi de aproximadamente 2 minutos, considerando que não havia tráfego de rede na nuvem. Se essa imagem estivesse no cache local do nodo, a inicialização

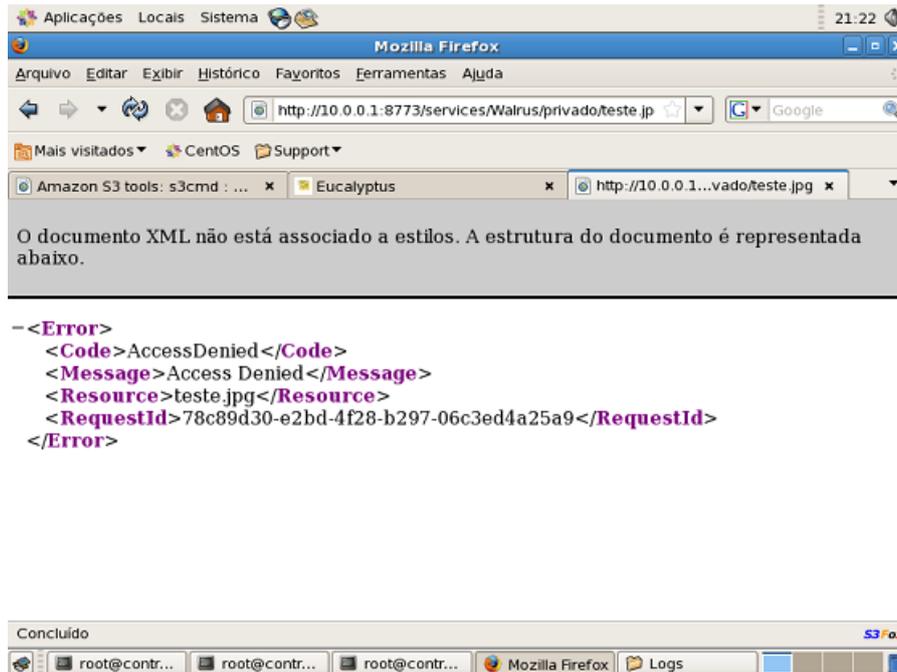


Figura 4.9: Acesso a um arquivo privado

```
1-[20:09:16] [EUCAINFO] walrus_request(): saved image in \
/usr/local/eucalyptus//admin/i-415606F2/root
2-[20:09:16] [EUCAINFO] vrun(): \
[cp -a /usr/local/eucalyptus//admin/i-415606F2/root \
/usr/local/eucalyptus//eucalyptus/cache/emi-5B6112C2/root]
```

Figura 4.10: *Logs* do nodo da nuvem descrevendo o *cache* de imagens

seria quase instantânea.

Através desse cenário, podemos concluir que sem a utilização do cache de imagens para os nodos da nuvem, a performance da mesma estaria bem abaixo do esperado, uma vez que cada instância, ao ser solicitada, seria transferida por completo, implicando em um tempo maior de transferência e um maior tráfego na rede.

4.4.7 Transferência de arquivos entre cliente e *Walrus*

Uma das ideias do armazenamento de arquivos em uma nuvem computacional é melhorar o atendimento a requisições. Para testar esse objetivo, foi realizado testes de transferência de arquivos para medir a performance apresentada pelo *Walrus*.

Para o monitoramento, utilizou-se o pacote do Linux chamado *sysstats*. Os índices monitorados através do *sysstats* na máquina em que o *Walrus* está executando, foram:

- Utilização da rede: monitoramento da quantidade de pacotes recebidos e transmitidos;

- Utilização de CPU: monitoramento da porcentagem de *cpu* utilizada e a porcentagem de *iowait*, que é a porcentagem de tempo que a *CPU* está *idle*, enquanto o sistema executa operações de entrada e saída;
- Utilização de memória: monitoramento da porcentagem de memória utilizada durante a cópia dos arquivos para o servidor.

Dentro desse cenário, foi realizado *upload* de arquivos de 900 MB através da rede local. No primeiro teste realizado, foram enviados 3 arquivos de 900 MB simultaneamente para um *bucket* e, a transferência demorou aproximadamente 4 minutos. Já, no segundo teste realizado foram enviados 8 arquivos de 900 MB para 2 *buckets* (em cada *bucket* é possível armazenar apenas 5 GB), e o tempo de transferência foi de aproximadamente 11 minutos. As taxas de transferências, em ambos os testes foram:

- Transferência de 1 arquivo: a taxa de transferência média foi de 4.89 MB/s o tempo de transferência foi de 3.2 minutos;
- Transferência de 3 arquivos simultâneos: a taxa de transferência média foi de 3.71 MB/s e o tempo médio de transferência foi de 4 minutos;
- Transferência de 8 arquivos simultâneos: a taxa de transferência média foi de 1.5 MB/s e o tempo médio de transferência foi de 10 minutos e 30 segundos.

Os resultados do monitoramento mostraram que a utilização de CPU e de memória foram praticamente iguais em todos os testes. O *iowait* aumentou enquanto os arquivos estavam sendo transferidos, mas permaneceu em valores aceitáveis, abaixo de 25%, sendo o tráfego de rede o índice monitorado em que pôde se observar maior aumento em comparação com o estado do controlador, anterior aos testes de transferência. Todas as requisições enviadas foram atendidas pelo *Walrus*, que centralizou o processamento. Enquanto os arquivos estavam sendo transferidos, a máquina onde o *Walrus* estava executando apresentou um alto tráfego na rede, o que vem a ser um gargalo na infra-estrutura. Além disso, a máquina apresentou altos índices de entrada e saída de *paginação*, além de serem registradas também uma alta atividade de operações de entrada e saída (*input/output*). Através dos testes realizados, verificou-se que o atendimento a requisições e toda a transferência de arquivos é realizada a partir do computador que executa o *Walrus*, desta forma, não tendo vantagens em relação a um sistema de armazenamento centralizado. Afim de reduzir esse problema, pode-se aumentar a capacidade de processamento, velocidade dos discos, etc. Porém, esse computador continuará sendo um gargalo na infra-estrutura de armazenamento de arquivos desenvolvida com a ferramenta *Eucalyptus*.

Frequentemente, os acessos ao *Walrus* serão realizados através de conexões mais lentas, onde a velocidade de transferência irá refletir na porcentagem de CPU utilizada. Hoje, o uso do *Walrus* pela ferramenta *Eucalyptus* concentra-se no armazenamento e disponibilização das imagens das máquinas virtuais para a nuvem computacional, porém, espera-se que assim como é possível criar um *cluster* de nodos na nuvem, seja possível também criar um *cluster* envolvendo o sistema de arquivos da nuvem.

4.5 Considerações Finais

Nesse capítulo foi descrito em detalhes os objetivos da infra-estrutura desenvolvida, bem como os principais componentes que formam essa infra-estrutura. Após, foi apresentada a estrutura física da nuvem computacional, o número de máquinas, a configuração nos computadores utilizados e a forma de conexão entre eles.

Além da infra-estrutura física, nesse capítulo foram apresentados todos os *softwares* utilizados no desenvolvimento da nuvem computacional, suas funções e configurações específicas. Por fim, foram apresentados os testes para consolidar a infra-estrutura desenvolvida. Esses testes tiveram como principais objetivos verificar se os objetivos propostos foram atingidos e verificar as funcionalidades do software de gerenciamento utilizado.

5 CONCLUSÃO

Nesse trabalho estudou-se a computação nas nuvens, e os conceitos da mesma foram utilizados para o desenvolvimento de uma infra-estrutura de nuvem computacional para o armazenamento de arquivos. Após o desenvolvimento desse trabalho pode-se realizar testes, analisando a performance da nuvem desenvolvida. A computação nas nuvens é uma tecnologia em constante crescimento e mostrou-se uma alternativa aos *clusters* e *grids* computacionais, devido ao seu baixo custo financeiro se comparada à capacidade de processamento e a elasticidade que é possível obter através da sua utilização.

Devido ao fato da computação nas nuvens ser uma solução recente no mercado, o acesso a bibliografias para auxiliar essa pesquisa foi um obstáculo. A pesquisa sobre a tecnologia resumiu-se a artigos acadêmicos e poucos livros que foram publicados nos últimos meses. Os serviços de computação nas nuvens existentes também foram base para esse trabalho, assim como, manuais e guias de ferramentas estudadas para o desenvolvimento da infra-estrutura proposta.

A ferramenta utilizada para o desenvolvimento da nuvem computacional, o *Eucalyptus*, é uma ferramenta em constante desenvolvimento que vem, frequentemente, sendo utilizada em ambientes acadêmicos e até comerciais. A ferramenta, porém, não possui ainda algumas funcionalidades e facilidades que são essenciais para a utilização da mesma em ambientes de produção, como por exemplo, a *clusterização* do serviço de armazenamento, migração das máquinas virtuais que executam na nuvem, suporte a tolerância a falhas, entre outras.

O armazenamento de arquivos na nuvem computacional deve ser feito de forma distribuída e escalável, de forma a atender um grande número de requisições. Através da utilização da ferramenta *Eucalyptus* e do seu componente de armazenamento *Walrus*, foi possível prover grande parte das funcionalidades que o serviço da *Amazon* provê, porém, com algumas limitações. O *Walrus* por si só não permite que os arquivos sejam, armazenados de forma distribuída e escalável, fazendo uso de terceiros para realizarem essa operação, como sistemas de armazenamento e arquivos distribuídos.

Assim, alguns dos objetivos propostos nesse trabalho não puderam ser alcançados, uma vez que a ferramenta utilizada nesse trabalho atualmente não possui suporte a elas. Por exemplo, um ambiente de armazenamento de arquivos na nuvem, deve prover o armazenamento de arquivos de forma distribuída e escalável, como uma espécie *cluster* de armazenamento. Além disso o atendimento de requisições não deve concentrar-se apenas na máquina em que o *web service* que controla o armazenamento está sendo executado.

A utilização do armazenamento de arquivos na nuvem computacional ainda encontra muita resistência dos usuários, pois os arquivos estarão em algum lugar na internet e não mais sob o seu total controle. Os dados estarão dispostos para acesso de qualquer lugar, porém o usuário ainda questiona a segurança dos dados dispostos na nuvem, mesmo que, os dados dispostos na nuvem estejam armazenados em uma infra-estrutura muito mais complexa e segura que o local onde o usuário hoje armazena esses dados.

5.1 Trabalhos Futuros

Esse trabalho, além de dar início a estudos sobre computação nas nuvens e o desenvolvimento de um ambiente de armazenamento de arquivos baseados nessa tecnologia, cria diversas opções para trabalhos futuros.

Várias ferramentas de gerenciamento de uma nuvem computacional estão disponíveis, porém esse trabalho escolheu uma delas para utilizar no desenvolvimento da infra-estrutura de nuvem computacional. Ferramentas como o *OpenNebula* (OPEN-NEBULA, 2009), *Enomalism* (ENOMALISM, 2009), entre outras podem ser utilizadas por trabalhos futuros visando analisar os resultados obtidos através da utilização dessas ferramentas.

Um outro trabalho que poderia ser realizado futuramente, é a utilização de ferramentas, como por exemplo, o *Hadoop* (WHITE, 2009) para o processamento de dados complexos na nuvem. O *Hadoop* é um *software* de código aberto que provê entre outras ferramentas, um sistema de arquivos distribuídos que pode ser utilizado em uma nuvem computacional, além de prover um *framework* de *MapReduce*, que possibilita a utilização de computação distribuída para o processamento de grande quantidade de dados na nuvem.

REFERÊNCIAS

AMAZON. **Amazon Web Services (aws)**. Disponível em: <http://aws.amazon.com/> Acesso em: setembro 2009.

AMAZON. **Amazon Elastic Compute Cloud**. Disponível em: <http://aws.amazon.com/ec2/> Acesso em: setembro 2009.

AMAZON. **Amazon Simple Storage Service**. Disponível em: <http://aws.amazon.com/s3/> Acesso em: setembro 2009.

ARMBRUST Michael et. al. Above the Clouds: A Berkeley View of Cloud Computing. **Electrical Engineering and Computer Sciences University of California at Berkeley**, v.0, february 2009.

BUYYA Rajkumar et.al. Cloud Computing: and Emerging IT Plataforms: Vision, Hype and Reality for Delivering Computing as the 5th Utility. **Future Generation Computer Systems**, p.599–616, 2008.

CHAPPELL David. **Introducing The Azure Services Platform**. Disponível em: <http://www.microsoft.com/windowsazure/> Acesso em: setembro 2009.

CITRIX. **Citrix Cloud Center**. Disponível em: <http://www.citrix.com/> Acesso em: outubro 2009.

ENOMALISM. **Enomaly's Elastic Computing Platform**. Disponível em: <http://www.enomalism.com> Acesso em: outubro 2009.

EUCALYPTUS. **Eucalyptus Open-Source Cloud Computing Infrastructure - An Overview**. Disponível em: <http://www.eucalyptus.com/> Acesso em: setembro 2009.

GANTZ John et. al. The Diverse and Exploding Digital Universe. **International Data Group**, 2008. Disponível em: <http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf> Acesso em: julho 2009).

GOOGLE. **Google App Engine**. Disponível em: <http://appengine.google.com>
Acesso em: setembro 2009.

HAIZEA. **Haizea**: An Open Source VM-based Lease Manager. Disponível em:
<http://haizea.cs.uchicago.edu> Acesso em: outubro 2009.

HARWOOD Ted. **Inside Citrix Metaframe XP**. 4.ed. Indianápolis, USA: John
Wait, 2004.

LEAVITT Neal. Is Cloud Computing Really Ready for Prime Time? **Computer**,
v.42, n.1, p.15–20, january 2009.

MICROSOFT. **Microsoft Azure**. Disponível em:
<http://www.microsoft.com/azure/> Acesso em: setembro 2009.

MILLER Michael. **Cloud Computing**: Web-Based Application that Change the
Way You Work and Collaborate Online. 1.ed. Indianápolis, USA: QUE Publishing,
2009. 289p.

NIMBUS. **Nimbus Open Source Iaas Cloud Computing Software**. Disponível
em: <http://workspace.globus.org/> Acesso em: outubro 2009.

NIRVANIX. **Nirvanix Storage Delivery Network**. Disponível em:
<http://www.nirvanix.com/> Acesso em: setembro 2009.

OPENNEBULA. **Opennebula**: The Open Source Toolkit for Cloud Computing.
Disponível em: <http://www.opennebula.org/> Acesso em: outubro 2009.

RACKSPACE. **The Rackspace Cloud**. Disponível em
<http://www.rackspacecloud.com/> Acesso em: setembro 2009.

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

SALESFORCE. **SalesForce Force.com**. Disponível em:
<http://www.salesforce.com/cloudcomputing> Acesso em: setembro 2009.

SOTOMAYOR Borja et. al. Capacity Leasing in Cloud Systems using the
Opennebula Engine. **Workshop on Cloud Computing and its Applica-
tions (CCA08)**, 2008. Disponível em [http://www.cca08.org/papers/Paper20-
Sotomayor.pdf](http://www.cca08.org/papers/Paper20-Sotomayor.pdf) Acesso em: outubro 2009.

STANLEY Morgan. Technology Trends, 2008. Disponível em:
[http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-
universe.pdf](http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf) Acesso em: setembro 2009.

TAN Tingxi; KIDDLE Cameron. An Assessment of Eucalyptus Version 1.4. **Grid Research Centre**, CANADA, may 2009.

TANEMBAUM, A. **Distributed Operating Systems**. New Jersey, USA: Prentice-Hall, 1995.

VAQUERO Luis M et. al. A Break int the Clouds: Towards a Cloud Definition. **ACM SIGCOMM Computer Communication Review**, 2009.

VMWARE. **VMware**. Disponível em: <http://www.vmware.com/> Acesso em: outubro 2009).

VMWARE. **Vware vsphere**. Disponível em: <http://www.vmware.com/products/vsphere> Acesso em: outubro 2009.

WARDLEY Simon; GOYER Etienne; BARCETS Nick. **Ubuntu Enterprise Cloud Architecture**. www.canonical.com: Canonica, 2009. 18p. v.0.

WHITE Tom. **Hadoop: The Definitive Guide**. California, USA: O'reilly, 2009.

6 ANEXOS

6.1 Instalação do *Eucalyptus*

1. Instalação da versão 5.4 do Sistema Operacional CentOS, através do DVD de instalação, disponível em <http://www.centos.org>.
2. Fazer o *download* do pacote de instalação do *Eucalyptus*, o *eucalyptus-1.6.1-centos-i386.tar.gz*, disponível em <http://open.eucalyptus.com.br>. Após o *download*, descompactar o instalador da seguinte forma:

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

6.1.1 Instalação Específica nos Nodos

1. A Instalação do *software hipervisor Xen*, é realizada da seguinte forma:

```
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
```

Configurações do arquivo */etc/xen/xend-config.sxp*:

```
(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 '')
```

2. Instalação dos pacotes de dependência do *Eucalyptus*, que são: *aoetools*, um pacote de ferramentas para acessar dispositivos de armazenamento *SATA* através da rede; *euca-axis2c*, uma ferramenta para a construção de *web services*, e o *euca-rampartc* uma biblioteca do *Eucalyptus* para segurança de *web services*.

```
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 ..
```

3. Instalar o serviço de controlador de nodo:

```
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
```

4. Inicializar o serviço do controlador de nodo:

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

6.1.2 Instalação Específica no Controlador da Nuvem

1. Instalação dos *softwares* necessários para o funcionamento do controlador da nuvem, como por exemplo, o Java e o servidor DHCP:

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

2. Instalação dos pacotes de dependência do controlador, que são: assim como no nodo os pacotes *aoetools*, *euca-axis2c* e *euca-rampartc*. Além desses, devem ser instalados o *vblade*, que é uma ferramenta que permite que exportar capacidade de armazenamento para hospedeiros através da rede local; o *groovy*, que é uma linguagem de programação; o *vtun*, uma ferramenta para um tunelamento nas conexões que utilizam o protocolo *TCP-IP*; e o *lzo* uma biblioteca para compactação.

```

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 ..

```

3. Instalação dos pacotes do controlador de *cluster* e controlador da nuvem, que são:

```

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

```

4. Inicializar os serviços de controlador da nuvem e controlador de *cluster* na máquina controladora:

```

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

```

6.1.3 Registrar os Componentes da Nuvem ao Controlador

1. Registrar o *Walrus*

```
euca_conf --register-walrus 10.0.0.1
```

2. Registrar o *cluster*

```
euca_conf --register-cluster cluster 10.0.0.1
```

3. Registrar o *Storage Controller*:

```
euca_conf --register-sc cluster 10.0.0.1
```

4. Registrar os Nodos ao *Cluster*:

```
euca_conf --register-nodes '10.0.0.10 10.0.0.20'
```

6.1.4 Alterar o arquivo de configurações do *Eucalyptus* nos nodos

Alterar o arquivo de configurações `/etc/eucalyptus/eucalyptus.conf`, alterando a configuração `MAX_CORES` para 3, dessa forma permitindo instanciar até 3 máquinas de 1 *core* em cada nodo. Se essa configuração estiver inativa, apenas será possível instanciar a quantidade de instâncias *small* de acordo com o número de *cores* que as máquinas nodo possuem.

6.2 Configuração do Serviço de *NTP*

1. Instalação do serviço: para instalar o serviço de *NTP*, em cada computador deve-se executar:

```
yum install -y ntp
```

2. Configuração do servidor de *NTP*: as configurações adicionais feitas no Arquivo `/etc/ntp.conf` do servidor de *NTP* (controlador da nuvem) são as seguintes:

3. 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
```

4. Configuração do cliente de *NTP*: as configurações adicionais feitas no Arquivo `/etc/ntp.conf` dos clientes *NTP* (nodos) são as seguintes:

```
server 10.0.0.1
restrict 10.0.0.1 mask 255.255.255.255 nomodify notrap noquery
```

6.3 Configuração do Servidor de *DHCP*

Configuração do servidor de *DHCP* que executa no controlador da nuvem. As configurações foram realizadas no arquivo `/etc/dhcp.conf`

```
# 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;
# HORÁRIO (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;

```

6.4 Configurações Adicionais no CentoOS

Alterações realizadas no *CentOS* nas máquinas nodo, de forma que o *Xen* conseguisse comunicar-se através da rede:

1. Parâmetros de *kernel*: alterada a linha do *boot* do *grub.conf* no arquivo */etc/boot/grub/menu.lst*, adicionando o parâmetro *noacpi* a linha do arquivo do *kernel*.
2. Configuração do *modprobe*: para não carregar o modulo *usb* da máquina, foi alterado o arquivo */etc/modprobe.d/blacklists* e adicionada a linha *ehci-hcd* ao final do arquivo.

6.5 Arquivo de Configuração do s3cmd

```

#chave de acesso
access_key = WKy3rMzOWPouV0xK1p3Ar1C2uRBwa2FBXnCw
acl_public = False
bucket_location = US
debug_syncmatch = False
default_mime_type = binary/octet-stream
delete_removed = False
dry_run = False
encrypt = False
force = False
gpg_command = /usr/bin/gpg
gpg_decrypt = %(gpg_command)s -d --verbose --no-use-agent --batch --yes
--passphrase-fd %(passphrase_fd)s -o %(output_file)s %(input_file)s
gpg_encrypt = %(gpg_command)s -c --verbose --no-use-agent --batch --yes

```

```

--passphrase-fd %(passphrase_fd)s -o %(output_file)s %(input_file)s
gpg_passphrase =
guess_mime_type = False
#localização do walrus
host_base = 10.0.0.1:8773
host_bucket = 10.0.0.1:8773
service_path = /services/Walrus
human_readable_sizes = False
preserve_attrs = True
proxy_host =
proxy_port = 0
recv_chunk = 4096
#secret key para acesso a nuvem
secret_key = x2dxrGhh098PXCupj9L61fE8CoKWKYX0xp20A
send_chunk = 4096
use_https = False
verbosity = WARNING

```

6.6 Utilização do *euca2ools*

6.6.1 Instalação e obtenção das credenciais

Para efetuar o download das credenciais obtidas através do acesso a *interface web* do controlador da nuvem e utilizar o *euca2ools*. Para a instalação do *euca2ools* os seguintes procedimentos devem ser seguidos:

1. Instalar o programa *swig*, através do comando:

```
yum install -y swig
```

2. Efetuar o *download* do arquivo *euca2ools-1.1-centos-i386.tar.gz* no site do *Eucalyptus*.
3. Descompactar os arquivos da ferramenta através do comando:

```
tar zxvf euca2ools-1.1-centos-i386.tar.gz
```

4. Entrar no diretório onde os arquivos foram descompactados e instalar a ferramenta através do comando:

```
rpm -Uvh phyton25-2.5.1-bashton1.i386.rpm \
phyton25-libs-2.5.1-bashton1.i386.rpm euca2ools-1.1-i386
```

5. Acessar a *interface* do controlador da nuvem computacional através dos endereços `https://10.0.0.1:8443` ou `https://200.233.17.193:8443`, e efetuar o *download* das credenciais, descompactar e executar o arquivo *eucarc* através do seguinte comando:

```
source eucarc
```

6.7 *Backup* dos arquivos da nuvem computacional

Para realizar a cópia dos dados da nuvem computacional, deve-se:

1. Finalizar as instâncias que estiverem rodando nos nodos, através do comando:

```
euca-terminate-instances <id da instância>
```

2. Finalizar todos os serviços que estão rodando nos componentes da nuvem computacional. Para garantir que todos os processos do *Eucalyptus* tenham sido finalizados, em todos os componentes da nuvem foi executado:

```
ps aux | grep euca
kill -9 <id do processo>
```

Após garantir que o *Eucalyptus* não estava mais executando, foram copiados os arquivos necessários através do seguinte comando, de forma a garantir que todas as permissões também fossem copiadas juntamente com os arquivos:

```
tar cvf /root/Backup/eucalyptus-backup.tar \
/etc/eucalyptus/eucalyptus.conf \
var/lib/eucalyptus/keys \
var/lib/eucalyptus/db \
/var/lib/eucalyptus/bukkits \
/var/lib/eucalyptus/volumes
```