

UNIVERSIDADE FEDERAL DE SANTA CATARINA

BRUNO ORTALE GONÇALVES

**GERÊNCIA E MONITORAMENTO DE UMA NUVEM PRIVADA**

Universidade Federal de Santa Catarina

Centro Tecnológico

Florianópolis

2011

BRUNO ORTALE GONÇALVES

GERÊNCIA E MONITORAMENTO DE UMA NUVEM PRIVADA

Produção Técnico-Científica apresentada como requisito parcial para a obtenção do título de Bacharel em Ciências da Computação, na Universidade Federal de Santa Catarina, Centro Tecnológico

Orientador: Prof. Carlos Becker Westphall

Universidade Federal de Santa Catarina

Florianópolis

2011

## AGRADECIMENTOS

- *Agradeço primeiramente aos meus pais (Neto e Lenita) e meus irmãos (Rodrigo e Tatiana) que sempre me deram apoio e sempre acreditaram em mim.*
- *Aos meus familiares que estão distantes, mas não sendo menos importantes.*
- *Agradeço meus amigos que estiveram ao meu lado ao decorrer da minha vida, e sempre contribuíram com bons momentos de descontração, especialmente ao Toco, Vitti, Daniel e Rafael que me ajudaram com o projeto.*
- *à Deus por me dar muitas oportunidades e bons momentos nesta vida.*

## **LISTA DE TABELAS**

Tabela 01: Tabela adaptada de comparação de infraestrutura virtual.....	41
---	----

## LISTA DE FIGURAS

Figura 01: Convergência de tecnologias.....	13
Figura 02: Comparação entre Notebooks e Netbooks.....	16
Figura 03: Porcentagem de usuários que visitam o site Amazon por mês em 2010.....	17
Figura 04: Arquitetura de emulação.....	20
Figura 05: Arquitetura de virtualização total.....	20
Figura 06: Arquitetura de para-virtualização.....	21
Figura 07: Arquitetura de virtualização em nível de sistema operacional.....	22
Figura 08: Arquitetura de virtualização em nível de sistema operacional aplicação...	23
Figura 09: Modelos de Serviço.....	25
Figura 10: Nuvem Híbrida.....	28
Figura 11: Gráfico problemas relacionado a computação em nuvem.....	32
Figura 12: Arquitetura adaptada de computação em nuvem.....	33
Figura 13: Arquitetura para soluções de alta demanda.....	34
Figura 14: Arquitetura para soluções de alta quantidade de processos.....	35
Figura 15: Arquitetura para soluções de alta demanda em um período irregular.....	35
Figura 16: Arquitetura Eucalyptus adaptada.....	39
Figura 17: Modos de configuração de rede do Eucalyptus .....	48
Figura 18: Configuração Eucalyptus .....	49
Figura 19: Tipos de MV e controlador de armazenamento.....	50
Figura 20: Layout do projeto .....	52
Figura 21: Status Nagios .....	54
Figura 22: Status de carga média.....	55
Figura 23: Modelo definição de métricas.....	56
Figura 24: Envio de requisição .....	57
Figura 25: MV rodando .....	58
Figura 26: Resultado Nagios .....	59
Figura 27: Resultado web .....	59

## **LISTA DE ABREVIATURA E SIGLAS**

SO – Sistema Operacional

MV – Máquina Virtual

QoS – Quality of Service (Qualidade do Serviço)

SLA – Service Level Agreement (Acordo de Nível de Serviço)

SaaS – Software as a Service (Software como Serviço)

PaaS – Platform as a Service (Plataforma como Serviço)

IaaS – Infrastructure as a Service (Infraestrutura como Serviço)

TI – Tecnologia da Informação

EC2 – Elastic Compute Cloud

## LISTA DE TRADUÇÕES

Cloud Computing – Computação em nuvem

Utility Computing – Computação utilitária

Public Cloud – Nuvem Pública

Private Cloud- Nuvem Privada

Hybrid Cloud – Nuvem Híbrida

Community Cloud – Nuvem Comunitária

Pay per use – Pagar por uso

Toolkit– Conjunto de Ferramentas

Software as a Service (SaaS) – Software como serviço

Platform as a Service(PaaS) – Plataforma como serviço

Infrasructure as a Service(IaaS) – Infraestrutura como serviço

## **RESUMO**

Com os avanços tecnológicos surge uma nova tendência de tecnologia que tem por objetivo dar suporte e proporcionar serviços à tecnologia da informação. Estamos falando da computação em nuvem, um paradigma que promete ser a questão chave para mudar a economia e o cotidiano das pessoas. Porém a falta de conhecimento e um relacionamento confiável de seus serviços torna este tema mais nebuloso e as vezes atrasando esta migração natural de tecnologia. Neste trabalho serão apresentadas as características e os conceitos gerais que envolvem a computação em nuvem. Também foi criado um sistema web que implementa as métricas de acordo de nível de serviço que proporciona um relacionamento confiável entre os usuários e os prestadores de serviço constituindo de um monitoramento de uma nuvem privada que também foi construída. Demonstrando a viabilidade de uma implantação como esta, tendo a satisfação do cliente aos serviços obtidos e por fim clareando as idéias sobre a computação em nuvem.

.

Palavras-chave: Computação em nuvem, Monitoramento, Satisfação dos clientes, Viabilidade.

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>10</b>
1.1 Justificativa .....	10
1.2 Objetivos .....	11
1.2.1 Objetivo Geral .....	11
1.2.2 Objetivos Específicos .....	11
<b>2 CLOUD COMPUTING .....</b>	<b>12</b>
2.1 Fundamentação Teórica .....	12
2.2 História .....	13
2.1 Definição .....	15
2.2 Descrição e análise da situação anual.....	15
2.3 Virtualização .....	19
2.4 Tipos de virtualização .....	20
2.5 Computação Utilitária.....	23
2.6 Camadas e Modelos .....	24
2.7 Acordo de Nível de Serviço (SLA) .....	26
2.8 Tipos .....	27
2.9 Prós e Contras .....	28
2.10 Segurança .....	30
2.11 Arquitetura .....	32
2.12 Padrões .....	36
<b>3 FERRAMENTAS E TECNOLOGIAS .....</b>	<b>38</b>
3.1 Eucalyptus .....	38
3.2 Amazon EC2 .....	40
3.3 OpenNebula .....	40
3.4 Nagios .....	42
3.5 Icinga .....	43
3.6 HybridFox .....	43

3.7 PCMONS .....	44
3.8 Euca2ools .....	44
<b>4 ESTUDO DE CASO .....</b>	<b>45</b>
4.1 Definição da proposta .....	45
4.2 Detalhamento do ambiente e das ferramentas do processo .....	45
4.3 Detalhamento do processo de instalação dos softwares e ferramentas .....	47
4.4 Desenvolvimento .....	51
4.5 Layout do projeto .....	52
4.6 Definição de métricas.....	53
4.7 Sistema Web .....	56
4.8 Testes e Resultados .....	57
4.9 Conclusão .....	60
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>62</b>
5.1 Trabalhos Futuros .....	62
<b>6 REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>63</b>

## 1 INTRODUÇÃO

A Internet está presente na vida de bilhões de pessoas e tem potencial para revolucionar ainda mais a vida contemporânea.

O grande potencial e a evolução da Internet é chamada computação em nuvem. O termo nuvem é uma metáfora à forma como a Internet é mostrada nos diagramas de rede, que de modo geral é uma combinação de tecnologias e paradigmas. Utilizar serviços dessa nuvem, como armazenamento, processamento de dados, entre outros, é fazer computação em nuvem.

Computação em nuvem é um assunto novo na área de tecnologia, muito comentado e estudado no momento, por isso apresenta muitas questões em aberto. Toda essa curiosidade em relação ao tema levou ao interesse em se conhecer qual a viabilidade de se construir um ambiente de computação em nuvem e monitorá-lo através de ferramentas de código aberto para entregar um serviço com uma completa infraestrutura computacional. Tipicamente na entrega são fornecidas máquinas virtuais para o cliente que as usa conforme suas necessidades fazendo a infraestrutura centralizada no *datacenter* do provedor ou espalhada na nuvem deixando o usuário sem detalhes sobre a origem do seu fornecimento.

Este trabalho define o conceito de Nuvem e quais as vantagens e desvantagens desse novo paradigma.

### 1.1 Justificativa

A computação em nuvem vem crescendo cada vez mais, segundo Merrill Lynch (Lynch 2008) a computação em nuvem será um mercado de 160 bilhões de dólares em 2011. Muitas pessoas usam recursos em nuvem, mas não sabem, como por exemplo, o gmail da Google, assim veio a motivação de embasar conceitualmente o que é este novo paradigma e o que ele tem a oferecer.

A migração para a computação em nuvem aparentemente é a escolha certa e parece que tudo está tendendo para isso, porém um dos grandes problemas para esta troca de paradigma é justamente na confiabilidade entre o suposto serviço oferecido e o que realmente é prestado. Porém é preciso ter o controle disto tudo, com isso veio à necessidade de primeiro definir métricas adequadas para um acordo justo e um verdadeiro controle do que está sendo fornecido, e também de ter um sistema onde possa ser mostrado o que realmente está acontecendo em termo das métricas estabelecidas para com isso obter a confiabilidade e contribuir com a futura migração para as nuvens.

## **1.2 Objetivos**

### 1.2.1 Objetivo Geral

O objetivo geral deste trabalho é apresentar conceitos e definições deste paradigma, criar um sistema web de relacionamento entre usuário e servidor com definições de métricas para um relacionamento confiável além de obter resultados da implantação prática de uma nuvem.

### 1.2.2 Objetivos específicos

Os objetivos específicos que podem ser citados são:

- Definir e conceituar as tecnologias e o ambiente que envolve a computação em nuvem;
- Identificar e analisar a situação geradora do projeto;
- Simular uma nuvem;
- Definir e descrever os procedimentos que envolvem as ações projetadas;
- Apresentar indicadores que demonstram a viabilidade deste sistema para o uso da computação em nuvem.

## 2 CLOUD COMPUTING

### 2.1 Fundamentação Teórica

Na última década, os engenheiros da computação andam com a cabeça bem acima das montanhas. Eles agora vivem nas nuvens. “Cloud computing”, ou “computação em nuvem”, é o futuro. Todos os seus documentos, e arquivos irão estar virtualmente pairando sobre nós. Você acessa seus dados de qualquer computador, em qualquer lugar. E mais do que isso os programas também ficam nas nuvens.

Essa possibilidade de acesso remoto e armazenamentos de arquivos na rede estavam virando uma necessidade, graças à crescente digitalização de dados do mundo real. A proposta era de habilitar o acesso de grandes quantias de poder computacional em uma maneira totalmente virtualizada, para com isso criar uma ilusão da disponibilidade de recursos infinitos, acessáveis sob demanda e eliminando a necessidade de adquirir e provisionar recursos antecipadamente. Outra importante visão desta tecnologia era usar essa computação como serviço, o que hoje em dia é chamado de computação utilitária.

Computação utilitária é o modelo de negócio onde é feito um acordo sobre o uso do fornecimento dos recursos computacionais de *datacenters*, permitindo que as empresas usem os recursos na quantidade que forem necessários, aumentando e diminuindo a capacidade computacional de forma dinâmica (VOORSLUYS et al. 2011).

Se tudo acontecer como imaginam os engenheiros da computação, num futuro próximo os computadores poderão ser muito mais baratos e usarão programas oferecidos quase sempre de graça, pela Internet.

"Eu diria que o computador do futuro é a Internet", afirma Eric Schmidt, atual presidente do Google. "Hoje, se você tem um problema no computador, está tudo perdido, é terrível. Mas,

com a computação nas nuvens, não importa se você usa o celular, o computador ou qualquer outro aparelho, tudo estará guardado na internet."

Como tudo seria guardado nas "nuvens" só precisaria ter um computador com acesso a Internet assim contribuindo com a inclusão digital da população mais pobre.

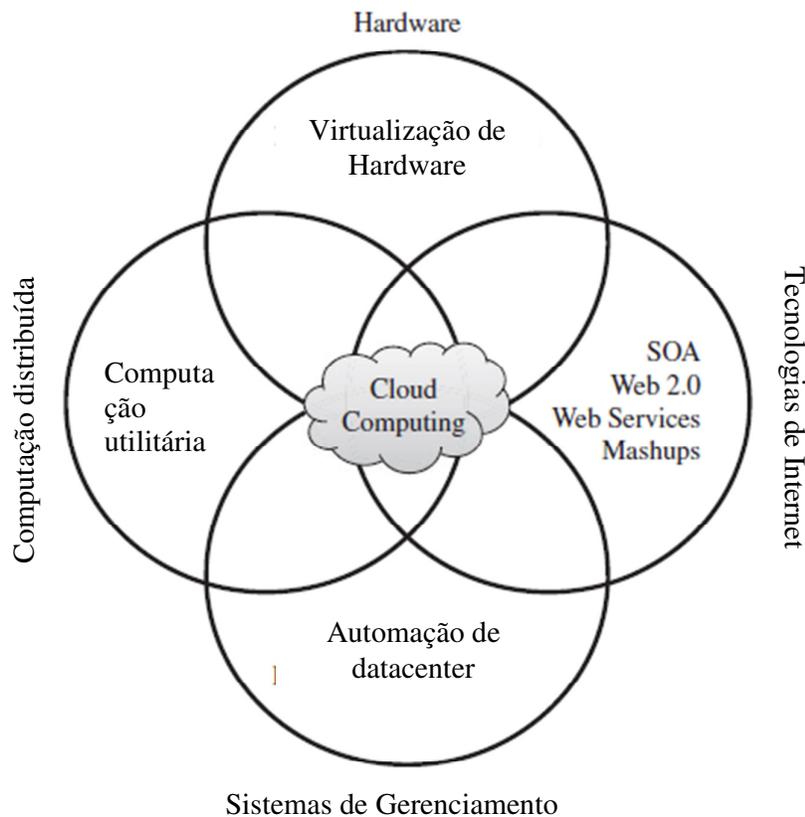
"Computação em nuvem é um negócio estimado em 160 bilhões de dólares" (LYNCH, 2008). Quando a população criar consciência sobre as atividades desenvolvidas, este setor poderá cada vez mais incentivar a demanda e com isso crescer em todos os sentidos não só computacionalmente.

## 2.2 História

O conceito por de trás da computação em nuvem vem de 1960 quando John McCarthy comentou sobre utilidade pública, mas só em 1997 que o termo "cloud computing" foi usado como uma definição acadêmica.

A raiz da computação em nuvem é um avanço e junção de várias tecnologias, na parte de hardware como virtualização e processadores, na parte de tecnologia sobre Internet temos *web services*, arquitetura orientada a serviço, *web 2.0*, na computação distribuída são clusters e grids, e no gerenciamento de sistemas temos automação de data centers, como mostra a figura 1 (VOORSLUYSS, 2011).

Figura 01– Convergência de tecnologias



Fonte : VOORSLUYSS, 2011

Com esses avanços em 2001 a Amazon modernizou seus data centers e implementou uma arquitetura em nuvem, e com resultados positivos começou a iniciar seu novo produto para providenciar computação em nuvem para os consumidores. Em 2005 foi criado o serviço Amazon EC2 (Elastic Compute Cloud), que usa a idéia de computação utilitária que cobrava por demanda.

Com isso após um ano a IBM, Google, AT&T, Microsoft e universidades pelo mundo começaram pesquisar e investir neste paradigma, surgindo várias ferramentas para criação de novos serviços.

No Brasil a realidade é que a computação em nuvem está em estágio inicial de adoção, mas os líderes de TI devem anteceder como ela poderá mudar a TI, as habilidades necessárias e as relações de negócio a fim de extrair todo o potencial oferecido pela nuvem. O Brasil já possui um sistema de computação em nuvem totalmente nacional, o PCiO, um sistema da MIPC informática, empresa brasileira fundada em 2005 (SOUZA FILHO, 2009). O principal

problema que a empresa encontra em relação a esse sistema é a resistência dos usuários, principalmente devido a questões relacionadas à segurança. Além disso, o serviço de Internet é muito inferior se comparado com os serviços de outros países, isso contribui também para que o sistema não seja muito difundido.

### **2.3 Definição**

Segundo (Taurion, 2009) o termo Cloud Computing ou computação em nuvem surgiu em 2006 em uma palestra de Eric Schmidt, do Google, sobre como sua empresa gerenciava seus datacenters. É assim chamado, pois reflete a idéia que a computação não está ocorrendo no seu computador e sim nos *datacenters*, não sabendo-se o local exato.

A idéia da nuvem é utilizar recursos ociosos de computadores independentes, sem preocupação com localização física e sem investimentos em hardware.

A computação em nuvem nada mais é que um termo para descrever um ambiente de computação baseado em uma imensa rede de servidores, sejam estes virtuais ou físicos. Uma definição simples seria a nuvem é um grande reservatório de recursos virtualizados que são acessíveis a todo o momento, sendo estes recursos como hardware, plataformas de desenvolvimentos e outros tipos de serviços. Esses recursos podem ser configurados e ajustados dinamicamente conforme desejado (VAQUERO et al. 2008).

Para os usuários utilizarem os serviços, necessitam apenas ter um computador com um sistema operacional com acesso a Internet, todos os recursos que iriam fazer os processamentos e armazenamentos de dados estarão disponíveis e sendo utilizados na “nuvem”.

### **2.4 Descrição e análise da situação atual**

A “nuvem”, a grande promessa de ser o fator principal para a mudança da economia e do nosso cotidiano. A chave do futuro é saber utilizar com sabedoria toda a informação relacionada a esse assunto. No presente, estamos aprendendo como processar e armazenar, para poder disponibilizar da melhor maneira possível, você nem precisar estar com o seu computador ligado, a nuvem faz tudo isso pra você por meio de datacenters espalhados pelo mundo porém conectados entre si.

Mesmo sem saber, você pode estar nas nuvens, vários aplicativos e programas na Internet recorrem a serviços de processamento e armazenamento nas nuvens, como por exemplo o Gmail, ou as planilhas do Google Docs. Tendo esta idéia de que tudo está ao redor das nuvens e de que ela está em todo lugar, e seus arquivos podem ser acessados de qualquer lugar, não está preso ao seu computador. Consequência disso é que começa a tendência de que o seu computador não precisa de todos os recursos para fazer a grande parte das suas tarefas, ou seja, aparece os novos modelos de computadores como os netbooks, que são mais baratos e compactos, mais viável a essa nova realidade que tem o crescimento dos netbooks em relação aos notebooks, isso já no começo de 2009 como mostra a Figura 2

Figura 02– Comparação entre Notebooks e Netbooks

Category	Q2'08	Q1'09	Q2'09
Mini-Note	5.6%	17.8%	22.2%
Notebook PC	94.4%	82.2%	77.8%

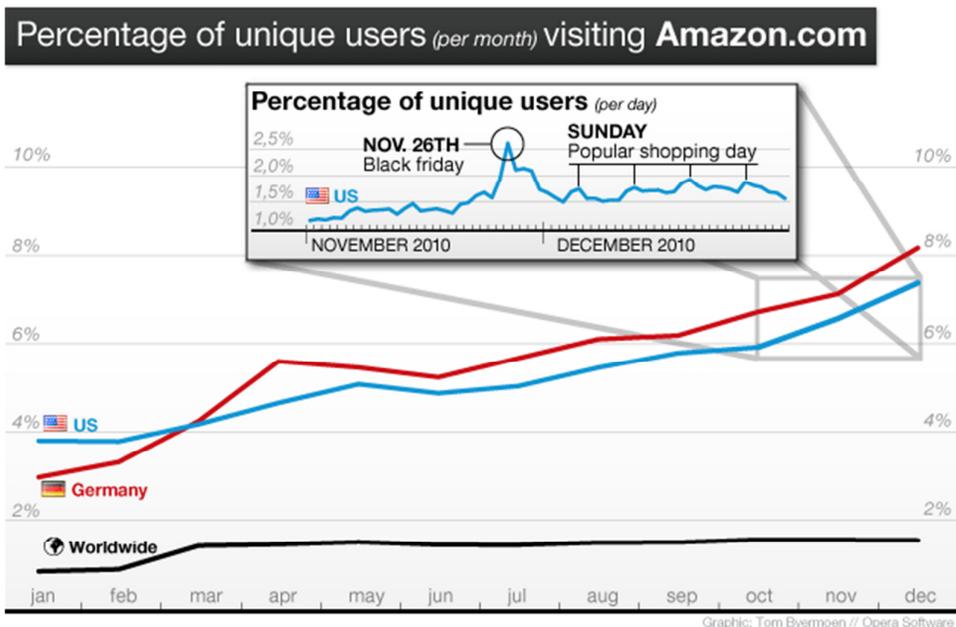
Fonte: DisplaySearch, 2009

Este conceito de trabalhar na grande nuvem fazendo com que os preços dos computadores sejam reduzidos gera também outras consequências como mais pessoas tendo acesso aos diversos produtos oferecidos pelas empresas, e assim firmando a frase de Ray Ozzie, substituto de Bill Gates no desenvolvimento de software da Microsoft que diz: “A era da nuvem já começou. Terá um impacto revolucionário na maneira como as pessoas lidam com a tecnologia e vai determinar a história da computação nos próximos cinquenta anos”.

A base da computação em nuvem são os *datacenters*, imensos aglomerados de computadores conectados a rede, alguns data centers são milhares de servidores enfileirados em longos corredores. Este conceito já é comum em algumas empresas como Google, Amazon, Microsoft, que mantêm vários parques computacionais com milhares de máquinas. Por exemplo, a Google que tem vários data centers espalhados pelo mundo, sua infraestrutura é chamada de Googleplex, compreende de mais de 200 petabytes de disco e cerca de 500 mil servidores. Antes de a Google ter esses imensos parques computacionais a Google e Amazon tinham algumas infraestruturas com data centers apenas para operar seus próprios negócios, pois servidores em salas frias já existem desde a década de 60, só que na época redistribuir as funções entre os grupos de máquinas era caro e trabalhoso. Por essa razão 80% da capacidade de processamento não era utilizada em certos períodos. A tecnologia que transformou esses parques de computadores em nuvem foi a virtualização.

Graças a este gerenciamento de poder de processamento e armazenamento sob demanda que a Amazon definiu o seu serviço de nuvem EC2(Elastic Compute Cloud) que “aumenta e diminui” conforme a necessidade. Um exemplo do como isso é necessário no dia a dia é um site de compras online na semana do Natal, o que acontece é que o acesso ao site aumenta significativamente seu número de acessos, e para suportar isso são precisos recursos computacionais como mostra a figura 3. Só que o problema todo é que se você comprar todo esse recurso para apenas suportar uma semana em um ano inteiro você estará jogando dinheiro fora, pois gastou muito dinheiro apenas para uma semana.

Figura 03– Porcentagem de usuários que visitam o site Amazon por mês em 2010



Fonte: Tetzchner, 2010

Por isso essa computação elástica encaixou tão bem com a realidade na qual estamos, com ela é possível no exemplo dado pagar por mais recurso apenas na semana do Natal e com isso as empresas não precisarão mais gastar fortunas para montar estruturas de tecnologia que tendem a ficarem desatualizadas rapidamente. Elas usarão a estrutura oferecida pelas grandes empresas que oferecem serviços de nuvem.

Isso que não estamos falando de um dos pontos cruciais, que é o caso de economia de energia. Está idéia é chamada de Green Cloud Computing, que é em todas as aplicações computacionais ter o mínimo de desperdício energético. Como sabemos para manter estes *datacenters* é necessário um ambiente asséptico, a temperatura tem que ser mantida a 21 graus, ou seja, quanto mais data centers e aumento de recursos, mais energia terá que ser gasta para mantê-las.

Green cloud computing é um assunto que já está sendo muito pesquisado e muito visado pois é uma questão que está atingindo um futuro bem próximo.

As principais características da computação em nuvem são os motivos por que ela será revolucionária para os negócios, que segundo (TAURION, 2009) consiste em:

- Eliminar a necessidade de adquirir recursos antecipadamente.
- Criar uma ilusão de disponibilidade de recursos infinitos, acessáveis sob demanda.
- Oferecer elasticidade, permitindo que as empresas usem os recursos na quantidade que forem necessários, aumentando e diminuindo a capacidade de processar de forma dinâmica.
- Oferecer serviços em nuvem e o pagamento ser feito pela quantidade de recursos utilizados

## **2.5 Virtualização**

Uma das principais tecnologias que permitiu a computação em nuvem é a virtualização, que é a capacidade de fornecer uma abstração dos recursos de computação. Hoje em dia as plataformas de infraestrutura são predominantemente de dois tipos, que são os ambientes totalmente virtualizados ou para-virtualizados (CHANTRY 2009).

Essa é uma tecnologia que não é exatamente recente, ela já é usada nos mainframes desde a década de 1970. A virtualização de servidores proporciona redução nos custos de aquisição dos servidores físicos, facilita a administração, diminui os recursos de infraestrutura necessários para hospedar os servidores e o consumo de energia.

Há outras variações para a virtualização , porém esses 2 tipos são os mais comuns e usados na computação em nuvem.

## 2.6 Tipos de Virtualização

- **Emulação** – Neste tipo de virtualização, o ambiente virtual emula uma arquitetura de hardware exigida de outro sistema operacional. Um dos casos comuns em que você encontra o hardware emulado é com dispositivos móveis. Os desenvolvedores de aplicativos usam um ambiente emulado para testar os aplicativos que são projetados para rodar em smartphones ou PDAs (CHANTRY 2009).

Figura 04- Arquitetura de emulação



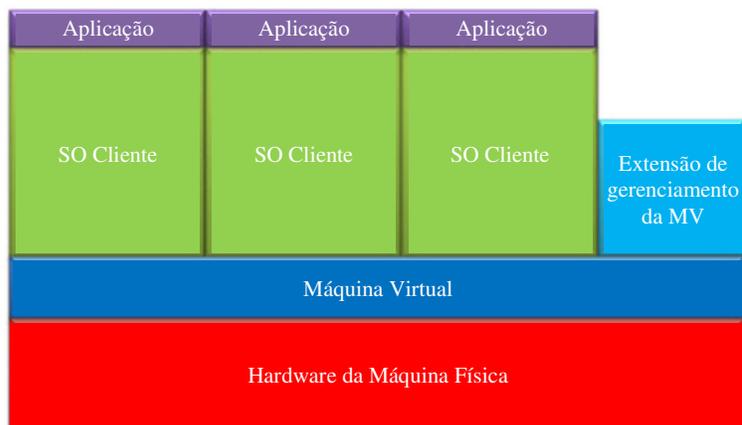
Fonte: CHANTRY 2009

**Prós:** Simula um ambiente de hardware, que é completamente diferente do hardware subjacente. Um exemplo disto seria um dispositivo móvel, como um smartphone emulado em um computador.

**Contras:** Péssimo desempenho e alto uso de recursos.

- **Virtualização Total** - Na virtualização completa, uma imagem completa de outro sistema operacional é feita e executado dentro de um ambiente virtualizado. A diferença entre a virtualização completa e emulação é que todos os convidados virtuais executam na mesma arquitetura de hardware. Todos os clientes suportam o mesmo hardware, o que permite que o cliente execute várias instruções diretamente no hardware, assim, proporcionando melhor desempenho (CHANTRY 2009).

Figura 05- Arquitetura de virtualização total



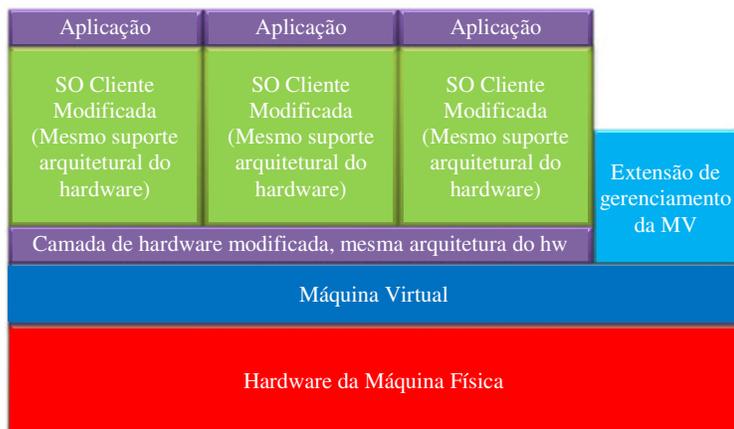
Fonte: CHANTRY 2009

**Prós:** A capacidade de executar várias versões do sistema operacional de múltiplos fornecedores como Microsoft, Linux, UNIX.

**Contras:** Imagens virtualizadas são instalações completas do sistema operacional e podem ser arquivos extremamente grandes. Com operações de entrada e saída em uma aplicações com grande intensidade, o desempenho pode ter quedas significativas.

- **Para-Virtualização** - Na paravirtualização, o hypervisor exporta uma cópia modificada do hardware. A camada de exportação tem a mesma arquitetura que o hardware do servidor. Assim tendo um ótimo desempenho pois se utiliza de drivers reais.(CHANTRY 2009)

Figura 06- Arquitetura de para-virtualização



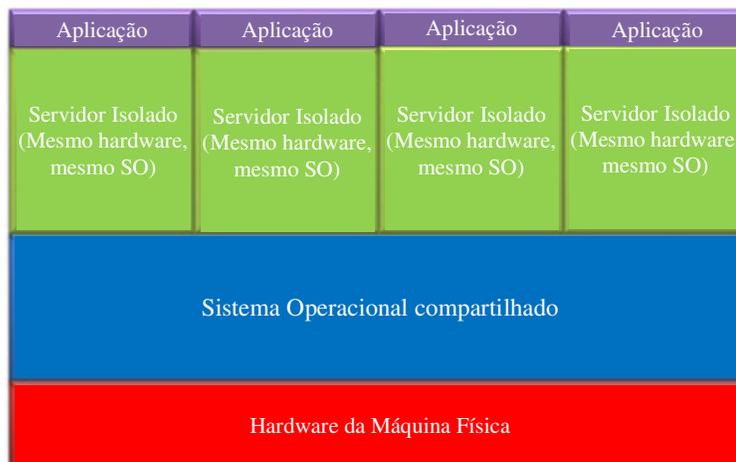
Fonte: CHANTRY 2009

**Prós:** Leve e rápido. Os tamanhos das imagens são significativamente menores, e o desempenho pode chegar a velocidades próximas da original. Permite a virtualização das arquiteturas que normalmente não suportam a virtualização completa.

**Contras:** Requer modificações para o sistema operacional cliente.

- **Virtualização em nível de sistema operacional** - Na virtualização do sistema operacional, não há nenhuma máquina virtual, a virtualização é feita completamente dentro de um único sistema operacional. Os sistemas de hóspedes compartilham características comuns e drivers do sistema operacional subjacente, porém são tratados como computadores completamente distintos. Cada instância hóspede terá o seu próprio sistema de arquivo, IP e configuração do servidor, e executam aplicações completamente diferentes. (CHANTRY 2009)

Figura 07- Arquitetura de virtualização em nível de sistema operacional



Fonte: CHANTRY 2009

**Prós:** Rápido, leve e eficiente, com a capacidade de suportar um grande número de instâncias virtuais.

**Contras:** Todas as instâncias virtuais devem suportar o mesmo sistema operacional.

- **Virtualização da aplicação** - A virtualização de aplicação, como qualquer outro tipo de virtualização, exige uma camada de virtualização. A aplicação é completamente abstraída da plataforma física e interage apenas com a camada de virtualização, isso permite que aplicativos que são incompatíveis uns com os outros executem lado a lado (CHANTRY 2009).

Figura 08- Arquitetura de virtualização em nível de sistema operacional aplicação



Fonte: CHANTRY 2009

**Prós:** Melhora a portabilidade de aplicações, permitindo que funcionem em diferentes ambientes operacionais. Permite que aplicativos incompatíveis executem lado a lado.

**Contras:** Nem todos os softwares podem ser virtualizados, por isso não é uma solução completa.

## 2.7 Computação Utilitária

Segundo (CHANTRY, 2009) a computação utilitária refere-se a utilização de recursos computacionais como, armazenamento, infraestrutura de serviços, processamento, da mesma

forma que você usaria eletricidade ou água, ou seja, como um serviço fornecido no qual você só paga pelo que usa. Este serviço pode eliminar a necessidade de comprar, executar e manter hardware, servidores e plataformas de aplicativos.

Em ambientes de computação utilitária, os usuários atribuem um valor de utilidade a sua respectiva tarefa, onde a utilidade é uma avaliação fixa ou variável no tempo que capta várias restrições de qualidade de serviço. O valor é a quantidade que o usuário está disposto a pagar a um provedor de serviços para satisfazer suas demandas. Os prestadores de serviço em seguida, tentam maximizar sua própria utilidade, pois o lucro está diretamente relacionado com o tipo de serviço oferecido (VOORSLUYS, 2011).

Tradicionalmente, para tratar picos de carga, as organizações muitas vezes projetavam os data centers com poder de processamento suficiente para gerencia-lo, o que significa que para a maioria do tempo os data centers não foram totalmente utilizados. Ao utilizar nuvem, uma organização pode construir um centro de dados com as especificações que permitirá à entidade executar todas as cargas de trabalho normal do dia a dia dentro de seu ambiente e, em seguida, usar provedores de cloud computing para fornecer recursos adicionais para gerenciar cargas de pico.

Computação utilitária é freqüentemente associada a algum tipo de plataforma de virtualização que permite que uma quantidade quase infinita de armazenamento e processamento possa ser disponibilizado para as plataformas de usuários por meio de *datacenters*. A evolução da computação em nuvem é agora expandir a definição de computação utilitária para incluir serviços além dos de infraestrutura. (CHANTRY, 2009)

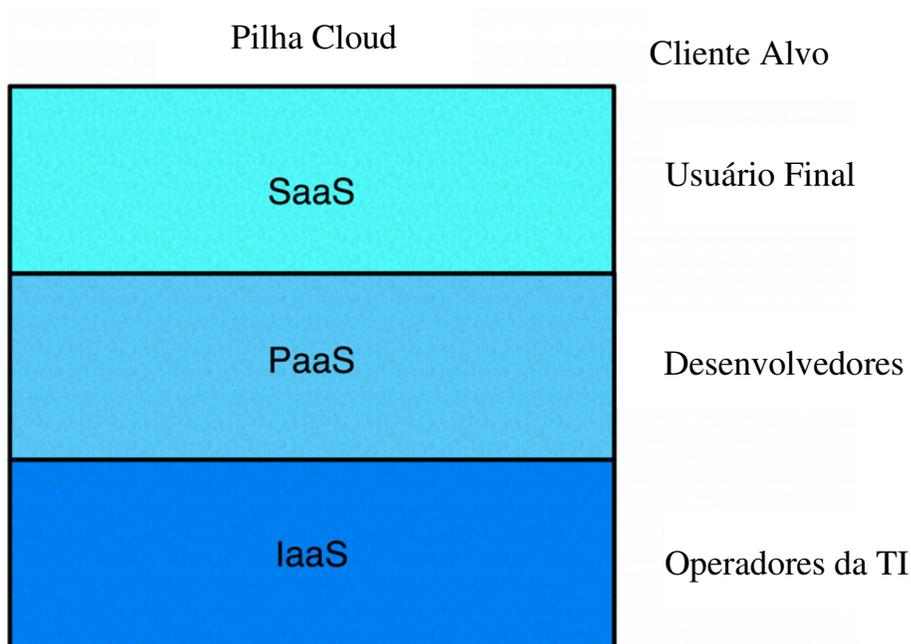
## **2.8 Camadas e modelos**

Os modelos de computação em nuvem são divididos em três camadas, de acordo com o nível de abstração da capacidade fornecida e do modelo de serviço dos prestadores: IaaS

(infraestrutura como serviço), PaaS (plataforma como serviço), e SaaS (software como um serviço). A figura 09 mostra esta organização (VOORSLUYS et al. 2011).

- **Infraestrutura como serviço (IaaS):** Nesse cenário o serviço possui uma infraestrutura de hardware, que é responsável pelo processamento e armazenamento de dados. Nessa infraestrutura é presente a tecnologia de virtualização. O princípio básico da virtualização é o compartilhamento da mesma máquina física por diferentes máquinas virtuais. Pode-se oferecer essas máquinas virtuais para diversos clientes, dividindo seus recursos de máquina entre eles. Esses clientes utilizam esses recursos virtualizados para oferecer seus serviços. Por exemplo, é oferecida para o cliente uma máquina virtual com um sistema operacional instalado e o cliente irá usa-la para executar as suas aplicações (VAQUERO et al. 2008).
- **Plataforma como serviço (PaaS):** Uma plataforma de nuvem é a capacidade de construir, testar, implementar, executar e gerenciar aplicativos na nuvem. Plataformas de Cloud oferecem alternativas a essas ações, por exemplo, a experiência de construir pode ser apenas online ou apenas off-line, ou uma combinação dos dois (CHANTRY, 2009). Temos como exemplo Google AppEngine, uma de plataforma como serviço, oferece um ambiente escalável para o desenvolvimento e hospedagem de aplicações Web.
- **Software como serviço (SaaS):** Neste modelo o objetivo é permitir ao cliente a execução de determinadas aplicações na Nuvem, em alternativa à execução local dessas aplicações, que em alguns casos necessita de recursos de alto valor. Assim, esse tipo de serviço possui um nível de abstração acima do PaaS, oferecendo ao cliente apenas o uso da aplicação (VAQUERO et al. 2008). Por exemplo, pode ser oferecido ao cliente um editor de texto, como o Google Docs.

Figura 09- Modelos de Serviço



Fonte: Randybias, 2009

Os modelos só funcionam se tudo for compatível com acordos de nível de serviço (Service Level Agreement – SLA) definidos com os clientes.

## 2.9 Acordo de Nível de Serviço (SLA)

O acordo de nível de serviço (SLAs) é oferecidos por Prestadores de IaaS para expressar seu compromisso com a entrega de uma qualidade de serviço (QoS) determinada. Para os clientes isto serve como uma garantia. Um SLA geralmente inclui a garantia de disponibilidade e desempenho. Além disso, medidas devem ser acordadas por todas as partes, bem como as penalidades pela violação ou não cumprimento dessas expectativas. A maioria dos provedores IaaS concentram seus termos de SLA na garantia de disponibilidade, especificando o percentual mínimo de tempo que o sistema estará disponível durante um determinado período. Por exemplo, a Amazon EC2 diz que "se o tempo de funcionamento anual para um cliente cai abaixo de 99,95% para o ano de serviço, o cliente é elegível a receber um crédito de serviço equivalente a 10% do valor acertado pelo serviço (VOORSLUYS et al. 2011).

SLA é essencial para o controle dos serviços prestados, assim assegurando em ambas as partes a garantia do serviço.

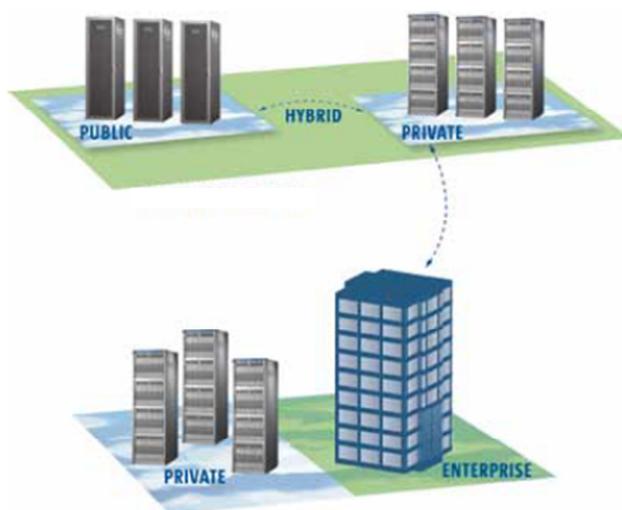
## 2.10 Tipos

As nuvens podem ser classificadas em três tipos básicos: públicas, privadas, comunitárias e híbridas. A escolha entre elas depende das necessidades das aplicações. Abaixo, esses tipos de nuvem são descritos (DIKAIKOS et al. 2009)(SUN, 2009).

- **Nuvem Privada:** A infraestrutura de nuvem é operada exclusivamente em uma organização ou empresa. Pode ser gerenciado pela empresa ou por um terceiro e pode existir no local ou fora do local. (NIST, 2011), a nuvem privada não consiste em vender serviço mas prover um serviço melhor em nuvem para sua empresa, sem ter os riscos e problemas de segurança caso estivesse usando uma nuvem pública.
- **Nuvem Pública:** A nuvem pública é fornecida por um prestador de serviços para o público geral usando como base a computação utilitária que tem o modelo de consumo pay-per-use. Os recursos da nuvem são geralmente hospedados nas instalações do prestador de serviços mas estes data centers podem estar em qualquer lugar do mundo. Exemplo de nuvens públicas são Amazon (EC2), Google gmail, e Azure da Microsoft (MOHAN, 2011).
- **Nuvem Comunitária:** A infraestrutura da nuvem é compartilhada por várias empresas e suporta uma determinada comunidade que visam o mesmo objetivo ou meta como, por exemplo, a missão, os requisitos de segurança, a política, entre outros. Pode ser gerenciado pela empresa ou por um terceiro e pode existir no local ou fora do local (NIST, 2011).

- Nuvem Híbrida: A nuvem híbrida é a composição de duas ou mais nuvens (privada, comunitária, ou pública) que permanecem entidades únicas, mas estão unidas por tecnologia padronizada que permite a portabilidade de dados e de aplicativos (MOHAN, 2011).

Figura 10- Nuvem Híbrida



Fonte: SUN 2009

É válido destacar que as nuvens híbridas introduzem a complexidade de determinar a maneira como as aplicações são distribuídas entre nuvens públicas e privadas para obter as vantagens dos dois tipos de nuvem.

## 2.11 Prós e Contras

Como todo paradigma, a computação em nuvem tem suas vantagens e desvantagens. As principais vantagens são:

- Elasticidade rápida: A computação em nuvem dá a ilusão de infinitos recursos computacionais disponíveis sob demanda. Portanto, os usuários esperam que a nuvem forneça rapidamente os recursos em qualquer quantidade a qualquer momento. Em particular, espera-se que os recursos adicionais possam ser fornecidos automaticamente, quando uma carga de aplicativos aumenta e também tenham seus recursos liberado quando diminui a carga (VOORSLUYS, 2011).
- Serviço Medido: Sistemas de cloud automaticamente controlam e otimizam a utilização dos recursos, alavancando a capacidade de medição em algum nível de abstração adequado para o tipo de serviço como por exemplo, armazenamento, processamento, banda, entre outros. Uso de recursos pode ser monitorado, controlado e relatado proporcionando transparência para o fornecedor e o consumidor do serviço utilizado (NIST, 2011).
- Disponibilidade e acesso em qualquer lugar a qualquer hora: Com os benefícios da internet, a mobilidade torna-se uma excelente característica nessa nova tecnologia. Como os dados agora estão na Nuvem, eles podem ser acessados de qualquer computador que tenha acesso à Internet, e em qualquer lugar.
- Economia: O modelo *pay per use* tende a baratear, pois só é gasto o que realmente é usado. Também podem ser considerados os casos onde não é mais necessário pagar por uma licença definitiva de um determinado programa, já que em alguns casos é mais viável a tarifação do uso específico do software.

As principais desvantagens são:

- Confiabilidade : Um sistema é dito confiável se ele não falha com frequência e, mais importante, se ele não perde os dados ao falhar (SUN, 2011). As aplicações desenvolvidas para a computação em nuvem devem ser confiáveis, ou seja, elas devem possuir uma arquitetura que permita que os dados permaneçam intactos mesmo

que haja falhas ou erros em um ou mais servidores ou máquinas virtuais sobre os quais essas aplicações estão sendo utilizadas. Isto também é um assunto que tem que ser definido na SLA, pois caso ocorra uma falha, uma penalidade deve estar descrita no acordo. O problema não é qual a penalidade a ser aplicada e sim que os dados não podem de jeito algum ser perdido. Essa característica está associada à realização de cópias de segurança dos dados. O armazenamento dessas cópias deve ser feito em local seguro para que, caso haja alguma falha nas aplicações e elas percam os dados, estes, ou pelo menos uma parte deles, possam ser recuperados.

- **Segurança:** O grande e maior problema da computação em nuvem é a segurança. Este problema está mais aplicado na implementação de uma nuvem pública, que no caso os *datacenters* armazenam as informações que os usuários tradicionalmente armazenariam em seus próprios computadores, ou seja, esses usuários desconhecem tanto a localização exata de seus dados quanto a fonte dos dados que estão armazenados junto aos deles (KAUFMAN, 2009). Assim, a proteção da privacidade dos usuários e a integridade das informações devem ser consideradas pelos prestadores de infraestrutura e de serviços.
- **Padrões:** Paradigma novo, em seu estado atual temos alguns padrões porém nada muito sólido, dificultando troca de provedores ou de recursos por outras nuvens.
- **Interoperabilidade:** A interoperabilidade diz respeito à capacidade dos usuários de executar os seus programas e os seus dados em diferentes nuvens e plataformas. Isso permite, por exemplo, que as aplicações não fiquem restritas a somente uma nuvem. Essa é uma característica amplamente desejável no ambiente da computação em nuvem. Ultimamente, muitas aplicações têm sido desenvolvidas considerando esse fator. Porém, ainda há a necessidade da implementação de padrões e interfaces para que essa portabilidade seja possível (DIKAIAKOS et al. 2009).

## 2.12 Segurança

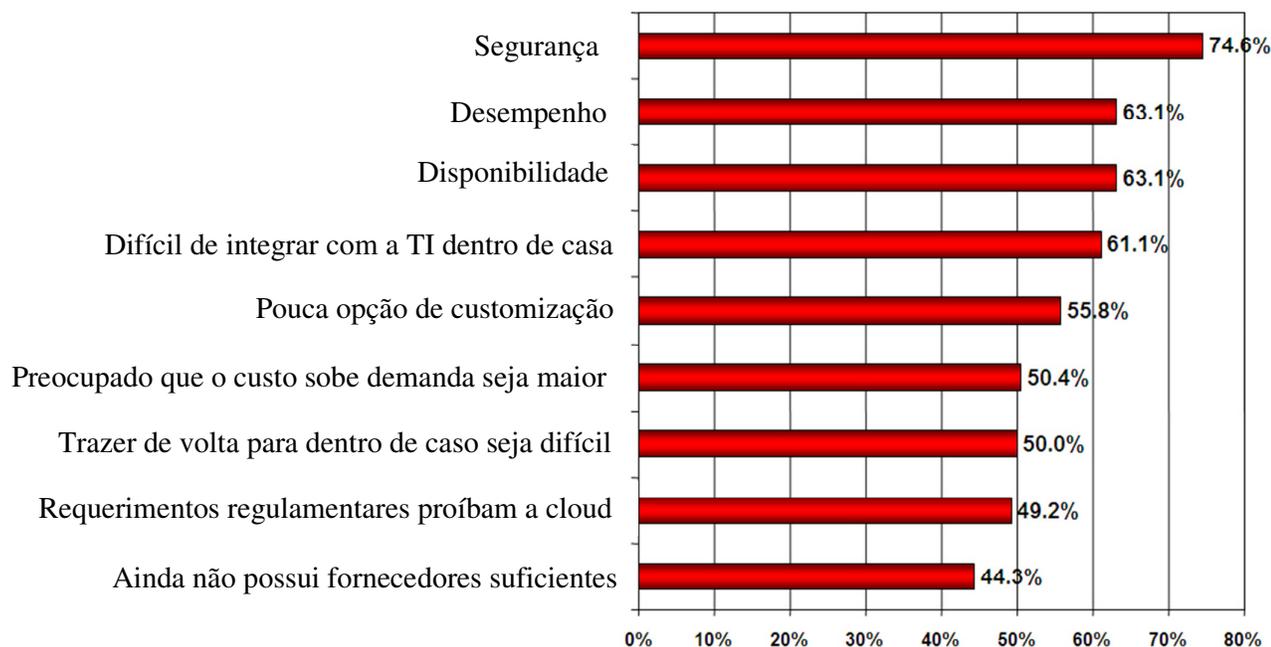
Segurança, uma palavra crítica em computação em nuvem, de acordo com a figura 11, segurança é o maior problema da migração para as nuvens.

Levar a informação e torná-la segura de modo que somente você ou determinadas pessoas possam ter acesso, obviamente não é um conceito novo. No mundo real, mesmo a informação guardada a sete chaves está sujeito a roubo e é certamente susceptível de utilização acidental ou mal intencionada. No mundo digital, essa analogia de proteção a sete chaves pode ser vista como a informação estando criptografada. Mas mesmo a nossa tentativa de proteger a informação digital tem-se revelado não eficaz, devido às limitações inerentes proteger uma informação criptografada, e não no conteúdo dessa informação. Esta limitação se tornou mais evidente à medida que entramos na era da computação em nuvem, onde a informação tem muito mais dinamismo ao invés da informação que é estática em um computador pessoal ou em uma pasta de rede, por isso agora nós precisamos começar a pensar em uma nova maneira de proteger informações (SUSAN MORROW, 2011).

Segundo (DIKAIAKOS et al. 2009), na computação em nuvem, os data centers contêm informações que os que mais tradicionalmente os usuário teriam armazenados em seus computadores. Isto gera uma preocupação para o usuário em questão da proteção de privacidade, pois agora os dados estão sendo armazenado por terceiros. Além disso, a mudança para serviços centralizados podem afetar a privacidade e a segurança em questão das interações dos usuários. As ameaças de segurança podem acontecer no provisionamento de recursos e durante a execução de aplicativos distribuídos. Por exemplo, os hackers podem utilizar a infraestrutura virtualizada como um lançamento para novos ataques. Serviços de nuvem devem preservar a integridade dos dados e a privacidade do usuário e ao mesmo tempo, devem aumentar a interoperabilidade entre vários prestadores de serviços em nuvem.

Neste contexto, temos de investigar novos tipos de mecanismos para segurança de dados, para garantir a privacidade dos dados, garantia de segurança sobre os recursos e os direitos autorais do conteúdo.

Figura 11- Gráfico problemas relacionado a computação em nuvem



Fonte: Gens, 2008

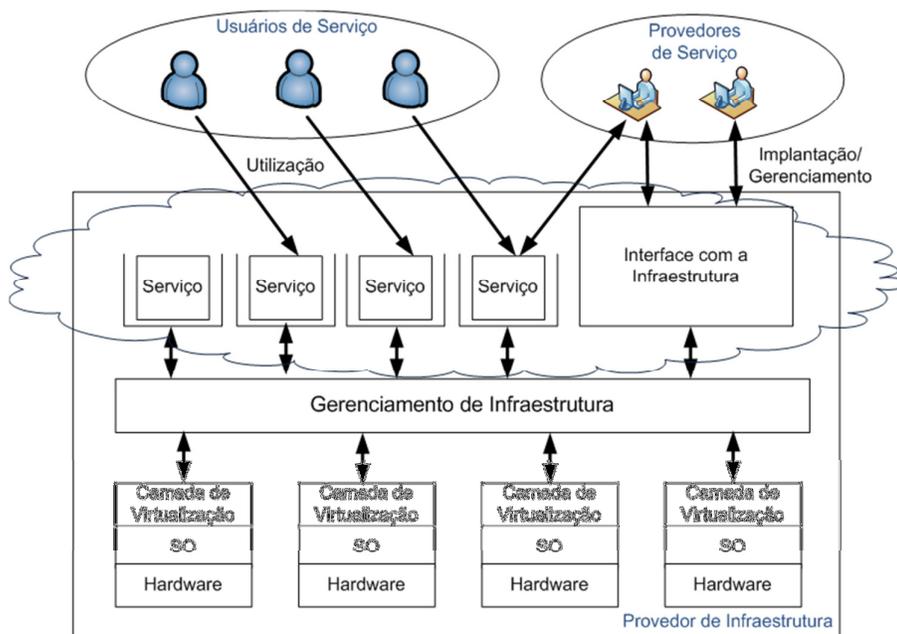
### 2.13 Arquitetura

Assim como diversas questões sobre computação em nuvem, ainda não existe uma definição única de arquitetura ideal para esse tipo de tecnologia (NURMI et al. 2009). Em uma arquitetura de computação em nuvem, diversos usuários compartilham recursos de um mesmo provedor. Para haver redução de custos e manter todos os serviços funcionando, um provedor deve alocar os recursos de seu *datacenters* para cada cliente de forma eficiente. Em (BUYA et al. 2009) é definida uma arquitetura de computação em nuvem orientada a mercado, isto é, que suporte alocação de recursos do provedor baseado na necessidade ou no contrato com cada cliente. De acordo com (BUYA et al. 2009), um gerenciamento de recursos orientado a mercado é necessário para regular a oferta e a demanda dos recursos da Nuvem com

objetivo de atingir equilíbrio de mercado, ou seja, é necessário regular a oferta para que ela seja igual à demanda, assim tanto os provedores quanto os clientes beneficiar-se-ão em termos de redução de custos promovida pela diferenciação de serviços.

Antes de mostrar alguns exemplos de arquitetura, vamos mostrar alguns elementos da arquitetura, segundo (VAQUERO, 2009). A computação em nuvem é composta por três atores principais: os prestadores de serviços, os usuários dos serviços, e os prestadores de infraestrutura. Os prestadores de serviços são aqueles que desenvolvem e deixam os serviços acessíveis aos usuários. Esses serviços, por sua vez, necessitam de recursos, de uma infraestrutura, sobre a qual estará instalada, essa infraestrutura é fornecida como um serviço pelos prestadores de infraestrutura, esta relação é mostrada na figura 12.

Figura 12- Arquitetura adaptada de computação em nuvem



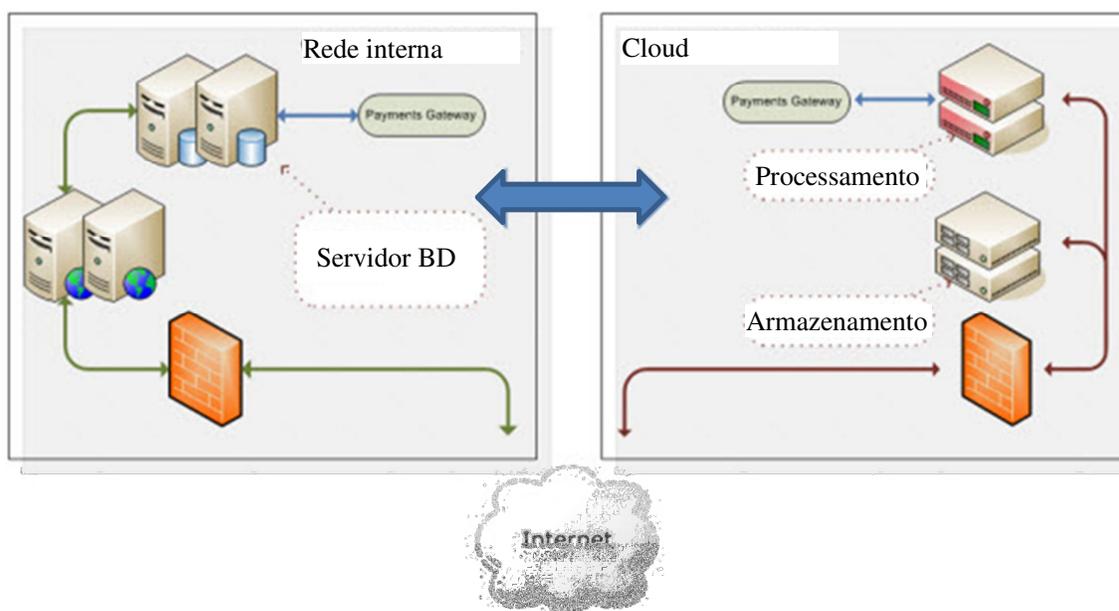
Fonte: Buyya et al. 2009

Como a computação em nuvem tem a sua arquitetura voltada a mercado temos algumas principais soluções arquiteturais de serviço de nuvem tais como:

- Arquitetura para solução de alta demanda por um curto período de tempo:

Neste caso os acessos aos *datacenters* estão frequentemente sobrecarregados durante os períodos de alta demanda, com isso a necessidade de recursos computacionais elevado é extremamente requisitado. Com isso vem a capacidade de executar instâncias de máquinas virtuais para cobrir esses períodos, até às vezes tendo que ligar as máquinas físicas.

Figura 13- Arquitetura para soluções de alta demanda



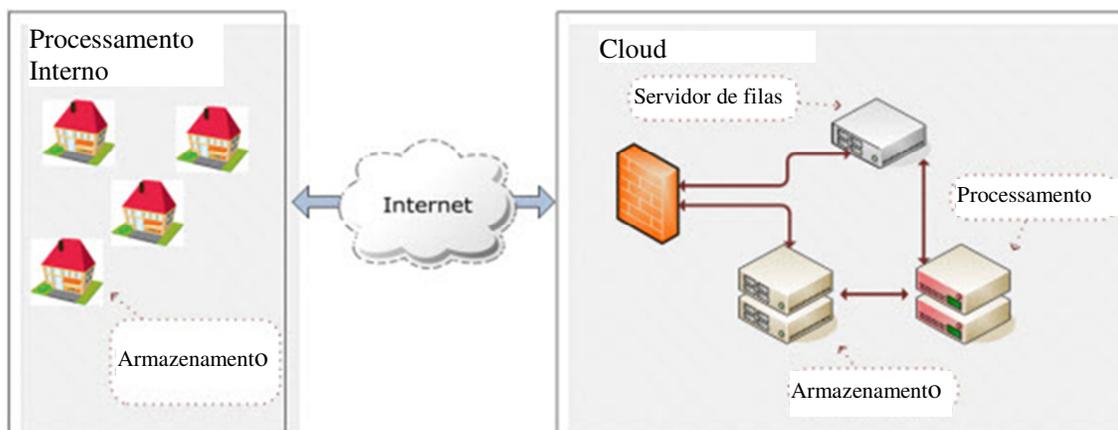
Fonte: Chantry 2009

Como mostrado na figura 13, esta é uma arquitetura para solucionar este problema, fazendo a divisão da carga de processamento e armazenamento assim uma parte acontecendo na nuvem e a outra internamente (CHANTRY, 2009).

- Arquitetura para solução de alta quantidade de processos:

Neste caso não temos recursos interno para o processamento e armazenamento de dado. Com isso há necessidade de se fazer as tarefas em um serviço de nuvem público que guarda e processa os dados devidamente tendo o seu acesso pela Internet (figura 14).

Figura 14- Arquitetura para soluções de alta quantidade de processos

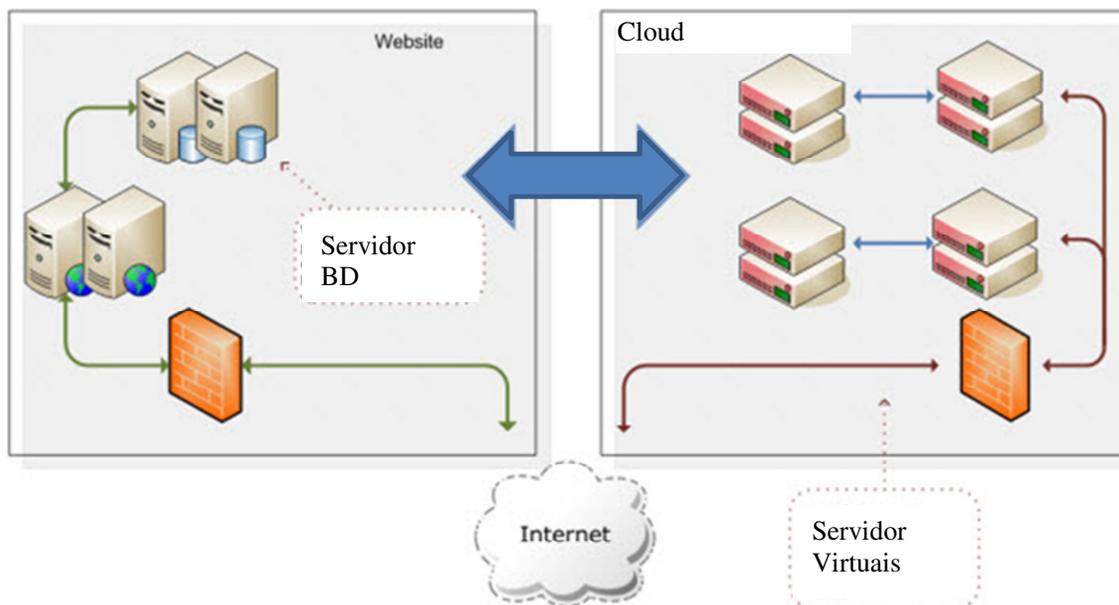


Fonte: Chantry 2009

- Arquitetura para solução de alta demanda em um período irregular:

Neste caso os acessos aos *datacenters* são sobrecarregados durante certos períodos, porém sem previsibilidade. Por isso seria impraticável passar o serviço para a nuvem em um determinado pico de carga. Assim há necessidade de criar uma cópia do seu modelo interno e, alocá-lo em nuvem como é demonstrado na figura 15. (CHANTRY, 2009)

Figura 15- Arquitetura para soluções de alta demanda em um período irregular



Fonte: Chantry 2009

## 2.14 Padrões

- A padronização é importante para garantir a interoperabilidade entre fornecedores de gerenciamento de virtualização, entre as máquinas virtuais produzidos por cada um deles, e a computação em nuvem. Como a padronização na computação em nuvem não é sólida temos dois principais padrões, Open Cloud Computing Interface e Open Cloud Consortium

:

- Open Cloud Computing Interface (OCCI)

A OCCI tem como objetivo a entrega de uma especificação de API para o gerenciamento remoto da infraestrutura da computação em nuvem e para permitir o desenvolvimento de ferramentas de interoperabilidade para tarefas comuns, incluindo a implantação, dimensionamento autônomo e monitoramento. O escopo da especificação é a cobertura de uma funcionalidade de alto nível necessário para gerir o ciclo de vida da máquinas virtuais,

rodando em tecnologias de virtualização, e dando suporte aos serviços elásticos. A nova API de interface IaaS vai permitir vários aspectos, tais como:

- Os consumidores possam interagir com a infraestrutura de computação em nuvem numa base *ad hoc*.
- Integradores possam oferecer serviços avançados de gestão, agregadores possam oferecer uma única interface comum a vários provedores.
- Provedores possam oferecer uma interface padrão que seja compatível com as ferramentas disponíveis.

O Open Cloud Computing Interface é um protocolo de fronteira e API que funciona como um serviço de front-end para framework de gerenciador interno do provedor. Os consumidores de serviço podem ser tanto os usuários finais quanto instâncias de outro sistema. OCCI é adequado para ambos os casos, a característica chave é que OCCI pode ser usado como uma API de gerenciamento para todos os tipos de recursos e, ao mesmo tempo manter um nível elevado de interoperabilidade (OCCI, 2011).

- Open Cloud Consortium

O Open Cloud Consortium (OCC) é uma organização dirigida por membros que apoia o desenvolvimento de padrões para computação em nuvem e quadros para interoperar entre as nuvens, desenvolve padrões de referência para a computação em nuvem, suporta implementações de referência para a computação em nuvem, de preferência, implementações de referência de código aberto, gerencia um *testbed* para Cloud Computing chamado Open Cloud Testbed (OCT) e patrocina workshops e outros eventos relacionados com computação em nuvem. A OCC tem um foco especial em nuvens de grandes dados. Ele desenvolveu o benchmark MalStone, e está trabalhando em um modelo de referência para as nuvens de grandes dados.

### 3 FERRAMENTAS E TECNOLOGIAS

Existem muitas ferramentas e tecnologias para implementação de computação em nuvem, nesta seção iremos descrever algumas delas.

#### 3.1 Eucalyptus

O Eucalyptus foi um dos primeiros projetos open source a se concentrar na criação de nuvens IaaS. Ele foi desenvolvido com a intenção de fornecer uma implementação open source quase idêntica em funcionalidade ao Amazon Web Services API. Portanto, os usuários podem interagir com uma nuvem Eucalyptus, usando as mesmas ferramentas que eles usam para acessar a Amazon EC2. Ele também distingue-se de outras ferramentas, pois fornece um API de nuvem de armazenamento emulando o API Amazon S3 para armazenar dados de usuários em geral e imagens de máquinas virtuais. Em resumo, o Eucalyptus oferece os seguintes recursos: Controlador baseado em Linux com a administração de um portal Web; compatibilidade EC2, S3, Xen, KVM e VMWare, compatibilidade com dispositivos de armazenamento virtual Amazon EBS, e interface para a nuvem pública Amazon EC2, e redes virtuais.(VOORSLUYS, 2011)

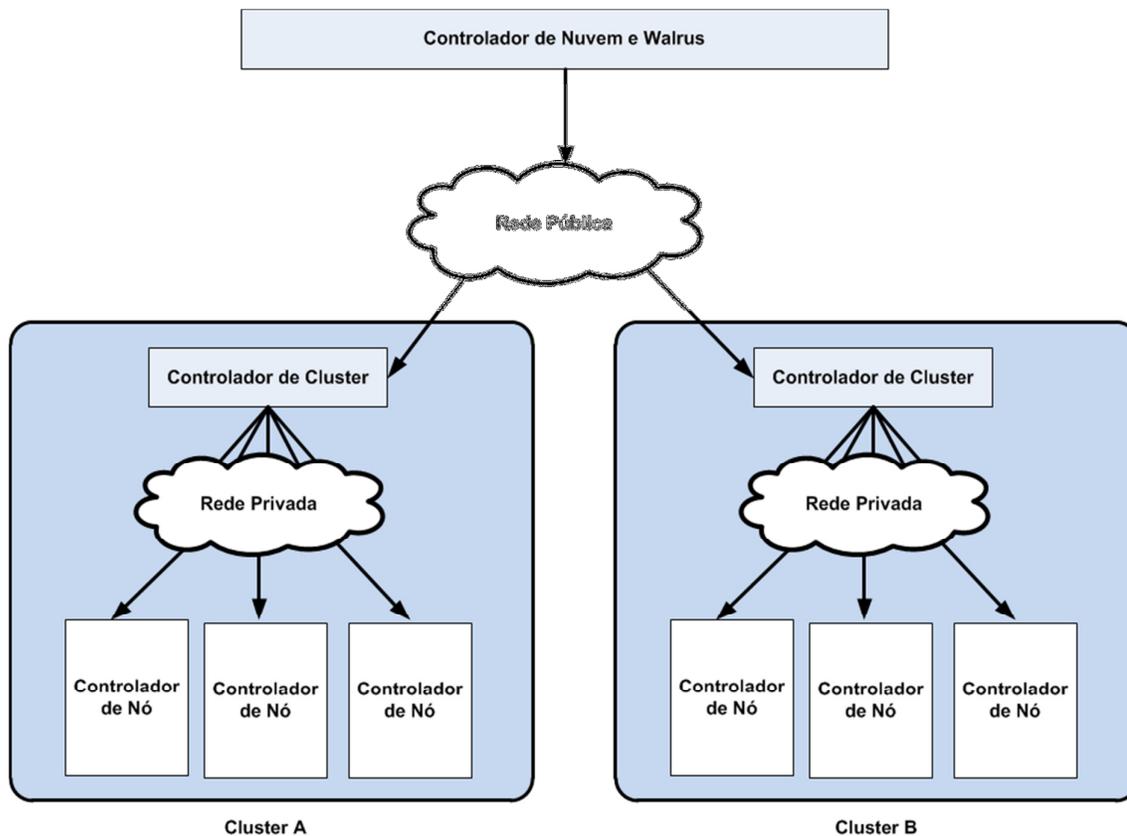
A arquitetura do Eucalyptus é modular e, como possui código aberto, facilita que pesquisadores desenvolvam diversas modificações à sua estrutura original. A Figura 16 mostra a sua estrutura e seus componentes. Esses componentes são implementados como web services e são detalhados a seguir segundo (NURMI et al. 2009):

- Controlador de armazenamento (Walrus): Esse componente é um serviço de armazenamento e possui uma interface com o usuário compatível com a API Amazon Simple Storage Service (Amazon S3). Com essa API os usuários podem colocar e retirar dados do walrus. Além de armazenar dados do usuário, o walrus armazena imagens de máquinas virtuais. Uma imagem de máquina virtual possui o sistema de arquivos, o kernel e o ramdisk necessários para instanciar as máquinas virtuais. Os nós da arquitetura podem enviar imagens de máquina virtual para o walrus bem como

baixá-las para poderem instanciar as máquinas. Para esse envio e recebimento de imagens, os nós utilizam ferramentas do amazon EC2.

- Controlador de nó: Esse componente executa em todo o nó, ou na máquina física designada para rodar as máquinas virtuais. O controlador de nó é responsável pelo controle das máquinas virtuais de uma determinada máquina física, assim esse componente pode iniciar, parar e obter informações das máquinas virtuais. Essas ações sobre a plataforma de virtualização são realizadas pela comunicação entre o controlador de nó e o hipervisor. O hipervisor realiza o gerenciamento do hardware da máquina física para o seu compartilhamento entre as diversas máquinas virtuais.
- Controlador de nuvem: Esse componente é a porta de entrada do sistema para usuários e administradores. Esse componente é responsável por atender às requisições dos usuários, como manipulação das propriedades das máquinas virtuais ou das redes, bem como requisições de informações sobre o estado das máquinas virtuais. Ele também controla o armazenamento de dados dos usuários e de estado do sistema, além disso, possui um serviço no qual são oferecidas interfaces para usuários ou administradores obterem acesso ao sistema e alterarem suas propriedades.
- Controlador de clusters: A partir de requisições aos controladores de nó, esse componente recolhe informações sobre as máquinas virtuais de um *cluster*, ou de uma rede privada, e realiza o escalonamento das requisições de instanciação das máquinas virtuais entre os diversos Controladores de nó e também é responsável por gerenciar redes, que são formadas cada uma por um grupo de instâncias das máquinas virtuais.

Figura 16 – Arquitetura Eucalyptus adaptada



Fonte: NURMI et al. 2009

### 3.2 Amazon EC2

O Amazon Elastic Compute Cloud, ou Amazon EC2, fornece um IaaS no modelo *pay per use*. Ele provê um ambiente virtual que permite o desenvolvimento e a execução de aplicações baseadas em Linux. Para isso, os prestadores de serviços devem criar sua própria máquina virtual, que contém todas as aplicações, os dados e as bibliotecas necessárias. Alternativamente, os prestadores de serviços também podem usar imagens de máquinas virtuais que já estão prontas e disponíveis (BUYA et al. 2008).

### 3.3 OpenNebula

Segundo (VOORSLUYS et al. 2011), o OpenNebula é um dos gerenciadores de infraestruturas virtuais open source mais rica em recursos. Ele foi inicialmente concebido para

gerir uma infraestrutura virtual local, mas também incluiu interfaces remotas que tornam viável a construção de nuvens públicas. Ao todo, quatro APIs de programação estão disponíveis: XML-RPC e libvirt para a interação local, um subconjunto do EC2 (consulta) e os APIs Cloud OpenNebula API para acesso público.

Sua arquitetura é modular, abrangendo vários plug-ins. O módulo core orquestra servidores físicos e seus hipervisores, nós de armazenamento e estrutura da rede. As operações de gerenciamento são realizados através drivers conectados, que interagem com as APIs de hipervisores, e as nuvens públicas. O módulo Scheduler, que está em responsável pela atribuição de pedidos pendentes das máquinas virtuais aos hosts físicos, oferece alocação dinâmica de recursos disponíveis. Os administradores podem escolher entre diferentes objetivos de programação, tais como a manter as máquinas virtuais em poucas máquinas físicas ou manter a carga equilibrada. Em resumo, OpenNebula oferece os seguintes recursos: controlador baseado em Linux, consulta compatível com EC2, Xen, KVM e Vmware, interface para as nuvens públicas, alocação dinâmica de recursos, e reserva antecipada de capacidade.(VOORSLUYS et al. 2011). Em seguida é mostrada na tabela 1 uma comparação entre os recursos de alguns gerenciadores de infraestrutura virtual.

Tabela 1- Tabela adaptada de comparação de infraestrutura virtual

	Licença	Plataforma de instalação do controlador	Cliente UI, API, Linguagem	Hypervisor	Virtualização de armazenamento	Interface para nuvem pública	Redes Virtuais	Alocação dinâmica de recursos	Adiantamento de reserva de capacidade	Disponibilidade alta	Proteção de dados
Eucalyptus	BSD	Linux	EC2 WS, CLI	Xen, KVM	No	EC2	Yes	No	No	No	No
Nimbus	Apache v2	Linux	EC2 WS, WSRF, CLI	Xen, KVM	No	EC2	Yes	Via integration with OpenNebula	Yes (via integration with OpenNebula)	No	No
OpenNEbula	Apache v2	Linux	XML-RPC, CLI, Java	Xen, KVM	No	Amazon EC2, Elastic Hosts	Yes	Yes	Yes (via Haizea)	No	No
OpenPEX	GPL v2	Multiplatform (Java)	Portal, WS	XenServer	No	No	No	No	Yes	No	No
oVirt	GPL v2	Fedora Linux	Portal	KVM	No	No	No	No	No	No	No
Platform ISF	Proprietary	Linux	Portal	Hyper-V, XenServer, VMWare ESX	No	EC2, IBM CoD, HP Enterprise Services	Yes	Yes	Yes	Unclear	Unclear
Platform VMO	Proprietary	Linux, Windows	Portal	XenServer	No	No	Yes	Yes	No	Yes	No
VMWare vSphere	Proprietary	Linux, Windows	CLI, GUI, Portal, WS	VMware ESX, ESXi	VMware vStorage VMFS	VMware vCloud partners	Yes	VMware DRM	No	Yes	Yes

Fonte: Voorsluys et al. 2011

### 3.4 Nagios

Nagios é um sistema de monitoramento que permite às organizações a identificar e resolver problemas de infraestrutura de TI antes que eles afetem os processos críticos de negócio. Projetado com escalabilidade e flexibilidade, Nagios tem como missão fazer que os seus processos de negócios não serão afetados por interrupções desconhecido. Nagios é uma ferramenta que lhe proporciona avisos imediatos sobre os principais objetivos da infraestrutura de TI da sua empresa. Nagios permite detectar e reparar e atenuar futuros problemas antes que eles afetem os usuários finais (NAGIOS 2011).

A filosofia do Nagios pode ser descrita em seis etapas:

- Monitoramento: Configurar os componentes críticos da infraestrutura de TI, incluindo métricas, protocolos, aplicações, serviços, e infraestrutura de rede.
- Alerta: Nagios manda um alerta quando o componente crítico da infraestrutura falha e se recupera, assim avisando os administradores sobre eventos importantes.
- Resposta: A equipe de TI pode reconhecer alertas e começar a resolver falhas e investigar sobre elas imediatamente. Os alertas podem ser escalados para grupos diferentes se os alertas não forem reconhecidos em tempo
- Relatar: Os relatórios fornecem um registro histórico de falhas, eventos, notificações, e resposta de alertas. A disponibilidade de relatórios ajuda a garantir se os SLAs estão sendo cumpridos.
- Manutenção: Manutenção programada impede alertas durante atualizações e manutenções.

- Planejamento: Gráficos de tendências e planejamento de capacidade e relatórios permitem identificar melhorias necessárias da infra-estrutura antes de ocorrer uma falha.

### 3.5 Icinga

Icinga é um *fork* do Nagios e é totalmente compatível, assim, as configurações do Nagios como plugins e *addons* podem ser usados com o Icinga. Embora Icinga mantenha todas as características existentes do seu antecessor, foram acrescentadas muitas atualizações que foram solicitadas por usuários da comunidade. Escalável e extensível, o Icinga pode monitorar ambientes complexos e grandes.

### 3.6 HybridFox

HybridFox é uma extensão para o navegador Firefox que nada mais é que uma interface amigável que tem o papel de interagir com Web Services compatíveis com o S3 e o Amazon EC2. As suas principais funções são:

- Gerenciar imagens;
- Rodar e parar instâncias;
- Gerenciar instancia;
- Gerenciar endereços Ips
- Simplificar o uso do ferramenta.

### **3.7 PCMONS**

O PCMONS( Private Clouds Monitoring Systems) é um sistema de monitoramento extensível e modular para nuvens privadas. O PCMONS foi criado em 2010 por Rafael Uriarte e Shirlei Aparecida, e atua principalmente na recuperação, coleta e preparação de informações relevantes para o monitoramento de dados, e é especialmente focado em máquinas virtuais. Na sua primeira versão ele é compatível com o Eucalyptus e com o Nagios (URIARTE, 2011).

### **3.8 EUCA2TOOLS**

Euca2ools são ferramentas de linha de comando para interagir com serviços Web. Esta ferramenta pode ser usada tanto em serviços da Amazon quanto na instalação do Eucalyptus. Foi inspirada pelas ferramentas de linha de comando distribuídas pela Amazon e aceita as mesmas opções e variáveis do ambiente. No entanto, essas ferramentas foram implementadas do zero, feita em Python com a biblioteca Boto e M2Crypto toolkit (EUCALYPTOS, 2011).

## **4 ESTUDO DE CASO**

### **4.1 Definição da proposta**

Com o projeto pretende-se construir um sistema web com métricas de acordo SLA, e construir um ambiente de computação em nuvem utilizando software open source, onde o sistema se comunica com o ambiente com o fim de manter um histórico e gerar relatórios para o cliente e o prestador de serviço, para futuros ajustes no acordo ou novos meios de gerenciamento de máquinas virtuais e ou físicas fornecidas.

Suas ações têm como objetivo viabilizar o projeto desde a preparação do ambiente, passando pela implantação do sistema e do relacionamento arquitetural.

### **4.2 Detalhamento do ambiente e das ferramentas do processo**

Para o desenvolvimento do projeto foram utilizados dois computadores que seguem as seguintes configurações:

Para o computador com a finalidade de ser o Controlador de Cluster (CC), Controlador de Armazenamento (SC), Controlador de Nuvem (CLC), Walrus, e o Controlador de Nó (NC) temos:

- Processador: AMD Phenom (TM)9650 Quad-Core @ 2.3 GHz
- Cache do Processador: 2MB L2 Cache
- Memória: 4GB DDR2 667MHZ
- Disco Rígido: 750GB Interface Serial ATA II

Neste mesmo computador foram usados os seguintes softwares e ferramentas:

- Eucalyptus 2.0.2
- Xen 3.4.1
- OpenSUSE 11.2
- Nagios 3.2.3
- Euca2ools
- PCMONS

Para o outro computador que tem a finalidade de ser somente o Controlador de Nó (NC) temos:

- Processador: Intel(R)Core(TM)2 Quad Q8200 @ 2.33 GHz
- Cache do Processador: 2MB L2 Cache
- Memória: 3GB DDR2 333MHZ
- Disco Rígido: 320GB Serial ATA 3.0Gbps

Neste computador foram usados os seguintes softwares e ferramentas:

- Eucalyptus 2.0.2
- Xen 3.4.1
- OpenSUSE 11.2

As escolhas dos softwares apresentados foram:

- Eucalyptus: A sua escolha para a plataforma de computação em nuvem foi feita pois além de oferecer suporte para várias versões do Linux, é bastante documentada, foi usada a versão Community, e um dos principais motivos também, foi por ter uma interface parecida com a da Amazon EC2, um dos serviços para nuvem mais utilizados.
- Xen: Um hipervisor de código aberto, ele foi escolhido pois além de ser compatível com o Eucalyptus, que no caso é o nosso software de computação em nuvem escolhido, ele usa o mecanismo da paravirtualização, não precisando de suporte nativo à virtualização pelo hardware, que no nosso caso as duas máquinas usadas, nenhuma delas tem seus respectivos processadores com suporte à virtualização.
- OpenSUSE: O nosso sistema operacional foi escolhido pela compatibilidade com o Eucalyptus e também ao Xen, e principalmente pela facilidade obtida ao instalar, pois foi tentado usar o Debian para o experimento mas a dificuldade de instalação e erros obtidos sem ter a sua respectiva resposta, levou a ser usado o OpenSUSE.
- Nagios: A ferramenta para o monitoramento foi escolhida pois é compatível com o Eucalyptus e também é muito bem documentada, tendo também muitas opções de incrementos através de plugins.

### **4.3 Detalhamento do processo de instalação dos softwares e ferramentas**

Para o desenvolvimento do projeto foi usado a primeira máquina descrita em 6.2, porém antes de realizar os procedimentos na máquina, já existia o sistema operacional Debian nela instalado, só que ao tentar configurar todo o ambiente não obtive sucesso devido a vários erros sendo gerados ao tentar instalar os outros softwares necessários. Tendo estes problemas, a solução foi instalar o OpenSUSE.

Para instalar o OpenSUSE, que possui suporte ao XEN, tendo um kernel modificado necessário para instalação do mesmo, e também ao Eucalyptus, foi usado a versão 11.2, uma instalação normal nas duas máquinas dedicando o disco rígido inteiro para esta aplicação dando suporte as requisições de máquinas virtuais.

Uma das opções escolhida para esta instalação foi o não uso da interface gráfica, pois nenhuma das ferramentas necessita de interface. Para controle e uso da máquina foi instalado o ssh.

O Xen foi instalado via uma ferramenta chamada Yast que por padrão tem o hypervisor Xen, com isso a instalação foi bem fácil assim como a sua configuração.

No caso do Eucalyptus, na sua instalação foi usado os pacotes disponibilizados no site da ferramenta, e seguindo o tutorial de instalação este procedimento foi um pouco mais tranquilo.

- Para a configuração do Eucalyptus, neste caso na parte de rede, foi necessário escolher uma das opções fornecidas pelo Eucalyptus como mostra a figura 17, que são:

Figura 17 – Modos de configuração de rede do Eucalyptus

Modo de Rede	Requisitos físicos			Características do EUCALYPTUS					
	VLAN limpa	IP público	DHCPD ≠ Eucalyptus	Conectividade	Controle IP	Grupos de segurança	IPs Elásticos	Serviço de metadados	Isolamento MV
MANAGED	✓	✓		✓	✓	✓	✓	✓	✓
MANAGED-NOVLAN		✓	X	✓	✓	✓	✓	✓	✓*
STATIC		✓	X	✓	✓			✓	
SYSTEM			✓	✓				✓	

Fonte: Eucalyptus, 2011

- Gerenciado (Managed): Neste modo, o Eucalyptus gerencia a rede local das máquinas virtuais instanciadas e fornece todas as funcionalidades, incluindo isolamento de rede das MV, segurança, Ips elásticos e serviços de metadados.
- Gerenciado sem VLAN (Managed-NOVLAN): Neste modo, o Eucalyptus irá gerenciar totalmente a rede local das máquinas virtuais e fornece todas as funcionalidades exceto o isolamento de rede das MV.
- Sistema (System): Neste modo existe pouca configuração necessária para usar o Eucalyptus, neste caso o gerenciamento de redes não é feito pela ferramenta.
- Estático (Static): Neste modo, as atribuições dos endereços IP das máquinas virtuais são gerenciados pelo Eucalyptus, e isso é feito através do seu próprio servidor DHCP com uma entrada estática por MV.

Tendo em vista essas opções foi escolhida a opção System, pois temos um servidor externo DHCP que fica responsável pelas atribuições de IP. Tanto quanto o servidor, é necessário também configurar a ponte do hipervisor Xen que associa a interface de rede da máquina virtual com a interface física, e as configurações foram DHCP = UFSC e o bridge (ponte)= br0, as outras configurações foram definidas podendo ser vistas na interface web que o Eucalyptus fornece (Figura 18).

Figura 18- Configuração Eucalyptus

The screenshot shows a web browser window with the address bar displaying 'https://150.162.63.25:8443/#conf'. The browser tabs include 'First-time Setup (2.0) | Euc...', 'Sed - An Introduction and T...', and 'libvirt: The virtualization API'. The main content area is divided into three sections:

- Cloud configuration:**
  - Cloud Host:
  - Default kernel:
  - Default ramdisk:
  - Buttons:  and
- DNS configuration:**
  - Domain name:
  - Nameserver:
  - IP:
  - Buttons:  and
- Walrus Configuration:**
  - Walrus host:
  - Maximum bucket size (MB):
  - Maximum buckets per user:
  - Buckets Path:
  - Space reserved for unbundling images (MB):
  - Space reserved for snapshots (GB):

The bottom of the browser window shows the title bar: 'Eucalyptus - Mozilla Fir...'

Em seguida foi utilizada a ferramenta euca2ools para habilitar que as imagens escolhidas das máquinas virtuais possam ser uma entidade executável, utilizando três comandos EC2 é possível carregar as imagens para o Eucalyptus, para posteriormente usar o HybridFox para inicializar as MV.

Na configuração do Controlador de Armazenamento (SC), temos as seguintes configurações, que são mostradas na figura 19. Nesta aba podemos ver que o Eucalyptus nos dá opções de tipos de máquinas virtuais, ou seja, com essa especificação ao instanciar uma máquina virtual é necessário dizer qual o tipo, e para cada tipo de MV segue a correspondente configuração listada na figura.

Figura 19- Tipos de MV e controlador de armazenamento

150.162.63.25 https://150.162.63.25:8443/#conf

ime Setup (2.0) | Euc... ✕ Sed - An Introduction and T... ✕ libvirt: The virtualiz

### Storage Controller

Max volume size:

Disk space reserved for volumes:

Storage Interface:

Volumes path:

Zero-fill volumes

Clusters up to date

### VM Types:

Name	CPUs	Memory (MB)	Disk (GB)
m1.small	<input type="text" value="1"/>	<input type="text" value="128"/>	<input type="text" value="10"/>
c1.medium	<input type="text" value="1"/>	<input type="text" value="256"/>	<input type="text" value="20"/>
m1.large	<input type="text" value="2"/>	<input type="text" value="512"/>	<input type="text" value="30"/>
m1.xlarge	<input type="text" value="2"/>	<input type="text" value="1024"/>	<input type="text" value="40"/>
c1.xlarge	<input type="text" value="4"/>	<input type="text" value="2048"/>	<input type="text" value="50"/>

Saved VM Types to server

otus - Mozilla Fir...

Para finalizar, a instalação do Nagios foi feita via a ferramenta Yast que estava disponível pelo OpenSUSE, e foi configurado via um arquivo de configuração inicial fornecido pelo Eucalyptus, que monitora sus principais componentes. E por último foi instalado o PCMONS que está disponível em seu site (URIARTE, 2011).

#### 4.4 Desenvolvimento

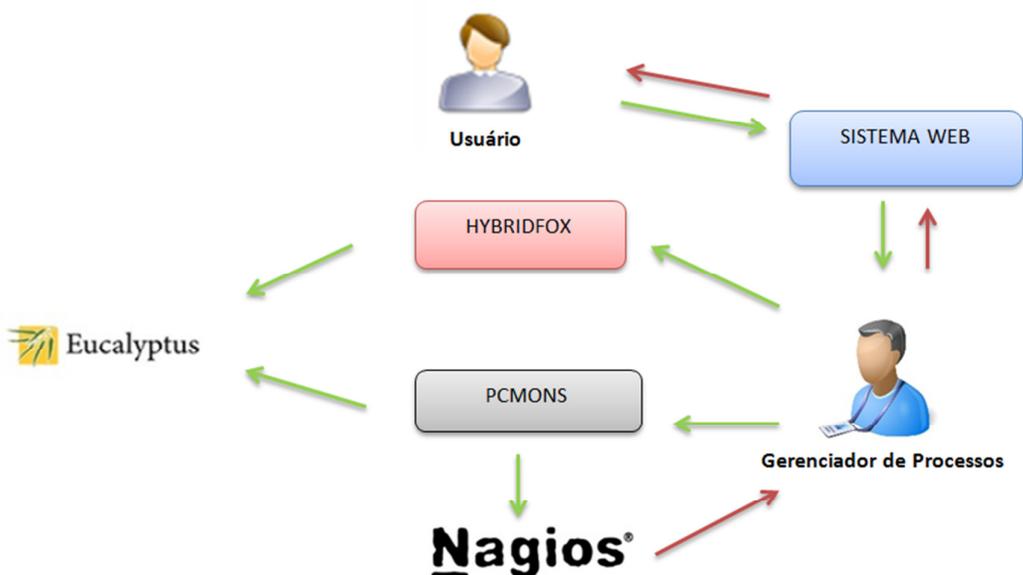
Considerando o ambiente de implantação da nuvem foi visto que a idéia de se ter este serviço é muito bom, porém para isto acontecer é necessário ser primeiramente viável. Como sabemos

a viabilidade deste processo está em ser possível a utilização desse serviço. E para isto acontecer é preciso de requisitos mínimos a serem seguidos. Além disso, é preciso ter um grau de confiabilidade neste serviço.

Após uma análise deste ambiente foi visto que é preciso uma definição de métricas simples para a viabilidade do processo contando com as necessidades do usuário final e também um sistema confiável de interação entre o usuário e o gerenciador do processo para mostrar o que realmente o serviço está oferecendo, e se não estiver, mostrar o que não foi cumprido para as devidas ações.

#### 4.5 Layout do projeto

Figura 20- Layout do Projeto



A estrutura do projeto é uma linha de conexões onde primeiro temos o usuário, que se comunica com o gerenciador de projetos através do sistema web. Em seguida o gerenciador de processos atua diretamente com o PCMONS e com o Eucalyptus via HybridFox, e também o Eucalyptus se comunica com o PCMONS, passando informações das MVs para que o

PCMONS possa configurar o Nagios a fim de mostrar os resultados do monitoramento para o gerenciador de processos que por fim passa as informações para o usuário via o sistema web.

#### 4.6 Definição de métricas

Antes de concluir quais métricas temos que usar, é necessário entender qual o propósito e quais as recomendações que temos hoje em dia, e feito isso, adaptar para a realidade do projeto.

As métricas estão para dar suporte ao acordo onde podem ser definidos como metas a serem cumpridas para obter um controle do serviço e dando garantia. Existem métricas comum para tipos de aplicações como por exemplo um gerenciamento de serviço de TI, que teria métricas como tempo médio de resposta, tempo para completar uma tarefa, porcentagem de ligações que desligam o telefone enquanto aguardam atendimento. Agora mais próximo do tema, temos as seguintes métricas que devemos tomar como as principais segundo Strassman em 2011:

- Uptime: O tempo necessário para disponibilizar o serviço caso tenha sido desligado ou interrompido;
- Latency: O tempo de resposta do servidor;
- Disponibilidade: O quanto este serviço fica ativo em termos de tempo;
- Performance: O desempenho esperado pelo serviço adquirido.

Tendo estas informações e sabendo do ambiente em específico criado, as seguintes métricas foram tomadas:

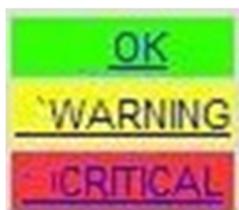
- Numero de máquinas virtuais a serem usadas e suas respectivas configurações;

- Carga média de uso da CPU;
- Estabelecer conexão segura;
- Ping.

A escolha dessas métricas básicas é referente ao desempenho e disponibilidade, no qual para o nosso estudo de caso é suficiente para testar a viabilidade da aplicação prática da implantação de uma nuvem privada. Um dos motivos dessa escolha foi devido a facilidade de gerenciamento tendo em vista ao uso do PCMONS, que por definição já monitora o Ping, que na verdade é utilizado para testar a conectividade das máquinas virtuais, também já monitora o modo de se conectar no sistema que seria a métrica SSH, e também possui o monitoramento da carga média utilizada pelas CPU, que nesta versão os valores são produzidos em intervalos de 1, 5 e 15 minutos. A escolha de máquinas virtuais veio em vista de ser uma métrica para o gerenciador de processos, pois o como o servidor no caso a máquina onde estão instalados todos os controladores tem um limite de recurso, tendo estas informações pelo acordo é possível planejar o que é possível disponibilizar para outro usuário.

Tendo em vista as métricas para possibilidade da aplicação prática, foi necessário entender o sistema de monitoramento do Nagios para fazer as mudanças de acordo com o pedido do usuário. O Nagios possui três status de monitoramento que são demonstrados na figura 21

Figura 21- Status Nagios



Com essa base de monitoramento é possível configurar as requisições do usuário tendo em vista que CRITICAL seria a forma de quebra do acordo, OK seria que está de acordo, e

WARNING também estaria de acordo, mas que se a métrica específica não tiver uma atenção especial pode ser que ela passe para o estado crítico e conseqüentemente ocorra a quebra de acordo de serviço.

No caso temos a métrica Carga Média onde no PCMONS já é definida, e para ser atrelada ao requisito do usuário, o gerenciador de processos deve mudar os parâmetros que assim foram definidos no acordo, como mostra a figura 22.

Figura 22- Status de carga média

```
def process_notification(self,data, ipAddress):
    print 'data : ', data
    instance_id = self.db.get_id_by_ip(ipAddress)
    vm = self.db.get_vm_info(instance_id['instance_id'])
    hostname = "%s_%s_%s"%(vm['user'],vm['instance_id'],vm
['node_hostname'])

    memPercentage = int( 100*(data['memory']['used']/float
(data['memory']['total']))
    self.register_passive_check(hostname, 'RAM', 0, '''+str
(memPercentage) + ';' +str((data['memory']['used']/1024))+ '/' +str
((data['memory']['total']/1024))+''')

    #CPU
    if data['loadavg']['1'] >= X:
        output = "CRITICAL - load average: %.2f, %.2f, %.2f"
% (data['loadavg']['1'],data['loadavg']['5'],data['loadavg']
['10'])
    elif data['loadavg']['1'] >= Y:
        output = "WARNING - load average: %.2f, %.2f, %.2f"
% (data['loadavg']['1'],data['loadavg']['5'],data['loadavg']
['10'])
    else:

        print 'data: ', data['loadavg']['1']
        output = "OK - load average: %.2f, %.2f, %.2f" %
(data['loadavg']['1'],data['loadavg']['5'],data['loadavg']
['10'])

    self.register_passive_check(hostname, 'LOADAVG', 0,
'''+output+''')
```

Neste trecho de código do arquivo NagiosPassiveServer.py do PCMONS, o X indica o valor de limite no qual o usuário definiu no acordo de serviço e Y seria um valor próximo a X só para indicar que está quase quebrando o acordo.

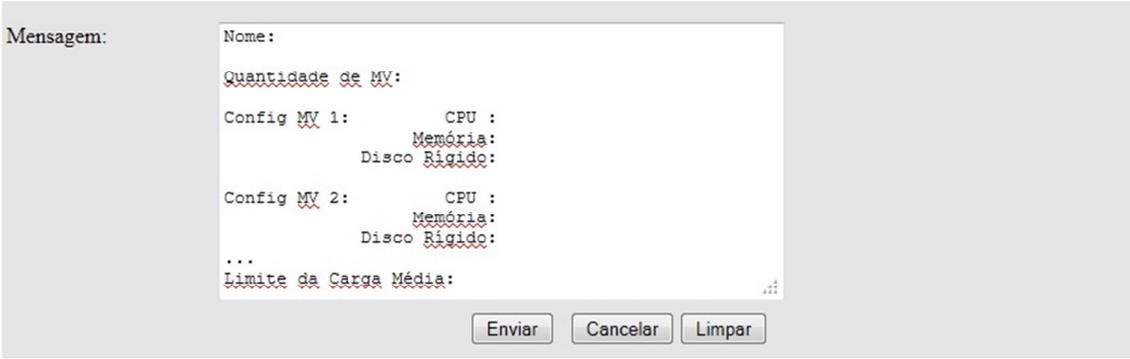
Para a métrica de número de máquinas virtuais e as suas configurações, o usuário posta o que necessita para o seu uso, no caso quantas máquinas necessita e a sua configuração, no caso seria a quantidade de CPUs, quantidade de memória e quantidade de disco rígido, onde o gerenciador de projeto irá atribuir esses valores, conforme mostrado na figura 19.

#### 4.7 Sistema Web

A construção do sistema foi feita usando o NetBeans 7.0, e também é usado o servidor Apache Tomcat 7.0.11. O sistema funciona da seguinte forma, como o ambiente é de uma nuvem privada, para cada usuário que vem a utilizar o serviço é fornecido um login e uma senha, tendo se autenticado o sistema é composto por duas camadas que são, a rede onde está a sua nuvem privada, e uma camada inferior onde diz qual o nome do servidor que está oferecendo seus recursos, isso feito para caso de ter mais nuvens privadas no mesmo ambiente.

A comunicação entre o usuário e o gerenciador de processos é feito de forma que o usuário define as suas necessidades postando no sistema através do botão escrever e isso deve ser feito através de um modelo mostrado na figura 23, e o gerenciador posta resposta relevante ao dia do que aconteceu em relação dos seus serviços prestados.

Figura 23- Modelo definição de métricas



```
Mensagem:
Nome:
Quantidade de MV:
Config MV 1:      CPU :
                  Memória:
                  Disco Rígido:
Config MV 2:      CPU :
                  Memória:
                  Disco Rígido:
...
Limite da Carga Média:
Enviar  Cancelar  Limpar
```

## 4.8 Testes e Resultados

Para testarmos o funcionamento do projeto e por fim obter os resultados, as etapas do processo foram feitas da seguinte forma:

1. Foi criado um usuário com login = bruno e senha = 1234, e o gerenciador de processos utiliza o sistema usando login de administrador;
2. Após o usuário ter tido acesso ao sistema, foi escolhida a sessão LRG, depois a sessão CIRRUS;
3. Postou a requisição de serviço via modelo de definições de métricas como mostrado na figura 23;
4. Enviou a requisição conforme é mostrado na figura 24;

Figura 24- Envio de requisição

Perfil  
19:28:44 - 23 de Maio de 2011

Home Pesquisar Membros

Logout

Registro Bem vindo bruno

Usuários registrados: 3

RESPONDER

CIRRUS

**Autor**

**bruno**  
Foto

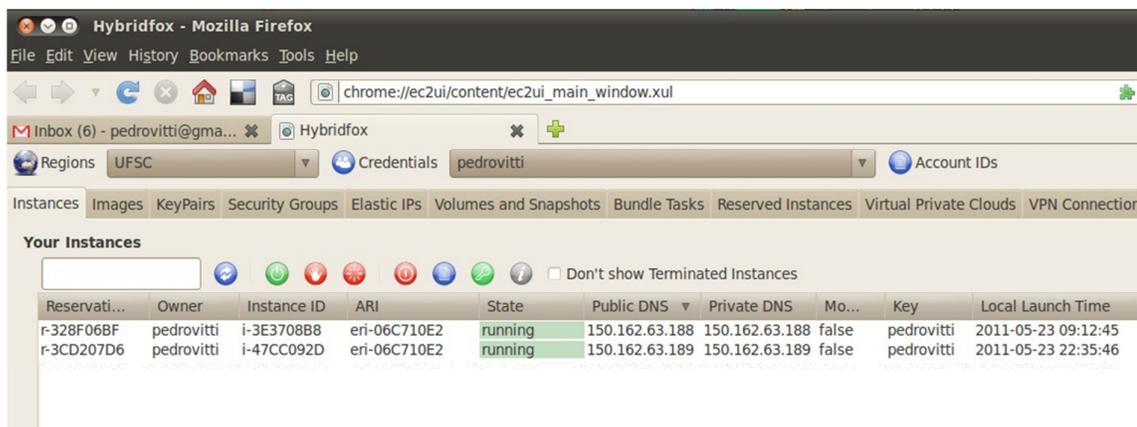
Nome: Bruno Ortale Gonçalves Quantidade de MV: 2 Config MV 1: CPU: 1 Memória: 128MB Disco Rígido: 10GB  
Config MV 2: CPU: 2 Memória: 512MB Disco Rígido: 30GB Limite da Carga Média: 3.2

[Voltar ao topo](#)

RESPONDER

5. Gerenciador de processos recebe a solicitação mostrada no sistema e muda a configuração no PCMONS como mostrado na figura 21 com o valor de  $X = 3.2$  e  $Y = 1.7$ ;
6. Gerenciador de processos atribui os tipos de máquinas virtuais que é mostrado na figura 19 conforme a configuração exigida pelo usuário;
7. Usando a ferramenta HybridFox, o gerenciador instância as duas máquinas virtuais conforme o acordo, as duas máquinas virtuais geradas estão usando o sistema operacional Debian 5.0 (figura 25).

Figura 25- MV rodando



8. Feito isso a interface de monitoramento Nagios mostra o relatório como na figura 26;

Figura 25- Resultado Nagios

The screenshot shows the Nagios monitoring interface with a table of system metrics. All metrics are in an "OK" state, indicated by green bars. The metrics include CPU Load, HTTP Connections, PING, RAM, and SSH.

Cpu Load	OK	23-05-2014 01:12:23	0d 1h 47m 34s	1/4	OK - load average: 0.00, 0.00, 0.00
HTTP Connections	OK	23-05-2014 01:12:23	0d 1h 47m 34s	1/4	8
PING	OK	23-05-2014 01:11:45	0d 1h 46m 29s	1/4	PING OK - Packet loss = 0%, RTA = 0.15 ms
RAM	OK	23-05-2014 01:12:23	0d 1h 47m 34s	1/4	30/39127
SSH	OK	23-05-2014 01:08:32	0d 1h 44m 42s	1/4	SSH OK - OpenSSH_5.1p1 Debian-5ubuntu1 (protocol 2.0)

9. O gerenciador usa o sistema web para postar os resultados obtidos do monitoramento, demonstrado pela figura 27.

Figura 27- Resultado web

[Home](#)   [Pesquisar](#)   [Membros](#)

[Perfil](#)  
22:58:44 - 23 de Maio de 2011

[Registro](#) Bem vindo

administrador

Usuários registrados: 3

 **RESPONDER**

CIRRUS

**Autor**

**bruno**  
Foto

Nome: Bruno Ortale Gonçalves Quantidade de MV: 2 Config MV 1: CPU: 1 Memória: 128MB Disco Rígido: 10GB  
Config MV 2: CPU: 2 Memória: 512MB Disco Rígido: 30GB Limite da Carga Média: 3.2

[Voltar ao topo](#)

**administrador**  
Foto

Acordo de serviço com Bruno Ortale Gonçalves cumprido: 2 MV running | Status de todas as métricas = OK

[Voltar ao topo](#)

 **RESPONDER**

10. Por último o gerenciador toma as devidas atitudes que combinou com usuário dependendo do resultado obtido.

## 4.9 Conclusão

O processo de desenvolvimento apresentado demonstra uma maneira simples de garantir a viabilidade e confiabilidade de uma implantação e gerenciamento de uma nuvem privada que pode ser modificada através de métricas simples assim definidas, e isso é feito também

utilizando um sistema web que dá o suporte para a interação entre o gerenciador de processos e o usuário.

O resultado mostra as ações e os papéis de cada elemento dentro deste ambiente que tem a finalidade de ser um exemplo mesmo que pequeno, de como deve ser um processo de gerenciamento de uma nuvem privada que deseja proporcionar um relacionamento confiável e obter um controle do serviço oferecido.

## 5 CONSIDERAÇÕES FINAIS

Depois de mostrado os principais conceitos, as principais ferramentas e tecnologias, o processo de desenvolvimento para uma solução de gerenciamento em computação em nuvem privada, podemos dizer que é já um passo para incrementar este novo paradigma, que há ainda muito que fazer.

A computação em nuvem vai crescer isto é certo e como estamos no começo dessa nova era, podemos concluir que esses pequenos passos que estamos tomando, vão contribuir e muito lá na frente além de agregar um conhecimento que este vai servir para estar adiantado na hora que está mudança se concluir.

Com isso este trabalho ajuda no processo de migração para as nuvens e com isso facilitando e ajudando com o crescimento tecnológico para o futuro.

Em um mundo globalizado e cheio de mudanças em que vivemos hoje, não se pode mais ficar parado, é preciso estar em constante adaptação para mantermos competitivos e oferecendo serviços e bens de qualidade, e quanto antes adaptarmos mais bem preparados estaremos.

### 5.1 Trabalhos Futuros

Para trabalhos futuros, pretende-se automatizar os processos do modelo, onde ao usar o sistema web não seja necessário ter um gerenciador de processos para dar continuidade no projeto. Outra abordagem seria dar suporte a outras ferramentas de plataforma assim não restringindo somente aos softwares usados neste projeto.

Por fim seria expandir o conceito do modelo, podendo ser mais geral e satisfazer qualquer implantação de uma nuvem privada.

## REFERÊNCIAS BIBLIOGRÁFICAS

BUYYA R. , Yeo C.S. , Venugopal S. , Broberg, J. , Brandic, I , "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility" em Future Generation Computer Systems, 2009.

BUYYA, R.; YEO, C. S.; VENUGOPAL, S. Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, Dalian, China, setembro 2008.

CHANTRY, Darryl. Mapping Applications to the Cloud. janeiro de 2009. Disponível em: <<http://msdn.microsoft.com/en-us/library/dd430340.aspx>>. Acesso em: 18 maio 2011.

CHAVES, S. A. ; URIARTE, R. B. ; WESTPHALL, C. B. . Implantando e Monitorando uma Nuvem Privada. In: VIII Workshop em Clouds, Grids e Aplicações - Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2010, Gramado. VIII Workshop em Clouds, Grids e Aplicações. Porto Alegre : SBC, 2010. v. 1. p. 31-42. Link: [sbrc2010.inf.ufrgs.br/anais/data/pdf/wcga/st01\\_03\\_wcga.pdf](http://sbrc2010.inf.ufrgs.br/anais/data/pdf/wcga/st01_03_wcga.pdf)

DISPLAYSEARCH. Mini-Note PC (Netbook) Shipments Grow at Twice the Rate of Notebook PCs in Q2'09. Disponível em: <[http://www.displaysearch.com/cps/rde/xchg/displaysearch/hs.xsl/090831\\_mini\\_note\\_pc\\_netbook\\_shipments\\_grow\\_at\\_twice\\_rate\\_notebook\\_pcs\\_q2\\_09.asp](http://www.displaysearch.com/cps/rde/xchg/displaysearch/hs.xsl/090831_mini_note_pc_netbook_shipments_grow_at_twice_rate_notebook_pcs_q2_09.asp)>. Acesso em: 18 maio 2011.

DIKAIKOS, M. D.; PALLIS, G.; KATSAROS, D.; MEHRA, P.; VAKALI, A. Cloud Computing – Distributed Internet Computing for IT and Scientific Research. IEEE Internet Computing, 13(5): 10-13, outubro 2009.

EUCALYPTUS Network Configuration Disponível em: <[http://open.eucalyptus.com/wiki/EucalyptusNetworkConfiguration\\_v2.0](http://open.eucalyptus.com/wiki/EucalyptusNetworkConfiguration_v2.0)>. Acesso em: 14 maio 2011.

GENS, Frank. IT Cloud Services User Survey, pt.2: Top Benefits & Challenges. Disponível em: <<http://blogs.idc.com/ie/?p=210>>. Acesso em: 18 maio 2011.

HYBRIDFOX: Elasticfox for Eucalyptus Disponível em: <<http://architects.dzone.com/news/hybridfox-elasticfox>>. Acesso em: 18 maio 2011.

KAUFMAN, L. M. Data Security in the World of Cloud Computing. IEEE Security and Privacy, 7(4): 61-64, julho/agosto 2009.

KREUTZ, Diego. Cloud Computing. Disponível em: <[http://eventos.unipampa.edu.br/tchelinix2010/files/2010/12/TcheLinux2010\\_DiegoKreutz\\_Cloud\\_Computing.pdf](http://eventos.unipampa.edu.br/tchelinix2010/files/2010/12/TcheLinux2010_DiegoKreutz_Cloud_Computing.pdf)>. Acesso em: 18 maio 2011.

LYNCH, Merrill (2008), "The Cloud Wars: \$100+ billion at stake".

MOHAN T. S. CLOUD COMPUTING Principles and Paradigms. New Jersey: Wiley, 2011. Migrating into a Cloud.

MOTA, Wélton Costa da; ZANOTTO, Mariana Susan; RUSCHEL, Henrique. Computação em Nuvem. Disponível em: <<http://www.ppgia.pucpr.br/~jamhour/RSS/TCCRSS08B/Welton%20Costa%20da%20Mota%20-%20Artigo.pdf>>. Acesso em: 18 maio 2011.

MORROW Susan. CLOUD COMPUTING Principles and Paradigms. New Jersey: Wiley, 2011. Data Security in the Cloud.

NAGIOS Disponível em: <<http://www.nagios.org/>>. Acesso em: 18 maio 2011.

NIST. The NIST Definition of Cloud Computing (Draft). janeiro de 2011. Disponível em: <[http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145\\_cloud-definition.pdf](http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf)>. Acesso em: 18 maio 2011.

NURMI, D.; WOLSKI, R.; GRZEGORCZYK, C.; OBERTELLI, G.; SOMAN, S.; YOUSEFF, L.; ZAGORODNOV, D. The Eucalyptus Open-Source Cloud Computing System. CCGRID-09, p. 124-131, maio 2009.

OCCI Disponível em: <<http://occi-wg.org/about/>>. Acesso em: 18 maio 2011.

RANDYBIAS. Cloud Standards are Misunderstood. Disponível em: <<http://cloudscaling.com/blog/cloud-computing/cloud-standards-are-misunderstood>>. Acesso em: 18 maio 2011.

SOUZA FILHO, F. Cloud Computing totalmente nacional. PC Magazine, setembro 2009.  
Douglas Parkhill's 1966 book, *The Challenge of the Computer Utility*.

STRASSMAN, Paul. HOW to Read Service Level Agreements Disponível em: <<http://pstrassmann.blogspot.com/2011/04/how-to-read-service-level-agreements.html>>. Acesso em: 15 maio 2011.

SUN MICROSYSTEMS, INC. Take your Business to a Higher Level. Sun Cloud Computing, março 2009.

SUN MICROSYSTEMS, INC. Introduction to Cloud Computing Architecture. White Paper, 1ª edição, junho 2009.

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

TETZCHNER. State of the Mobile Web, December 2010. 2010. Disponível em: <<http://www.opera.com/smw/2010/12/>>. Acesso em: 18 maio 2011.

URIARTE, Rafael Bruno; APARECIDA, Shirlei. Private Clouds MONitoring Systems. Disponível em: <<http://code.google.com/p/pcmons/>>. Acesso em: 20 maio 2011.

VAQUERO L.M. , Rodero-Merino L. , Caceres, J. , Lindner, M. , "A break in the clouds: towards a cloud definition" em ACM SIGCOMM Computer Communication Review, 2008

VOORSLUYS, William ; BUYYA, Rajkumar; BROBERG, James. CLOUD COMPUTING Principles and Paradigms. New Jersey: Wiley, 2011. Introduction to the Cloud.