



UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

Julio Alba Soto

**OpenNebula: implantação de uma nuvem privada e
orquestração das máquinas virtuais no paradigma
da Computação em Nuvem**

FORTALEZA – CEARÁ
MAIO 2011

Autor:

Julio Alba Soto

Orientador:

Prof. Dr. Danielo Gonçalves Gomes

OpenNebula: implantação de uma nuvem privada e orquestração das máquinas virtuais no paradigma da Computação em Nuvem

Monografia de Conclusão de Curso apresentada à Coordenação do Curso de Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará como parte dos requisitos para obtenção do grau de **Engenheiro de Teleinformática.**

FORTALEZA – CEARÁ

MAIO 2011

JULIO ALBA SOTO

**OpenNebula: implantação de uma nuvem privada e orquestração das
máquinas virtuais no paradigma da Computação em Nuvem**

Esta Monografia foi julgada adequada para a obtenção do diploma de Engenheiro do Curso de Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará.

Julio Alba Soto

Banca Examinadora:

Prof. Dr. Danielo Gonçalves Gomes
Universidade Federal do Ceará - UFC

Prof. Alexandre Augusto da Penha Coelho
Universidade Federal do Ceará - UFC

Prof. Augusto José Venâncio Neto

Fortaleza, 2011

Resumo

A popularização da Internet na década de 1990 e o rápido crescimento da produção e comercialização de dispositivos computacionais móveis foram fatores determinantes para a formação do panorama tecnológico observado na atualidade. À medida que a conectividade dos indivíduos aumentava, uma poderosa plataforma de serviços surgia inteiramente on-line. Deste modo, formava-se um ambiente propício ao surgimento da Computação em Nuvem. Computação em Nuvem é um modelo de computação atraente, pois permite que os recursos sejam fornecidos de acordo com a demanda, ou seja, os usuários podem alugar os recursos da nuvem que forem necessários. Um desafio considerável a cerca da nuvem é relativo à necessidade de padronização. Este modelo tem motivado varias instituições acadêmicas e não-acadêmicas para desenvolver soluções de código aberto da nuvem. Este trabalho apresenta as soluções de código aberto mais populares para Computação em Nuvem. A atividade de pesquisa tem como foco a análise da ferramenta de código aberto OpenNebula, que busca oferecer meios para a implementação de ambientes distribuídos com suporte à virtualização e baseado no paradigma da Computação em Nuvem. Após essa etapa inicial, a ferramenta é utilizada para a implementação de uma nuvem privada e a orquestração das máquinas virtuais.

Palavras-chaves: Computação em Nuvem, Sistemas Distribuídos, Virtualização, Ferramentas de código aberto, OpenNebula.

Abstract

Since 1990, the Internet popularization and the fast growth of production and distribution of mobile computing devices were key factors for the developed technologic situation obtained nowadays. While users' connectivity was growing, a powerful services platform was arising totally on-line. By this way, an appropriated environment was formed due cloud computing emergence. Cloud Computing is an attractive computation model, it allows that resources being provided according to the demand, i.e., the users can rent the cloud resources that they need. An important challenge around the cloud is relative to standardization. This model has been motivated some academic and non-academic institutions to develop cloud open source solutions. This work presents the most popular Cloud Computing open source solutions. The research has focus in the analysis of OpenNebula open source tool that seek to offer ways for the development of distributed environment with support to virtualization and based on cloud computing paradigm. After this, the tool is used for the development of a private cloud and the orchestration of virtual machines.

Keywords: Cloud Computing, Distributed Systems, Virtualization, Open-source tools, OpenNebula.

Dedico este trabalho à minha família e aos meus amigos, a quem tudo devo.

Agradecimentos

Agradeço primeiramente aos meus pais Luis e Encarnita, pelos valores ensinados, pela paciência e pelo apoio incondicional durante os longos anos de graduação.

Aos meus irmãos Juan e Luis, pela amizade e companheirismo.

À minha segunda família, Jose, Joni, Víctor e Mateo, por todo o apoio durante o ano de convívio em Fortaleza.

Ao professor Danielo Gonçalves Gomes, meu orientador, pela confiança e pelo apoio imprescindível durante a realização do trabalho.

Ao Thiago Teixeira Sá, pela paciência e esforço compartilhado auxiliando na orientação deste trabalho.

À Universidade Federal do Ceará, por proporcionar ensinamentos de altíssima qualidade.

Aos amigos do curso de graduação em Engenharia de Teleinformática, especialmente a Raymundo.

Dêem-me um ponto de apoio e moverei a terra.

Arquimedes

Sumário

| | |
|--|-------------|
| Lista de Figuras | viii |
| Lista de Tabelas | ix |
| Lista de Siglas | ix |
| 1 Introdução | 1 |
| 1.1 Motivação | 2 |
| 1.2 Objetivos | 2 |
| 1.3 Organização da monografia | 2 |
| 2 Computação em Nuvem | 3 |
| 2.1 Introdução | 3 |
| 2.2 Contexto Histórico | 7 |
| 2.3 Arquitetura | 9 |
| 2.3.1 Sistemas distribuídos | 9 |
| 2.3.2 Middleware | 11 |
| 2.3.3 Grades computacionais | 11 |
| 2.3.4 Computação utilitária | 12 |
| 2.3.5 Virtualização | 13 |
| 2.3.6 Web services | 14 |
| 2.3.7 Computação verde | 15 |
| 2.4 Modelos de serviço | 15 |
| 2.4.1 Software como serviço (SaaS) | 16 |
| 2.4.2 Plataforma como serviço (PaaS) | 17 |
| 2.4.3 Infraestrutura como serviço (IaaS) | 18 |
| 2.5 Modelos de implementação | 18 |
| 2.5.1 Nuvem pública | 18 |
| 2.5.2 Nuvem privada | 19 |
| 2.5.3 Nuvem híbrida | 19 |
| 2.6 Esforços de normalização | 19 |
| 2.7 OpenNebula | 22 |
| 2.7.1 Introdução | 22 |

| | | |
|----------|---|-----------|
| 2.7.2 | História | 23 |
| 2.7.3 | Arquitetura OpenNebula | 24 |
| 2.7.4 | Objetivos | 26 |
| 2.8 | Outras soluções de código aberto | 27 |
| 2.8.1 | Eucalyptus | 27 |
| 2.8.2 | OpenQRM | 28 |
| 2.8.3 | Nimbus | 29 |
| 2.8.4 | TPlataform | 30 |
| 2.8.5 | <i>Xen Cloud Plataform (XCP)</i> | 30 |
| 2.8.6 | Resumo | 31 |
| 3 | Análise da ferramenta OpenNebula | 33 |
| 3.1 | Equipamento utilizado | 33 |
| 3.2 | Criação de nuvem privada | 34 |
| 3.3 | Lançamento de máquina virtual | 35 |
| 3.4 | Comandos | 37 |
| 4 | Resultados | 38 |
| 4.1 | Resultados da criação de uma nuvem privada | 38 |
| 4.2 | Resultados do lançamento de uma máquina virtual | 40 |
| 5 | Conclusões | 43 |
| 5.1 | Considerações finais | 43 |
| 5.2 | Perspectivas para trabalhos futuros | 44 |
| | Referências Bibliográficas | 47 |

Lista de Figuras

| | | |
|------|---|----|
| 2.1 | Diagrama de rede (COULOURIS et al., 2005) | 4 |
| 2.2 | Arquitetura Von Neumann (Adaptado de BREY, 2008) | 7 |
| 2.3 | Sistema distribuído típico (Adaptada de TANENBAUM, 2006) | 10 |
| 2.4 | Modelos de serviço da Computação em Nuvem (Adaptada de STANOEVSKA-SLABEVA et al., 2010) | 16 |
| 2.5 | Logotipo projeto OpenNebula (OPENNEBULA, 2011) | 23 |
| 2.6 | Arquitetura padrão de uma nuvem OpenNebula (OPENNEBULA, 2011) | 26 |
| 2.7 | Processos OpenNebula (OPENNEBULA,2011) | 26 |
| 2.8 | Componentes do espaço de trabalho Nimbus (KEAHAY, 2009) | 29 |
| 2.9 | Arquitetura XCP (TAKAKO, 2010) | 31 |
| 2.10 | Soluções Computação em Nuvem (Adaptado de TAKAKO, 2010) | 32 |
| | | |
| 3.1 | Nuvem privada implementada | 34 |
| 3.2 | Máquina virtual (VM1) implementada | 35 |
| 3.3 | Conteúdo padrão do arquivo de texto <i>vmnet.template</i> | 35 |
| 3.4 | Conteúdo padrão do arquivo de texto <i>primeiraVM.template</i> | 36 |
| | | |
| 4.1 | Comando <i>onehost list</i> | 38 |
| 4.2 | Comando <i>onehost show <host_id></i> | 39 |
| 4.3 | Comando <i>onehost disable <host_id></i> | 40 |
| 4.4 | Comando <i>onehost enable <host_id></i> | 40 |
| 4.5 | Comando <i>onehost delete <host_id></i> | 40 |
| 4.6 | Comando <i>onevnet list</i> | 40 |
| 4.7 | Comando <i>onevnet show <network_id></i> | 41 |
| 4.8 | Comando <i>onevnet delete <network_id></i> | 41 |
| 4.9 | Comando <i>onevm list</i> | 41 |
| 4.10 | Comando <i>onevm delete <vm_id></i> | 42 |

Lista de Tabelas

| | | |
|-----|---|----|
| 2.1 | Evolução projeto OpenNebula (OPENNEBULA, 2011) | 25 |
| 2.2 | Comparação das soluções de código aberto da Computação em Nuvem | 32 |
| 3.1 | Equipamento utilizado no desenvolvimento do trabalho | 33 |
| 4.1 | Estados das máquinas virtuais | 42 |

Lista de Siglas

- PDA** *Personals Digital Assistant* (Assistentes Digitais Pessoais)
- 3G** *3rd Generation mobile telecommunications* (Terceira Geração)
- API** *Application Programming Interface* (Interface de Programação de Aplicativos)
- IaaS** *Infrastructure as a Service* (Infraestrutura como Serviço)
- DMTF** *Distributed Management Task Force*
- SLA** *Service Level Agreement* (Acordos de Nível de Serviço)
- VCL** *Virtual Computing Lab*
- PaaS** *Platform as a Service* (Plataforma como Serviço)
- AWS** *Amazon Web Service* (Serviço Web Amazon)
- SaaS** *Software as a Service* (Software como Serviço)
- NIST** *National Institute of Standards and Technology*
- PC** *Personal Computer* (Computador Pessoal)
- VLSI** *Very Large Scale Integration* (Integração em Escala Muito Alta)
- IT** *Information Technology* (Tecnologia da Informação)
- XML** *Extensible Markup Language* (Linguagem de Marcação Extensível)
- SOA** *Service-Oriented Architecture* (Arquitetura Orientada a Serviços)
- CaaS** *Communication as a Service* (Comunicação como Serviço)
- MaaS** *Management as a Service* (Gerenciamento como Serviço)
- DaaS** *Data as a Service* (Dados como Serviço)
- ITaaS** *Information Technology as a Service* (Tecnologia da Informação como Serviço)

-
- XaaS** *Everything as a Service* (Tudo como Serviço)
- EC2** *Elastic Compute Cloud*
- SOAP** *Simple Object Access Protocol* (Protocolo Simples de Acesso a Objetos)
- REST** *Representational State Transfer* (Transferência de Estado Representacional)
- CSA** *Cloud Security Alliance*
- OVF** *Open Virtualization Format* (Formato Aberto de Virtualização)
- OCC** *Open Cloud Consortium*
- OASIS** *Organization for the Advancement of Structures Information Standards*
- VM** *Virtual Machine* (Máquina Virtual)
- VWS** *Virtual Workspace Service*
- IP** *Internet Protocol* (Protocolo de Internet)
- XCP** *Xen Cloud Platform*
- TFS** *Tianwang File System* (Sistema de Arquivos de Tianwang)
- GFS** *Google File System* (Sistema de Arquivos de Google)
- LAN** *Local Area Network* (Rede de Área Local)

Capítulo 1

Introdução

A Computação em Nuvem está se tornando um conceito importante no mundo da *Information Technology* (Tecnologia da Informação) (IT). Existem diferentes formas de usar a Computação em Nuvem. É possível usar esta tecnologia para mudar uma aplicação crítica de negócio e que ela seja executada em uma nuvem pública. Nesta configuração só se paga pelo que se usa. Existem muitos fornecedores de Computação em Nuvem que implementam este tipo de serviço. É chamado *Software as a Service* (Software como Serviço) (SaaS). Além do software, também é possível criar uma infraestrutura de rede baseada em uma nuvem *Infrastructure as a Service* (Infraestrutura como Serviço) (IaaS). Esta forma de computação é mais eficiente do que a criação tradicional de um datacenter. Além disso, Computação em Nuvem pode ser usada para fazer o datacenter mais "verde" e criar servidores redundantes com mais facilidade.

O campo da Computação em Nuvem tem recebido grande atenção da academia desde sua eclosão. Devido a seu caráter recente, muitas informações conflitantes são freqüentemente encontradas na literatura. Contudo, o conceito é tratado no mundo acadêmico de forma próxima à unanimidade devido ao seu grande potencial e o papel chave que certamente desempenhará em um futuro próximo em diversas áreas da tecnologia.

Uma característica comum presente em todas as nuvens é que os fornecedores compreensivelmente evitam revelar as características. São muitos os esforços feitos para desenvolver uma solução de código aberto. A observação e a classificação dessas soluções podem alavancar mais ainda a área de pesquisa da Computação em Nuvem

fornecendo um bom ponto de partida para descobrir diferentes abordagens para os problemas presentes nos ambientes de Computação em Nuvem.

Uma das soluções de código aberto mais populares é a ferramenta OpenNebula, que se encaixa nos atuais ambientes de datacenter para construir qualquer tipo de implantação IaaS na nuvem. Alguns benefícios que podem ser obtidos com a utilização de um ambiente em nuvem OpenNebula são: respostas mais rápidas às necessidades de infraestrutura, gerenciamento centralizado, melhor utilização dos recursos existentes, ganhos operacionais como redução de custos com espaço e energia, suporte a ambientes de execução heterogêneos, implementação em qualquer datacenter com suporte à virtualização, etc.

1.1 Motivação

A principal motivação deste trabalho é demonstrar a viabilidade de implantação e configuração de OpenNebula.

1.2 Objetivos

Esse trabalho objetiva analisar a ferramenta de código aberto OpenNebula através da configuração de ambientes para a criação de uma nuvem privada e a orquestração das máquinas virtuais no paradigma da Computação em Nuvem.

1.3 Organização da monografia

Após a Introdução, o trabalho se divide em mais quatro capítulos. O Capítulo 2 aborda os conceitos gerais que definem o novo paradigma de computação distribuída denominado Computação em Nuvem e constitui o embasamento teórico deste trabalho. No Capítulo 3 será fundamentada toda a metodologia utilizada, ou seja, a criação de uma nuvem privada e a orquestração das máquinas virtuais mediante o uso da ferramenta OpenNebula. O Capítulo 4 analisa os resultados obtidos. Enquanto o Capítulo 5 apresenta considerações finais acerca do trabalho realizado.

Capítulo 2

Computação em Nuvem

Nesse capítulo, serão abordados aspectos que definem o campo de pesquisa e negócios hoje intitulado Computação em Nuvem. As Seções 2.1 e 2.2 trazem definições e uma contextualização histórica, enquanto a Seção 2.3 analisa os elementos que compõem a arquitetura deste modelo. Em seguida, são abordados os principais modelos de serviço e os modos de implantação existentes nas Seções 2.4 e 2.5. Enquanto a Seção 2.6 discute os esforços de normalização na área da Computação em Nuvem. As seções finais analisam as soluções de código aberto atualmente disponíveis no contexto da Computação em Nuvem.

2.1 Introdução

O grande crescimento nos últimos anos da conectividade de usuários (laptops, palmtops, *Personals Digital Assistant* (Assistentes Digitais Pessoais) (PDA), celulares *3rd Generation mobile telecommunications* (Terceira Geração) (3G), etc.) em todo o mundo atrelado à evolução da rede mundial de computadores de um simples meio de comunicação a uma plataforma real de negócios e redes sociais, tornou necessária a criação de ambientes computacionais capazes de gerenciar dinamicamente uma demanda exponencial por dados e serviços on-line, e um acesso banda larga à Internet a qualquer hora, de qualquer lugar. Precisava-se de um novo campo de pesquisa e negócios, de uma nova visão para discutir possíveis soluções para esta problemática, hoje conhecido como: Computação em Nuvem (em inglês, Cloud Computing).

O termo nuvem ou “cloud” é uma metáfora em relação à forma como a Internet é

usualmente mostrada nos diagramas de rede, como uma nuvem, como pode ser visto na Figura 2.1. Nesses diagramas, o ícone da nuvem representa todas as tecnologias que fazem a Internet funcionar, abstraindo a infraestrutura e a complexidade que ela engloba. No caso de Computação em Nuvem, refere-se de modo geral a uma combinação de tecnologias e paradigmas (virtualização, computação utilitária, computação em grade, arquitetura orientada a serviços, entre outros) arquitetados e utilizados de uma forma na qual se abstrai os detalhes do usuário, o qual não precisa ter todo o conhecimento necessário para manter essa infraestrutura. Utilizar serviços dessa nuvem (armazenamento, banco de dados, processamento, entre outros) é fazer Computação em Nuvem.

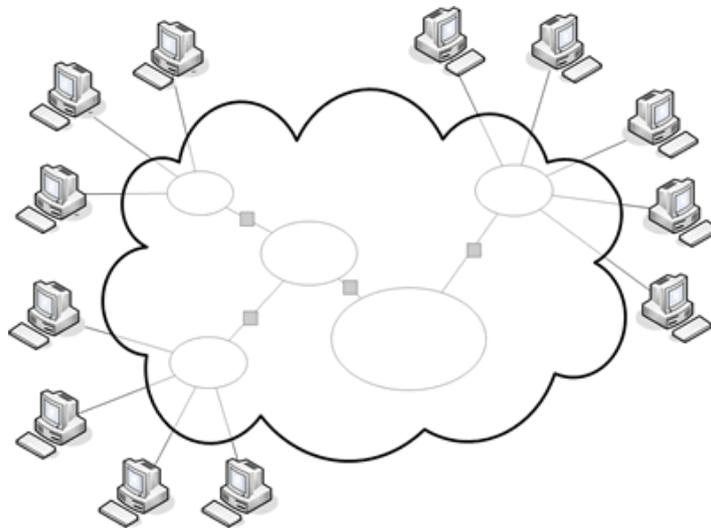


Figura 2.1: Diagrama de rede (COULOURIS et al., 2005)

A Computação em Nuvem é um conceito relativamente novo, a origem do termo é controversa. De acordo com (1), o emprego técnico do termo “nuvem” data dos anos 1990 em referência à capacidade de comutação dinâmica de circuitos para o balanceamento de utilização das redes de telecomunicações, o que reiterava seu caráter virtualizado. No ano de 2001, segundo matéria publicada pelo jornal americano *The New York Times*, a Microsoft utilizou o termo durante uma apresentação sobre o .Net Framework para designar o conjunto de computadores que constituem a Internet. Todos os profissionais falam do conceito Computação em Nuvem e, as vezes, há mal entendidos. Existem tantas definições como autores têm tentado definir na literatura. Uma parte do problema de compreender o que é a Computação em Nuvem é, justamente, que não têm definição concreta.

Segundo o *National Institute of Standards and Technology* (NIST), a Computação

em Nuvem é “um modelo que possibilita o acesso via rede de forma conveniente e sob demanda a um conjunto compartilhado de recursos computacionais (como redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente providos e liberados com interação com o provedor de serviços e esforços de gerenciamento mínimos. O modelo promove a disponibilidade e é composto por cinco características essenciais, três modelos de serviço, e quatro modos de implantação” (2). Discutiremos os modelos de serviço e os modos de implantação da Computação em Nuvem nas seções 2.4 e 2.5 respectivamente. As cinco características essenciais são descritas a seguir:

Atendimento self-service e sob demanda: O usuário pode alocar recursos computacionais unilateralmente sem a necessidade de interação humana com o provedor do serviço. Empresas que atuam na área de IaaS oferecem uma *Application Programming Interface* (Interface de Programação de Aplicativos) (API) própria, que pode ser utilizada para a requisição dinâmica de recursos através de scripts personalizados. A Computação em Nuvem não é mais que um modelo de prestação de serviços. Neste tipo de computação, tudo o que pode oferecer um sistema de computação é fornecido como um serviço.

Amplo acesso à rede: Infraestrutura computacional, plataformas de desenvolvimento e aplicações são acessadas via rede através de protocolos padrão. Isto possibilita a utilização dos serviços por máquinas clientes que variam de desktops robustos a dispositivos móveis com severas limitações de recursos. A Computação em Nuvem desenvolve a idéia da internet com aplicações remotas.

Pooling de recursos: O provedor detém um conjunto de recursos físicos e virtuais que são alocados para usuários e liberados pelos mesmos de forma dinâmica e de acordo com a demanda existente. A informação e aplicações já não são dependentes de nossos computadores ou sistema operacional. A alocação de um mesmo recurso é muitas vezes feita para mais de um usuário simultaneamente, prática comumente definida como multi-arrendamento (em inglês, multi-tenancy). Ademais, o usuário não possui controle ou conhecimento preciso do ponto geográfico de origem do serviço que está utilizando.

Elasticidade: Os recursos providos por um ambiente computacional em nuvem

são inerentemente escaláveis. O termo elasticidade é utilizado para transmitir a idéia de que o usuário pode, a qualquer momento, aumentar ou diminuir a quantidade de recursos utilizados. Essa característica cria a ilusão de que os recursos oferecidos são ilimitados e que o usuário pode fazer uso da quantidade que lhe for conveniente. O consumidor do serviço pode requisitar dinamicamente mais ou menos recursos para sua aplicação para se adaptar à demanda dos seus usuários. Por exemplo, em períodos de pico o consumidor solicita à nuvem mais recursos computacionais (como, por exemplo, servidores adicionais), podendo depois liberar tais recursos, quando os mesmos não forem mais necessários.

Medição de serviços: Os recursos oferecidos podem ser monitorados, controlados e reportados ao provedor e ao usuário de forma transparente. Este mecanismo permite que sejam cobrados valores referentes ao grau de utilização dos recursos e estipulados por meio de contrato entre provedor e usuário. Os consumidores pagam aos provedores de serviço de nuvem de acordo com o consumo efetuado (modelo de pagamento pelo uso semelhante a utilidades como energia e gás).

Embora não raro mencionada equivocadamente como uma tecnologia específica, a Computação em Nuvem constitui um ponto de convergência entre modelos tecnológicos diversos. Trata-se de um campo multidisciplinar, onde antigos e alguns novos conceitos, como modelo cliente-servidor, virtualização, middleware, grades computacionais (em inglês, grid computing), sistemas distribuídos, técnicas de armazenamento de dados, segurança e serviços web se unem para a formação de um ambiente computacional complexo, cujo potencial atrai, com igual interesse, o mundo corporativo e a academia. A Computação em Nuvem criou muito interesse nos últimos anos devido à sua facilidade de escalar os seus recursos. É possível aumentar ou diminuir os seus recursos e só paga pelos recursos que você usa. A Computação em Nuvem é uma fase em utility computing, o que significa uma embalagem de recursos de computação, tais como a computação e armazenamento, como um serviço semelhante a um de utilidade pública tradicional (tais como eletricidade, água, gás natural ou rede de telefone).

Por tudo isso, é importante pensar como é que foi a evolução da computação para se tornar o que hoje conhecemos como Computação em Nuvem. Esta pergunta será respondida na próxima seção.

2.2 Contexto Histórico

Como discutimos na seção anterior, a Computação em Nuvem é um estilo de computação “onde os atributos, escalável e flexível, são provisionados como um serviço aos clientes externos através da tecnologia Internet” (3). É importante compreender a evolução dos sistemas de computação até o momento de eclosão da Computação em Nuvem.

A primeira evidência de uma arquitetura computacional foi proposta por Charles Babbage no ano de 1856. O chamado Motor Analítico representa, ainda hoje, a forma mais básica de arquitetura dos sistemas empregados nos computadores digitais modernos (4). Até anos recentes, quando o trabalho de Babbage ainda não havia sido redescoberto, esta arquitetura era atribuída a John Von Neumann, de onde provém a popular expressão “arquitetura Von Neumann” (Figura 2.2).

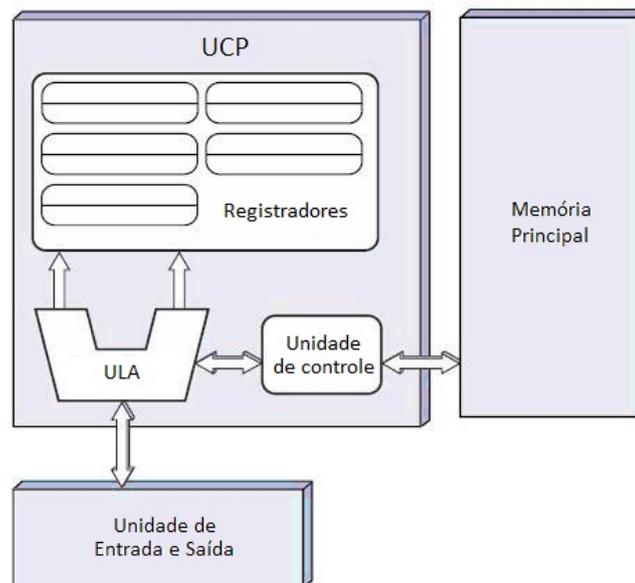


Figura 2.2: Arquitetura Von Neumann (Adaptado de BREY, 2008)

É muito comum na literatura dividir a evolução da computação em gerações. Segundo (3), a primeira geração de computadores remonta ao ano de 1943, quando a IBM financiou a criação do Mark I, um computador programável eletromecânico de propósito geral desenvolvido pela Universidade de Harvard. No mesmo ano, o primeiro dispositivo de computação programável, eletrônico e digital, chamado Colossos, foi concebido na Inglaterra. Era o tempo das válvulas, do armazenamento de dados através de cartões perfurados e de computadores que ocupavam grandes galpões.

A invenção dos transistores marcou o surgimento da segunda geração de computadores. Ainda se tratavam de máquinas grandes e de custo muito elevado, o que limitava sua utilização as universidades e órgãos governamentais.

Com o surgimento dos circuitos integrados, deu-se início à terceira geração de computadores. As máquinas desta geração, chamadas minicomputadores, eram mais baratas e compactas, tornando possível sua utilização por empresas de menor porte.

O advento dos microprocessadores na década de 1970 fez com que surgissem os computadores que caracterizam a quarta geração. O chamado *Personal Computer* (Computador Pessoal) (PC) oferecia maior capacidade de processamento, armazenamento de dados, e tamanho bastante reduzido. A *Very Large Scale Integration* (Integração em Escala Muito Alta) (VLSI) colaborou para a massificação comercial destes dispositivos, tornando-os ainda menores e mais baratos.

A popularização da Internet na década de 1990 e o rápido crescimento da produção e comercialização de dispositivos computacionais móveis foram fatores determinantes para a formação do panorama tecnológico observado na atualidade. À medida que a conectividade dos indivíduos aumentava, uma poderosa plataforma de serviços surgia inteiramente on-line. A tecnologia começava a embutir-se onde antes não se imaginava, concretizando o conceito da Computação Ubíqua, termo empregado pela primeira vez por Mark Weiser ainda na década de 1980.

Deste modo, ao final da década de 1990, formava-se um ambiente propício ao surgimento da Computação em Nuvem. A grande quantidade de serviços web disponíveis e a necessidade de acessá-los através de dispositivos com recursos físicos cada vez mais limitados somaram-se ao avanço das tecnologias de virtualização de hardware para a promoção de uma evolução na forma de lidar com o consumo de recursos computacionais e com o desenvolvimento e utilização de software. Por conseguinte, a Computação em Nuvem, ao contrario do que muitos apregoam, não se trata de uma revolução e também não se trata de uma nova tecnologia, é uma evolução tecnológica natural.

Atualmente, o mercado da Computação em Nuvem tornou-se incrivelmente forte, estima-se "superior a 42.000 milhões de dólares em 2012"(5). Se nos centramos no mercado brasileiro, a Computação em Nuvem vai crescer exponencialmente. O Brasil é o país que mais tem interesse por Computação em Nuvem quando comparado com os demais da América Latina. Até hoje, 18% das médias e grandes empresas brasileiras já utilizam alguma aplicação de Computação em Nuvem. Estima-se que

até 2013, esta fatia deve saltar para 30% a 35%, número que é aproximadamente 60% maior do que a base atual. Nos Estados Unidos, entre 45% e 55% das companhias médias e grandes já utilizam algum serviço de Computação em Nuvem. Na Europa o número está entre 35% e 40% (5).

O interesse da indústria em adotar a Computação em Nuvem se dá principalmente pelo fato deste modelo permitir utilizar os recursos computacionais de forma mais econômica e otimizada, possibilitando uma redução de custos operacionais (menos máquinas, manutenção simplificada, redução de gastos com energia e resfriamento) (6), o que representa um fator extremamente relevante considerando-se o atual cenário de crise econômica mundial e redução de gastos com IT. Esta economia ocorre devido ao fato que empresas clientes das nuvens não precisam investir para adquirir (comprando hardware, coolers, alugando espaço físico) e administrar (contratando pessoal) sua própria infraestrutura. Ao invés disso, estas empresas podem optar por alugar a infraestrutura necessária de outras empresas que possuem enormes datacenters (como Google e Amazon) e, com isso, conseguem atingir economias de escala oferecendo tais serviços a um baixo custo a qualquer cliente.

2.3 Arquitetura

Como falamos na seção 2.1, a Computação em Nuvem constitui um ponto de convergência entre modelos tecnológicos diversos, onde antigos e alguns novos conceitos se unem para a formação do ambiente computacional complexo. As tecnologias que ele envolve é um assunto bastante comentado e estudado no momento, e ainda apresenta muitas questões em aberto, além de uma grande variedade de produtos e serviços oferecidos com essa finalidade. Nesta seção vamos tentar discutir mais um pouco sobre as principais tecnologias e modelos relacionados da Computação em Nuvem.

2.3.1 Sistemas distribuídos

O emprego do termo nuvem remete à utilização de serviços de infraestrutura de hardware e software provindos de um ambiente capaz de oferecer recursos virtualmente ilimitados e de forma "elástica". Embora não fique claro ao usuário final, este ambiente é formado por um conjunto que datacenters que, unidos, tornam possível o pooling de recursos, que, juntamente com a elasticidade, constitui

das principais características da Computação em Nuvem. O aspecto coletivo do funcionamento destes datacenters caracteriza o que é denominado um sistema computacional distribuído.

Um sistema distribuído é definido como um conjunto de máquinas independentes que se apresentam ao usuário como um sistema único e consistente (7). Nos sistemas distribuídos os computadores se comunicam e coordenam suas ações via rede através da troca de mensagens (8). Duas das principais vertentes dos sistemas computacionais distribuídos são a computação em clusters e as grades computacionais. No primeiro, um agrupamento de máquinas com o mesmo sistema operacional é conectado localmente, enquanto o último é formado por máquinas provenientes de diferentes organizações e indivíduos e que, embora integradas como um único sistema, estão inseridas em domínios administrativos diferentes e possuem características de hardware e software nitidamente heterogêneas (7). Falaremos mais profundamente do conceito das grades computacionais na Seção 2.3.3.

A Figura 2.3 ilustra um sistema distribuído típico, no qual a camada de middleware é composta pelo software responsável pelo provimento de uma interface de comunicação única e pela coordenação de usuários e aplicações para a utilização de máquinas distintas, cada uma com seu próprio sistema operacional.

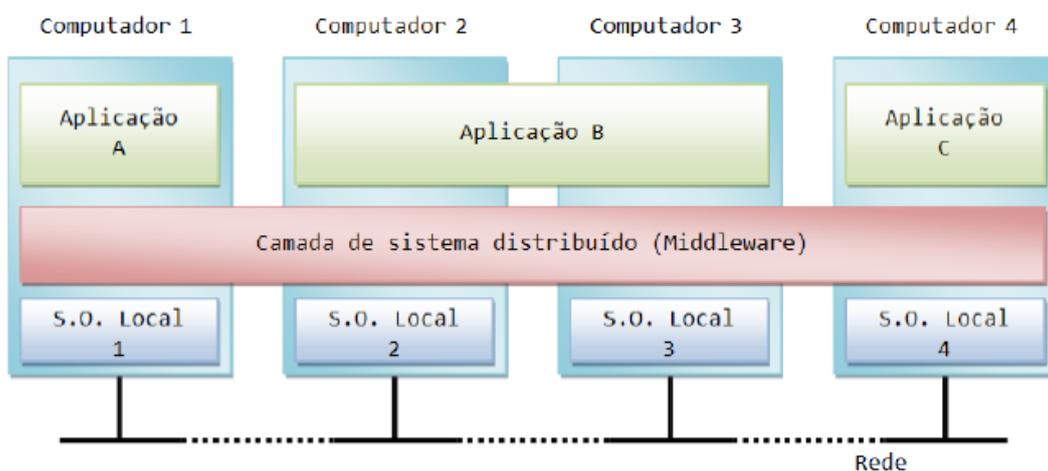


Figura 2.3: Sistema distribuído típico (Adaptada de TANENBAUM, 2006)

É correto afirmar, portanto, que a utilização do termo nuvem está diretamente relacionada à existência e à utilização de um grande sistema distribuído.

2.3.2 Middleware

O conceito de middleware é muito utilizado no campo da Computação em Nuvem. O middleware é definido como um programa de computador que faz e mediação entre software e demais aplicações (9). É utilizado para mover ou transportar informações e dados entre programas de diferentes protocolos de comunicação, plataformas e dependências do sistema operacional. Seu objetivo é mascarar a heterogeneidade, e facilitar o desenvolvimento de aplicações, tipicamente as distribuídas, assim como facilitar a integração de sistemas legados ou desenvolvidos de forma não integrada automática (9).

Em grande parte, sua utilização se aplica diretamente à função exercida pela camada de software (Figura 2.3) que exerce a coordenação e o gerenciamento das máquinas e aplicações que constituem um sistema distribuído.

Os serviços de IaaS existentes atualmente funcionam através da utilização de interfaces de web services oferecidas por estes middlewares, através das quais os clientes alocam recursos contidos na infraestrutura do provedor.

2.3.3 Grades computacionais

Um ambiente de grades computacionais, ou computação em grades, consiste em uma "infraestrutura de hardware e software capaz de prover acesso a recursos de computação de alta qualidade de forma segura, consistente e barata"(10). Uma definição mais recente e amplamente aceita pela comunidade científica trata as grades computacionais como "o compartilhamento seguro, confiável e eficiente de recursos computacionais e dados entre instituições organizacionais independentes"(11).

Dentre os principais tipos de recursos que podem ser compartilhados através das grades computacionais, estão o processamento e armazenamento de dados, largura de banda e ferramentas de software. Há de se ressaltar que as grades se diferenciam da computação distribuída por focar em questões como alto desempenho, coordenação, gerenciamento e compartilhamento destes recursos (12).

Um dos elementos fundamentais na implantação de um ambiente de grades computacionais é o middleware utilizado para tal. Através do middleware, cria-se uma camada de virtualização e gerência de compartilhamento sobre a qual as aplicações dos usuários são executadas e administradas.

Todos os modos de implantação da Computação em Nuvem empregam tecnologias e termos inerentes ao campo das grades computacionais. O conceito de compartilhamento de recursos entre usuários e organizações em todo o mundo criado pelas grades é adaptado para um panorama onde um provedor compartilha sua infraestrutura entre muitos clientes, utilizando mecanismos de gerência e coordenação muito similares aos existentes nas grades.

Conforme mencionado anteriormente, é importante ressaltar as diferenças entre os conceitos Computação em Nuvem e computação em grade para evitar confusão (12):

- i. Modelo de negócio: o modelo de computação em nuvem é um modelo de pagar pelo uso similar a utilidades como eletricidade. Os provedores de serviços de nuvem se beneficiam de economias de escala para realizar lucros e oferecer seus serviços a um baixo custo para os clientes. A computação em grade é normalmente um modelo mais voltado para a pesquisa que envolve a cooperação de grandes centros de pesquisa do governo e universidades. A idéia da grade também consiste de compartilhar recursos para que os pesquisadores submetam processos para resolver problemas computacionais complexos (por exemplo: projetos na área genética ou de previsão de tempo, simulação, etc.). Normalmente a computação em grade oferece tais recursos computacionais para os membros participantes de um determinado projeto.
- ii. Arquitetura e distribuição: normalmente em grades os recursos computacionais compartilhados podem estar em regiões geográficas e pertencer a instituições diferentes. Isto em geral não ocorre na computação em nuvem, cujas soluções são normalmente oferecidas pela mesma empresa embora também se possa haver distribuição geográfica (Amazon, por exemplo, oferece seus serviços na costa leste e oeste dos EUA e na Europa). Outra diferença é que a distribuição de aplicações em nuvem dá-se normalmente através de máquinas virtuais, enquanto que em grades estas normalmente consistem de processos submetidos para execução em batch.

2.3.4 Computação utilitária

Como discutimos anteriormente, uma das características da Computação em Nuvem consiste na medição dos serviços oferecidos. Os recursos envolvidos devem

ser monitorados e seu grau de utilização deve ser reportado tanto ao provedor quanto ao cliente de forma transparente. Nos casos onde há a necessidade de tarifação do modelo, esta característica torna-se vital para a viabilidade comercial do negócio. Assim como no fornecimento de bens utilitários como água, gás, telefonia e energia elétrica, o campo da Computação Utilitária (em inglês, Utility Computing) busca desenvolver métodos para a provisão de recursos computacionais onde o cliente é cobrado com base na quantidade e na qualidade daquilo que utiliza.

Tendo em vista os tipos de serviços existentes, como entrega de capacidade de processamento, armazenamento de dados e redes de comunicação, e considerando-se que os mesmos devem atender automaticamente a flutuações de demanda, a Computação Utilitária deve atender a questões como o aumento de confiabilidade, melhoria de desempenho e redução do custo total na utilização de recursos (13).

Paradigmas como grades computacionais e computação peer-to-peer propunham, desde sua concepção, a transformação dos recursos computacionais em um quinto bem utilitário (assim como o provimento de energia, água, gás e telefone). A Computação em Nuvem renova esta proposição através do fornecimento de infraestrutura, plataformas de desenvolvimento e software (14).

2.3.5 Virtualização

Nos últimos anos, os custos provenientes de espaço físico, energia, refrigeração e manutenção estão, mais uma vez, revertendo o panorama da utilização de grandes datacenters, e a virtualização surge como peça fundamental para uma melhor utilização dos recursos computacionais disponíveis.

A virtualização consiste na técnica de prover a ilusão de acesso exclusivo a um recurso compartilhado. Com isto, é possível a execução de um grande número de workloads em um mesmo sistema de hardware de forma previsível, segura e isolada (15). As diferentes modalidades existentes de virtualização oferecem benefícios diversos, como uma melhor utilização do sistema através do compartilhamento de recursos físicos, execução de aplicações em ambientes isolados, eliminação de conflitos de domínio (como utilização das mesmas portas TCP), utilização simultânea de vários sistemas operacionais em uma única máquina e isolamento de falhas. A seguir, são descritos os três principais tipos de virtualização:

Virtualização para gerência de recursos: Ao executar diferentes workloads em

um único sistema, torna-se aparente a necessidade de gerenciar os recursos de hardware utilizados. Esta é a forma mais básica de virtualização, e pode ser exemplificada pelo gerenciamento realizado pelos sistemas operacionais ao executar simultaneamente um grande número de aplicações.

Virtualização em sistema operacional: Ao executar diferentes workloads em um único sistema, torna-se aparente a necessidade de gerenciar os recursos de hardware utilizados. Esta é a forma mais básica de virtualização, e pode ser exemplificada pelo gerenciamento realizado pelos sistemas operacionais ao executar simultaneamente um grande número de aplicações.

Virtualização completa: Neste tipo de virtualização, sistemas operacionais diversos rodam simultaneamente na mesma máquina com um leve gerenciador de recursos, chamado hipervisor, sendo executado sobre o hardware. As instâncias executadas neste modelo são chamadas de máquinas virtuais. Há dois tipos básicos de hipervisores. O primeiro tipo é executado diretamente no hardware, enquanto o segundo tipo necessita de um sistema operacional hospedeiro.

A virtualização é a peça chave para a construção de um ambiente com tais funcionalidades de multi-arrendamento. O caráter "elástico" da nuvem de que temos falado, com a provisão de recursos virtualmente ilimitados, é obtido através da criação de novas instâncias de máquinas virtuais quando se faz necessário, oferecendo ao cliente upgrades de recursos instantâneos e automatizados.

2.3.6 Web services

O funcionamento da grande variedade de serviços disponíveis na internet, comumente designados como web services, segue uma série de protocolos que, em grande parte, empregam o padrão de linguagem *Extensible Markup Language* (Linguagem de Marcação Extensível) (XML) como elemento funcional básico. Adicionalmente, linguagens de descrição são utilizadas para a composição de interfaces através das quais clientes têm acesso aos serviços.

Os web services são oferecidos através da construção de sistemas que implementam *Service-Oriented Architecture* (Arquitetura Orientada a Serviços) (SOA). O surgimento destas arquiteturas está intrinsecamente relacionado a necessidades de mercado, quando um número crescente de empresas sentiu a

necessidade de integração e interoperabilidade entre sistemas demasiadamente heterogêneos (16).

A Computação em Nuvem torna possível a utilização de infraestrutura de hardware e software remotamente, oferecendo-os como serviços aos usuários finais. Isto é possível devido à utilização de interfaces de web services, através das quais requisições são traduzidas para o processamento por parte dos servidores de gerenciamento dos provedores, que administram o provimento de recursos de acordo com suas políticas internas de segurança e *Service Level Agreement* (Acordos de Nível de Serviço) (SLA) estabelecidos com seus clientes.

2.3.7 Computação verde

O conceito computação verde (em inglês, green computing) refere-se à utilização eficiente dos recursos de computação, minimizando o impacto ambiental, maximizando a sua viabilidade econômica e assegurar direitos sociais. Já segundo (17) a computação verde é "o estudo e a prática de projeto, fabricação, uso e descarte de computadores, servidores e subsistemas associados (tais como monitores, impressoras, dispositivos de armazenamento e de rede e sistemas de comunicação), de forma eficiente e eficaz com um mínimo ou nenhum impacto sobre o ambiente".

Por tudo o que foi discutido até o momento, algumas das tecnologias são classificadas como verdes, pois contribuem para a redução do consumo de energia ou das emissões de dióxido de carbono, são Computação em Nuvem, grades computacionais e virtualização em datacenters.

2.4 Modelos de serviço

O novo paradigma proposto pela Computação em Nuvem baseia-se na entrega de uma vasta variedade de recursos através de serviços disponíveis em escala mundial. Isto implica em um impacto profundo na forma como empresas e pessoas utilizam o hardware e o software de que necessitam, além de elevar ainda mais a importância das redes de comunicação através das quais esses recursos se tornam acessíveis.

A variedade de serviços disponíveis em uma nuvem faz com que sua classificação divirja sobremaneira. Nomenclaturas como *Communication as a Service* (Comunicação como Serviço) (CaaS), *Management as a Service* (Gerenciamento como Serviço) (MaaS), *Data as a Service* (Dados como Serviço) (DaaS), *Information Technology as a Service* (Tecnologia da Informação como Serviço)

(ITaaS) e até mesmo *Everything as a Service* (Tudo como Serviço) (XaaS) são encontradas com relativa facilidade no mercado e na academia (18). Todavia, a estrutura da Computação em Nuvem é visivelmente formada por três camadas fundamentais (19). Logicamente interconectados (Figura 2.4), os elementos que compõem esta visão triádica da nuvem são a SaaS, a *Platform as a Service* (Plataforma como Serviço) (PaaS) e o IaaS.

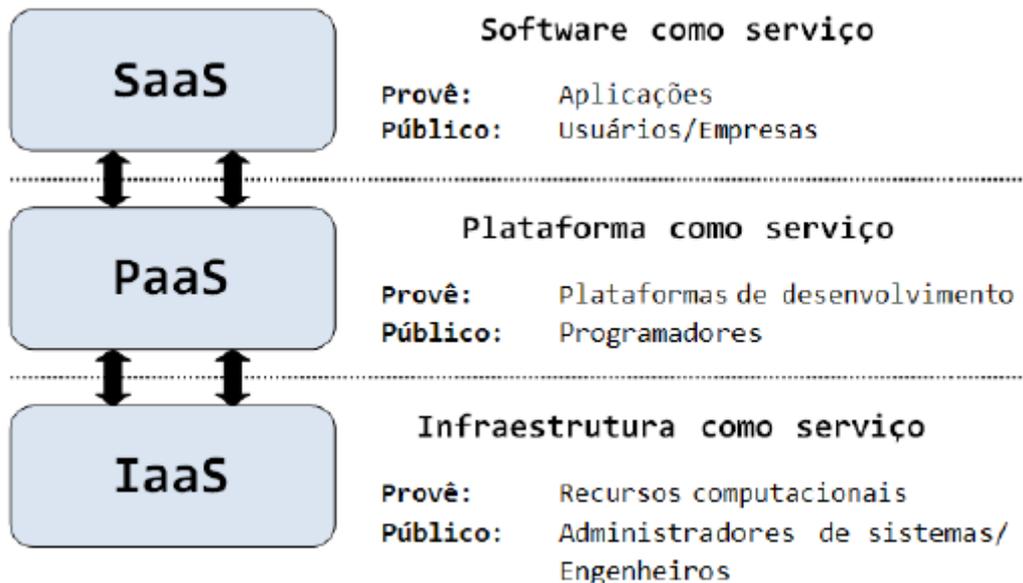


Figura 2.4: Modelos de serviço da Computação em Nuvem (Adaptada de STANOEVSKA-SLABEVA et al., 2010)

Atualmente, este tipo de classificação é amplamente aceito na literatura, pois engloba grande parte dos subtipos encontrados e define suas funcionalidades de forma coerente e concisa.

2.4.1 Software como serviço (SaaS)

O modelo de Software como Serviço (SaaS) consiste no provimento via rede de programas executados em sistemas e infraestruturas de terceiros, através de uma técnica de cobrança baseada na taxa de utilização do usuário. Comumente chamada de “pay-per-use”, essa técnica permite que clientes eliminem grande parte dos custos atrelados ao uso de um determinado software, excetuando-se os relativos à utilização das redes de comunicação. Não obstante, o emprego de SaaS gera uma grande dependência do usuário em relação a recursos que se encontram além de seu domínio administrativo, o que levanta muitos questionamentos a respeito da viabilidade de uma adesão integral ao modelo.

Embora seja a camada mais visível ao usuário final, o funcionamento do SaaS está intimamente ligado ao emprego de plataformas e infraestrutura como serviço. A comunicação vertical entre os elementos estruturais da Computação em Nuvem representada na Figura 2.4 é realizada em sua plenitude através do uso de SaaS, pois as aplicações oferecidas são executadas em ambientes baseados no modelo de PaaS, que, por sua vez, estão configurados sobre hardware virtualizado por meio de IaaS. A atuação conjunta destes elementos forma um sistema computacional complexo e inovador, cujo potencial ainda está sendo explorado tanto pelo mercado quanto pela academia.

O fato de toda a infraestrutura empregada estar inteiramente na nuvem cria a possibilidade de utilização dos mais diversos tipos de aplicação por meio de dispositivos com sérias limitações de recursos. Desta forma, dispositivos móveis podem ter acesso via web services a programas sofisticados e que necessitam de grande capacidade de processamento para sua execução.

2.4.2 Plataforma como serviço (PaaS)

A camada de Plataforma como Serviço (PaaS) é direcionada aos programadores, oferecendo-os um ambiente autoconfigurável através do qual é possível o desenvolvimento e o teste de aplicações sem a preocupação com a infraestrutura necessária para tal. Por meio do modelo de PaaS, o programador pode criar e executar uma aplicação inteiramente na nuvem. As variações no uso da aplicação são então gerenciadas de forma integral pela plataforma. Caso seu nível de utilização cresça, um acréscimo na quantidade de recursos alocados é efetuado (20). Uma ação análoga é realizada nos casos de decréscimo do nível de utilização. Ademais, o ambiente oferecido pela Plataforma como Serviço pode cobrir todas as etapas de desenvolvimento de software ou especializar-se em uma área específica, como o gerenciamento de conteúdo (21).

Percebe-se, neste ponto, a comunicação vertical existente entre as camadas que compõem a estrutura da Computação em Nuvem. Enquanto utiliza interfaces de provedores para o consumo de recursos computacionais no modelo IaaS, a Plataforma como Serviço, por sua vez, oferece interfaces próprias para a comunicação com aplicações desenvolvidas pelos programadores que a utilizam. Trata-se, portanto, de uma camada de abstração entre a IaaS e a SaaS (19).

2.4.3 Infraestrutura como serviço (IaaS)

Em contrapartida à venda direta de componentes de hardware e servidores completos, o modelo de Infraestrutura como Serviço (IaaS) oferece ao consumidor poder de processamento, armazenamento de dados, estrutura de rede e uma série de outros recursos básicos de forma virtualizada e seguindo os preceitos do self-service sob demanda, elasticidade e multi-arrendamento. Instituições que aderem a este tipo de serviço não têm controle direto sobre a infraestrutura física, mas podem utilizar os recursos virtualizados que lhes são oferecidos da forma que lhes convier, salvo restrições estabelecidas pelo provedor.

Ao eliminar a necessidade de posse de uma infraestrutura particular, o emprego de IaaS têm impacto direto em custos relacionados ao consumo de energia, espaço físico e manutenção de hardware.

Em conformidade com as cinco características principais da Computação em Nuvem, o modelo de IaaS fornece infraestrutura de forma dinâmica, automatizada e inerentemente escalável. Com isto, o cliente do serviço pode aumentar ou diminuir a quantidade de recursos alocados para si de acordo com sua necessidade. Ao abrir mão de uma infraestrutura própria através da adesão ao IaaS, uma organização vê muitos de seus dados ultrapassarem seus domínios, sendo armazenados na nuvem, um ambiente sobre o poder administrativo de terceiros. Por conseguinte, muitas questões relacionadas à segurança da informação são levantadas, sendo este um dos principais pontos de discussão do campo da Computação em Nuvem.

2.5 Modelos de implementação

Além da categorização feita na Seção 2.4 com relação aos serviços oferecidos, também há uma categorização com relação à abrangência e acesso à nuvem. A classificação geralmente citada na literatura é a de: nuvem pública, nuvem privada e nuvem híbrida. A seguir, descrevemos brevemente cada uma delas.

2.5.1 Nuvem pública

De acordo com (6), o software e hardware de um datacenter é a nuvem e se essa nuvem for disponibilizada para o público em geral no modo “pay-per-use”, ela é chamada nuvem pública.

2.5.2 Nuvem privada

Por outro lado, o datacenter interno de uma organização, não disponível ao público em geral, é a nuvem privada. Segundo (22), o objetivo principal de uma nuvem privada não é vender capacidade pela Internet, através de interfaces acessíveis ao público em geral, mas sim dar aos usuários locais uma infraestrutura ágil e flexível para suportar workloads de serviços dentro de seu próprio domínio administrativo.

2.5.3 Nuvem híbrida

Uma nuvem privada, no entanto, pode dar suporte à uma nuvem híbrida, através da complementação da capacidade da infraestrutura local com a capacidade computacional de uma nuvem pública (22). A nuvem privada/híbrida também pode permitir acesso remoto através de interfaces remotas, como a de Web Services que a Amazon EC2 utiliza (22).

2.6 Esforços de normalização

Desde a popularização do termo Computação em Nuvem com a IBM Blue Cloud (23), diversas iniciativas se tornaram provedoras de computação em nuvem: Amazon e sua *Elastic Compute Cloud* (EC2), Google com a ferramenta Google App, Microsoft com sua plataforma Windows Azure, Salesforce e sua Force.com e demais. Apesar dessas soluções se encaixarem na definição de computação em nuvem, elas se diferem nas suas programabilidades. Segundo (24) o conceito programabilidade é relativo a programar as características que a rede pode oferecer aos desenvolvedores, medindo assim quanta liberdade os mesmos teriam para poder manipular os recursos e dispositivos.

Esse conceito pode ser aplicado para comparar as soluções de Computação em Nuvem. Nuvens mais programáveis oferecem ambientes onde os desenvolvedores estão livres para escolher seus próprios paradigmas, linguagem e plataforma de programação, possuindo total controle sobre os seus recursos virtualizados. Nuvens menos programáveis restringem os desenvolvedores de certa forma: forçando um determinado conjunto de linguagens de programação, ou permitindo suporte para apenas um paradigma de aplicação. Por outro lado, um elevado nível de programabilidade torna difícil o manejar a nuvem porque o fornecedor da nuvem terá muitos ambientes distintos para manejar. Por exemplo, se o provedor da nuvem

permitir seus usuários a utilizarem qualquer sistema operacional na máquina virtual, o operador da nuvem terá que lidar com um elevado número de soluções para prover serviços de tolerância à falha. Além disso, soluções menos programáveis abstraem alguns problemas operacionais (comunicação entre processos, acesso à rede, funções de armazenamento, etc.) por alguns modelos de middleware. Um exemplo de solução em nuvem com alta programabilidade é a Amazon EC2, onde os usuários podem escolher o sistema operacional de um conjunto de sistemas suportados dados pela Amazon e eles podem configurar suas máquinas virtuais para trabalhar como eles acreditam que encaixe. A ferramenta Google App é um exemplo de uma solução pouco programável, visto que ela restringe os desenvolvedores ao paradigma web e a poucas linguagens e programação.

Uma característica comum presente em todas as nuvens é que os donos das nuvens compreensivelmente evitam revelar as características por trás de suas soluções, visto que isso é parte de suas informações estratégicas. Apesar disso, são muitos os esforços feitos por várias instituições acadêmicas e não acadêmicas para desenvolver uma solução de código aberto. A observação e a classificação dessas soluções podem alavancar mais ainda a área de pesquisa da Computação em Nuvem fornecendo um bom ponto de partida para descobrir diferentes abordagens para os problemas presentes no ambiente de Computação em Nuvem.

O desenvolvimento de soluções em Computação em Nuvem trás diversos desafios técnicos para os desenvolvedores. Esses desafios podem ser agrupados em três principais áreas: negociação, decisão e operação. Na área de negociação, os desafios são relativos à como desenvolvedores de interface de aplicação com a nuvem assim como a descrição das ofertas da nuvem. Isso inclui também a definição do nível de programabilidade que a nuvem oferecerá. A área de decisão lida com o principal problema que as nuvens enfrentam por trás das cenas: como os recursos virtuais podem ser agendados para encontrar as exigências dos usuários, por exemplo? Por fim, a área de operação é associada à execução de decisões e comunicação entre elementos da nuvem.

Um desafio considerável presente em várias discussões levantadas a cerca da nuvem é relativo à necessidade de padronização. Todas as três áreas enfrentam o desafio da padronização de certa forma, mas o desafio principal ocorre na área de negociação. Atualmente, provedores de nuvem oferecem interfaces proprietárias para acessar seus serviços. Isso bloqueia usuários dentro de um determinado

provedor como eles não podem migrar suas aplicações e serviços com facilidades entre provedores (25). É esperado que os provedores enxerguem esse problema e trabalhem juntos para oferecer uma API padronizada baseada na abertura de padrões como *Simple Object Access Protocol* (Protocolo Simples de Acesso a Objetos) (SOAP) e *Representational State Transfer* (Transferência de Estado Representacional) (REST).

A partir do desenvolvimento exponencial da Computação em Nuvem, vários padrões abertos são introduzidos. Estes destinam-se a dar orientações para implementar ou desenvolver a Computação em Nuvem da melhor forma possível. Estas normas são revistas por vários profissionais e a maioria é de uso gratuito. As diretrizes de esses padrões são baseadas nas melhores práticas. Muitos dos padrões abertos para Computação em Nuvens estão ainda em desenvolvimento, mas vamos dar uma visão geral dos mais interessantes:

- i. ***Open Cloud Manifesto***: Um esforço importante na normalização vem do Open Cloud Manifesto (26). Esta é uma iniciativa apoiada por centenas de empresas que pretende discutir com as organizações da nuvem uma maneira de produzir padrões abertos para a Computação em Nuvem. Suas doutrinas principais são a colaboração e a coordenação de esforços na normalização, a adoção de padrões abertos sempre que seja necessário e no desenvolvimento de normas baseadas em requisitos do cliente. O manifesto destaca os requisitos que precisam ser padronizados em um ambiente de nuvem, para garantir a interoperabilidade nos cenários mais típicos de interação.
- ii. ***Cloud Security Alliance (CSA)***: A Cloud Security Alliance foi criada para promover a utilização das melhores práticas para garantir a segurança no ambiente da Computação em Nuvem, e fornecer educação sobre o uso da Computação em Nuvem para ajudar a proteger todas as outras formas de computação.
- iii. ***Open Virtualization Format (Formato Aberto de Virtualização) (OVF)***: Esta especificação descreve um formato aberto, seguro, eficiente e extensível para o acondicionamento e distribuição de software para rodar em máquinas virtuais. É gerenciado pelo *Distributed Management Task Force* (DMTF).

- iv. *Open Cloud Consortium (OCC)*: Para promover o desenvolvimento de padrões para Computação em Nuvem e diretrizes para operar entre as nuvens.
- v. *Organization for the Advancement of Structures Information Standards (OASIS)*: OASIS impulsiona o desenvolvimento, convergência e adoção de padrões abertos para a sociedade global da informação.

Por tudo isso e, como falamos anteriormente, são muitos os esforços feitos para desenvolver uma solução de código aberto. A atividade de pesquisa a realizada vai ter como foco a análise da ferramenta OpenNebula, que busca oferecer meios para a implementação de ambientes distribuídos com suporte à virtualização e baseados no paradigma da Computação em Nuvem. Após esta etapa inicial, a ferramenta vai ser utilizada para a implementação de uma nuvem privada de recursos. Na próxima Seção 2.7 vamos discutir profundamente sobre o projeto OpenNebula. Mas existem muitas outras soluções de código aberto em desenvolvimento, algumas das mais importantes serão descritas na Seção 2.8.

2.7 OpenNebula

Nesta seção vamos nos centrar analisar a ferramenta de código aberto OpenNebula, o núcleo da nossa pesquisa.

2.7.1 Introdução

"OpenNebula é uma ferramenta aberta e flexível que se encaixa nos atuais ambientes de datacenter para construir qualquer tipo de implantação IaaS na nuvem"(27). OpenNebula pode ser usado principalmente como uma ferramenta de virtualização para gerenciar sua infraestrutura virtual no datacenter ou cluster, o que é normalmente referido como nuvem privada. OpenNebula suporta nuvem híbrida para combinar infraestrutura local com infraestrutura baseada em nuvem pública, permitindo ambientes de hospedagem altamente escaláveis. OpenNebula também suporta nuvem pública ao fornecer interfaces da nuvem para expor a sua funcionalidade para máquina virtual, armazenamento e gerenciamento da rede. Em outras palavras, o projeto OpenNebula orquestra armazenamento, rede, virtualização, monitoramento e tecnologias de segurança para permitir o posicionamento dinâmico de serviços multi-tier (grupos de máquinas virtuais

interligados) em infraestruturas distribuídas, combinando recursos do datacenter e recursos da nuvem remota, acordo com as políticas de alocação (27).

OpenNebula.org é um projeto que visa à construção da ferramenta padrão de código aberto da Computação em Nuvem para gerenciar a complexidade e heterogeneidade das infraestruturas dos datacenters distribuídos. Essa é a missão do projeto OpenNebula.



Figura 2.5: Logotipo projeto OpenNebula (OPENNEBULA, 2011)

Aqui estão algumas das principais características que fazem OpenNebula diferente de outras ferramentas de gestão da nuvem (22):

Capacidades para gestão da nuvem: OpenNebula é o conjunto de ferramentas mais avançadas para a criação de nuvens privadas, públicas e híbridas, oferecendo não só uma implementação de código aberto das interfaces mais comuns da nuvem, mas também oferece recursos exclusivos para a gestão da nuvem e dos datacenters virtuais.

Capacidades para integração: Arquitetura, interfaces e componentes abertos, flexíveis e extensíveis que se encaixam em qualquer datacenter existente, e permitir a sua integração com qualquer produto e serviço da nuvem e qualquer ferramenta de gestão dos datacenters.

Capacidades para produção de ambientes: Escalabilidade e desempenho testado em infraestruturas compostas por milhares de núcleos, com os níveis de segurança e tolerância a erros exigidos em sistemas de produção.

Software totalmente open source: OpenNebula é verdadeiramente código fonte aberto, distribuído sob a licença Apache, que é rigorosamente testado e amadurecido através de uma comunidade vibrante.

2.7.2 História

OpenNebula foi criado como um projeto de pesquisa em 2005 por Ignacio M. Llorente e Rubén S. Montero. Desde sua primeira versão pública do software

de março de 2008, evoluiu através de versões de código aberto e agora opera como um projeto open source. OpenNebula é o resultado de muitos anos de pesquisa e desenvolvimento em gestão eficiente e escalável de máquinas virtuais em infraestruturas distribuídas, em estrita colaboração com a comunidade de usuários (27).

A tecnologia OpenNebula amadureceu graças a uma comunidade ativa de usuários e desenvolvedores. Diferentes projetos, grupos de pesquisa e empresas construíram novos componentes da nuvem para complementar e melhorar a funcionalidade fornecida por este conjunto de ferramentas de código aberto.

Em março de 2010, os principais autores de OpenNebula criam C12G Labs para permitir que o projeto OpenNebula não seja vinculado exclusivamente ao financiamento público. OpenNebula.org é agora um projeto gerido pelos C12G Labs (27).

A tabela 2.1 fornece um resumo bastante detalhado das principais etapas do projeto OpenNebula.

2.7.3 Arquitetura OpenNebula

O modo de operação da nuvem construída por meio da ferramenta OpenNebula é baseado em uma arquitetura semelhante ao modelo clássico dos clusters, onde um nó de *front-end* é responsável pela execução dos serviços de administração da infraestrutura, enquanto os demais nós computacionais executam as máquinas virtuais que consumirão os recursos disponíveis. Neste modelo, ao menos uma rede de comunicação física é utilizada para a interligação de todos os nós. A Figura 2.6 ilustra o ambiente constituído pela nuvem privada em questão.

O nó de *front-end* executa o OpenNebula Daemon, que representa o núcleo do sistema e administra o ciclo de vida das máquinas virtuais, além de outros aspectos relativos ao funcionamento dos demais nós (como rede, armazenamento e hipervisores). Neste nó também são executados os drivers da ferramenta, responsáveis pela constituição de interfaces entre o núcleo do sistema e elementos externos, como hipervisores e sistemas de arquivos específicos. O nó de *front-end* pode, ainda, atuar como repositório de imagens de máquinas virtuais, embora este não seja seu modo de funcionamento ideal.

Os demais nós têm hipervisores habilitados para a execução de máquinas

| | |
|--|------------|
| OpenNebula 2.2 Released | 28/03/2011 |
| OpenNebula 2.2 Beta 1 Release | 02/03/2011 |
| OpenNebula 2.0 Release | 24/10/2010 |
| OpenNebula 2.0 Beta 1 Release | 28/07/2010 |
| OpenNebula Cloud Toolkit Goes Commercial | 05/05/2010 |
| New Web Site for OpenNebula.org | 24/02/2010 |
| OpenNebula 1.4 Released | 16/12/2009 |
| OpenNebula Cloud Announcement | 18/11/2009 |
| OpenNebula 1.4 RC Released | 18/11/2009 |
| OpenNebula 1.4 Beta 2 Released | 30/10/2009 |
| New Research Grants to Fund OpenNebula until 2012 | 04/09/2009 |
| OpenNebula Tarantula Stable version 1.2.1 | 29/07/2009 |
| OpenNebula 1.4 Beta1 Codename Hourglass out for Testing | 23/07/2009 |
| Ubuntu 9.04 (Jaunty Jackalope) has been released today bringing OpenNebula | 23/04/2009 |
| OpenNebula Wins the Best Demo Award at OGF25/4th EGEE-UF | 09/03/2009 |
| OpenNebula 1.2 release | 06/02/2009 |
| OpenNebula 1.2 beta release | 05/12/2008 |
| Release of OpenNebula 1.0 for Data Center Virtualization and Cloud Solutions | 24/07/2008 |
| New Technology Preview (TP2) of the OpenNebula (ONE) Virtual Infrastructure Engine | 24/07/2008 |
| Technology Preview of the OpenNebula (ONE) Virtual Infrastructure Engine | 24/07/2008 |

Tabela 2.1: Evolução projeto OpenNebula (OPENNEBULA, 2011)

virtuais. A comunicação entre o nó de *front-end* e os nós com hipervisores se dá por meio de canais de SSH. Alternativamente, a criptografia destes canais pode ser desabilitada, resultando em um melhor tempo de transmissão das imagens das máquinas virtuais a serem executadas em um dado nó.

OpenNebula compreende a execução de três tipos de processos, como pode ser visto na Figura 2.7.

- i. **O OpenNebula daemon (oned):** Um processo centralizado que gerencia o ciclo de vida das *Virtual Machine* (Máquina Virtual) (VM) e orquestra a

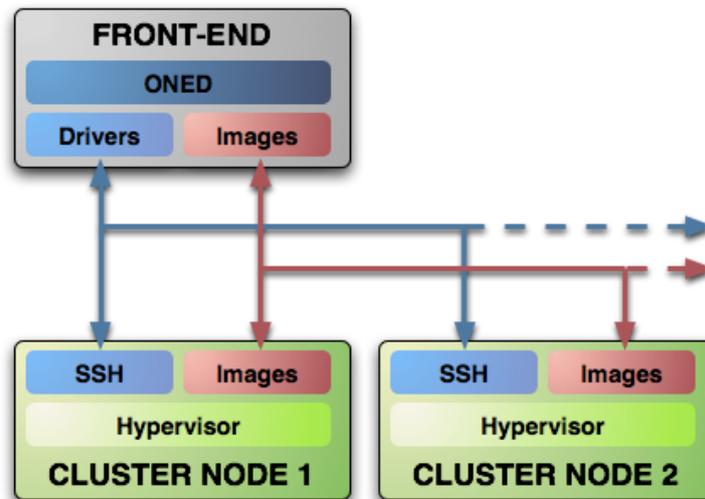


Figura 2.6: Arquitetura padrão de uma nuvem OpenNebula (OPENNEBULA, 2011)

operação de todos os módulos.

- ii. **Os drivers:** Para acessar aos sistemas específicos do cluster (por exemplo, armazenamento ou hypervisors). Fornecem uma abstração da camada de virtualização subjacente. Assim, OpenNebula não está vinculado a nenhum ambiente específico, proporcionando uma camada uniforme de gestão, independente da tecnologia de virtualização utilizada.
- iii. **O scheduler:** Para ajustar a locação das VM com base em um conjunto de políticas pré-definidas.



Figura 2.7: Processos OpenNebula (OPENNEBULA,2011)

2.7.4 Objetivos

O projeto OpenNebula tem os seguintes objetivos, a fim de liderar a inovação na nuvem (27):

- i. Desenvolver o conjunto de ferramentas mais avançada, altamente escalável e adaptável para o gerenciamento da Computação em Nuvem.
- ii. Garantir estabilidade e qualidade das ferramentas de software.
- iii. Colaborar com os usuários mais exigentes das ferramentas de gestão da Computação em nuvem.
- iv. Dar suporte aos componentes de código aberto que estão sendo criados em torno do projeto OpenNebula.
- v. Apoiar à comunidade de usuários e desenvolvedores que contribuem para o projeto.
- vi. Colaborar com os principais projetos de investigação e inovação da Computação em Nuvem.

2.8 Outras soluções de código aberto

Além de projeto OpenNebula, existem outras muitas soluções de código aberto: Eucalyptus, OpenQRM, Nimbus, TPlatform, XCP, Apache *Virtual Computing Lab* (VCL), Enomaly Elastic Computing Platform, etc. Nesta seção vamos descrever algumas destas soluções de forma rápida.

2.8.1 Eucalyptus

Eucalyptus é uma infraestrutura de software de código aberto para implementar Computação em Nuvem. Começou como um projeto de pesquisa no Departamento de Ciências da Computação, na Universidade da Califórnia, Santa Bárbara (29). Eucalyptus é um acrônimo para "Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems". "O sistema Eucalyptus é feito para proporcionar aos administradores e pesquisadores a habilidade de implantar uma infraestrutura para criação de máquinas virtuais e controle sobre recursos existentes"(29). Eucalyptus tenta oferecer a mesma funcionalidade dos serviços Amazon EC2, mas dentro de uma nuvem privada.

Com Eucalyptus é possível construir sua própria nuvem privada por trás do firewall da sua companhia. Um dos pontos fortes do Eucalyptus é que "é muito modular e o API que usa é compatível com *Amazon Web Service* (Serviço Web

Amazon) (AWS)"(29). Com essa compatibilidade (com uma das maiores nuvens de provedores públicos), Eucalyptus oferece ótima exibição, desse modo você não está limitado a sua própria nuvem e pode facilmente trocar.

Uma comparação detalhada entre a funcionalidade do OpenNebula e do Eucalyptus é (30):

- i. OpenNebula tem uma interface de administração superior. As máquinas virtuais podem migrar e suspender. Eucalyptus tem apenas a funcionalidade oferecida pelo EC2 (ou seja, não suspende, nem faz migração de qualquer tipo).
- ii. OpenNebula oferece uma interface de host física flexível para acompanhar, e gerenciar os recursos físicos da nuvem. Esse gerenciamento de interface está faltando nas versões atuais do Eucalyptus.
- iii. OpenNebula permite o gerenciamento de conjuntos de máquinas virtuais e das configurações de rede (VLANs) entre eles. Rede de isolamento é fornecida através de tabelas de transição. Eucalyptus não permite a definição de redes virtuais.
- iv. OpenNebula fornece uma API poderosa para expandir as suas capacidades. Eucalyptus apenas fornece a interface EC2-SOAP para interagir com ele.
- v. OpenNebula pode construir nuvens híbridas. Então você pode implantar suas máquinas virtuais localmente ou em outra nuvem. Isso não pode ser feito com Eucalyptus.

2.8.2 OpenQRM

OpenQRM é "uma solução de código aberto de gestão de infraestrutura muito flexível"(31). Ele é completamente conectável com uma arquitetura que foca a automaticidade, rapidez, monitoramento, alta disponibilidade, Computação em Nuvem e especialmente suporte a múltiplas tecnologias de virtualização. OpenQRM não vem com muitas funcionalidades. Ele oferece todos os tipos de recursos para a instalação de plugins que tornam OpenQRM uma importante ferramenta para gestão de infraestrutura. Através desses plugins são utilizadas as plataformas de virtualização, armazenagem, controle e implantação da Computação em Nuvem.

OpenQRM tem as seguintes limitações (32):

- i. **Zonas de disponibilidade:** A documentação não fala sobre isso, não há suporte.
- ii. **OVF:** OpenQRM não possui implementações para o formato OVF. Suas imagens virtuais possuem seu próprio formato.
- iii. **Documentação:** Há uma limitada documentação disponível. OpenQRM foca principalmente nos seus produtos em geral, com a Computação em Nuvem sendo o único plugin. Não existe muita documentação e experiência sobre o assunto.

2.8.3 Nimbus

Nimbus (33) é "uma solução de código aberto (distribuído sob a licença Apache) para tornar clusters em uma infraestrutura como serviço (IaaS), focado principalmente nas aplicações científicas". Essa solução dar aos usuários a possibilidade de alocar e configurar recursos remotos implantando máquinas virtuais, conhecido como *Virtual Workspace Service* (VWS). Um VWS é um gerenciador de máquinas virtuais que invocar diferentes interfaces.

Para implantar as aplicações, Nimbus oferece uma configuração "cloudkit" que consiste em um serviço de gerenciamento de hospedagem e um repositório de imagem. Os componentes da área de trabalho são mostrados na Figura 2.8.

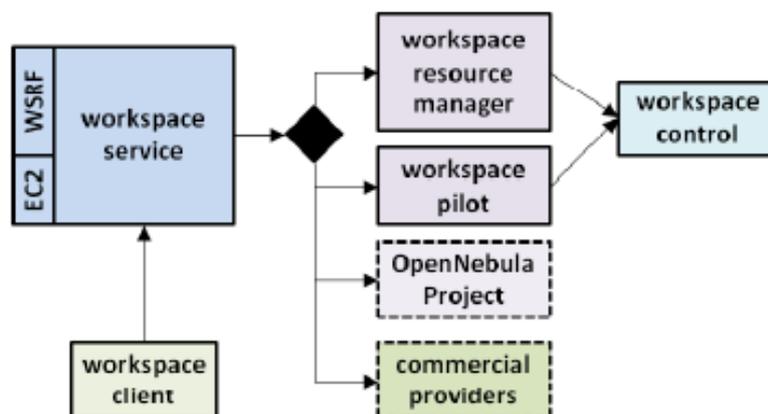


Figura 2.8: Componentes do espaço de trabalho Nimbus (KEAHAY, 2009)

Podemos ver os seguintes componentes:

- i. **Workspace service:** é baseado em web services e fornece segurança com a

autenticação e autorização GSI. Atualmente, Nimbus suporta duas interfaces: Amazon EC2 e WSRF.

- ii. **Workspace control:** é responsável por controlar as instâncias das máquinas virtuais, gerenciando e reconstruindo imagens, intergrando uma máquina virtual com a internet atribuindo endereços IP e MAC. As ferramentas de controle da área de trabalho operam com o Xen hypervisor e podem também operar com KVM5.
- iii. **Workspace resource management:** é uma solução de código aberto para gerenciar diferentes máquinas virtuais, mas podem ser substituídas por outras tecnologias como OpenNebula.
- iv. **Worspace pilot:** é responsável por prover virtualização com pequenas mudanças em uma célula de operação.

2.8.4 TPlatform

TPlatform (34) é "uma solução da nuvem que prover uma plataforma de desenvolvimento para aplicações via web, que é inspirada nas tecnologias da nuvem Google, e que age como uma solução plataforma como serviço (PaaS)". Suas infraestruturas são suportadas por três tecnologias: um sistema de arquivos escalável chamado *Tianwang File System* (Sistema de Arquivos de Tianwang) (TFS) que é similar ao *Google File System* (Sistema de Arquivos de Google) (GFS), o mecanismo de armazenamento de dados BigTable, e o modelo de programação MapReduce.

2.8.5 XCP

O Xen hypervisor (35) é "uma solução para virtualização de infraestrutura que prover uma camada de abstração entre o hardware do servidor e o sistema operacional. Xen hypervisor permite cada servidor físico execute vários servidores virtuais lidando com o sistema operacional e seus aplicativos de setor físico. A solução Xen é usada por muitas soluções de Computação em Nuvem como Amazon EC2, Nimbus e Eucalyptus". Recentemente, a Xen.org anunciou a XCP (35) como uma solução para infraestrutura de virtualização da nuvem. Mas, diferentemente das soluções de código aberto das nuvens existentes, a XCP não fornece uma arquitetura global para serviços da nuvem. Seu objetivo é prover uma ferramenta para copiar com configuração automática e manutenção de plataformas da nuvem (35).

A arquitetura XCP, como pode ser visto na Figura 2.9, é baseada no XCP hosts que são responsáveis por hospedar as máquinas virtuais. De acordo (35), esses hóspedes podem ser agregados em uma variedade de recursos XCP e usar um armazenamento compartilhado, assim as máquinas virtuais podem ser iniciadas e reiniciadas em qualquer host XCP. O Master XCP Host oferece uma interface de administração e encaminha mensagens de comando para os outros hosts XCP.

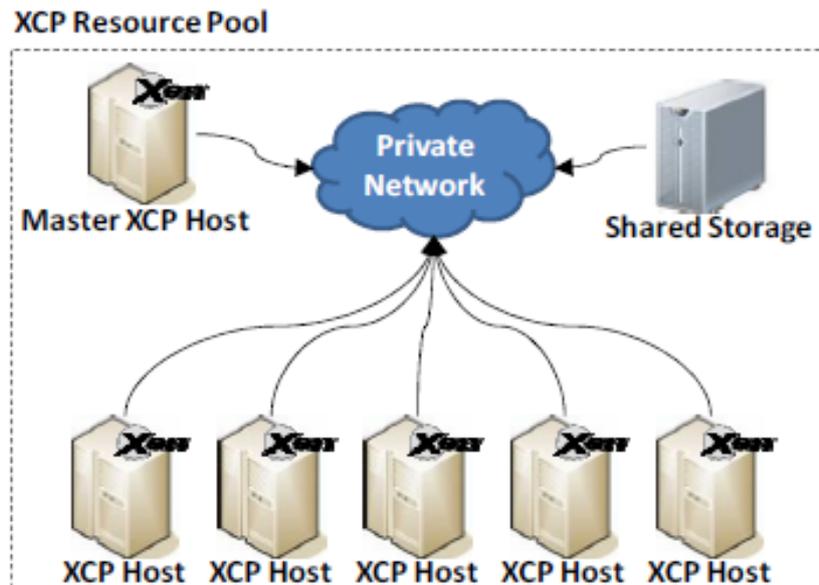


Figura 2.9: Arquitetura XCP (TAKAKO, 2010)

2.8.6 Resumo

Fazendo um resumo para obter uma idéia geral, apresentamos a Figura 2.10 onde mostramos as soluções de código aberto e algumas das soluções comerciais mais importantes da computação em nuvem. O gráfico compara as soluções e seu modelo de negócio (hardware, middleware e nível de usuário), de acordo com a flexibilidade de configuração.

Finalmente, a Tabela 2.2 apresenta um quadro comparativo das soluções de código aberto da nuvem descritas nesta seção, em termos de tipo de serviço (SaaS, PaaS e IaaS), as principais características e as tecnologias de infraestrutura.

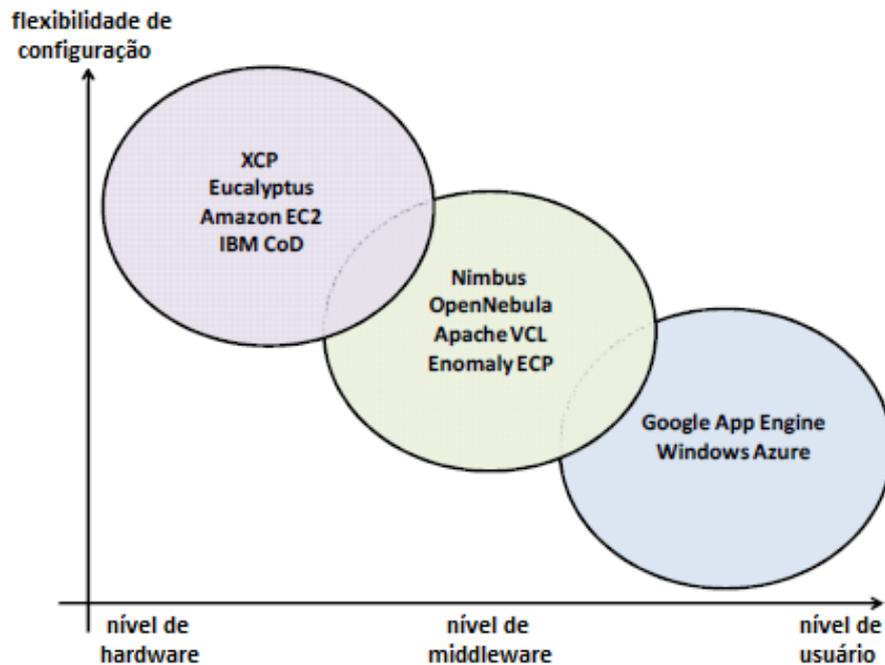


Figura 2.10: Soluções Computação em Nuvem (Adaptado de TAKAKO, 2010)

| Soluções | Serviço | Principais Características | Infraestrutura |
|------------|---------|--|---|
| OpenNebula | IaaS | Alocação de recursos | Xen hypervisor, KVM e VMware |
| Eucalyptus | IaaS | Arquitetura hierárquica | Xen hypervisor e KVM |
| OpenQRM | IaaS | Documentação limitada | Xen hypervisor, KVM, VMware e Linux VServer |
| Nimbus | IaaS | Tornar clusters em uma IaaS para Computação em Nuvem | Xen hypervisor e KVM |
| TPlatform | PaaS | Centra-se em aplicações web de mineração de textos | TFS, BigTable e MapReduce |
| XCP | IaaS | Apenas uma ferramenta para manutenção automática de nuvens | Xen |

Tabela 2.2: Comparação das soluções de código aberto da Computação em Nuvem

Capítulo 3

Análise da ferramenta OpenNebula

Nesse capítulo são apresentados, inicialmente, os equipamentos utilizados para a análise da ferramenta OpenNebula, o software open source para Computação em Nuvem, que constitui o objetivo principal deste trabalho. Nas Seções 3.2 e 3.3 são descritas as metodologias utilizadas para a obtenção dos ambientes. Especificamente, a Seção 3.2 analisa as etapas necessárias para criar uma nuvem privada. Em seguida, a Seção 3.3 descreve como lançar uma VM. A Seção final discute os comandos OpenNebula executados na metodologia, utilizados para a obtenção dos resultados.

3.1 Equipamento utilizado

No desenvolvimento desse trabalho os ambientes foram desenvolvidos em PCs conectados a um roteador wireless, resumidamente descritos pela Tabela 3.1.

| | Computador pessoal | Computador pessoal | Roteador Wireless |
|-------------------|--------------------|--------------------|-------------------|
| Fabricante | ACER | ACER | D-Link |
| Modelo | Aspire 5741 | Aspire 5741 | DI-524, 54Mbps |
| Processador | Intel Core i3-350M | Intel Core i3-350M | - |
| RAM | 4GB | 4GB | - |
| OS | Linux Ubuntu 10.10 | Linux Ubuntu 10.10 | - |
| Papel no Ambiente | Front-end | Nó | Servidor DHCP |

Tabela 3.1: Equipamento utilizado no desenvolvimento do trabalho

Um dos computadores pessoais foi empregado como nó de *font-end*, enquanto ao outro foi responsável pela execução do hypervisor e pela alocação de máquinas virtuais.

3.2 Criação de nuvem privada

A nuvem privada implementada é composta por dois nós. Como falamos na Seção anterior, o nó de *front-end* é constituído por um dos computadores pessoais. O nó computacional adicional é constituído pelo outro PC. A Figura 3.1 ilustra o ambiente construído.

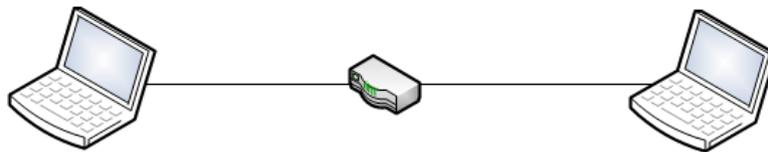


Figura 3.1: Nuvem privada implementada

O processo de instalação da nuvem privada pode ser dividido em cinco etapas. Inicialmente, os pacotes do OpenNebula (versão atual 2.2) são instalados no nó de front-end. Para tal, utiliza-se o seguinte comando no terminal:

```
$ sudo apt-get install opennebula
```

Este comando realiza várias ações de configuração, como a criação de uma conta de administrador, geração de um par de chaves RSA e a inicialização do *daemon* do OpenNebula.

O segundo passo consiste em adicionar os nós computacionais ao ambiente. Para tal, o seguinte comando é utilizado, sendo *desktop01* o hostname de nossa segunda máquina e tendo em vista que o hypervisor utilizado é o KVM:

```
$ onehost add desktop01 im _kvm vmm _kvm tm _ssh
```

Em seguida, configura-se o acesso SSH entre os nós. Como só há um nó computacional além do nó de *front-end*, apenas comando abaixo é suficiente:

```
$ sudo -u oneadmin ssh desktop01
```

A quarta etapa consiste em instalar os pacotes necessários do OpenNebula no nó computacional. Para isto, nesta máquina, executa-se o comando abaixo:

```
$ sudo apt-get install opennebula-node
```

Por fim, são realizadas as configurações de autenticação no escopo de acesso SSH. Assim, o *front-end* vai se comunicar com os controladores dos nós via SSH sem digitar nenhuma senha. OpenNebula usa o usuário *oneadmin* para gerenciar tudo, é por isso que o seu par de chaves precisa ser conhecido por todos os nós. Para tal, no nó computacional, é executado o comando abaixo, onde (*output*) deve ser substituído pela saída obtida no terminal após a utilização do comando *onehost*:

```
$ sudo tee /var/lib/one/.ssh/authorized_keys < < EOT > (output)
> EOT
```

Os nós existentes no ambiente podem ser então listados através do comando:

```
$ onehost list
```

3.3 Lançamento de máquina virtual

Uma vez que a nuvem privada está rodando, o nó pode iniciar as suas VMs. Toda a configuração do ambiente deve ser feita no nó *front-end*, que orquestra e administra o lançamento e ciclo de vida das máquinas virtuais. A Figura 3.2 ilustra o ambiente construído.

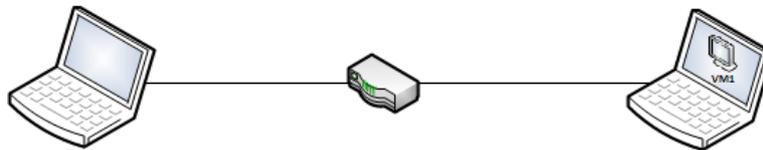


Figura 3.2: Máquina virtual (VM1) implementada

Primeiro, precisa-se preparar uma rede virtual para a nossa VM. Cria-se um arquivo de texto *vmnet.template* para gerar uma rede virtual com apenas um *Internet Protocol* (Protocolo de Internet) (IP), contendo o seguinte (muda-se de acordo com o endereço *Local Area Network* (Rede de Área Local) (LAN)):

```
NAME = "VLAN"
TYPE = FIXED
BRIDGE = eth0
LEASES = [IP=192.168.0.1]
```

Figura 3.3: Conteúdo padrão do arquivo de texto *vmnet.template*

Em seguida, utiliza-se o seguinte comando no terminal para criar a rede virtual recém configurada:

\$ onevnet create vmnet.template

É importante verificar a criação correta da rede. Para mostrar uma lista de todas as redes virtuais disponíveis é executado o comando abaixo:

\$ onevnet list

Além disso, para mostrar os detalhes da rede virtual criada é interessante utilizar o seguinte comando:

\$ onevnet show

Depois de ter configurada a rede virtual, com o comando *onevm* de OpenNebula é possível lançar uma máquina virtual do *front-end* para o nó. Para construir a VM, cria-se um arquivo de texto (*primeiraVM.template*) para apresentar a imagem dessa máquina virtual. A imagem vai estar alocada em um diretório do *front-end* (por exemplo: */opt/nebula/images*). O arquivo gerado vai ter o seguinte conteúdo:

```
NAME = VM1
CPU = 0.5
MEMORY = 128
OS = [
  kernel = "/boot/vmlinuz-2.6.18-4-xen-amd64",
  initrd = "/boot/initrd.img-2.6.18-4-xen-amd64",
  root = "sda1" ]
DISK = [
  source = "/opt/nebula/images/disk.img",
  target = "sda1",
  readonly = "no" ]
DISK = [
  type = "swap",
  size = 1024,
  target = "sdb"]
NIC = [ NETWORK = "VLAN" ]
```

Figura 3.4: Conteúdo padrão do arquivo de texto *primeiraVM.template*

Por fim, depois de ter adaptados os requisitos para a imagem da máquina virtual, lança-se a VM:

\$ onevm submit primeiraVM.template

Finalmente, pode-se usar novamente o comando *onevm* de OpenNebula para identificar e controlar o estado da máquina virtual:

\$ onevm list

3.4 Comandos

OpenNebula oferece seis comandos para interagir com o sistema: *onehost*, *onevnet*, *onevm*, *oneuser*, *oneimage*, *onecluster*, *oneauth*.

Especificamente, na metodologia descrita no capítulo utilizam-se três comandos:

onehost: Este comando permite ao usuário gerenciar hosts no servidor de OpenNebula (*front-end*). Ele fornece a funcionalidade de adicionar, eliminar e controlar um determinado host ou listar todos os hosts disponíveis na nuvem.

onevnet: Este comando permite ao usuário gerenciar redes virtuais no servidor OpenNebula (*front-end*). Ele fornece a funcionalidade de adicionar, excluir e monitorar uma determinada rede virtual.

onevm: Este comando permite ao usuário gerenciar máquinas virtuais em OpenNebula. O usuário pode implantar, migrar e controlar uma máquina virtual.

No próximo capítulo, serão mostrados os resultados da execução de alguns destes comandos no ambiente implementado.

Capítulo 4

Resultados

Nesse capítulo são exibidos e descritos os resultados obtidos da realização deste trabalho. Especificamente, os resultados obtidos da execução dos comandos OpenNebula na criação dos ambientes analisados no capítulo anterior. Eles podem ser divididos em duas Seções: resultados da criação de nuvem privada e resultados do lançamento de máquina virtual.

4.1 Resultados da criação de uma nuvem privada

Uma vez feita configuração da nuvem privada descrita na Seção 3.2, os resultados obtidos são os seguintes (sendo *desktop01*: 192.168.0.73):

- i. Executando o comando *onehost list* podem ser listados os nós existentes no ambiente:

```
$ onehost list
HID NAME          RVM  TCPU  FCPU  ACPU  TMEM  FMEM  STAT
0   192.168.0.73    0    100   49    100  1001448  694096  on
```

Figura 4.1: Comando *onehost list*

Onde:

HID: ID do Host.

NAME: Nome do Host.

RVM: Número de Máquinas Virtuais em execução.

TCPU: CPU Total (percentual).

FCPU: CPU Livre (percentual).

ACPU: CPU Disponível (não alocadas pelas VMs).

TMEM: Memória Total.

FMEM: Memória Livre.

STAT: Estado do Host.

- ii. Executando o comando `onehost show <host_id>` pode obter-se informação dos nós existentes na nuvem implementada:

```

$ onehost show 0
HID           = 0
HOSTNAME      = 192.168.0.73
IM MAD        = im_kvm
VMM MAD       = vmm_kvm
TM MAD        = tm_ssh
MANAGED       = 1
ATTRIBUTES

                ARCH=i686
                CPUSPEED=48.4
                FREEMEMORY=694096
                HOSTNAME=ubuntu
                MODELNAME=Intel (R)
                NETRX=
                NETTX=
                PUBKEY=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAC941rM21/1dp
                KtLgSdjTJOtkbktj5VwRA7g7YcT/Fj/fw4BqtoI5+7j8NwEvt41BC6K2+9A3bUXhKD
                4Cyc7SypJ2XB9x4pPa2GykGvTO2ypxLf9TSbrTUL9ARmLR2KdU+A4E2Atm4lvB/0Qt
                VLA638UNnH8T8KrczlhjJzToLKoWkyISdxhHJGhtq+Ok2Ih2w7m6Qa7GcaXUUJNdZn
                A41pwLJprAHuDA57YpsIzbM10b19cu8THkzk+3sE4vgAnnh2uZUaxUwohb/I2j7jae
                bUmHjnevkwiImmSobiTzi0QyOO5GG7keYXCNHcgT3ffbhhegsbiJ6npu8sshraP0KBN
                oneadmin@ubuntu
                TOTALCPU=100
                TOTALMEMORY=1001448
                USEDGPU=51.6
                USEDMEMORY=806380
HOST SHARES
HID           = 0
ENDPOINT      =
MAX_CPU       = 100
MAX_MEMORY    = 1001448
MAX_DISK      = 0
CPU_USAGE     = 0
MEMORY_USAGE  = 0
DISK_USAGE    = 0
RUNNING_VMS   = 0

```

Figura 4.2: Comando `onehost show <host_id>`

- iii. Executando o comando `onehost disable <host_id>` pode desabilitar-se qualquer nó existente na nuvem implementada, como pode ser visto na Figura 4.3.
- iv. Executando o comando `onehost enable <host_id>` pode habilitar-se novamente qualquer nó com estado *off*, como pode ser visto na Figura 4.4.

```

$ onehost disable 0
$ onehost list
HID NAME          RVM  TCPU  FCPU  ACPU  TMEM  FMEM  STAT
0  192.168.0.73    0    100   49    100  1001448  694096  off

```

Figura 4.3: Comando *onehost disable <host_id>*

```

$ onehost enable 0
$ onehost list
HID NAME          RVM  TCPU  FCPU  ACPU  TMEM  FMEM  STAT
0  192.168.0.73    0    100   49    100  1001448  694096  on

```

Figura 4.4: Comando *onehost enable <host_id>*

- v. Executando o comando *onehost delete <host_id>* pode apagar-se qualquer nó existente na nuvem implementada:

```

$ onehost delete 0
$ onehost list
HID NAME          RVM  TCPU  FCPU  ACPU  TMEM  FMEM  STAT

```

Figura 4.5: Comando *onehost delete <host_id>*

4.2 Resultados do lançamento de uma máquina virtual

Uma vez feita configuração do lançamento da máquina virtual descrita na Seção 3.3, os resultados obtidos são os seguintes:

- i. Executando o comando *onevnet list* podem ser listadas as redes virtuais existentes no ambiente:

```

$ onevnet list
NID NAME          TYPE  BRIDGE
0  VLAN            1    eth0

```

Figura 4.6: Comando *onevnet list*

Onde:

NID: ID da rede.

NAME: Nome da rede virtual.

TYPE: Tipo de rede virtual (0=ranged, 1=fixed).

BRIDGE: Interface associada à rede virtual.

- ii. Executando o comando *onevnet show <network_id>* pode obter-se informação das redes virtuais existentes no ambiente:

```

$ onevnet show 0
NID           : 0
UID           : 0
Network Name  : VLAN
Type          : Fixed
Bridge        : eth0

....: Template :....
      BRIDGE=eth0
      LEASES=IP=192.168.0.1, MAC=00:03:c0:a8:00:01
....: Leases :....
IP = 192.168.0.1  MAC = 00:03:c0:a8:00:01  USED = 0  VID = -1

```

Figura 4.7: Comando *onevnet show <network_id>*

- iii. Executando o comando *onevnet delete <network_id>* pode apagar-se qualquer rede virtual existente no ambiente:

```

$ onevnet delete 0
$ onevnet list
NID NAME           TYPE   BRIDGE

```

Figura 4.8: Comando *onevnet delete <network_id>*

- iv. Executando o comando *onevm list* podem ser listadas as máquinas virtuais existentes no ambiente:

```

$ onevm list
ID   NAME  STAT CPU   MEM   HOSTNAME           TIME
0    VM1  prol  0     0     192.168.0.73      00 0:00:06

```

Figura 4.9: Comando *onevm list*

Onde:

ID: ID da VM.

NAME: Nome da VM.

STAT: Estado da VM.

CPU: Percentual CPU usada pela VM.

MEM: Memória usada pela VM.

HOSTNAME: Host onde a VM está sendo ou foi executada.

TIME: Tempo desde o lançamento da VM (dias horas:minutos:segundos).

Na Figura 4.9 ilustra-se, por exemplo, a máquina virtual criada no estado *prol*, ou seja, o sistema está transferindo os arquivos da VM pro nó. Os estados possíveis da VM são mostrados na Tabela 4.1.

| Estado | Descrição |
|-------------|--|
| <i>pend</i> | No modo padrão, uma VM inicia-se no estado pendente, esperando os recursos para correr. |
| <i>hold</i> | VM não é executável. |
| <i>stop</i> | A máquina virtual é interrompida. O estado da VM foi salvo e foi transferido de volta junto com as imagens de disco. |
| <i>susp</i> | Como stop, mas os arquivos são derivados no nó remoto para depois reiniciar a VM. |
| <i>done</i> | A VM está pronta. VMs neste estado não serão mostradas com o comando <i>onevm list</i> , mas são mantidas no banco de dados para fins contábeis. |
| <i>prol</i> | O sistema está transferindo os arquivos da VM. |
| <i>runn</i> | A VM está sendo executada. |
| <i>migr</i> | A VM está migrando de um nó para outro. |
| <i>epil</i> | Neste estado o sistema "limpa"o nó usado para hospedar a VM. |
| <i>fail</i> | VM falhar. |

Tabela 4.1: Estados das máquinas virtuais

- v. Executando o comando *onevm delete <vm_id>* pode apagar-se qualquer VM existente no ambiente implementado:

```

$ onevm delete 0
$ onevm list
ID      NAME STAT CPU      MEM      HOSTNAME      TIME

```

Figura 4.10: Comando *onevm delete <vm_id>*

Capítulo 5

Conclusões

Nesse capítulo, são feitas conclusões acerca do trabalho realizado. A Seção 5.1 lista as considerações finais provindas do mesmo, enquanto a Seção 5.2 trata das perspectivas para trabalhos futuros.

5.1 Considerações finais

OpenNebula oferece um rico conjunto de ferramentas para a construção de ambientes computacionais em nuvem, sejam eles públicos ou privados. Para a implementação de uma nuvem privada, sua boa integração com a distribuição Ubuntu do sistema operacional Linux faz com que a instalação de todos os componentes necessários seja feita de forma simples e rápida. A adição de novos nós computacionais a um ambiente previamente configurado também é facilitada, podendo ser feita sem qualquer influência no funcionamento do sistema. Além disso, a orquestração das máquinas virtuais pode também ser feita sem maiores complicações, utilizando dos comandos OpenNebula básicos: *onevnet* e *onevm*. Desta forma, um vasto conhecimento em administração de sistemas Linux não é necessário para a configuração do ambiente.

Por fim, as ferramentas oferecidas por OpenNebula são abertas e desenvolvidas sob o paradigma do software open source, com forte participação de uma comunidade de desenvolvedores, que contribui com a melhoria de funcionalidades existentes, criação de novos módulos para o sistema e detecção e correção de “bugs”. OpenNebula trata-se de uma solução realmente interessante. É possível se ter um ambiente de Computação em Nuvem sem maiores investimentos.

A atratividade criada em torno da nuvem privada se dá também em relação a uma questão bastante em voga na atualidade: o consumo consciente. Ao se utilizar melhor os recursos computacionais já existentes no datacenter da empresa, pode-se reduzir o consumo de energia, postergar a aquisição de novos equipamentos e em última instância, reduzir o volume de lixo eletrônico, que causa contaminação e outros problemas ambientais e sociais em diversos lugares do mundo.

Os esforços devem continuar. É necessário mais trabalho para remover a maior parte do misticismo e dos conflitos que envolvem o uso das nuvens.

5.2 Perspectivas para trabalhos futuros

Com a realização deste trabalho tive uma primeira experiência com o ambiente OpenNebula. O trabalho futuro é analisar quantitativamente a ferramenta através de medidas de avaliação de desempenho. Testar um ambiente mais real, com vários hosts e orquestrando múltiplas máquinas virtuais, sem limitações de recursos computacionais.

É também considerada a possibilidade de analisar outras soluções open source no paradigma da Computação em Nuvem e realizar uma comparação quantitativa entre as ferramentas.

Referências Bibliográficas

- 1 MALIS, A. Routing over large clouds (rolc) charter. *Part of the 32 IETF meeting minutes*, 1993.
- 2 MELL, P.; GRANCE, T. The nist definition of cloud computing. 2009.
- 3 RITTINGHOUSE, J. W.; RANSOME, J. F. Cloud computing. implementation, management and security. *CRC Press*, 2010.
- 4 BREY, B. B. *Applying PIC18 Microcontrollers. Architecture, Programming, and Interfacing using C and Assembly*. 2008.
- 5 MAHOWALD, R. P. Market analysis perspective: Worldwide saas and cloud services 2010: New models for delivering software. *International Data Corporation (IDC)*, 2011.
- 6 ARMBRUST, M. et al. Above the clouds: A berkeley view of cloud computing. *University of California at Berkeley*, 2009.
- 7 TANENBAUM, M. V. S. A. S. *Distributed Systems. Principles and Paradigms*. 2006.
- 8 COULOURIS, G.; DOLLMORE; KINDBERG. *Distributed Systems: Concepts and Design*. 2005.
- 9 TALIA, D.; YAHYAPOUR, R.; ZIEGLER, W. *Grid Middleware and Services: Challenges and Solutions*. 2005.
- 10 FOSTER, I. *Computational Grids*. 1998.

- 11 WEISHAUPL, T. et al. Business in the grid: The big project. *Grid Economics and Business Models*, 2005.
- 12 FOSTER, I.; KESSELMAN, C.; TUECKE, S. The anatomy of the grid: Enabling scalable virtual organization. *International Journal of High Performance Computing Applications*, 2001.
- 13 BUNKER, D. T. G. *Delivering Utility Computing - Business-driven IT Optimization*. 2006.
- 14 BUBENDORFER, K.; BUYYA, R. *Market-Oriented Grid and Utility Computing*. 2009.
- 15 SOLTER, N. A.; JELINEK, G.; D., M. *OpenSolaris Bible*. 2009.
- 16 ENDREI, M. et al. *Patterns: Service-Oriented Architecture and Web Services*. 2004.
- 17 MURUGESAN, S. Harnessing green it: Principles and practices. *IEEE Internet Computing*, 2008.
- 18 PALLIS, G. Cloud computing: The new frontier of internet computing. *View from the cloud*, 2010.
- 19 STANOEVSKA-SLABEVA, K.; WOZNIAK, T.; RISTOL, S. *Grid and Cloud Computing - A Business Perspective on Technology and Applications*. 2010.
- 20 RIGHTSCALE. *Define Cloud Computing*. 2011. RightScale Blog, disponível em <http://blog.rightscale.com/2008/05/26/define-cloud-computing>.
- 21 SUN. A guide to getting started with cloud computing. *Sun white paper*, 2010.
- 22 SOTOMAYOR, B. et al. Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing*, v. 13, n. 5, p. 14–22, 2009.
- 23 VOUK, M. A. Cloud computing - issues, research and implementations. *Computing and Information Technology. University of Zagreb, Croatia*, 2008.
- 24 CHOWDHURY, N. M.; BOUTABA, R. A survey of network virtualization. *Computer Networks*, v. 54, p. 862–876, 2010.

- 25 BUYYA, R. et al. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 2009.
- 26 OPENCLOUD. *The Open Cloud Manifesto*. 2009. Disponível em <http://www.opencloudmanifesto.org/>.
- 27 OPENNEBULA. *Projeto OpenNebula*. 2011. Disponível em <http://opennebula.org/>.
- 28 SOTOMAYOR, B. et al. Capacity leasing in cloud systems using the opennebula engine. 2008.
- 29 EUCALYPTUS. *Projeto Eucalyptus*. 2011. Disponível em <http://open.eucalyptus.com/>.
- 30 MONTERO, R. Opennebula/eucalyptus comparision. 2009.
- 31 OPENQRM. *Projeto OpenQRM*. 2011. Disponível em <http://openqrm.com/>.
- 32 RAMBHADJAN, M.; SCHUTLIJSER, A. Surfnet cloud computing solutions. *Universiteit van Amsterdam*, 2010.
- 33 KEAHEY, K. Nimbus: Open source infrastructure as a service cloud computing software. *Workshop on adapting applications and computing services to multi-core and virtualization*, 2009.
- 34 PENG, B.; CUI, B.; LI, X. Implementation issues of a cloud computing platform. *IEEE Data Engineering Bulletin*, v. 32, 2009.
- 35 XEN. *Projeto Xen Cloud Platform*. 2011. Disponível em <http://xen.org/>.