Documentos

ISSN 1677-9274
Dezembro, 2011
116

Computação em nuvem: entendendo e implementando uma nuvem privada

Embrapa

ISSN 1677-9274
Dezembro, 2011

Empresa Brasileira de Pesquisa Agropecuária
Embrapa Informática Agropecuária
Ministério da Agricultura, Pecuária e Abastecimento

Documentos 116

Computação em nuvem: entendendo e implementando uma nuvem privada

Jorge Luiz Corrêa
Marcos Cezar Visoli

Embrapa Informática Agropecuária
Campinas, SP
2011

Embrapa Informática Agropecuária
Av. André Tosello, 209 - Barão Geraldo
Caixa Postal 6041 - 13083-886 - Campinas, SP
Fone: (19) 3211-5700 - Fax: (19) 3211-5754
www.cnptia.embrapa.br
sac@cnptia.embrapa.br

1ª edição on-line 2011

Dados Internacionais de Catalogação na Publicação (CIP)
Embrapa Informática Agropecuária

Corrêa, Jorge Luiz.

Computação em nuvem: entendendo e implementando uma nuvem privada / Jorge Luiz corrêa, Marcos Cezar Visoli. - Campinas : Embrapa Informática Agropecuária, 2011.

58p. : il. – (Documentos / Embrapa Informática Agropecuária ; ISSN 1677-9274, 116).

1. Computação em nuvem. 2. IaaS . 3. Openstack . 4. Virtualização. I. Visoli, Marcos Cezar. II. Embrapa Informática Agropecuária. II. Título. III. Série.

006.78 CDD (21. ed.)

Autores

Jorge Luiz Corrêa
Mestre em Ciência da Computação
Analista da Embrapa Informática Agropecuária
Av. André Tosello, 209, Barão Geraldo
Caixa Postal 6041 - 13083-970 - Campinas, SP
Telefone: (19) 3211-5882
e-mail: jorge@cnptia.embrapa.br

Marcos Cezar Visoli
Mestre em Ciência da Computação
Pesquisador da Embrapa Informática Agropecuária
Av. André Tosello, 209, Barão Geraldo
Caixa Postal 6041 - 13083-970 - Campinas, SP
Telefone: (19) 3211-5734
e-mail: visoli@cnptia.embrapa.br

Apresentação

Nas últimas décadas, a computação tem apresentado inovações em diversas áreas. Esse desenvolvimento tem permitido o fornecimento de diferentes tipos de serviços aos usuários. Mais recentemente, a computação em nuvem tem se tornado foco em pesquisas e desenvolvimentos. Esse novo modelo de computação surge em meio a um cenário onde necessidades comerciais influenciam os rumos da computação. No entanto, embora as características mercadológicas sejam preponderantes, essa nova tecnologia pode ser utilizada com diversas outras aplicações.

O uso da computação em nuvem no gerenciamento de infraestrutura altera, diretamente, o modo até então utilizado para realização de atividades de gerência de Tecnologia da Informação (TI). O controle no uso de ativos computacionais, provisionamento de infraestrutura de hardware (processamento e armazenamento), criação e liberação de serviços, gerenciamento de máquinas virtuais e capacidade de aumentar ou diminuir recursos computacionais, tudo sob demanda, são algumas dessas atividades cuja gerência torna-se extremamente versátil. Essa versatilidade fez surgir um novo modelo de negócios baseado na venda de serviços computacionais sob demanda.

Além disso, começam a surgir diversos cenários não comerciais onde as nuvens podem ser empregadas. Um deles é nas atividades de pesquisa e inovação. Nesse contexto, nuvens permitem diversas facilidades que auxiliam pesquisadores de diversas áreas, fornecendo plataformas de desenvolvimento, softwares e, principalmente, infraestrutura de hardware, tudo aliado à rapidez e às facilidades características das nuvens.

Este documento tem por objetivo introduzir os principais conceitos relativos às nuvens computacionais, desde os novos modelos comerciais até sua aplicação em atividades de pesquisa, notadamente na área agropecuária. Serão apresentadas características funcionais, técnicas e alguns cenários onde essa tecnologia pode ser empregada em prol do desenvolvimento de projetos de pesquisa dentro da Empresa Brasileira de Pesquisa Agropecuária.

Kleber Xavier Sampaio de Souza
Chefe-geral
Embrapa Informática Agropecuária

Sumário

Computação em nuvem: entendendo e implementando uma nuvem privada

Jorge Luiz Corrêa
Marcos Cezar Visoli

Introdução

A Embrapa Informática Agropecuária tem como missão viabilizar soluções em tecnologia da informação para o agronegócio. Diversas são as áreas de pesquisas, tais como Novas Tecnologias, Geotecnologias, Modelagem Agroambiental, Inteligência Computacional, Bioinformática, Software Livre, Organização da Informação Eletrônica, Redes de Computadores e Matemática Computacional. Uma característica comum a todas elas é a utilização de recursos computacionais como suporte ao desenvolvimento de suas atividades.

Frente a essa necessidade, o provisionamento de infraestrutura computacional passa a ser bastante relevante. Estabelecer um serviço ou servidor de maneira rápida e eficaz para utilização em um projeto de pesquisa representa um avanço em relação ao suporte prestado e, certamente, à própria atividade de pesquisa. Além dessa característica, as nuvens computacionais trazem também a otimização de uso de recursos computacionais, a versatilidade gerencial e a possibilidade de se utilizar diversos cenários dentro da estrutura da nuvem como o estabelecimento de um cluster computacional e o armazenamento redundante de dados.

Neste documento serão apresentados conceitos introdutórios sobre nuvens computacionais, visando um melhor esclarecimento sobre os pontos em que essa tecnologia pode ser útil no auxílio às atividades desenvolvi-

das na unidade. Além de uma revisão teórica, o documento apresenta os resultados obtidos pela realização de alguns experimentos práticos com o software para infraestrutura de nuvem “OpenStack” . Pretende-se que, ao final do documento, seja atingido um grau de esclarecimento básico acerca dessa tecnologia, de modo que possibilite a proposição de novos cenários onde a computação em nuvem atue como facilitadora da realização de atividades que envolvam tecnologia da informação, seja em tarefas administrativas ou na condução de pesquisas científicas

Conceituação

Durante a evolução da computação nas últimas décadas, diversos modelos de fornecimento de serviços foram utilizados. Em geral, esses modelos foram criados para atender determinadas demandas ou para estender a capacidade técnica das tecnologias em cada época. As redes locais foram estendidas pela interconexão de redes distribuídas, surgiram modelos de distribuição da informação diferentes do modelo cliente/servidor, terminais virtuais deram origem à computação local, sistemas passaram a ser virtualizados para melhor utilização do hardware, dentre diversas tecnologias que visam adaptar o fornecimento de um serviço ou infraestrutura às necessidades de computação de cada momento. A computação em nuvem é, hoje, foco de diversas pesquisas dentro da ciência da computação.

No contexto atual da computação, referindo-se ao fornecimento de serviços, duas características em especial têm impulsionado a existência da computação em nuvem: a disponibilidade e a escalabilidade. Dentro da disponibilidade deve-se considerar não só a capacidade de um sistema ou o fato de o serviço estar sempre disponível aos seus usuários, mas também a capacidade de, rapidamente, tornar esse sistema ou serviço disponível. Já a escalabilidade refere-se à necessidade de um constante crescimento na capacidade de fornecimento de serviços, acompanhando, assim, o crescimento da demanda. Além dessas características consideradas fundamentais, existem outras que refletem o momento atual da computação, como é o caso das questões de negócios referentes à tarifação pelo uso dos recursos computacionais e, até mesmo, questões ambientais,

quando analisando do ponto de vista de otimização do uso dos recursos disponíveis para economia de energia.

Este trabalho tem por objetivo realizar um estudo sobre a computação em nuvem, procurando esclarecer do que se trata, quais os objetivos desse novo conceito, as vantagens e desvantagens de seu uso e suas aplicações. Para tal, além de uma parte conceitual, será estabelecido um laboratório experimental e serão realizados alguns ensaios que auxiliarão na consolidação desses conceitos, bem como tentarão definir facilidades e restrições da utilização das nuvens computacionais. Todo esse trabalho reflete uma visão sobre um cenário atual de computação em nuvem. Devido ao atual estágio de desenvolvimento dessa área, deve-se considerar que alguns conceitos, práticas, softwares e funcionalidades aqui discutidos poderão ser fácil e rapidamente alterados em curto período de tempo posterior a este trabalho.

O que é computação em nuvem

A definição de computação em nuvem não é imediata, estando normalmente influenciada pela área de aplicação. Em outras palavras, a definição se adapta a diversos nichos, de forma que fornecedores de hospedagem de serviços definem de uma forma mais voltada aos seus objetivos, enquanto departamentos de Tecnologia da Informação (TI) definem, sem enfatizar a comercialização de serviços, as capacidades que uma nuvem pode fornecer para o ambiente de TI. Pela necessidade de uma definição mais formal, o National Institute of Standards and Technology (NIST) define a computação em nuvem como sendo *"*um modelo para permitir acesso via rede, conveniente e sob demanda, a um conjunto de recursos computacionais configuráveis e compartilhados (como redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados e liberados com o mínimo de esforço de gerenciamento e interação com o provedor do serviço*"*.

De modo geral, trata-se de um novo paradigma de computação, cujo objetivo é fornecer disponibilidade, escalabilidade, redução de custos, otimização de recursos e versatilidade na implementação de serviços, abstraindo a localização dos recursos computacionais utilizados para prover esses

serviços. Como será abordado mais adiante, um serviço, nesse contexto, pode ser entendido tanto como uma aplicação, tal como um banco de dados, portal web, aplicação de suíte de escritórios como editores de textos, quanto uma infraestrutura de hardware, como um servidor com memória e capacidades de processamento e armazenamento. Uma nuvem compreende um conjunto de hardware gerenciado de tal modo que o estabelecimento de um serviço demandado seja rápido e transparente do ponto de vista de qual hardware está sendo dedicado a esse serviço. Essa é justamente uma das características que possibilitam a otimização da utilização de recursos computacionais. Não há uma relação forte de hospedagem de um serviço em determinado hardware. A nuvem em si pode ser entendida como um repositório de recursos de hardware que serão alocados dinamicamente para os serviços que nela executarão.

Classificação das nuvens quanto ao gerenciamento

Uma das classificações inicias de um sistema de computação em nuvem é baseada em quem é responsável pela gerência da nuvem e a quem ela está acessível. A gerência da nuvem diz respeito ao responsável pela sua manutenção e configuração, inclusive quanto aos ativos de hardware. A acessibilidade refere-se a quem pode acessar os serviços da nuvem. Essas características permitem classificá-las em uma das categorias a seguir.

Nuvem pública

Uma nuvem pública é aquela acessível de qualquer lugar da internet e gerenciada por terceiros. Sua funcionalidade se assemelha a um hospedeiro de serviços, de forma que os clientes pagam para utilizar serviços na estrutura dessa nuvem. O diferencial da computação em nuvem, quando comparada aos serviços de hospedagens comuns, está na abstração da localização do serviço na nuvem. O serviço vendido é alocado dinamicamente dentro dela de forma que, tanto para o cliente quanto para o provedor, independe onde esse serviço esteja fisicamente executando. O objetivo é fornecer ao cliente exatamente o que ele necessita, usando para isso recursos computacionais alocados sob demanda. Como será discuti-

do a seguir, existe um modelo de negócios por trás das nuvens públicas, como é o caso do Amazon Web Services (AWS), baseado na tarifação pelo uso de recursos em uma nuvem. Um exemplo de uso é um cliente com a necessidade de executar um servidor web com banco de dados durante um mês. Este serviço pode ser contratado e será alocado dentro da infraestrutura da nuvem pública, fora do seu ambiente corporativo, sob gerência do fornecedor. Esse, por sua vez, terá a capacidade de alocar tais recursos versátil e dinamicamente para o cliente. Ainda, dado o grau de versatilidade que essa tecnologia oferece, o próprio cliente pode alocar os recursos que necessitar e o provedor apenas o tarifará de acordo com o uso e o tipo de recurso alocado, sem intervir no provisionamento.

Nuvem privada

Uma nuvem privada é aquela cujo acesso é restrito a um conjunto de usuários e toda a infraestrutura pertence à organização onde ela está estabelecida. Em outras palavras, uma nuvem privada não possui acessibilidade pública pela internet, sendo sua gerência de responsabilidade interna e não de um provedor. Normalmente, uma nuvem privada é estabelecida dentro do *datacenter* de uma organização, visando prover serviços para os usuários da própria organização (os clientes da nuvem são internos). Compreende, então, uma maneira de otimizar a disponibilização e utilização de recursos de hardware para o fornecimento de serviços, internamente.

Nuvem híbrida

Uma nuvem híbrida faz uso tanto de uma nuvem pública quanto de uma privada, sendo que o acesso à nuvem pública é diferenciado, caracteristicamente de extensão. Um exemplo de utilização é uma nuvem privada, para disponibilização de serviços internamente, estendida para uma nuvem pública, conectadas por meio de uma *Virtual Private Network* (VPN). Caso a demanda por serviços seja alta para a nuvem privada, seus recursos podem ser ampliados para a nuvem pública, fora do ambiente, a fim de manter a disponibilidade do serviço. Essa é uma das grandes características das nuvens híbridas.

A possibilidade de direcionar parte da carga demandada por algum serviço para uma nuvem pública permite uma redução de custos de hardware. O modelo até então utilizado é tal que uma infraestrutura deve suportar os momentos de pico do serviço. Assim, a compra de hardware para a infraestrutura deve considerar que em tais ocasiões a disponibilidade não seja afetada. Independente do valor do pico de carga de um serviço, pode-se afirmar que ao se adquirir infraestrutura que atenda esses momentos, na maior parte do tempo restante essa infraestrutura ficará ociosa. Assim, além da subutilização de recursos na maior parte do tempo, o custo empregado é elevado.

A Figura 1 apresenta um gráfico que retrata a demanda por infraestrutura ao longo do tempo. As linhas A, B e C indicam casos de uso de uma infraestrutura privada ou pública, de forma que valores acima dessas linhas representam o uso de uma nuvem pública, enquanto valores abaixo dessas linhas representam o uso de uma nuvem privada. Assim, para o caso

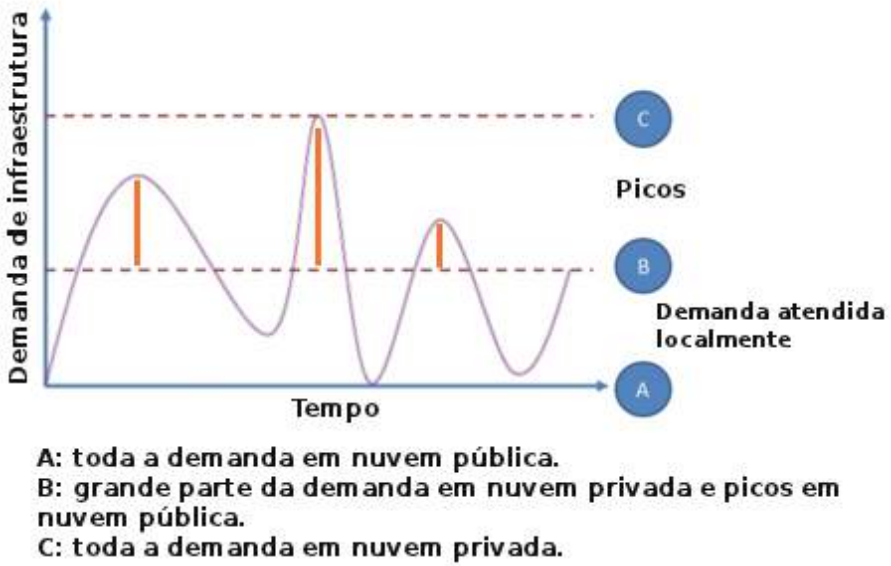


Figura 1. Demanda de infraestrutura ao longo do tempo: nuvens híbridas permitem que a infraestrutura privada não seja superdimensionada, resultando em ociosidade e alto custo.

Fonte adaptada de Amazon Enables Bursting to the Public Cloud (2009).

'A', toda a infraestrutura utilizada é pública. Para o caso 'B', a parte inferior indicada por 'Demanda atendida localmente' faz uso de uma nuvem privada enquanto os 'Picos', parte superior, são executados em uma nuvem pública. No caso 'C', toda a demanda é atendida por uma nuvem privada.

A possibilidade de se estender uma nuvem privada utilizando uma nuvem pública, aliada ao modelo de negócios com tarifação por uso, discutido em seções seguintes, permite que uma infraestrutura local não seja construída tomando por base os momentos de pico dos serviços. Torna-se possível manter a disponibilidade sem a necessidade de grandes investimentos em infraestrutura, pagando-se apenas pelo fornecimento de capacidade extra quando esta é necessária.

Classificação dos serviços em nuvem (IBM, 2010)

Como mencionado anteriormente, o conceito de serviço no contexto de uma nuvem computacional é mais amplo do que o comumente utilizado na internet. Sempre que o termo serviço for utilizado ele deve ser relacionado com alguma das categorias a seguir.

***Software as a Service* (SaaS)**

O serviço fornecido por esse tipo de nuvem é um software cujo uso é baseado em web. É o tipo mais comum de serviço baseado em nuvem. O exemplo mais conhecido é o *Google Docs*, serviço que provê uma suíte de utilitários de escritório acessados por meio de um *frontend* web.

***Platform as a Service* (PaaS)**

O serviço entregue é uma plataforma de desenvolvimento de softwares e produtos baseada em web. São fornecidas APIs (*Application Programming Interfaces*) e GUIs (*Graphical User Interfaces*) a fim de facilitar toda a criação de um software, envolvendo todas as suas etapas como *design*, desenvolvimento, testes, controle de versões, entre várias outras características. Todas essas funcionalidades são vistas como uma solução inte-

grada acessada pela web. As ferramentas desenvolvidas são executadas pelo provedor de serviço PaaS.

Infrastructure as a Service (IaaS)

O serviço entregue é infraestrutura computacional bruta, tal como instâncias de servidores virtuais, armazenamento, um banco de dados ou algum serviço semelhante, possibilitando que o usuário a customize a fim de adequá-la às suas necessidades. Esse tipo de serviço permite, por exemplo, que um usuário adquira rapidamente uma instância de computação, para execução de qualquer tipo de aplicação, sem se preocupar com a gerência do hardware, ou mesmo um sistema de arquivos com determinada capacidade, de forma que tarefas como a replicação de informação para *backup* ficam totalmente sob responsabilidade da nuvem.

A Figura 2 resume essa classificação mostrando alguns exemplos oferecidos atualmente para cada tipo de serviço.

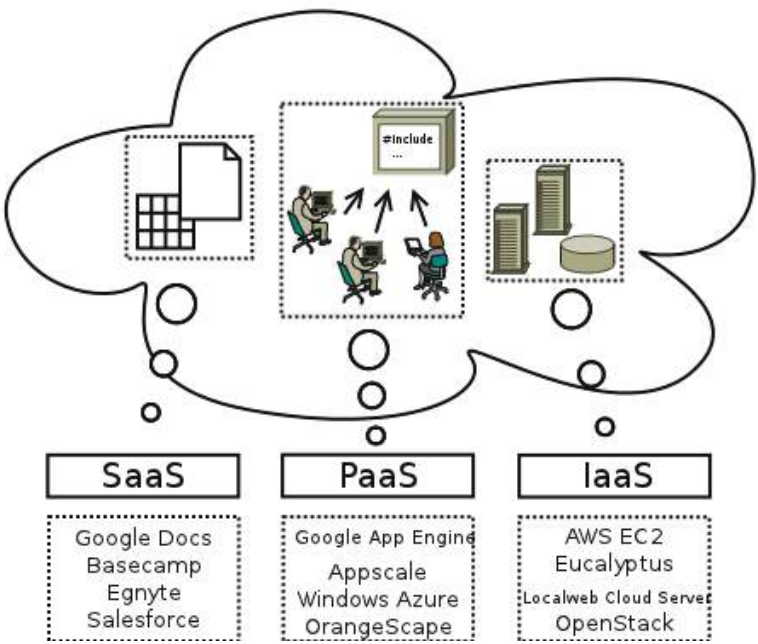


Figura 2. Classificação dos serviços em nuvem.

Modelo de negócios baseado em computação em nuvem

Existe, atualmente, um modelo de negócios por trás das nuvens públicas baseado na venda de serviços que são alocados dentro da nuvem. As diferenças para a venda de serviços fora da nuvem é que esses serviços podem ser contratados temporariamente e possuem elasticidade de capacidades.

Até então, ao se contratar a hospedagem de algum serviço, como um *host* virtual ou um banco de dados, pagava-se de acordo com planos anuais ou relacionados com alguma métrica, tal como a capacidade de processamento e memória do *host* contratado, a quantidade de espaço para armazenamento, entre outras. Caso o cliente não utilizasse esses recursos durante o ano todo, eles ficavam ociosos. Os serviços em nuvem permitem que o cliente utilize os recursos por períodos de tempo, de acordo com o que ele necessita. Normalmente, os serviços são tarifados por horas de uso. Desta maneira, o cliente acaba não pagando pela ociosidade, mas apenas pelo tempo em que efetivamente utiliza os recursos.

Ainda, muitas vezes, o cliente pagava por determinado recurso, mas não o utilizava em sua totalidade. O dimensionamento de recursos era difícil, uma vez que o cliente deveria prever o fornecimento do serviço nas situações de sobrecarga e, como essas situações não representam a maior parte do tempo de uso, o cliente acabava subutilizando o serviço contratado. A elasticidade é a característica que permite o ajuste de capacidades dos recursos de maneira bem versátil. Se um cliente necessita de um serviço o ano todo, mas possui demanda alta apenas durante um mês no ano, é possível que ele aumente a capacidade da infraestrutura apenas no mês de sobrecarga. Assim, ele pagará pelo baixo uso durante todo o ano e pelo alto uso apenas durante o mês discrepante. Não há desperdício financeiro nem de recursos computacionais. Esse modelo é chamado de “pague pelo uso”.

Principais provedores de serviços em nuvem

Existe, atualmente, uma grande quantidade de provedores de serviços em nuvem. Isso é explicado pelo investimento que os provedores de hospeda-

gem fizeram e estão fazendo no sentido de capacitar suas estruturas para fornecimento de serviços utilizando essa tecnologia. O que deve ser salientado é que esses provedores podem ser classificados segundo as categorias de serviços fornecidos, conforme discutido na seção "Classificação dos serviços em nuvem". Assim, alguns desses provedores estão voltados ao fornecimento de serviços como infraestrutura, outros como softwares e outros como plataforma. Existe ainda o caso de provedores que fornecem mais de um tipo de serviço na nuvem, de acordo com suas estruturas. Nesta seção serão apresentados alguns provedores e as características de seus serviços, com o intuito de exemplificar esse novo modelo de negócios.

Amazon Web Services (*AWS*)

A empresa Amazon oferece uma diversidade de serviços em nuvem, sendo que os mais conhecidos são o *Elastic Compute Cloud* (EC2) e o *Simple Storage Service* (S3). O EC2 provê infraestrutura para os clientes na forma de capacidade computacional reconfigurável. Um cliente da EC2 pode escolher as configurações de sua infraestrutura, tal como capacidade dos *hosts* e quais os sistemas operacionais. Já o S3 é um serviço de armazenamento acessado por meio de uma interface web.

A tarifação praticada pela Amazon é um grande exemplo do modelo "pague pelo uso". Em suas políticas, a empresa descreve o *Service Level Agreement* (SLA), direitos e deveres, mas sem exigir nenhum tipo de contrato. Basta ser um usuário cadastrado, com um cartão de crédito, para que a cobrança pelo serviço solicitado seja possível. O usuário escolhe os seus serviços em um sistema web, sendo taxado apenas pelo que utilizou. A tarifação praticada no EC2 é baseada em hora por instância. Uma instância é um tipo de serviço solicitado, como um computador com determinada configuração. Por exemplo, a Figura 3 descreve os valores e os tipos de instâncias que o cliente pode solicitar.

Como pode-se observar, o EC2 fornece *hosts* com sistemas Linux/Unix e Windows, categorizados em *Small, Large e Extra Large*. Essas categorias se diferenciam pela capacidade de processamento e de memória de cada *host*. Existe ainda uma categoria *Micro* que permite a instanciação de *hosts* com baixa capacidade, para pequenos testes. As demais categorias,

Region: US East (Virginia)

	Linux/UNIX Usage	Windows Usage
Standard On-Demand Instances		
Small (Default)	$0.085 per hour	$0.12 per hour
Large	$0.34 per hour	$0.48 per hour
Extra Large	$0.68 per hour	$0.96 per hour
Micro On-Demand Instances		
Micro	$0.02 per hour	$0.03 per hour
Hi-Memory On-Demand Instances		
Extra Large	$0.50 per hour	$0.62 per hour
Double Extra Large	$1.00 per hour	$1.24 per hour
Quadruple Extra Large	$2.00 per hour	$2.48 per hour
Hi-CPU On-Demand Instances		
Medium	$0.17 per hour	$0.29 per hour
Extra Large	$0.68 per hour	$1.16 per hour
Cluster Compute Instances		
Quadruple Extra Large	$1.60 per hour	N/A*
Cluster GPU Instances		
Quadruple Extra Large	$2.10 per hour	N/A*
* Windows® is not currently available for Cluster Compute or Cluster GPU Instances		

Figura 3. Tarifação da Amazon para EC2.

Fonte: Amazon (2011a).

Hi-Memory, Hi-CPU e *Cluster* representam serviços de infraestrutura para clientes que requerem altas performances.

Para o serviço S3, a Amazon tarifa de acordo com a quantidade de dados armazenada. A Figura 4 mostra a tarifação praticada. A coluna Reduced Redundancy Storage é um serviço de armazenamento para dados não críticos, sendo, portanto, mais barato que o padrão do S3 que fornece redundância de dados por padrão.

Region: US Standard

	Standard Storage	Reduced Redundancy Storage
First 1 TB / month	$0.140 per GB	$0.093 per GB
Next 49 TB / month	$0.125 per GB	$0.083 per GB
Next 450 TB / month	$0.110 per GB	$0.073 per GB
Next 500 TB / month	$0.095 per GB	$0.063 per GB
Next 4000 TB / month	$0.080 per GB	$0.053 per GB
Over 5000 TB / month	$0.055 per GB	$0.037 per GB

Figura 4. Tarifação da Amazon para S3.

Fonte: Amazon (2011b).

Google (*App Engine* e *Docs*)

O Google é outro importante provedor de serviços em nuvem, destacando-se pelo fornecimento de SaaS e PaaS. Como SaaS possui a conhecida família de softwares *Google Docs*, capaz de prover uma suíte completa de ferramentas de escritório. Como PaaS disponibiliza o *Google App Engine*. Esse serviço fornece uma plataforma para que o usuário desenvolva seus aplicativos e os executem na própria infraestrutura do Google. Embora seja um serviço do tipo PaaS, após desenvolvida uma aplicação, o usuário contará com serviços semelhantes aos fornecidos como IaaS, ou seja, não deverá se preocupar com *hosts* para hospedagem de sua aplicação nem com questões de escalabilidade. O modelo utilizado pelo Google é o "pague pelo uso", sendo que o uso inicial é gratuito e a taxação ocorre somente após o usuário atingir determinadas métricas relativas à quantidade de armazenamento e tráfego de rede (GOOGLE, 2011). O uso gratuito inclui 500 MB de espaço para armazenamento, processamento e largura de banda consideradas suficientes para suportar uma aplicação com até 5 milhões de visualizações de páginas por mês.

Windows Azure

O Windows Azure é uma plataforma da Microsoft (PaaS) que permite aos usuários o desenvolvimento de aplicações no Visual Studio, .NET Framework, além de PHP, Java e Ruby, sem se preocupar com questões sobre o gerenciamento de rede e infraestrutura na qual essas aplicações

executarão. O Windows Azure “oferece aos desenvolvedores a funcionalidade de compilar, hospedar e gerenciar aplicativos completos” utilizando uma nuvem da própria Microsoft (MICROSOFT, 2011).

Assim como esses provedores, mais conhecidos no contexto da computação em nuvem, vários outros fornecem serviços semelhantes, como é o caso da IBM, HP, Rackspace, RedHat, Oracle, Salesforce e outros.

Nuvem privada provendo serviço de infraestrutura (IaaS)

A computação em nuvem criou uma série de oportunidades além do modelo de negócios discutido. Embora seja uma tendência comercial para provedores de serviços na internet, ela tem possibilitado que departamentos de TI revejam suas estruturas computacionais. O uso de uma nuvem privada, provendo infraestrutura como serviço (IaaS) dentro de um departamento de TI, permite que o uso dos recursos computacionais sejam otimizados, bem como facilita o gerenciamento dos serviços. Deste modo, além de ser explorado comercialmente por diversas empresas que vendem serviços na nuvem, esse paradigma representa uma nova maneira de se fornecer serviços dentro de uma organização, utilizando sua própria estrutura de TI. Para esse objetivo, existem conjuntos de ferramentas capazes de prover todas as funcionalidades necessárias para o estabelecimento de uma nuvem privada.

Analisando as ferramentas *Open Source*, de uso gratuito, quatro delas se destacam e são adotadas como referências na comunidade de software livre: Euclyptus (EUCALYPTUS, 2011), OpenNebula (OPENNEBULA, 2011), Nimbus (NIMBUS, 2011) e OpenStack (OPENSTACK, 2011). Essas ferramentas permitem que uma nuvem privada seja estabelecida com diversos recursos. No entanto, cada uma possui determinadas características que permitem diferenciá-las de acordo com o perfil de uso da nuvem privada a ser criada.

Uma comparação detalhada entre as três primeiras pode ser observada em (SEMPOLINSKI; THAIN, 2010). Segundo esses autores, alguns fatores são muito importantes nas análises desses conjuntos de ferramentas, entre eles a capacidade de customização da nuvem, o grau de transparência na interface com o usuário e a distância existente na interação do usuário

com o administrador da nuvem. Por capacidade de customização deve-se entender a flexibilidade que cada conjunto de ferramentas dá ao administrador para poder configurar a nuvem de modo que atenda todas as suas necessidades. O grau de transparência diz respeito ao nível de abstração do uso da nuvem em relação aos usuários. Para o usuário final dos serviços que estão na nuvem, quanto mais abstrata esta parecer, melhor será o serviço, visto que muitos usuários não querem nem necessitam conhecer detalhes estruturais e técnicos da infraestrutura. E, quanto à distância entre o usuário e o administrador, os autores consideram que a utilização desses conjuntos de ferramentas para o estabelecimento de uma nuvem privada deve aproximar o usuário do administrador de modo a facilitar o fornecimento de serviço, estabelecendo certo nível de confiança entre as duas partes, confiança esta que não existe no contexto da comercialização de serviços (nuvens públicas). Tomando esses fatores como base e outras características como a filosofia do conjunto de ferramentas, aspectos sobre a configuração de rede do ambiente, aspectos de segurança interna e do usuário, tais autores expressaram o que consideram o cenário ideal para cada um desses conjuntos de ferramentas. Essas informações são mostradas na Tabela 1.

Como é possível observar, o cenário considerado ideal para o uso do Eucalyptus é aquele com um grande grupo de máquinas e um grupo de usuários relativamente confiáveis. Essa confiabilidade está pautada na forma de uso do sistema. Por exemplo, usuários internos de uma organização são considerados mais confiáveis que os usuários públicos de uma nuvem onde os serviços são vendidos. É mais provável que abusos ocorram na infraestrutura de uma nuvem pública do que em uma nuvem privada, pois nesta última é menos comum a presença de um usuário malicioso. No caso do OpenNebula, a indicação é para um grupo menor de máquinas e usuários altamente confiáveis. Esta última recomendação ocorre devido ao modo como o conjunto de ferramentas do OpenNebula opera, principalmente referente aos níveis de acesso nas máquinas, de forma que, se um usuário não é confiável, ele pode representar alto risco para toda a infraestrutura da nuvem. No caso do Nimbus, a indicação é para um ambiente totalmente voltado à pesquisa científica. Uma de suas vantagens é utilizar uma estrutura de certificação digital, garantindo maior controle sobre os usuários.

Além desses três conjuntos de ferramentas bem conhecidos na comunidade *open source*, existe um quarto, denominado OpenStack, que será

Tabela 1. Comparação entre os conjuntos de ferramentas Eucalyptus, OpenNebula e Nimbus.

	Eucalyptus	OpenNebula	Nimbus
Philosophy	Mimic Amazon EC2	Private, highly customizable cloud	Cloud resources tailored to scientific researchers
Customizability	Some for admin, less for user	Basically everything	Many parts except for image storage and globus credentials
DHCP	On cluster controller	Variable	On individual compute node
Internal Security	Tight. Root required for many things	Looser, but can be made more tight if needed	Fairly tight, unless deploying a fully private cloud.
User Security	Users are given custom credentials via a web interface	User logs into head (unless optional front-end used)	Users x509 credential is registred with cloud
An Ideal Setting	Large group of machines for buch of semi-trusted users	Smaller group of machines for highly trusted users	Deploy for less to semi-trusted users family with x509
Network Issues	Dhcpd on cluster controller	Admin must set manually but has many options	Dhcpd on every node and Nimbus assigns MAC

Fonte: Sempolinski e Thain (2010), adaptada pelo autor.

discutido mais adiante. Trata-se de um conjunto de ferramentas que tem ganhado notoriedade, principalmente com sua adoção como plataforma base de nuvem computacional na distribuição Ubuntu.

Portanto, a opção por um ou outro conjunto de ferramentas depende da análise de diversos aspectos. Dentro dos aspectos técnicos, deve-se analisar, principalmente, qual a dimensão da utilização dos serviços em nuvem, o nível de administração que se deseja e qual o perfil dos usuários, mesmo dentro de uma organização.

Entendendo melhor o interior da nuvem

Os conjuntos de ferramentas apresentados anteriormente permitem a criação de uma nuvem computacional dentro do modelo IaaS. O termo

"conjunto de ferramentas" busca deixar claro que não se trata de apenas uma ferramenta, capaz de prover todos os recursos necessários. Nesse contexto, como bem definem Sempolinski e Thain, trata-se de uma pilha de ferramentas cuja interação é essencial para o resultado final. Assim sendo, é possível identificar nas ferramentas de código aberto seis componentes básicos: hardware e sistema operacional, rede, monitor de máquina virtual, imagem de disco virtual, o *frontend* com o usuário e o próprio *framework* de nuvem.

Uma vez que a nuvem utilizará instâncias de máquinas virtuais, é importante verificar o tipo de hardware empregado para se determinar o tipo de virtualização (CINTRA, 2010). Dependendo do processador disponível, é possível utilizar virtualização total, auxiliada por hardware, ou empregar paravirtualização.

A rede refere-se ao modo como esta é configurada dentro de cada ambiente. Cada um dos conjuntos de ferramentas citados possui determinadas características de configuração. No entanto, é importante entender como ocorre a distribuição de endereços IPs para as máquinas físicas, como é o controle de endereço físico das máquinas virtuais (*MAC Address*) e como ocorre a resolução de nomes (DNS).

O monitor de máquina virtual, também conhecido como *hypervisor*, diz respeito a qual sistema de hospedagem de máquinas virtuais será utilizado. Essa escolha não é livre e cada um dos conjuntos de ferramentas citados tem suporte a determinados monitores de máquinas virtuais.Por exemplo, o Eucalyptus e o OpenNebula que possuem suporte a Xen, KVM e VMWare, enquanto o Nimbus apenas suporta Xen e KVM. Neste ponto é importante ficar clara a distinção entre os conceitos de virtualização e nuvem computacional. A virtualização é uma tecnologia utilizada para o estabelecimento de uma nuvem, não sendo, portanto, sinônimos. Uma nuvem não é estabelecida por um único software capaz de prover todas as funcionalidades esperadas, mas sim por um conjunto de ferramentas das quais a virtualização é apenas uma dessas ferramentas. Ainda vale considerar que, embora as descrições desses conjuntos de ferramentas afirmem haver suporte para diversos monitores de máquinas virtuais, muitas vezes a documentação não mostra como a implementação (configuração) deve ser realizada para se utilizar determinado monitor.

As imagens de disco virtual possuem toda a instalação de um sistema operacional e dão origem a uma máquina virtual instanciada na nuvem.

Em uma ocasião de uso comum, o disco virtual é criado durante o processo de criação da própria máquina virtual. No entanto, para uma nuvem computacional, diversas máquina virtuais serão instanciadas e descartadas a qualquer momento, tornando-se inviável que esses discos virtuais sejam criados neste momento e o sistema operacional instalado para que a instanciação ocorra. Assim, os *frameworks* de computação em nuvem normalmente trabalham com um repositório de discos virtuais. Ao ser requisitada uma máquina virtual, uma imagem do repositório é copiada para a máquina física onde a instância executará e determinadas modificações são realizadas, como por exemplo a adição de uma área de troca para memória virtual (*swap*). A imagem do repositório é chamada de modelo (*template*) enquanto a imagem adaptada, que executa na máquina física, é chamada de imagem de tempo de execução (*runtime image*).

O *frontend* é o ponto pelo qual o usuário acessa a parte de fornecimento de serviços da nuvem. É a partir dele que o usuário pode requisitar uma máquina, podendo especificar parâmetros de configuração. Ele também é responsável por realizar controle de credenciais do usuário e gerenciar o uso dos recursos da nuvem, controlando as requisições de acordo com a capacidade física da nuvem.

Por fim, o último componente é o próprio *framework*, responsável por manter a interação de todos os outros componentes. É ele quem recebe as entradas do *frontend*, obtém a imagem do repositório de discos virtuais, define a máquina física onde essa instância será alocada, avisa o monitor de máquina virtual desse nó para iniciar uma máquina virtual e passa informações para a configuração de rede para a máquina virtual sendo instanciada.

A Figura 5, retirada de (SEMPOLINSKI; THAIN, 2010), mostra a arquitetura de um sistema genérico de nuvem computacional, destacando cada um dos componentes anteriormente discutidos.

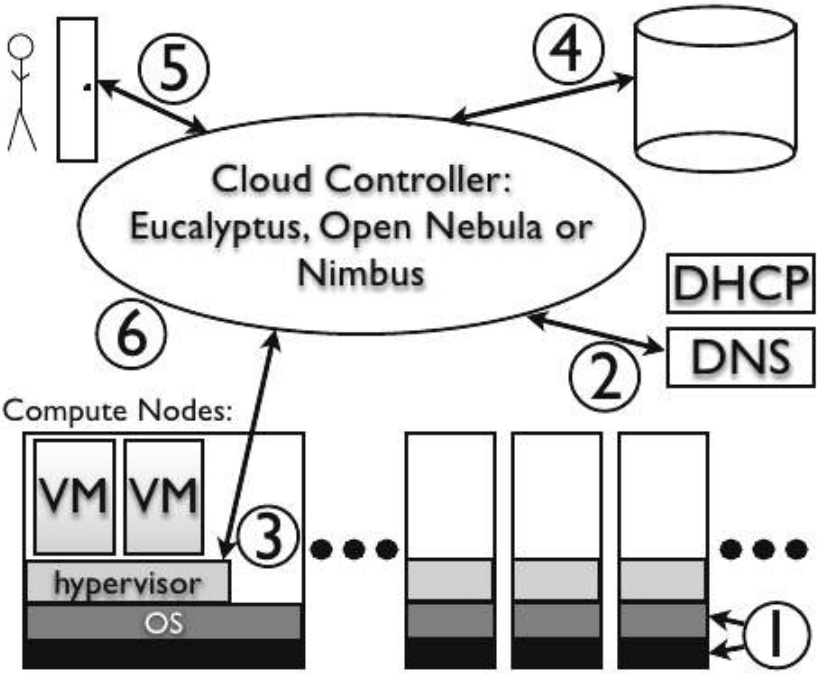


Figura 5. Componentes de uma arquitetura genérica de computação em nuvem: (1) hardware e sistemas operacionais; (2) rede; (3) monitor de máquina virtual; (4) imagens de discos virtuais; (5) frontend; (6) o próprio *framework*.

O conceito de Service Orchestration

Os *frameworks* até então discutidos são capazes de prover as funcionalidades para o estabelecimento de uma nuvem IaaS. Esse tipo de serviço tem por objetivo principal prover infraestrutura para os usuários e não aplicações *online*. Por exemplo, se um usuário possui a demanda por um servidor com uma capacidade relativamente grande para executar uma aplicação por um dia, tem-se um cenário ideal, cuja resolução se dará de maneira rápida utilizando uma nuvem IaaS. No entanto, os *frameworks* analisados não tangem a aplicação do usuário em si. A nuvem proporcionará a infraestrutura rapidamente, de forma que o usuário não deverá se preocupar com a aquisição de um computador, alocação de espaço físico para posicioná-

-lo e instalação do sistema operacional. No entanto, fica ao seu encargo a configuração das aplicações, mesmo que sejam aplicações básicas.

É nesse contexto que o conceito de *Service Orchestration* ganha relevância. No exemplo anterior, o usuário obteve a parte de infraestrutura e deveria configurar sua aplicação. Tomando como exemplo que essa aplicação utilize um banco de dados, o conceito de orquestração de serviços surgiu justamente para facilitar esta implantação. Utilizando um framework de orquestração de serviços torna-se mais fácil a configuração e implantação deles, fazendo com que serviços/aplicações (como servidores web, banco de dados, sistemas de gerenciamento de conteúdo, etc) acompanhem a versatilidade do provisionamento de infraestrutura na nuvem IaaS. Assim, em uma nova leitura, um usuário poderia solicitar para a nuvem um servidor que possua um banco de dados e esse provisionamento ocorreria de forma direta, sem o usuário ter que se preocupar com a configuração do sistema gerenciador de banco de dados.

Um exemplo de *framework* de orquestração de serviços em constante avanço é o Ensemble (ENSEMBLE, 2011), cujo nome será futuramente alterado para *Juju*. Esse projeto, mantido pela Canonical, responsável pelo desenvolvimento do Ubuntu Linux, é baseado no uso de fórmulas. As fórmulas são conjuntos de instruções utilizados não só para a implantação de um serviço, mas para seu gerenciamento de modo geral. Por exemplo, existem fórmulas para o estabelecimento de um banco de dados PostgreSQL e para o gerenciador de conteúdos Drupal. As fórmulas permitem também que esses dois serviços sejam interligados de maneira simples e prática.

Enquanto as nuvens IaaS são inicialmente entendidas como facilitadoras do provisionamento de infraestrutura de TI, a orquestração de serviços pode ser entendida como uma facilitadora da gerência da nuvem, permitindo que melhores práticas sejam compartilhadas e reusadas na nuvem. O próprio Ensemble conta com um repositório de fórmulas prontas desenvolvidas por profissionais de desenvolvimento e operação (*DevOp*). Essas fórmulas podem ser utilizadas diretamente ou alteradas para refletir determinadas necessidades. Com tais funcionalidades é possível otimizar ainda mais o tempo, pois, além do pronto estabelecimento de uma infraestrutura computacional, tal como um servidor, pode-se, com a mesma versatilidade, estabelecer determinados serviços dentro dessa própria infraestrutura de nuvem.

Computação em nuvem e a TI Verde

Além de todas as características técnicas que a computação em nuvem pode proporcionar para gestores de TI e usuários, é conveniente lembrar que existe uma relação entre ela e o conceito de TI Verde. A TI Verde reflete a preocupação com o desenvolvimento sustentável das empresas, considerando as relações entre a tecnologia e seus produtos com o meio ambiente, considerando desde materiais utilizados para produção de componentes, descartes, reciclagem, a matéria prima utilizada para fabricação desses produtos, até a poluição gerada nesse processo e o impacto do consumo de energia, muito comum se tratando de dispositivos eletrônicos. Pautar as estratégias de negócio tendo a sustentabilidade como um dos quesitos básicos tem se tornado quase uma obrigação para diversas empresas. E, no âmbito da Empresa Brasileira de Pesquisa Agropecuária (Embrapa), tal característica é tida como essencial, visto que a sustentabilidade, em suas atividades, é considerada como sua missão.

No entanto, executar suas atribuições sustentavelmente requer adequações e controle para que esse requisito não passe a ser apenas teórico. Como bem destaca Torres (2008), “a implantação de Sistemas de Gestão Ambiental por parte das organizações deve ser feita considerando-se, principalmente, os recursos e estruturas existentes para que não fique em descompasso com a sua realidade e seja simplesmente uma mera declaração de intenções”. Em outras palavras, deve ser um processo não abrupto e que não interfira drasticamente no andamento das atividades de uma organização.

Por tais considerações, pode-se afirmar que a computação em nuvem é uma das maneiras pela qual a tecnologia da informação pode se tornar mais sustentável. Primeiramente pela otimização no uso dos recursos computacionais, de modo a evitar ociosidade e por conseguinte evitar o desperdício de hardware. Por outro, reflete diretamente no consumo de energia. A computação em nuvem permite que *datacenters* sejam reorganizados de modo diminuir a quantidade de máquinas e, consequentemente, a energia gasta com sistemas de refrigeração. Além disso, existem atualmente tecnologias capazes de otimizar ainda mais esses usos. As placas-mãe mais atuais possuem BIOS capazes de aceitar conexões remotas e efetuar o desligamento ou a ligação de um *host*. Aliando-se tal tecnologia

com a característica de elasticidade, é possível que em situações de ociosidade dentro da nuvem (o que já é um ganho em relação à ociosidade fora da nuvem) seja possível desligar máquinas que não estejam sendo utilizadas e religá-las quando sob novas demandas.

Tendências *open source* e projetos Canonical

Considerando a comunidade *open source* de desenvolvimento, é comum se encontrar vários projetos e ferramentas para um mesmo nicho de aplicação. Essa concorrência, de certo modo, faz com que essas ferramentas evoluam mais rapidamente. No caso das nuvens computacionais, os softwares utilizados para sua criação e gerência, denominados pilha de softwares (*stack*) uma vez que não é apenas um único programa, mas um conjunto que opera colaborativamente, também se enquadram nesse cenário. São vários os conjuntos de softwares que se pode utilizar para o estabelecimento de uma infraestrutura como serviço (IaaS), alguns deles citados anteriormente.

A escolha por um ou outro, muitas vezes, é pautada nas características técnicas, de forma que o escolhido possui todas as funcionalidades requisitadas pelo usuário. No cenário das nuvens computacionais, pelo menos no cenário atual, essa definição de qual conjunto de softwares utilizar entra, em certo ponto, na esfera política da comunidade *open source*. Embora existam vários conjuntos de softwares, grande parte da evolução de cada uma dessas ferramentas ainda está por vir. Trata-se de um campo bastante novo em que grande parte das funcionalidades estão em desenvolvimento e ganharão proporção nos próximos anos. Logo, algumas considerações além das características técnicas devem ser ponderadas para a seleção de um conjunto de ferramentas para estabelecer uma nuvem privada. Uma dessas considerações é uma visão do atual estágio de desenvolvimento das distribuições Linux.

Não existe, atualmente, uma maneria de se determinar qual a distribuição Linux mais utilizada. O que existe são informações que podem refletir a popularidade no uso de cada distribuição. O *Google Insights* é um exemplo que possibilita o acesso a esse tipo de informação, mostrando qual

o volume de buscas ocorreu para um determinado termo na base do Google. Comparando o comportamento nas buscas pelos nomes de cinco distribuições bem conhecidas, estimadas como as mais utilizadas atualmente, pode-se verificar que o Ubuntu Linux figura como a mais popular. A Figura 6 mostra a comparação entre as distribuições Ubuntu, Debian, Mint, CentOS e Slackware (constantes em várias pesquisas como as mais utilizadas atualmente). Os valores das ordenadas representam uma porcentagem em relação ao número total de buscas considerando todos os termos (valor normalizado). Os "Totais" indicam uma média para o período considerado.

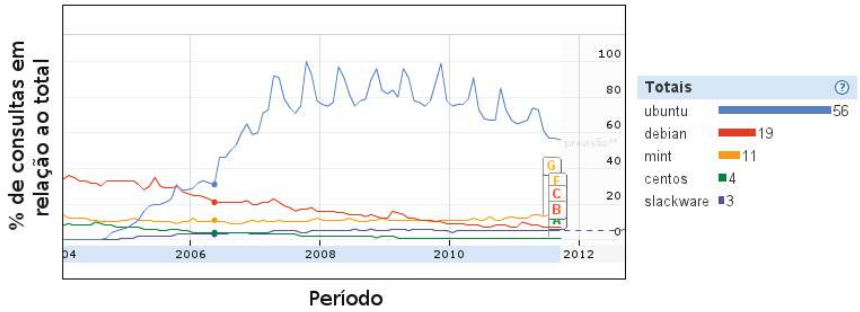


Figura 6. Informações relativas à popularidade das distribuições indicadas, segundo o volume de pesquisas no site de busca Google (gráfico gerado pelo Google Insights). Ubuntu é a distribuição mais popular.

Sendo o Ubuntu a distribuição mais popular atualmente, convém, nesse ponto, analisar alguns fatores relativos ao seu desenvolvimento, fatores estes que fazem parte da esfera política mencionada e são importantes na decisão de qual pilha de softwares utilizar. Embora exista toda uma comunidade colaborativa, empenhada nas melhorias do Ubuntu, seu desenvolvimento é mantido pela Canonical. Trata-se de uma empresa que, conforme consta em sua apresentação, "trabalha com a comunidade *open source* para entregar o Ubuntu" gratuitamente, além de vender serviços que "auxiliam pessoas e negócios a reduzirem custos ao redor do mundo, aumentar a eficiência e elevar a segurança" (CANONICAL, 2011). A Canonical é responsável também por diversos outros projetos de softwares, tanto livres, como o Ensemble, quanto comerciais, como o Landscape, um gerenciador

de *hosts*. Nesse sentido, existe por parte dela, um esforço e foco em determinados softwares, o que pode ser considerado um respaldo de qualidade quando analisando a evolução do Ubuntu nos últimos anos. Entre esses softwares está, por exemplo, o KVM (KVM, 2011), tomado como o sistema padrão de virtualização do Ubuntu.

Assim como o KVM, a Canonical dedica esforços em outros softwares que ela considera como o padrão em determinada área dentro da distribuição Ubuntu. No caso das pilhas de softwares para construção de nuvens privadas, a Canonical adotava o Eucalyptus como pilha padrão. No entanto, por questões gerenciais, de compatibilidade de métodos de desenvolvimento de projetos e também técnicas, a partir da versão 11.10, a Canonical adotará o OpenStack como pilha padrão para nuvem. Além disso, projetos relacionados, como o Ensemble (Juju), terão seu foco voltado para a compatibilidade com o OpenStack.

Por todas essas considerações, tanto no contexto político quanto no técnico, fica definido que o presente trabalho utilizará o Ubuntu Linux, sendo a nuvem IaaS baseada no OpenStack. Além disso, o KVM será utilizado como *hypervisor* devido tanto à sua tendência de uso com o OpenStack quanto a este ser o *hypervisor* padrão da Canonical adotado no Ubuntu. Pretende-se que, com tal decisão, este trabalho fique alinhado ao que a Canonical está por desenvolver e, consequentemente, com a evolução que os softwares por ela adotados como padrão no Ubuntu venham a trilhar.

Aspectos práticos: implementação e testes

Os aspectos práticos envolvem toda a parte do estabelecimento do ambiente de nuvem, desde as definições de sistemas operacionais e ferramentas até a instalação e configuração de cada item da pilha de softwares. Por fim, alguns testes serão realizados com a estrutura montada a fim de esclarecer como a nuvem se comportaria perante alguns cenários de demanda de serviços.

OpenStack

O OpenStack é um conjunto de ferramentas livres que possibilita a criação de uma nuvem computacional IaaS, incluindo serviços de armazenamento de dados. Os maiores contribuidores, para que o OpenStack atingisse o grau de desenvolvimento atual, foram a NASA e a empresa Rackspace. A partir dessas duas entidades formou-se o OpenStack e, posteriormente, um consórcio que conta com diversos outros participantes, como a International Business Machines (IBM) , Dell, Citrix e Canonical.

O OpenStack é organizado em três famílias de serviços: Compute Infrastructure (Nova), Storage Infrastructure (Swift) e Imaging Service (Glance). Cada uma dessas famílias possui um ou mais utilitários responsáveis por desempenharem determinada função dentro da nuvem.

Compute Infrastructure (Nova)

Esse conjunto de ferramentas é o controlador da nuvem. Todas as atividades necessárias para que um instanciamento de máquina virtual ocorra são gerenciadas por essa entidade como, por exemplo, recursos computacionais, rede, autorização e escalabilidade. No entanto, trata-se de uma plataforma de gerenciamento, não existindo nenhuma capacidade de virtualização. Todo o processo de virtualização no OpenStack é possibilitado por algum dos *hypervisors* suportados, de modo que a gerência do processo de virtualização é executado na plataforma Nova. Essa gerência é baseada na API da biblioteca *libvirt*, utilizada por diversos sistemas de virtualização. Segundo a documentação atual do OpenStack, são suportados o Xen, XenServer, KVM, UML e Hyper-V. Dentro do conjunto OpenStack Infrastructure Compute, tem-se as seguintes ferramentas:

API Server (nova-api)

O *API Server* é responsável por traduzir as chamadas do mundo externo, como requisições de usuários, para ações dentro da nuvem, permitindo que atividades de gerenciamento sejam realizadas. Esse gerenciamento é realizado por meio de chamadas de Web Services utilizando a API EC2,

como nos sistemas da Amazon. O *API Server* se comunica, então, com outros componentes da infraestrutura da nuvem utilizando um modelo de troca de mensagens implementado pelo *Message Queue*.

Message Queue (rabbit-mq server)

A comunicação entre os componentes de controle da infraestrutura de nuvem do OpenStack ocorre pelo protocolo AMQP (*Advanced Message Queue Protocol*). Esse componente controla, portanto, a troca de mensagens entre as outras ferramentas do OpenStack que operam colaborativamente para o gerenciamento da nuvem.

Compute Workers (nova-compute)

Esse componente trata do gerenciamento do ciclo de vida das instâncias. Sua função é receber mensagens de gerenciamento de ciclo de vida de instâncias, pelo Message Queue, e realizar as operações necessárias. Dentro de uma infraestrutura de nuvem OpenStack podem existir diversos Compute Workers. Em suma, trata-se do controle principal que deve existir em cada nó de processamento da infraestrutura. Uma instância poderá ser executada em qualquer *host* que possua esse componente. A escolha por qual *host* da infraestrutura executará uma instância dependerá do algoritmo de escalonamento sendo utilizado no *Scheduler*, outro componente discutido a seguir.

Network Controller (nova-network)

Trata da configuração de rede das máquinas da infraestrutura, realizando ações como a alocação de endereços IP para as máquinas virtuais, configurações de *Virtual Local Area Network* (VLAN) para os chamados projetos que executam na nuvem, implementação de grupos de segurança e configurações de redes para os nós da nuvem.

Volume Worker (nova-volume)

Esse componente trata do gerenciamento de volumes *Logical Volume Manager* (LVM). O LVM é uma forma de se organizar informações dentro de um disco. Em vez de se utilizar uma partição com determinado sistema de arquivos, cria-se uma partição do tipo LVM e dentro dela é permitido criar-se volumes, com diferentes tamanhos e sistemas de arquivos. Assim, torna-se mais versátil a utilização do espaço em disco podendo-se criar, apagar, redimensionar e duplicar os chamados volumes LVM, dentre outras ações.

Uma instância de máquina virtual no OpenStack, assim como outros softwares de nuvem, possui dois tipos de armazenamento: não-persistente e persistente. Quando um instância é iniciada, existirá um sistema de arquivos referente ao sistema operacional dessa instância. Todo dado armazenado nesse sistema de arquivos é considerado não-persistente uma vez que, ao se desligar essa instância, todos os dados se perderão. O *Volume Worker* existe justamente para possibilitar o armazenamento de dados de modo persistente. Ele permite que volumes sejam criados e vinculados a uma instância, de modo que todo dado gerado durante seu ciclo de vida possa ser armazenado nesse volume e recuperado posteriormente. O OpenStack trabalha com o LVM para criação dos volumes persistentes utilizados pelas instâncias e com o protocolo iSCSI para montagem remota.

Scheduler (nova-scheduler)

O *Scheduler* executa na forma de um *daemon* sendo responsável por mapear as chamadas geradas no *API Server* (nova-api) para os componentes apropriados do OpenStack. Assim, uma ação de usuário é primeiramente interpretada pelo *API Server* e uma ou mais chamadas Web Services são criadas. O *Scheduler*, então, será responsável por mapear essas chamadas para ações dentro da infraestrutura da nuvem, comunicando-se com os componentes necessários para que a ação do usuário seja implementada. Ele se utiliza dos recursos disponíveis na infraestrutura se comunicando com os componentes nova-compute, nova-network e nova-volume para alocá-los. Sua decisão pode ser baseada em diversos

fatores como carga de processamento, distância física da zona de disponibilidade*, memória, arquitetura da CPU, etc. Os algoritmos de escalonamento são:

- *chance*: o *host* para execução da instância é escolhido aleatoriamente entre as zonas de disponibilidade;
- *availability zone*: similar ao *chance*, mas o *host* é escolhido aleatoriamente dentro de uma zona de disponibilidade;
- *simple*: o *host* será escolhido com base na menor carga de processamento.

Storage Infrastructure (*OpenStack* Object Storage - Swift)

O Swift é uma infraestrutura de armazenamento de dados na forma de objetos. Deve ficar claro que o Swift é diferente do nova-volume, um componente do OpenStack responsável por possibilitar que discos virtuais sejam utilizados nas instâncias. Considerando o que foi explicado sobre a característica de um serviço na nuvem, o Swift pode ser entendido como um sistema capaz de fornecer armazenamento como serviço, semelhante ao S3 da Amazon. Uma vez que seu objetivo é fornecer armazenamento, diversas características são inerentes de sua utilização, como por exemplo, gerenciamento de redundância e *failover*, capacidade de servir gráficos e vídeos (*stream*), escalabilidade, dentre outras. Trata-se, portanto, de uma maneira de se armazenar dados na nuvem com diversas características de suporte e otimização relativa a tal atividade, buscando abstrair para o usuário toda a complexidade do processo de custódia da informação armazenada.

Imaging Service (Glance)

O Glance é um serviço de pesquisa e recuperação de imagens de máquinas virtuais. Além da tarefa de gerenciamento, dentro do OpenStack ele

---

* *Zonas de disponibilidade correspondem a clusters separados dentro de uma nuvem, inclusive geograficamente, a fim de se manter determinada redundância e disponibilidade.*

pode ser entendido como o repositório das imagens das máquinas virtuais. Em relação ao armazenamento, existem 4 diferentes mecanismos que podem ser utilizados pelo Glance:

- Armazenamento local (no *host* que executa o Glance).
- OpenStack *Object Store* para armazenamento das imagens.
- S3 *Storage* diretamente.
- S3 *Storage* utilizando o Object Store como intermediário para acesso ao S3.

Uma vez que o Swift provê armazenamento de forma independente do fornecimento de infraestrutura de processamento como serviço, o mais comum é que o Glance utilize o mecanismo de armazenamento local. Assim, as imagens das máquinas virtuais são simplesmente armazenadas no sistema de arquivos do *host* que o executa.

KVM

O KVM, *Kernel-based Virtual Machine* (KVM, 2011) é uma solução de virtualização para sistemas Linux que utilizam processadores com arquitetura x86 e possuem extensões de virtualização em hardware, como é o caso das tecnologias Intel VT e AMD-V. O KVM é um módulo do *kernel* Linux que provê a infraestrutura de núcleo necessária para se utilizar virtualização. O componente do KVM interno ao *kernel* do Linux está inserido neste desde a versão 2.6.20.

No entanto, o KVM por si só, não permite que todas as atividades necessárias para se criar um sistema virtual sejam executadas. Para se criar máquinas virtuais que utilizem o KVM como *hypervisor*, é necessário a utilização de outra ferramenta, denominada QEMU (QEMU, 2011). O QEMU é um emulador e virtualizador de máquinas. Como emulador ele é capaz de executar sistemas operacionais desenvolvidos para diferentes hardwares, como processadores ARM, em um PC comum, realizando tradução de chamadas de sistemas. Como virtualizador, o QEMU utiliza algum *hypervisor*, como o Xen ou KVM, para permitir a execução, de fato, de uma máquina virtual.

Conforme discutido, o KVM será o *hypervisor* utilizado neste trabalho. Todas as instâncias de máquinas virtuais executadas na nuvem o utilizarão como solução de virtualização.

Implementação de uma nuvem IaaS na Embrapa Informática Agropecuária

Com o objetivo de testar a pilha de ferramentas do OpenStack, uma arquitetura foi elaborada. A infraestrutura física foi composta por 3 computadores que formam a nuvem e um computador utilizado como cliente. Todos os computadores da nuvem utilizavam como sistema operacional o Ubuntu Server 11.04 x86_64. O cliente utilizava a versão Desktop do Ubuntu 11.04. Para o cliente é necessário que seja instalado um ambiente gráfico, possibilitando, assim, a criação das imagens.

Arquitetura da nuvem – processamento como serviço

A arquitetura elaborada é apresentada na Figura 7. Os computadores utilizados foram organizados segundo as descrições seguintes.

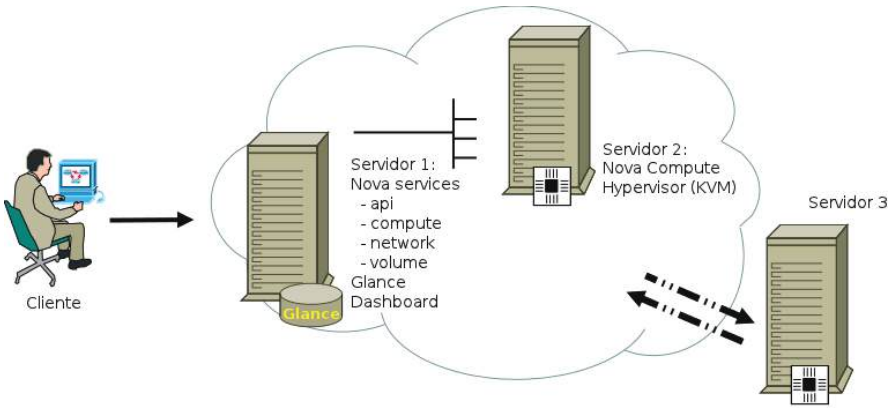


Figura 7. Arquitetura da nuvem da Embrapa Informática Agropecuária: o servidor 1 executa diversos serviços Nova sendo o nó central; o servidor 2 é um nó de computação executando máquinas virtuais; o servidor 3 é utilizado para testes de inserção e remoção de hardware na nuvem.

- **Cliente**: utilizado para gerar imagens e testar o acesso à nuvem.
- **Servidor 1**: é o nó controlador da nuvem, sendo o ponto único de acesso aos serviços por ela fornecidos. Executa todos os componentes do OpenStack, como o nova-api, nova-compute, nova-network e nova-volume. Executa ainda o Glance, o repositório de imagens das máquinas virtuais, e a Dashboard, uma interface facilitadora da interação usuário/nuvem.
- **Servidor 2**: nesta organização, é utilizado como nó de processamento. Trata-se de um *host* que executará instâncias das máquinas virtuais até sua capacidade máxima em termos de memória e processamento.
- **Servidor 3**: este servidor é utilizado para testes de realocação de hardware, ou seja, com o objetivo de simular a inserção e remoção de hardware na estrutura da nuvem.

A principal utilização da nuvem computacional proposta é para o rápido provisionamento de capacidade de processamento para o desenvolvimento de pesquisas agropecuárias. A arquitetura provê a capacidade de se instanciar e liberar servidores de maneira muito versátil. Se um grupo de pesquisa necessita que determinado servidor seja estabelecido, este é prontamente instanciado. Além disso, um servidor virtual pode ser acessado por mais de um grupo. Por exemplo, se um determinado projeto de pesquisa envolve pesquisadores de diversos grupos diferentes, cada um poderá acessar o servidor virtual que pertencerá ao projeto. É possível, pela organização da nuvem, que o acesso a cada servidor virtual seja configurado para refletir a participação de usuários de grupos distintos, cada qual com sua colaboração. A Figura 8 apresenta essa organização, representando diferentes grupos participando de diferentes projetos de pesquisa. Embora cada grupo possua uma infraestrutura de rede independente, no âmbito das pesquisas todos convergem para um ambiente único e controlado, a nuvem privada.

Arquitetura da nuvem – armazenamento como serviço

Uma maneira simplista de se considerar uma pesquisa é entender que se trata de atividade cujos resultados gerados, normalmente, são dados a serem armazenados. Outra característica dessa arquitetura de nuvem é permitir que dados resultantes de pesquisas possam ser facilmente

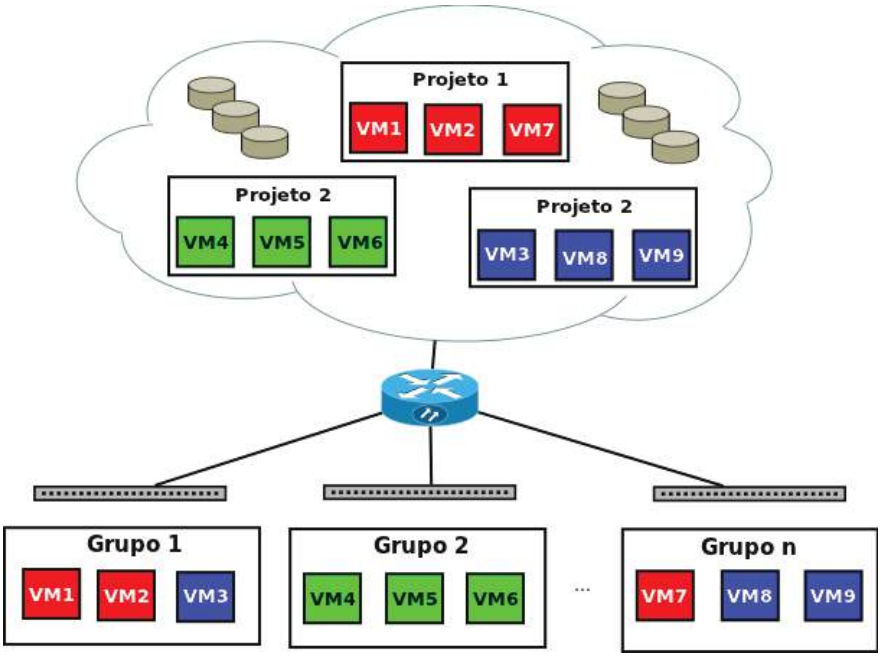


Figura 8. Nuvem privada: diferentes grupos de pesquisa participando de diferentes projetos, todos executando em um ambiente único, controlado e com recursos otimizados.

compartilhados entre os pesquisadores ou entre os grupos de pesquisa. Esses dados podem ficar armazenados na própria nuvem, com controle de acesso e várias características de segurança, tal como a redundância automática de dados. A Figura 9 apresenta a organização implementada para o fornecimento de armazenamento como serviço.

O Swift é o componente do OpenStack responsável por prover armazenamento como serviço. Seu funcionamento se dá por meio de diversos nós que disponibilizam espaço em disco para que dados sejam armazenados. Utilizando a mesma estrutura de hardware que provê processamento, uma vez que ambos os serviços podem coexistir na mesma estrutura, foi implementado um sistema de armazenamento em nuvem. Esse sistema é composto por um nó de controle, que executa o componente Proxy do Swift, mais dois nós de armazenamento denominados *containers*. Os *containers* são componentes básicos de armazenamento sendo independentes de

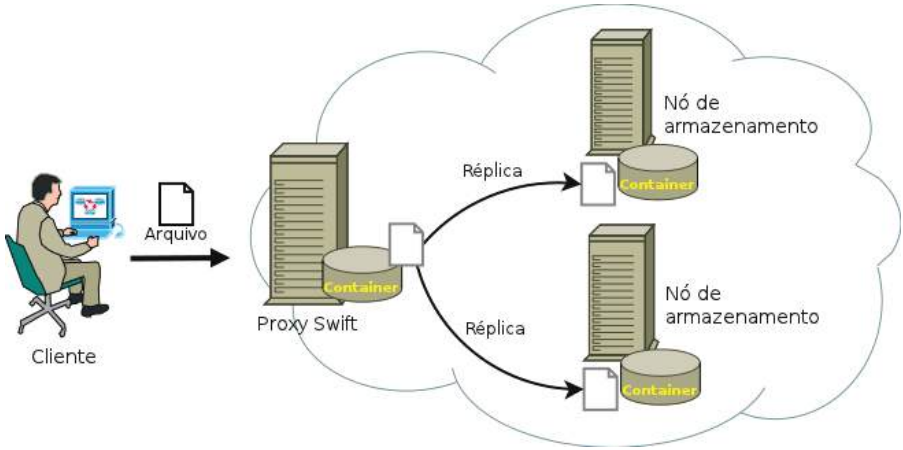


Figura 9. Armazenamento como serviço: uso do Swift dentro da estrutura de nuvem proposta. Um Proxy para interação e nós de armazenamento para replicação de dados, todos em sincronia.

qualquer outro serviço. Assim, o próprio Proxy, controlador, também é um *container*.

Toda a interação dos usuários ocorre com o Proxy por meio de utilitários de linha de comando que permitem operações como mostrar o estado do sistema de armazenamento, enviar arquivos e fazer o download de arquivos armazenados. O Swift opera conjuntamente com o Rsync, um sistema de sincronização de arquivos, de modo que as modificações realizadas pelos usuários são replicadas por toda a nuvem. O próprio modelo de armazenamento provê a característica de redundância de modo que sistemas como o RAID são desnecessários e até mesmo não indicados para uso conjunto com o Swift. Em sua operação, o Swift utiliza um padrão de acesso a disco que compreende o pior caso para sistemas RAID, de forma que a performance diminui rapidamente, principalmente para as configurações RAID 5 e RAID 6. O Swift compreende, portanto, uma nova maneira de se oferecer a característica de redundância que sistemas RAID oferecem, com diversas outras características.

O serviço de armazenamento busca ao máximo abstrair do usuário as complicações das tarefas de armazenamento consistente. Embora tenha sido utilizada apenas uma zona de disponibilidade, o Swift é capaz de

operar em diversas delas, instaladas em estruturas física e geograficamente diferentes, aumentando ainda mais as características de redundância e disponibilidade.

Características técnicas de funcionamento

A arquitetura proposta possui algumas características técnicas de funcionamento importantes para o entendimento do processo de comunicação entre os *hosts* da nuvem e entre os usuários e as instâncias virtuais. A nuvem possui uma rede própria, com endereçamento próprio. Embora o recomendado seja cada *host* da nuvem possuir duas interfaces de rede, é possível realizar uma implementação utilizando apenas uma interface, atuando em modo *bridge*. O *host* controlador possuirá dois endereços IP, sendo um privado da nuvem, para comunicação com os outros nós, e um utilizado para acessos oriundos dos clientes. Assim, este segundo endereço é o ponto de acesso por onde clientes da LAN acessarão a nuvem privada.

O nó controlador da nuvem atua como *gateway*. Todo o tráfego de todas as instâncias é direcionado a esse nó que tomará decisões de roteamento. Assim, é possível que as instâncias se comuniquem entre si e também possuam acesso à internet. O OpenStack permite ainda que determinados endereços sejam reservados para determinadas instâncias. Assim, é possível que uma instância responda por um endereço da LAN como se fosse uma máquina desse domínio. Todo o direcionamento de tráfego é realizado pelo controlador que nesse caso realiza traduções de endereços (NAT). O componente responsável por executar essas ações é o nova-network. Em versões futuras será possível atribuir diferentes redes privadas para cada projeto dentro da nuvem, de modo que sejam completamente independentes.

Outra característica importante do OpenStack é a abstração de projeto. A criação de instâncias está sempre vinculada a um projeto cadastrado no sistema. Assim, a criação de um projeto é o passo inicial para o uso da nuvem. As alterações subsequentes, como a criação de usuário e de instâncias virtuais, estarão relacionadas a esse projeto. Essa organização objetiva permitir que diversos projetos diferentes sejam executados simultaneamente na nuvem, como apresentado na Figura 8. Desta forma,

usuários diferentes podem ter acesso a instâncias diferentes desde que o administrador do projeto autorize.

Quanto ao acesso a uma instância virtual, ele é configurado durante o instanciamento da máquina. Para sistemas Linux é possível personalizar as imagens de modo que algumas ações sejam executadas durante o processo de carga da máquina virtual. Dentre essas ações está, por exemplo, a inserção de uma chave pública que permite ao usuário acessar a instância por meio do protocolo SSH. A chave privada é fornecida ao usuário por meio de comandos ou pela interface gráfica. Para outros tipos de sistemas que possam ser virtualizados, como o Windows, é necessário utilizar agentes, não tão comuns atualmente, dentro das instâncias.

No contexto do armazenamento de dados, o Swift não oferece atualmente opções de interface gráfica para o usuário. O upload e download de arquivos para o sistema se dá por meio de utilitários de linha de comando. Atualmente o Swift não permite, por padrão, o aninhamento de diretórios compondo uma hierarquia, tal como ocorre em sistemas de arquivos comuns. Os usuários devem utilizar *containers* e armazenar todos os arquivos nestas estruturas. As últimas versões de teste do Swift estão permitindo, por meio de algumas configurações, que essa hierarquia de diretórios seja simulada, de modo que o usuário possa fazer uso deste tipo de organização, embora internamente o Swift continue utilizando o sistema de *containers*.

Tecnicamente o Swift utiliza o Rsync para manter os dados sincronizados entre todos os nós. Quando uma modificação ocorre são executadas rotinas de sincronia entre os nós participantes da nuvem de armazenamento. O nó controlador mantem estruturas denominadas Rings. Os Rings são mapeamentos entre arquivos/containers e localidades na nuvem. Desta forma, o nó controlador (Proxy) consegue indexar onde está um arquivo, suas réplicas e o estado de cada réplica.

Testes

Esta seção discute os testes realizados com a arquitetura implementada.

Considerações gerais

A versão do OpenStack inicialmente utilizada foi a Cactus. No entanto, algumas novas funcionalidades e, principalmente, correções, foram inseridas em uma nova versão denominada Diablo. Por se tratar de um conjunto de ferramentas em desenvolvimento, ambas as versões não são completamente estáveis. Ainda, durante a implementação do ambiente, outra versão passou a ser foco de desenvolvimento, denominada Essex. Essa constante mudança dificulta a implementação, no sentido de que não é possível atualizar apenas parte das ferramentas para correções de determinados problemas, e limita as funcionalidades, uma vez que novas versões trazem novas características, mas normalmente são menos estáveis que as anteriores. O que se buscou foi manter um ambiente funcional com o mínimo de estabilidade, mesmo que nem todas as funcionalidades estivessem presentes.

Instanciamento e liberação de máquinas virtuais (linha de comando e interface)

O instanciamento e liberação de máquinas virtuais compreendem as funcionalidades básicas de uma nuvem IaaS, cujo serviço oferecido é infraestrutura de processamento. Essas tarefas puderam ser executadas tanto por meio de ferramentas de linha de comando quanto por meio da interface Web Dashboard. A ideia de que a criação de servidores pode ser executada de maneira muito rápida foi comprovada. Possuindo a imagem no repositório Glance, basta um comando para que diversas máquinas virtuais sejam criadas em alguns minutos.

```
euca-run-instances -n 10 -k chave -t m1.small ami-
-0000000f
```

O comando anterior instancia 10 máquinas virtuais. O parâmetro -k indica qual chave deve ser inserida nas máquinas para possibilitar o acesso via SSH. O parâmetro -t indica o tipo da instância. Nesse contexto, o tipo refere-se à configuração da máquina virtual. O OpenStack traz algumas configurações pré-cadastradas no sistema, sendo chamados *flavors*.

Essas configurações podem ser alteradas pelos administradores para se criar novos *flavors* a fim de adequá-los com os tipos de máquinas virtuais que se deseja. A lista a seguir mostra as configurações pré-definidas.

```
m1.medium: Memory: 4096MB, VCPUS: 2, Storage: 40GB
m1.large: Memory: 8192MB, VCPUS: 4, Storage: 80GB
m1.tiny: Memory: 512MB, VCPUS: 1, Storage: 0GB
m1.xlarge: Memory: 16384MB, VCPUS: 8, Storage: 160GB
m1.small: Memory: 2048MB, VCPUS: 1, Storage: 20GB
```

Assim, as máquinas criadas pelo comando anterior terão 2 GB de memória RAM cada, um processador e 20 GB de espaço virtual em disco. Esse espaço virtual é volátil e indica a quantidade de dados que cada instância virtual pode utilizar durante sua execução.

Além da maneira convencional, é possível instanciar máquinas virtuais por meio da interface Web Dashboard, de maneira muito mais intuitiva. Basta clicar na imagem do repositório, escolher qual o tipo de configuração, a quantidade de instâncias e a chave de acesso, tudo de forma visual. A Figura 10 apresenta uma visualização da parte de imagens da Dashboard, onde o usuário pode criar novas instâncias. Do lado direito da figura é mostrada a parte de seleção das configurações da instância que será criada.

Pode-se observar também que a Dashboard gerencia as instâncias criadas, chaves de acesso e os volumes de armazenamento permanente. As novas versões da interface possibilitarão ao administrador monitorar o estado do hardware da nuvem, tal como a capacidade total de memória usada e disponível, bem como quais *hosts* estão com maior carga de processamento.

Montagem de volume persistente em máquina virtual

Conforme já mencionado, as instâncias possuem discos virtuais para armazenamento, o que significa que os dados são voláteis. Ao ser liberada, uma instância descarta qualquer dado que tenha sido armazenado em seu sistema de arquivos raiz. Para fornecer armazenamento persistente para as instâncias, o OpenStack utiliza o componente nova-volume e permite que volumes de dados sejam montados nas instâncias. Os dados armazenados nesses volumes são permanentes e podem ser utilizados posteriormente à liberação de uma instância. Os volumes são criados por meio do

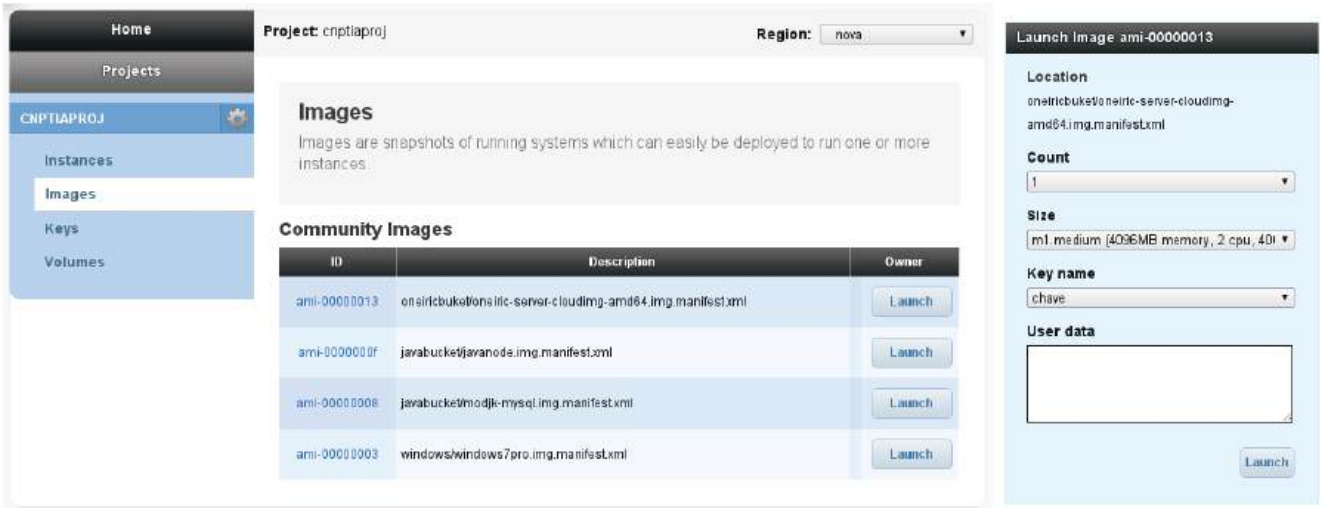


Figura 10. *Dashboard*: interface para interação com a nuvem capaz de executar diversas tarefas, como a criação de instâncias virtuais.

LVM e são montados nas instâncias por meio do protocolo iSCSI, via rede.

A montagem de volumes nas instâncias também pode ser realizada tanto por meio de ferramentas de linha de comando quanto por meio da Dashboard. O comando a seguir indica para a nuvem que o volume vol-0000001 deve ser montado na instância i-00000019, utilizando o dispositivo /dev/vdc.

```
euca-attach-volume -i  i-00000019 -d /dev/vdc  vol-
0000001
```

Embora o dispositivo seja indicado no comando `(/dev/vdc)`, nem sempre é possível utilizar o mesmo dispositivo na instância. Assim, pode ocorrer do *kernel* da instância atribuir automaticamente outro dispositivo, diferente do`/dev/vdc`. Faz-se necessária uma verificação nas mensagens do *kernel* para se determinar a qual dispositivo o volume foi atribuído. Isso pode ser feito por meio do comando `dmesg`. Neste ponto, o volume persistente deve se tornar disponível na instância em questão. Para que seu uso seja possível, é necessário montá-lo no sistema de arquivos local da instância, da mesma maneira que ocorre com qualquer dispositivo de armazenamento. Embora grande parte do processo seja manual, o objetivo das novas versões de software para nuvem, principalmente os agentes a serem instalados nas imagens das máquinas virtuais, é fazer com que esse processo seja automatizado. Pretende-se, por exemplo, que entradas sejam inseridas no arquivo de montagem de partições, o `/etc/fstab`, de modo que, durante o *boot* da instância, volumes persistentes sejam montados automaticamente. Estas modificações tornariam ainda mais versátil o gerenciamento de instâncias e volumes. Um exemplo prático de utilização desse modelo é para uma instância que utiliza um banco de dados. Uma imagem virtual, com o banco de dados já instalado, é instanciada e todos os dados referentes ao banco são armazenados em um volume persistente, montado automaticamente no *boot* do sistema.

Em suma, as instâncias virtuais em uma nuvem possuem um espaço de disco volátil, indicado no *flavor* durante sua criação, e opcionalmente, um espaço para armazenamento permanente por meio da montagem de volumes persistentes.

Geração e execução de imagens personalizadas

As imagens utilizadas na nuvem devem ser geradas previamente. Existem, atualmente, imagens prontas disponibilizadas na internet, como é o caso das imagens do Ubuntu voltadas para uso em nuvens. Embora essas imagens possuam todos os softwares necessários para customização de uso em nuvens, são apenas a instalação básica do sistema operacional.

Uma das características importantes do OpenStack é permitir que o usuário personalize suas imagens, instalando diferentes sistemas operacionais e diferentes softwares em cada uma delas. Durante os testes foi possível gerar duas imagens diferentes, uma contendo o sistema operacional Ubuntu Server 11.04 e outra contendo o sistema Windows 7. O acesso a uma instância da imagem do Ubuntu ocorre por meio do SSH, porém utilizando usuário e senha, sem o uso de chaves. A imagem gerada executou, normalmente, entrando na rede privada na nuvem. Para o sistema Windows deve-se utilizar o protocolo RDP de acesso remoto. A imagem deve possuir um servidor de RDP para que seja possível ao usuário conectar e visualizar o ambiente. A execução da imagem ocorreu com sucesso. No entanto, o Windows não entra automaticamente na rede da nuvem. Existem atualmente agentes para serem instalados em sistemas Windows a fim de facilitar a configuração da instância para a execução em nuvem. No entanto, esses agentes funcionam quando o *hypervisor* utilizado é o Xen. Uma vez que o KVM foi utilizado, não foi possível completar automaticamente o processo de instância e conexão a um sistema Windows com as versões de software utilizadas. Se uma conexão for realizada manualmente, utilizando portas que o próprio KVM disponibiliza para a conexão com a máquina virtual, é possível conectar na instância e verificar que sua execução está ocorrendo normalmente.

Por fim, duas outras imagens foram geradas a partir do Ubuntu Server 11.04. Uma dessas imagens possui o conjunto de servidores Web Apache e Tomcat. O Apache possui ainda o módulo `mod_jk`, utilizado para permitir acesso a aplicações Web em Java pelo Apache. A segunda imagem possui apenas o servidor Web Tomcat. Essas imagens foram geradas para se testar outras características da nuvem, conforme seção a seguir.

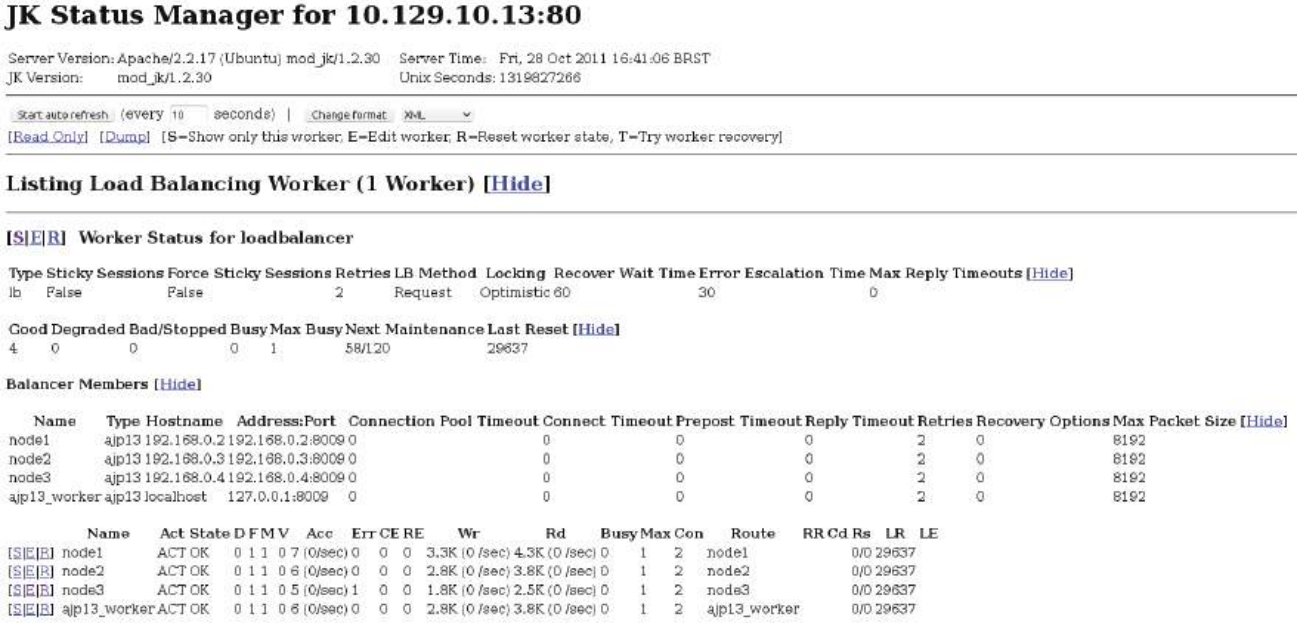
Teste de elasticidade utilizando aplicação Web em Java

Uma das grandes características de uma nuvem é fornecer elasticidade em relação à capacidade de determinados serviços. A quantidade de acessos a uma aplicação Web pode sofrer uma grande variação em diferentes períodos. Enquanto em determinados momentos toda a capacidade computacional do servidor é utilizada, em outros, esse permanece até mesmo ocioso. Administradores que lidam com tais situações devem provisionar o hardware para suprir os momentos de pico. Isso faz com que na maior parte do tempo a capacidade do servidor fique ociosa.

A elasticidade é uma característica que permite aumentar e diminuir a capacidade de hardware de determinado sistema de forma imediata. Na situação de pico do exemplo anterior, a capacidade do servidor web poderia ser aumentada e, após esse período, diminuída. Essa possibilidade dependerá do tipo de serviço sendo fornecido. Aplicações Web são um exemplo na qual a elasticidade surge como grande auxílio em sua gerência.

Conforme mencionado, para esse teste foram geradas duas imagens diferentes, uma contendo um sistema com o servidor Tomcat e outra contendo um sistema com o Apache e o módulo `mod_jk`. O `mod_jk` permite que o servidor Apache receba requisições para um site e execute um balanceamento de cargas, dividindo essas requisições entre vários servidores Tomcat em máquinas diferentes. Nesse cenário de teste, foi utilizado um nó executando o Apache com o módulo `mod_jk` mais três instâncias executando o servidor Tomcat. Como o nó com Apache também possuía um servidor Tomcat, a estrutura conta com 4 nós Tomcat para processamento das requisições.

Todos os servidores Tomcat foram configurados para operar junto ao nó balanceador. Uma aplicação Web simples foi criada com o intuito de verificar a ocorrência do balanceamento entre as instâncias, sendo instalada nas duas imagens. Essa aplicação simplesmente recebe uma conexão e mostra uma página contendo o endereço IP para qual foi destinada a requisição, equivalente ao IP do servidor onde o site está hospedado, e o IP da interface de rede física do *host* onde a requisição está sendo processada. A Figura 11 apresenta a interface de monitoramento do balanceador de cargas. Nela é possível observar a existência de 4 nós e o estado de cada um deles. O IP mostrado na parte superior, 10.129.10.13 é o IP pelo qual o



JK Status Manager for 10.129.10.13:80

Server Version: Apache/2.2.17 (Ubuntu) mod_jk/1.2.30 Server Time: Fri, 28 Oct 2011 16:41:06 BRST
JK Version: mod_jk/1.2.30 Unix Seconds: 1319827266

Start auto refresh (every 10 seconds) | Change format XML
[Read Only] [Dump] [S=Show only this worker, E=Edit worker, R=Reset worker state, T=Try worker recovery]

Listing Load Balancing Worker (1 Worker) [Hide]

[S|E|R] Worker Status for loadbalancer

Type	Sticky Sessions	Force Sticky Sessions	Retries	LB Method	Locking	Recover Wait Time	Error Escalation Time	Max Reply Timeouts [Hide]
lb	False	False	2	Request	Optimistic	60	30	0

Good	Degraded	Bad/Stopped	Busy	Max Busy	Next Maintenance	Last Reset [Hide]
4	0	0	0	1	58/120	29637

Balancer Members [Hide]

Name	Type	Hostname	Address:Port	Connection Pool Timeout	Connect Timeout	Prepost Timeout	Reply Timeout	Retries	Recovery Options	Max Packet Size [Hide]
node1	ajp13	192.168.0.2	192.168.0.2:8009	0	0	0	0	2	0	8192
node2	ajp13	192.168.0.3	192.168.0.3:8009	0	0	0	0	2	0	8192
node3	ajp13	192.168.0.4	192.168.0.4:8009	0	0	0	0	2	0	8192
ajp13_worker	ajp13	localhost	127.0.0.1:8009	0	0	0	0	2	0	8192

	Name	Act	State	D	F	M	V	Acc	Err	CE	RE	Wr	Rd	Busy	Max	Con	Route	RR	Cd	Rs	LR	LE
[S|E|R]	node1	ACT	OK	0	1	1	0	7 (0/sec)	0	0	0	3.3K (0 /sec)	4.3K (0 /sec)	0	1	2	node1			0/0	29637	
[S|E|R]	node2	ACT	OK	0	1	1	0	6 (0/sec)	0	0	0	2.8K (0 /sec)	3.8K (0 /sec)	0	1	2	node2			0/0	29637	
[S|E|R]	node3	ACT	OK	0	1	1	0	5 (0/sec)	1	0	0	1.8K (0 /sec)	2.5K (0 /sec)	0	1	2	node3			0/0	29637	
[S|E|R]	ajp13_worker	ACT	OK	0	1	1	0	6 (0/sec)	0	0	0	2.8K (0 /sec)	3.8K (0 /sec)	0	1	2	ajp13_worker			0/0	29637	

Figura 11. Página de monitoramento do balanceador de cargas `mod_jk` usado em aplicações Web: exemplo de elasticidade na nuvem.

site é acessado. Os IPs da rede 192.168.0.0 representam os IPs privados de instâncias da nuvem.

Essa organização provê grande elasticidade. Na ocasião dos 4 nós terem seus recursos exauridos devido a uma grande quantidade de requisições, basta que uma ou mais instâncias que possuem o Tomcat sejam iniciadas e pequenas configurações realizadas para que passem a receber requisições do balanceador. Isso implicará em rápido aumento na capacidade da aplicação.

A Figura 12 mostra a página da aplicação, identificando tanto o IP único do site (10.129.10.13) quanto o IP da instância. Conforme os acessos são realizados, é possível perceber a mudança do IP da instância, comprovando que as requisições são divididas entre os servidores Tomcat existentes.

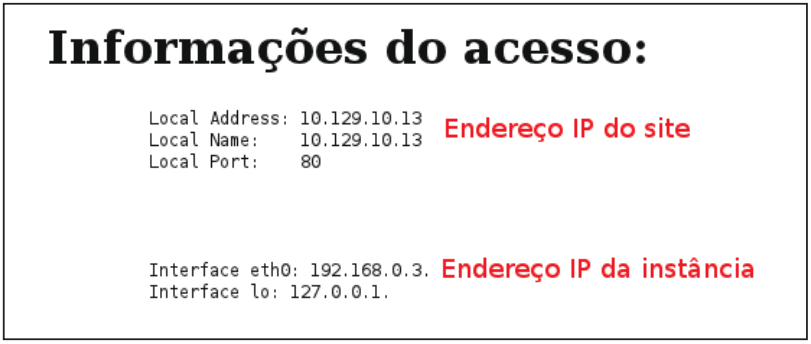


Figura 12. Página da aplicação Web balanceada com instâncias na nuvem: IPs indicam distribuição de cargas entre as máquinas virtuais.

Se em determinado momento as 4 instâncias não forem mais necessárias, pode-se diminuir esse número e liberar o hardware utilizado para outro tipo de atividade.

Testes de realocação de hardware

Os testes de realocação de hardware objetivavam elucidar a maneira pela qual novos hardwares seriam inseridos na nuvem e como hardwares com problemas ou antigos seriam retirados dela. O servidor 3 da arquitetura proposta foi utilizado para esse fim.

O OpenStack funciona como um conjunto de ferramentas colaborativas. Essa divisão em vários componentes traz certa independência entre cada um deles. Assim, em uma infraestrutura para provimento de servidores virtuais para processamento, a inserção de um hardware é realizada pela instalação do componente nova-compute. Este novo *host* deverá ser semelhante ao restante da nuvem em relação ao tipo de sistema operacional e possuir suporte à virtualização. Portanto, basta que alguns pacotes básicos de virtualização e o nova-compute sejam instalados e a configuração do nó controlador replicada para este *host* para que ele passe a fazer parte da nuvem. O nova-compute se encarregará de informar o nó controlador sobre suas capacidades e, a partir do momento que o controlador obtiver essas informações, máquinas virtuais podem ser instanciadas nesse novo nó.

Para a remoção de um hardware, é aconselhável que as instâncias em execução sejam finalizadas. Desta forma o nó controlador receberá informações para manter sua base de estados atualizada, sobre a existência das instâncias. A partir do momento que não há conectividade entre o controlador e um nó da nuvem, o controlador passa a informar ao administrador que não mais dispõe daquele nó. Portanto, a remoção de um nó simplesmente se dá pela desconexão do nó da nuvem. Caso alguma instância esteja em execução, o controlador pode manter informações desatualizadas.

Processo semelhante é utilizado para adicionar nós de armazenamento quando este é o serviço fornecido pela nuvem. O novo hardware é conectado à estrutura de rede da nuvem e componentes são instalados para que o nó se comunique com o nó controlador do serviço. No caso do armazenamento, especificamente, é necessário que o nó controlador seja reconfigurado no sentido de reconstruir as estruturas de indexação (*rings*), passando a utilizar também o novo nó.

Armazenamento de dados na infraestrutura Swift

O teste de armazenamento planejado consistia em gravar dados na infraestrutura provida pela nuvem, simular a perda de nós de armazenamento, analisar o estado do sistema e tentar recuperar uma informação inicialmente armazenada. Todos os passos puderam ser realizados com

sucesso. Conforme mencionado, a interação com o Swift ocorre por meio de ferramentas de linha de comando. Os comandos a seguir são exemplos de como se adicionar arquivos a um *container*, como verificar o estado do sistema e como recuperar um arquivo armazenado.

```
 swift -A https://<IP do proxy>:8080/auth/v1.0 -U
system:root -K <senha> upload <container> <qualquer ar-
quivo>

 swift -A https://<IP do proxy>:8080/auth/v1.0 -U
system:root -K <senha> stat
    Account: AUTH_a4e769fe-3b47-4509-96da-e082d914b260
    Containers: 1
    Objects: 1
    Bytes: 13905920
    Accept-Ranges: bytes

 swift -A https://<IP do proxy>:8080/auth/v1.0 -U
system:root -K <senha> download <container>
```

Avaliação e considerações finais

A utilização de nuvens computacionais é atualmente uma nova vertente na maneira de se utilizar a computação para o provimento de serviços. Nesse contexto, os serviços são bastante diferentes dos conceitos até então conhecidos, visto a possibilidade de se considerar como um deles o provisionamento de infraestrutura computacional.

Nessa etapa de maturação, é comum o surgimento de diversas dúvidas sobre o que é possível se fazer com o uso dessa tecnologia. São diversas as aplicações que uma nuvem computacional pode ter, de forma que seu uso é recomendado quando suas características facilitam e otimizam o trabalho de administradores e usuários em geral. Não existe, e possivelmente não existirá, uma categoria de problemas para o qual as nuvens devam ser empregadas. O que se deve ter em mente são as características que elas podem proporcionar frente ao problema que se busca resolver. A partir disso, tornam-se mais claros diversos cenários onde as nuvens podem

ser empregadas, desde modelos comerciais até modelos para atividades didáticas, contemplando diversos nichos do uso da computação.

A opção ou não por usar uma nuvem é de cada administrador de serviços de TI. Conhecendo as características de uma nuvem computacional torna-se possível elencar algumas diretivas para um melhor esclarecimento dos usos.

O que esperar de nuvens privadas

- Versatilidade no provisionamento e gerenciamento de serviços;
- Escalabilidade, de acordo com a quantidade de ativos de hardware;
- Elasticidade para aplicações preparadas para executarem em nuvem;
- Otimização de uso de recursos computacionais;
- Sistema redundante de armazenamento de arquivos (backup);
- Facilitação de gerenciamento e fornecimento de determinados serviços;
- Economia de energia.

O que não esperar de nuvens privadas

- Que desempenhem o papel da virtualização, pois são tecnologias com objetivos distintos;
- Que substitua ou faça exatamente o papel das estruturas de *Clusters* e *Grids* computacionais;
- Que sejam utilizadas como sistema de arquivos distribuído, pois estes tem diversas funcionalidades diferentes, principalmente quanto ao acesso aos dados.

A computação em nuvem não objetiva substituir nenhuma tecnologia existente atualmente. É comum existirem diversas dúvidas por parte dos profissionais de TI e usuários em geral nos primeiros contatos com o tema. Muitas delas surgem da relação da computação em nuvem com a virtualização e com armazenamento de arquivos *online*.

Existem diversas empresas que apresentam produtos por elas classificados como computação em nuvem. Outras, acreditam possuir uma estrutura

de nuvem privada quando na verdade possuem uma estrutura de virtualização de servidores, *desktops* e aplicações. A computação em nuvem é uma tecnologia que inclui as características de serviços monitorados e autogerenciáveis, rede de alta capacidade, uso de *pool* de recursos e elasticidade. A virtualização somada a uma camada de controle não pode ser considerada computação em nuvem, pois não apresenta diversas outras características presentes em sistemas reais de infraestrutura de nuvem, como o OpenStack.

Ainda em relação à virtualização, deve-se ter cuidado no entendimento do sentido inverso: computação em nuvem não é virtualização. Nem todas as aplicações que executam em ambientes virtualizados poderão ser migradas para uma nuvem resultando em vantagens. Com o avanço dos *frameworks* de nuvem, as aplicações desenvolvidas usarão APIs específicas e serão preparadas para a execução nesse tipo de ambiente. O fundamento básico é que a própria aplicação terá conhecimento que está executando em uma infraestrutura de nuvem. Isso permitirá o autogerenciamento de recursos e, consequentemente, elasticidade. Migrar uma aplicação existente, não preparada para execução em nuvem, pode representar demasiado esforço para se obter um resultado não tão vantajoso.

Em relação ao armazenamento de arquivos *online*, uma dúvida frequente é se os *frameworks* de nuvem, tal como o OpenStack, são repositórios de dados para que os usuários possam guardar arquivos e recuperá-los rapidamente. Essa ideia se consolidou pelo surgimento de diversos serviços de armazenamento *online*, como o Dropbox (DROPBOX, 2012), sendo chamados de computação em nuvem. No entanto, trata-se de um equívoco sobre o conceito de nuvens computacionais, principalmente em relação às nuvens privadas. *frameworks* de computação em nuvem são muito mais complexos do que o fornecimento de um serviço de armazenamento *online*, como é o caso do Dropbox. É bem possível que esse tipo de serviço possa executar em uma estrutura de nuvem, mas não se pode considerar o serviço de armazenamento de arquivos *online* como sinônimo de computação em nuvem. Outro fator que contribuiu para essa ideia foi denominar determinados tipos de serviços de TI como sendo computação em nuvem apenas por estarem fora dos domínios de uma instituição. Ao se analisar conjuntos de ferramentas como o Nimbus, Eucalyptus e OpenStack, percebe-se que a computação em nuvem é um novo modelo

de se utilizar a computação, sendo bem mais do que apenas um serviço de armazenamento.

Trabalhos futuros

Os trabalhos futuros envolvem a automatização cada vez maior do provisionamento de serviços na nuvem. A orquestração de serviços é uma das tecnologias que buscam acompanhar os desenvolvimentos das nuvens computacionais. Com ela será possível, por exemplo, manter imagens com instalações básicas de um sistema operacional e, durante o processo de instanciamento, realizar a atualização do sistema operacional e instalação dos softwares necessários. Desse modo, não será necessário gerar uma imagem com o servidor Tomcat ou com o Apache. Bastará que no instanciamento o usuário solicite que esses servidores sejam instalados na instância. Além disso, essa atualização do sistema operacional durante o instanciamento demonstra uma preocupação com a segurança das instâncias, visto que problemas dessa categoria são corrigidos diariamente e a necessidade de atualização é constante.

Ainda, outros cenários de uso poderão ser testados. Um desses cenários diz respeito à utilização de nuvens computacionais no apoio às atividades didáticas, como a realização de cursos práticos em laboratórios.

A realização de cursos de capacitação depende sempre de um ambiente computacional que deve ser previamente preparado. Essa preparação envolve a instalação dos sistemas operacionais e softwares em cada computador a ser utilizado durante o curso. Trata-se de uma tarefa repetitiva e que demanda certo tempo para ser finalizada. O uso de uma nuvem em laboratórios de ensino é certamente aplicável. A diferença básica é que a nuvem não necessariamente precisa estar hospedada em hardware de um datacenter. Dada a existência de um laboratório didático que conta com diversos computadores, uma nuvem pode ser estabelecida no próprio laboratório, utilizando o hardware local. Em vez de se preparar computador por computador, basta que uma única imagem, contendo tudo que é necessário para o curso, seja gerada e disponibilizada no nó controlador. Com isso, atinge-se um alto grau de otimização na preparação do ambiente e uma

grande versatilidade, visto que em questão de minutos diversas máquinas virtuais podem ser instanciadas nos computadores do próprio laboratório. O acesso a essas máquinas virtuais se dá localmente, por meio de protocolos de acesso remoto como o SSH ou *Remote Desktop* (VNC). A Figura 13 mostra esta organização. As etapas para o estabelecimento de um laboratório para um curso prático são representadas pelos três passos numerados. De posse da imagem previamente preparada, o instrutor solicita o instanciamento do número de máquinas virtuais que representa a quantidade de alunos (1). O controlador começará a instanciar essas máquinas utilizando o hardware disponível no laboratório como nós da nuvem (2). Terminado o instanciamento, cada participante poderá se conectar a uma instância virtual (3). Ainda, caso haja a necessidade do compartilhamento de dados gerados durante o curso, estes poderão ser armazenados na própria nuvem.

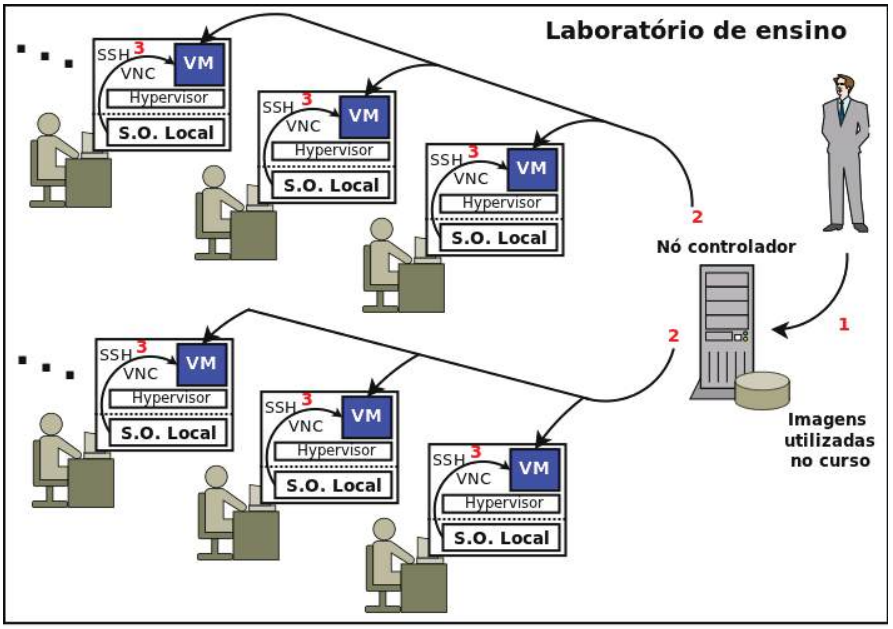


Figura 13. Laboratório para curso prático estruturado com base em uma nuvem privada: versatilidade e homogeneidade do ambiente.

Por fim, é importante que o acompanhamento das novas versões dos softwares de nuvem, tanto para novas funcionalidades quanto para a correção de erros, seja realizado constantemente. Trata-se de um novo arcabouço de ferramentas cujo interesse da comunidade tem se tornado cada vez maior, de forma que o avanço a ser atingido nos próximos anos será muito promissor para essa nova tecnologia dentro da computação.

Referências

AMAZON. **Amazon elastic compute cloud** (Amazon EC2). Pricing. Disponível em: <http://aws.amazon.com/ec2/#pricing>. Acesso em: 23 ago. 2011a.

_____. **Simple storage service** (Amazon S3). Pricing. Disponível em: <http://aws.amazon.com/s3/#pricing>. Acesso em: 23 ago. 2011b.

AMAZON Enables Bursting to the Public Cloud. **Virtual Lab Automation Blog**. 2009. Disponível em: <http://vmlab.wordpress.com/category/virtualization-platforms/cloud-computing--virtualization-platforms/>. Acesso em: 6 set. 2011.

CANONICAL. **Canonical**. Disponível em: <http://www.canonical.com/>. Acesso em: 6 set. 2011.

CINTRA, L. C. **Virtualização com o Xen**: instalando e configurando o ambiente. Campinas: Embrapa Informática Agropecuária, 2010. (Embrapa Informática Agropecuária. Comunicado técnico, 102).

DROPBOX. 2012. Disponível em: <https://www.dropbox.com/>. Acesso em: 6 jan. 2012.

ENSEMBLE. **Ensemble project**. 2011. Disponível em: <https://ensemble.ubuntu.com/>. Acesso em: 30 ago. 2011.

EUCALYPTUS. **Eucalyptus project**. 2011. Disponível em: <http://www.eucalyptus.com/>. Acesso em: 29 ago. 2011.

GOOGLE. **Google app engine**. 2011. Disponível em: <http://code.google.com/intl/pt-BR/appengine/docs/whatisgoogleappengine.html>. Acesso em: 23 ago. 2011.

IBM. **Serviços em nuvem para sua infraestrutura virtual, Parte 1**: Infrastructure-as-a-Service (IaaS) e Eucalyptus. 2010. Disponível em: <http://www.ibm.com/developerworks/br/library/os-cloud-virtual1/>. Acesso em: 23 ago. 2011.

KVM. **Kernel-based virtual machine**. 2011. Disponível em: <http://www.linux-kvm.org/>. Acesso em: 8 set. 2011.

MICROSOFT. **Windows azure**. 2011. Disponível em: <http://www.microsoft.com/windowsazure/pt/br/>. Acesso em: 23 ago. 2011.

NIMBUS. **Nimbus project**. 2011. Disponível em: <http://www.nimbusproject.org/>. Acesso em: 29 ago. 2011.

OPENNEBULA. **OpenNebula Project**. 2011. Disponível em: <http://opennebula.org/>. Acesso em: 29 ago. 2011.

OPENSTACK. **Openstack cloud software**. 2011. Disponível em: <http://www.openstack.org>. Acesso em: 29 ago. 2011.

QEMU. **Open source machine emulator and virtualizer**. 2011. Disponível em: <http://wiki.qemu.org/Main_Page>. Acesso em: 16 set. 2011.

SEMPOLINSKI, P.; THAIN, D. **A Comparison and critique of eucalyptus, OpenNebula and Nimbus**. In: IEEE INTERNATIONAL CONFERENCE ON CLOUD COMPUTING TECHNOLOGY AND SCIENCE, 2., 2010, Indianapolis. **Proceedings**...Washington: IEEE Computer Society, 2010. p. 417-426. CLOUDCOM '10.

TORRES, T. Z.; BERNARDES, R. M. **Reflexões sobre a implantação de um Programa de TI-Verde para a Embrapa: bases conceituais e metodológicas.** Campinas: Embrapa Informática Agropecuária, 2008. 25 p. (Embrapa Informática Agropecuária Documentos, 86).

Ministério da
Agricultura, Pecuária
e Abastecimento

GOVERNO FEDERAL
BRASIL
PAÍS RICO É PAÍS SEM POBREZA

CGPE 9787