

O Modelo de Computação em Nuvem e sua Aplicabilidade

E. Bacelar, P. Popiolek, V. Hax, J. Tyska, N. Duarte, O. Mendizabal

Resumo—Atualmente departamentos de TI precisam adaptar-se às mudanças de negócio sem ocasionar custo excessivo ou complexo gerenciamento para as organizações. Porém, melhorias em infraestruturas convencionais e esforços para adaptar aplicações existentes podem não ser compatíveis a esta demanda. Buscando soluções inteligentes para atender estes requisitos, este artigo trás uma visão geral sobre Computação em Nuvem, um novo paradigma para gerenciamento eficiente de recursos computacionais e provisão de serviços sob demanda. Maior flexibilidade, adaptação e economia são algumas de suas vantagens. Esta pesquisa inclui a terminologia atual e os tipos de serviço e nuvens disponíveis. Além disso, é apresentada uma implementação de nuvem computacional. Onde são discutidos detalhes sobre a infraestrutura, softwares utilizados, forma de gerenciamento e serviços a serem disponibilizados.

Palavra-Chave—Computação em Nuvem, Máquinas Virtuais, Eucalyptus, Virtualização.

I. INTRODUÇÃO

Com o crescimento e globalização das organizações, os sistemas de informação também necessitam evoluir rapidamente para acompanhar e atender novos requisitos de negócio. Cada vez mais há uma exigência para que aplicações corporativas adaptem-se às demandas do mercado em tempo e com custos satisfatórios, sem perda de qualidade.

Neste contexto, a maioria dos sistemas computacionais modernos baseia-se em infraestruturas amplamente distribuídas, possibilitando o desenvolvimento de aplicações

colaborativas, maior compartilhamento de recursos e aumento no desempenho e disponibilidade. Nas últimas décadas, várias aplicações e plataformas distribuídas foram propostas, desde modelos de aplicação cliente/servidor, redes ponto a ponto (P2P), arquiteturas multicamadas e orientadas a serviços, aglomerados e grades computacionais [1]-[2]-[3]-[4].

Embora os modelos de aglomerados e grades computacionais oferecessem compartilhamento de recursos em grande escala, considerando inclusive o acesso a recursos globalmente distribuídos, o gerenciamento e política de acesso aos nodos computacionais restringiram o uso destes modelos apenas à academia. Outra característica destes modelos de computação é o alto custo para a manutenção de uma infraestrutura de grande poder computacional.

Com o surgimento da Computação em Nuvem (*Cloud Computing*), os modelos de programação distribuída em larga escala tornam-se mais atrativos ao usuário final. O enfoque em serviços talvez seja a principal razão para a boa aceitação e utilização das nuvens computacionais. Inspirado em serviços comuns do cotidiano como, por exemplo, distribuição de energia elétrica ou serviços de telefonia, este modelo visa provisão de serviços computacionais sob demanda. Esta abordagem tira a responsabilidade das empresas em administrar serviços básicos de infraestrutura e serviços, permitindo que estas despendam mais recursos no negócio propriamente dito.

No modelo de Computação em Nuvem é possível oferecer, por exemplo, serviços de armazenamento de dados sem que o cliente disponha de uma infraestrutura própria dedicada. Portanto, tanto a infraestrutura quanto a interface da aplicação são disponibilizadas pela nuvem. Serviços podem ser adaptados às necessidades do cliente, sem que este participe da instalação, configuração ou manutenção do produto. Além disso, recursos, como máquinas virtuais, podem ser acessados pelo cliente em qualquer lugar, a qualquer momento, também sem a necessidade de envolvimento com o gerenciamento desta infraestrutura.

Além desta nova visão de como aplicações e infraestrutura podem ser oferecidas em ambientes distribuídos, outra razão para o sucesso das Nuvens Computacionais é decorrente da evolução tecnológica. O uso de virtualização associado a boas técnicas de provisão de recursos permite um melhor aproveitamento dos recursos oferecidos, maior economia e flexibilidade na configuração de ambientes. O aumento da largura de banda e maiores taxas de transferência de dados em redes de comunicação, assim como o aumento na capacidade

E. Bacelar é aluno de graduação em Sistemas de Informação na Universidade Federal do Rio Grande – FURG (e-mail: eduardobacelar@furg.br).

P. Popiolek é aluno de graduação em Engenharia de Computação na Universidade Federal do Rio Grande – FURG (e-mail: p.f.popiolek@furg.br).

V. Hax é mestrando em Engenharia de Computação na Universidade Federal do Rio Grande – FURG. Atualmente atua como Analista de Tecnologia de Informação na mesma instituição (e-mail: iniciushax@furg.br).

J. Tyska é mestre em Modelagem Computacional pela Universidade Federal do Rio Grande – FURG. Atualmente atua como Analista de Tecnologia de Informação na mesma instituição (e-mail: jontatyska@gmail.com).

N. Duarte é doutor em Informática pela Pontifícia Universidade Católica do Rio de Janeiro. Atua como professor no Centro de Ciências Computacionais da Universidade Federal do Rio Grande – FURG (e-mail: dmtndf@furg.br).

O. Mendizabal é mestre em Ciência de Computação pela Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS). Atua como professor no Centro de Ciências Computacionais da Universidade Federal do Rio Grande – FURG (e-mail: odoricomendizabal@furg.br).

de processamento dos computadores atuais, oferece melhor desempenho, possibilitando ao provedor de serviços garantir níveis de qualidade de serviço compatíveis aos requisitos dos usuários.

Este trabalho relata um estudo sobre Computação em Nuvem, envolvendo conceitos, tecnologias existentes e os modelos de serviço oferecidos. Além de descrever a terminologia associada aos modelos de serviço, também é feito um detalhamento sobre máquinas virtuais e sua utilização em Nuvens Computacionais. Além disso, será apresentada a Nuvem Computacional do C3 (Centro de Ciências Computacionais) da Universidade Federal do Rio Grande - FURG, que foi implantada como parte deste trabalho. Finalizando este estudo, também são discutidas algumas aplicações usuais no contexto das Nuvens Computacionais, que serão integradas futuramente à Nuvem do C3.

A próxima seção faz uma introdução sobre Nuvens Computacionais e seus conceitos. A Seção 3 apresenta máquinas virtuais e sua aplicação em computação em nuvem. A implementação de uma nuvem computacional é descrita na Seção 4 e conclusões e trabalhos futuros são discutidos na Seção 5.

II. COMPUTAÇÃO EM NUVEM

Embora a Computação em Nuvem esteja empregada na indústria e na academia, com modelos de serviços e tecnologias já consolidados, este conceito ainda é recente, não havendo consenso em sua definição. Segundo o *National Institute of Standards and Technology* (NIST) [5], Computação em Nuvem é “Um modelo que permite acessar, de maneira conveniente e sob demanda, recursos configuráveis de computação (por exemplo, rede, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente adquiridos e liberados com o mínimo de esforço de gerenciamento ou interação com o provedor dos serviços”. Um levantamento sobre definições e conceitos para Nuvem Computacional é apresentado em [6]. Os autores descrevem mais de 20 definições para Computação em Nuvem e demonstram como este modelo evoluiu do modelo de grades computacionais.

Neste trabalho será adotada a visão do NIST, que aponta as seguintes características sobre computação em nuvem: compartilhamento de recursos; acesso aos recursos sob demanda; provisão destes recursos usando um modelo de serviços; e facilidade em acessar e configurar serviços oferecidos.

Com relação ao provisionamento de recursos, diferentes tipos de recursos podem ser oferecidos. Desde aplicativos, plataformas para desenvolvimento, ou mesmo a disponibilização de servidores ou *desktops* virtuais, tudo através da Internet. As próximas seções apresentam uma visão geral dos principais modelos de serviços, de acordo com os tipos de recursos disponibilizados. Outros trabalhos também descrevem estes modelos de serviços [7]-[8]-[9], assim como [10], que propõe uma ampla taxonomia para os tipos de serviços comumente oferecidos.

A. SaaS (Software-as-a-Service)

Neste modelo de serviço, aplicações são executadas e hospedadas em uma Nuvem de uma organização. Estas aplicações são pagas por tempo de uso (*pay-as-you-go*) e não é necessário aquisição de uma cópia de uso. Toda a área de gerenciamento da aplicação, como atualizações, manutenção e monitoramento, ficam sob responsabilidade do provedor de serviço. Esta abordagem evita que o usuário se envolva com aspectos tecnológicos. Dessa forma, o cliente do serviço fará uso da aplicação com o foco apenas no negócio e não no gerenciamento de recursos e infraestrutura de sua organização.

Como exemplo, um editor de texto no modelo tradicional precisa de uma cópia para uso e sua respectiva licença para funcionar em um computador. O usuário adquire esta licença para uso 24h, 7 dias por semana, mesmo que não o utilize a noite ou nos finais de semana. Se ele tiver que trabalhar em casa para suprir o trabalho do escritório e tiver que usar outro computador, precisará de uma segunda licença, aumentando gastos e inflexibilidade ao usuário. No modelo SaaS, esse problema é resolvido, pois o usuário paga apenas pelo que usa e pode utilizar a mesma licença em qualquer lugar, a qualquer instante, bastando que o usuário tenha acesso um computador ligado à Internet.

Ferramentas corporativas, que antes exigiam investimento financeiro elevado e requisitos específicos de infraestrutura, como ferramentas para *Enterprise Resource Planning* (ERP), *Customer Relationship Management* (CRM), *Business Process Management* (BPM), encontram-se disponíveis no modelo SaaS [11]-[12]-[13]-[14]. Dentre as vantagens deste modelo destacam-se o valor de investimento atrativo, além de exigir pouca ou nenhuma mudança na infraestrutura da empresa.

B. IaaS (Infrastructure-as-a-Service)

Este modelo oferece infraestruturas virtuais para os clientes. Portanto, não é necessário que o cliente adquira máquinas físicas, mas sim permitir que este acesse máquinas virtuais com as características desejáveis. O cliente não tem controle sobre a infraestrutura em que a máquina reside, porém ele escolhe o sistema operacional, memória disponível, núcleos de processamento, etc.. É ainda permitido que ele utilize e instale aplicativos compatíveis a configuração da máquina virtual por ele escolhida. A Seção 3 descreve o funcionamento das máquinas virtuais que, além de serem usadas no modelo IaaS, também são utilizadas na construção da infraestrutura da nuvem em geral.

Neste tipo de serviço o cliente pagará um determinado valor por tempo de uso, sendo o custo depende das características da máquina virtual e/ou pelo número de instâncias utilizadas. Se ele alugar uma máquina com alta capacidade de processamento e armazenamento de dados, pagará um valor mais alto do que para uma máquina inferior. Empresas podem ainda adquirir um número elevado de máquinas virtuais para cada período do ano, aumentando e diminuindo sua infraestrutura conforme a necessidade de seu negócio. Este modelo, além de diminuir consideravelmente os gastos da empresa com hardware e suporte, também ocasiona economia

de energia e oferece melhor aproveitamento de espaço físico.

C. PaaS (Platform-as-a-Service)

Este modelo de serviço oferece ao usuário uma plataforma de desenvolvimento completa e remotamente hospedada na Nuvem. Taurion [15] define esta plataforma como uma plataforma para criação e teste de aplicações. Neste modelo de serviço são disponibilizadas ferramentas para desenvolvimento, administração e gerenciamento, além de serviços de runtime. Portanto, é possível utilizar IDEs (Integration Development Toolkit), compiladores, ferramentas para gerenciamento e controle de versão ou ferramentas para teste de sistemas, incluindo teste automatizado ou teste de desempenho, por exemplo. PaaS pode ser dividido em dois tipos distintos, chamados PaaS aberto e PaaS fechado. O primeiro utiliza plataformas padrão, mantendo compatibilidade com todos os outros PaaS abertos, o que possibilita a livre migração. Já o PaaS fechado, é composto por plataforma única ou proprietária, de impossível migração.

A forma de cobrança deste serviço varia de empresa para empresa. Por exemplo, a empresa Heroku calcula o valor a ser cobrado através dos “dynos”, onde quatro dynos equivalem a um núcleo de CPU. Quanto mais dynos o cliente exige mais ele pagará. O cliente pode também aumentar a capacidade de seu ambiente de teste para aplicativos (Shared Database), pagando por este aumento de capacidade. Exemplos deste modelo de serviço são o Google App Engine [16] e Aneka [17].

A Figura 1 ilustra o funcionamento geral de uma nuvem computacional e as atividades de um provedor de serviços em nuvem. Provedores são capazes de implantar novos serviços (SaaS, IaaS ou PaaS) e torná-los acessíveis aos clientes através da Internet. O gerenciamento da infraestrutura física na qual estes serviços estão alocados é de responsabilidade do provedor de serviço, que deve fazer sua manutenção e provisão de recursos, de modo a garantir níveis de qualidade de serviço estipulados em contrato. Os clientes, uma vez que tenham contrato de serviços com o provedor, acessam a infraestrutura da nuvem usufruindo dos serviços oferecidos.

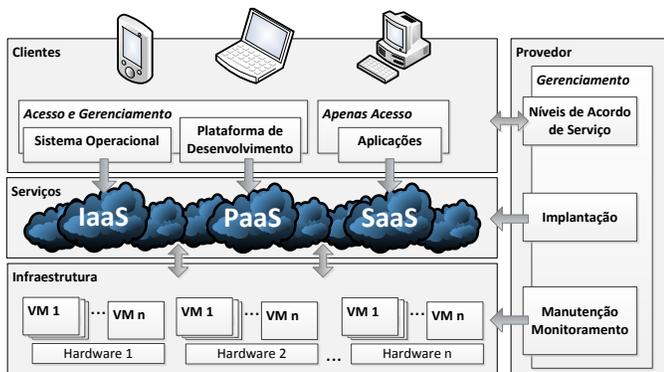


Fig. 1. Nuvem Computacional

Com relação à implementação, as Nuvens Computacionais também podem ser divididas em:

1. Nuvem Privada: Infraestrutura utilizada exclusivamente por uma ou mais empresas ou agências governamentais, onde

se faz necessário a utilização de níveis mais rigorosos de privacidade. Nesse caso, toda a infraestrutura de Nuvem é montada para um público restrito, não sendo oferecidos serviços ao mercado em geral.

2. Nuvem Pública: Infraestrutura criada para o público em geral ou grandes grupos industriais, normalmente utilizado por empresas que fornecem SaaS.

3. Nuvem Híbrida: Infraestrutura formada por no mínimo duas nuvens, uma pública e outra privada. Este modelo apresenta o benefício da utilização de recursos disponíveis em nuvens públicas, ao mesmo tempo em que se beneficia da segurança das nuvens privadas.

4. Nuvem Comunitária: Infraestrutura formada por uma Nuvem privada e compartilhada, onde há troca de informações, dados, pesquisas, etc. Normalmente é utilizada por grupos com o mesmo objetivo, como, por exemplo, uma infraestrutura de nuvem compartilhada entre universidades. Esta Nuvem pode ser administrada por uma das organizações do grupo ou por uma organização fora dele responsável apenas por administrá-la. A Figura 2 ilustra a política de administração de nuvens comunitárias com administração feita por provedor externo ou interno ao grupo que compõe a nuvem.

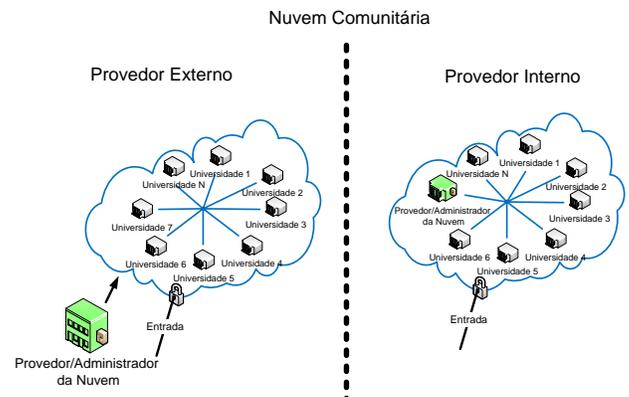


Fig. 2. Nuvem Comunitária com administração interna e externa

D. Tecnologias para Computação em Nuvem

Esta seção discute tecnologias relacionadas à implementação de infraestruturas e serviços para computação em nuvem. Empresas como a Amazon, Google e Microsoft, têm suas próprias infraestruturas de nuvem computacional, oferecendo serviços ao mercado.

1. Amazon Web Services - AWS

A Amazon, uma das empresas pioneiras em IaaS, oferece atualmente o conjunto de serviços Amazon Web Services (AWS). Dentro deste conjunto de serviços encontram-se serviços conhecidos como Amazon Simple Storage Service (S3), que fornece um ambiente virtual na Web para armazenamento ilimitado de qualquer tipo de dados e recuperação dos mesmos. O dado armazenado pode variar de 1 byte a 5 GB e seu local de armazenamento pode ser definido como um sistema de pastas e subpastas acessíveis pelos sistemas operacionais atuais. Este ambiente pode ser coletivo ou individual, sendo possível configurar restrições de acesso para cada objeto armazenado no S3 [18]-[19]. O Amazon

Elastic Compute Cloud (EC2) é outro serviço Web oferecido dentro do AWS. Basicamente ele disponibiliza máquinas virtuais baseadas no sistema operacional Linux. Este serviço garante total flexibilidade à infraestrutura do usuário, permitindo que aumento ou redução da capacidade de computação baseado na sua demanda. Se for necessário aumentar temporariamente a capacidade de computação, pode-se ativar instâncias virtuais rapidamente e após o uso excluí-las. Dentro do AWS existem outros serviços como *Amazon SimpleDB* (SDB), *Amazon Simple Queue Service* (SQS), *Amazon Virtual Private Cloud* (VPC) [20]- [9].

2. Google Application Engine

A empresa Google disponibiliza serviços PaaS, através da plataforma de desenvolvimento Web *Google Application Engine* ou *App Engine*. Nesta plataforma, é possível criar aplicações Web utilizando a nuvem da Google, que possui ambientes para desenvolvimento Java, Python e Go. Além disso, a *App Engine* foi projetada para ser independente de linguagem de programação, então futuramente terá compatibilidade com outras linguagens de programação. Este serviço oferece um sistema de armazenamento chamado *Datastore*, baseado em uma estrutura escalável de armazenamento de dados definida por *BigTable*, ambos desenvolvidos pela Google [20].

3. Azure

A Microsoft entrou no mercado com o *Windows Azure*, o PaaS da Microsoft, uma plataforma de serviços projetada como ambiente de desenvolvimento e hospedagem de serviços e aplicações Web. Esta plataforma possibilita ao seu utilizador criar aplicações baseados em .NET com as ferramentas do *Azure* e armazená-las na nuvem da Microsoft. Com o *Azure*, também é possível programar em C#, C++, PHP e outras linguagens suportados pela plataforma .NET. O *Windows Azure* fornece um conjunto de serviços para seus usuários, sendo os mais utilizados:

a) *Windows Azure AppFabric*: É um *middleware* para desenvolvimento, implementação e controle de aplicações dentro do *Windows Azure*. Junto a esta plataforma são disponibilizadas outras ferramentas para auxiliar o programador a desenvolver seus aplicativos: Há serviços orientados à mensagens baseado em *Service Bus*, garantindo persistência e fácil distribuição de mensagens; Controle de acesso à plataforma, o *Access Control*, para desenvolvimento em equipe, e o *Caching* para *cache* de dados em memória e outros componentes para integração de sistemas e automatização do desenvolvimento.

b) *SQL Azure*: É um Sistema Gerenciador de Banco de Dados relacional baseado no *Windows Server* e no *SQL Server*. Ele oferece a escalabilidade e funcionalidades de um *Datacenter* flexível, pois consegue lidar com diferentes tipos de carga. Ele também dimensiona e particiona os dados, enquanto seus dados crescem. Todo o gerenciamento e instalação são de responsabilidade da Microsoft.

c) *Windows Azure Storage Service*: É o serviço de armazenamento nas nuvens (Bancos de Dados da Microsoft) oferecido pelo *Azure*, seus dados podem ser armazenados de quatro formas: *Binary Large Objects* (BLOBs) que armazena

dados binários como arquivos ou imagens; *Queues* que armazena filas de mensagens, usadas para troca de informação entre os serviços do *Azure*; *Tables* que armazena informação estruturada com alta disponibilidade e escalabilidade; e *Windows Azure Drive* que funcionam como discos locais e é nele que estão armazenados os BLOBs.

4. MapReduce/Hadoop

Um serviço muito conhecido de nuvem computacional é o *MapReduce*, um modelo de programação e *framework* desenvolvido pelo Google, para facilitar o processamento e manipulação de grandes volumes de dados de forma paralela e/ou distribuída em aglomerados. O sistema controla as operações de tolerância a falha, balanceamento de carga e escalonamento de I/O permitindo ao usuário abstração de concorrência com alta disponibilidade.

O *MapReduce* processa dados em duas etapas básicas: a primeira, chamada de *Map* ou Mapeamento, gera um conjunto intermediário de pares chave/valor da cada entrada de dados (a forma deste conjunto é definida pelo usuário) antes de passar para segunda etapa. O *framework* ordena os conjuntos de pares pelas chaves e agrupa todos os valores de mesma chave em um par, criando um novo conjunto intermediário de pares chave (lista de valores). Em seguida é executada a segunda etapa, *Reduce* ou Redução. Nesta etapa, cada novo par intermediário processará todos os valores de uma mesma chave e gerará um novo conjunto final de pares chave/valor. Em resumo, na etapa de Mapeamento são encontrados dados que, depois de ordenados pelo *framework*, são indexados e os resultados agrupados.

Hoje existem diversas implementações baseadas em *MapReduce*, como: *Disco*, *Skynet*, *FileMap*, *Greenplum* e, em especial, *Hadoop* um *framework open source* desenvolvido em linguagem Java que permite a criação de aplicações distribuídas no modelo *MapReduce*. Essas aplicações são utilizadas para processar grandes volumes de dados em ambientes "clusterizados", o *Hadoop* pode armazenar e processar petabytes de informação facilmente, pois ele distribui dados e processamento através de *clusters* automaticamente. O *Hadoop* tem um sistema de arquivos distribuído próprio, o *Hadoop Distributed File System* (HDFS). Este separa os arquivos de uma entrada em blocos de dados e os divide entre os nodos do *cluster* criando um ambiente de execução paralela. Para maior segurança, o HDFS cria cópias idênticas dos blocos de arquivos de cada nodo em outros nodos, como meio de recuperar blocos de dados caso na ocorrência de falhas.

5. Eucalyptus

Para fins de pesquisa, há interesse em estudar desempenho, segurança e confiabilidade do modelo de Computação em Nuvem, não bastando, portanto, utilizar infraestrutura e serviços disponíveis no mercado. Neste contexto que se destaca o *Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems* ou *Eucalyptus* [21]-[22], desenvolvido pela Universidade da Califórnia e comercializado atualmente como *Eucalyptus Systems Inc*. O *Eucalyptus* é um *software* gratuito de código aberto, que permite criar nuvens públicas ou privadas, sendo compatível

com o API da Amazon, EC2 API, permitindo fácil migração de seus aplicativos de uma nuvem para outra. Esse *software* é, atualmente, uma boa solução de Computação em Nuvem privada para clientes VMware, suportando também outros gerenciadores de máquinas virtuais, como Xen e KVM.

Em [21]-[22] é apresentada sua arquitetura, composta por cinco componentes:

a) *Node Controller* ou Controlador de Nó (NC): Executa em todos os nodos da nuvem, ou seja, máquinas físicas da nuvem que hospedarão as máquinas virtuais. Este controlador é responsável pelo gerenciamento das máquinas virtuais (VMs – *Virtual Machines*), podendo iniciar, parar e colher informações sobre as VMs. O NC é subordinado ao Controlador de *Cluster* (CC), portanto, para acessá-lo é necessário fazê-lo por intermédio do CC.

b) *Cluster Controller* ou Controlador de *Cluster* (CC): É responsável por gerar as solicitações de execução das VMs. A partir de requisições aos NC ele recolhe dados sobre as VMs de um *Cluster* ou rede privada. Ele também é responsável pelo escalonamento das requisições de instanciação das máquinas virtuais entre os NC e balanceamento de carga. É este componente que decide em qual NC será melhor atribuir uma tarefa específica, tomando essa decisão baseado em dados coletados sobre requisições e carga dos nodos. Dentro de uma nuvem podem existir múltiplos Controladores de Clusters.

c) *Storage Controller* ou Controlador de Armazenamento (SC): É um bloco de armazenamento elástico mais interno do sistema. É neste módulo que são armazenados os dados dos Nodos em volumes. O SC é compatível com EC2, podendo ser utilizado como bloco de armazenamento. Sua utilização permite a persistência de dados nas instâncias do Eucalyptus.

d) *Walrus* (W): *Walrus* é um componente de armazenamento do tipo *put/get* que utiliza tecnologias de *Web Services*. Ele possui interface com usuário compatível com a API *Amazon Simple Storage Service* (Amazon S3), sendo possível inserir e retirar dados do *Walrus*. O *Walrus* armazena dados dos usuários e também imagens de VMs.

e) *Cloud Controller* ou Controlador de Nuvem (CLC): Este é o único componente da arquitetura que usuários possuem acesso direto a partir da Internet. Ele é responsável por atender requisições referentes à manipulação de máquinas virtuais e fornecimento de informações sobre o estado das VMs. Para obter estas informações, o CLC comunica-se diretamente com o CC. O CLC controla também o armazenamento de dados dos usuários e do sistema (*Walrus*), possibilitando ao usuário e/ou administrador obterem informações sobre os recursos disponíveis e o estado atual do sistema. É dentro do CLC que estão às interfaces para usuários e administradores, normalmente implementadas em SOAP, Query ou Web, tornando mais simples o uso e a visão da Nuvem.

A Figura 3 mostra uma possível estrutura de Nuvem Eucalyptus. Os elementos da arquitetura podem compartilhar um mesmo servidor ou encontrarem-se dispostos separadamente. Como exemplo, veja o SC e o CC no Cluster 1 e no Cluster 2. Esses elementos podem ser organizados na infraestrutura de diferentes maneiras. O SC e CC podem estar localizados em uma mesma máquina física ou podem estar em

máquinas independentes. No entanto, para se obter uma melhor escalabilidade e desempenho, é recomendada maior distribuição entre os elementos da arquitetura. Outro aspecto importante é a homogeneidade do hardware empregado nos nodos de processamento de cada cluster.

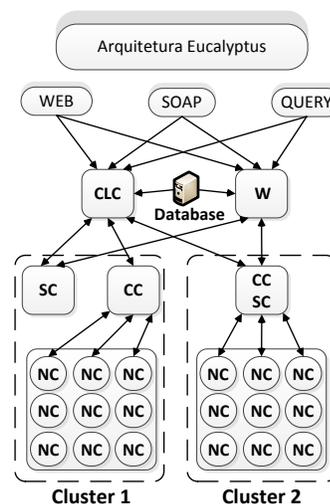


Fig. 3. Arquitetura do Eucalyptus

III. MÁQUINAS VIRTUAIS

A. Características Básicas

Virtualização de servidores refere-se à aplicação de técnicas que permitem o particionamento de recursos de servidores físicos entre vários sistemas operacionais que executam concomitantemente em uma mesma máquina física. A disponibilização de parte desses recursos é feita através do uso de Máquinas Virtuais (VMs - *Virtual Machines*) que, através de uma instância de um SO (Sistema Operacional), são capazes de acessar os recursos físicos do computador de modo transparente ao usuário. Portanto, esta é definida como uma duplicata isolada e eficiente de uma máquina real [23].

Em ambientes virtualizados, além da presença do SO que é executado diretamente sobre *hardware* físico (SO hospedeiro), VMs e SOs executados sobre *hardwares* virtuais (SOs hóspedes ou visitantes), existe o monitor de máquinas virtuais (VMM - *Virtual Machine Monitor*) [24]. O VMM, também conhecido por *Hypervisor*, é responsável pelo gerenciamento e controle dos recursos compartilhados pela máquina física (por exemplo, processador, memória, disco rígido e dispositivos de E/S). Em outras palavras, o VMM implementa uma camada de *software* com o encargo de fazer o escalonamento e o tratamento de instruções destas VMs. Também é responsabilidade do VMM fornecer ao SO visitante a abstração de VM [25]. A Figura 4 representa uma possível disposição dos elementos descritos.

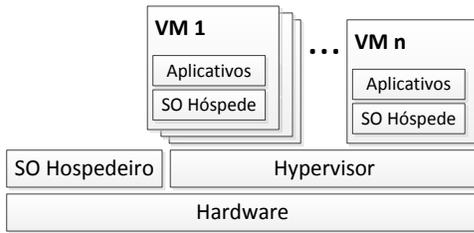


Fig. 4. Modelo de virtualização híbrida.

Atualmente, é possível encontrar diferentes abordagens na implementação de virtualização de servidores, as principais são: virtualização total e paravirtualização. O que essencialmente as difere é a maneira na qual os *Hypervisors* tratam a execução de instrução sensível por SOs visitantes. Assim chamada por Popek e Goldberg em 1974 [23], instrução sensível são todas as instruções que executam apenas em modo supervisor, como por exemplo: instruções que modificam ou lêem o estado de registradores; e instruções de comunicação com dispositivos (ex. instruções que geram interrupções de E/S). Esses pesquisadores também propuseram o conceito de instrução privilegiada, instruções que são capturadas por armadilha (*trap*) em programas executando no modo usuário. Estas armadilhas indicam ao SO qual interrupção deve ser executada em modo supervisor, permitindo que interrupções sejam geradas no modo usuário, porém tratadas posteriormente pelo sistema operacional em modo privilegiado.

A virtualização total pode ser implementada de duas maneiras. A mais atual utiliza *hypervisor* tipo 1, desenvolvido para CPUs com arquitetura x86 estendida com suporte a virtualização. Já para CPUs 386 e sucessoras, sem tal suporte, usa-se o *hypervisor* tipo 2. Em ambas as implementações, as VMs executam em modo usuário e os SOs hóspedes em modo supervisor. A união dessas duas características acarreta na execução de instrução sensível em modo usuário. Isso poderia ser um problema, pois, se não houvesse nenhum tipo de tratamento, estas falhariam e o SO hóspede pararia de funcionar. Cada um dos *hypervisors* possui um comportamento próprio ao lidar com essa situação. Estes comportamentos são conhecidos por: captura e emulação (*trap-and-emulate*) ou tradução binária [26].

Com a utilização do *hypervisor* tipo 1, ocorrerá *trap-and-emulate*. Em *hardware* com suporte a virtualização, existe uma armadilha a espera de instrução sensível executada em modo usuário. Quando a armadilha captura alguma instrução com esse perfil, o *hypervisor* inspeciona a origem da instrução capturada. Caso esta tenha sido executada por algum SO hóspede, o *hypervisor* executará a instrução. Caso contrário, o *hypervisor* emula a tomada de decisão que o *hardware* teria ao se deparar com uma instrução sensível executada em modo usuário por um processo qualquer.

A tradução binária, feita por *hypervisors* tipo 2, varre o código binário de execuções em busca de instruções sensíveis. Quando alguma é encontrada, esta é substituída por uma chamada a uma rotina de *hypervisor*, que, por sua vez, irá executar a instrução e simular a execução ao SO hóspede. As demais instruções são executadas diretamente no hardware.

A paravirtualização, diferentemente da virtualização total, necessita que os SOs hóspedes tenham o código fonte alterado. Essa alteração é feita para que os SOs hóspedes façam chamadas de *hypervisor* ao invés de executarem instruções sensíveis. Assim, é dito que o SO hóspede possui ciência de estar sendo executado sobre uma VM.

B. Principais VMMs Existentes

Existem diversas soluções em *software* disponíveis para distribuições Linux ou Windows, que provêm suporte a virtualização de servidores. Porém, nem todas são baseadas na virtualização total e/ou paravirtualização. Há outros tipos de virtualização: emulação (ou simulação) e virtualização em nível de sistema operacional. Também é importante salientar que a emulação não é considerada como um tipo de virtualização por alguns autores [27]. Serão apresentadas nesta seção as principais ferramentas de virtualização em uso atualmente.

1. VMware

Infraestrutura de virtualização que oferece diversas aplicações para suporte a ambientes virtualizados. A maioria são aplicações comerciais, sendo algumas gratuitas, com recursos reduzidos. A abordagem empregada na virtualização de servidores nos produtos VMware é a virtualização total. Os produtos gratuitos existentes que realizam essa técnica são: *VMware vSphere Hypervisor* e *VMware Server*. O primeiro executa diretamente sobre o hardware, sendo este um SO hospedeiro com *hypervisor* tipo 1. Já o segundo é instalado como aplicação no SO hospedeiro, ou seja, é um *hypervisor* tipo 2. Em se tratando de produtos da VMware, é comum a utilização dos termos *bare metal* e *hosted* para se designarem respectivamente os tipos 1 e 2 de *hypervisors*. O SO hospedeiro é quem fornece o suporte aos dispositivos através do *VMDriver*. Estas diferenças entre as duas tecnologias podem ser observada nas Figuras 5 e 6.

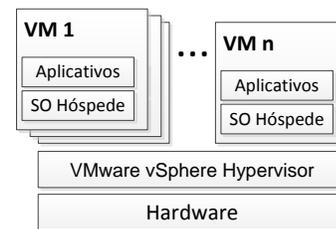


Fig. 5. VMware vSphere Hypervisor: virtualização total, *hypervisor* tipo 1

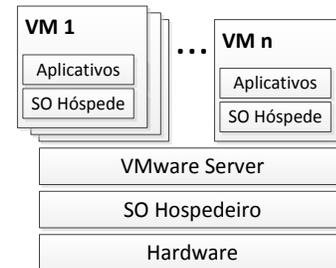


Fig. 6. VMware Server: virtualização total, *hypervisor* tipo 2

2. Xen Hypervisor

Software de código aberto sob licença GPL2 (GNU - *General Public License*), com suporte a paravirtualização e virtualização total (*hypervisor* tipo 1). Essa ferramenta faz distinção entre os SOs hóspedes totalmente virtualizados e paravirtualizados, nomeando-os respectivamente de *Domain U PV Guests* (PV - *Paravirtualization*) e *Domain U HVM Guests* (HVM - *Hardware Virtual Machine*). Os domínios U (domU) são chamados de não-privilegiados. Divergindo ao conceito de SO hospedeiro, existe um domínio privilegiado, formalmente conhecido por *Domain 0 Guest* (dom0). Conforme sugere o nome dos domínios, apenas existem SOs virtualizados (hóspedes) em ambientes com a presença do *Xen Hypervisor*. Além das diferenças já discutidas quanto aos tipos de virtualização, para entender o funcionamento dessa tecnologia é fundamental saber a diferença de domínio privilegiado (domínio 0) e domínios não-privilegiados (domínios U) [28].

O domínio 0 assim como os demais *PV Guests*, necessita de um núcleo modificado. O que o faz privilegiado é a possibilidade de acesso direto ao *hardware*. Isso é possível através dos *drivers* físicos de dispositivos de entrada/saída existentes nele. Também se encontram presentes *drivers* especiais que tratam requisições provenientes das VMs dos domínios U para acesso a rede e disco. A VM do domínio 0 sempre é a primeira a ser executada pelo *hypervisor*, pois, em adição às características recém descritas, o domínio 0 possui caráter administrativo. É através dele que são criadas, inicializadas e terminadas as outras VMs. Diferente dos *PV Guests*, *HVM Guests* não possuem *drivers* virtuais que interagem com os *drivers* especiais presentes no domínio 0. Essa ausência é devido ao fato de que os *HVM Guests* não foram modificados. Por esse motivo, os SOs que compõe esses domínios comportam-se da mesma maneira que se comportariam caso fossem executados sobre máquina física. Cabe ao *Xen virtual firmware* simular a existência da BIOS (*Basic Input/Output System*) e ao QEMU emular o *hardware* disponível para as VMs dos *HVM Guests* [29]. Veja na Figura 7 algumas vias de interação entre o *Xen Hypervisor* e seus domínios PV.

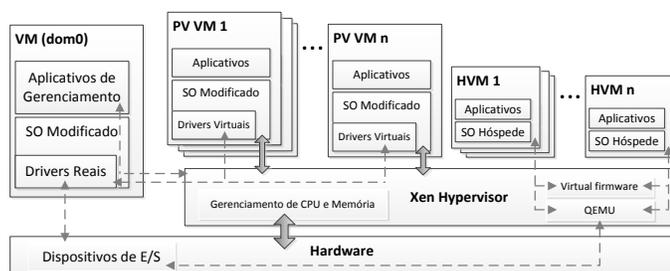


Fig. 7. Componentes do Xen, hypervisor e domínios [29]

3. KVM

Kernel-based Virtual Machine, *software* de código aberto para o suporte a virtualização total, possui *hypervisor* tipo 1. Esse *software* foi concebido para executar instâncias de servidores virtualizados sem interface gráfica. Nos últimos anos o KVM alcançou boa parte do mercado de virtualização de servidores para distribuições Linux. Distribuições como

Ubuntu e *Red Hat Enterprise Linux* que outrora davam suporte ao *Xen Hypervisor*, voltaram-se para adoção do KVM como principal VMM.

Assim como VMware e Xen, o KVM é capaz de efetuar *Live Migration*. Essa característica permite a transferência de uma máquina virtual em execução para outro servidor hospedeiro, sem que haja o desligamento ou perda de informações do estado atual da VM. Portanto, é possível efetuar manutenção de infraestrutura ou balanceamento de carga em servidores hospedeiros sem interrupção de execução.

C. Hypervisors em Nuvens

Grandes empresas que oferecem serviços em nuvem computacional possuem *hypervisors* próprios para a virtualização da infraestrutura. A implementação desses *hypervisors* visam extrair o melhor aproveitamento do tipo específico de suas infraestruturas computacionais. A empresa Amazon utiliza o *Xen Hypervisor* com alterações personalizadas no código fonte. O *Windows Azure* utiliza o *Wazure Hypervisor*, solução não comercial, baseada em uma solução fornecida pela própria empresa, o *Hyper-V*.

IV. IMPLEMENTAÇÃO DE UMA NUVEM

A. Infraestrutura Disponível

Atualmente o C3 possui um ambiente de teste exclusivo para estudos de computação em nuvem e computação de alto desempenho. A infraestrutura para implantação de nuvem computacional é composta por três servidores *Dell PowerEdge T300* e um *switch*. Cada servidor possui processadores Intel Xeon com quatro núcleos X3363 - 2,83GHz, 8GB de memória RAM DDR2, 1TB em disco rígido e quatro interfaces de rede padrão *Gigabit Ethernet*.

A seguir são apresentados alguns aspectos referentes à implementação da nuvem, seu gerenciamento e serviços oferecidos. A documentação completa com as instruções de instalação e configuração estão disponíveis em [30].

B. Softwares Utilizados e Instalação

Como sistema operacional utilizou-se o *Ubuntu Server* versão 11.04. Os demais *softwares* para a implantação da nuvem computacional foram o *Eucalyptus* e o *KVM*. A instalação dos componentes do *Eucalyptus* foi distribuída da seguinte maneira: *Cloud Controller* e *Walrus*, assim como o *Storage Controller* e *Cluster Controller* instalados em um mesmo servidor; o *Node Controller* foi configurado em um terceiro servidor. A Figura 8 mostra a disposição da infraestrutura da nuvem e sua rede interna.

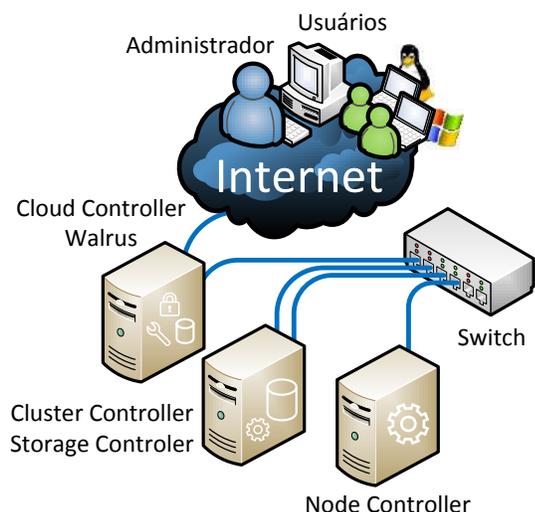


Fig. 8. Infraestrutura da nuvem computacional do C3

Nesta infraestrutura apenas o *Cloud Controller* e o *Walrus* tem acesso direto a rede externa, os demais componentes são acessados indiretamente pelo repasse de pacotes, ou seja, o 1º servidor (CLC/W) faz o roteamento dos pacotes para o 2º servidor (CC/SC) que, por sua vez, faz o roteamento para o 3º servidor (NC).

C. Gerenciamento

A administração da nuvem é feita remotamente. O controle pode ser feito através de interfaces gráficas como *Hybridfox* e *Eucalyptus Web User Interface*, ou pelo próprio terminal. Todas as formas usam direta ou indiretamente os comandos providos pela ferramenta *euca2ools*, que é um conjunto de ferramentas fundamental para o controle e gerenciamento da nuvem. Por exemplo, com o *euca2ools* é possível criar e manipular VMs, instâncias virtuais, *volumes* e *snapshots* (gerenciador de *backup* de *volumes*). A Tabela 1 mostra as combinações de VM que podem ser disponibilizadas pela Nuvem do C3. Cabe ressaltar, que estes tipos são configuráveis de acordo com a necessidade dos utilizadores da nuvem, porém podem ser alterados apenas usuários administradores da nuvem.

Table I
Variables to be considered on the evaluation of interaction techniques

| Tipo de VM | CPUs | Memória | Disco |
|------------|------|---------|-------|
| m1.small | 1 | 192 MB | 2 GB |
| c1.medium | 1 | 256 MB | 5 GB |
| m1.large | 2 | 512 MB | 10 GB |
| m1.xlarge | 2 | 1024 MB | 20 GB |
| c1.xlarge | 4 | 2048 MB | 20 GB |

D. Serviços Oferecidos

Serão apresentados nesta seção alguns serviços a serem implementados na nuvem computacional do C3. Estes serviços têm por objetivo atender demandas de ampliação de infraestrutura, de processamento de dados, além de propor aplicações específicas para a instituição, para suporte às atividades de ensino, pesquisa e extensão.

Mesmo que a universidade não vise lucro com a prestação de serviços, ainda assim obtêm-se benefícios em manter infraestrutura de serviços e infraestrutura em nuvem computacional. Por exemplo, o consumo de energia e gastos com infraestrutura de *hardware* deve reduzir, visto que vários servidores podem executar em instâncias virtuais, não havendo necessidade em executá-los em infraestrutura física dedicada. Além disso, é esperado que o tempo para configuração e manutenção de serviços e servidores seja diminuído, aumentando a disponibilidade e manutenibilidade da infraestrutura.

1. Serviços para Centro de Processamento de Dados

Diversos serviços comuns em CPD's (Centros de Processamento de Dados) apresentam uma demanda significativa de *hardware*. Com a utilização de uma nuvem computacional, é possível designar um conjunto de máquinas virtuais para execução de servidores de *e-mail*, sistema de arquivos remotos, servidores de ftp, etc. Com este modelo está prevista uma diminuição nos custos com *hardware* e consumo de energia, sem comprometer a qualidade dos serviços oferecidos. O modelo de computação em nuvem provê o gerenciamento destes recursos e a possibilidade de migração de máquinas virtuais para outros nodos computacionais, além de rápida recuperação de configurações pré-definidas, no caso de necessitar reparos.

2. Repositórios de Dados

Pretendem-se utilizar serviços da nuvem para controle e acesso dos dados na nuvem. Dados de um usuário poderão ser transferidos a qualquer momento para a nuvem, de qualquer dispositivo com acesso a interface desta aplicação. Desta forma, a facilidade de acesso e disponibilidade de dados deve ser incrementada. Sistemas de controle de versão, *middleware* orientados a mensagem, e técnicas de replicação de dados e tolerância a falhas, devem fazer parte da solução proposta. Além disso, para aumentar a acessibilidade dos dados por diversos dispositivos, pretende-se montar uma arquitetura orientada a serviços capaz de atender requisições provenientes dos mais diversos dispositivos, como computadores pessoais, dispositivos móveis, etc.

3. Pool de Aplicações

Em uma nuvem computacional é possível disponibilizar uma variedade de aplicações com acesso sob demanda. Portanto, o usuário poderá escolher, dentre um conjunto de aplicações oferecidas pela nuvem, quais ele quer usar sem se preocupar com a localização ou configuração da aplicação. Como exemplos podem ser citados aplicativos para gerência de tarefas, agendas pessoal ou global, fóruns, gerência e edição de sites, entre outras.

4. Pool de Recursos

De forma similar ao *pool* de aplicações, a nuvem também oferecerá um *pool* de recursos. O usuário poderá solicitar máquinas virtuais com configurações específicas e então trabalhar na instância solicitada. Esta abordagem é comum no processo de desenvolvimento de *software*, por exemplo, nas atividades de teste de sistema. Uma equipe de testes pode verificar a compatibilidade dos sistemas desenvolvidos em ambientes com diferentes configurações de *hardware* ou ainda com diferentes sistemas operacionais.

V. CONCLUSÃO E TRABALHOS FUTUROS

Este artigo apresentou um estudo sobre infraestruturas usadas na construção de nuvens computacionais. A definição de nuvem computacional, assim como os conceitos relacionados a este modelo de computação foram destacados, juntamente com a descrição de algumas ferramentas de suporte a nuvem e máquinas virtuais. Foi feita uma análise dos principais tipos de serviços oferecidos por nuvens computacionais, com a finalidade de mostrar ao leitor o potencial destas infraestruturas em ambientes corporativos e acadêmicos.

Para ilustrar um estudo de caso real, foi discutida a implementação de uma nuvem computacional no C3 da universidade FURG. Foram descritos o *hardware* utilizado, ferramentas de controle da nuvem e de gerenciamento de recursos virtuais. Uma das contribuições deste artigo é dar subsídios ao leitor, permitindo que esta infraestrutura possa ser reproduzida.

Após a implantação da nuvem computacional em nosso centro, vislumbramos a implementação de uma série de serviços. Tais serviços foram listados e detalhados ao longo deste artigo. A implementação destes serviços será feita em trabalhos futuros, bem como a avaliação dos benefícios que estes venham a trazer.

Como trabalhos futuros, além da implementação de novos serviços, pretende-se pesquisar técnicas de monitoramento e gerenciamento de recursos, utilizando a infraestrutura da nuvem do C3. Portanto, iremos avaliar como identificar, em tempo de execução, o desempenho das máquinas virtuais ativas. Baseado nos dados de monitoramento serão propostas técnicas preventivas para minimizar os efeitos de degradação de desempenho que certas aplicações podem ocasionar. Esta análise está relacionada à verificação de níveis de acordo de serviço e qualidade de serviços com cargas de trabalho variáveis.

REFERÊNCIAS

- [1] F. Cappello, S. Djilali, G. Fedak, T. Herault, F. Magniette, V. Néri, and O. Lodygensky, "Computing on Large-Scale Distributed Systems: XtremWeb Architecture, Programming Models, Security, Tests and Convergence with Grid," *Future Generation Computer Systems*, vol. 3, pp. 417–437, 2005.
- [2] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems Concepts and Design*. Boston: Addison Wesley, 1994.
- [3] A. Iosup, C. Dumitrescu, D. Epema, H. Li, H. and L. Wolters. "How are Real Grids Used? The Analysis of Four Grid Traces and its Implications," in: *IEEE/ACM International Conference on Grid Computing*, 7th, IEEE Computer Society, 2006.
- [4] J. A. Stankovic, "A perspective on distributed computer systems," *Computers, IEEE Transactions on*, C-33, vol. 12, pp. 1102–1115, 1984.
- [5] P. Mell, and T. Grance. (2009). The NIST Definition of Cloud Computing. *National Institute of Standards and Technology*. [Online] Gaithersburg: National Institute of Standards and Technology. Disponível em: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909616.
- [6] L. M. Vaquero, L. R. Merino, J. Caceres, and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2008.
- [7] A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm, "What's Inside the Cloud? An Architectural Map of the Cloud Landscape," in: *ICSE Workshop on Software Engineering Challenges of Cloud Computing, CLOUD '09, IEEE Computer Society: 2009*.
- [8] F. R. C. Sousa, L. O. Moreira, and J. C. Machado, "Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios," in: *ERCEMAPI 2009, SBC: 2009*.
- [9] J. Varia, "Cloud Architectures". Amazon, 2008. Disponível em: <http://jmeshvaria.s3.amazonaws.com/public/cloudarchitectures-varia.pdf>.
- [10] B. P. Rimal, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," in: *International Joint Conference on INC, IMS and IDC, 5th, NCM: 2009*.
- [11] Acumatica. *Acumatica Cloud ERP Software*. Disponível em: <http://www.acumatica.com/>.
- [12] Appian. *Appian Cloud BPM*. Disponível em: <http://www.appian.com/>.
- [13] Netsuite. *NetSuite ERP Cloud*. Disponível em: <http://www.netsuite.com>.
- [14] Salesforce. *SalesForce Social Enterprise*. Disponível em: <http://www.salesforce.com/br/?ir=1>.
- [15] C. Taurion, *Cloud Computing: Computação em Nuvem: Transformando o Mundo da Tecnologia da Informação*. São Paulo: Brasport, 2009.
- [16] Google. *Google App Engine*. Disponível em: <http://code.google.com>.
- [17] X. Chu, K. Nadiminti, C. Jin, S. Venugopal, and R. Buyya. "Aneka: Next-Generation Enterprise Grid Platform for e-Science and e-Business Applications." In: *IEEE International Conference on e-Science and Grid Computing, 3th, 2007*.
- [18] M. Ambrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia. "View of Cloud Computing". *Communications of the ACM*, vol. 4, pp. 50–58, 2010.
- [19] M. R. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel, "Amazon S3 for Science Grids: a Viable Solution?" in: *International Workshop on Data-Aware Distributed Computing, DADC '08, 2008*.
- [20] M. Ambrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia. (2009, Maio). Above the Clouds: A Berkeley View of Cloud Computing. *Berkeley University of California*. [Online]. Disponível em: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EICS-2009-28.pdf>. Acessado em: 02/04/2012.
- [21] S. Liu, Y. Liang, and M. Brooks, "Eucalyptus: a Web Service-Enabled e-Infrastructure," in: *Conference of the Center for Advanced Studies on Collaborative Research, CASCON '07, IBM: 2007*.
- [22] D. Nurmi, R. Wolski, C. Zgzegorzyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. "The Eucalyptus Open-Source Cloud-Computing System," in: *IEEE/ACM International Symposium on Cluster Computing and the Grid, 9th, 2009*.
- [23] G. J. Popek, and R. P. Goldberg, "Formal Requirements for Virtualizable Third Generation Architectures." *Communications of the ACM*, vol. 7, pp. 412–421, 1974.
- [24] J. E. Smith, and R. Nair, *An overview of virtual machine architectures*. Binghamton: Binghamton University, 2004. Disponível em: <http://www.cs.binghamton.edu/~kchiu/cs552-f04/papers/smith-04.pdf>.
- [25] D. M. F. Mattos. (2008). Virtualização: Vmware e xen. [Online] Disponível em: http://www.gta.ufgr.br/grad/08_1/virtual/artigo.pdf.
- [26] A. S. Tanenbaum, *Sistemas Operacionais Modernos*. SP: Pearson, 2009.

- [27] D. B. Gonçalves, J. C. J. Vahl, *White paper – virtualização*. Campinas: Polis de Tecnologia, 2008. Disponível em: http://www.sensedia.com/br/anexos/wp_virtualizacao.pdf.
- [28] P. Barham, P. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, “Xen and the art of virtualization,” in: *ACM Symposium on Operating Systems Principles. 2003 SOSP '03*.
- [29] A. Carissimi. “Virtualização: da teoria a soluções,” in: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), SBC: 2008*.
- [30] *Relatório Técnico 2012/01: Instalação de Nuvem Computacional utilizando o Eucalyptus*, C3, Centro de Ciências Computacionais. 2012. Disponível em: <http://www.hpc.c3.furg.br/arquivos/download/rt-2012-01-instalacao-eucalyptus.pdf>.