



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
TECNÓLOGO EM REDES DE COMPUTADORES

FRANCISCO RENATO CAVALCANTE ARAÚJO

**USO OPORTUNÍSTICO DE RECURSOS COMPUTACIONAIS
OCIOSOS ATRAVÉS DA VIRTUALIZAÇÃO**

**QUIXADÁ
2014**

FRANCISCO RENATO CAVALCANTE ARAÚJO

**USO OPORTUNÍSTICO DE RECURSOS COMPUTACIONAIS
OCIOSOS ATRAVÉS DA VIRTUALIZAÇÃO**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Tecnólogo em Redes de Computadores da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Tecnólogo.

Área de concentração: computação

Orientador Prof^o. Paulo Antonio Leal Rego

**QUIXADÁ
2014**

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca do Campus de Quixadá

A687u Araújo, Francisco Renato Cavalcante
 Uso oportunístico de recursos computacionais ociosos através da virtualização / Francisco
 Renato Cavalcante Araújo. – 2014.
 37 f. : il. color., enc. ; 30 cm.

 Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de
 Tecnologia em Redes de Computadores, Quixadá, 2014.
 Orientação: Prof. Me. Paulo Antonio Leal Rego
 Área de concentração: Computação

1. Computação em Nuvem 2. Virtualização (Computação) 3. Tecnologia da Informação I. Título.

CDD 004.36

FRANCISCO RENATO CAVALCANTE ARAÚJO

**USO OPORTUNÍSTICO DE RECURSOS COMPUTACIONAIS OCIOSOS ATRAVÉS
DA VIRTUALIZAÇÃO**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Tecnólogo em Redes de Computadores da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Tecnólogo.

Área de concentração: computação

Aprovado em: _____ / dezembro / 2014.

BANCA EXAMINADORA

Prof. MSc. Paulo Antonio Leal Rego (Orientador)
Universidade Federal do Ceará-UFC

Prof. MSc. Alisson Barbosa de Souza
Universidade Federal do Ceará-UFC

Prof. MSc. Michel Sales Bonfim
Universidade Federal do Ceará-UFC

Aos meus pais, Raimundo Alves de Araújo e Marilene Cavalcante de Araújo que apesar das dificuldades me deram educação e princípios, e meus irmãos pela união e preocupação entre ambos em todas as situações.

AGRADECIMENTOS

Agradeço a Deus por ter permitido mais essa conquista em minha vida e ter me dado a oportunidade de estar concluindo um curso superior.

Ao professor Paulo Antonio, pela orientação nesta pesquisa e na bolsa de iniciação à docência.

A todos os professores, que contribuíram para a minha formação acadêmica.

A todos os meus amigos do curso de redes de computadores, que tive a oportunidade de conhecer durante os três anos de curso.

Principalmente ao Marcos Ribeiro, por dividir seu apartamento durante um mês que o ônibus escolar da minha cidade não veio para Quixadá, evitando que eu perdesse aula.

Aos meus amigos do AP 102, Wellington Júnior, Joel Sousa e Matheus Silveira, por todos os momentos que passamos.

A minha namorada Lívia Gildo, pela compreensão e força durante os momentos difíceis. E a todos meus familiares e amigos que contribuíram pra que eu possa ter chegado aqui.

"A persistência é o menor caminho do êxito."
(Charles Chaplin)

RESUMO

A virtualização é uma tecnologia que permite o compartilhamento do *hardware* de máquinas físicas. Tal característica proporcionou o surgimento de novas tecnologias, dentre elas a computação em nuvem, que representa um modelo de abstração de recursos computacionais. Explorar a utilização dessas tecnologias propicia inúmeras vantagens, como confiabilidade dos dados, redução de espaços físicos e redução de investimentos financeiros na aquisição e manutenção de equipamentos de Tecnologia da Informação. Ao mesmo tempo, o avanço tecnológico vem possibilitando *desktops* mais robustos ao passo que instituições possuem muitos desses computadores que não tem o uso de seus recursos por completo. Este trabalho concentra-se em desenvolver uma solução para usar de forma oportunística recursos computacionais ociosos presentes em instituições, integrando esses recursos numa nuvem privada. Um estudo de caso foi realizado para validar a solução.

Palavras chave: Virtualização. Computação em nuvem. Computação oportunística.

ABSTRACT

Virtualization is a technology that allows the sharing of physical machines hardware. This feature gave the emergence of new technologies, among them cloud computing, which is a model of abstraction of computer resources. Explore the use of these technologies provides numerous advantages such as data reliability, reduction of physical spaces and reduction of financial investments in the acquisition and maintenance of information technology equipment. At the same time, technological advances has enabled more robust desktops while many of these institutions have computers that do not have the use of their resources completely. This work focuses on developing a solution to use opportunistically idle computing resources present in institutions, integrating these resources in a private cloud. A case study was conducted to validate the solution.

Keywords: Virtualization. Cloud computing. Opportunistic computing.

LISTA DE ILUSTRAÇÕES

Figura 1 – Abstração de recursos físicos computacionais na virtualização	15
Figura 2 – Virtualização de recursos computacionais.....	15
Figura 3 – Visão geral de uma nuvem computacional.	17
Figura 4 – Modelos de serviço da computação em nuvem.....	19
Figura 5 – Visão geral da proposta.....	25
Figura 6 – Integração de máquinas temporárias a nuvem	27
Figura 7 – Comunicação entre módulo cliente e módulo servidor.....	28
Figura 8 – Abstração da topologia proposta.....	31
Figura 9 – Tempo para ligar MFs.....	32
Figura 10 – Tempo para desligar MFs.....	33
Figura 11 – Tempo para desligar MF com MV em execução	33
Figura 12 – Cenários comuns	35
Figura 13 – Migração de MVs.....	36

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 Introdução	12
1.2 Objetivo Geral.....	13
1.3 Objetivos Específicos	14
1.4 Organização do Trabalho	14
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 Virtualização	14
2.2 Computação em nuvem	16
2.2.1 Características essenciais.....	17
2.2.2 Modelos de serviço.....	18
2.2.3 Modelos de implantação.....	20
2.3 Plataforma OpenNebula.....	20
2.4 Nuvem oportunística.....	21
3 TRABALHOS RELACIONADOS	22
4 PROPOSTA.....	24
4.1 Instalação da plataforma OpenNebula	25
4.2 Estratégia automatizada de integração de MFs à nuvem	26
4.2.1 Servidor	27
4.2.2 Cliente.....	28
4.3 Automatizar o desligamento/inicialização das MFs	30
5 ESTUDO DE CASO	30
5.1 Cenário.....	31
5.2 Experimentos	31
6 CONSIDERAÇÕES FINAIS	36
REFERÊNCIAS	37

1 INTRODUÇÃO

1.1 Introdução

Nos últimos anos, várias áreas do conhecimento têm adotado a computação para ajudá-las em suas pesquisas ao processar e armazenar dados, a fim de que eles estejam sempre guardados de forma segura, confiável e com alta disponibilidade. Essa dependência cria uma demanda quase insaciável de poder computacional, mas o acesso a esses recursos permanece um desafio, especialmente em países em desenvolvimento (OSORIO; CASTRO; BRASILEIRO, 2012). Este fato mostra o quanto a computação tem grande demanda, e que é preciso que esta área corresponda de forma crescente e acessível ao enorme número de usuários, que geram uma grande quantidade de informação a cada dia.

Com tanta informação a ser computada com alta complexidade, o que exige alto poder computacional, necessita-se de recursos suficientes e uma infraestrutura robusta para atender aos clientes e serviços. Para isso, é preciso investimento em equipamentos e conseqüentemente em manutenção, uma vez que a implantação de novos dispositivos na rede gera um grande gasto financeiro com material e pessoal (CASTRO et al., 2012), o que priva algumas organizações financeiramente de usarem esses serviços que detêm um alto custo computacional (DIAZ et al., 2013). Além disso, quanto maior for a quantidade de máquinas presentes em uma infraestrutura, maior será o consumo de energia e, futuramente, lixo eletrônico (WERNER, 2011).

A virtualização é uma técnica que permite a abstração de recursos computacionais (FERNANDES et al., 2010), possibilitando um melhor aproveitamento do *hardware* das máquinas, e tal característica abriu espaço para o surgimento de novas tecnologias. A evolução das técnicas de virtualização tem propiciado o desenvolvimento de aplicações que compartilham a mesma infraestrutura de *hardware*, alavancando o surgimento de soluções para os serviços em nuvem (VERDI et al., 2010). Segundo Azevedo et al. (2012), a computação em nuvem é um paradigma que prega a provisão de recursos dinamicamente escaláveis e normalmente virtualizados através da *Internet*.

As nuvens públicas são amplamente utilizadas, entretanto um número crescente de organizações está adotando o modelo de nuvem privada (BRANDWACHT et al., 2013). O conceito dessa tecnologia aborda a evolução da computação distribuída, a fim de alcançar um maior rendimento para solucionar problemas de computação de grande escala (RIMAL; CHOI; LUMB, 2009). A novidade está na forma em que os recursos físicos são distribuídos,

tais como armazenamento, CPU e rede, programação de *frameworks* e bibliotecas. Além disso, os serviços podem ser negociados como bens econômicos, oferecidos sob demanda através da *Internet* usando uma forma de pagamento pelo uso (LOUTAS et al., 2011).

A computação em nuvem se utilizou da computação oportunística, que segue o princípio de usar, quando existir oportunidade, capacidades de computação que estariam sendo desperdiçadas (OSORIO; CASTRO; VILLAMIZAR, 2012), para criar a chamada "nuvem oportunística". Surgem, assim, oportunidades de trabalhar na área de computação em nuvem com foco em nuvens privadas, para desenvolver soluções que facilitem a criação e gerenciamento de nuvens oportunísticas. A ideia é implementar uma maneira de tornar acessível e com menor custo esses recursos computacionais, uma vez que organizações acadêmicas e comerciais geralmente possuem computadores subutilizados, pois os usuários locais dificilmente utilizam todo o poder computacional presente na máquina, havendo um desperdício desses recursos (ROSALES; CASTRO; VILLAMIZAR, 2011).

O objetivo deste trabalho é desenvolver uma solução que possibilite a utilização dos recursos que estão ou possam ficar ociosos em máquinas físicas (MF) dos laboratórios de uma instituição, adicionando-as momentaneamente ao conjunto de recursos de uma nuvem privada, a fim de garantir um melhor aproveitamento global dos recursos da instituição e disponibilizar uma quantidade maior de processamento, memória e armazenamento para os usuários da nuvem privada. Em tal ambiente, o compartilhamento de recursos deve ser orquestrado de maneira a não prejudicar as atividades dos usuários locais, que tem prioridade sobre o uso de recursos.

A ideia é que instituições que possuam laboratórios de informática possam criar de maneira fácil e automatizada uma nuvem privada oportunística, a fim de ampliar o poder computacional de sua infraestrutura sem grandes investimentos em equipamentos, principalmente durante a execução de experimentos em suas pesquisas acadêmicas, oferecendo aos usuários da nuvem também recursos computacionais não dedicados.

1.2 Objetivo Geral

Implementar uma solução para o uso oportunístico de recursos computacionais ociosos, através da técnica de virtualização, para nuvens privadas institucionais.

1.3 Objetivos Específicos

- Elaboração de uma estratégia para a alocação de máquinas virtuais (MV) nos computadores dos laboratórios que estiverem com recursos ociosos.
- Definir solução para ligar e desligar os computadores dos laboratórios programaticamente.
- Desenvolvimento de um estudo de caso para validar a solução.

1.4 Organização do Trabalho

O restante do trabalho está organizado desta forma: a Seção 2 apresenta todos os conceitos relacionados a este trabalho. A Seção 3 apresenta os trabalhos relacionados que deram direcionamentos para esta pesquisa. A Seção 4 apresenta os procedimentos tomados para o desenvolvimento da proposta. A Seção 5 apresenta o estudo de caso realizado para validar a proposta e por fim, a Seção 6 apresenta as considerações finais encontradas com a realização desta pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Essa seção apresentada uma visão geral sobre os conceitos abordados nesse trabalho. Nas seções 2.1, 2.2, 2.3 e 2.4 são apresentados os conceitos relacionados à Virtualização, Computação em nuvem, Plataforma OpenNebula e Nuvem oportunística, respectivamente.

2.1 Virtualização

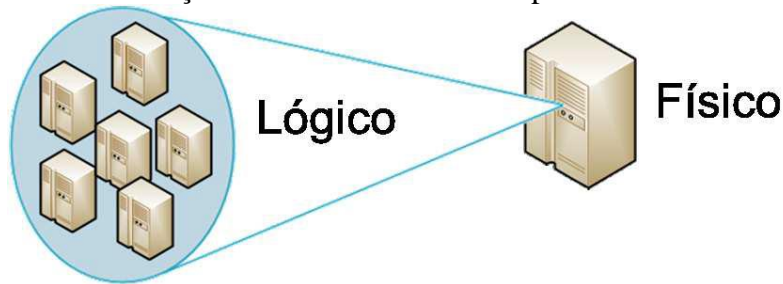
O conceito de virtualização surgiu com a IBM na década de 1960. Hoje, em ambientes virtualizados há um *software* sendo executado nas máquinas físicas, que é chamado monitor de máquina virtual (VMM, *Virtual Machine Monitor*) ou hipervisor (*hypervisor*) (SEO, 2009). Essa técnica foi desenvolvida para possibilitar um melhor aproveitamento do *hardware* dos grandes *mainframes*, um dos primeiros computadores criados.

Desde o início da virtualização, empresas como IBM, HP e Sun têm desenvolvido sistemas com suporte à virtualização. No entanto, o foco do mercado não era voltado para esses sistemas até o começo do ano 2000. A evolução dos sistemas de *hardware* aliada à crescente necessidade de se executar mais tarefas computacionais ao mesmo tempo com um

custo cada vez menor, impulsionou o crescimento da virtualização nos últimos anos (SEO, 2009).

A virtualização possibilita um melhor aproveitamento do *hardware*, de forma a proporcionar a instalação de diferentes sistemas dentro de uma mesma máquina física, a fim de compartilhar os recursos de *hardware* disponíveis da máquina física entre as máquinas virtuais criadas. Dessa forma, ambas podem executar ao mesmo tempo sem interferir em aplicações da outra, criando a ilusão de existir várias máquinas em uma infraestrutura, quando na verdade existem poucas máquinas físicas (SHIN; AKKAN, 2010).

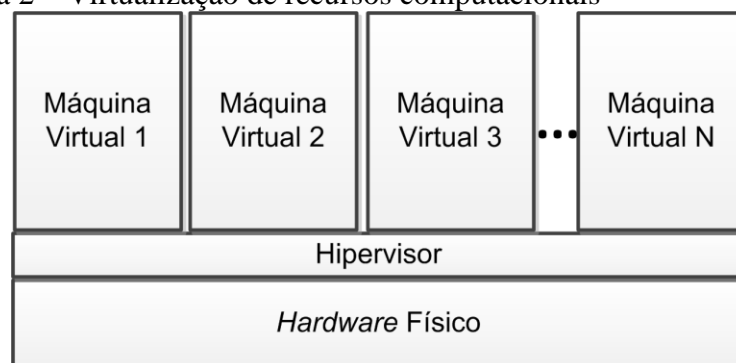
Figura 1 – Abstração de recursos físicos computacionais na virtualização



Fonte: Carvalho (2012)

A abstração de recursos físicos computacionais que permite existir várias máquinas virtuais executando em uma única máquina física simultaneamente é ilustrada na Figura 1, onde seis máquinas virtuais executam sobre uma máquina física, que se somam e totalizam sete máquinas em execução, quando na infraestrutura só existe uma máquina física. Essa camada de *software* que permite a abstração de recursos é o hipervisor e fornece às máquinas virtuais interfaces similares à interface real de *hardware* (CARVALHO, 2012). Na Figura 2, é possível ver internamente a organização das máquinas virtuais na máquina real.

Figura 2 – Virtualização de recursos computacionais



Fonte: Carvalho (2012)

De acordo com Rego (2012), existem as seguintes formas de virtualização:

- Virtualização de *desktops*: trata da configuração dos computadores pessoais em uma infraestrutura virtual, onde os sistemas operacionais e aplicações passam a ser executadas em centro de dados na forma de máquina virtual. Tais máquinas podem ser acessadas através da rede dispensando a necessidade de armazenamento local.
- Virtualização de armazenamento: uma abstração entre o conceito de armazenamento lógico e armazenamento físico, que permite que diversos discos físicos heterogêneos sejam vistos como um conjunto homogêneo de recursos de armazenamento, o que permite que administradores de sistemas tenham mais flexibilidade no gerenciamento de usuários e cotas de disco, uma vez que as complexidades de discos físicos são escondidas.
- Virtualização de redes: a forma de virtualização que permite a criação de ambientes de rede para grupos de usuários, apesar de compartilhar a mesma infraestrutura de rede física. Causando a impressão para os usuários finais, de uma rede própria, com recursos dedicados e políticas de segurança independentes. Um exemplo dessa forma de virtualização são as redes virtuais privadas (VPN, *Virtual Private Network*).
- Virtualização de servidores: tipo de virtualização mais comum. Permite que uma máquina física seja particionada para executar múltiplas máquinas virtuais. Havendo a necessidade de armazenamento local.

Neste trabalho, o modelo de virtualização de servidores será utilizado para implantar a ideia de particionamento de máquinas físicas, para que seja possível instalar várias máquinas virtuais.

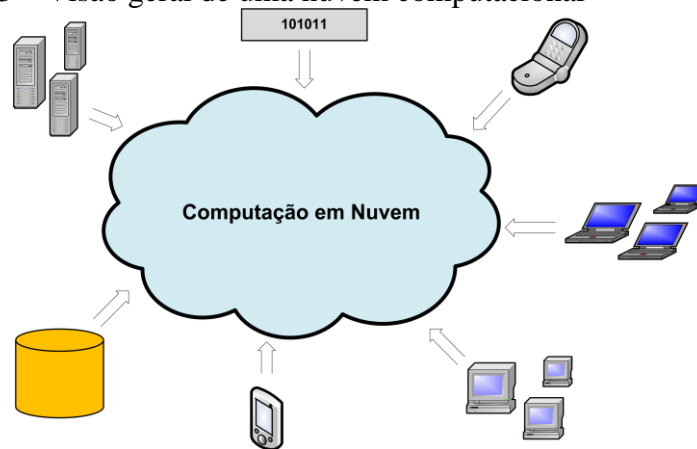
2.2 Computação em nuvem

Com a evolução das tecnologias e conseqüentemente a acessibilidade aos equipamentos, a sociedade moderna se tornou dependente dos serviços de Tecnologia da Informação (TI), gerando uma grande demanda por esses serviços o que proporcionou o surgimento de um novo paradigma de computação, a computação em nuvem (RIMAL; CHOI; LUMB, 2009).

De acordo com Sousa, Moreira e Machado (2009), a nuvem é uma metáfora para a *Internet* ou infraestrutura de comunicação entre os componentes arquiteturais, baseada em

uma abstração que oculta a complexidade de infraestrutura. Cada parte dessa infraestrutura é provida como um serviço, que normalmente são alocados em *datacenters*, que utilizam o *hardware* compartilhado para computação e armazenamento. Ou ainda, de acordo com Antonopoulos e Gillam (2010), esse termo faz referência ao *hardware*, *software* e aplicações entregues como serviço disponíveis através da *Internet*. A Figura 3 apresenta uma visão geral de uma nuvem computacional, com os vários equipamentos que podem compor essa infraestrutura diversificada, a abstração desta diversidade, como sendo homogênea.

Figura 3 – Visão geral de uma nuvem computacional



Fonte: Sousa, Moreira e Machado (2009)

Para o NIST (*National Institute of Standards and Technology*) (NIST, 2014) a computação em nuvem é um modelo para permitir o acesso conveniente, sob demanda de rede a um conjunto compartilhado de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados e liberados com um esforço mínimo de gerenciamento ou de interação do provedor de serviços.

Mell e Grance (2011) e NIST (2014) descrevem cinco características essenciais para o modelo de computação em nuvem, além de definir três modelos de serviços e quatro modelos de implantação, que são apresentados nas subseções a seguir.

2.2.1 Características essenciais

1) Serviço sob demanda

O usuário pode adquirir unilateralmente recurso computacional na medida em que necessite e sem precisar de interação humana com os provedores de cada serviço.

2) Amplo acesso

Os recursos computacionais são disponibilizados por meio da *Internet* e acessados através de mecanismos padronizados que permitem o uso por dispositivos móveis e estações de trabalho.

3) *Pooling* de recursos

Os recursos computacionais do provedor são utilizados para servir múltiplos usuários usando um modelo multi-inquilino (*multi-tenant*), com diferentes recursos físicos e virtuais, dinamicamente atribuídos e ajustados de acordo com a demanda dos usuários. Estes usuários não precisam ter conhecimento da localização física dos recursos computacionais, podendo somente especificar a localização em um nível mais alto de abstração, tais como o país, estado ou centro de dados.

4) Elasticidade rápida

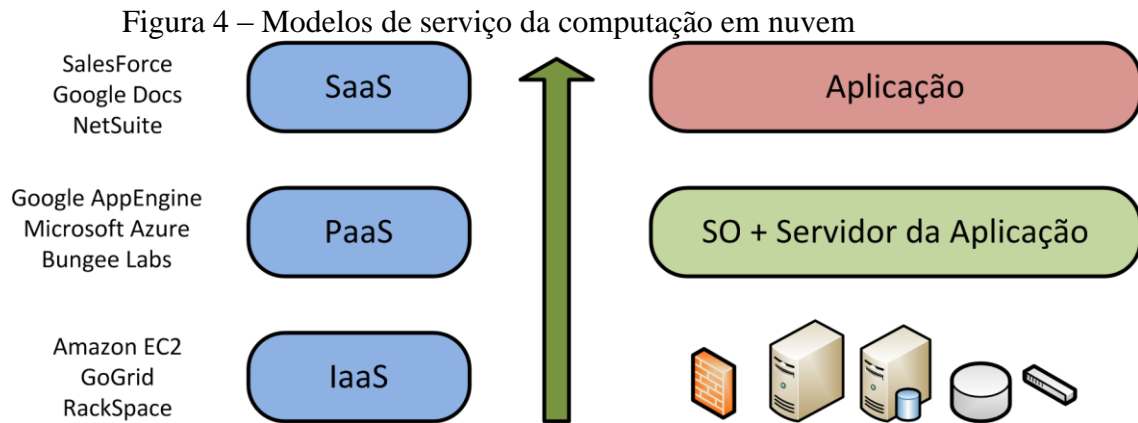
Recursos podem ser adquiridos de forma rápida e elástica, em alguns casos automaticamente, caso haja a necessidade de escalar com o aumento da demanda, e liberados, na retração dessa demanda. Para os usuários, os recursos disponíveis para uso parecem ser ilimitados e podem ser adquiridos em qualquer quantidade e a qualquer momento.

5) Medições de serviço

Os sistemas de gerenciamento utilizados para computação em nuvem controlam e monitoram automaticamente os recursos para cada tipo de serviço (armazenamento, processamento e largura de banda). O uso dos recursos deve ser transparente para o provedor do serviço, assim como, para o consumidor do serviço utilizado.

2.2.2 Modelos de serviço

Os recursos da nuvem são oferecidos através de abstrações de serviços, que atualmente são divididos em três categorias principais (ALANIS et al., 2013), conforme a Figura 4.



1) *Software* como um serviço (SaaS, *Software as a Service*)

Modelo no qual oferece *software* com propósito específico para os usuários finais, que acessam esses serviços por meio da *Internet* através de aplicações como navegador web. Nesse modelo o usuário não tem acesso a configurações aprofundadas do sistema. Hoje em dia, serviços oferecidos em nível de SaaS, como o Google Docs e Gmail, são muito difundidos e abstraem totalmente do usuário a infraestrutura da nuvem e detalhes técnicos (RIGHI, 2013).

2) Plataforma como um serviço (PaaS, *Platform as a Service*)

Nuvens PaaS fornecem um nível adicional de abstração sobre nuvens IaaS, proporcionando um ambiente em tempo real para a execução de códigos de aplicação e um conjunto de serviços de *software* adicional, como protocolo de comunicação, controle de acesso e persistência (CHAPMAN et al., 2011). Tal modelo fornece ambientes de desenvolvimento de aplicações, mas o consumidor desse serviço não gerencia nem controla a infraestrutura da nuvem, incluindo rede, servidores, sistemas operacionais ou armazenamento, mas tem controle sobre os aplicativos implementados e definições de configuração para o ambiente de hospedagem de aplicativos.

3) Infraestrutura como um serviço (IaaS, *Infrastructure as a Service*)

O objetivo deste modelo é tornar mais fácil e acessível o fornecimento de recursos, tais como servidores, rede, armazenamento e outros recursos de computação fundamentais para construir um ambiente de computação sob demanda, que pode incluir sistemas operacionais e aplicativos. Em geral, o usuário não administra ou controla a infraestrutura da nuvem, mas tem controle sobre os sistemas operacionais, armazenamento e aplicativos

implantados e, possivelmente seleciona componentes de controle de rede, tais como *firewalls*.

2.2.3 Modelos de implantação

1) Nuvem privada (*Private Cloud*)

A infraestrutura de nuvem é provisionada para uso exclusivo por uma única organização que agrupa vários consumidores. Podendo ser local ou remota e administrada pela própria empresa ou por terceiros.

2) Nuvem comunidade (*Community Cloud*)

Fornece uma infraestrutura compartilhada por uma comunidade de organizações com interesses em comum.

3) Nuvem pública (*Public Cloud*)

A infraestrutura de nuvem é disponibilizada para uso aberto do público em geral. A nuvem é disponibilizada publicamente através do modelo pague pelo uso (*pay-per-use*), tipicamente são oferecidos por companhias que possuem grandes capacidades de armazenamento e processamento (VERDI et al., 2010).

4) Nuvem híbrida (*Hybrid Cloud*)

A infraestrutura é composta por duas ou mais nuvens distintas (privada, comunidade, ou pública) que permanecem entidades únicas, mas conectadas através de tecnologia proprietária ou padronizadas.

Este trabalho utilizará o modelo de serviço de nuvem IaaS, que permite um maior controle sobre a infraestrutura da nuvem, tal característica será usada para possibilitar o objetivo do trabalho proposto, que visa facilitar a instalação de nuvem privada oportunística em instituições que apresentem computadores subutilizados. Para isso, será adotado o modelo de implantação de nuvem privada.

2.3 Plataforma OpenNebula

O OpenNebula é um *software* de código livre desenvolvido para gerenciar a complexidade e heterogeneidade de grandes infraestruturas distribuídas, e pode ser usado para a criação de nuvens públicas, privadas e híbridas. É um conjunto de ferramentas que tem como objetivo fornecer uma camada de gerenciamento flexível, aberta, escalável e abrangente

para gerenciar o funcionamento dos grandes centros de dados e transformar a infraestrutura existente em uma nuvem IaaS (WEN et al., 2012).

Este trabalho utilizará a plataforma de gerenciamento de nuvem OpenNebula, pois segundo Wen et al. (2012), o OpenNebula é adequado tanto para institutos de pesquisas, quanto para universidades e empresas, especialmente grandes centro de dados que querem construir uma nuvem aberta, flexível e escalável para suportar suas pesquisas e que estão interessadas na tecnologia de máquinas virtuais também é ideal para os usuários que querem executar rapidamente na nuvem um certo número de MVs.

Esta ferramenta fará o gerenciamento da nuvem privada a ser criada durante o projeto, pois ela é adequada para lidar com grandes quantidades de dados e alavancar as infraestruturas de TI existentes, protegendo investimentos e evitando dependência de provedores (OPENNEBULA, 2014). Segundo Sempolinski e Thain (2010), nessa ferramenta, os nós de computação não precisam de uma grande quantidade de recursos de disco rígido. Além de permitir a mudança em praticamente todos os seus componentes, incluindo o sistema de arquivos e o *front-end*.

2.4 Nuvem oportunística

A computação oportunística é um paradigma de computação distribuída que se baseia em oferecer serviços de computação para atender as necessidades de aplicações, aproveitando recursos de computação disponíveis e acessíveis no ambiente. Uma infraestrutura computacional oportunística fornece sob demanda serviços computacionais fundamentais, sistemas operacionais e aplicações (BARRERA; ROSERO; CANO, 2012). Além disso, explora todos os recursos disponíveis em um ambiente de maneira oportunística para fornecer uma plataforma para a execução de tarefas de computação distribuída (CONTI; KUMAR, 2010).

Um dos primeiros trabalhos na área de computação oportunística foi o sistema Condor, desenvolvido na Universidade de Wisconsin-Madison em 1988. Este projeto traz dois modelos de execução: computação intensiva e computação oportunística. O modelo de computação intensiva é adequado para usuários que necessitam de uma grande quantidade de computação durante um grande espaço de tempo. No modelo oportunístico os recursos são utilizados quando se apresentam ociosos, sendo que a prioridade é dos donos dos recursos (LITZKOW; LIVNY; MUTKA, 1988).

O projeto UnaCloud é um sucessor do projeto UnaGrid, foi desenvolvido para integrar os benefícios dos sistemas de computação em grade, computação voluntária e o

modelo IaaS da computação em nuvem, com o objetivo de proporcionar acesso sob demanda aos recursos computacionais personalizados (processamento, armazenamento e rede) usando uma infraestrutura oportunística capaz de suportar um modelo IaaS experimental a um menor custo e oferecendo os mesmos benefícios e funcionalidades disponíveis em modelos IaaS convencionais usando uma base de melhor esforço (DIAZ et al., 2013).

Os conceitos da UnaCloud foram utilizados, uma vez que esta é a primeira implementação funcional do conceito de computação oportunística em nuvem (OSORIO; CASTRO; BRASILEIRO, 2012).

3 TRABALHOS RELACIONADOS

Osorio, Castro e Brasileiro (2012) usam o conceito de computação oportunística em uma infraestrutura de computação em nuvem. Uma solução atraente para os usuários dos serviços que poderiam acessar grandes recursos computacionais com a facilidade da computação em nuvem a custos baixos, e para os gerentes de serviços, que poderiam oferecer uma solução de computação de alto desempenho, sem fazer grandes investimentos em equipamentos e manutenção. Essa ideia tem sido desenvolvida na Universidade de *Los Andes*, na Colômbia, através de um projeto *open source* denominado UnaCloud.

O UnaCloud foi criado para implantar, gerenciar e entregar um modelo IaaS oportunística com base em recursos pré-existentes, de computação não dedicada, distribuídos e heterogêneos que fazem parte de diferentes domínios administrativos. Apesar de ser bem sucedida, essa solução apresenta alguns requisitos para sua instalação, como sistema operacional *Windows Server 2003* ou *Windows 7*, e o hipervisor *VMware Workstation 6* ou superior, *software* que precisam de licença e não tem seu código livre. Para a implementação da ideia, os autores usaram a linguagem de programação Java.

Propomos desenvolver uma solução própria, para a adição automatizada e oportunística dos recursos à nuvem privada, baseada no conceito de UnaCloud, mas completamente *open source*, desenvolvida em *Shell Script* e *Python*, e utilizando o hipervisor KVM e a ferramenta OpenNebula para o gerenciamento da nuvem, que busque uma otimização no uso de recursos ociosos presentes em infraestruturas pré-existentes.

No trabalho de Alanis et al. (2013), os autores propõem uma solução de nuvem privada oportunística que permite o uso dos recursos ociosos de uma infraestrutura pré-existente para a execução de aplicações que não exigem uma alta qualidade de serviço (QoS, *Quality of Service*), as chamadas aplicações BoT (*Bag-of-Tasks*, Saco de Tarefas). BoT são

aplicações paralelas cujas tarefas são independentes entre si. Os autores desenvolveram uma plataforma tecnológica capaz de fornecer suporte a aplicações BoT, adequada a ambientes oportunistas e heterogêneos, de uma forma amigável, dinâmica, flexível e a um baixo custo.

Para criar a nuvem privada, eles utilizaram a plataforma Eucalyptus (SEMPOLINSKI e THAIN, 2010), e tiveram que modificar o componente *Node Controller* (NC), responsável por controlar atividades das máquinas virtuais, pois este utilizava originalmente o hipervisor Xen¹, que impõe alguns requisitos de sistema para a sua implantação em um ambiente heterogêneo. Eles então reimplementaram o NC em Java e decidiram usar o hipervisor VirtualBox². Apesar de ser bem sucedido, esse trabalho é voltado a aplicações BoT em um ambiente de nuvem oportunística que busca utilizar os recursos computacionais ociosos fora do horário de trabalho dos usuários (e. g., depois do fim do expediente).

Nossa solução busca a utilização de recursos ociosos inclusive em horários de trabalho, contando com um sistema de monitoramento da utilização dos recursos das máquinas, a fim de contar com essas máquinas enquanto os usuários locais estão usando. Para isso, é preciso lidar com as flutuações das cargas de trabalho e possíveis falhas e desligamentos não programados da máquina.

Marshall, Keahey e Freeman (2011) propõem uma solução para um melhor uso de infraestrutura de nuvens, que combina alocação sob demanda de recursos e provisionamento oportunístico dos ciclos ociosos dos nós da nuvem, para que estes possam ser utilizados por outros processos por meio da implantação de MVs, pois os provedores de nuvem tendem a manter uma infraestrutura de nuvem dedicada capaz de atender as requisições de uma grande quantidade de usuários, mas pode haver momentos em que o acesso a nuvem apresente queda deixando alguns nós ociosos.

Os autores investigam soluções de uso desses nós ociosos em provedores de nuvem, para isso utilizaram o *design* original do Condor, e a plataforma de nuvem Nimbus (SEMPOLINSKI e THAIN, 2010), para a implantação de MVs em nós da nuvem que não apresentem uso para o processamento de carga de trabalho. Assim, foi criada uma extensão para o Nimbus em Java, que contém a maior parte do código de implantação de MVs.

O trabalho proposto busca um melhor uso de recursos presentes em ambientes institucionais que apresentem computadores subutilizados, através da criação de uma nuvem privada oportunística que possa usar de forma paralela ao usuário da máquina física os

¹ <http://www.cl.cam.ac.uk/research/srg/netos/xen/>

² <https://www.virtualbox.org/>

recursos que seriam desperdiçados por usuários que não consomem toda a capacidade computacional presente na máquina, além de usar outras soluções e *software* já citados.

4 PROPOSTA

O trabalho visa desenvolver uma solução para possibilitar o uso de recursos computacionais presentes em máquinas físicas subutilizadas de uma instituição.

O código desenvolvido para a implantação da solução proposta está disponível no repositório GitHub³.

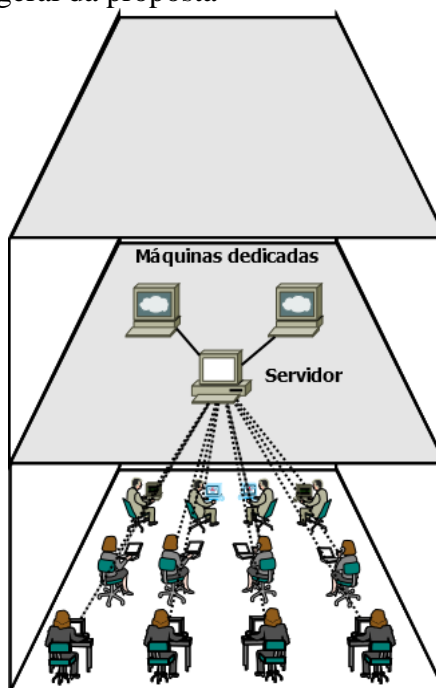
Uma visão geral da proposta deste trabalho está representada na Figura 5, onde se pode observar uma nuvem privada com apenas duas máquinas dedicadas e que reúne em si os recursos que se apresentam ociosos nas demais máquinas do laboratório, aumentando assim a capacidade computacional da nuvem e eliminando a necessidade de investimentos adicionais na compra de novas máquinas.

A integração e utilização dessas máquinas pela nuvem ocorrem simultaneamente ao uso dos usuários finais, sem prejudicá-los em suas atividades cotidianas, ficando transparente para os mesmos.

A reunião dos recursos computacionais ociosos de cada máquina é compartilhada na nuvem formando uma agregação de recursos que juntos tornam a nuvem capaz de executar aplicações mais complexas e executar uma maior quantidade de máquinas virtuais.

³ <https://github.com/Renato2012/NuvemOportunistica.git>

Figura 5 – Visão geral da proposta



Fonte: elaborada pelo autor.

Durante o desenvolvimento deste trabalho, os passos descritos nas subseções seguintes foram seguidos.

4.1 Instalação da plataforma OpenNebula

A instalação se deu de forma autocontida usando o OpenNebula versão 4.8.0, numa máquina física com sistema operacional Ubuntu 14.04 LTS 64 bits do Laboratório de Redes de Computadores, e tal máquina é responsável por gerenciar as máquinas virtuais criadas com o hipervisor KVM presentes nas máquinas físicas, de acordo com informações obtidas do controlador. Para utilizar o KVM é necessário instalar e configurar o hipervisor nas máquinas que farão parte da nuvem privada a ser criada.

O controlador é uma máquina física que executa *scripts* para coletar informações sobre o estado de memória RAM, CPU e disco das máquinas físicas que contenham máquinas virtuais, de modo a tomar decisões com base nas informações coletadas. Em especial, algumas das decisões que precisam ser consideradas são as de migrar MVs, principalmente quando o usuário da MF iniciar o processo de desligamento da mesma.

A instalação do hipervisor foi feita de maneira automatizada usando *script* desenvolvido na linguagem *Shell Script*, que pode ser executado para instalar o hipervisor nos clientes de forma remota e automática, como no servidor de forma iterativa com passagem de parâmetros que serão solicitados ao administrador em tempo de execução.

4.2 Estratégia automatizada de integração de MFs à nuvem

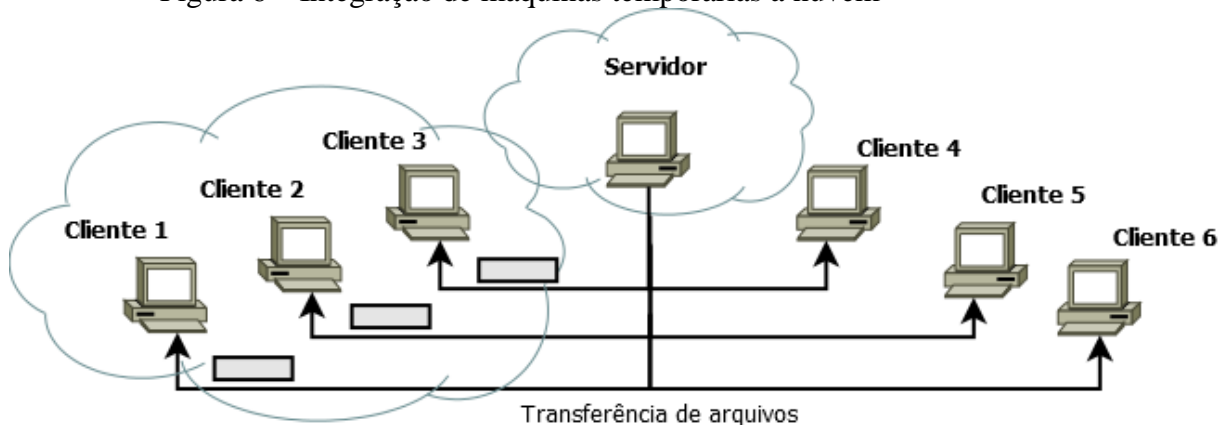
Foi desenvolvida uma solução para prover a integração de maneira automatizada de máquinas físicas com a nuvem privada criada. A solução foi implementada usando as linguagens *Shell Script* e *Python* para interligar alguns computadores à nuvem privada, para que esta tenha um aumento em seu poder computacional sem a necessidade de adquirir novos equipamentos.

No servidor, foi necessário coletar e salvar os endereços de rede das máquinas que fariam parte da nuvem, como também os endereços físicos das mesmas. Faz-se necessário ter a lista de endereços IPs das máquinas do laboratório no servidor para tomar como entrada para a execução do *script* responsável por transferir de forma automatizada e remota uma pasta com o conjunto de *scripts* de configuração para todas as máquinas físicas que seriam integradas à nuvem. A transmissão desses arquivos e as conexões necessárias se deram de forma segura usando o protocolo SCP (*Secure Copy*) e SSH (*Secure Shell*) respectivamente.

Houve ainda a necessidade de se criar um *script* servidor com *thread* e um cliente usando *Python* para permitir uma interação entre a máquina temporária e a máquina servidora, que aguarda conexões dos clientes, cujos clientes foram transferidos para as máquinas temporárias junto com os *scripts* de configuração.

Na máquina servidora, foi implementado um *script* que transfere os arquivos de configuração necessários para a adição das máquinas temporárias a nuvem, e após concluir a transferência, estabelece uma conexão SSH e executa de forma remota o instalador que desencadeia toda a configuração necessária do cliente na máquina, e repete esse ciclo para cada máquina que se pretende adicionar à nuvem. A Figura 6 apresenta uma visão geral da solução para a integração de máquinas temporárias à nuvem de forma automatizada.

Figura 6 – Integração de máquinas temporárias a nuvem



Fonte: elaborada pelo autor.

Uma vez configurada, a máquina remota se torna um membro da nuvem privada, e tem seus recursos monitorados pelo OpenNebula, e ficam à disposição da nuvem para executar máquinas virtuais.

Pode-se observar que existem dois módulos na solução: Servidor e Cliente, que são descritos nas próximas seções.

4.2.1 Servidor

No servidor, o próprio OpenNebula se encarrega de monitorar o estado de CPU e memória das máquinas adicionadas a nuvem, gerando arquivos de *log* contendo tais informações.

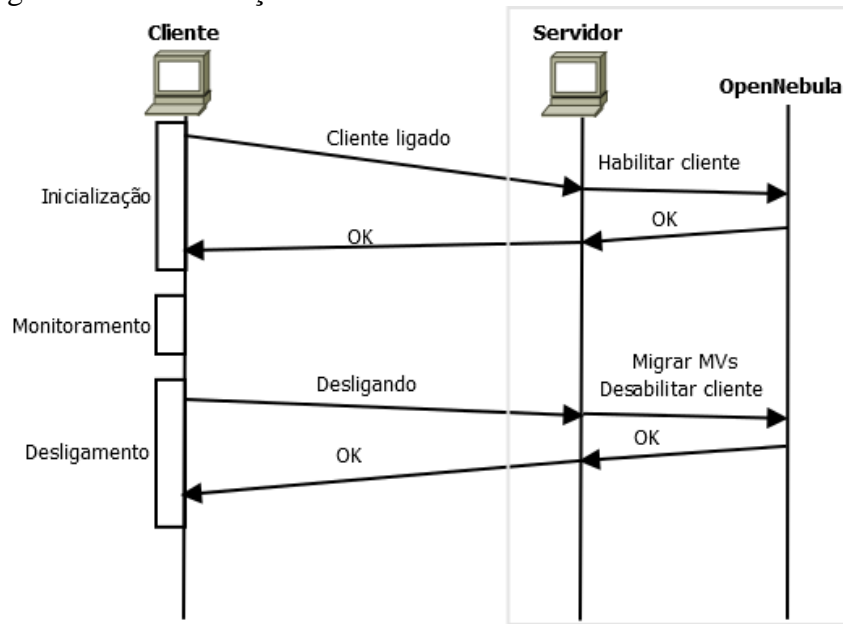
A técnica de ativação remota (WoL, *Wake on LAN*) foi empregada neste trabalho para prover a inicialização automatizada das máquinas físicas pertencentes à nuvem. WoL permite que um computador com rede seja ligado através do envio de um pacote de dados especiais (*Magic Packet*) para a máquina de destino. Quando o pacote é recebido no destino, o dispositivo de rede (NIC, *Network Interface Controller*) inicializa a máquina, mesmo que esta esteja desligada, pois o NIC permanece ativo na máquina verificando a rede (UBUNTU DOCUMENTATION, 2014).

O módulo servidor desenvolvido é responsável por verificar continuamente a quantidade de recursos disponíveis na nuvem para atender à quantidade de máquinas virtuais instanciadas, nesse processo deve ligar automaticamente máquinas temporárias quando não houver recursos suficientes para a execução de alguma máquina virtual instanciada.

O módulo servidor aguarda as conexões dos clientes, e ao recebê-las, chama o serviço OpenNebula para adicionar as máquinas na nuvem essas conexões ficam ativas na espera por novas mensagens.

Nos casos de desligamento do cliente, a mensagem recebida informa tal evento e o módulo deve verificar se o cliente possui máquinas virtuais em execução para migrá-las em tempo real para outras máquinas que possuam recursos suficientes para receber as instâncias. A Figura 7 apresenta as mensagens trocadas entre o cliente e o servidor nas fases de inicialização e desligamento de uma máquina da nuvem, bem como as ações que o módulo impõe ao OpenNebula.

Figura 7 – Comunicação entre módulo cliente e módulo servidor



Fonte: elaborada pelo autor.

Esse módulo, faz o intermédio do cliente com o OpenNebula e baseado nas mensagens que recebe do cliente informa o que o OpenNebula deve fazer.

Podem ocorrer casos de desligamento simultâneo das máquinas temporárias, o que impediria o módulo servidor de ligar as máquinas físicas no momento necessário, já que estas se encontram em processo de desligamento e impede a técnica WoL de liga-las assim não haveria hospedeiro disponível para as MVs o que causaria falha no sistema e a perda dessas MVs. Para evitar esses casos, é necessário reservar algumas máquinas para atuarem de forma dedicada a nuvem.

4.2.2 Cliente

Nas máquinas físicas que compõem a nuvem, foi implantado um *script daemon* para que, na inicialização, reinicialização e desligamento do sistema, o módulo cliente em *Python* seja executado, e assim comunique-se com o servidor para que este tome as ações necessárias com as máquinas virtuais em execução, se houver alguma. O *daemon* criado foi baseado no modelo que as versões do Ubuntu Debian disponibilizam, tal modelo está localizado no diretório `/etc/init.d/skeleton`.

O módulo cliente é controlado pelo *daemon* e responsável pelas tarefas listadas a seguir:

- 1) Inicialização da máquina

Na inicialização do sistema operacional, junto a outros serviços, o cliente é ativado e fica verificando a disponibilidade do serviço de rede, após a disposição desse serviço, o cliente monta o sistema de arquivo de rede (NFS, *Network File System*) para que os recursos da máquina possam ser compartilhados na nuvem e logo em seguida se comunica com o módulo servidor para que o mesmo o habilite na nuvem, para que o OpenNebula possa monitorá-lo.

2) Durante a execução

Após ser habilitada e ter seus recursos compartilhados na nuvem a máquina fica pronta para receber instâncias de MVs, dependendo apenas da quantidade de recursos ociosos na máquina.

Durante o período de execução da máquina o módulo cliente verifica continuamente as pretensões do usuário para saber o momento exato em que o mesmo inicia o processo de desligamento da máquina.

Nessa fase não há troca de mensagens entre os módulos cliente e servidor, existindo apenas as trocas de mensagens das sondas de monitoramento do OpenNebula, que por padrão se dão através de um agente que utiliza conexões com o protocolo UDP.

3) Desligamento da máquina

Ao perceber que o usuário iniciou o processo de desligamento da máquina o módulo cliente envia uma mensagem para o servidor informando o evento e bloqueia o processo de desligamento na espera por resposta do servidor, que, após processar a mensagem recebida e tomar as ações necessárias, irá retornar a mensagem de liberação para o cliente que ao recebê-la desbloqueia o processo e prossegue com o desligamento normal do sistema.

O bloqueio do processo de desligamento do sistema se fez necessário para os casos de desligamento em que haja máquinas virtuais em execução para tornar possível a migração antes do desligamento da máquina física para isso foi necessário intervir no processo de desligamento normal do sistema, de forma a causar uma espera para que as MVs possam completar a migração, só então é que o desligamento deve prosseguir normalmente.

Para interferir no processo de desligamento padrão do sistema operacional das máquinas temporárias da nuvem, foi necessário implementar um *script* que tem sua execução iniciada no momento desse evento com a função de aguardar uma resposta do módulo servidor para prosseguir com o desligamento do sistema. Esse *script* deve ser chamado

exatamente após a linha “*emits desktop-shutdown*” que é responsável por divulgar a mensagem de desligamento do sistema para os processos, no qual está no arquivo `/etc/init/lightdm.conf` do sistema *Upstart*. Tal sistema é um sistema de inicialização moderno e totalmente baseado em eventos e vem substituindo o antigo *System V*, originário do Unix (WIKI DEBIAN, 2014).

Como o *System V* atualmente ainda atua em paralelo nas distribuições Debian, ainda foi necessário implantar o *script* de interferência em seu diretório o `/etc/init.d/` e criar os *links* apontando para os diretórios `/etc/rc0.d` e `/etc/rc6.d`, responsáveis pelo o desligamento e reinicialização do sistema respectivamente.

4.3 Automatizar o desligamento/inicialização das MFs

O desligamento automatizado das máquinas dos laboratórios foi desenvolvido para auxiliar os administradores na gerência das máquinas, para que estes apresentem um maior controle sobre as máquinas e evitem deixá-las ligadas, quando não há necessidade.

Na inicialização das máquinas, além da forma padrão (por usuários), o administrador tem autonomia para ligar, desligar ou reiniciar uma ou mais máquinas remotamente.

Além da inicialização de forma automática através do módulo servidor, que executa de maneira contínua na máquina servidora, para monitorar a quantidade de recursos computacionais disponíveis na nuvem e tomar decisões quando houver máquinas virtuais pendentes por falta destes recursos, iniciando o processo de *boot* nas máquinas temporárias desligadas para suprir a falta de recursos.

5 ESTUDO DE CASO

Nesta seção são apresentados os cenários e seus respectivos resultados da execução da proposta deste trabalho.

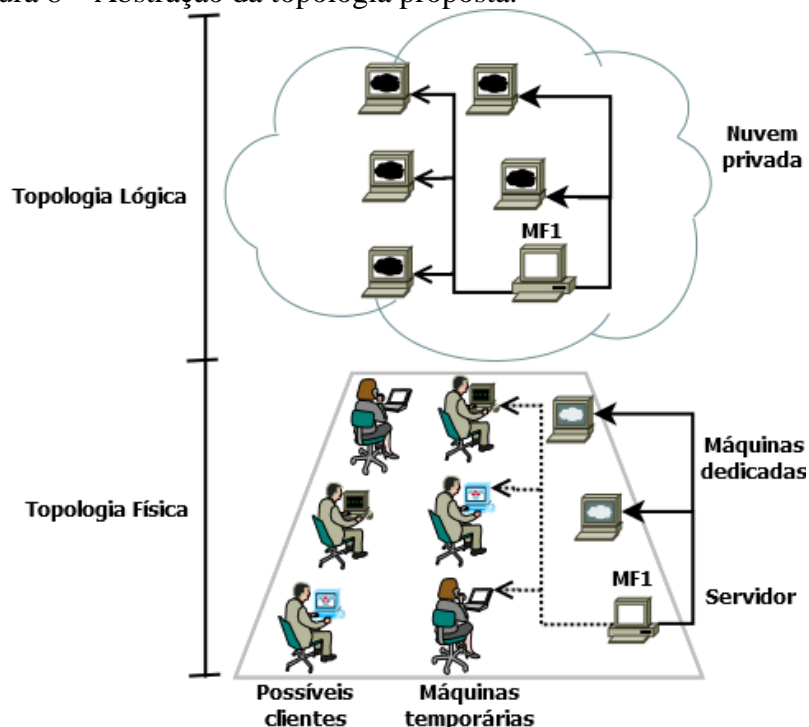
Foi realizado um estudo de caso no Campus Quixadá da Universidade Federal do Ceará, onde os conceitos de computação em nuvem e virtualização são utilizados para estender, de forma oportunística, os recursos da nuvem privada criada para o desenvolvimento da proposta no Laboratório de Redes de Computadores.

5.1 Cenário

Para os experimentos, foram utilizadas seis máquinas físicas do Laboratório de Redes de Computadores da UFC Campus Quixadá. Sendo a MF1 o controlador com duas máquinas dedicadas à nuvem e três máquinas integradas de forma oportunística. A Figura 8 é uma representação da topologia adotada.

Para os clientes locais fica transparente a utilização da máquina pela nuvem, que agrega todas as máquinas temporárias junto às dedicadas, formando um aglomerado de máquinas que juntas tornam a nuvem mais robusta.

Figura 8 – Abstração da topologia proposta.



Fonte: elaborada pelo autor.

5.2 Experimentos

Foram realizados seis experimentos para calcular as métricas abaixo:

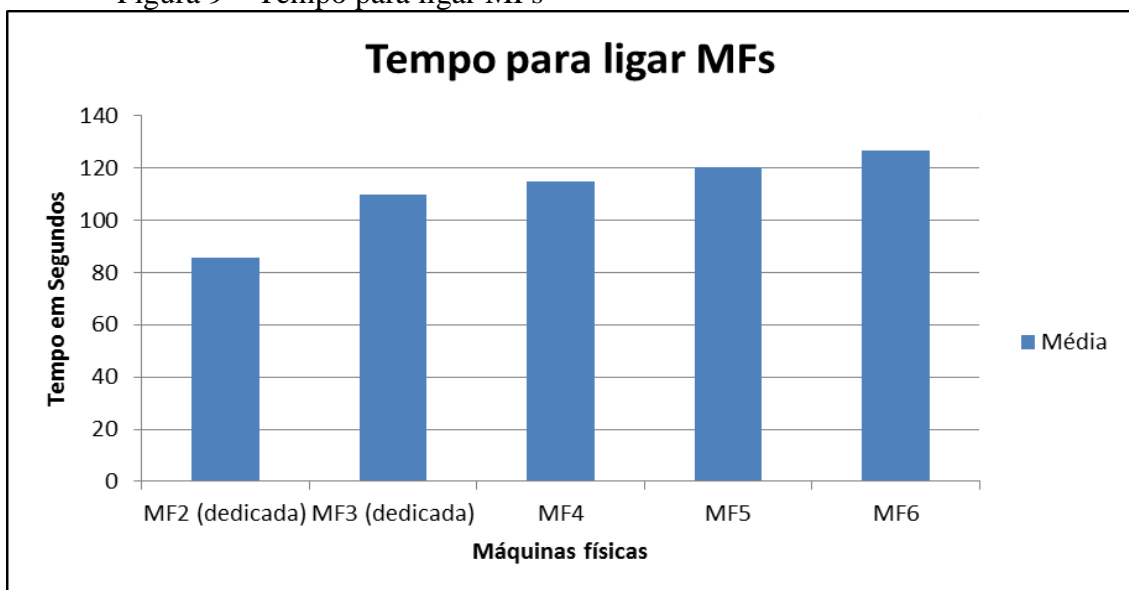
- Tempo para ligar as máquinas
- Tempo para desligar as máquinas
- Tempo para desligar MFs com MVs em execução

Com a obtenção dessas métricas, é possível analisar o comportamento da solução quanto ao tempo necessário para que as máquinas temporárias sejam habilitadas na nuvem quando ligadas, sejam desabilitadas da nuvem quando desligadas nos casos sem MVs e com máquinas virtuais em execução.

O tempo para a ativação remota das máquinas apresentado é o tempo total necessário para que as máquinas temporárias sejam iniciadas e conectadas à nuvem privada. Vale ressaltar que as máquinas físicas usadas nos experimentos tem o disco particionado com dois sistemas operacionais diferentes, o sistema Windows e o Ubuntu, sendo este configurado com *boot* padrão em todas as máquinas, no qual há uma espera de 10 segundos para uma possível intervenção do usuário final na escolha do sistema.

Os resultados obtidos seguem nas Figuras abaixo, respectivamente:

Figura 9 – Tempo para ligar MFs

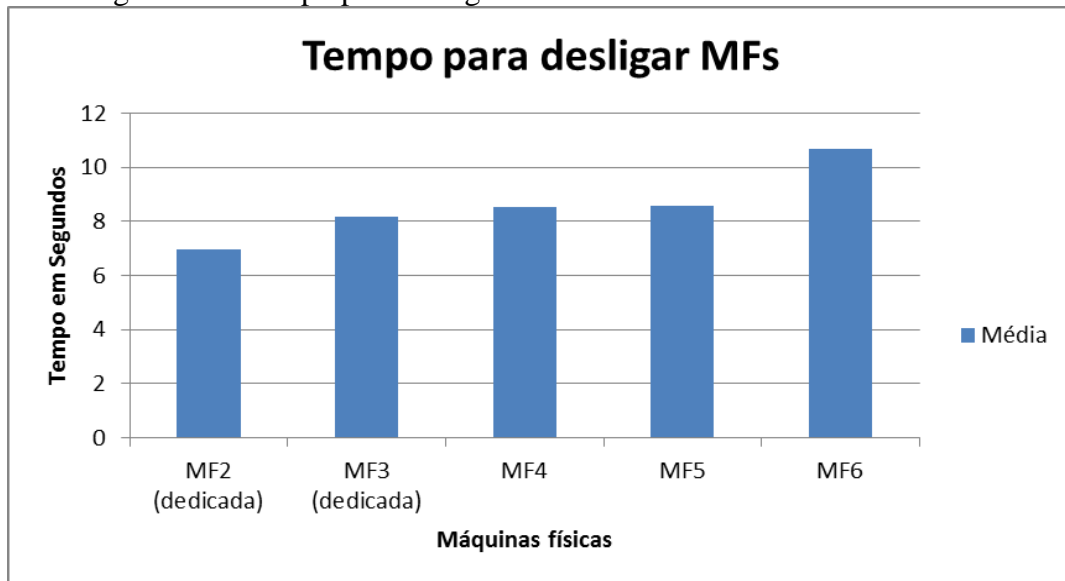


Fonte: elaborada pelo autor.

Na Figura 9 é possível observar que na medida em que aumenta a quantidade de máquinas físicas a média de tempo para ligá-las também aumenta, pois para cada máquina a ativação remota se dá em uma iteração diferente e crescente do *script* de ativação remota, assim quanto maior for o número da máquina maior será a espera para que a iteração chegue à mesma e a ativação remota aconteça.

A Figura 10 representa os resultados dos experimentos quanto ao tempo de desligamento das máquinas temporárias.

Figura 10 – Tempo para desligar MFs

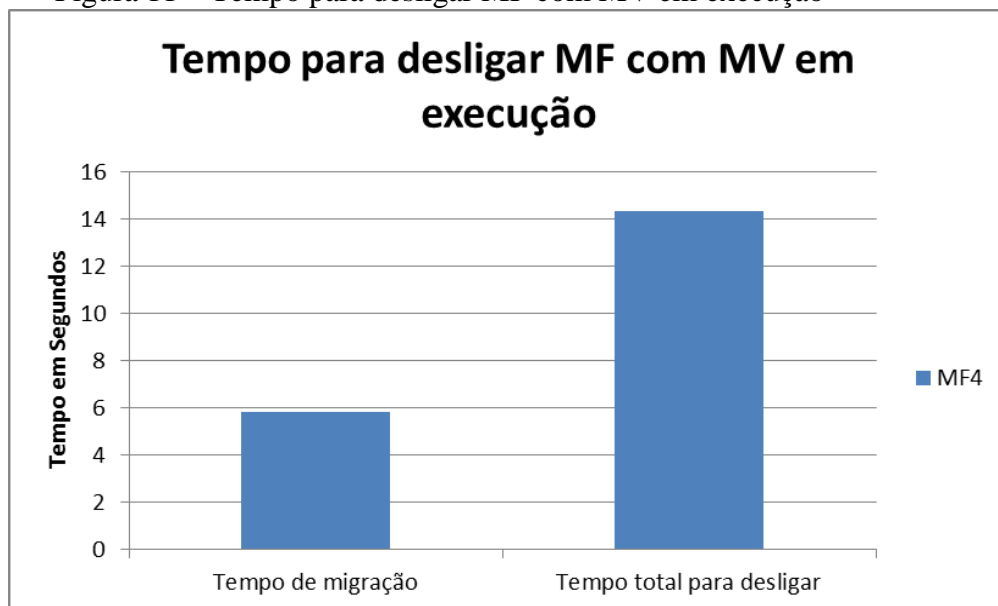


Fonte: elaborada pelo autor.

Na análise dos resultados mostrados na Figura 10, pode-se notar que a média de tempo também acontece de forma crescente, o que é explicável seguindo o mesmo raciocínio sobre as iterações do *script* de ativação e desligamento remoto.

Para os experimentos de cálculo do tempo de migração de máquinas virtuais, foram criadas MVs com 1 GB de memória RAM, cujos valores dos experimentos estão representados na Figura 11.

Figura 11 – Tempo para desligar MF com MV em execução



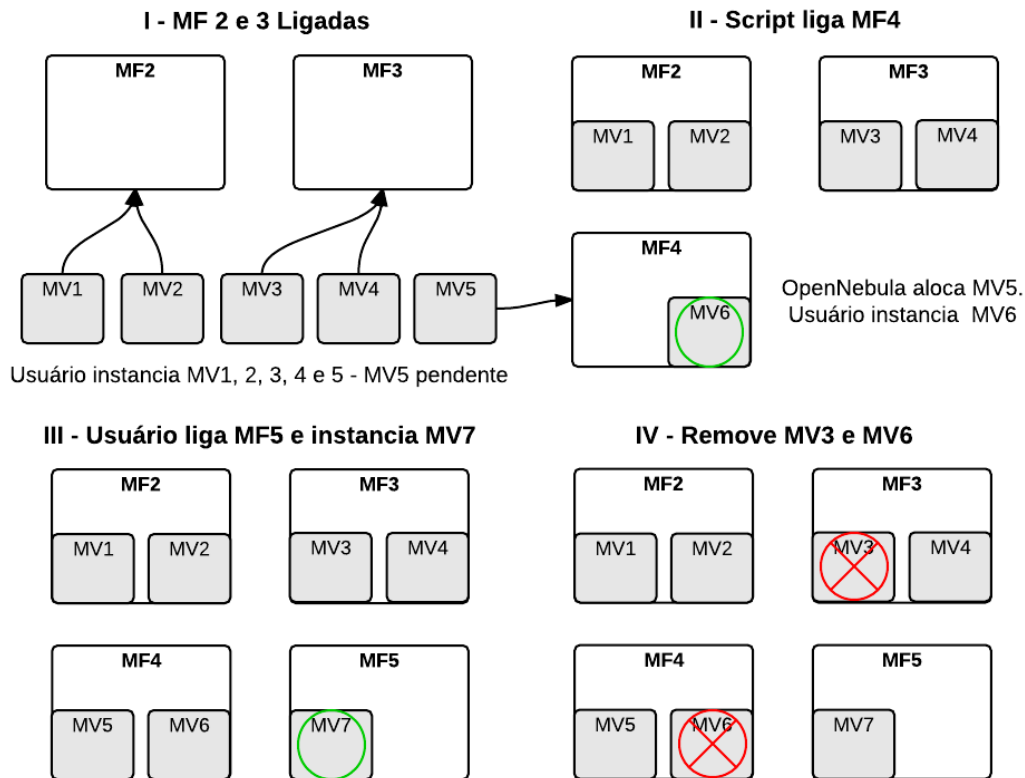
Fonte: elaborada pelo autor.

Na Figura 11, é mostrada a média dos resultados obtidos a partir do desligamento da MF4 com uma MV em execução. Por está com MV em execução no momento do desligamento, o tempo para que a máquina física 4 desligue ultrapassou o tempo de desligamento quando não há MV executando, pois além do tempo de desligamento padrão foi acrescentado o tempo para que a MV seja migrada para outra MF.

Um cenário foi construído com várias possibilidades que poderiam ocorrer no cotidiano de laboratórios de informática. A Figura 12 retrata bem essas ocorrências.

- Parte I: nesse cenário se podem observar duas máquinas físicas ligadas prontas pra receber máquinas virtuais. O usuário da nuvem precisa de cinco MVs e as instancia, mas não há espaço suficiente para atender às requisições do usuário, pois as máquinas físicas 2 e 3 cada uma só comportam duas MVs, com as configurações solicitadas. Assim a última MV, no caso a MV5, ficará no estado pendente, portanto, o módulo servidor verifica a falta de recursos na nuvem e aloca uma nova máquina pertencente à nuvem que esteja desligada.
- Parte II: o módulo servidor liga a máquina física 4 e o OpenNebula se encarrega de alocar a MV5 que estava pendente à nuvem. Após isso, o usuário da nuvem manda criar uma nova máquina virtual (MV6), que é alocada à MF4, uma vez que ela contém recursos necessários para receber a MV.
- Parte III: o usuário decide ligar a MF5, enquanto que o usuário da nuvem instancia uma nova máquina virtual, MV7 que é alocada à máquina MF5 que acabou de ser ligada.
- Parte IV: o usuário da nuvem decide remover as máquinas virtuais MV3 e MV6, ocasionando liberação parcial de recursos nas máquinas físicas MF3 e MF4, que estariam aptas a receber novas instâncias já que tiveram recursos liberados. A MF5 também suportaria mais uma instância.

Figura 12 – Cenários comuns.

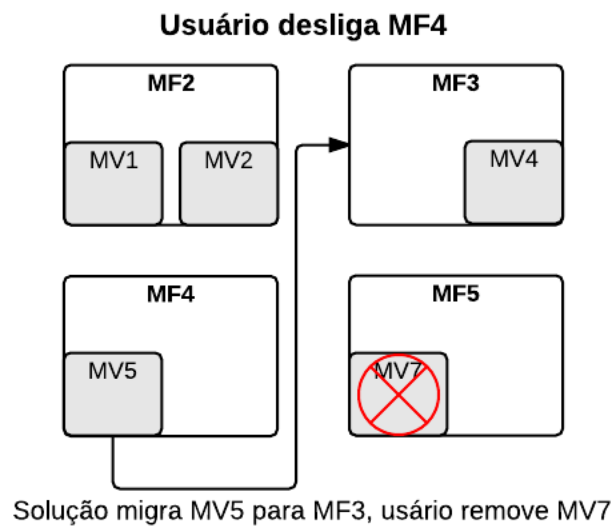


Fonte: elaborada pelo autor.

O reinício e desligamento de *desktops* com máquinas virtuais em execução pode acontecer. Nesse caso, os usuários não precisam se preocupar com MVs que, por ventura, estejam sendo executadas, pois a solução proposta lida com esse cenário.

A Figura 13 apresenta essa situação, a máquina física 4 se encontra ligada com uma máquina virtual em execução, a MV5. O usuário do laboratório inicia o processo de desligamento nesta máquina, então a solução deve migrar a MV5 para a máquina física 3 ou máquina física 5, pois ambas possuem recursos suficientes. Assumindo que a MV em questão foi movida para a MF3 a nuvem contaria apenas com a parcialidade de recursos da MF5 disponíveis, já que a MF2 e MF3 estariam em sua capacidade máxima. O usuário da nuvem decide remover a MV7 que se encontra em execução na MF5 liberando os recursos dessa máquina que estaria apta a receber novas instâncias.

Figura 13 – Migração de MVs.



Fonte: elaborada pelo autor.

A migração das possíveis MVs que estejam em execução no momento do desligamento da máquina física deve ser automatizada e com certa rapidez para máquinas físicas que tenham recursos suficientes disponíveis.

6 CONSIDERAÇÕES FINAIS

Após concluirmos esse estudo, pode-se notar que a computação em nuvem junto da virtualização está se tornando tendência de mercado, e juntas são tecnologias que oferecem vastos campos de pesquisa.

A computação em nuvem dispõe de inúmeras vantagens, motivos aos quais está levando diversas organizações a adotá-la dentro de seu setor de TI. Pesquisas nessa área são constantes e mesmo assim, essa tecnologia tem muito mais benefícios a oferecer, que juntos estão auxiliando empresas como as *startups* a surgirem no mercado, mesmo sem grandes investimentos em aquisição de equipamentos ou manutenção.

Grandes empresas provedoras como a *Amazon Web Services (AWS)*⁴ e *Microsoft Azure*⁵, lideram o setor atualmente, mas empresas de todos os portes também podem montar

⁴ <http://aws.amazon.com/pt/>

⁵ <http://azure.microsoft.com/pt-br/>

sua própria nuvem privada para suprir suas necessidades internas e externas de informação e armazenamento de dados.

Este trabalho apresentou uma maneira automatizada para a implantação de uma nuvem privada oportunística, que pode ser adotado por pequenas e médias organizações que buscam inovação e crescimento tecnológico sem maiores gastos financeiro e provendo um uso eficiente de seus *desktops* através da virtualização. A solução desenvolvida está disponível para a comunidade e pode ser utilizada para lidar com a inicialização e desligamento de máquinas físicas (considerando inclusive a necessidade de migrar máquinas virtuais), bem como com a inclusão e remoção dessas máquinas físicas na nuvem.

É importante ressaltar que a solução desenvolvida está implementada para funcionar apenas com nuvens criadas com OpenNebula e para máquinas temporárias que executam sistema operacional Ubuntu. Entretanto, a solução pode ser estendida para trabalhar com outras configurações, uma vez que parte dos *scripts* desenvolvidos e os módulos em *Python* não são específicos para OpenNebula e Ubuntu.

REFERÊNCIAS

ALANIS, P. et al. Uma nuvem privada oportunista para execução de aplicações Bag-of-Tasks. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. 31., 2013, Brasília, DF. **Anais...Brasília**, 2013. p. 1102-1108.

ANTONOPOULOS, Nick; GILLAM, Lee. **Cloud Computing**. London: Springer London, 2010. 382p.

AZEVEDO, E. et al. Nuvem pública versus privada: Variações no desempenho de infraestrutura para elasticidade. In: Workshop em Clouds e Aplicações, 5., 2012, Ouro Preto-MG. **Anais...Ouro Preto, MG: SBC**, 2012. P.110-123.

BARRERA, H. E. C.; ROSERO, E. E. R.; CANO, M. J. V. Clusters computacionales para la investigación: personalizables, eficientes, amigables y a costo cero. **Revista de Ingeniería**, Bogotá-COL, v. 37, p. 13–18, jul./dez. 2012.

BRANDWACHT, L. et al. Models and Guidelines for Dimensioning Private Clouds. In: 6th International Conference on Cloud Computing, 6., 2013. **Anais...Santa Clara, CA: IEEE**, 2013. p. 880-886.

CARVALHO, H. E. T. **Voltaic: um Sistema de Gerência Automatizada de Recursos de Nós Virtuais para Computação em Nuvens**. 2012. Dissertação (Pós-Graduação em Engenharia Elétrica) – COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Dezembro/2012.

CASTRO, H. et al. Green flexible opportunistic computing with task consolidation and virtualization. **Cluster Computing**, Bogotá-COL, v. 16, n. 3, p. 545–557, 18 jul. 2012.

CHAPMAN, C. et al. Software architecture definition for on-demand cloud provisioning. **Cluster Computing**, London, v. 15, n. 2, p. 79–100, 27 fev. 2011.

CONTI, M.; KUMAR, M. Opportunities in opportunistic computing. **Computer**, v. 43, n. 1, p. 42–50, jan. 2010.

DIAZ, C. O. et al. Energy-aware VM Allocation on an Opportunistic Cloud Infrastructure. In: 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, 13., 2013, Delft. **Anais...Delft: IEEE**, 2013. P. 663-670.

FERNANDES, N. C. et al. Virtual networks: isolation, performance, and trends. **annals of telecommunications - annales des télécommunications**, v. 66, n. 5-6, p. 339–355, 7 out. 2011.

LITZKOW, M. J.; LIVNY, M.; MUTKA, M. W. Condor - A Hunter of Idle Workstations. **Department of Computer Sciences University of Wisconsin Madison**, p. 1-8, 1988.

LOUTAS, N. et al. Cloud Computing Interoperability: The State of Play. In: 3th International Conference on Cloud Computing Technology and Science (CloudCom). 3., 2011. **Anais...Athens: IEEE**, 2011. p. 752-757.

MARSHALL, P.; KEAHEY, K.; FREEMAN, T. Improving Utilization of Infrastructure Clouds. IN: 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 11., 2011. **Anais...Newport Beach, CA: IEEE**, 2011. p. 205-214.

MELL, P.; GRANCE, T. The NIST Definition of Cloud Computing - Recommendations of the National Institute of Standards and Technology. **NIST Special Publication**, set. 2011.

NIST. **NIST Cloud Computing Program**. 2014. Disponível em: <<http://www.nist.gov/itl/cloud/>>. Acesso em: 29 abr. 2014.

OPENNEBULA. **The Open Source Toolkit for Cloud Computing**. 2014. Disponível em: <<http://opennebula.org/>>. Acesso em: 30 abr. 2014.

OSORIO, J. D.; CASTRO, H.; BRASILEIRO, F. Perspectives of UnaCloud: An Opportunistic Cloud Computing Solution for Facilitating Research. In: 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 12., 2012. **Anais...Ottawa, ON: IEEE**, 2012. p. 717-718.

OSORIO, J. D.; CASTRO, H.; VILLAMIZAR, M. An opportunistic cloud for e-science. In: 38th Conferencia Latinoamericana En Informatica (CLEI), 38., 2012. **Anais...Medellin: IEEE**, 2012. p. 1-9.

REGO, P. A. L. **FAIRCPU : Uma Arquitetura para Provisionamento de Máquinas Virtuais Utilizando Características de Processamento**. 2012. Dissertação (Pós-Graduação em Ciência da Computação) – Departamento de Computação, Universidade Federal do Ceará, Fortaleza, Março/2012.

RIGHI, R. D. R. Elasticidade em cloud computing: conceito, estado da arte e novos desafios. **Revista Brasileira de Computação Aplicada**, Passo Fundo, RS, v. 5, n. 2, p. 2–17, out. 2013.

RIMAL, B. P.; CHOI, E.; LUMB, I. A Taxonomy and Survey of Cloud Computing Systems. In: 5th INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on. 5., 2009. **Anais...**Seoul: IEEE, 2009. p. 44-51.

ROSALES, E.; CASTRO, H.; VILLAMIZAR, M. UnaCloud : Opportunistic Cloud Computing Infrastructure as a Service. In: 2th The Second International Conference on Cloud Computing, GRIDs, and Virtualization, 2., 2011. **Anais...**Rome, Italy: IARIA, 2011. p. 187-194.

SEMPOLINSKI, P.; THAIN, D. A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus. In: 2th Second International Conference on Cloud Computing Technology and Science, 2., 2010. **Anais...**Indianapolis, IN: IEEE, 2010. p. 417-426.

SEO, C. E. Virtualização – Problemas e desafios. **IBM Linux Technology Center**, v. 1, n. 008278, p. 1–19, 2009.

SHIN, D.; AKKAN, H. Domain-based virtualized resource management in cloud computing. In: 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 6., 2010. **Anais...**Chicago, IL: IEEE, 2010. p. 1-6.

SOUSA, F.; MOREIRA, L.; MACHADO, J. Computação em nuvem: conceitos, tecnologias, aplicações e desafios. In: 3th Escola Regional de Computação Ceará, Maranhão e Piauí ERCEMAPI, 3., 2009. **Anais...**Parnaíba, PI: EDUFPI, 2009. p. 150-175.

UBUNTU DOCUMENTATION. **WakeOnLan**. 2014. Disponível em: <<https://help.ubuntu.com/community/WakeOnLan>>. Acesso em: 05 jun. 2014.

VERDI, F. L. et al. Novas Arquiteturas de Data Center para Cloud Computing. In: 28th Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 28., 2010. **Anais...**Gramado, RS, 2010.

WEN, X. et al. Comparison of open-source cloud management platforms: OpenStack and OpenNebula. In: 9th International Conference on Fuzzy Systems and Knowledge Discovery, 9., 2012. **Anais...**Sichuan: IEEE, 2012. p. 2457-2461.

WERNER, J. **Uma abordagem para alocação de máquinas virtuais em ambientes de computação em nuvem verde**. 2011. Dissertação (Programa de Pós-Graduação em Ciência da Computação) – Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, Fevereiro/2011.

WIKI DEBIAN. **Upstart Position Statement**. 2014. Disponível em: <<https://wiki.debian.org/Debate/initsystem/upstart>>. Acesso em: 10 nov. 2014.