

Uso Oportunístico de Recursos Computacionais em Nuvens Privadas

Francisco R. C. Araújo¹, Paulo A. L. Rego^{1,2}, Michel S. Bonfim¹,
Alisson B. de Souza¹, José N. de Souza²

¹Redes de Computadores - Campus de Quixadá
Universidade Federal do Ceará
Quixadá, Ceará, Brasil

²Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
Universidade Federal do Ceará
Fortaleza, Ceará, Brasil

f.renato@alu.ufc.br, {pauloalr, michelsb, alisson, neuman}@ufc.br

Abstract. *In recent years, the demand for computing resources in several areas of knowledge has grown. Companies, universities and government agencies have sought technologies capable of providing memory, processing and storage with reliability, efficiency, and economy. At the same time, the technological progress has enabled personal computers increasingly robust, while in many institutions and laboratories, several of these equipment are underutilized. In this context, this paper proposes a solution to use "idle" computing resources in an opportunistic manner, using concepts such as Virtualization and Private Clouds in order to ensure a better use of resources and increase the computing power without large investments in equipment. We developed a case study and performed experiments to evaluate the proposed solution.*

Resumo. *Nos últimos anos, a demanda por recursos computacionais tem crescido em diversas áreas do conhecimento. Empresas, universidades e órgãos governamentais têm buscado tecnologias capazes de prover memória, processamento e armazenamento com confiabilidade, eficiência e economia. Ao mesmo tempo, o avanço tecnológico vem possibilitando computadores pessoais cada vez mais robustos, ao passo que, em muitas instituições e laboratórios, vários desses equipamentos são subutilizados. Neste contexto, este trabalho propõe uma solução para usar de forma oportunística recursos computacionais "ociosos", usando conceitos de Virtualização e Nuvens Privadas, a fim de garantir um melhor aproveitamento dos recursos e ampliar o poder computacional sem grandes investimentos em equipamentos. Um estudo de caso e experimentos foram realizados para validar a solução proposta.*

1. Introdução

A computação se tornou indispensável para o desenvolvimento de outras tecnologias e áreas do conhecimento, incluindo biotecnologia, farmacêutica, medicina e engenharia. O uso de simulação, modelagem e outras tecnologias computacionais ajudam esses campos da ciência 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 [Osorio et al. 2012a]. Assim, necessita-se de recursos tecnológicos e infraestrutura robusta para atender aos clientes e serviços.

Para tanto, é preciso investir em equipamentos e, conseqüentemente, em manutenção, podendo gerar um grande gasto financeiro com material e pessoal e restringir algumas organizações [Osorio et al. 2012a]. Além disso, outro custo importante é o de consumo de energia. Quanto maior for a quantidade de máquinas presentes em uma infraestrutura, maior será o consumo de energia e o custo envolvido [Beldiceanu et al. 2015].

Para possibilitar um melhor aproveitamento do *hardware* das máquinas, surgiram várias tecnologias. Dentre elas, destaca-se a virtualização - técnica que permite a abstração de recursos computacionais e propicia o desenvolvimento de aplicações que compartilham a mesma infraestrutura de *hardware* [Kumar and Singh 2015]. Para aplicar essas técnicas de virtualização a fim de alcançar um maior rendimento em problemas de computação de grande escala, surgiu a Computação em Nuvem - paradigma que prega a provisão de recursos dinamicamente escaláveis através da *Internet* [Rimal et al. 2009]. Neste paradigma, os recursos físicos (CPU, rede, armazenamento) são distribuídos como serviços, usando uma forma de pagamento baseado no uso.

As nuvens - conjunto de *hardware*, redes, CPU, memória, aplicações e plataformas disponibilizados como serviços - possuem diferentes modelos de implantação. Destacamos dois modelos: nuvens públicas e privadas [Kumar and Singh 2015]. As nuvens públicas (e.g., Google Cloud Platform, Amazon EC2 e Microsoft Azure) estão sendo amplamente utilizadas, pois disponibilizam um meio confiável e escalável para as empresas que queiram “exportar” suas infraestruturas de TI, juntamente com os seus serviços.

Entretanto, um número crescente de organizações está adotando o modelo de nuvem privada, que engloba as infraestruturas de nuvens restritas a uma empresa sendo, muitas vezes, construídas a partir de recursos computacionais subutilizados existentes no local. Com este aumento na demanda de nuvens privadas, cria-se um cenário onde este tipo de nuvem deve estar constantemente ampliando o seu poder computacional, para atender as novas aplicações [Rego et al. 2015]. Contudo, para ampliar a sua capacidade é necessário realizar investimentos em novos equipamentos, o que pode aumentar consideravelmente o custo de capital da empresa [Brandwacht et al. 2013].

Como forma de reduzir ainda mais o impacto desses investimentos, a Computação em Nuvem pode se utilizar do conceito de Computação Oportunística, cujo princípio é utilizar capacidades de computação que estejam sendo desperdiçadas para criar a chamada “Nuvem Oportunística”, quando existir oportunidade [Osorio et al. 2012b] [Conti and Kumar 2010]. A ideia é que quaisquer instituições ou empresas possam criar, de maneira fácil e automatizada, uma nuvem privada oportunística. Isso é especialmente importante em locais que possuem recursos computacionais ociosos (por exemplo, computadores em laboratórios de informática). Nesses ambientes, os usuários locais dificilmente utilizam toda a capacidade disponível dos recursos computacionais, havendo um desperdício destes [Rosales et al. 2011]. O objetivo da “Nuvem Oportunística” é ampliar temporariamente o poder computacional de suas infraestruturas sem investir em novos equipamentos, oferecendo aos usuários da nuvem novos recursos computacionais quando for necessário e viável.

Este trabalho propõe uma solução que possibilita 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. O objetivo é 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 possuem prioridade sobre o uso de recursos. Foram realizados estudo de caso e experimentos para validar a solução, utilizando máquinas do Laboratório de Redes, no Campus da UFC em Quixadá.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta o estado da arte, enquanto a Seção 3 apresenta a solução proposta. Na Seção 4, os experimentos e resultados são apresentados, e por fim, a Seção 5 apresenta as considerações finais obtidas com a realização deste estudo.

2. Trabalhos Relacionados

O conceito de computação oportunística tem sido utilizado em diversos trabalhos nos últimos anos, destacando-se os projetos Condor [Litzkow et al. 1988], BOINC [Anderson and Fedak 2006] e UnaCloud [Osorio et al. 2012a]. O Condor foi um dos primeiros trabalhos na área de computação oportunística e dependia de máquinas com recursos homogêneos. O sistema proposto desencadeou o interesse pela computação ociosa presente em laboratórios, mas com escala limitada. A evolução desta ideia trouxe o poder da computação ociosa usando a *Internet*, a chamada computação voluntária. Nesse mesmo período surgiram as grades computacionais, caracterizadas por um escopo maior com milhares de computadores.

No contexto de grades computacionais, o projeto BOINC [Anderson and Fedak 2006], da Universidade de Berkeley, permite que pesquisadores utilizem uma infraestrutura de computação compartilhada para fazer pesquisa. Nesse projeto, os participantes disponibilizam equipamentos completamente ou parcialmente ociosos para compor o *pool* de recursos. Diferente do nosso trabalho, a solução é baseada em um agente executado nas máquinas físicas, que permite a execução de processos (paralelos ou não), mas não de máquinas virtuais (MVs).

Por outro lado, o projeto UnaCloud [Osorio et al. 2012a], da Universidade de los Andes na Colombia, foi o primeiro a implementar de forma funcional os conceitos de computação oportunística em nuvem. O UnaCloud foi desenvolvido para prover uma infraestrutura oportunística (distribuída, heterogênea e de diferentes domínios administrativos) capaz de suportar um modelo de nuvem IaaS (*Infrastructure as a Service*) experimental a um menor custo e oferecer os mesmos benefícios e funcionalidades disponíveis em modelos de nuvem convencionais. Assim, grandes recursos computacionais poderiam ser acessados pelos usuários com a facilidade e baixos custos da computação em nuvem e os gerentes de serviços teriam à disposição uma solução de computação de alto desempenho sem fazer grandes investimentos em equipamentos e manutenção.

O UnaCloud foi implementado utilizando a linguagem de programação Java e necessita do ambiente Windows para ser executado, além de depender dos hipervisores *VMware* e *VirtualBox*. Apesar do projeto UnaCloud ter inspirado nosso trabalho, propomos o desenvolvimento de uma solução própria completamente implementada com

soluções de código aberto. Assim, o presente trabalho diferencia-se do UnaCloud por ter sido implementado em *Shell Script* e *Python*, e por utilizar o hipervisor KVM e a ferramenta OpenNebula para o gerenciamento da nuvem.

A ferramenta *Lobster* [Woodard et al. 2015] foi desenvolvida na Universidade de *Notre Dame* para gerenciar os ciclos computacionais disponíveis nas máquinas da instituição. A ferramenta, que consiste de um gerenciador de arquivos e tarefas, bem como ferramentas de monitoramento e diagnósticos, processa cargas de trabalho submetidas por usuários de acordo com a disponibilidade de recursos. O sistema usa uma política de filas para gerenciar as tarefas entre os núcleos de processamento oportunistas ou dedicados, e os dados são manipulados localmente através de *Chirp* e *Hadoop*, além de contar com uma área de federação mais ampla através do *XrootD*. A solução foi desenvolvida com foco direcionado apenas às atividades de pesquisa e conta com uma arquitetura estática, voltada especialmente para tarefas analíticas de HEP (*High Energy Physics*), além de não prover o suporte à criação de MVs.

Por fim, no trabalho [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 BoT (*Bag-of-Tasks*), que não exigem uma alta qualidade de serviço. Os autores desenvolveram uma plataforma tecnológica dinâmica, flexível e a um baixo custo capaz de fornecer suporte para aplicações BoT. A solução é baseada na plataforma *Eucalyptus* [Sempolinski and Thain 2010], que foi modificada para suportar o hipervisor *VirtualBox*. Apesar de bem sucedido, esse trabalho é voltado para aplicações BoT e visa utilizar o ambiente de nuvem oportunística somente fora do horário de trabalho dos usuários (e. g., durante a madrugada ou feriados). Por outro lado, nossa solução utiliza os recursos ociosos inclusive em horários de trabalho, contando com um sistema de monitoramento da utilização dos recursos das máquinas físicas, a fim de possibilitar a utilização das máquinas enquanto os usuários locais as estão usando. Para isso, é preciso lidar com possíveis falhas e desligamentos não programados das máquinas não dedicadas.

3. Solução Proposta

Este artigo propõe uma solução para possibilitar a integração e uso oportunístico de recursos computacionais ociosos (i.e., presentes em máquinas físicas não dedicadas e subutilizadas) a uma nuvem privada. O objetivo é proporcionar um melhor aproveitamento da infraestrutura presente em instituições (e.g., universidades) e, por conseguinte, aumentar a quantidade de recursos disponíveis para criação de máquinas virtuais e execução de tarefas computacionais e pesquisa, sem a necessidade de grandes investimentos financeiros.

Para tanto, criamos uma nuvem privada oportunística que, além de usar os recursos ociosos de máquinas que não estão sendo utilizadas (desligadas), também usam os recursos utilizados em tempo-real pelo usuário local. Esta última aparece como opção quando os usuários não consomem toda a capacidade computacional presente na máquina. Para isso, contamos com um sistema de monitoramento da utilização dos recursos das máquinas físicas e gerenciamos as flutuações das cargas de trabalho e possíveis falhas e desligamentos não programados destas máquinas.

O desenvolvimento da solução proposta neste estudo foi guiado pelos seguintes requisitos:

- A integração e utilização de máquinas não dedicadas à nuvem deve ocorrer simultaneamente ao uso dos usuários finais, sem prejudicá-los em suas atividades cotidianas, ficando assim transparente para os mesmos;
- Deve ser possível integrar estações de trabalho inicialmente desligadas à nuvem (i.e., as máquinas devem ser ligadas programaticamente);
- Deve ser possível desligar máquinas físicas não dedicadas quando os recursos não são mais necessários;
- Deve-se tratar possíveis eventos de desligamento das máquinas não dedicadas por parte dos usuários finais, a fim de evitar que as máquinas virtuais em execução sejam perdidas ou corrompidas;
- A solução deve ser baseada em *software* livre;
- As máquinas que são oportunisticamente integradas à nuvem devem executar sistema operacional Linux e suportar virtualização.

A arquitetura da solução, bem como seus principais componentes e funcionalidades são apresentados nas próximas seções.

3.1. Arquitetura da Solução

A arquitetura da nossa solução é baseada no conceito de UnaCloud [Osorio et al. 2012a], mas completamente implementada em código aberto. Ela foi desenvolvida usando *Shell Script* e *Python*, além de utilizar o hipervisor KVM e a ferramenta OpenNebula para o gerenciamento da nuvem.

Utilizamos o hipervisor KVM por ele apresentar melhores desempenhos em pesquisas de Computação de Alto Desempenho (CAD), principalmente em termos de capacidade de computação e de expansibilidade de memória [Younge et al. 2011]. Já o OpenNebula foi escolhido pelo fato de que os nós de computação não precisam de uma grande quantidade de recursos de disco rígido e porque ele permite mudanças em praticamente todos os seus componentes, incluindo o sistema de arquivos e o *front-end* [Sempolinski and Thain 2010]. Além disso, ele é adequado tanto para institutos de pesquisas, quanto para universidades e empresas [Wen et al. 2012].

A solução proposta segue o modelo de arquitetura gerente-agente, onde um gerente é executado em uma das máquinas dedicadas da nuvem privada, enquanto os agentes são executados nas máquinas não dedicadas. Uma visão geral da arquitetura da solução é apresentada na Figura 1. Do lado esquerdo, pode-se observar um exemplo de topologia física de um prédio composto por dois ambientes de trabalho (e.g., laboratórios) e uma sala de telecomunicações, interconectados utilizando *switches*. Na sala de telecomunicações, pode-se observar *racks* com equipamentos dedicados à nuvem privada, onde são executados o Controlador e o *Storage* da nuvem, bem como o Gerente da solução proposta. Já nos ambientes de trabalho, pode-se observar diversas máquinas que não são dedicadas à nuvem e que executam os Agentes.

Já do lado direito, pode-se observar a topologia lógica, onde as máquinas da sala de telecomunicações compõem o *pool* de recursos dedicado à nuvem privada, enquanto as máquinas presentes nas áreas de trabalho são oportunisticamente adicionadas e removidas do *pool* de recursos da nuvem, formando uma nuvem oportunística. Com essa solução, a nuvem privada pode ser expandida para usar parte ou todos os recursos computacionais disponíveis no prédio, bem como retraída para usar apenas as máquinas dedicadas,

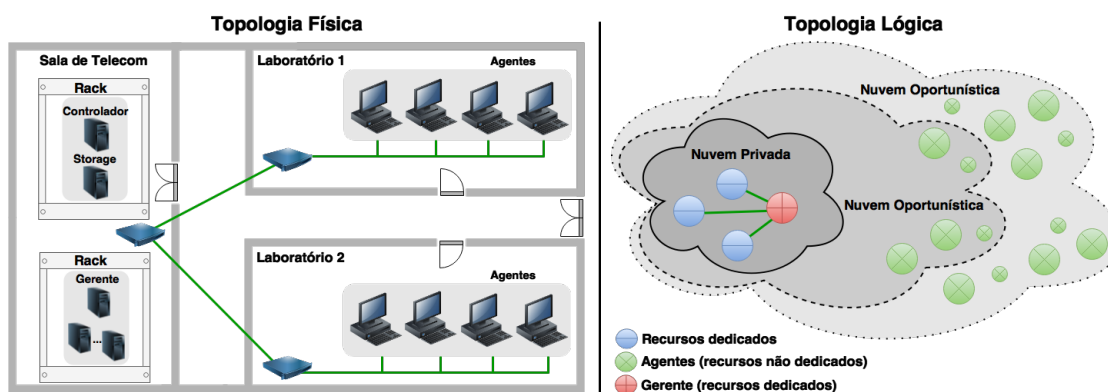


Figura 1. Visão geral da arquitetura da solução proposta. A topologia física apresenta a disposição dos equipamentos no prédio, enquanto a topologia lógica ilustra como a nuvem privada pode se expandir utilizando os recursos não dedicados, formando a nuvem oportunística.

fazendo assim um melhor uso dos recursos disponíveis. Além disso, é possível ajustar os recursos da nuvem às variações de demanda sem a necessidade de fazer investimentos com a compra de novos equipamentos.

Storage representa o sistema de armazenamento distribuído da nuvem, que é necessário para que a técnica de *live migration* possa ser utilizada para transferir máquinas virtuais de uma máquina física para outra sem desconectar clientes e aplicações ou interromper os serviços em execução nas máquinas virtuais. Neste estudo, NFS (*Network File System*) é o sistema utilizado.

O Controlador da nuvem é o módulo responsável por coordenar e monitorar as máquinas físicas, bem como gerenciar o ciclo de vida das máquinas virtuais. Além disso, ele é o módulo que gerencia a nuvem e provê acesso a serviços e APIs, que são acessados pelo módulo Gerente para controlar o OpenNebula.

O módulo Gerente é executado em uma das máquinas dedicadas à nuvem e é responsável pela comunicação e gerenciamento das máquinas não dedicadas, através do módulo Agente. O Gerente utiliza os serviços oferecidos pelo Controlador para verificar continuamente os recursos disponíveis na nuvem, a fim de atender à quantidade de máquinas virtuais requisitadas pelos usuários. Para tal, pode ser preciso adicionar máquinas físicas não dedicadas à nuvem, caso os recursos disponíveis sejam insuficientes para a instanciação e execução de alguma máquina virtual. Além disso, pode ser preciso ligar automaticamente máquinas que por ventura estejam desligadas. A funcionalidade de ligar e desligar máquinas não dedicadas, bem como adicioná-las e removê-las da nuvem envolve comunicação entre os módulos Gerente e Agente, e será explicada na próxima seção.

O módulo Agente é executado nas máquinas físicas não dedicadas e foi implantado como um *daemon* no Linux em `/etc/init.d`, seguindo o modelo do Ubuntu/Debian, para ser executado sempre que o sistema operacional está em processo de inicialização, reinicialização ou desligamento. Para isso, quando o módulo é executado, ele se comunica com o Gerente para que este tome as ações de pós-inicialização ou pré-desligamento necessárias. Tais ações fazem o Agente trabalhar em três modos de

operação: Inicialização, Monitoramento e Desligamento. A Figura 2 apresenta as mensagens trocadas entre o Agente, Gerente e Controlador durante os modos de operação mencionados.

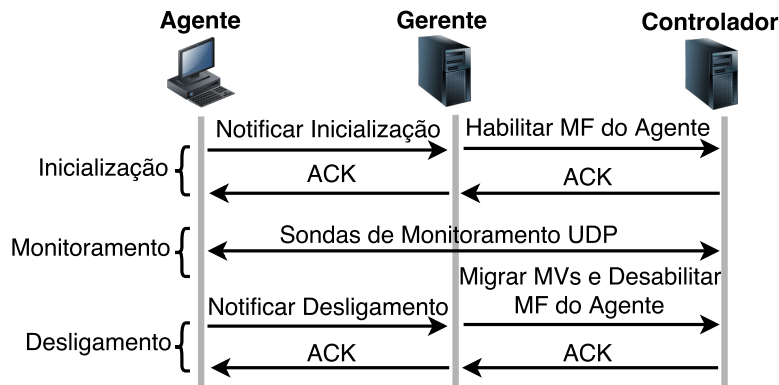


Figura 2. Comunicação entre os módulos da solução

No modo de Inicialização, o Agente é inicializado junto com os outros serviços durante o processo de inicialização do sistema operacional. Assim que é inicializado, o Agente constantemente monitora a disponibilidade do serviço de rede, a fim de notificar o Gerente sobre a inicialização da máquina em que ele é executado e acessar o sistema de arquivos NFS do *Storage*. Após ser notificado, o Gerente habilita a máquina do Agente na nuvem, que por sua vez envia mensagem ao Agente confirmando o início do modo de monitoramento.

Neste modo, após ser habilitada na nuvem, a máquina em que o Agente é executado passa a ter seus recursos integrados ao *pool* de recursos da nuvem, e a máquina física fica pronta para receber instâncias de máquinas virtuais, obedecendo a quantidade de recursos não reservados. Nesse estado de operação, não há troca de mensagens entre os módulos Agente e Gerente. Entretanto, o sistema de monitoramento do OpenNebula, que por padrão utiliza conexões baseadas no protocolo UDP, envia mensagens de monitoramento ao Controlador da nuvem para informar a quantidade de recursos não utilizados.

Por outro lado, ao perceber que o usuário mandou desligar a máquina, o Agente começa a operar no modo de desligamento. Primeiro ele notifica o Gerente sobre a intenção do usuário de desligar o equipamento, e em seguida bloqueia o processo de desligamento enquanto espera por mensagem de confirmação do Gerente. O Gerente por sua vez, toma as ações necessárias para evitar que as máquinas virtuais em execução naquela máquina física sejam desligadas ou corrompidas (e.g., migrando as máquinas virtuais para outra máquina física) e, após isso, notifica o Agente para que este continue o processo de desligamento/reinicialização normal do sistema operacional.

3.2. Funcionalidades

Esta seção apresenta detalhes de implementação de algumas das funcionalidades da solução proposta. Mesmo incluindo um conjunto de funcionalidades, nossa solução ainda se encontra em um estágio inicial de desenvolvimento, carecendo de algumas funcionalidades importantes, como mecanismos de reescalonamento de máquinas virtuais.

3.2.1. Gerência das máquinas

O administrador da nuvem tem autonomia para ligar, desligar ou reiniciar uma ou mais máquinas físicas remotamente. Tal funcionalidade permite um maior controle sobre a infraestrutura, uma vez que permite o administrador desligar todas as máquinas que não estão sendo utilizadas com um único comando. Além disso, permite também que o administrador ligue todas as máquinas disponíveis na instituição por algumas horas durante a madrugada para fazer algum experimento.

A funcionalidade de ligar as máquinas remotamente foi implementada utilizando a técnica de ativação remota *Wake-On-Lan*, que permite que computadores sejam ligados através do envio de um pacote de dados especial para a máquina alvo. Nas máquinas que suportam essa tecnologia, a placa de rede continua ligada e verificando a rede mesmo quando a máquina está desligada. Assim, uma vez que o pacote é recebido, a placa de rede liga a máquina.

3.2.2. Instalação automatizada do módulo Agente

Para instalar o Agente nas máquinas, o administrador precisa apenas do endereço de rede das novas máquinas e as credenciais de acesso. Em posse desses dados, ele pode executar um *script* que transfere o instalador do módulo Agente para as novas máquinas e o executa. Tal instalador configura todo o ambiente para execução da nuvem oportunística (e.g., troca de chaves de segurança, instalação do hipervisor, instalação do cliente NFS, etc).

3.2.3. Migração automática de máquinas virtuais

Quando o usuário manda desligar a máquina física, o Agente notifica o Gerente, e este verifica se existem máquinas virtuais sendo executadas naquela máquina física. Em caso positivo, o Gerente solicita ao Controlador da nuvem a migração das máquinas virtuais para outras máquinas físicas. A migração das máquinas virtuais se dá através da técnica conhecida como *live migration*, que permite que as máquinas sejam migradas sem interrupção dos serviços em execução. Tal técnica migra o processamento, memória e fluxo de rede para a nova máquina física, e depende de um sistema de arquivos compartilhado e distribuído, por isso o NFS é utilizado. Após a migração das máquinas, o Gerente notifica o Agente, que permite que o processo de desligamento da máquina física continue.

3.2.4. Elasticidade com base no melhor esforço

O módulo Gerente procura atender todas as requisições para criação de máquinas virtuais. Para isso, ele utiliza as funcionalidades descritas em Gerência das máquinas (Subseção 3.2.1) para inicializar automaticamente máquinas não dedicadas e integrá-las à nuvem. Entretanto, todo esse processo é feito baseado no melhor esforço e, por enquanto, não existe mecanismo de reescalonamento de máquinas virtuais para otimizar o uso de recursos uma vez que as máquinas virtuais estão em execução. Isso pode levar a falta de recursos quando todas as máquinas não dedicadas já estiverem sendo utilizadas, o que

vai deixar novas máquinas virtuais em estado pendente. Já com relação ao desligamento, uma máquina física é automaticamente desligada quando as máquinas virtuais que estiverem sendo executadas nela forem desligadas ou migradas para outra máquina física, e não houver usuário logado na máquina. Nesse caso, o Gerente desabilita a máquina e notifica o Agente, que desliga a máquina.

4. Estudo de Caso

Com o objetivo de validar a solução proposta, um estudo de caso foi elaborado no Laboratório de Redes de Computadores, no Campus Quixadá da Universidade Federal do Ceará. As máquinas do laboratório foram utilizadas para estender, de forma oportunística, os recursos de uma nuvem privada.

A Figura 3 ilustra a topologia adotada como cenário do estudo de caso. A infraestrutura permanente da nuvem privada é composta por três máquinas físicas dedicadas (MF1, MF2 e MF3), sendo a MF1 o controlador da nuvem. Por outro lado, cinco máquinas físicas (MF4, MF5, MF6, MF7 e MF8) podem ser utilizadas temporariamente na nuvem, sendo integradas de forma oportunística.

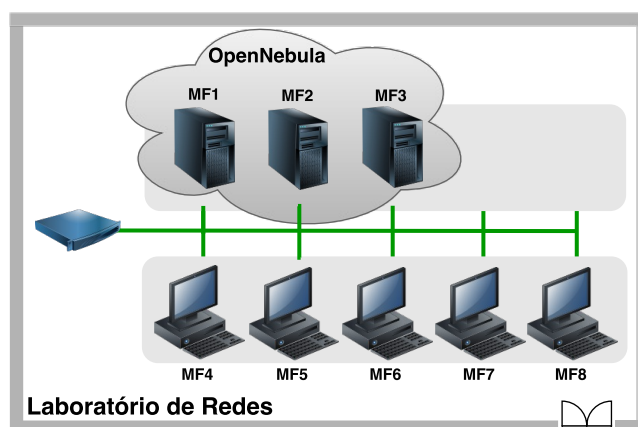


Figura 3. Cenário de Experimentos

As oito máquinas físicas utilizadas tinham as mesmas configurações, processador Intel Core I5 (4 cores), 8 GB de memória RAM e 1 TB de espaço em disco; enquanto todas as máquinas virtuais criadas tinham 1 VCPU (CPU virtual) e 2 GB de memória RAM.

As seguintes premissas foram consideradas:

1. Todos os recursos das máquinas físicas dedicadas podem ser utilizados para alocar máquinas virtuais (i.e., 4 cores de processamento e 8 GB de memória). Máximo de 4 VMs em cada máquina física.
2. Apenas metade dos recursos das máquinas físicas temporárias podem ser utilizados para alocar máquinas virtuais (i.e., 2 cores de processamento e 4 GB de memória). Máximo de 2 VMs em cada máquina física.

Diante do cenário e das premissas apresentadas, o estudo de caso considera diferentes possibilidades que podem ocorrer no cotidiano de um laboratório de informática.

O experimento realizado é descrito a seguir e o estado de alocação de recursos nas máquinas físicas da nuvem pode ser visualizado na Figura 4.

1. Etapa I: Inicialmente, pode-se observar duas máquinas físicas dedicadas (MF2 e MF3) ligadas e prontas pra receber máquinas virtuais. O usuário solicita à nuvem a criação de 9 MVs (MV1, MV2, ..., MV9). Entretanto, não há espaço suficiente para atender às requisições, pois as máquinas físicas MF2 e MF3 comportam apenas quatro MVs cada uma. Assim a MV9 fica no estado pendente. Nesse caso, o módulo Gerente verifica a falta de recursos na nuvem e aloca oportunisticamente uma máquina física temporária (que pode estar sendo utilizada por algum usuário do laboratório ou mesmo desligada);
2. Etapa II: O Gerente liga a MF4 e o escalonador do OpenNebula se encarrega de alocar a MV9, que estava pendente. Em seguida, o usuário da nuvem solicita uma nova máquina virtual (MV10), que é alocada na MF4, uma vez que esta contém os recursos necessários para receber a MV;
3. Etapa III: Em dado momento, a MF5 é ligada por um usuário do laboratório, e pouco depois uma nova máquina virtual (MV11) é solicitada à nuvem. A MV11 é, então, alocada à máquina física MF5;
4. Etapa IV: O usuário da nuvem decide remover as máquinas virtuais MV8 e MV9, ocasionando liberação parcial de recursos das máquinas físicas MF3 e MF4, que se encontram aptas a receber novas MVs, uma vez que tiveram recursos liberados. A MF5 também tem recursos livres, já que executa apenas uma MV.

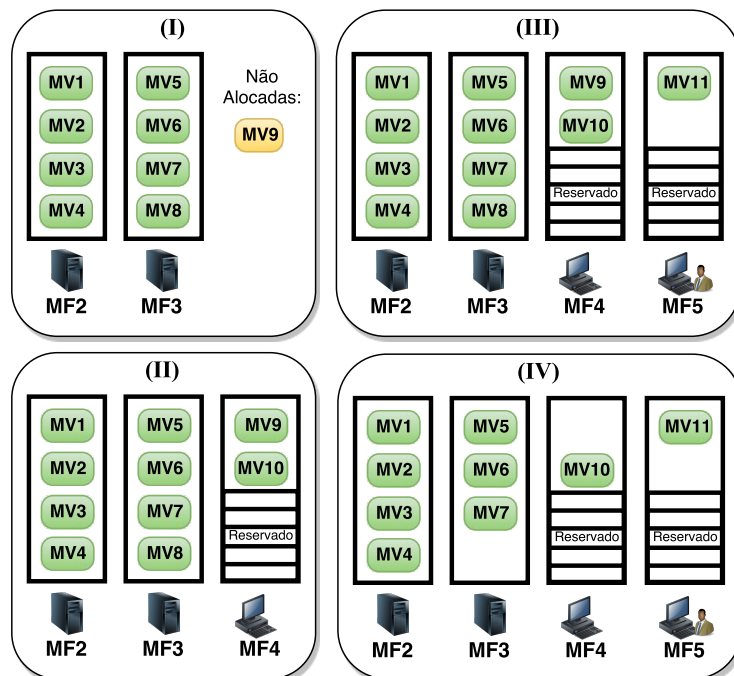


Figura 4. Alocação de recursos na nuvem durante as etapas do estudo de caso

A reinicialização e o desligamento de *desktops* com máquinas virtuais em execução pode acontecer. Um experimento foi realizado para contemplar esta situação. Tal experimento utilizou o estado final das máquinas físicas e virtuais do cenário anterior (Figura 4 - IV).

Na Figura 5 - I pode-se visualizar que a MF5 encontra-se ligada e está executando a máquina virtual MV11. O usuário do laboratório inicia o processo de desligamento desta máquina, logo a solução deve migrar automaticamente a MV11 para a máquina física 3 ou 4, pois ambas possuem recursos livres. A MV em questão é então movida para a MF3. Por outro lado, na Figura 5 - II, quando a MV10 é desligada pelo usuário, a solução automaticamente libera os recursos da MF4 e a desliga, pois esta encontra-se sem MVs e sem usuários logados, já que foi ligada pelo módulo Gerente.

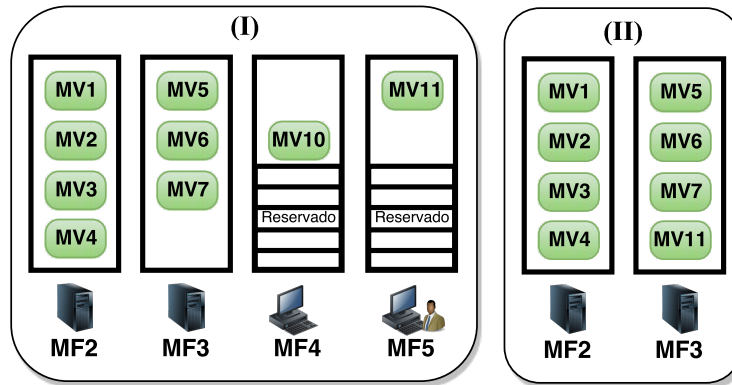


Figura 5. Alocação de recursos na nuvem após processo de desligamento

Como forma de verificar a viabilidade da solução em ambientes de produção, foram feitas medições do tempo de inicialização da MF4, até o momento em que a mesma se torna disponível para o OpenNebula alocar MVs. Foram realizados 30 experimentos para uma análise estatística (nível de confiança de 95%), onde os dados obtidos indicam um tempo médio de inicialização de 53,8 segundos, com um intervalo de confiança de 0,29. Como nossa nuvem privada leva em média 60 segundos para instanciar uma MV, podemos considerar aceitável o tempo de aproximadamente 2 minutos para instanciar uma MV em uma máquina física temporária que precisa ser ligada. Tal valor pode ser comparado ao tempo médio para instanciar MVs na Amazon EC2, que de acordo com a avaliação de desempenho apresentada em [Ostermann et al. 2009], varia aproximadamente de 40 a 110 segundos, dependendo do tipo de instância requisitada.

4.1. Análise de Desempenho

Para ilustrar as vantagens de utilizar o conceito de nuvem oportunística, um experimento foi realizado com o sistema de recomendação de fotos MMedia2UE (*Mobile Media to You Extended*), proposto em [Rego et al. 2013].

O MMedia2UE utiliza o paradigma *MapReduce* para analisar uma base de dados de fotos anotadas com informações contextuais, a fim de paralelizar a tarefa de recomendar fotos em um contexto similar ao do usuário, sendo essa similaridade uma função de três dimensões contextuais (espacial, social e temporal)[Lemos et al. 2012]. O serviço de recomendação foi implementado com o *framework* Apache *Hadoop* e bases de dados de diferentes tamanhos foram geradas. A Tabela 1 apresenta a relação entre o tamanho da base de dados e a quantidade de fotos.

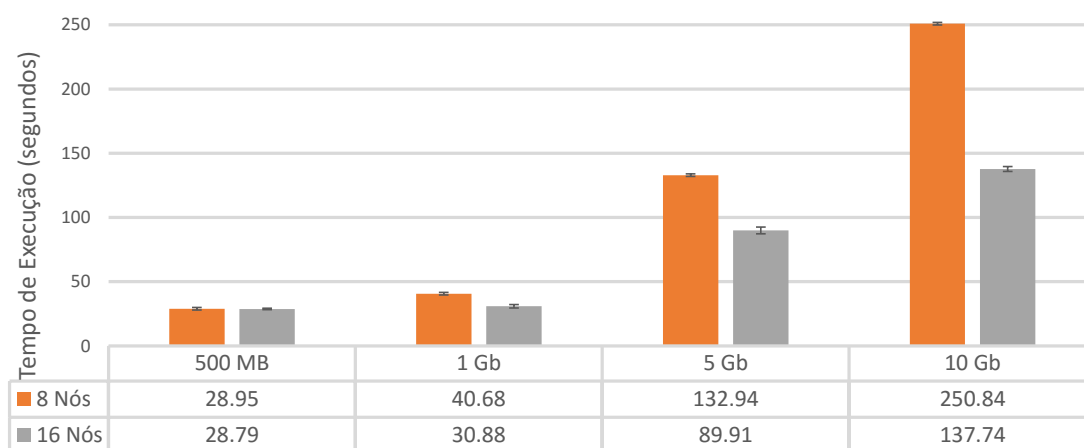
Em [Rego et al. 2013], foi avaliado o desempenho do sistema de recomendação ao utilizar *clusters Hadoop* com 2, 4 e 8 nós. Neste trabalho, o sistema de recomendação

Tabela 1. Relação entre o tamanho da base de dados e quantidade de fotos

Tamanho	500 MB	1000 MB	5 GB	10 GB
Quantidade de Fotos	2.062.103	4.124.206	20.735.400	41.234.562

foi executado em dois cenários: (i) nuvem privada apenas com máquinas dedicadas e (ii) nuvem oportunística. No primeiro caso, um *cluster Hadoop* com 8 nós foi criado, enquanto um *cluster* com 16 nós foi criado no segundo caso. Como cada MV tinha 1 VPCU e 2 GB de memória, a única maneira de alocar 16 nós *Hadoop* é utilizando MFs temporárias para estender a quantidade de recursos da nuvem privada.

A Figura 6 apresenta o tempo médio de execução do sistema de recomendação em cada cenário e para diferentes tamanhos de base de dados. Cada experimento foi executado 30 vezes a fim de calcular a média com 95% de confiança. É importante destacar que, em todos os experimentos, o sistema de arquivos do *Hadoop* (HDFS - *Hadoop Distributed File System*) foi configurado com os valores padrões para taxa de replicação e tamanho de bloco, 3 e 64 MB respectivamente.

**Figura 6. Comparação entre o tempo de execução na nuvem com máquinas dedicadas (8 Nós) e nuvem oportunística (16 Nós)**

Para a base de dados de 500 MB, pode-se observar que o tamanho do *cluster* tem pouca influência no tempo de execução. Tal resultado é explicado pelo fato do tamanho do bloco do HDFS ser igual a 64 MB. Assim, apenas 8 nós do *cluster* são utilizados durante o processamento, uma vez que é preciso executar apenas oito tarefas *Map* (teto da divisão de 500 por 64). Portanto, para tal tamanho de base de dados, não há vantagem de se utilizar o maior *cluster*.

Por outro lado, para as bases de dados de 1 Gb a 10 Gb, pode-se observar o ganho de desempenho ao utilizar a nuvem oportunística. A redução no tempo de execução chegou a aproximadamente 45% ao utilizar a nuvem oportunística com uma base de dados de 10 Gb. Ou seja, quanto maior a base de dados, mais vantajoso é ter mais nós *Hadoop* para processar as tarefas, o que demonstra uma evidente vantagem das nuvens oportunísticas.

5. Conclusão

A adoção da computação em nuvem traz inúmeras vantagens para as organizações. Diante disto, grandes empresas como Amazon, Microsoft e Google têm oferecido serviços da chamada nuvem pública. Entretanto, é importante ressaltar que qualquer empresa, independente do porte, pode criar uma nuvem privada com recursos próprios a fim de suprir suas necessidades de computação e armazenamento de dados.

Nesse contexto, este trabalho apresentou uma solução inicial para possibilitar a integração e uso oportunístico de recursos computacionais ociosos a uma nuvem privada OpenNebula. Tal solução pode ser adotada por pequenas e médias organizações que busquem melhor aproveitar sua infraestrutura e equipamentos, e por conseguinte aumentar a quantidade de recursos disponíveis para criação de máquinas virtuais e execução de tarefas computacionais e pesquisa, sem a necessidade de grandes investimentos financeiros. Um estudo de caso e experimentos foram apresentados para validar algumas das funcionalidades da solução proposta e mostrar o ganho de desempenho de uma aplicação *Hadoop* ao utilizar a nuvem oportunística.

Por fim, a solução desenvolvida está disponível para a comunidade¹ e pode ser utilizada e estendida para atender diferentes casos de uso e suportar outros *software* de gerenciamento de nuvem, como OpenStack. Como trabalhos futuros, pretende-se investigar mecanismo de reescalonamento de máquinas virtuais e implementar e avaliar o suporte à virtualização baseada em *containers*.

Referências

- Alanis, P., Barros, A., Brasileiro, F., and Nóbrega, M. (2013). Uma nuvem privada oportunista para execução de aplicações bag-of-tasks. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 1102–1108.
- Anderson, D. P. and Fedak, G. (2006). The computational and storage potential of volunteer computing. In *Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on*, volume 1, pages 73–80. IEEE.
- Beldiceanu, N., Feris, B. D., Gravey, P., Hasan, S., Jard, C., Ledoux, T., Li, Y., Lime, D., Madi-Wamba, G., Menaud, J. M., Morel, P., Morvan, M., Moulinard, M. L., Orgerie, A. C., Pazat, J. L., Roux, O., and Sharaiha, A. (2015). The epoc project: Energy proportional and opportunistic computing system. In *Smart Cities and Green ICT Systems (SMARTGREENS), 2015 International Conference on*, pages 1–7.
- Brandwacht, L., Meeuwissen, E., van den Berg, H., and Ivkovic, M. (2013). Models and guidelines for dimensioning private clouds. In *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*, pages 880–886.
- Conti, M. and Kumar, M. (2010). Opportunities in opportunistic computing. *Computer*, 43(1):42–50.
- Kumar, D. and Singh, A. S. (2015). A survey on resource allocation techniques in cloud computing. In *Computing, Communication Automation (ICCCA), 2015 International Conference on*, pages 655–660.

¹Código fonte disponível em: <https://github.com/Renato2012/NuvemOportunistica2.0.git>

- Lemos, F. D., Carmo, R. A., Viana, W., and Andrade, R. M. (2012). Improving photo recommendation with context awareness. In *Proceedings of the 18th Brazilian Symposium on Multimedia and the Web, WebMedia '12*, pages 321–330, New York, NY, USA. ACM.
- Litzkow, M., Livny, M., and Mutka, M. (1988). Condor—a hunter of idle workstations. In *Distributed Computing Systems, 1988., 8th International Conference on*, pages 104–111.
- Osorio, J., Castro, H., and Brasileiro, F. (2012a). Perspectives of unacloud: An opportunistic cloud computing solution for facilitating research. In *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on*, pages 717–718.
- Osorio, J., Castro, H., and Villamizar, M. (2012b). An opportunistic cloud for e-science. In *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En*, pages 1–9.
- Ostermann, S., Iosup, A., Yigitbasi, N., Prodan, R., Fahringer, T., and Epema, D. (2009). A performance analysis of ec2 cloud computing services for scientific computing. In *Cloud computing*, pages 115–131. Springer.
- Rego, P. A. L., Bonfim, M. S., Ortiz, M. D., Bezerra, J. M., Campelo, D. R., and de Souza, J. N. (2015). An openflow-based elastic solution for cloud-cdn video streaming service. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7.
- Rego, P. A. L., Lemos, F. D., Viana, W., Trinta, F., and de Souza, J. N. (2013). Mapreduce performance evaluation for knowledge-based recommendation of context-tagged photos. In *Proceedings of the 19th Brazilian Symposium on Multimedia and the Web, WebMedia '13*, pages 249–256. ACM.
- Rimal, B., Choi, E., and Lumb, I. (2009). A taxonomy and survey of cloud computing systems. In *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*, pages 44–51.
- Rosales, E., Castro, H., and Villamizar, M. (2011). Unacloud: Opportunistic cloud computing infrastructure as a service. *Cloud Computing*, pages 187–194.
- Sempolinski, P. and Thain, D. (2010). A comparison and critique of eucalyptus, opennebula and nimbus. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pages 417–426.
- Wen, X., Gu, G., Li, Q., Gao, Y., and Zhang, X. (2012). Comparison of open-source cloud management platforms: Openstack and opennebula. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pages 2457–2461.
- Woodard, A., Wolf, M., Mueller, C., Tovar, B., Donnelly, P., Anampa, K. H., Brenner, P., Lannon, K., Hildreth, M., and Thain, D. (2015). Exploiting volatile opportunistic computing resources with lobster. In *Journal of Physics: Conference Series*, volume 664. IOP Publishing.
- Younge, A., Henschel, R., Brown, J., von Laszewski, G., Qiu, J., and Fox, G. (2011). Analysis of virtualization technologies for high performance computing environments. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 9–16.