

Plataforma de áreas de trabalho virtuais escalável para nuvens privadas

Demis Gomes¹, Nichene Verçosa¹, Victor Medeiros¹, Glauco Gonçalves¹

¹Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco (UFRPE)
CEP 52.171-900 – Recife – PE – Brazil

{demis.gomes,nichene.vercosa,victor.wanderley,glauco.goncalves}@ufrpe.br

Abstract. *Virtual desktops platforms aim to centralize computing resources in a Cloud and allow users to access these resources through terminals. Such platforms can be a scalable and efficient solution for private clouds by reducing energy consumption and operating costs. We propose a platform that meets these requirements and is based on virtualized sessions, low-cost terminals, open-source software and cloud management algorithms. Finally, we conducted automatized experiments on the proposed platform, simulating a use case of the system, hence demonstrating the low memory consumption caused by the use of virtualised sessions and the high variation of network consumption and CPU.*

Resumo. *Plataformas de áreas de trabalho virtuais buscam centralizar os recursos computacionais em uma Nuvem, permitindo o acesso do usuário por meio de terminais. Tais plataformas podem ser uma solução escalável e eficiente para nuvens privadas, reduzindo o consumo de energia elétrica e dos custos operacionais. Propomos uma plataforma visando atender à estes requisitos baseada em sessões virtualizadas, terminais de baixo custo, software open-source e algoritmos de gerenciamento da Nuvem. Por último, realizamos experimentos com a plataforma proposta, simulando um caso de uso do sistema, demonstrando o baixo consumo de memória causado pelo uso de sessões virtualizadas e a alta variação do consumo de rede e de CPU.*

1. Introdução

Computadores convencionais ainda são a escolha mais comum em redes privadas de pequeno e médio porte, como laboratórios de informática, departamentos, pequenas empresas e *callcenters*. Nestes ambientes, porém, as atividades realizadas pelos usuários tendem a ser homogêneas, isto é, esses executam um conjunto previsível de aplicações na maior parte do tempo em que utilizam seu computador. Contudo, a padrão de uso de aplicações neste tipo de ambiente implicam em uma maior ociosidade dos recursos dos computadores quando comparado a outros cenários, sugerindo que o compartilhamento desses recursos computacionais potencializaria menor consumo de energia elétrica e diminuição dos custos.

Tal nível de compartilhamento de recursos pode ser alcançado por meio do conceito de área de trabalho remota, que prega a execução da área de trabalho, com aplicações e arquivos do usuário em um servidor remoto. A área de trabalho seria apenas mostrada ao usuário na tela de seu dispositivo. O controle da área de trabalho é feito pelos dispositivos de entrada, como mouse, tela de toque e teclado. Com o emprego deste conceito,

os dispositivos teriam a única função de acessar o servidor, tendo a possibilidade de adquirir equipamentos de menor capacidade computacional e menor consumo de energia elétrica, reduzindo o custo global da infraestrutura. Uma destas opções são os *thin clients* [Deboosere et al. 2012], dispositivos que têm apenas a função de acessar o servidor remoto, possibilitando reduzir o consumo de energia elétrica em cerca de 80% em relação aos computadores convencionais [Dasilva et al. 2012].

O conceito de área de trabalho remota também pode ser associado à Computação em Nuvem, oferecendo benefícios como escalabilidade, disponibilidade e gerenciamento simples. A Nuvem também pode auxiliar a reduzir os custos. Pesquisadores estimaram uma redução de 42% nos custos de operação caso universidades utilizassem a Nuvem em vez de adquirir computadores convencionais [Chandra and Borah 2012]. Desta forma, a área de trabalho remota busca prover uma redução de custos de implantação e de operação, reduzindo o tempo de instalação e atualização de novos softwares uma vez que o controle é centralizado, além de escalar o sistema conforme a demanda dos usuários.

Plataformas para acesso às áreas de trabalho virtuais remotas são consideradas um objeto de estudo tanto no aspecto conceitual quanto de sua implementação. Dentre os trabalhos que implementaram uma solução de plataforma de áreas de trabalho virtuais de acesso remoto, podemos destacar [Tuminauskas et al. 2012], [Yoo et al. 2012], [Junior et al. 2004] e [De Bona et al. 2008], indicando um interesse acadêmico e comercial em soluções baseadas na migração das áreas de trabalho do computador local para um servidor remoto. Para prover uma solução ainda mais eficaz, porém, é necessário verificar quais os fatores mais importantes para a criação desta plataforma, os softwares mais adequados para a realização desta tarefa e os possíveis gargalos de desempenho que podem ocorrer. Desta forma, o objetivo deste trabalho é apresentar uma solução eficiente, escalável e robusta de uma plataforma de áreas de trabalho virtual para nuvens privadas baseada em virtualização, uso de software livre e de ferramentas de gerenciamento de recursos computacionais e usuários. A partir de experimentos, também buscamos entender em que pontos a plataforma deve ser melhorada e, por meio de experimentos, mostramos o comportamento do sistema em diferentes condições de uso.

O artigo está dividido em cinco Seções. A segunda Seção discorre sobre os trabalhos relacionados que contribuíram para a motivação dos objetivos deste artigo. A Seção 3 apresenta a plataforma, discutindo os motivos que levaram a propô-la, suas características e funcionalidades, além de discutirmos o protótipo implementado, justificando os componentes de hardware e software escolhidos. A Seção 4 apresenta os experimentos de carga no sistema. Na última seção, concluímos o artigo e propomos futuros desenvolvimentos desta pesquisa.

2. Trabalhos relacionados

A utilização do conceito de área de trabalho remota remonta ao uso dos servidores de acesso remoto. Neste sentido, pode-se destacar iniciativas como o projeto Servidor de Estações de Trabalho [Junior et al. 2004] desenvolvido pela UFPA. O principal objetivo do projeto foi reduzir os custos com laboratórios de informática educacionais centralizando as atividades e aplicações do usuário em um servidor. Os computadores dos alunos teriam apenas a função de enviar os comandos através do teclado e do mouse e recebiam a área de trabalho, visualizando-a em seus monitores. O projeto utilizou computadores

convencionais obsoletos e sem HD como terminais, pois na época eram mais baratos que os *thin clients*, dispositivos específicos para acesso remoto. Ainda no contexto brasileiro e utilizando tecnologias similares, pode-se relatar a aplicação desenvolvida no Paraná, no projeto chamado Paraná Digital [Schmelzer et al. 2011]. Este projeto, liderado pela UFPR, teve como objetivo informatizar as escolas do estado através da criação de uma rede de alta velocidade interligando as escolas; do emprego de computadores multi-terminais (cerca de 4 terminais por CPU) para criação de laboratórios; e da utilização do Linux Educacional como sistema operacional padrão da solução, ajudando a reduzir custos de software proprietários. Cerca de 2100 escolas do estado participaram do projeto [De Bona et al. 2008].

Contudo, essas iniciativas se mostram limitadas quanto à escalabilidade e gerência de recursos físicos. Para alcançar tais propriedades faz-se necessário o uso de conceitos e ferramentas da virtualização e computação em nuvem. Neste sentido, a experiência na universidade Kaunas na Lituânia envolveu a virtualização de toda a infraestrutura computacional da universidade [Tuminauskas et al. 2012]. Nesta solução utilizou-se software Microsoft para hospedar máquinas virtuais em servidores que poderiam ser acessados a partir do site da universidade. Esta solução atendeu inclusive questões específicas como a possibilidade de executar aplicações antigas em sistemas operacionais mais recentes, virtualizando a aplicação com o sistema operacional compatível correspondente. Os pesquisadores apontaram como benefícios do sistema a atualização rápida e simples do software e a escalabilidade. Como problema, os autores mencionaram a falta de um gerenciamento eficiente de falhas, pois uma falha no servidor faria com que as máquinas virtuais hospedadas nele ficassem inacessíveis, afetando a satisfação do usuário.

Uma experiência semelhante também foi implantada em um hospital de Seul [Yoo et al. 2012]. Por meio da solução, pacientes e seus familiares acessam prontuários, exames de raios-x e informações sobre medicamentos através de tablets, diretamente dos servidores do hospital. Segundo os autores, os médicos, apesar da falta de afinidade com a tecnologia, acessaram e aprovaram o sistema. Futuramente, hospitais de menor porte tecnológico podem utilizar uma infraestrutura mais simples que permita o acesso de prontuários e exames de pacientes através do histórico armazenado no hospital, auxiliando os médicos a realizarem um acompanhamento mais detalhado do paciente. Os pacientes e seus familiares podem acessar exames de maneira mais simples, evitando a burocracia.

Nossa proposta tem como objetivo utilizar a virtualização de servidores para melhorar a escalabilidade, elasticidade e tolerância à falhas, algo que os trabalhos implementados em [Junior et al. 2004] e [Schmelzer et al. 2011] não utilizaram, e um baixo custo de implantação e manutenção com o uso de software livre, diferentemente dos trabalhos de [Tuminauskas et al. 2012] e [Yoo et al. 2012].

3. Plataforma proposta

Nesta Seção iremos inicialmente apresentar a plataforma de modo conceitual, definindo a arquitetura do sistema. Em seguida, detalharemos a criação do protótipo baseado na arquitetura proposta, informando quais componentes de hardware e software foram utilizados em cada camada.

3.1. Apresentação da plataforma

Uma plataforma de área de trabalho remota precisa prover requisitos como escalabilidade, redundância, disponibilidade, tolerância à falhas, administração facilitada e execução remota de aplicações. Tais requisitos podem ser atendidos de forma eficiente com o auxílio da virtualização, de conceitos relacionados à gerência da Nuvem e de algoritmos com o intuito de utilizar os recursos da Nuvem economizando energia. A virtualização, aliada a gerência da Nuvem, facilita a administração do sistema permitindo a geração de cópias de servidores virtuais e o fácil armazenamento dos estados das máquinas virtuais (*Virtual Machines* - VMs). O servidor virtual que hospeda a área de trabalho do usuário é chamado de servidor de aplicações. Este servidor tem a função de executar a área de trabalho e as aplicações do usuário. Na plataforma, vários deles são utilizados em conjunto para prover uma melhor qualidade de experiência ao usuário.

Um importante aspecto na especificação de uma plataforma de área de trabalho remota é a escolha da forma de virtualização da área de trabalho. Neste caso, há duas opções: sessão virtualizada e VDI (*Virtual Desktop Infrastructure*). Na sessão virtualizada, um servidor permite o acesso de vários usuários a partir de sessões remotas no sistema operacional. Uma vez que cada servidor é uma VM, esta VM é portanto compartilhada entre diversos usuários. Na VDI, há a criação de uma máquina virtual para cada usuário, ou seja, cada usuário tem total domínio sobre sua VM [Dernbecher et al. 2013].

Em nosso cenário, escolhemos a sessão virtualizada devido a fatores como desempenho e gerenciamento. As sessões fazem melhor uso dos recursos dos servidores e são mais fáceis de gerenciar [Shabaitah 2014], pois aplicações podem ser instaladas apenas uma vez no servidor de aplicações designado, estando disponível para todos assim que instalada. Na VDI, tal tarefa é feita de modo mais complexa. Uma solução seria criar uma imagem raiz de uma VM e imagens filhas que herdam as configurações da raiz e são distribuídas para os clientes. Quando uma mudança é feita na raiz, deve-se atualizar todas as imagens filhas dos clientes. A implementação de uma solução de área de trabalho remota utilizando sessões também é mais simples, uma vez que menos etapas são necessárias para a instalação de todos os componentes requeridos em relação à VDI. O menor consumo de recursos computacionais - que potencializa o menor consumo de energia elétrica - e a simplicidade provida por esta abordagem são aspectos que se adequam a cenários que contém usuários de comportamento homogêneo, mantendo a linha inicial da nossa proposta.

A VDI tem benefícios como um maior nível de customização da área de trabalho, em que até mesmo parâmetros do sistema operacional podem ser ajustados para se adequar às necessidades de um usuário, e a maior disponibilidade de ferramentas de gerência de VMs na nuvem [Endo et al. 2010]. A Tabela 1 resume as principais características das duas formas de virtualização de áreas de trabalho.

A arquitetura proposta neste trabalho é mostrada na Figura 1. Por meio dessa arquitetura pretende-se permitir que os clientes tenham acesso às características operacionais do sistema, através de um conjunto de serviços gerenciados na nuvem. Os serviços serão detalhados a seguir.

O terminal, também chamado de cliente, é um dispositivo que acessa o servidor e mostra ao usuário apenas a tela de sua área de trabalho. Esse dispositivo tem a função de

Características	VDI	Sessão
Consumo de recursos	Maior	Menor
Custo	Maior	Menor
Administração	Mais complexa	Mais simples
Disponibilidade de ferramentas	Maior	Menor
Customização do usuário	Melhor	Pior

Tabela 1: Comparação entre VDI e sessão virtualizada



Figura 1: Arquitetura da plataforma

enviar ao servidor por meio de protocolos específicos, os comandos referentes as ações do usuário, que são recebidas através da interação do usuário com os dispositivos de entrada como mouse, tela de toque e teclado. Opções como os *thin clients* são criadas especificamente para o acesso remoto à servidores, com baixo custo de aquisição e de consumo de energia e com a única função de acessar servidores remotos através de protocolos de acesso. Contudo, dispositivos como *smartphones* e *tablets* podem ser utilizados complementarmente.

O Serviço de Desktop Virtual é o principal serviço do sistema, que oferece uma área de trabalho virtual hospedada em uma máquina virtual na plataforma. Este serviço transmite a tela remotamente do servidor ao terminal, através de protocolos de acesso específicos. Tais protocolos podem ser classificados de duas maneiras distintas: protocolos baseados em objetos e protocolos baseados em *pixels*.

No primeiro caso, cada aplicação executando na VM envia requisições para que o dispositivo desenhe objetos gráficos como botões, barras de rolagem e outros, diretamente na tela do usuário. A principal vantagem desta abordagem é o baixo consumo da capacidade da rede já que as requisições enviadas são simples indicações de posição na tela. Entretanto, o dispositivo do usuário precisa ser mais robusto em termos de software e hardware, uma vez que é preciso uma conversão rápida dos comandos recebidos da aplicação em telas para o usuário. X11¹, RDP (*Remote Desktop Protocol*)² e HDX³ são exemplos de protocolos que compartilham esta característica.

Os protocolos baseados em *pixels* enviam imagens da área de trabalho de um servidor para um cliente, como se o sistema operacional enviasse capturas de tela ao dis-

¹RFC X11 - <https://tools.ietf.org/html/rfc1198>

²RFC RDP - <https://tools.ietf.org/html/rfc1151>

³Citrix HDX - https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/citrix-hdx-technologies.pdf

positivo do usuário. Por depender da qualidade da imagem e enviar sucessivas imagens, esta transmissão requer boa capacidade de rede. Contudo, o cliente não precisa possuir um hardware ou software tão robusto, pois não é necessário converter requisições como no exemplo anterior. PCoIP (*Personal Computer-over-Internet Protocol*)⁴ e RFB (*Remote Framebuffer*)⁵ são alguns exemplos de protocolos deste tipo. A plataforma deve se adaptar com a conexão de rede do usuário, transmitindo a tela com protocolos mais robustos quando a conexão estiver estável e com alta largura de banda, e utilizar protocolos mais leves nos momentos em que a conexão estiver congestionada. Assim, utilizando protocolos mais leves, a latência é diminuída quando a conexão estiver congestionada, melhorando a experiência do usuário uma vez que a conexão cliente-servidor depende da qualidade da rede.

O Serviço de Administração consiste na gerência de usuários de modo simples e centralizado, a fim de viabilizar o controle dos administradores do sistema. A gerência de usuários se caracteriza pela criação, edição e remoção de usuários. Deste modo, os administradores do sistema podem gerenciar usuários em um único local, evitando o trabalho repetitivo de criar, editar ou remover usuários em cada computador da infraestrutura. Além disso, o administrador pode atribuir uma política de uso do sistema que inclui permissões específicas para grupos de usuários, provendo acesso a arquivos, sites, aplicações ou diretórios apenas para determinados grupos. A gerência de servidores virtuais também é necessária para criar servidores de determinadas configurações de CPU, memória e armazenamento de forma facilitada.

O Serviço de Boot Remoto compreende a entrega do IP aos terminais menos capazes por meio do DHCP (*Dynamic Host Configuration Protocol*), o *boot* remoto que geralmente ocorre através do protocolo PXE [Cruz et al. 2010] (*Preboot Execution Environment*) e a escolha de qual dispositivo os recursos serão utilizados, ou seja, se o terminal utilizará seus recursos ou apenas acessará o servidor. O PXE carrega uma imagem de inicialização do sistema que é armazenada no terminal, e dependendo da capacidade do terminal e dos recursos disponíveis no servidor, esta imagem de inicialização pode ser apenas um *boot* mínimo, com o terminal acessando o servidor após a autenticação, ou um *boot* de um sistema operacional mais completo, no qual o terminal utiliza os seus recursos para executar o sistema, com a plataforma gerenciando os arquivos e as aplicações do usuário.

A gerência da nuvem tem as funções de balanceamento de carga, migração e eficiência de consumo de energia. Em uma plataforma de consumo eficiente de recursos, é necessário a utilização de algoritmos adequados para estas funções a fim de manter um bom desempenho do sistema e utilizar a capacidade dos servidores físicos de maneira mais eficiente. O balanceamento de carga [Gonçalves et al. 2013], [Palhares et al. 2014] permite que cada usuário possa ser alocado a um servidor mais ocioso, oferecendo melhor desempenho. Caso um novo servidor virtual seja necessário, um algoritmo pode verificar os servidores menos carregados para que a VM seja alocada. O consumo de energia pode ser reduzido caso a plataforma execute algoritmos de consolidação que percebam a possibilidade de desligamento automático de um servidor físico ocioso sem perda de desempenho [Deboosere et al. 2012]. Neste caso, as VMs contidas neste servidor seriam migradas para servidores ociosos.

⁴Teradici PCoIP - <http://www.teradici.com/pcoip-technology>

⁵RFC RFB - <https://tools.ietf.org/rfc/rfc6143.txt>

A virtualização auxilia a escalabilidade, a administração e o backup dos dados, uma vez que máquinas virtuais de diferentes configurações são criadas rapidamente e podem ter seu estado salvo a qualquer momento. A flexibilidade proporcionada pela virtualização também permite que usuários acessem diferentes tipos de sistemas operacionais em diferentes configurações de RAM, CPU e capacidade de HD. Os usuários podem acessar o seu *desktop* na Nuvem através de uma máquina virtual dedicada a ele ou em uma sessão remota em um servidor virtual. O *hypervisor*, ou VMM (*Virtual Machine Monitor*), é a camada que se situa entre o hardware e a máquina virtual, alocando recursos da VM no hardware. O *hypervisor* tem funções como ligar, desligar e reiniciar máquinas, além de salvar o estado atual da máquina virtual (*snapshot*) e clonar máquinas virtuais.

A plataforma deve prover um sistema eficiente e escalável para o armazenamento das imagens das máquinas virtuais e dos dados dos usuários. A utilização de um backup dos dados, que pode ser em outro sistema externo à plataforma, é essencial para a redundância de dados caso um servidor falhe, impedindo que o usuário tenha seus arquivos e aplicações perdidas. A taxa de I/O (entrada/saída) de disco deve ser alta neste serviço, pois o armazenamento é um dos principais pontos de gargalo de um sistema de acesso remoto [Kibe et al. 2011].

Por último, os servidores e dispositivos de rede correspondem aos equipamentos físicos que hospedam os serviços. Os servidores estarão conectados por *switches* e roteadores, e nestes servidores serão executados programas de gerenciamento e compartilhamento de recursos. Como os servidores físicos hospedarão servidores virtuais através dos *hypervisors*, técnicas de otimização dos *hypervisors* devem ser realizadas a fim de mitigar a perda de desempenho causada pelo uso de máquinas virtuais, algo que sobrecarrega o hardware.

3.2. Protótipo

A Figura 2 define os componentes de hardware e software a serem utilizados em cada camada definida na Subseção 3.1.

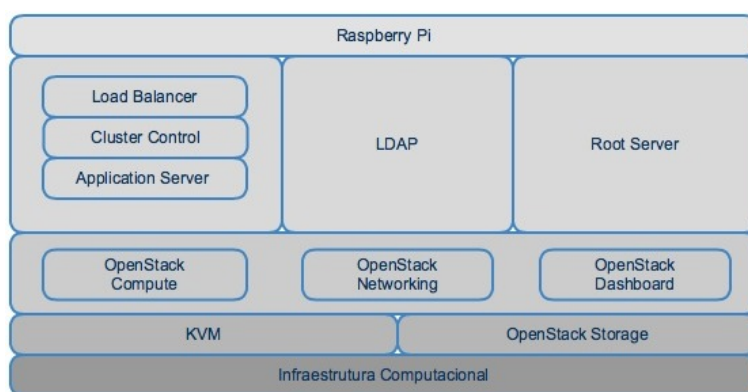


Figura 2: Protótipo

O terminal adotado para concepção do protótipo foi a placa Raspberry Pi. Tal placa foi escolhida devido a sua alta adaptabilidade como *thin client*, ao mesmo passo que se encontra atualmente como uma das opções mais baratas de implementação entre outros *thin clients*. A terceira edição da placa, chamada Raspberry Pi 2, foi lançada em fevereiro de 2015 e conta com um processador de quatro núcleos de 900 MHz e 1GB

de memória RAM. O sistema operacional utilizado foi o Raspbian, sistema operacional oficial da Raspberry Pi Foundation ⁶, que possui simples implementação e versatilidade.

Em termos de *desktop* virtual, o LTSP Cluster⁷ foi utilizado para a comunicação entre clientes e servidores com o protocolo de acesso X11. Na documentação da ferramenta, é indicada a criação de quatro servidores: *Root Server*, *Cluster Control Server*, *Load Balancer Server* e *Application Server*. O *Root Server* tem a função de ser o servidor DHCP e armazenar a imagem mínima de inicialização enviada ao cliente utilizando o TFTP (*Trivial File Transfer Protocol*)⁸. O *Cluster Control* configura o LTSP Cluster, oferecendo uma interface web para facilitar a administração de toda a infraestrutura, além de criar o arquivo com a configuração necessária para o cliente iniciar, contendo informações como o *Application Server* definido para o usuário, linguagem do sistema e do teclado. O *Load Balancer* verifica qual *Application Server* está em melhores condições de atender as requisições do próximo cliente, verificando status de memória e CPU de todos os *Application Servers*. O *Application Server* é responsável por executar as aplicações dos usuários. O número de *Application Servers* pode crescer a partir da demanda desejada.

A princípio, mantivemos a configuração padrão do LTSP Cluster com quatro servidores, fazendo do *Cluster Control Server* o servidor LDAP (*Lightweight Directory Access Protocol*)⁹ para a autenticação, criação e remoção de usuários de maneira centralizada. Apesar da possibilidade de criarmos vários *Application Servers* no Serviço de Desktop Virtual, em nossa avaliação iremos utilizar apenas um *Application Server* para medir o impacto dos clientes neste servidor. Neste caso de apenas um único *Application Server*, foi verificado que a melhor utilização da máquina que hospeda tais servidores virtuais seria concretizada através da diminuição da quantidade de servidores. Desta forma, agregamos os serviços oferecidos pelo *Cluster Control* e o *Load Balancer* junto ao *Root Server*. Destaca-se que esta alteração só deve ser efetuada para a conformação com poucas unidades de *Application Server*, como ocorre em nossa avaliação. No cenário com número significativo de *Application Servers*, a melhor utilização dos recursos se dá através da recomendação original da documentação do LTSP cluster. Deste modo, em nossa arquitetura, o Serviço de Administração é feito pelo LDAP em conjunto com o *Cluster Control*, enquanto que o Serviço de *Boot* Remoto é atribuído ao *Root Server*.

Como gerenciador de nuvem é recomendado a utilização do OpenStack¹⁰. O OpenStack é um sistema operacional em nuvem dividido em três componentes: *Compute*, *Networking* e *Storage*. A gerência é feita por meio do *Dashboard* através de uma interface web, facilitando a automatização de vários processos e possibilitando a construção de ferramentas para gerenciar os recursos. O OpenStack Compute provê e gerencia redes de máquinas virtuais escaláveis e elásticas. O Networking é um sistema para o gerenciamento de redes e endereços IP. O Storage permite armazenamento com requisitos diferentes e bastante desempenho, inclusive em cluster.

Os servidores virtuais são hospedados através do KVM (*Kernel-based Virtual Machine*), que é um módulo do *kernel* do Linux que permite a virtualização completa na arquitetura x86. O KVM ao contrário de outros *hypervisors*, concentra-se na

⁶Raspbian for Raspberry Pi - <https://www.raspberrypi.org/downloads/raspbian/>

⁷LTSP-Cluster - <https://www.ltsp-cluster.org/>

⁸RFC TFTP - <https://www.ietf.org/rfc/rfc1350.txt>

⁹RFC LDAP - <https://tools.ietf.org/html/rfc4511>

¹⁰OpenStack - <https://www.openstack.org/>

implementação de gestão do hóspede da virtualização, permitindo que o *kernel* do Linux opere como o *hypervisor* [Opsahl 2013]. A comunicação do OpenStack com o KVM é direta, mantendo a coesão entre os componentes da plataforma. Em estudos prévios, ([Gomes et al. 2015]), pudemos corroborar os resultados de [Opsahl 2013], observando que, para ambientes de áreas de trabalhos virtuais, o KVM apresentou melhor *hypervisor* desempenho em termos de CPU e por isto foi selecionado para a plataforma.

4. Experimentos de avaliação de carga

Os experimentos realizados neste trabalho avaliam apenas o serviço que é núcleo da plataforma proposta: o serviço de áreas de trabalho virtuais. A avaliação conduzida foca na medição do desempenho de um único *Application Server*, por isso o ambiente de avaliação utilizado não conta com o uso do OpenStack. A função de servidor de arquivos foi executada pelo NFS (*Network File System*), que tem como propósito montar o diretório básico do usuário (*/home*) no *Application Server*. O NFS foi utilizado neste ambiente de teste pela sua simplicidade. Com relação à infraestrutura computacional, a parcela referente à rede necessita de duas interfaces distintas, uma para a gerência e outra para a aplicação. A interface de rede de gerência é o meio pelo qual o sistema operacional e o LTSP Cluster se comunicam. Já a interface de rede de aplicação contém o tráfego entre os *desktop* virtuais e os clientes.

Para a realização dos experimentos são definidos dois tipos de clientes, baseado em [Shabaitah 2014]: leve e pesado. A categorização dos perfis foi baseada nas aplicações que um usuário típico da categoria utiliza e na carga que exerce no servidor. O cliente do tipo leve acessa dois sites no Mozilla Firefox, executa o LibreOffice Writer, escreve um parágrafo de três linhas em um novo documento, abre uma nova planilha no LibreOffice Calc, edita catorze células e abre sua pasta *home*. O cliente pesado acessa um site no Firefox, abre um arquivo de 1,2 MB no LibreOffice Writer e edita um parágrafo de três linhas, abre uma planilha de 256 KB e edita catorze células, abre uma apresentação de 700 KB, acessa outro site, executa o gcalctool e abre um documento PDF de 3,2 MB.

A ação dos clientes de cada tipo é automatizada por meio de uma ferramenta de emulação de clientes desenvolvida exclusivamente para este trabalho. A carga no servidor é medida pelo software *dstat*, ferramenta que captura o estado do sistema por meio de diferentes métricas que são analisadas após a execução do experimento. Como métricas, são coletados o consumo de memória, a ocupação de CPU e a vazão de dados de saída e entrada do servidor de áreas de trabalho.

Para analisar a carga no servidor com o aumento do número de terminais, executamos experimentos em um computador de 64 GB de RAM, com dois processadores Intel Xeon de 6 cores cada e frequência de 2.40 GHz e sistema operacional Linux Debian. Nele são executadas diversas máquinas virtuais, divididas em clientes e servidores virtualizados através do *hypervisor* KVM. Os terminais são emulados por meio de VMs com 256 MB de RAM, e 1 CPU virtual (vCPU). O *Root Server* possui 2 GB de RAM, e 4 vCPUs, enquanto que o *Application Server* possui 20 GB de RAM e 20 CPUs virtuais. Ambos os servidores executam o Ubuntu 12.04. O Root Server é configurado com o LDAP e o NFS para operações de autenticação e armazenamento, respectivamente.

As VMs que emulam os terminais se comunicam com os servidores através de uma *bridge* virtual, executam o boot remoto através do PXE, recebem a imagem de

inicialização do sistema do *Root Server* e executam suas aplicações no *Application Server*, como mostra a Figura 3. Nosso experimento executa os clientes leves e pesados e utiliza dois fatores: o número de terminais, variando entre 5, 10, 15, 20, 25, 30, 35 e 40 terminais, e a porcentagem de clientes do tipo pesado que varia entre 0%, 25%, 50%, 75% e 100%. Um experimento é realizado a partir da junção destes parâmetros, e cada experimento é realizado cinco vezes.

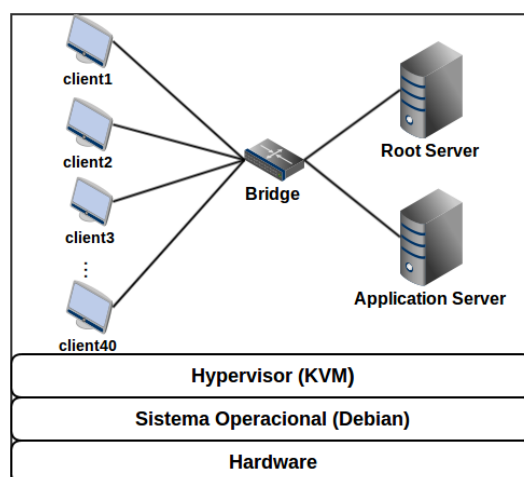


Figura 3: Ambiente de execução dos experimentos na máquina física de 64 GB de RAM e dois processadores Xeon de 6 cores. A estrutura foi virtualizada no hypervisor KVM.

Em relação ao consumo de memória, como mostra a Figura 4, o crescimento do uso deste recurso apresenta um comportamento linear com o número e o tipo dos clientes. Neste caso, o sistema com 40 máquinas executando todos os clientes pesados utiliza no máximo pouco mais de 80% da memória do *Application Server*, cerca de 17 GB. Aqui deve-se destacar uma das vantagens do emprego de sessões virtualizadas frente ao VDI. Trabalhos anteriores mostram que cada cliente em um sistema de VDI aloca entre 0,5 GB e 2,5 GB de memória ([Jo and Egger 2013], [Uehara 2014], [Tuminauskas et al. 2012]), assim um ambiente utilizando VDI para o atendimento de 40 clientes teria um consumo de memória entre 20 e 100 GB, mais alto do que no ambiente utilizando sessões. Além disso, o consumo de 80% no *Application Server* indica que, mesmo com 40 clientes pesados, o servidor provavelmente não sofrerá gargalos de desempenho em relação ao consumo de memória, uma vez que é recomendado que o sistema atenda às requisições sem gargalos de desempenho quando o consumo máximo de memória situa-se entre 70% e 80% [Shabaitah 2014].

Na taxa de dados enviados e recebidos da rede, verificamos os valores máximos e o 75 percentil. A taxa de saída de dados de rede geralmente corresponde a transferência da tela do usuário do servidor ao terminal e a comunicação entre *Root Server* e *Application Server* para a sincronização do NFS, enquanto que os dados de entrada são relacionados a comunicação com o *Root Server* para a sincronização do NFS e os comandos enviados pelos clientes. Como mostra a Figura 5(a), os valores máximos de entrada chegam a ficar entre 300 e 400 Mbps, e a taxa de saída fica em torno de 900 Mbps. Apesar destes valores, o 75 percentil (Figura 5(b)) mostra que todos os valores estão abaixo dos 100 Mbps, ou seja, em 75% do tempo, os valores ficaram abaixo dos 100 Mbps, tornando viável a solução quando os clientes estão em redes Fast-Ethernet, de 100 Mbps. A interface de

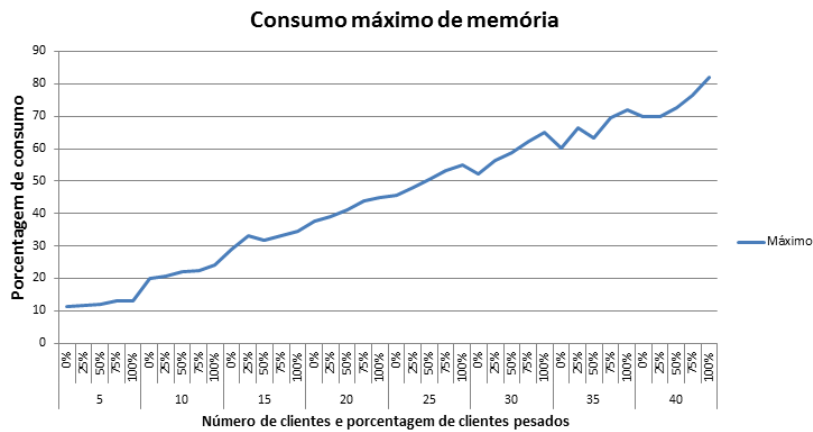


Figura 4: Consumo máximo de memória dos experimentos realizados

rede do *Application Server*, porém, deve ser Gigabit, para comportar os picos dos valores das amostras e a alta comunicação entre Root e *Application Server*.

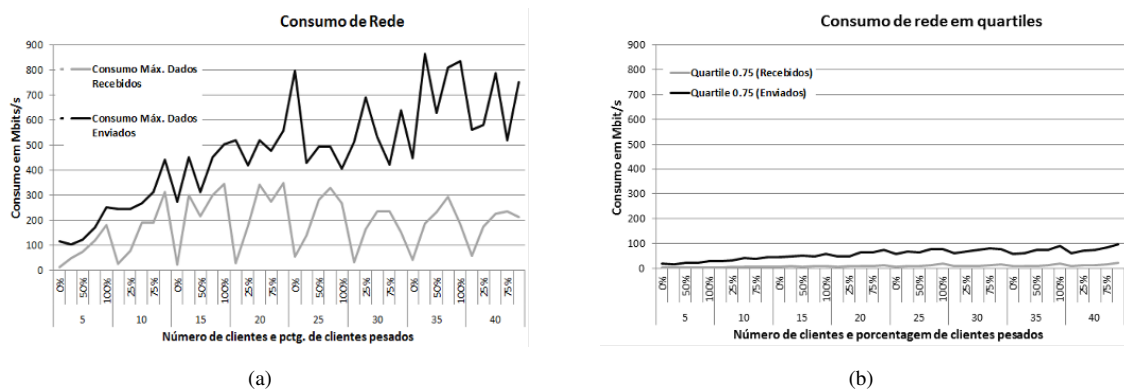


Figura 5: Consumo de rede máximo no quartil 0.75

Em relação ao consumo de CPU, diferentes análises foram feitas devido a variabilidade destes dados. A análise dos valores foi realizada com base no pico sustentado. Esta análise consiste em reter apenas os valores das amostras que ultrapassaram um limiar por um determinado período de tempo. No nosso caso, fixamos o limiar em 95%, pois neste valor de consumo de CPU o *Application Server* não responde, levando o usuário a esperar o consumo diminuir para prosseguir com suas ações. A partir de [Tolia et al. 2006], o qual apresenta estudos do impacto do tempo de resposta de ações de usuários, é relatado que uma aplicação torna-se inutilizável quando o tempo de resposta é maior que cinco segundos. Desta forma, o gráfico da Figura 6 é preenchido com o tempo durante o qual a ocupação de CPU ultrapassou 95% por 5 segundos ou mais.

Analizamos o impacto gerado pela variação da porcentagem de clientes leves e pesados. O gráfico na Figura 6 mostra que, até 25 clientes, o sistema mostrou-se responsivo na maior parte do tempo, com apenas 2% do tempo não responsivo. A partir de 30 clientes, os resultados demonstram que houve aumentos significativos, principalmente nos casos em que os clientes do tipo pesado são maioria - quando a porcentagem destes clientes está em 75% ou 100%. Acreditamos que, quando a porcentagem acima do pico sustentado com limiar 95% e com cinco ocorrências estiver em torno de 2% do tempo,

o sistema ainda pode ser utilizável devido a baixa ocorrência destes valores. Acima de 2% a situação fica mais instável, levando os usuários a ficarem mais decepcionados com o atraso de resposta da plataforma.

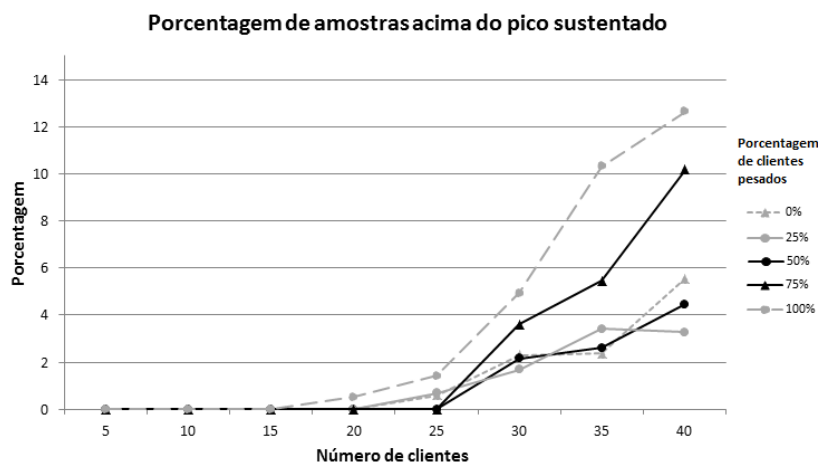


Figura 6: Porcentagem do tempo em que as amostras ficaram acima do pico sustentado

Como esperado, nossos experimentos demonstraram que os clientes pesados utilizam mais CPU do que os leves. Seguindo nossa abordagem de que o sistema é utilizável quando as amostras acima do pico sustentado estão em torno dos 2%, o sistema oferece um desempenho aceitável quando o número de clientes leves é 30 e o de clientes pesados é 25. Desta forma, a plataforma pode se adaptar com relação ao número de clientes de cada tipo, criando um novo *Application Server* e disponibilizando-o no sistema a partir da demanda. Com 40 clientes pesados, o servidor não respondeu durante 12% do período de execução, tempo considerado alto.

O consumo de CPU não se comportou de forma esperada em todos os casos, como por exemplo, no caso com 25% dos clientes pesados, há uma queda de consumo de CPU mesmo com o aumento de 35 para 40 clientes, e, no caso com 40 clientes, o consumo de CPU com todos os clientes leves (0%) é maior do que com 25% ou 50% de clientes pesados. Isto ocorre porque, os clientes leves concluem as atividades definidas na ferramenta de emulação de clientes entre 20% e 30% mais rápido do que os clientes pesados. Desta forma, quando o número de clientes pesados é menor ou igual ao dos clientes leves, na maioria das vezes, os clientes leves terminam o experimento primeiro e diminuem a carga no sistema. Esta diminuição é menos aparente quando a maioria dos clientes são pesados, pois esses geram maior carga no sistema. Além disso, estas variações também são impactadas pela execução das VMs dos clientes na mesma máquina física dos servidores virtuais. Diferentemente da memória, cujo isolamento entre VMs é forte, a CPU é compartilhada entre elas. Desta forma, quanto mais clientes são executados, maior a ocupação de CPU, aumentando e variando ainda mais o consumo deste recursos no *Application Server*.

5. Conclusão

A plataforma de área de trabalho virtuais é uma alternativa para sistemas homogêneos, nos quais os usuários executam tarefas semelhantes como em *callcenters*, laboratórios de informática e pequenas empresas. A solução proposta visa melhorar a eficiência dos

sistemas atuais, reduzindo os custos referentes à implantação e manutenção do sistema com o emprego de terminais mais leves e a centralização das atividades na Nuvem, tornando o gerenciamento de usuários e de máquinas mais simples e robusto. Algoritmos de consolidação propostos pela academia podem ser utilizados na plataforma para reduzir o consumo de energia, e o balanceamento de carga aliado a migração de máquinas virtuais é uma abordagem fundamental para melhorar a experiência do usuário, evitando a má qualidade do serviço causado por servidores sobrecarregados e migrando VMs de servidores com alta carga de trabalho para servidores mais ociosos.

Analisamos os tipos de acesso dos clientes aos servidores e concluímos que a melhor opção para nosso objetivo seria a sessão virtualizada. Realizamos experimentos que verificavam a carga exercida em um serviço de áreas de trabalho virtuais em medições com número variável de clientes, no qual observamos o alto consumo de CPU e sua grande variação, a linearidade do consumo de memória e os picos de consumo de rede existentes. Apesar da variabilidade dos dados de CPU, verificamos que em 12% do tempo, o sistema fica inutilizável quando 40 clientes eram emulados, porém quando submetido à carga de até 25 clientes, o tempo em que o sistema fica inutilizável é menor que 2%. Deste modo, o sistema atenderia bem as requisições de 25 usuários com servidores de configuração semelhantes ao do experimento.

Nos próximos trabalhos, pretendemos avaliar proceder uma análise minuciosa dos tempos de execução de cada atividade dos clientes leve e pesado, com o intuito de levantar aspectos de qualidade de experiência além dos aspectos de qualidade de serviço apresentados. Além disso, pretendemos avaliar como configurações de hardware impactam no sistema. Um exemplo de configuração que pode melhorar o desempenho desta solução é o mapeamento de cores ("pinagem"), que permite associar cada core de uma VM com um dado core físico do processador. Desta forma, pode-se diminuir a latência de comunicação entre processos da VM mapeando os cores virtuais em determinados cores físicos do processador que potencializem o uso de barramentos mais rápidos.

6. Agradecimentos

Agradecemos à FACEPE e à UFRPE pelo apoio na forma de Bolsa de Iniciação Científica através do projeto Desenvolvimento de Plataforma de DaaS para Laboratórios de Ensino de Informática.

Referências

- Chandra, D. G. and Borah, M. D. (2012). Cost benefit analysis of cloud computing in education. In *Computing, Communication and Applications (ICCCA), 2012 International Conference on*, pages 1–6.
- Cruz, T., Simões, P., Bastos, F., and Monteiro, E. (2010). Integration of PXE-based desktop solutions into broadband access networks. In *2010 International Conference on Network and Service Management (CNSM)*, pages 182–189.
- Dasilva, D.-A., Liu, L., Bessis, N., and Zhan, Y. (2012). Enabling Green IT through Building a Virtual Desktop Infrastructure. In *2012 Eighth International Conference on Semantics, Knowledge and Grids (SKG)*, pages 32–38.
- De Bona, L. C., Castilho, M., Silva, F., Weingartner, D., Lourenço, L. H., and Ribas, B. C. (2008). Managing a grid of computer laboratories for educational purposes. page 6.

- Deboosere, L., Vankeirsbilck, B., Simoens, P., De Turck, F., Dhoedt, B., and Demeester, P. (2012). Cloud-Based Desktop Services for Thin Clients. *IEEE Internet Computing*, 16(6):60–67.
- Dernbecher, S., Beck, R., and Toenker, M. (2013). Cloudifying Desktops—A Taxonomy for Desktop Virtualization.
- Endo, P. T., Gonçalves, G. E., Kelner, J., and Sadok, D. (2010). A survey on open-source cloud computing solutions. *Brazilian Symposium on Computer Networks and Distributed Systems*, pages 3–16.
- Gomes, D., Verçosa, N., Carvalho, D., Medeiros, V., and Gonçalves, G. (2015). Avaliação de desempenho entre kvm e virtualbox em um ambiente de desktop-as-a-service.
- Gonçalves, G. E., Endo, P. T., Palhares, A. A., Santos, M. A., Kelner, J., and Sadok, D. (2013). On the load balancing of virtual networks in distributed clouds. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 625–631. ACM.
- Jo, C. and Egger, B. (2013). Optimizing Live Migration for Virtual Desktop Clouds. pages 104–111. IEEE.
- Junior, R. V., Favero, E. L., and Gonçalves, G. E. (2004). SET-uma solução de computador popular de rede para laboratórios de informática.
- Kibe, S., Uehara, M., and Yamagiwa, M. (2011). Evaluation of Bottlenecks in an Educational Cloud Environment. pages 520–525.
- Opsahl, J. M. G. (2013). Open-source virtualization: Functionality and performance of Qemu/KVM, Xen, Libvirt and VirtualBox.
- Palhares, A., Santos, M., Endo, P., Vitalino, J., Rodrigues, M., Goncalves, G., Sadok, D., Sefidcon, A., and Wuhib, F. (2014). Joint allocation of nodes and links with load balancing in network virtualization. In *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*, pages 148–155. IEEE.
- Schmelzer, S., von Suchodoletz, D., Schneider, G., Weingaertner, D., De Bona, L. C. E., and Carvalho, C. (2011). Universal remote boot and administration service. In *Network Operations and Management Symposium (LANOMS), 2011 7th Latin American*, pages 1–6. IEEE.
- Shabaitah, A. R. (2014). Server-Based Desktop Virtualization.
- Tolia, N., Andersen, D., and Satyanarayanan, M. (2006). Quantifying Interactive User Experience on Thin Clients. *Computer*, 39(3):46–52.
- Tuminauskas, R., Ambraziene, D., Miseviciene, R., and Pazereckas, N. (2012). Educational Infrastructure Using Virtualization Technologies: Experience at Kaunas University of Technology. *Informatics in Education-An International Journal*, (Vol11_2):227–240.
- Uehara, M. (2014). Supersaturated Cloud for Multicore Systems. pages 845–849. IEEE.
- Yoo, S., Kim, S., Kim, T., Kim, J. S., Baek, R.-M., Suh, C. S., Chung, C. Y., and Hwang, H. (2012). Implementation Issues of Virtual Desktop Infrastructure and Its Case Study for a Physician’s Round at Seoul National University Bundang Hospital. *Healthcare Informatics Research*, 18(4):259.