

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MARCIO RODRIGO FALVO

**UMA AVALIAÇÃO EXPERIMENTAL DO USO
DE DESKTOPS VIRTUAIS**

ORIENTADOR: PROF. DR. HERMES SENGER

São Carlos – SP

Janeiro/2014

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MARCIO RODRIGO FALVO

**UMA AVALIAÇÃO EXPERIMENTAL DO USO
DE DESKTOPS VIRTUAIS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Sistemas Distribuídos e Redes

Orientador: Prof. Dr. Hermes Senger

São Carlos – SP

Janeiro/2014

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

F197ae Falvo, Marcio Rodrigo.
Uma avaliação experimental do uso de desktops virtuais /
Marcio Rodrigo Falvo. -- São Carlos : UFSCar, 2014.
66 f.

Dissertação (Mestrado) -- Universidade Federal de São
Carlos, 2014.

1. Sistemas distribuídos. 2. Virtualização. 3. Desktop
Virtual. 4. VDI. I. Título.

CDD: 005.43 (20^a)

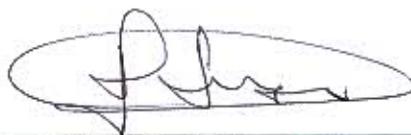
UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
Programa de Pós-Graduação em Ciência da Computação

**“Uma Avaliação Experimental do Uso de
Desktops Virtuais”**

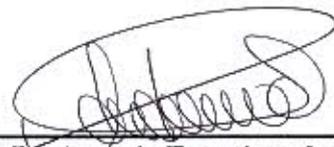
MARCIO RODRIGO FALVO

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

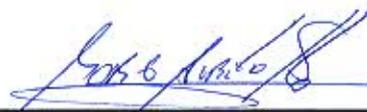
Membros da Banca:



Prof. Dr. Hermes Senger
(Orientador – DC/UFSCar)



Prof. Dr. Antonio Francisco do Prado
(DC/UFSCar)



Prof. Dr. Marco Aurélio Stelmar Netto
(IBM Research)

São Carlos - SP
Fevereiro/2014

À toda minha família.

AGRADECIMENTOS

Gostaria de agradecer a minha família pelo apoio e compreensão pelos meus momentos de ausência. Ao meu orientador pela paciência e por acreditar em mim. A diretoria e colegas de trabalho da Secretaria Geral de Informática pela ajuda vinda de várias maneiras, desde uma simples conversa para motivação até a ajuda técnica. Por fim, agradeço a Deus por ter vivido está experiência e conhecido novos amigos. Nada é por acaso.

"I gotta keep breathing. Because tomorrow the sun will rise. Who knows what the tide could bring?"

Nafrago (2000)

RESUMO

Um conjunto de servidores conectados à Internet pode prover *desktops virtuais* em um ambiente virtual de computação. Cada usuário pode ter seu próprio desktop virtual, acessando através da Internet por meio de dispositivos como um computador comum ou dispositivos móveis, como um notebook, *tablet* ou celular. Dentre os benefícios da virtualização de *desktops*, incluem-se a melhoria na utilização dos recursos de computacionais como processador, memória, disco e rede, a redução de custos para a administração dos recursos (que passa a atuar de forma centralizada e mais eficiente), backup centralizado e possibilidade de acesso remoto, onde quer que o usuário esteja. Muitas empresas já vem adotando o uso de desktops virtuais no ambiente de trabalho dos seus funcionários e colaboradores. Entretanto, as instituições de ensino tradicionalmente utilizam *desktops* locais, tanto para uso do corpo docente, corpo administrativo e técnico, como também para os seus alunos.

Como o tema é bem pouco explorado no ambiente de universidades e instituições de ensino superior, o presente trabalho tem como objetivo principal responder algumas questões sobre a utilização de desktops virtuais. O que se pode esperar em termos da qualidade de experiência dos usuários de desktops virtuais em função de sua execução remota? Quais aplicações típicas apresentam maior ou menor degradação em termos de qualidade de experiência à medida que alguns parâmetros da rede (como por exemplo a latência) se degradam? Qual é o consumo de recursos (processador, memória, E/S e rede) para a execução de desktops virtuais? Como principais contribuições, este trabalho apresenta uma avaliação experimental de uma infraestrutura de *desktops* virtuais implementada na Secretaria de Informática da UFSCar. Foram realizados experimentos para avaliar a qualidade de experiência do usuário com um conjunto de cinco aplicações típicas de um laboratório de ensino da universidade. Foram testados seis níveis de qualidade de serviço da rede para medir o seu impacto na qualidade de experiência dos usuários. Também foram feitos testes variando-se a quantidade de *desktops* virtuais executando simultaneamente para verificar o consumo de recursos, que servirá de base para um futuro planejamento de capacidade.

Palavras-chave: Virtualização; Desktop Virtual; Desktop Remoto; VDI; DaaS

ABSTRACT

A server cluster connected to the Internet can support virtual desktops in a virtual computing environment. Each user can have his/her own virtual desktop, accessed through the Internet by means of devices such as laptop, PC, notebook, tablet or smartphone. The benefits of desktop virtualization include the possibility of improving utilization of computational resources such as processor, memory, disk and network; a more efficient management; centralized backup, and remote accessibility independently on the user location.

Education institutions have traditionally used local desktops for use of faculty, administrative and technical staffs, as well as for their students. Managing these desktops requires a technical staff to take care of installation, configuration, updating and maintaining in a personalized manner, according to each user profile. In this manner, administration is costly and inefficient. Several companies increasingly adopted virtual desktops delivered by an infrastructure of centralized servers, while education institutions are late in this process. The diversity of applications used by different areas of teaching create demand for studies to better understand the performance of virtual desktop infrastructure.

The purpose of this study is to evaluate virtual desktop infrastructure from the perspective of user quality of experience. To accomplish this, a virtual desktop infrastructure have been deployed at UFSCAR, and a set of experiments were carried out at our labs employing five different applications. To evaluate the influence of the networking quality of service on the user experience, we produced 5 scenarios with different communication latencies. Finally, we also executed experiments to evaluate the demand for computational resources (e.g., processor, memory, I/O, and network) as more virtual desktops are spawned. This experiment can provide important information for future capacity planning.

Keywords: Virtualization; Virtual Desktop; Remote Desktop; VDI; DaaS

LISTA DE FIGURAS

2.1	Fluxo de comunicação do protocolo PCoIP	23
2.2	Pilha do protocolo RDP	25
2.3	Formato de uma mensagem RDP	26
2.4	Visão geral dos componentes de uma arquitetura VDI	28
2.5	Visão geral dos componentes de uma arquitetura VDI distribuída em diferentes <i>data centers</i> . Fonte (CALYAM, 2011)	29
2.6	Visão geral da arquitetura de VDI Xendesktop. Adaptado de Wang (WANG; LIANG, 2011)	30
2.7	Exemplo da arquitetura de VDI da Microsoft. Adaptado de Wang (WANG; LIANG, 2011)	34
2.8	Exemplo da arquitetura de VDI da VMWare Horizon View. Fonte (VMWARE, 2012c)	35
2.9	Exemplo da arquitetura de administração do vCenter. Fonte (VMWARE, 2008)	36
2.10	Arquitetura básica da disciplina de fila do Linux. Adaptado de (HEMMINGER et al., 2005)	38
3.1	Infraestrutura de <i>Desktops</i> Virtuais	45
3.2	Infraestrutura do ambiente de teste	47
3.3	Quantidade de testes realizados	48
3.4	Arranjo de teste sequencial (A) e não sequencial (B)	49
3.5	Qualidade de experiência por aplicação - Digitação de textos	51
3.6	Qualidade de experiência por aplicação - Vídeo	52
3.7	Qualidade de experiência por aplicação - Internet Explorer	53

3.8	Qualidade de experiência por aplicação - Eclipse	54
3.9	Qualidade de experiência por aplicação - Matlab	55
3.10	Média da qualidade de experiência das aplicações em diferentes condições de rede	56
3.11	Média da taxa de transferência de dados entre Cliente e Servidor de Conexão .	57
3.12	Média de utilização de CPU por aplicação e quantidade de desktops virtuais . .	58
3.13	Média de utilização de memória por aplicação e quantidade de desktops virtuais	59
3.14	Operações de leitura durante a carga do Eclipse em 25 desktops virtuais	60
3.15	Média de operações de E/S de leitura por aplicação e quantidade de desktops virtuais	61
A.1	Backbone atual e futuro da OARnet. Fonte (OARNET, 2013b)	68

LISTA DE TABELAS

2.1	Tipos de mensagens msgType do protocolo RDP	27
2.2	Tipos de mensagens msgFlags do protocolo RDP	27
2.3	Escala de valores do modelo MOS	38
3.1	Tipo de teste executado em cada aplicação	42
3.2	Localização dos laboratórios de informática no campus UFSCar São Carlos . .	43
3.3	Latências utilizadas nos experimentos	44
3.4	Rótulo dos eixos das abcissas dos gráficos de qualidade de experiência por aplicação	50

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	12
1.1 Objetivos	14
1.2 Metodologia	14
1.3 Justificativa	15
1.4 Estrutura da dissertação	15
CAPÍTULO 2 – TECNOLOGIAS E TRABALHOS RELACIONADOS	17
2.1 Virtualização	17
2.2 Computação em nuvem	18
2.3 Desktops locais e desktops virtuais	20
2.4 Protocolos de distribuição de <i>desktops</i> virtuais	21
2.4.1 O protocolo PC over IP (PCoIP) da Teradici	22
2.4.2 O protocolo ICA/HDX da Citrix	23
2.4.3 Remote Desktop Protocol (RDP)	24
2.5 Infraestrutura de desktops virtuais (VDI)	26
2.5.1 Citrix: XenDesktop	30
2.5.2 Microsoft RDS	32
2.5.3 VMware Horizon View	34
2.6 Emulação de redes com NetEm	37
2.7 Avaliação subjetiva utilizando Mean Opinion Score	38

CAPÍTULO 3 – UMA AVALIAÇÃO EXPERIMENTAL DO USO DE DESKTOPS VIRTUAIS	40
3.1 Objetivos dos experimentos	40
3.2 Seleção das aplicações	41
3.3 Qualidade de serviço da rede da UFSCar	42
3.4 Descrição do ambiente de experimentos	44
3.5 Projeto de experimentos	46
3.5.1 Testes para a avaliação da qualidade de experiência dos usuários	46
3.5.2 Testes de capacidade	50
3.6 Resultados obtidos	50
3.6.1 Resultados dos testes de qualidade de experiência	50
3.6.2 Resultados dos testes de capacidade	55
3.7 Conclusão	56
CAPÍTULO 4 – CONCLUSÃO	62
4.1 Trabalhos futuros	63
REFERÊNCIAS	64
GLOSSÁRIO	67
APÊNDICE A – OARNET E O PROJETO VDPILOT	68
APÊNDICE B – PROCEDIMENTOS DE TESTES DE APLICAÇÕES	69
B.1 Procedimento de teste com Microsoft Word	69
B.2 Procedimento de teste com Microsoft Media Player	70
B.3 Procedimento de teste com Microsoft Internet Explorer	70
B.4 Procedimento de teste com Eclipse	71
B.5 Procedimento de teste com Matlab	71

Capítulo 1

INTRODUÇÃO

As instituições de ensino brasileiras encontram-se face a uma série de desafios na área de tecnologia da informação. Com a virtualização e a computação em nuvem, abrem-se possibilidades para oferecer uma plataforma para melhorar a qualidade de ensino e a partilha de recursos educacionais.

Os alunos do ensino presencial geralmente dependem dos laboratórios informatizados da universidade para terem acesso a hardware e software necessários para sua aprendizagem. Para poder continuar seus estudos fora da universidade, os alunos são obrigados a instalar em seu computador pessoal as ferramentas necessárias e ainda enfrentam os problemas de requisitos de hardware e de licenciamento de alguns softwares. Há também a situação dos alunos do ensino a distância, que enfrentam os mesmos problemas, além de não poderem contar com laboratórios informatizados. Logo, o desafio das instituições de ensino é provisionar recursos computacionais para aprendizado para além dos laboratórios, indo para as casas dos alunos e professores, podendo ter acesso a toda a plataforma de ensino a qualquer hora, dentro do seu ritmo.

Do ponto de vista da instituição de ensino, os desafios estão voltados a própria manutenção da infraestrutura. Para manter os laboratórios informatizados funcionais, a administração responsável pelos laboratórios precisa ter uma equipe de suporte, investir periodicamente na substituição dos computadores e gerenciar as licenças de softwares.

A equipe de suporte tem por função prestar manutenção ao hardware e ao software dos computadores dos laboratórios. O trabalho de manutenção dos computadores dos laboratórios envolve a troca de peças danificadas, como discos rígido, fontes de alimentação de energia, placas de rede, processadores, placas mãe, etc. Outra função da equipe é atualizar o sistema operacional, antivírus, aplicar correções de segurança e instalar softwares específicos para as

disciplinas lecionadas em cada um dos laboratórios. Contudo, o trabalho da equipe demanda tempo, e também é repetitivo pois a manutenção de software é feita máquina por máquina.

A substituição dos equipamentos antigos dos laboratórios por novos também é um custo necessário por conta da demanda crescente dos softwares por maior capacidade de processamento e armazenamento do hardware.

A respeito dos custos de licenciamento, um *desktop* de laboratório que ficar apenas oito horas por dia acessível utiliza portando apenas um terço da utilidade da licença. Logo, quanto mais for estendido o tempo de utilização das licenças, mais seus custos serão amortizados devido ao crescimento da sua taxa de utilização. Frentes a estes desafios e das possibilidades de virtualização e de computação em nuvem, algumas universidades estão utilizando *desktops* virtuais para o provisionamento de recursos para atividade de ensino.

Desktops virtuais são computadores pessoais virtualizados, executando em uma infraestrutura gerenciada por hipervisor. Esta infraestrutura fica localizada em um data center dentro da própria instituição ou em um provedor de nuvem. É composta por um conjunto de máquinas com grande capacidade de processamento e memória e também de uma grande área de armazenamento. O usuário acessa seu *desktop* utilizando um software cliente que faz primeiro uma autenticação e depois a conexão com o *desktop* virtual. O cliente envia para o *desktop* comandos de mouse e teclado e o *desktop* envia ao cliente apenas os pixels necessários para atualizar a imagem vista pelo usuário no cliente.

A adoção de *desktops* virtuais pode ajudar as instituições de ensino e pesquisa a diminuir seus custos de gerenciamento e aquisição de equipamentos, além aumentar a disponibilidade de acesso do aluno aos recursos de ensino e pesquisa normalmente restritos nos laboratórios. A equipe de suporte poderá criar perfis de *desktops* virtuais, com softwares específicos para cada disciplina, e disponibilizar a quantidade de instâncias necessárias para cada laboratório dentro do tempo necessário. Ao final do uso, estas instâncias podem ser apagadas. As atualizações de sistema operacional e correções de segurança podem ser aplicadas uma única vez e replicadas para as instâncias. Logo, o esforço de gerencia da equipe de suporte é menor e mais eficiente. Quanto a substituição das máquina antigas dos laboratórios para novas, a frequência pode diminuir porque o *desktop* virtual executa no data center, logo, do lado do cliente, é necessário pouco recurso computacional para executar. Por fim, o aluno pode acessar o *desktop* do laboratório quando e onde quiser.

1.1 Objetivos

Uma infraestrutura de *desktops* virtuais permite o acesso a *desktops* sem as limitações impostas por um ambiente físico, como disponibilidade e localização. A disponibilidade diz respeito ao tempo permitido para acesso ao recurso, e a localização assume que o aluno deve se deslocar até o ambiente físico, como o laboratório. Já o acesso ao *desktop* virtual é feito através da rede, logo a qualidade de experiência do usuário é fortemente influenciada pelo desempenho da rede que interliga a infraestrutura que provisiona o *desktop* e o cliente final.

Outro fator que vai influenciar na qualidade de experiência do usuário são os tipos de aplicações que serão executadas no *desktop* virtual. Uma ferramenta que produz gráficos tridimensionais, por exemplo, demanda mais recursos que outra que manipula texto. Isto acontece devido ao funcionamento do protocolo, que recebe do dispositivo do cliente comandos de mouse e teclado, e retorna atualizações de pixels da tela do cliente. Quanto mais a aplicação demandar atualização de imagens, maior será o tráfego na rede.

O objetivo deste trabalho é avaliar a viabilidade do uso de *desktops* virtuais nos laboratórios das instituições de ensino e pesquisa, substituindo o uso do computador pessoal com sistema operacional e aplicações instaladas localmente, visando diminuir os custos de gerencia dos laboratórios e facilitar o acesso ao ambiente de estudo.

A principal foco deste trabalho é avaliar o impacto do uso de *desktops* virtuais (em vez dos locais) sobre a qualidade de experiência do usuário. Outro enfoque relacionado é investigar a influencia dos diferentes níveis de qualidade de serviço de rede sobre os diferentes tipos de aplicações que o usuário pode utilizar. A respeito da infraestrutura necessária para viabilizar o uso de *desktops* virtuais, este trabalho procura levantar quais são os requisitos em termos de recursos computacionais necessários para implementar uma infraestrutura de *desktops* virtuais.

1.2 Metodologia

Para atingir os objetivos deste trabalho foram realizados experimentos *in vitro*, nos quais um ambiente foi preparado e oferecido para avaliação por parte de usuários, a fim de medir a qualidade de experiência no uso de *desktops* virtuais, comparado com o uso de estações de trabalho locais. O estudo teve início com a escolha de algumas aplicações representativas tipicamente utilizadas em laboratório. A seguir, foram levantadas as condições de latência da rede tipicamente observadas em um ambiente real, que foram reproduzidas nos experimentos. Para a análise da qualidade de experiência dos usuários, foi utilizado um modelo de avaliação cen-

trado na qualidade percebida pelos usuários. Com estes parâmetros foram criados experimentos e aplicados aos usuários. Para o levantamento dos requisitos em termos de recursos computacionais necessários para implementar um infraestrutura de desktops virtuais, foram feitos testes de uso de até vinte e cinco desktops virtuais executando simultaneamente um mesmo tipo de aplicação. Esta execução foi feita por meio de macros de gravação de ações de teclado e mouse, logo os testes foram feitos sem a presença de usuários.

O tipo de teste aplicado nos experimentos com usuários foi o subjetivo. Os testes subjetivos dão uma medida da percepção do usuário frente ao sistema proposto. Já o teste de capacidade de infraestrutura foi objetivo, produzindo informações de uso de processamento, memória e disco para cada tipo de aplicação e quantidade de desktops executando simultaneamente.

1.3 Justificativa

O cenário alvo do estudo são as instituições de ensino e pesquisa, com um ou mais campi, conectadas a uma infraestrutura de rede internet de alta velocidade, como é a rede Ipê (RNP, 2013). Com esta infraestrutura de conectividade disponível, é possível agregar serviços que possam facilitar o acesso a aplicações necessárias para estudo, ensino e pesquisa. A infraestrutura de *desktops* virtuais pode disponibilizar para acesso remoto, *desktops* com aplicações e capacidade de hardware necessárias.

A rede Ipê é uma rede de fibra ótica operada pela Rede Nacional de Ensino e Pesquisa (RNP). Ela conecta instituições de ensino e pesquisa a fim de haver comunicação e colaboração entre elas, além de servir de laboratório de testes e desenvolvimento de aplicações. Este conjunto de instituições é formado por unidades de pesquisa e instituições públicas de ensino superior. No Brasil, cerca de 800 instituições estão conectadas.

1.4 Estrutura da dissertação

Esta dissertação está dividida em 4 capítulos, incluindo o capítulo introdutório:

- O **Capítulo 2** apresenta o levantamento dos conceitos e tecnologias utilizados na dissertação: virtualização, computação em nuvem, protocolos de distribuição de *desktops*, infraestruturas de *desktops* virtuais, emulação de redes, análise de tráfego e MOS (*Means Score Opinion*).
- O **Capítulo 3** apresenta os objetivos específicos da parte experimental, também apresenta

projeto de experimentos, resultados e análises.

- O **Capítulo 4** apresenta as conclusões.

Capítulo 2

TECNOLOGIAS E TRABALHOS RELACIONADOS

Este capítulo descreve as principais tecnologias relacionadas com este trabalho. São apresentados os conceitos de virtualização, computação em nuvem e *desktop* virtual. Também são descritos conceitos, arquiteturas de infraestrutura de *desktops* virtuais e protocolos relacionados.

2.1 Virtualização

Em Silberschatz, Galvin e Gagne (2009) os autores definem virtualização como a tecnologia que permite aos sistemas operacionais executarem como aplicações dentro de outros sistemas operacionais. Com a virtualização, um sistema operacional compilado para uma determinada arquitetura executa em um outro sistema operacional também nativo aquela CPU.

A ideia fundamental por trás de uma máquina virtual é abstrair o hardware de um computador em vários ambientes de execução, de tal forma a criar a ilusão de que cada ambiente está sendo executado em seu próprio computador individual.

A implementação de máquinas virtuais requer vários componentes. Na base está o hospedeiro, o sistema de hardware que executa as máquinas virtuais. O gestor de máquinas virtuais (VMM), também conhecido como hipervisor, instancia e controla as máquinas virtuais, fornecendo uma interface idêntica à do hospedeiro (exceto no caso de paravirtualização). Cada máquina virtual hospedada (do inglês *guest*) receberá uma cópia virtual do hospedeiro. Normalmente, o cliente é um sistema operacional chamado de hospede. Logo, uma única máquina física pode executar vários sistemas operacionais hospedados ao mesmo tempo, cada um em sua própria máquina virtual.

A virtualização é uma das tecnologias importante para a chave da computação em nuvem,

uma vez que ela provê a capacidade de alocar e desalocar dinamicamente recursos computacionais de um aglomerado de servidores para aplicações, de acordo com a demanda (ZHANG; CHENG; BOUTABA, 2010). A virtualização também permite a computação sob demanda, ou computação utilitária que é, modelo de fornecimento de recursos computacionais, como CPU, memória e disco, que são disponibilizados apenas quando necessário em vez de alocados de forma estática. Através do provisionamento dinâmico de máquinas virtuais, da consolidação da carga de trabalho, e ainda do ligamento e desligamento de servidores quando necessário, os operadores de *datacenter* podem manter a qualidade serviço pretendida enquanto conseguem uma alta utilização dos servidores e alta eficiência energética (KUSIC, 2009).

2.2 Computação em nuvem

Em MARSTONA (2011), computação em nuvem é definida como:

“É um modelo de serviço onde serviços computacionais (ambos hardware e software) são fornecidos sob demanda aos consumidores, pela rede e de maneira *self-service*, independente do dispositivo computacional utilizado ou localização do usuário. Os recursos necessários para proporcionar o nível de qualidade de serviço requisitada são compartilhados, dinamicamente escaláveis, rapidamente provisionados, virtualizados e liberados com a mínima interação do provedor de serviço. Os usuários pagam pelo serviço como uma despesa operacional sem incidir em despesas significantes com capital inicial, com os serviços em nuvem utilizando um sistema de medição que divide os recursos computacionais em blocos apropriados”

Na definição do NIST (US National Institute of Standards and Technology):

“Computação em nuvem é um modelo que possibilita de forma conveniente, o acesso sob demanda a um conjunto compartilhado de recursos computacionais (como rede, servidores, armazenamento, aplicativos e serviços) que podem ser rapidamente provisionados e liberados com o mínimo esforço de gerenciamento ou interação do provedor de serviço” (MELL P; GRANCE, 2011).

Ainda de acordo com o NIST, o modelo de computação em nuvem apresenta cinco características essenciais, que são:

- **Serviços sob demanda:** um consumidor pode obter capacidades computacionais, como tempo de utilização de servidor ou armazenamento em rede conforme a necessidade, sem precisar de interação humana com seu provedor;
- **Acesso amplo à rede:** capacidades estão disponíveis pela rede e acessadas através de mecanismos padrão que promovem a utilização de plataformas heterogêneas tais como telefones móveis, computadores portáteis e PDAs;

- **Agrupamento de recursos:** Os recursos computacionais dos provedores são agrupados para atender múltiplos consumidores utilizando o modelo multi-inquilino ou multi-arrendatário (do inglês *multitenancy*). Nesse modelo, uma aplicação atende a múltiplos usuários individuais (inquilinos ou *tenants*) porém de forma isolada. É diferente do modelo *on-premise* em que uma cópia do software deve ser instalada nos servidores do cliente;
- **Rápida elasticidade:** As capacidades computacionais podem ser rapidamente ajustadas de acordo com a demanda, até mesmo de forma automática. Para o consumidor, a capacidade de provisionamento pode parecer ilimitada e pode ser adquirida em qualquer quantidade e qualquer momento;
- **Serviço mensurado:** A utilização dos recursos pode ser monitorada, controlada e reportada ao consumidor, proporcionando transparência no fornecimento do serviço.

O NIST também define três modelos básicos de entrega de serviços em nuvem. No modelo denominado Software como Serviço (SaaS), a aplicação executa na nuvem do provedor de serviço, eliminando a necessidade de instalar e executar a aplicação no computador do cliente. Assim, este modelo utiliza as funcionalidades de software, sem precisar saber qual tecnologia é necessária para funcionar, como também não precisa se preocupar com atualizações e correções de segurança. Como exemplo de SaaS, podemos citar os serviços da Salesforce, Netsuite, Google Apps para aplicação de e-mail GMail, Facebook ou Twiter. O modelo denominado Plataforma como Serviço (PaaS) é fornecido uma plataforma para desenvolvimento, implantação e gerenciamento de aplicações sem os custos e a complexidade de compra e gerenciamento do hardware e software. Como exemplo, podemos mencionar o Windows Azure, Force.com, Google App Engine. Por fim, no modelo Infraestrutura como Serviço (IaaS) existe um conjunto de componentes de hardware, tais como storage, ciclos de CPU, capacidade de memória, banda de rede, que são entregues como serviços através da rede por empresas como Amazon, Terramark e Rackspace.

O NIST define ainda quatro modelos de implementação de nuvens. No modelo denominado *Nuvem Privada*, os recursos servem apenas a uma determinada organização e seus usuários, oferecendo como vantagem seu total controle sobre os mesmos. Em contrapartida exige um investimento grande por parte da organização para montar sua infraestrutura. No modelo de *Nuvem Comunitária*, a infraestrutura de nuvem é compartilhada por uma comunidade de organizações com interesses em comum. Já o modelo de *Nuvem Pública* oferece seus recursos como serviços ao público em geral. Este modelo apresenta vantagens aos seus clientes, que não precisam dispor de investimento inicial em infraestrutura para hospedar seus serviços, transferindo a respon-

sabilidade da infraestrutura para o provedor de nuvem. Por fim, o modelo de *Nuvem Híbrida*, uma combinação das vantagens das Nuvens Privadas e Públicas, é possível deixar parte dos serviços em cada uma delas, como também mandar para a Nuvem Pública o excedente da carga de trabalho na Nuvem Privada.

Com o avanço da sociedade moderna, serviços essenciais água, esgoto, energia elétrica, gás encanado e telefonia são considerados *commodities*, que traduzido para o português significa mercadoria. Tais serviços estão disponíveis para que qualquer pessoa possa acessá-los sem a preocupação de onde ou como são fornecidos. Neste modelo de negócio, o usuário apenas deve pagar pelo que consome. Da mesma forma, a computação vem sendo *comoditizada*, sendo chamada de computação utilitária. O paradigma que vem possibilitando esta comoditização é a computação em nuvem, permitindo que os usuários utilizem a computação como um serviço, pelo mesmo modelo denominado Pay-as-You-Go (BUYA, 2009).

2.3 Desktops locais e desktops virtuais

O *desktop* local, também denominado cliente pesado (do inglês *thick client*), é um computador de uso pessoal ou coletivo, em que o processamento é feito todo localmente, nele sendo instalados o sistema operacional, softwares aplicativos e utilitários. Esse modelo tradicional de uso de *desktops* feito pelas organizações, está fundamentado em fornecer a cada usuário um computador com a maior capacidade de processamento e armazenamento possíveis, com todos os softwares necessários para que o usuário possa utilizar. Como o *desktop* permanece fisicamente na organização, o usuário necessita se deslocar ao local onde ele está para fazer uso, além de depender dos horários de disponibilidade do recurso. O mesmo problema acontece nas salas de ensino de escolas e universidades.

De acordo com Sridharan (2011), o gerenciamento deste tipo de ambiente é custoso pois requer uma equipe de suporte dedicada à instalação, configuração, atualização e manutenção de hardware e software, e da implementação de medidas de segurança, como instalação de anti-vírus e *firewall*. Além disso, este modelo intrinsecamente subutiliza estes recursos, resultando em um baixo retorno de investimento (ROI). Além disso, como os dados ficam localmente armazenados, em caso de dano ou roubo da máquina, há perda de dados e também a possibilidade de serem acessados por pessoas não autorizadas.

Segundo a empresa VMWare (VMWARE, 2008), devido às desvantagens na utilização de *desktops* tradicionais, seu uso vem sendo reavaliado. Muitas companhias têm procurado migrar os seus *desktops* para máquinas virtuais e executando em servidores centralizados, possibili-

tando a centralização de recursos e melhorias no gerenciamento .

De acordo com Casas (2013), um *desktop* virtual pode ser definido a partir do conceito de *desktop* remoto. O autor explica que o paradigma do *desktop* remoto permite ao usuário acessar o conteúdo e aplicações executadas em seus computadores pessoais através de uma conexão de rede. Quando estes computadores são provisionados por meio de um gerenciador de máquina virtual ou hipervisor, os *desktops* remotos se transformam no paradigma dos *desktops* virtuais remotos. Neste caso, o *desktop* virtual é renderizado por um servidor, e o usuário necessita apenas de um dispositivo com capacidade de visualização (por exemplo: um computador pessoal, *tablet*, *smartphone*, etc), um software client e uma conexão de rede para acessar o *desktop* virtual de qualquer local. Este serviço tem sido tradicionalmente instalado em redes locais de empresas, onde o desempenho da rede é ótimo e a qualidade de experiência sofrida por estes usuários geralmente não é um problema.

O projeto VDPilot e o Laboratório de Computação Virtual são exemplos destas iniciativas. O projeto VDPilot (OARNET, 2013c), iniciado pelo Concelho Consultivo do Conselho de Regentes de Ohio (OHIOHIGTERED, 2013), é um estudo de viabilidade de uma “nuvem de *desktops* virtuais” para os laboratórios e salas de aula das instituições conectadas a OARnet(OARNET, 2013a). O projeto VDPilot utiliza o acesso de alta velocidade das universidades, de modo a:

- Avaliar a experiência do usuário ao acessar *desktops* virtuais, comparado ao acesso físico de um laboratório;
- Analisar os desafios e reduções de custos, devido aos recursos compartilhados entre as instituições membros da OARnet.

A Universidade do Estado da Carolina do Norte (NCSU) criou em 2003 o Laboratório de Computação Virtual (VCL) (NCSU, 2013), que veio fornecer através da Internet, a mais de trinta mil alunos e professores, acesso remoto a *desktops* virtuais com diversas configurações de hardware e software. Para isso, a universidade desenvolveu um software que dinamicamente provisiona recursos computacionais aos usuários, mediante agendamentos (LI; TODERICK; LUNSFORD, 2009). Esse software mais tarde foi encubado pelo Projeto Apache (APACHE, 2013).

2.4 Protocolos de distribuição de *desktops* virtuais

A interação entre o *desktop virtual* localizado em um *data center* e o usuário remoto acontece, por exemplo, quando o este movimentar o cursor ou utiliza o teclado e recebe imediata-

mente do desktop as respostas aos comandos em forma de imagens que vão se atualizando. Esta comunicação para acontecer de forma remota precisa de um protocolo específico.

Aqui são descritos os principais protocolos que permitem a distribuição de *desktops* virtuais. Esses protocolos permitem a comunicação entre os clientes por meio dos quais os usuários interagem com os servidores que executam os desktops virtuais

2.4.1 O protocolo PC over IP (PCoIP) da Teradici

PCoIP é um protocolo proprietário da Teradici Corporation, destinado a renderizar as imagens geradas no servidor hospedeiro localizado na infraestrutura de VDI no *data center* e transmiti-las de forma comprimida aos dispositivos clientes. Foi adotado como protocolo padrão nos produtos da empresa VMWare para sua plataforma de VDI, chamada VMWare View. Outro protocolo suportado é o RDP. De acordo com o fabricante (TERADICI, 2013), as três características essenciais do protocolo são a renderização no servidor hospedeiro, a utilização de múltiplos codificadores de vídeo e áudio e a adaptação do protocolo às condições de rede.

Assim que a imagem é renderizada no hospedeiro, o protocolo transmite ao cliente apenas os pixels compactados através da rede, e não os dados. Logo o cliente precisa apenas decodificar estes pixels, atualizando a imagem mostrada ao usuário. Pelo fato do cliente não receber dados e sim pixels, evitam-se problemas de incompatibilidade entre servidor hospedeiro e cliente, pois não existe dependência do cliente em relação à controladora gráfica, ou mesmo em relação à aplicação ou sistema operacional executados no hospedeiro. Como o protocolo realiza a entrega de pixels de forma unilateral (*one-way delivery*) para os clientes, isto propicia a completa independência sobre a latência da rede e limitação de banda de transmissão, possibilitando uma melhor experiência do usuário com o ambiente de *desktop* virtual.

Como a imagem do hospedeiro é feita de diferentes elementos, tais como gráficos, textos, fotografias, ícones e vídeos, o protocolo PCoIP decompõe estes diversos elementos e faz uso de diversos codificadores de acordo com cada tipo, a fim de otimizar a transmissão e decodificação, além de consumir menor banda de transmissão.

O protocolo pode ser configurado para que automaticamente varie a qualidade da imagem enviada do hospedeiro ao cliente, de acordo com as condições da banda de transmissão, diminuindo a qualidade quando a rede está congestionada, ou aumentando quando não. O PCoIP utiliza UDP para transmitir os pixels, o mesmo utilizado em outras aplicações de tempo real como voz sobre Ip e IPTV.

A Figura 2.1 mostra o fluxo de comunicação do PCoIP, as portas utilizadas entre origem,

o software client, e destino, o servidor de conexão. Também mostra a comunicação entre o servidor de conexão e o *desktop* hospedado no aglomerado de virtualização.

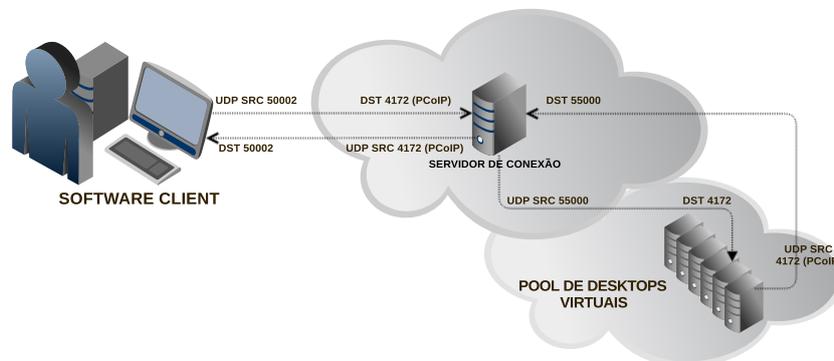


Figura 2.1: Fluxo de comunicação do protocolo PCoIP

2.4.2 O protocolo ICA/HDX da Citrix

O protocolo de Arquitetura de Computação Independente (ICA), é proprietário da Citrix. Seu principal objetivo é permitir que um conjunto centralizado de servidores baseados em Microsoft Windows entreguem aplicativos ou máquinas virtuais para usuários de outras plataformas, como Mac OS e Linux (AMAZON, 2013).

Pela perspectiva do modelo OSI, o protocolo ICA pertence à camada de apresentação, que é definida para serviços de conversão de dados, compressão e descompressão, encriptação e decríptação. A transferência de dados entre servidor e cliente geralmente inclui movimentos de mouse e eventos de teclado.

A arquitetura do protocolo ICA é baseada em 32 canais virtuais com prioridades diferentes cada um. O fornecimento de QoS é baseado na priorização, assim, dados menos importantes são descartados se necessário (WANG; LIANG, 2011).

Com o surgimento de outras soluções de VDI, como PCoIP, a Citrix adicionou novas tecnologias e componentes do lado do servidor, rede e cliente, a fim de integrar com o protocolo ICA e assim surgiu a marca HDX.

HDX, que é o acrônimo de “*High Definition User Experience*”, não é um protocolo ou tecnologia, mas sim uma marca ou termo guarda-chuva que engloba outros sub-componentes desenvolvidos pela Citrix ou adquiridos por ela de outros fabricantes. Exemplos disso incluem o HDX MediaStrem, HDX RealTime, HDX 3D e HDX IntelliCache, este último adquirido de outra empresa. Cada um destes tem uma variedade de tecnologias, como por exemplo HDX

Plug-in-Play, que dá suporte a dispositivos USB, impressoras, múltiplos monitores, *smart cards* e digitalizadores (CITRIX, 2013).

De acordo com a Citrix, existem mais de sessenta aspectos relacionados ao HDX, um deles é a Orquestração Adaptativa, em que os componentes se adaptam de acordo com as condições de rede e do dispositivo cliente.

2.4.3 Remote Desktop Protocol (RDP)

O protocolo RDP (*Remote Desktop Protocol*) é uma tecnologia proprietária da Microsoft, que estende a família T-120 de padrões de protocolos do ITU (ITU, 2009). O RDP é um protocolo multicanal que permite a troca de dados entre dispositivos de forma serial, permite o envio de informações de licenciamento, dados criptografados (teclado, atividade do mouse), pixels e quaisquer interações entre o desktop remoto e o cliente. O RDP oferece um serviço multiponto, permitindo inclusive múltiplas sessões com vários participantes.

O RDP foi concebido para operar sobre várias tecnologias e protocolos como rede ISDN, IPX, NetBIOS e TCP/IP. As versões mais recentes utilizam apenas TCP/IP. O RDP trata das camadas de aplicação e de apresentação do modelo RM-ISO/OSI, e utiliza outros serviços de camadas inferiores, tais como sessão e de transporte.

A pilha de protocolos do RDP, mostrada pela Figura 2.2, é composta pelos seguintes componentes, que são o MCS (*Multipoint Communication Service*), o GCC (*Generic Conference Control*):

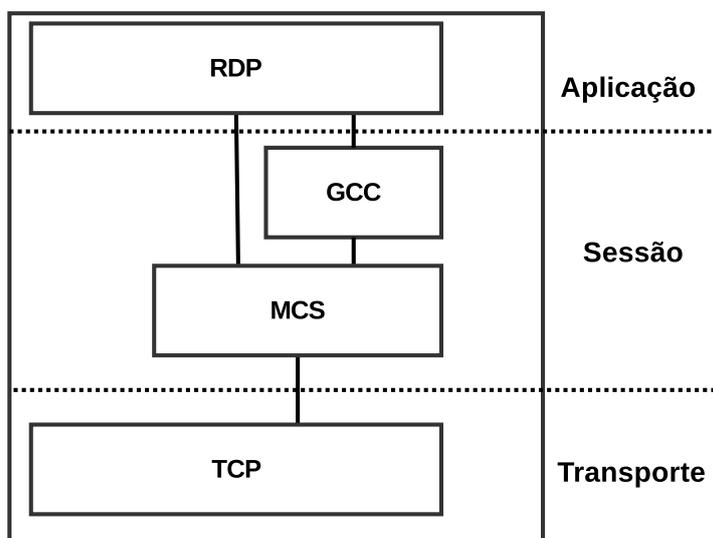


Figura 2.2: Pilha do protocolo RDP

- O MCS controla a atribuição de canais (fazendo multiplexação de dados em canais virtuais predefinidos no protocolo), os níveis de prioridade e a segmentação dos dados que estão sendo enviados. Ele reduz as pilhas múltiplas de RDP a uma única entidade, da perspectiva do GCC. O MCS é parte da família T.120 do ITU (ITU, 2009), sendo composto por dois padrões: o T.122, que define os serviços multiponto e o T.125, que especifica o protocolo de transmissão de dados.
- O GCC implementa um serviço de sessões. Ele permite a criação e a exclusão de conexões de sessão e controla os recursos fornecidos pelo MCS. Cada protocolo do Terminal Server (atualmente, apenas o RDP e o ICA da Citrix recebem suporte) tem um parâmetro na instância da pilha de protocolos carregada (uma pilha de escuta aguardando um pedido de conexão). O driver de dispositivo do Terminal Server, implementado pelo módulo Wdtshare.sys, coordena e gerencia as atividades dos protocolos RDP para transferência, compressão, criptografia, e enquadramento de dados de interface. O GCC utiliza o serviço de transporte do TCP implementado pelo módulo Tdtcp.sys.

A Figura 2.3 mostra o formato das mensagens do protocolo RDP e as Tabelas 2.2 e 2.2 listam os tipos de mensagens e seu significado (MICROSOFT, 2013).

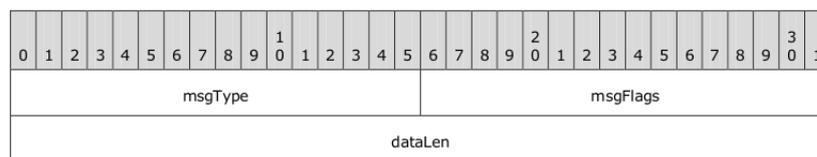


Figura 2.3: Formato de uma mensagem RDP

2.5 Infraestrutura de desktops virtuais (VDI)

De acordo com a VMWare (VMWARE, 2006), a Infraestrutura de *Desktops* Virtuais (VDI) é uma solução de computação baseada em servidores que fornece ambientes de *desktop* como um serviço corporativo hospedado. VDI permite aos administradores manter e gerenciar em um *data center*, todos os aplicativos dos usuários, dados e sistema operacional. Também fornece isolamento entre os *desktops* enquanto compartilha recursos de hardware subjacentes, além disso, permite que administradores forneçam *desktops* com todos os benefícios da virtualização, como a gestão central, consolidação de hardware e flexibilidade de recursos.

Segundo a empresa (VMWARE, 2006), com VDI é possível montar uma infraestrutura utilizando *thin-client* que otimize a usabilidade, gerência, TCO e flexibilidade. Ambientes completos de *desktops* podem executar em máquinas virtuais em servidores de *data centers* que podem

Tabela 2.1: Tipos de mensagens msgType do protocolo RDP

Mensagem	Significado
CB_MONITOR_READY 0x0001	Monitor Ready PDU
CB_FORMAT_LIST 0x0002	Format List PDU
CB_FORMAT_LIST_RESPONSE 0x0003	Format List Response PDU
CB_FORMAT_DATA_REQUEST 0x0004	Format Data Request PDU
CB_FORMAT_DATA_RESPONSE 0x0005	Format Data Response PDU
CB_TEMP_DIRECTORY 0x0006	Temporary Directory PDU
CB_CLIP_CAPS 0x0007	Clipboard Capabilities PDU
CB_FILECONTENTS_REQUEST 0x0008	File Contents Request PDU
CB_FILECONTENTS_RESPONSE 0x0009	File Contents Response PDU
CB_LOCK_CLIPDATA 0x000A	Lock Clipboard Data PDU
CB_UNLOCK_CLIPDATA 0x000B	Unlock Clipboard Data PDU

Tabela 2.2: Tipos de mensagens msgFlags do protocolo RDP

Mensagem	Significado
CB_MONITOR_READY 0x0001	Monitor Ready PDU
CB_RESPONSE_OK 0x0001	Utilizada pelas PDUs Format List Response, Format Data Response, e File Contents Response para indicar que as solicitações associadas as PDUs Format List, Format Data Request e File Contents Request foram processadas com sucesso.
CB_RESPONSE_FAIL 0x0002	Utilizada pelas PDUs Format List Response, Format Data Response e File Contents Response para indicar que as PDUs associadas Format List, Format Data Request e File Contents Request não foram processadas com sucesso.
CB_ASCII_NAMES 0x0004	Utilizada pela variável Short Format Name da PDU Format List Response para indicar que o formato dos nomes estão em ASCII 8.

ser acessados pelos usuários de qualquer computador ou *thin-client* da rede da instituição. A solução de VDI oferece controle centralizado dos *desktops* e de seus dados, assim como a capacidade de consolidar máquinas virtuais e otimizar recursos de utilização do *data center*. Os usuários ganham a flexibilidade de poder acessar seus *desktops* de qualquer lugar.

A Infraestrutura de *desktops* virtuais oferece os benefícios de:

- Um ambiente de *desktop* isolado dos outros;
- Os dados estão mais seguros em um *data center*;
- Todas as aplicações executam em uma máquina virtual;
- As ferramentas de gerência também funcionam em máquinas virtuais;
- As imagens são gerenciadas de forma centralizada;
- *Desktops* podem ficar sempre ligados e conectados;
- Os usuários podem ter acesso aos seus *desktops* de qualquer lugar.

A Figura 2.4 mostra uma visão geral dos vários componentes de uma arquitetura de *desktop* virtual. Do lado do servidor temos um hipervisor utilizado para criar grupos de *desktops* virtuais com aplicações comuns como Word, Excel, Internet Explorer, Media Player e outras mais especializadas como, por exemplo, Matlab e Eclipse. Cada grupo contém instâncias do mesmo perfil de *desktop* virtual e os usuários são vinculados a cada um destes grupos. Os *desktops* virtuais compartilham os recursos computacionais do mesmo hardware físico e sistema de armazenamento de dados. Do lado do cliente, os usuários se conectam ao servidor de conexão via um protocolo de distribuição de desktops virtuais, como o PCoIP, ICA/HDX ou RDP. Este servidor de conexão autentica o usuário consultando o serviço de autenticação que posteriormente autoriza o usuário a conectar ao desktop virtual por meio de um *thin-client* ou software cliente instalado nos dispositivos como *tablets*, celulares ou computadores.

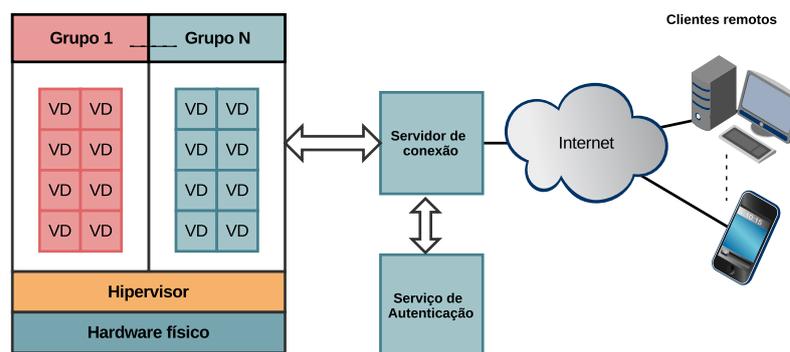


Figura 2.4: Visão geral dos componentes de uma arquitetura VDI

Calyam (2011) relaciona algumas das características que impulsionam o VDI, das quais se incluem o suporte ao sistema operacional, aplicações e atualizações de segurança, que são mais fáceis de gerenciar de forma centralizada através de ferramentas de gerenciamento, do que deslocar equipes até o local onde estariam os *desktops* fisicamente. Outra característica

está relacionada ao usuário de dispositivos móveis, que tem acesso amplo as suas aplicações e dados uma vez que pode utilizar, por exemplo, seu celular, *tablet* e *smartphone*, para acessar seu *desktop* virtual. Um outro aspecto citado pelo autor é a maior segurança dos dados, já que estão alocados na infraestrutura do *data center*. E por fim, o número de *desktops* subutilizados consumindo energia é reduzido. De acordo com Dasilva (2012), dispositivos *thin-clients* não utilizam mais que 40 watts, ao contrário dos 300 watts ou mais utilizados por um computador pessoal.

A Figura 2.5 mostra a visão geral de uma arquitetura com dois ou mais *data centers*, onde são executados os *desktops* virtuais e gerenciadas as conexões destes com os *thin-clients*. Em cada *data center* um hipervisor executa os *desktops* virtuais. Do lado do cliente, este se conecta remotamente a um servidor de conexão (*broker*). O servidor de conexão autentica o usuário utilizando a base de usuário do serviço Active Directory.

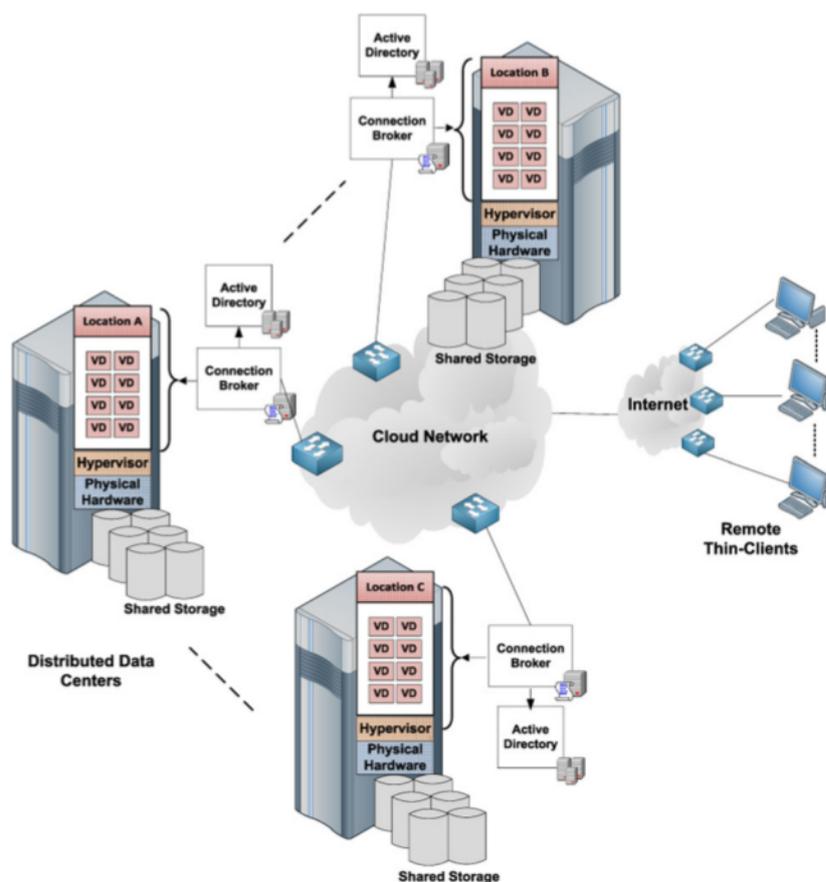


Figura 2.5: Visão geral dos componentes de uma arquitetura VDI distribuída em diferentes *data centers*. Fonte (CALYAM, 2011)

Existem alguns estudos sobre métodos de otimização da alocação de *desktops* virtuais em uma infraestrutura de VDI com vários servidores hospedeiros. Sridharan (2011) identifica e propõe uma solução ao problema de fragmentação de recursos computacionais, análogo ao

problema de fragmentação de disco, que pode ocorrer durante a alocação e desalocação de *desktops* virtuais em uma infraestrutura de VDI com múltiplos *data centers*.

2.5.1 Citrix: XenDesktop

XenDesktop é o nome do produto comercial de VDI da empresa Citrix. A solução suporta apenas *desktops* virtuais com sistema operacional Microsoft Windows, enquanto que o lado do cliente suporta plataformas diferentes como Windows, Linux e Mac OS. O hipervisor utilizado pode ser o próprio Citrix XenServer ou de outros fabricantes, como Microsoft Hyper-V e VMWare ESXi. O protocolo de comunicação utilizado para VDI é o ICA (WANG; LIANG, 2011).

A Figura 2.6 mostra os principais componentes da arquitetura da plataforma Xendesktop. Além destes componentes, podem ser instalados outros suplementares, como um servidor de licenças, DNS e DHCP. Os principais componentes da arquitetura são:

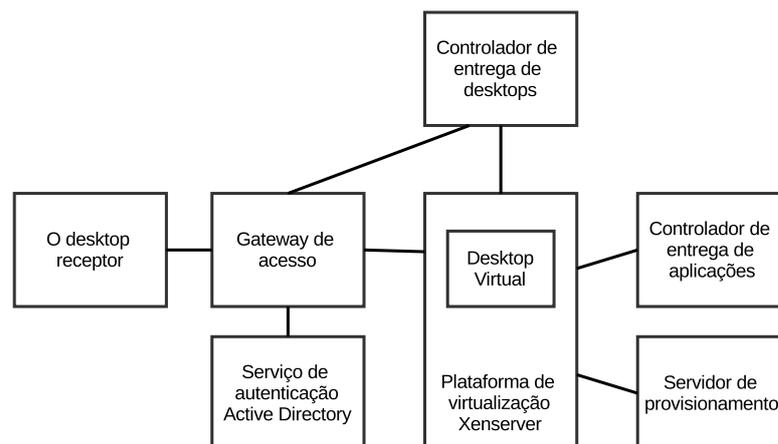


Figura 2.6: Visão geral da arquitetura de VDI Xendesktop. Adaptado de Wang (WANG; LIANG, 2011)

- *O desktop receptor*: é o software executando no lado do cliente, ele faz a conexão no *desktop* remoto através do protocolo ICA.
- *Gateway de acesso*: para o cliente se conectar ao *desktop* remoto, antes ele deve se conectar ao *gateway* para autenticação. Esta autenticação é feita via conexão segura SSL.
- *Controlador de entrega de desktops*: responsável por gerenciar a conexão entre o *desktop* remoto e o cliente. Quando um *desktop* inicia sua execução ele é registrado no controlador. Quando um cliente inicia uma conexão, o controlador seleciona o *desktop* remoto apropriado para conectar ao cliente e estabelece finalmente esta conexão.

- *Plataforma de virtualização*: é o hipervisor responsável em gerenciar as máquinas virtuais. O XenServer é a plataforma de virtualização da Citrix, porém outros hipervisores podem ser utilizados.
- *Servidor de provisionamento*: responsável em fornecer as imagens dos sistemas operacionais. Essas imagens tem um perfil de uso já pré-configurado, exceto as aplicações. Quando um desktop virtual é inicializado, sua imagem é fornecida pelo servidor de provisionamento.
- *Controlador de entrega de aplicações*: responsável pela entrega de aplicações. Cada perfil de usuário tem uma lista de aplicações associadas, esse usuário pode então escolher as aplicações a serem carregadas.
- *Desktop virtual*: é o *desktop* executando do lado do servidor. Para que este *desktop* possa ser conectado a um cliente ele tem um software agente instalado que aceita as conexões com o protocolo ICA.
- *Serviço de diretório*: responsável pela autenticação do usuário.

Para que o usuário acesse um desktop virtual através de seu cliente, ele deve passar por um processo de autenticação, cuja sequência é:

1. O usuário através de um navegador de Internet se conecta ao gateway de acesso, este gera uma página de autenticação;
2. O usuário informa suas credenciais e o gateway autentica as credenciais no serviço de diretório;
3. Após autenticado com sucesso, o gateway de acesso conecta no controlador de entrega de desktop;
4. O Controlador recupera as credenciais do usuário no Gateway de Acesso e as autentica novamente no Serviço de Diretório;
5. Após autenticado com sucesso o Controlador relaciona ao usuário quais desktops estão disponíveis a ele;
6. O Gateway de Acesso retorna ao usuário uma página contendo a lista de desktops virtuais disponíveis.

Com a lista de desktops disponíveis exibida no navegador, o usuário escolhe qual deseja acessar, e o sistema segue os procedimentos:

1. O Gateway de Acesso envia ao Controlador a informação de qual desktop foi escolhido pelo usuário;
2. O Controlador verifica se existe um desktop virtual disponível para o usuário através do seu status, caso não houver, o Controlador requisita ao hipervisor que seja inicializado um novo. Após inicializado o desktop abre uma porta ICA para esperar a conexão do cliente;
3. O Controlador cria um arquivo ICA do desktop virtual e transfere este arquivo ao Gateway de Acesso, que repassa ao cliente;
4. O cliente executa o arquivo ICA e envia uma requisição de conexão ao Gateway de Acesso;
5. O Gateway de Acesso repassa a requisição de conexão ao desktop virtual;
6. O desktop virtual envia ao Controlador um pedido de validação de autenticação;
7. Para a validação, o Controlador obtém uma licença no Servidor de Licenças e a envia junto com as credenciais ao desktop virtual. A conexão é finalmente criada após o desktop virtual autenticar as credenciais no Serviço de Diretório.

2.5.2 Microsoft RDS

O Serviço de *Desktop* Remoto (RDS) é proprietário da Microsoft. O protocolo utilizado pelo RDS para conectar o cliente e o servidor é o RDP. Os principais componentes do RDS são mostrados na Figura 2.7. Eles são (WANG; LIANG, 2011):

- *Cliente de Desktop Remoto*: é o software cliente executado no dispositivo, utilizado para acessar o *desktop* virtual. Esse cliente conecta o dispositivo ao *desktop* virtual utilizando o protocolo RDP;
- *Gateway de Desktop Remoto*: autoriza a conexão de clientes e fornece uma conexão segura para estes clientes acessarem o *desktop* virtual;
- *Serviço de Conexão*: é responsável por atribuir o *desktop* virtual correto ao cliente e também realizar balanceamento de carga de sessão. Um usuário pode ter um *desktop*

pessoal ou receber um dinamicamente. No caso do *desktop* pessoal, esse é atribuído ao mesmo usuário. No caso de uma atribuição dinâmica, o Serviço de Conexão atribui ao usuário qualquer *desktop* virtual disponível do grupo que o usuário tem autorização de acesso;

- *Servidor de Sessão*: responsável por hospedar programas baseados no Windows ou a área de trabalho completa do Windows para clientes. Os usuários podem se conectar a um servidor de sessão para executar programas, salvar arquivos e usar recursos de rede nesse servidor;
- *Servidor de Virtualização*: contém o hipervisor Hyper-V da Microsoft para hospedar e fornecer máquinas virtuais;
- *Serviço de Diretório*: responsável pela autenticação do usuário.

Os procedimentos para um cliente se conectar a um desktop remoto são:

1. O cliente se conecta ao Servidor de Sessão, que redireciona a conexão para o Serviço de Conexão;
2. O Serviço de Conexão consulta no Serviço de Diretório quais são as configurações do desktop virtual atribuído ao usuário;
3. O Serviço de Conexão recupera o desktop virtual do usuário no Servidor de Virtualização. Caso o desktop não estiver em execução, ele é inicializado pelo Servidor de Virtualização;
4. O Servidor de Conexão retorna ao Servidor de Sessão as informações do desktop virtual recuperado;
5. O Servidor de Sessão repassa estas informações ao cliente;
6. O cliente se conecta ao desktop virtual.

2.5.3 VMware Horizon View

O VMWare Horizon View é a solução de VDI apresentada pela VMware (VMWARE, 2013b). Até a versão 5.1 da plataforma o nome do produto era apenas View, na versão atual 5.2 foi acrescentado o Horizon no nome. A Figura 2.8 mostra os componentes da solução e a relação entre eles (VMWARE, 2012c). Nela estão representados os dispositivos cliente, tais como tablets, *thin-clients*, computadores com sistema operacional Windows, Mac e Linux. O usuário através

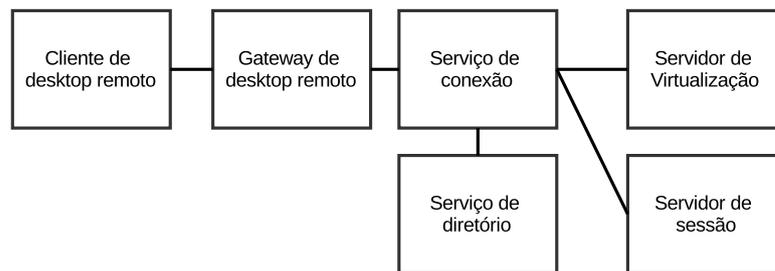


Figura 2.7: Exemplo da arquitetura de VDI da Microsoft. Adaptado de Wang (WANG; LIANG, 2011)

de um cliente de conexão (View Client) instalado nestes dispositivos, pode acessar o *desktop* virtual.

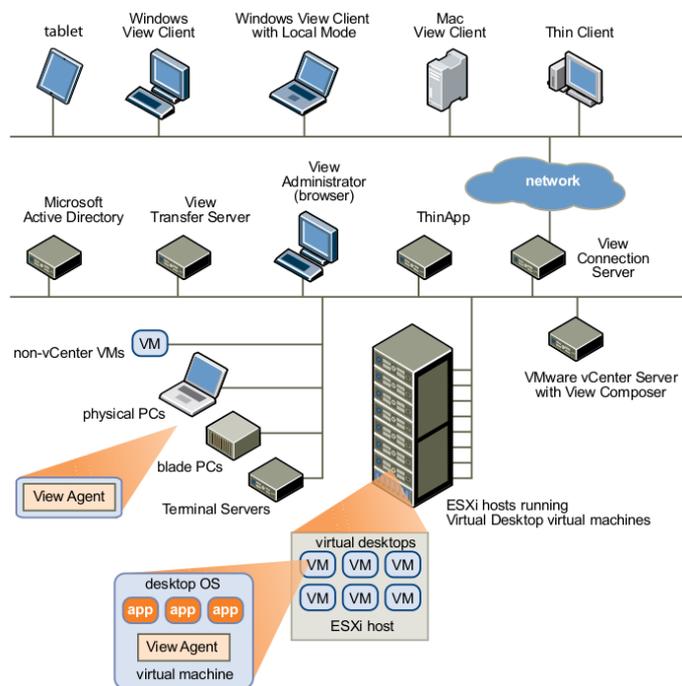


Figura 2.8: Exemplo da arquitetura de VDI da VMware Horizon View. Fonte (VMWARE, 2012c)

O *software cliente view* instalado nos dispositivos requer autenticação do usuário pelas credenciais do Active Directory ou via token. Após autenticado, o usuário pode selecionar o *desktop* virtual de uma lista de *desktops* autorizados para seu uso, além de poder escolher o protocolo de distribuição, PCoIP (padrão) ou RDP, e também a qualidade da imagem do *desktop* virtual.

O *software client com modo local* permite ao usuário carregar todo o arquivo do *desktop* virtual em seu dispositivo cliente, especificamente em sistemas operacionais Windows, e utilizar o *desktop* virtual sem conexão de rede. Nesse caso, o *desktop* é executado localmente. O servidor de transferência do view é o componente responsável por transferir os arquivos do

desktop virtual entre o *data center* e o *desktops* local do usuário, quando este utiliza o cliente view em modo local, além de também sincronizar o *desktop* virtual em modo local com o *desktop* do *data center*, replicando as alterações realizadas pelo usuário.

O *servidor de conexão do view* gerencia toda a comunicação entre o cliente view e o *desktop* virtual, também é responsável pelo processo de autenticação do usuário, vinculação do usuário a um *desktop* virtual específico e estabelecimento de uma conexão segura entre os usuários e os *desktops*. Dentro de uma infraestrutura de *desktop* virtual, são instalados dois ou mais servidores de conexão.

Quando o servidor de conexão é instalado, uma aplicação de administração web também é instalada, ela permite aos administradores gerenciar o servidor de conexão de qualquer lugar, sem precisar instalar uma aplicação no computador local.

O *agente do view* é instalado em todas as máquinas virtuais, sistemas físicos e servidores que serão utilizados como fontes para a virtualização de *desktops*. Em máquinas virtuais, este agente se comunica com o cliente para disponibilizar recursos como monitoramento de conexão, impressão e acesso a dispositivos USB conectados localmente. Se uma máquina virtual for a fonte de virtualização, primeiro o agente é instalado nela e em seguida ela pode ser usada como modelo para outras derivadas dela ou vinculadas a ela. Ao criar um conjunto de máquinas derivadas, nestas o agente é automaticamente instalado. O agente também propicia uma forma de autenticação única (do inglês *single sign-on*), em que os usuários são solicitados a fazer autenticação apenas quando se conectam no servidor de conexão view, não sendo solicitada uma segunda vez para conectar a um *desktop* virtual.

O *vCenter* é o serviço central de administração de todos os servidores instalados com a plataforma de virtualização VMware ESX/ESXi. Como mostra a Figura 2.9, o *vCenter* é o ponto central para configuração, provisionamento e gerenciamento de máquinas virtuais no *data center*, atuando na distribuição de máquinas virtuais nos servidores físicos e de armazenamento, na atribuição de CPU e recursos de memória para as máquinas virtuais.

Com relação à virtualização de *desktops*, são utilizados modelos para criar grupos de virtualização, além de hospedar outras máquinas virtuais da própria infraestrutura de virtualização de *desktops*, como o servidor de conexão, de Active Directory, e instâncias dele mesmo. Pode-se instalar o *serviço compositor* no mesmo servidor instalado o *vCenter*, um servidor Windows Server 2008, para assim poder criar grupos de clones de máquinas virtuais vinculadas (VMWARE, 2012a) (VMWARE, 2012b), em que dados duplicados são compartilhados de forma eficiente.

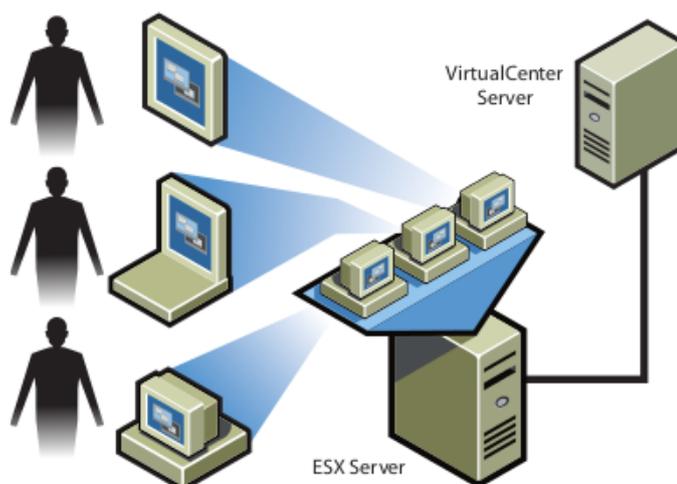


Figura 2.9: Exemplo da arquitetura de administração do vCenter. Fonte (VMWARE, 2008)

De acordo com o fabricante (VMWARE, 2013a), no gerenciamento de grupos é possível criar grupos com vários *desktops* virtuais, em que uma imagem deste *desktop* é utilizada como base. Ao utilizar uma máquina virtual, pode-se automatizar a criação de quantas cópias forem necessárias, configurando um número mínimo e máximo de *desktops* virtuais gerados no grupo.

O gerenciamento de grupos permite também aplicar políticas de gerencia em todos os *desktops* virtuais, tais como políticas de vinculação de usuários aos *desktops* do grupo. Neste caso existem duas formas. A primeira é a vinculação do usuário a um *desktop* virtual, em que ele sempre volta a se conectar, podendo personalizá-lo, instalar aplicações e armazenar dados. Na segunda forma, não existe vínculo, o *desktop* virtual pode ser apagado e recriado após o uso, e qualquer usuário pode se vincular a ele.

2.6 Emulação de redes com NetEm

NetEm é uma melhoria nas possibilidades de controle de tráfego do Linux que permite a adição de atraso, perda de pacote e outros cenários. Ele é construído utilizando os módulos de Qualidade de Serviço (QoS) e Serviços Diferenciados (Diffserv) do kernel do Linux. A motivação por trás do NetEm é prover uma maneira de reproduzir um ambiente de rede de longa distância em um laboratório. Netem é controlado pela ferramenta de linha de comando ‘tc’, que é parte do pacote de ferramentas iproute2 (HEMMINGER et al., 2005).

Emuladores de rede são ferramentas importantes para fazer a pesquisa e desenvolvimento relacionados com protocolos e aplicações de rede. Com a emulação de rede, é possível realizar testes de cenários de rede realistas, de forma controlada, o que não é possível usando apenas os dispositivos de rede reais, sem recursos de emulação. Um dos emuladores de rede mais

populares no mundo da pesquisa é NetEm. Esta ferramenta de código livre é amplamente utilizada em diferentes tipos de cenários de redes experimentais (JURGELIONIS, 2011).

O NetEm é constituído de duas partes, que são um pequeno módulo do kernel que cuida da disciplina de fila e uma ferramenta de linha de comandos para configurar estas filas. A comunicação entre a ferramenta e o módulo do kernel é feita por uma interface chamada Netlink. A Figura 2.10 mostra a arquitetura de disciplina de filas do Linux. Nela, a disciplina de filas está posicionada entre a saída da pilha de protocolo TCP/IP e a interface de rede.

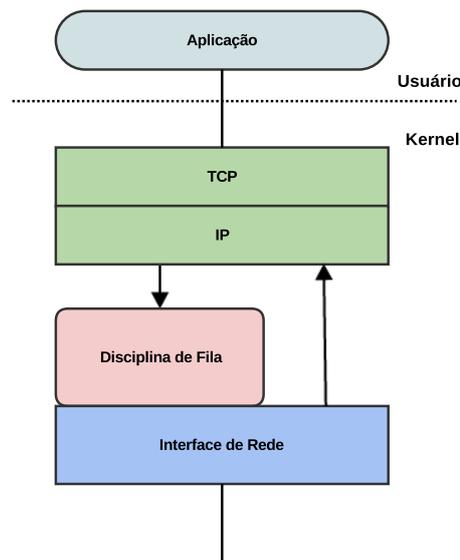


Figura 2.10: Arquitetura básica da disciplina de fila do Linux. Adaptado de (HEMMINGER et al., 2005)

A disciplina de fila tem duas interfaces principais. Uma interface enfileira pacotes para serem enviados e a outra interface libera os pacotes para o dispositivo de rede. A disciplina de fila segue a política de envio de pacotes passados como parâmetros. O Linux tem uma rica variedade de disciplinas de enfileiramento, priorização e políticas de controle de taxa, a disciplina padrão de enfileiramento é a FIFO.

O usuário especifica os parâmetros para o emulador através de argumentos para o comando 'tc'. Sem nenhum parâmetro, o NetEm funciona com uma fila FIFO, sem atraso, perda, duplicação ou reordenação de pacotes.

2.7 Avaliação subjetiva utilizando Mean Opinion Score

O *Mean Opinion Score* (MOS) é uma escala recomendada pelo ITU-T P.800 (SECTOR, 1996) para pontuar as opiniões subjetivas de usuários sobre uma aplicação avaliada. Foi origi-

nalmente criada para testes de julgamento de qualidade de transmissão de voz, mas também foi bastante utilizada para medir a qualidade de experiência para aplicações envolvendo VDI.

A escala de classificação vai de 1 (péssimo) a 5 (excelente) de acordo com a Tabela 2.3:

Tabela 2.3: Escala de valores do modelo MOS

Valor	Qualidade	Perturbação
1	Péssimo	Muito Importuno
2	Ruim	Importuno
3	Razoável	Um pouco importuno
4	Bom	Perceptível mas não importuno
5	Excelente	Imperceptível

Os testes subjetivos dão uma visão real do usuário frente ao sistema proposto. Em Wang (2012), os usuários da China Telecom avaliaram a experiência de uso de alguns aplicativos e classificaram de acordo com um modelo MOS adaptado. Em Nakazawa, Koizumi e Hirasawa (2012), os autores estudam a influência do uso de desktops virtuais no ensino a distância, nas condições das redes móveis do Japão. Neste estudo 10 usuários realizam diversos testes e classificam também de acordo com o MOS. Também em Casas (2013), o autor apresenta um estudo de qualidade de experiência aplicado em 52 usuários utilizando desktop virtual em um laboratório de testes dedicado, baseado na plataforma de VDI da Citrix. Aqui os usuários também utilizam o modelo MOS para pontuar a QoE. Este estudo fornece resultados sobre o comportamento do usuário quando confrontado com condições variáveis de rede com relação a seu comportamento em desktops locais padrão.

Capítulo 3

UMA AVALIAÇÃO EXPERIMENTAL DO USO DE DESKTOPS VIRTUAIS

Este capítulo descreve o projeto e o desenvolvimento dos experimentos realizados como parte deste trabalho. Inicialmente, foram selecionadas algumas aplicações típicas encontradas nas salas de aula informatizadas. Como os trabalhos relacionados apontam para a latência de rede como sendo o fator mais influente na qualidade de experiência do usuário, foram definidas faixas de latência a serem investigadas ao longo dos experimentos. A escolha dessas faixas se baseia em alguns testes preliminares realizados em ambiente real, incluindo-se algumas latências maiores com objetivo de investigar zonas de limite para o teste *in vitro*. O capítulo também apresenta os resultados obtidos e as análises realizadas.

3.1 Objetivos dos experimentos

De modo geral, os experimentos realizados mostram a influência de parâmetros de desempenho da infraestrutura (tais como latência de rede e perda de pacotes) na percepção de qualidade por parte do usuário. Idealmente, as diferentes zonas de latência deverão incluir valores típicos observados em ambientes reais, além de evidenciar zonas de latência limítrofes (por exemplo, o que usuário considera como qualidade aceitável e ruim), para diferentes tipos de aplicações. O projeto de experimentos deve levar em conta a noção de qualidade sob o ponto de vista da percepção do usuário. Portanto, os resultados devem se basear na avaliação feita com usuários reais. Ao final, espera-se estabelecer as diferentes faixas de valores dos parâmetros que resultam em diferentes níveis de qualidade segundo a percepção dos usuários.

3.2 Seleção das aplicações

A experiência do usuário em utilizar um *desktop* virtual depende do tipo de aplicação a ser executada. Uma ferramenta que lida com gráficos tridimensionais, por exemplo, demanda mais recursos ou recursos diferentes daqueles que outra que manipula textos. Para cobrir um bom número de ferramentas representativas utilizadas pelos usuários, foi feito um levantamento de quais softwares são instalados nas salas de aula informatizadas da Secretaria Geral de Informática da UFSCar, utilizadas para as aulas práticas, e depois classificados de acordo com seu uso.

As aplicações instaladas nos laboratórios foram classificadas em ferramentas de uso geral e de uso específico. As ferramentas de uso geral são utilizadas pela maioria dos usuários para elaborar documentos, navegar na Internet e assistir a vídeos. As selecionadas para tal aplicação foram o Microsoft Word, Windows Media Player e Internet Explorer. Já as ferramentas de uso específico são aquelas que apenas alguns grupos de usuários precisam utilizar, de acordo com a especificidade de sua área. As ferramentas escolhidas estão voltadas para cálculo matemático e estatístico e desenvolvimento e estão instaladas em todos os laboratórios informatizados do campus da UFSCar em São Carlos. Foram usados o Matlab (MATHWORKS, 2014), utilizado para realização de cálculo numérico e estatística, e o Eclipse (ECLIPSE FOUNDATION, 2014), utilizado para desenvolvimento de software.

Uma vez identificadas as ferramentas, elas foram instaladas em um *desktop* físico e um outro virtualizado. O primeiro foi utilizado como base de comparação, pois as aplicações estavam instaladas localmente e não dependiam das condições da rede. O *desktop* virtualizado serviu de modelo a ser distribuído para os usuários envolvidos no experimento. A distribuição do *desktop* virtual consiste no processo de se criar várias instâncias de um modelo de *desktop*, associar este conjunto de instâncias aos usuários e, através de um programa cliente, o usuário acessa e utiliza este *desktop*. No final do uso da instância, ela volta ao seu estado original, como se não tivesse sido utilizada.

Para cada ferramenta foi feito um planejamento das atividades que o usuário devia executar, explorando uma das funcionalidades de cada aplicação. Os testes estão relatados no Apêndice B. A Tabela 3.1 mostra na primeira coluna o nome da aplicação e na segunda coluna o objetivo do teste aplicado.

- **Microsoft Word:** o objetivo deste teste é avaliar a qualidade de experiência do usuário em aplicações de digitação de textos. Caso os recursos de hardware alocados não sejam suficientes, pode ocorrer atraso entre o tempo de digitação e o aparecimento do caractere

Tabela 3.1: Tipo de teste executado em cada aplicação

Aplicação	Teste
Microsoft Word	Qualidade de interação para digitação
Internet Explorer	Teste de rolagem de tela
Microsoft Media Player	Qualidade de vídeo
Matlab	Teste de arrastar e soltar e renderização 3D
Eclipse	Teste de execução de funções gráficas

digitado. Do mesmo modo, durante o uso de um *desktop* virtual as condições de rede podem degradar ocasionando atrasos de resposta, trazendo desconforto ao usuário.

- **Microsoft Internet Explorer:** o objetivo deste teste foi a movimentação da tela através do botão de rolagem do mouse ou pelas teclas *Page Up* e *Page Dow* do teclado. A rolagem de tela exige atualização rápida da imagem enviada ao usuário utilizando um *desktop* virtual, implicando em um envio de grande quantidade de pixels do *desktop* virtual ao cliente, e as condições da rede podem influenciar no tempo de resposta, logo influenciando na qualidade de experiência do usuário.
- **Microsoft Media Player:** foi escolhido um vídeo de 30 segundos com resolução de 640 por 360 pixels, com imagens reais. As imagens exploram mais a quantidade de cores do que uma animação. O vídeo foi assistido pelo usuário na proporção 1:1, desta forma não ocupando a tela toda.
- **Matlab:** foi explorado o tempo de resposta da ação de arrastar e soltar um polígono gerado por uma função matemática, implicando em uma nova renderização tridimensional.
- **Eclipse:** foi testado o tempo de resposta de algumas funções gráficas bidimensionais.

3.3 Qualidade de serviço da rede da UFSCar

Em um ambiente típico de uma universidade com múltiplos prédios espalhados pelo campus ou com mais de um campus, é comum ter diversos tipos de redes e tecnologias de acesso com diferentes atrasos de rede. Para podermos prever o comportamento de uma ambiente de *desktop* virtual em diferentes condições de rede, foram definidas seis faixas de latência, reproduzindo as principais condições de testes realizados nos laboratórios do campus da UFSCar em São Carlos, como também as latências até o campus de Araras e de Sorocaba.

Estudos prévios mostram que a latência e a perda de pacotes são os parâmetros que mais influenciam na qualidade de experiência. Em Tolia, Andersen e Satyanarayanan (2006), é es-

tudado o tempo de resposta de diferentes tipos de aplicações em um *desktop* remoto acessado através do aplicativo VNC. Foram testadas latências diferentes em redes com largura de banda de 10 e 100 Mbps. Os autores concluíram que aplicações mais interativas são mais sensíveis a latência de rede e não à largura de banda. Em Lai e Nieh (2006), os autores comparam diversos sistemas para *thin-clients*, com foco no consumo de banda e latência em redes locais e remotas. Concluem que embora muitos destes sistemas tratem da eficiência no uso de largura de banda, a latência é o fator principal no desempenho em redes remotas.

Os testes foram executados na rede cabeada e também na rede sem fio de alguns laboratórios de ensino informatizados do campus da UFScar São Carlos, como também até o *gateway* de entrada da rede de Sorocaba e Araras. No campus de São Carlos foram escolhidos um laboratório da área Sul, dois na área central e um na área norte. Na Tabela 3.2 está indicado onde cada um destes laboratórios estão localizados.

Tabela 3.2: Localização dos laboratórios de informática no campus UFSCar São Carlos

Área	Localização
Área Sul	AT2
Área Central	SIn, AT4, BCO
Área Norte	AT6

O teste de latência na rede cabeada de cada um dos laboratórios foi feito utilizando um laptop com sistema operacional Linux Ubuntu (CANONICAL, 2013) e a ferramenta ping. Os testes foram feitos em dias letivos, durante um período de 4 horas. Os resultados mostraram que o tempo médio de latência entre os laboratórios e a rede com a infraestrutura de *desktop* virtual é menor que 1 ms, com níveis de perda de pacote negligíveis, mostrando uma excelente qualidade da rede cabeada.

Também foi feito um teste de latência entre a rede onde está localizada a infraestrutura de *desktops* virtuais até o roteador de borda dos campus UFSCar de Sorocaba e Araras. Todos os campus da UFSCar estão conectados ao *backbone* da rede Ipê, que é uma infraestrutura de rede Internet voltada para a comunidade de ensino e pesquisa, mantida pela RNP. Um servidor Linux Ubuntu conectado na mesma rede da infraestrutura de *desktops* virtuais enviou durante 4 horas pacotes ICMP também utilizando a ferramenta ping, e o resultado foi uma latência média de 13 ms até Sorocaba e 9,7 ms até Araras. Não foram detectadas perdas de pacotes nos testes. Também em ambos os casos temos uma conexão estável.

Foram feitos alguns testes na rede sem fio do campus da UFSCar em São Carlos, nos prédios de aulas teóricas AT2, AT4 e AT6, na SIn e Biblioteca Comunitária (BCO). Para o teste foi

utilizado um laptop com o sistema operacional Linux Ubuntu e a ferramenta ping. O laptop foi posicionado perto dos pontos de acesso, a uma distância de dez metros, sem obstáculos físicos como paredes e vidros. O laptop então foi conectado a rede sem fio, que recebeu sua configuração de rede via DHCP. Com a conexão estabelecida, foram coletados os tempos de latência. Na rede sem fio do AT2, a latência média foi de 541 ms com perda de pacotes de 4%. Na rede sem fio do AT4 os resultados foram de 863 ms de latência média, com 23% de perda de pacotes. Na rede do AT6, a latência média foi de 64,5 ms. Na BCO a latência média foi de 24 ms com perda de 9%. Por fim, no saguão da SIn, a latência média ficou em 3,5 ms sem perda de pacotes.

Para cobrir as faixas de latência verificadas e também investigar zonas de limite, prevendo assim acessos externos, da rede pública, isto é, da casa do usuário, foram escolhidas as latências mostradas na Tabela 3.3.

Tabela 3.3: Latências utilizadas nos experimentos

Grupos	Latências
1	menor que 1 ms
2	20 ms
3	65 ms
4	120 ms
5	250 ms
6	400 ms

Para os experimentos com latência de 400 ms, foi incluído 1% de perda de pacotes a fim de simular condições mais difíceis de redes, como o usuário acessando de sua casa.

3.4 Descrição do ambiente de experimentos

Para os testes com os usuários e de capacidade de desktops, primeiramente foi criada uma infraestrutura de virtualização de *desktops*, composta por dois servidores físicos, *storage* e *switches*. Em seguida foram instalados outros dois servidores físicos para monitorar a rede e gerar atrasos na rede de teste. A infraestrutura de virtualização foi montada no *data center* da Secretaria Geral de Informática (SIn), em separado do ambiente de produção. A infraestrutura de geração de atrasos e monitoramento também foi feita no mesmo prédio, e os testes com os usuários foram feitos do lado desta infraestrutura. Os testes de capacidade foram feitos utilizando um dos laboratórios da SIn.

De acordo com a Figura 3.1, a infraestrutura de virtualização é composta por dois servidores

sendo um deles um HP com 2 CPUs Intel com quatro núcleos de 2.53GHz, com 32 Gigabytes de memória e o outro servidor Dell com 2 CPUs Intel com quatro núcleos de 2GHz e 8 Gigabytes de memória. Ambos os servidores tem uma de suas interfaces de rede Ethernet conectadas nas portas de 100Mbps de um *switch* CISCO. Estas portas do *switch* foram configuradas para que o tráfego de rede aconteça de forma isolada através de um rede virtual (VLAN).

Para a montagem dos sistemas de arquivos nos *desktops* virtuais, foi utilizado um *storage* com sessenta discos. Os servidores foram conectados a ele por uma interface dedicada denominada *Host Bus Adapter* (HBA), que é uma placa PCI Express de velocidade de 4 Gbps instalada nos servidores, e delas partem fibras óticas até o *switch* óptico do *storage*. Nesse *storage* foi reservada uma unidade lógica de armazenamento de 1 Terabyte para abrigar o sistema de arquivos dos *desktops* e servidores virtuais.

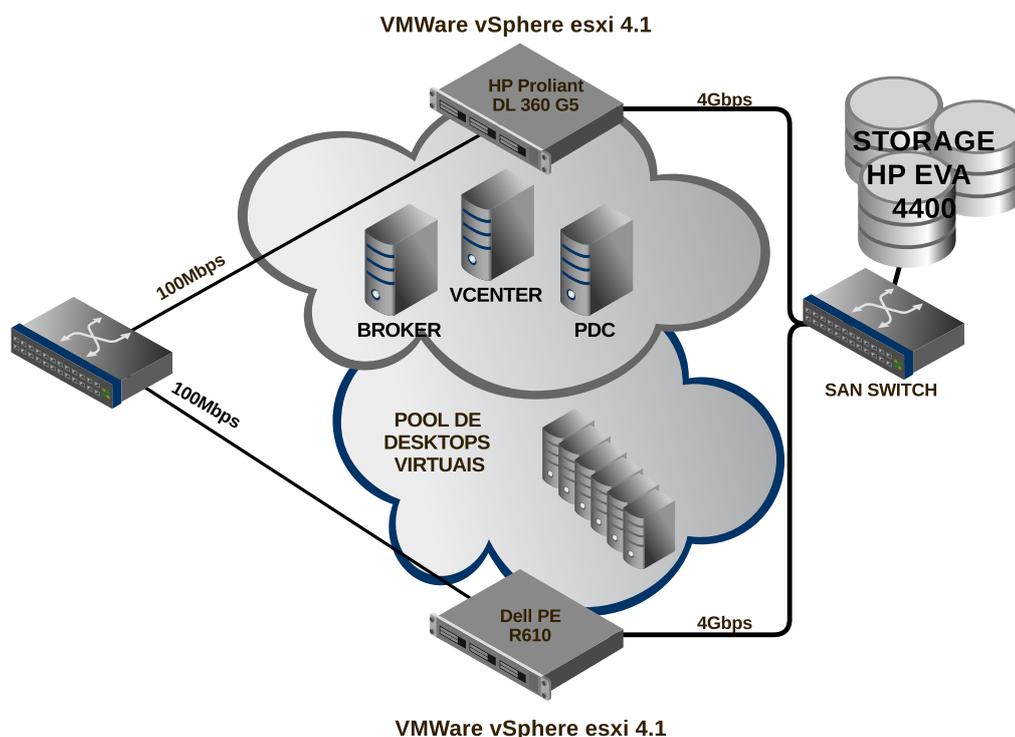


Figura 3.1: Infraestrutura de Desktops Virtuais

Nos servidores hospedeiros foi instalado o sistema operacional VMWare ESXi 4.1 Enterprise Plus, e ambos foram agrupados em um único cluster de virtualização. Também foram instalados e configurados dentro desta infraestrutura três servidores virtuais Windows 2008 R2 com VMWare Vcenter, VMWare View e Microsoft Active Directory, estes que formam a base para controle da virtualização de *desktops*. Estes três servidores foram hospedados em separado no servidor físico HP, enquanto que o outro servidor físico ficou reservado apenas para executar as instâncias de *desktops* virtuais.

Para a infraestrutura do ambiente de testes, ilustrada na Figura 3.2, dois servidores físicos foram preparados, sendo um dedicado à coleta de tráfego e outro para a emulação de redes com diferentes latências. O servidor dedicado à coleta tem uma interface de rede conectada a uma porta do *switch* de rede onde também estão os outros dois servidores de virtualização. Esta porta foi configurada para espelhar o tráfego de rede dos dois servidores, e a ferramenta *tcpdump* foi utilizada para coletar os pacotes desta interface para posterior análise. O servidor de emulação tem por objetivo criar condições controláveis de diferentes níveis de qualidade de serviço da rede, também está configurado para atribuir endereços IP privados a rede de teste além de fazer reescrita nos endereços de rede (NAT). Ele possui duas interfaces de rede, uma conectada à infraestrutura de *desktops* virtuais e outra conectada à intranet de testes. Neste servidor, diferentes condições de redes foram simuladas com a ferramenta Netem (HEMMINGER et al., 2005), como visto na Figura 3.3.

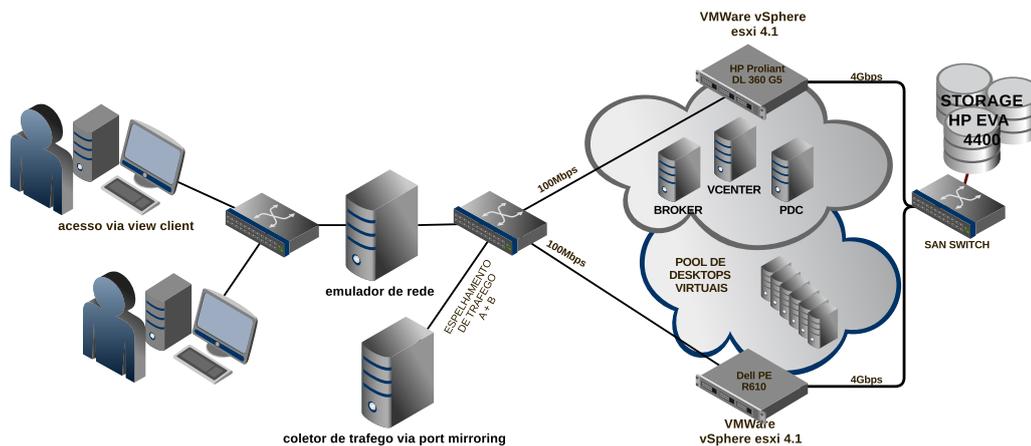


Figura 3.2: Infraestrutura do ambiente de teste

3.5 Projeto de experimentos

Foram realizados dois tipos de experimentos. O primeiro experimento avalia com o usuário a qualidade de experiência utilizando um desktop virtual. O segundo tipo de experimento observa o consumo de hardware da infraestrutura de desktops virtuais.

3.5.1 Testes para a avaliação da qualidade de experiência dos usuários

O primeiro conjunto de experimentos tem como objetivo avaliar a qualidade de experiência dos usuários ao realizar algumas tarefas em um *desktop* virtual. Como base para os experimentos, também foi incluído um teste com *desktop* local.

Os participantes selecionados para o experimento foram os mais próximos, de acordo com a técnica de amostragem não-probabilística por conveniência (WOHLIN, 2000): dez membros da equipe técnica da própria Secretaria Geral de Informática, que atuam na área de desenvolvimento e infraestrutura de redes participaram do experimento de forma voluntária.

Para o usuário avaliar a qualidade de experiência foi utilizado o método subjetivo *Mean Opinion Score (MOS)* apresentado nos trabalhos relacionados, no Capítulo 2. Cada vez que o usuário completava um teste, ele classificava a experiência de 1 até 5.

Para os testes, o computador pessoal ficou parecido com as configurações do *desktop* virtual. Para ambos foram reservados 4 núcleos de processamento e 4 gigabytes de memória RAM e 40 gigabytes de espaço em disco. Foram instalados o sistema operacional Windows 7 Enterprise 64 bit, e as aplicações necessárias para os testes, como o Microsoft Word, Internet Explorer, Windows Media Player, Matlat e Eclipse.

Para obter os resultados para posterior análise, foram realizados 350 testes a partir da definição de 35 experimentos distintos, repetidos dez vezes. Destes 35 experimentos, 30 foram definidos utilizando *desktops* virtuais e outros 5 utilizando *desktops* tradicionais. Para cada um dos 30 experimentos foram executados 10 testes, logo somando 300 testes com *desktops* virtuais. Os outros 5 experimentos com *desktop* local foram executados em um conjunto de 10 testes, somando 50. A Figura 3.3 mostra os números relatados.

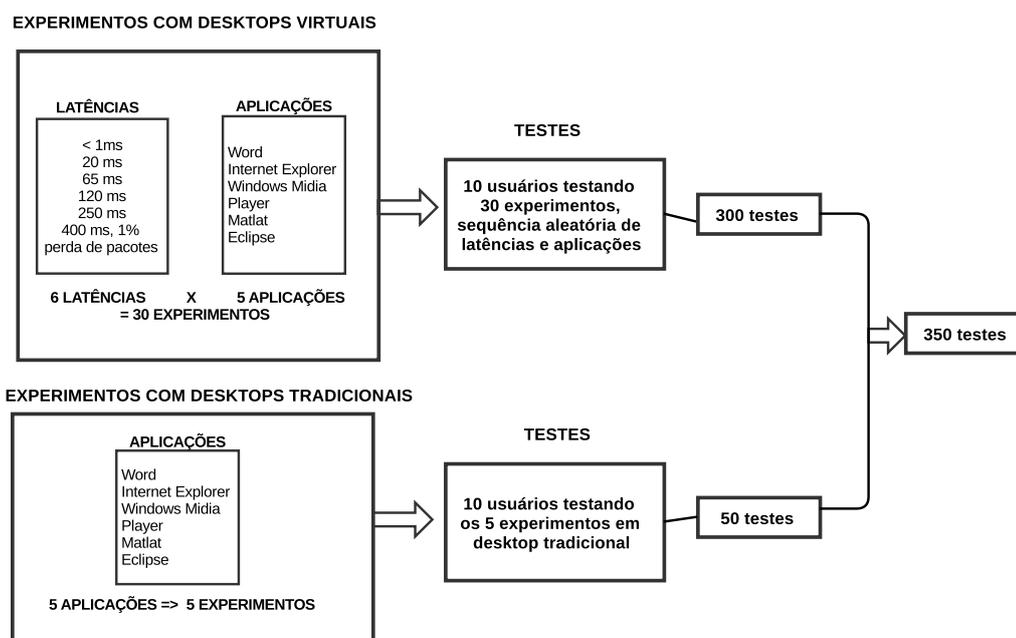


Figura 3.3: Quantidade de testes realizados

Para a execução dos experimentos, inicialmente foi definido que o primeiro conjunto de

testes com usuário seria feito utilizando um *desktop* local. O objetivo dos experimentos com *desktop* local é ter uma referência do usuário ao utilizar as aplicações em um computador pessoal, em que tanto o sistema operacional quanto as aplicações funcionem no mesmo hardware, sem dependerem das condições de rede.

Após os testes com *desktop* local, foi definido que o usuário passaria para os testes com *desktop* virtual nos dias posteriores. O intervalo dos testes entre o *desktop* local e o virtual foi necessário para tentar eliminar ou diminuir qualquer tendência do usuário em comparar com o desempenho do *desktop* local.

A sequência de testes escolhida também foi aleatória. Para criá-la de forma aleatória, primeiramente foi feito um *script* para associar cada uma das 5 aplicações a um tipo de latência, criando arquivos com uma estrutura, ainda sequencial, contendo o nome da aplicação, a latência e a porcentagem de perda de pacotes, conforme mostra a Figura 3.4A. Como durante os testes o usuário poderia perceber o aumento da latência e vir a classificar sua experiência de forma tendenciosa, já supondo que deveria existir uma piora ou melhora a cada vez que se repetia a sequência das aplicações, foi aplicado um algoritmo simples para embaralhar a sucessão de testes de forma a não manter nem a mesma sequência de aplicações nem de latências. A Figura 3.4B mostra um exemplo de como ficou o arranjo de testes.

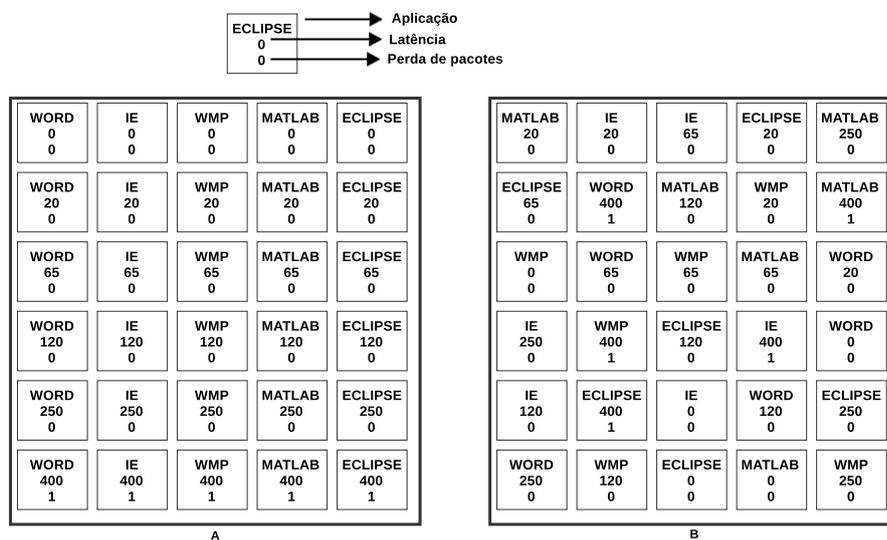


Figura 3.4: Arranjo de teste sequencial (A) e não sequencial (B)

Os testes com os usuários foram feitos de forma individual com o acompanhamento de uma pessoa para orientar qual aplicação ele deveria executar. Tanto no teste com *desktop* local quanto com o virtual, o usuário ficou em uma sala apenas acompanhado de um instrutor. O papel do instrutor era orientar o usuário sobre qual aplicação ele deveria abrir, quais procedimentos ele deveria executar e por fim, fechar a aplicação e responder ao questionário, no qual o usuário

validava a experiência vivenciada, em um escala de 1 a 5.

3.5.2 Testes de capacidade

Os testes de capacidade foram realizados utilizando a mesma infraestrutura do experimento anterior, sem a presença de usuários. O objetivo dos testes é observar o comportamento da infraestrutura de virtualização quando vários desktops virtuais são utilizados ao mesmo tempo, tal como acontece em uma aula, em que os alunos utilizam uma aplicação ao mesmo tempo.

Os testes foram feitos com 1, 5, 10, 15, 20 e 25 desktops virtuais simulando a interação do usuário com cada uma das aplicações: Word, Internet Explorer, Windows Media Player, MatLab e Eclipse. Para simular a interação do usuário com as aplicações, foi utilizado um software chamado Pulover's Macro Creator (PULOVERS MACRO CREATOR, 2014) que grava e repete as ações de teclado e mouse. Foram um total de 35 testes.

Para cada um dos 35 testes foram coletados dados de CPU, memória e acesso a unidade de armazenamento de dados do storage. A ferramenta que coletou estes dados foi a do próprio sistema de virtualização VMWare ESXi. Destes dados foi feita a média de utilização de cada um dos recursos.

3.6 Resultados obtidos

3.6.1 Resultados dos testes de qualidade de experiência

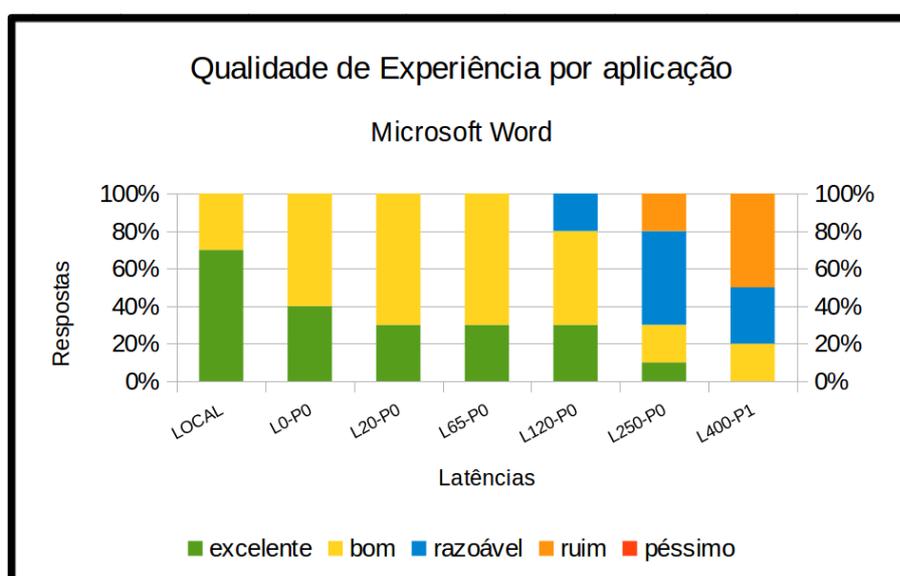
Os resultados obtidos são mostrados em gráficos de colunas separados por aplicação. Cada um deles mostra a porcentagem de usuários que classificaram a sua qualidade de experiência dentro dos valores de 1 a 5, de acordo com o modelo MOS, em que o valor 1 significa uma experiência péssima e 5 significa uma experiência excelente. Em cada gráfico o eixo da abscissa começa com o rótulo denominado *LOCAL*, representando o valor da qualidade de experiência do usuário ao utilizar um desktop local. Os outros rótulos representam os valores de latência e perda de pacotes de acordo com a Tabela 3.4.

A Figura 3.5 exibe os resultados da qualidade de experiência do usuário ao digitar um texto utilizando o Microsoft Word. Nela observamos que 80% dos usuários avaliam como boa ou excelente a qualidade de experiência até uma latência de 120 ms, enquanto apenas 20% classificam como razoável, isto é, um pouco importuna, mas ainda aceitável. A partir de 250 ms observamos que 20% já classificam como ruim. A 400 ms, metade classifica como ruim e outra

Tabela 3.4: Rótulo dos eixos das abcissas dos gráficos de qualidade de experiência por aplicação

Rótulo	Significado
L0-P0	Latência menor que 1 ms e níveis de perda de pacote negligíveis
L20-P0	Latência de 20 ms e níveis de perda de pacote negligíveis
L65-P0	Latência de 65 ms e níveis de perda de pacote negligíveis
L120-P0	Latência de 120 ms e níveis de perda de pacote negligíveis
L250-P0	Latência de 250 ms e níveis de perda de pacote negligíveis
L400-P1	Latência de 400 ms e níveis de perda de pacote de 1%

parte como boa a razoável. Como estes dados podemos concluir que para o tipo de aplicação de escrita de documentos, até uma latência de 250 ms temos grande parte dos usuários ainda satisfeitos com o serviço, porém com 400 ms e com perda de pacotes de 1% metade dos usuários já consideram seu uso importuno. Também podemos observar que ninguém classificou como muito importuno, o que seria o valor 1. Nesse caso, podemos concluir que apesar da aplicação estar ruim de manipular, o fato de estar funcionando é considerado positivo por parte do usuário.

**Figura 3.5: Qualidade de experiência por aplicação - Digitação de textos**

Na Figura 3.6 o usuário assiste a um vídeo de trinta segundos com imagens reais. No gráfico percebemos que até a latência de 250 ms temos uma ótima aceitação por todos os usuários. Em 80% dos casos, é classificado de boa a excelente e apenas 20% classificam como razoável. Porém a 400 ms e perda de pacotes em torno de 1%, 90% dos usuários classificam de ruim a péssimo. Ao contrário do gráfico anterior relacionado a digitação de texto, apesar do vídeo executar sem travar, a perceptível perda de áudio e vídeo é classificada como muito ruim. O bom funcionamento do vídeo em alta latência de 250 ms pode ser explicado pela utilização de buffer e pela latência constante. Logo, o início da execução do vídeo tinha um atraso de cerca

de 250 ms mas depois este atraso não era mais perceptível.

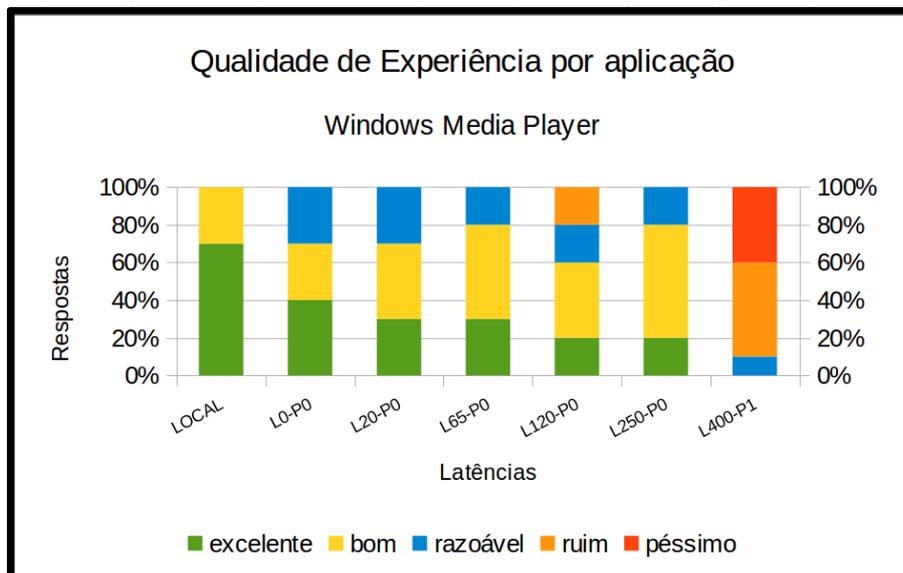


Figura 3.6: Qualidade de experiência por aplicação - Vídeo

A Figura 3.7 exibe a classificação do usuário ao navegar por uma tela de um navegador de internet. Nestes resultados podem ser destacados que mesmo com latência menor que 1 ms, 43% dos usuários já classificam a experiência com razoável, aumentando este patamar para 50% em latência de até 65 ms. Já na latência de 120 ms observamos que apenas 30% classificam com razoável, 10% como ruim e 30% como excelente, o que parece estranho já que com latências menores a classificação foi pior. Podemos supor que neste caso ocorreu algum fator não observado que tenha influenciado nos resultados. Já na latência de 400 ms, 80% dos usuários classificaram como ruim a péssimo, sem condições de utilização.

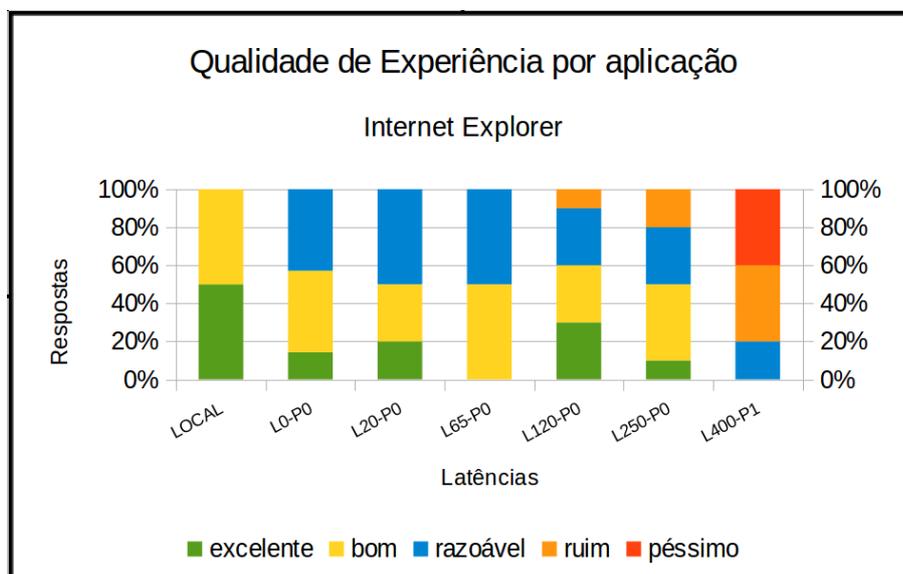


Figura 3.7: Qualidade de experiência por aplicação - Internet Explorer

Na Figura 3.8 os usuários classificaram sua experiência ao executar vários programas gráficos que acompanham o pacote do Eclipse e os resultados foram bastantes positivos até a latência de 250 ms. Diferente de todas as outras aplicações, 50% dos usuários classificaram como excelente a qualidade de experiência tanto na utilização dos aplicativos executados em desktop local como em latência em torno de 1 ms. Como no caso do navegador, houve uma classificação fora do padrão no experimento com latência de 65 ms, em que 10% dos testes foram considerados ruins. Nos testes com latência de 400 ms e perda de pacotes, 90% classificaram entre ruim e péssimo. Conclui-se que a aplicação eclipse pode ser utilizada com latência de até 250 ms.

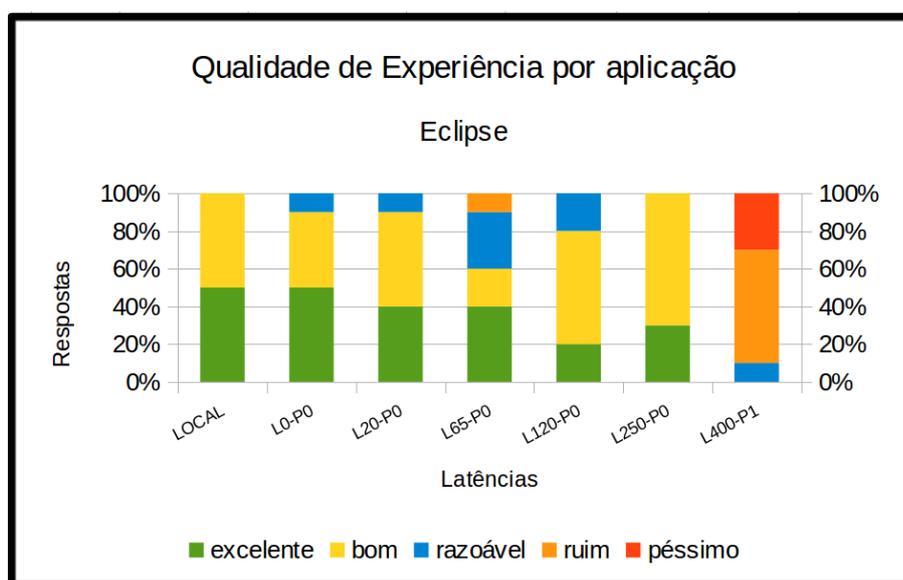


Figura 3.8: Qualidade de experiência por aplicação - Eclipse

Na Figura 3.9 os usuários classificam sua experiência em manipular um gráfico tridimensional gerado pelo Matlab. Em latência de até 65 ms a qualidade da experiência foi considerada em média boa e de 120 ms a 250 ms considerada razoável. Para latência de 400 ms, 70% dos usuários classificaram sua experiência de ruim a péssima. Também neste caso, concluímos que o Matlab também pode ser utilizado em latências de até 250 ms em média.

A Figura 3.10 mostra o valor médio da qualidade de experiência dos usuários nas diferentes aplicações quando executadas em desktop local e também nas diferentes condições de rede testadas. Esta média descarta um valor máximo e outro mínimo. Observamos que a qualidade de experiência na menor latência *L0-P0* diminui cerca de meio ponto comparado a qualidade de experiência em um *desktop* local, exceto para a aplicação de navegação na Internet com diferença de um ponto. Até a latência de 250 ms todas as aplicações estão acima de três pontos. Em latência de 400 ms com perda de pacotes todas as aplicações foram classificadas como ruins de serem utilizadas.

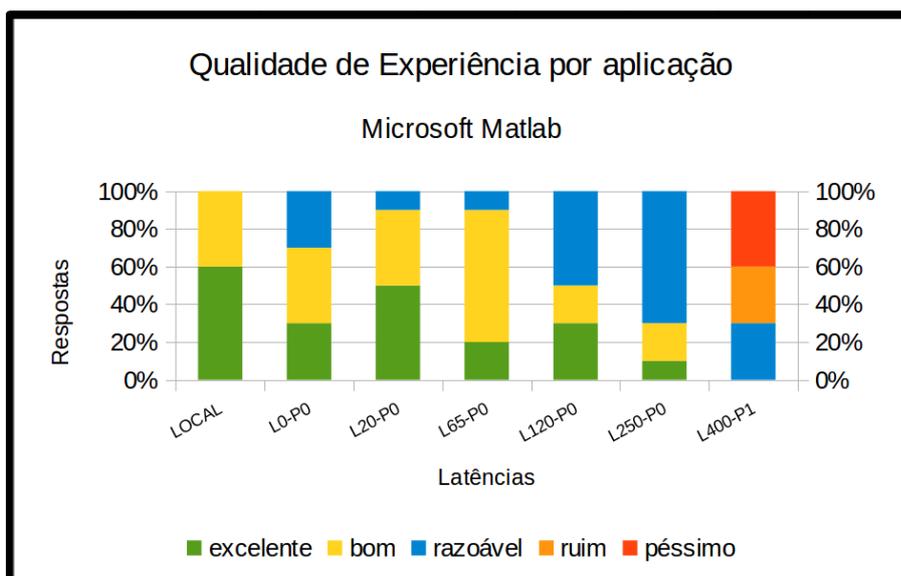


Figura 3.9: Qualidade de experiência por aplicação - Matlab

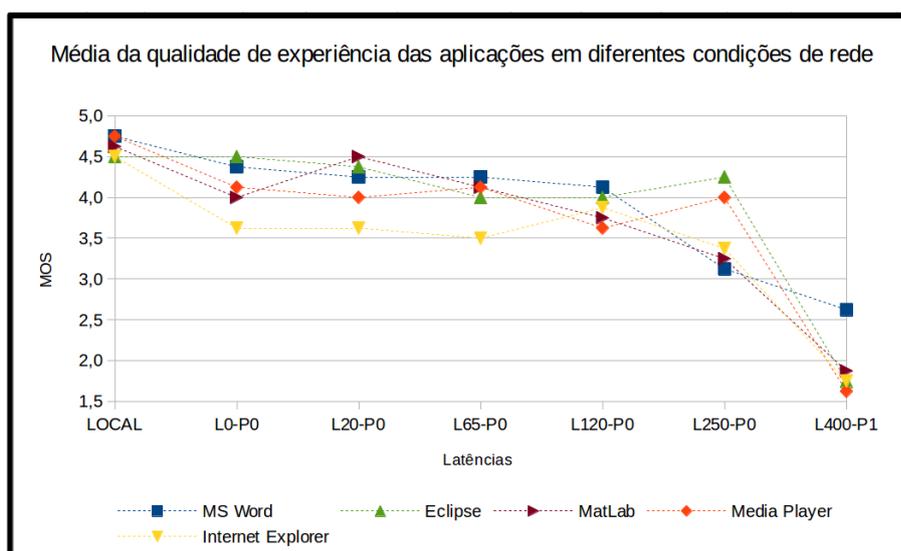


Figura 3.10: Média da qualidade de experiência das aplicações em diferentes condições de rede

A Figura 3.11 mostra a média da taxa de transferência de dados de cada uma das aplicações nos testes com usuário com latência menor que 1 ms. O fluxo de *upstream* é o fluxo de pixels que são enviados do *desktop* virtual até o software cliente onde o usuário interage. O fluxo de *downstream* são os dados enviados do cliente para o *desktop* virtual, como a posição do mouse, código das teclas acionadas no teclado. O gráfico mostra que os fluxos de *upstream* são muito pequenos, indo de 29kbps à 58kbps. Já os fluxos de *downstream* variam bastante. Na aplicação Word, pela pouca atualização de pixels, temos um pequeno fluxo de 85 kbps. As aplicações MatLab e Eclipse tiveram fluxos próximos, entre 640 kbps a 776 kbps. Já as aplicações Internet Explorer e Media Player tiveram fluxo médio de 2 Mbps à quase 6 Mbps. Nesse caso, a transferência de vídeo gerou três vezes mais pixels para atualização do que a

rolagem de tela do navegador, misturando texto e imagens.

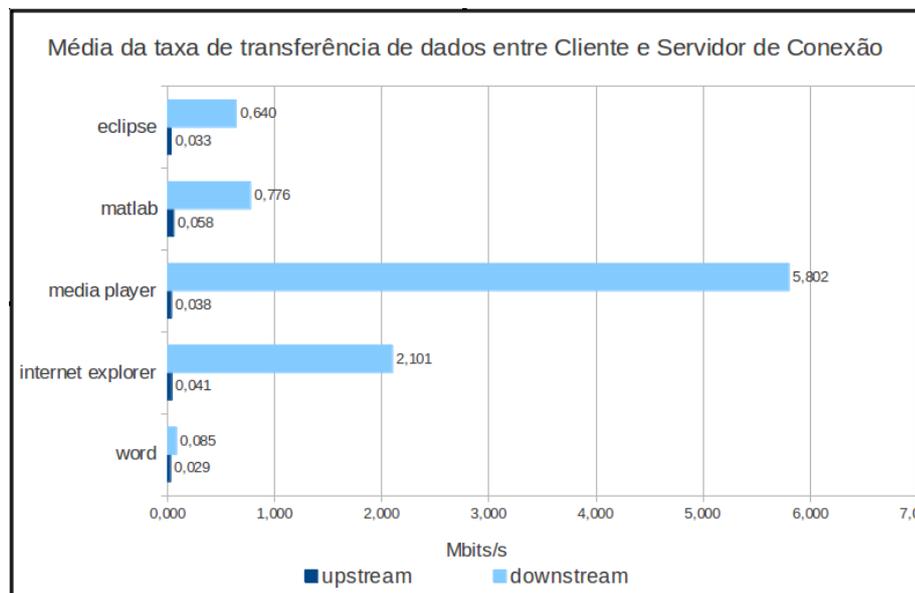


Figura 3.11: Média da taxa de transferência de dados entre Cliente e Servidor de Conexão

3.6.2 Resultados dos testes de capacidade

As Figuras mostram a taxa de utilização (média) dos recursos computacionais simulando a utilização de um usuário em cada uma das aplicações.

A Figura 3.12 mostra qual foi a utilização do processador do servidor hospedeiro ao executar cada tipo de aplicação com carga variando de 1 até 25 desktops virtuais. Observamos que as aplicações Media Player e Eclipse com 20 desktops ativos alcançam 90% de utilização de memória e com 25 desktops chegam a 100% de utilização.

A Figura 3.13 mostra a porcentagem de utilização de memória do servidor hospedeiro. Notamos que a curva de crescimento do consumo de memória decai gradativamente com a crescente utilização de mais *desktops* virtuais. Isso acontece pois quando o hospedeiro encontra-se com pouca memória disponível, o hipervisor utiliza técnicas de recuperação de memória para liberar memória ociosa ou inativa do servidor físico alocada para máquinas virtuais e a atribui a outras máquinas virtuais que vão utilizá-las efetivamente.

A Figura 3.15 mostra a taxa média de operações de E/S de cada aplicação, variando de 1 até 25 *desktops* executando ao mesmo tempo. Podemos notar um grande número de operações de leitura nos testes com 25 e 20 desktops, em todas as aplicações, porém menos evidente no caso do Word. Esta grande quantidade de operações de E/S com os testes com 25 e 20 desktops pode ser explicada pela ordem que aconteceram os testes. O primeiro teste foi com 25 desktops, foi a

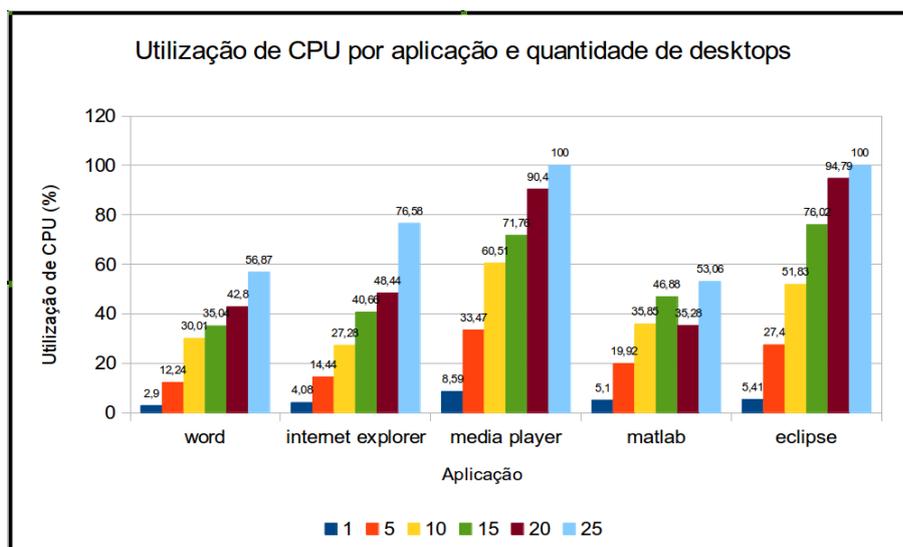


Figura 3.12: Média de utilização de CPU por aplicação e quantidade de desktops virtuais

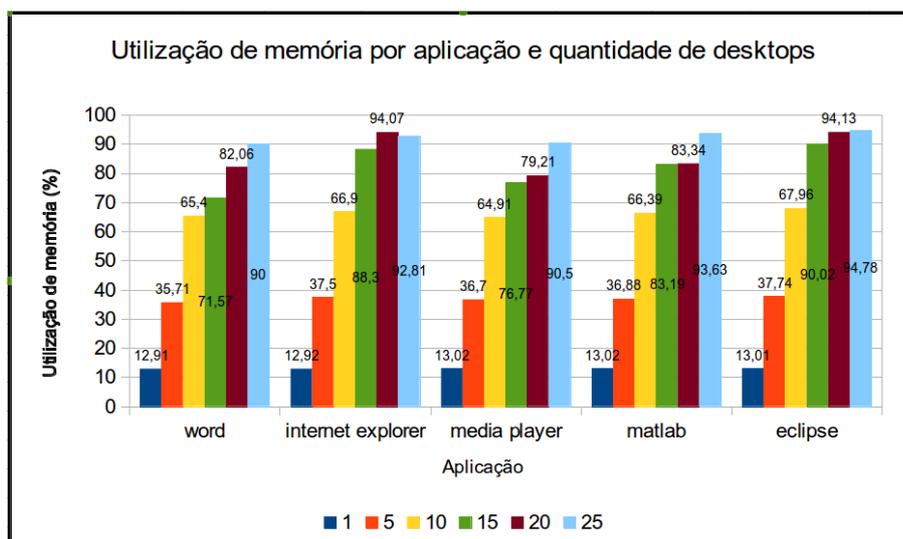


Figura 3.13: Média de utilização de memória por aplicação e quantidade de desktops virtuais

primeira vez que as aplicações foram executadas, logo ocorreram várias leituras em disco para executar as aplicações e criar, por exemplo, diretórios de projeto, como é o caso do Eclipse. Os testes com 20 desktops ocorreram no dia posterior, assim como os outros na sequência. A Figura 3.14, mostra que para carregar a aplicação Eclipse pela primeira vez em 25 desktops, foram em média 1033 operações de E/S por segundo, uma média de 41 operações por desktop virtual. Estes picos de E/S devem ser tratados pela infraestrutura de VDI.

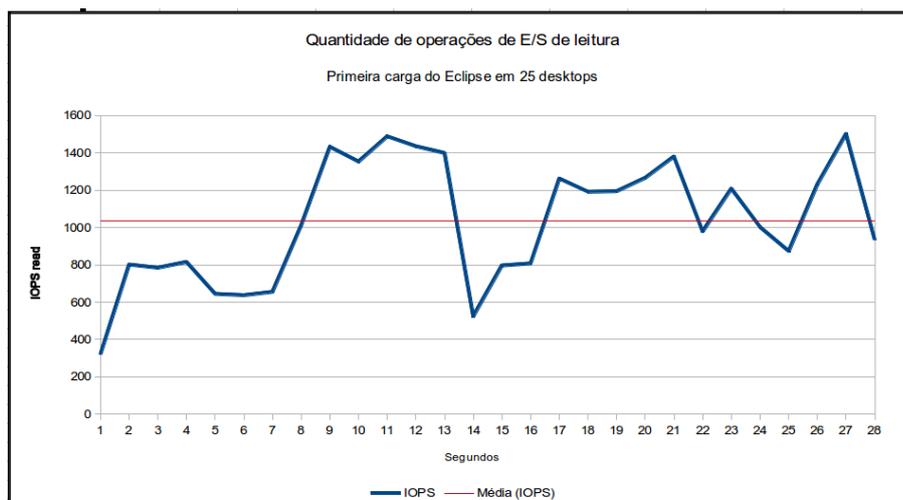


Figura 3.14: Operações de leitura durante a carga do Eclipse em 25 desktops virtuais

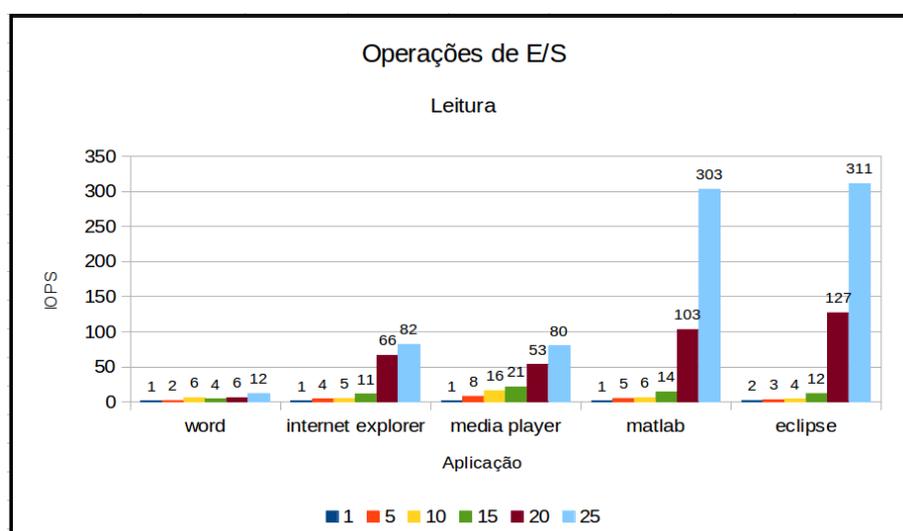


Figura 3.15: Média de operações de E/S de leitura por aplicação e quantidade de desktops virtuais

3.7 Conclusão

Este capítulo apresentou os experimentos realizados na infra-estrutura montada na Secretaria de Informática da UFSCAR. A suite de ferramentas VMWare View Horizon foi instalada em dois servidores no *data center* da Secretaria de Informática. A infra-estrutura é composta por um servidor HP com 2 processadores de 4 núcleos e 32 GB de memória, que executou serviços básicos da suite de ferramentas da VMware, tais como o *broker*, o *vCenter* e o *Active Directory*. Um segundo servidor Dell com 2 processadores de 4 núcleos cada e 8 GB de memória foi utilizado para executar os *desktops* virtuais.

Um primeiro conjunto de experimentos avaliou a qualidade de experiência dos usuários com cinco softwares comumente utilizados nos laboratórios da UFSCAR: Word, Internet Explorer,

Media Player, Matlab e Eclipse. Os testes foram feitos em um desktop local, e em desktops virtuais com latência de rede de menos de 1 ms, 20 ms, 65 ms, 120 ms, 250 ms e 400 ms. Foram executados, portanto, 35 experimentos, cada um replicado 10 vezes. Para cada experimento uma nota média foi extraída. Os experimentos mostram que, em linhas gerais, os usuários se mostraram satisfeitos para latências de até 250 ms, apesar de perceberem queda na qualidade da experiência desde os testes com latência menor que 1 ms, quando comparado ao desktop local. A exceção seria o Eclipse, que foi igualmente classificado com a mesma qualidade, comparado ao desktop local. Também percebemos que a aplicação Word demanda pouco processamento, memória e capacidade de rede. Este resultado fornece um parâmetro de desempenho mínimo da rede para que *desktops* virtuais sejam utilizados de forma satisfatória. Isto pode ser facilmente atingido quando o usuário encontra-se dentro de um campus da UFSCAR. Mesmo estando fora do campus (por exemplo, acessando de sua residência), a qualidade de experiência pode ser satisfatória se este limite for respeitado.

Um segundo experimento revelou que processador é o recurso crítico no caso do Media Player e Eclipse, permitindo até 25 *desktops* virtuais. O uso de memória foi mais elevado com Internet Explorer e Eclipse, chegando próximo a 95% de utilização. Este experimento traz inúmeras informações úteis para um planejamento de capacidades. Por exemplo, observa-se que como regra geral, que é possível executar de 2 a 3 *desktops* virtuais por núcleo de processador com relativo grau de conforto dos usuários. Vimos no gráfico 3.13 que os mecanismos de gestão de memória do hipervisor permitem atribuir as máquinas virtuais mais memória do que a memória física existente, técnica denominada sobrecarga de memória. Que a infraestrutura deve tratar picos de operações de E/S, como é o caso do Eclipse fazendo em média 1033 OIOPS.

Capítulo 4

CONCLUSÃO

A utilização de desktops virtuais transfere a complexidade de administração do ambiente computacional para uma infraestrutura centralizada, geralmente composta por um cluster de servidores conectado à Internet. Essa centralização tem demonstrado diversas vantagens, sobretudo na melhoria da utilização dos recursos computacionais, redução do consumo de energia, simplificação das tarefas e melhoria da eficiência de equipes de administração e manutenção de desktops, facilitando a realização de backups e atualizações de software, e tornando o ambiente disponível e acessável a partir de qualquer local através da Internet e por meio de dispositivos móveis. Empresas, de modo geral já vem adotando esse modelo há algum tempo. Nas universidades, porém, o uso de desktops virtuais ainda é bastante incipiente.

O presente trabalho tem como objetivo principal avaliar o uso de uma infraestrutura de desktops virtuais para uso didático em laboratórios tomando como base o cenário da UFSCAR. Para isto, uma suite de ferramentas para virtualização de desktops foi implantada no datacenter da Secretaria de Informática da UFSCAR. A suite escolhida foi o VMWare View Horizon, compatível com a infraestrutura de software disponível no datacenter. Analisando a bibliografia especializada, nota-se que a latência de rede tem sido determinante para a qualidade de experiência dos usuários. Por esse motivo, foram conduzidos experimentos para identificar as latências típicas nas instalações dos três *campi* da UFSCAR. Em seguida, foram executados experimentos nos quais um grupo de usuários utilizou e avaliou a qualidade do uso de desktops virtuais com um conjunto de aplicações típicas, com diferentes latências de rede. Este conjunto de experimentos permitiu identificar, para cada aplicação, qual é o impacto da latência de rede na percepção dos usuários. Um segundo conjunto de experimentos avaliou o consumo de recursos necessários para a execução de desktops virtuais. Os resultados deste segundo experimento poderão subsidiar o planejamento de capacidades futuramente em projetos de implantação de desktops virtuais na UFSCAR.

De modo geral, o presente trabalho contribui de forma bastante prática, fornecendo informações relevantes para que gestores de instituições de ensino planejem a implantação de desktops virtuais para uso em laboratórios e atividades didáticas.

4.1 Trabalhos futuros

Como possíveis trabalhos futuros, pretendemos medir o consumo de recursos nos elementos adjacentes da infraestrutura, tais como o *broker*, o *textitActive Directory* e *vCenter*. Tais componentes apresentam menor consumo de recursos e são mais escaláveis. Entretanto, seria interessante caracterizar o consumo de recursos nesses pontos. Se faz necessário, também, desenvolver um modelo específico para avaliação da qualidade de serviço em VDI, como é o caso do modelo E, que fornece uma medida objetiva para sistemas de comunicação por voz (ITU-T, 2013). Seria interessante desenvolver algo semelhante para a entrega de desktops virtuais.

REFERÊNCIAS

- AMAZON. *Technical Deep Dive: ICA Protocol and Acceleration*. 2013. http://s3.amazonaws.com/legacy.icmp/additional/ica_acceleration_0709a.pdf. Último acesso em: 13/11/2013.
- APACHE. *Apache VCL*. 2013. <http://vcl.apache.org/>. Último acesso em: 13/11/2013.
- BUYYA, R. et al. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, Elsevier, Amsterdam, v. 25, n. 6, p. 599–616, Junho 2009.
- CALYAM, P. et al. Utility-directed resource allocation in virtual desktop clouds. *Computer Networks*, Elsevier, v. 55, n. 18, p. 4112–4130, 2011.
- CANONICAL. *Ubuntu Desktop*. 2013. <http://http://www.ubuntu.com/desktop>. Último acesso em: 12/11/2013.
- CASAS, P. et al. Quality of experience in remote virtual desktop services. In: IEEE. *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*. [S.l.], 2013. p. 1352–1357.
- CITRIX. *XenDesktop Tech Info*. 2013. <http://www.citrix.com/products/xendesktop/tech-info.html#hdx>. Último acesso em: 13/11/2013.
- DASILVA, D.-A. et al. Enabling green it through building a virtual desktop infrastructure. In: IEEE. *Semantics, Knowledge and Grids (SKG), 2012 Eighth International Conference on*. [S.l.], 2012. p. 32–38.
- ECLIPSE FOUNDATION. *Eclipse*. 2014. <https://www.eclipse.org/>. Último acesso em: 21/01/2014.
- HEMMINGER, S. et al. Network emulation with netem. In: CITESEER. *Linux Conf Au*. [S.l.], 2005. p. 18–23.
- ITU. *T.120 : Data protocols for multimedia conferencing*. 2009. <http://www.itu.int/rec/T-REC-T.120>. Último acesso em 14/01/2014.
- ITU-T. *About the Ohio Board of Regents*. 2013. <https://www.itu.int/ITU-T/studygroups/com12/emodelv1/>. Último acesso em: 29/01/2014.
- JURGELIONIS, A. et al. An empirical study of netem network emulation functionalities. In: IEEE. *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*. [S.l.], 2011. p. 1–6.

- KUSIC, D. et al. Power and performance management of virtualized computing environments via lookahead control. In: . [S.l.: s.n.], 2009. p. 1–15.
- LAI, A. M.; NIEH, J. On the performance of wide-area thin-client computing. *ACM Transactions on Computer Systems (TOCS)*, ACM, v. 24, n. 2, p. 175–209, 2006.
- LI, P.; TODERICK, L. W.; LUNSFORD, P. J. Experiencing virtual computing lab in information technology education. In: ACM. *Proceedings of the 10th ACM conference on SIG-information technology education*. [S.l.], 2009. p. 55–59.
- MARSTONA, S. et al. Cloud computing the business perspective. In: . [S.l.: s.n.], 2011. p. 176–189.
- MATHWORKS. *MatLabg*. 2014. <http://www.mathworks.com/products/matlab/>. Último acesso em: 21/01/2014.
- MELL P; GRANCE, T. The nist definition of cloud computing. *(Draft)?Recommendations of the National Institute of Standards and Technology - Special publication*, 2011.
- MICROSOFT. *Remote Desktop Protocol:Clipboard Virtual Channel Extension*. 2013. [http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/\[MS-RDPECLIP\].pdf](http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/[MS-RDPECLIP].pdf). Último acesso em 14/01/2014.
- NAKAZAWA, M.; KOIZUMI, D.; HIRASAWA, S. The influence of qos on e-learning environment under virtual desktop infrastructure. In: IEEE. *Communications, Computers and Applications (MIC-CCA), 2012 Mosharaka International Conference on*. [S.l.], 2012. p. 174–178.
- NCSU. *Virtual Computing Lab*. 2013. <http://vc1.ncsu.edu/>. Último acesso em: 13/11/2013.
- OARNET. *About OARnet*. 2013. <https://oar.net/about/>. Último acesso em: 21/11/2013.
- OARNET. *OARnet History*. 2013. <https://oar.net/about/history/>. Último acesso em: 21/11/2013.
- OARNET. *VDPilot Project*. 2013. <http://vdpilot.oar.net/>. Último acesso em: 21/11/2013.
- OHIOHIGTERED. *About the Ohio Board of Regents*. 2013. <https://www.ohiohighered.org/board/>. Último acesso em: 21/11/2013.
- PULOVERS MACRO CREATOR. *Pulovers Macro Creator*. 2014. <http://www.macrocreator.com>. Último acesso em 20/01/2014.
- RNP. *Rede Ipe*. 2013. <http://http://www.rnp.br/ipe/>. Último acesso em: 01/02/2014.
- SECTOR, I. T. U. T. S. *Methods for Subjective Determination of Transmission Quality*. [S.l.]: International Telecommunication Union, 1996.
- SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. *Operating system concepts*. [S.l.]: J. Wiley & Sons, 2009.

- SRIDHARAN, M. et al. Defragmentation of resources in virtual desktop clouds for cost-aware utility-optimal allocation. In: IEEE. *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. Washington: IEEE, 2011. p. 253–260.
- TERADICI. *Tecnologia PCoIP explicada*. 2013. <http://www.teradici.com/pcoip-technology.php/>. Último acesso em: 5/09/2013.
- TOLIA, N.; ANDERSEN, D. G.; SATYANARAYANAN, M. Quantifying interactive user experience on thin clients. *Computer*, IEEE, v. 39, n. 3, p. 46–52, 2006.
- VMWARE. *VDI Server Sizing and Scaling*. 2006. http://www.vmware.com/pdf/vdi_sizing_vi3.pdf. Último acesso em 18/12/2013.
- VMWARE. *Linked Virtual Machines*. 2012. http://pubs.vmware.com/vsphere-50/index.jsp#com.vmware.wssdk.pg.doc_50/PG_Ch11_VM_Manage.13.4.html. Último acesso em: 16/10/2013.
- VMWARE. *Linked Virtual Machines*. 2012. http://pubs.vmware.com/vsphere-50/index.jsp#com.vmware.wssdk.pg.doc_50/PG_Ch11_VM_Manage.13.4.html. Último acesso em: 16/10/2013.
- VMWARE. *VMware Horizon View Architecture Planning*. 2012. http://www.vmware.com/support/pubs/view_pubs.html. Último acesso em: 16/10/2013.
- VMWARE. *Managing Desktop Pools from a Central Location*. 2013. <http://pubs.vmware.com/view-52/index.jsp#com.vmware.view.planning.doc/GUID-C54115CD-2A2B-4307-A80B-973F59E8722F.html>. Último acesso em: 16/10/2013.
- VMWARE. *Portal VMWare Brasil*. 2013. <http://www.vmware.com/br>. Último acesso em: 17/10/2013.
- WANG, F. et al. Benchmark driven virtual desktop planning: A case study from telecom operator. In: IEEE. *Cloud and Service Computing (CSC), 2012 International Conference on*. [S.l.], 2012. p. 204–211.
- WANG, J.; LIANG, L. Survey of virtual desktop infrastructure system. 2011.
- VMWARE. Addressing desktop challenges. *VMWare*, 2008.
- WOHLIN, C. et al. *Experimentation in software engineering: an introduction*. MA, USA: Kluwer Academic Publishers Norwel, 2000. ISBN 0792386825.
- ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: State-of-the-art and research challenges. In: . [S.l.]: Springer, 2010. v. 1, p. 7–18.

GLOSSÁRIO

HDX – *Citrix High Definition User Experience*

ICA – *Independent Computing Architecture*

IaaS – *Infraestrutura como Serviço*

NIST – *US National Institute of Standards and Technology*

PCoIP – *Teradici PC-over-IP*

PaaS – *Plataforma como Serviço*

RDP – *Remote Desktop Protocol*

RDS – *Remote Desktop Service*

RD – *Desktop Remoto*

ROI – *Retorno sobre investimento*

RVD – *Desktop Virtual Remoto*

SaaS – *Software como Serviço*

TCM – *Total cost management*

VDI – *Virtual Desktop Infrastructure*

Apendice A

OARNET E O PROJETO VDPiLOT

O projeto VDPiLot, iniciado pelo Conselho Consultivo do Conselho de Regentes de Ohio, é um estudo de viabilidade de uma "nuvem de desktops virtuais" para os laboratórios e salas de aula das instituições conectadas a OARnet(OARNET, 2013c).

O Conselho de Regentes de Ohio, é um conselho consultivo nomeado pelo governador do estado de Ohio, com o conselho e consentimento do Senado. Algumas das responsabilidades do conselho incluem fazer um relatório anual sobre as condições da educação no estado de Ohio e emitir uma avaliação de desempenho do reitor. O conselho também é responsável em orientar o reitor sobre questões que afetam a educação(OHIOHIGTERED, 2013).

A Rede de Recursos Acadêmicos de Ohio (OARnet), foi criada pelo Conselho de Regentes de Ohio, para oferecer aos pesquisadores o primeiro acesso "online" ao Centro de Supercomputação de Ohio, em Columbus, nos Estados Unidos.

Atualmente o backbone da OARnet, de acordo com a figura A.1 tem quase 3 mil quilômetros de fibra óptica escura, fornecendo conectividade as faculdades e universidades de Ohio, escolas de educação infantil e ensino médio, emissoras públicas, centros médicos acadêmicos, e organizações parceiras de pesquisa, estaduais e federais(OARNET, 2013a).



Figura A.1: Backbone atual e futuro da OARnet. Fonte (OARNET, 2013b)

Apendice B

PROCEDIMENTOS DE TESTES DE APLICAÇÕES

B.1 Procedimento de teste com Microsoft Word

Utilizando o aplicativo MS Word o usuário abre um arquivo de texto existente e deve digitar um parágrafo no começo do documento, rolar o texto, salvar e fechar. Durante este teste o usuário deve observar a velocidade de execução das tarefas e a qualidade gráfica exibida. Manter todas as outras aplicações fechadas no desktop virtual.

Procedimentos

1. Abrir o MS Word clicando no atalho na área de trabalho.
2. Quando aparecer a tela intitulada Microsoft Office Activation Wizard, aperte a tecla C.
3. Maximize a janela do Word caso seja necessário.
4. Clicar em File -> Open -> Documents, selecionar o arquivo CF1998 e clicar em Open.
5. Caso aparecer um aviso em amarelo na parte superior da janela, intitulado Protected View, clique em Enable Editing.
6. No início do primeiro parágrafo, que se encontra incompleto, digitar o texto: *Nós, representantes do povo brasileiro, reunidos em Assembléia Nacional Constituinte para instituir em Estado democrático, destinado a assegurar o exercício dos direitos sociais*
7. Salvar o documento clicando em File -> Save as, coloque o nome diferente do original.
8. Fechar o Word clicando em File -> Exit.
9. Responda ao questionário.

B.2 Procedimento de teste com Microsoft Media Player

Utilizando o Windows Media Player, o usuário deve assistir os vídeos em HD e o outro de média qualidade (480px). Durante a reprodução, o usuário deve notar a qualidade do vídeo, a fluência e suavidade da reprodução.

Manter todas as outras aplicações fechadas no desktop virtual.

Procedimentos

- 1.Clique no ícone do Windows Media Player na barra de ferramentas.
- 2.Clique em Vídeos.
- 3.Clique com o botão esquerdo sobre o vídeo Wonders of Nature.mp4 e clique em Play.
Assista o vídeo.
- 4.Feché o Windows Media Player clicando na X no canto superior direito da janela.
- 5.Responda ao questionário.

B.3 Procedimento de teste com Microsoft Internet Explorer

Utilizando o Internet Explorer, abrir o site de notícias <http://www.folha.uol.com.br/> e navegar pelos seus menus. Durante o procedimento o usuário deve notar a fluência e suavidade durante a rolagem do site.

Manter todas as outras aplicações fechadas no desktop virtual

Procedimentos

- 1.Abra o Internet Explorer clicando no seu atalho na barra de tarefas.
- 2.Após carregar o site, maximize a tela do IE com F11.
- 3.Abrir o site de notícias <http://www.folha.uol.com.br>.
- 4.Role a tela até o final utilizando o botão scrow do mouse e depois retorne.
- 5.Passe o mouse sobre cada um dos menus de notícias e veja que cada um deles abrirá submenus.
- 6.Feché o Internet Explorer.

7.Responda ao questionário.

B.4 Procedimento de teste com Eclipse

O propósito deste teste é simular uma típica utilização do Eclipse no desenvolvimento de aplicações java com o uso de recursos gráficos. Neste teste o usuário irá executar alguns projetos exemplo do eclipse e interagir com eles, devendo observar o tempo de resposta da interface e suavidade na execução.

Manter todas as outras aplicações fechadas no desktop virtual.

Procedimentos

- 1.Abrir o Eclipse através do seu atalho na área de trabalho.
- 2.Caso aparecer uma janela intitulada WorkSpace Launcher, clique no botão OK.
- 3.A janela do Eclipse estará aberta com os pacotes de exemplos, expanda o pacote org.eclipse.swt.examp
- 4.Expanda a pasta src.
- 5.Clique no pacote org.eclipse.swt.examples.graphics.
- 6.Clique em notão Run, e uma janela intitulada Run As abrirá, selecione Java Application e clique em OK, e o exemplo será executado.
- 7.Na janela do exemplo intitulada Graphics Example clique em Alpha e em seguida em Next, passando por cada um dos exemplos a cada 2 segundo em média.
- 8.Ao chegar no último exemplo, feche a janela Graphics Example.
- 9.Fechar o Eclipse clicando em File -> Exit.
- 10.Responda ao questionário.

B.5 Procedimento de teste com Matlab

O propósito deste teste é simular uma interação com o Matlab, manipulando um dos modelos tridimensionais de exemplo que ele oferece. O usuário deve observar neste teste o quão rápido e suave o matlab responde enquanto interage com o modelo 3D.

Manter todas as outras aplicações fechadas no desktop virtual.

Procedimentos:

1. Iniciar o Matlab com um duplo clique no seu atalho na área de trabalho.
2. Na janela Command Windows clicar em Demos, uma outra janela intitulada Help será aberta.
3. Expandir a pasta Gallery e clicar em Spherical Surface Harmonic.
4. Clicar em Run this demo.
5. Como o modelo carregado, manipular com zoom, rotação e movimentos.
6. Fechar a janela do modelo e depois o Matlab clicando em File -> Exit Matlab.
7. Responder ao questionário.