

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UM MODELO PARA ESTIMATIVA DO
CONSUMO DE ENERGIA DE DESKTOPS
VIRTUAIS**

PEDRO WILSON BIGNATTO JUNIOR

ORIENTADOR: PROF. HERMES SENGER

São Carlos – SP

21 de junho de 2016

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UM MODELO PARA ESTIMATIVA DO
CONSUMO DE ENERGIA DE DESKTOPS
VIRTUAIS**

PEDRO WILSON BIGNATTO JUNIOR

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. Hermes Senger

São Carlos – SP

21 de junho de 2016

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar
Processamento Técnico
com os dados fornecidos pelo(a) autor(a)

B593m Bignatto Junior, Pedro Wilson
Um modelo para estimativa do consumo de energia
de desktops virtuais / Pedro Wilson Bignatto Junior.
-- São Carlos : UFSCar, 2016.
89 p.

Dissertação (Mestrado) -- Universidade Federal de
São Carlos, 2016.

1. Computação em nuvem. 2. Desktops virtuais. 3.
Estimativa energética em desktops virtuais. 4.
Controladores de nuvem. 5. Eficiência energética em
desktops virtuais. I. Título.



UNIVERSIDADE FEDERAL DE SÃO CARLOS
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a defesa de dissertação de mestrado do candidato Pedro Wilson Bignatto Junior, realizada em 07/03/2016.

Prof. Dr. Hermes Senger
(UFSCar)

Prof. Dr. Hélio Crestana Guardia
(UFSCar)

Prof. Dr. Mário Dantas
(UFSC)

Certifico que a sessão de defesa foi realizada com a participação à distância do membro Prof. Dr. Mário Dantas e, depois das arguições e deliberações realizadas, o participante à distância está de acordo com o conteúdo do parecer da comissão examinadora redigido no relatório de defesa do aluno Pedro Wilson Bignatto Júnior.

Prof. Dr. Hermes Senger
Presidente da Comissão Examinadora
(UFSCar)

Dedico este trabalho a minha família que tanto amo, meus irmãos Rodrigo Amaro Bignatto e Karina Amaro Bignatto que me ajudaram desde que nasci a chegar onde estou agora.

Claro que eu não esqueceria da pessoa que é a razão de tudo nessa vida para mim, minha querida e amada mãe Maria Angelica Amaro Bignatto, a quem tudo devo nessa vida, obrigado do fundo do coração.

Este trabalho eu também dedico a quem me ensinou muito em pouco tempo que estivemos juntos nessa vida, mas sei que olha por mim todos os dias, meu querido pai Pedro Wilson Bignatto.

AGRADECIMENTOS

Esta dissertação é um passo fundamental na minha carreira profissional e estou muito feliz por ter concluído meu trabalho de mestrado.

Agradeço a todos que me ajudaram neste projeto, meu Professor, orientador e amigo Hermes Senger, com quem trabalho junto a mais de três anos e sempre me ajudou em tudo. Meus amigos que fiz durante a graduação, da pós e meus amigos do trabalho que me deram forças para chegar até aqui.

Gostaria de agradecer também minha namorada Marcela que tem me apoiado tanto nessa empreitada e meus amigos da minha cidade natal José Bonifácio, pequena cidade do interior paulista onde tive a honra de conhecer o melhor conjunto de amigos que eu poderia ter, são vários mais eles merecem ser citados, pois todos são parte fundamental da minha vida, obrigado: Affonso, Bruno, Fernando, Heitor, João, Jonas, José, Marcelo, Matheus, Olavo, Rafael B., Rafael E., Renan, Rodrigo, Tales, Tiales, Tomaz, Sérgio e Vinícius meu primo.

Muito Obrigado a todos!

RESUMO

A virtualização de recursos computacionais está se tornando cada vez mais frequente para diversas finalidades. Um exemplo é a virtualização de *desktops*, em que a estação de trabalho do usuário é virtualizada e executada em um centro de dados, e são entregues aos seus usuários sob a forma de um serviço. Os usuários podem acessar seus *desktops* virtuais via Internet, de qualquer lugar e a qualquer hora, utilizando clientes simplificados como *smartphones* ou *tablets*. Além da mobilidade, a virtualização de *desktops* traz enorme redução nos custos de gerência e administração. Esta dissertação propõe um modelo que permite estimar o custo energético de um serviço de virtualização de *desktops*. Esse modelo tem por finalidade auxiliar a tomada de decisões sobre o provisionamento de recursos, melhorando a eficiência energética. Para essa estimativa, são considerados os custos energéticos que compõem a entrega do *desktop* virtual, desde a instanciação até sua execução no centro de dados. A partir dos experimentos realizados, aplicando o modelo proposto, foi possível concluir que para aplicações com taxa de transmissão de quadros e comandos baixa ou regular, como um navegador web por exemplo, consome-se menos energia para executar um VD do que executar um *desktop* local. Isso acontece porque o custo energético de execução é amortizado por meio do compartilhamento de recursos entre os diversos usuários conectados ao mesmo servidor, além do baixo custo de transmissão. A melhor eficiência energética obtida foi resultado de instanciar um VD em um *datacenter* (DC) com acesso a rede local (LAN), em que o consumo energético de transmissão por bit é menor e tão pequeno que compensou no decorrer do tempo, inclusive, o custo energético da migração do VD entre diferentes DCs.

Palavras-chave: Computação em Nuvem, *desktops* virtuais, estimativa energética em *desktops* virtuais, controladores de nuvem, eficiência energética em *desktops* virtuais

ABSTRACT

The virtualization of computing resources is becoming increasingly frequent for several purposes. An example is desktop virtualization, in which the user's workstation is virtualized, executed in a data center, and delivered to users as a service. Users can access their virtual desktops over the network, from anywhere, anytime, using simplified clients as smartphones or tablets. Besides mobility, desktop virtualization brings significant reduction in management costs and administration of desktops. In the present work we propose a model that estimates the energy consumption of a desktop virtualization service. The proposed model aims to assist decision-making on the provision of resources and improving energy efficiency. The model takes into account energy consumption during all the lifecycle of a virtualized desktop since the instantiation. From the experiments, applying the model, it was concluded that for applications with low transmission rate of frames and commands, such as a web browser, it is better to instantiate an VD instead of running as a local desktop. It happens because the energy cost of running is amortized through the sharing of resources between the various connected users into the server, in addition to the low cost of transmission. The best energy efficiency has been obtained as a result of instantiating an VD on a datacenter (DC) with accessible through a local area network (LAN), in which the energy consumption per bit transmission is sufficiently low to compensate energy cost of VD migration between different DCs.

Keywords: Cloud computing, desktops virtuals, energy estimate on virtual desktops, cloud controllers, energy efficiency in virtual desktops

LISTA DE FIGURAS

2.1	Diferentes tipos de máquinas virtuais (SMITH; NAIR, 2005)	17
2.2	VDS, adaptado de (BOSING; KAUFMANN, 2012) e <i>commons.wikimedia.org</i> . . .	19
2.3	Consolidação e migração de VDs, adaptado de <i>mirantis.com</i>	20
2.4	Modelos de computação em Nuvem	23
2.5	Tipos de serviço da computação em nuvem	23
2.6	Arquitetura de softwares VMware, adaptado de (WANG; LIANG, 2011)	28
2.7	Domínios e Hypervisors, adaptado de (CARISSIMI, 2008)	29
2.8	Arquitetura de softwares Citrix, adaptado de (WANG; LIANG, 2011)	30
2.9	Cabeçalho protocolo ICA, adaptado de (KANTER, 1998)	31
2.10	Funções protocolo ICA, adaptado de (KANTER, 1998)	32
2.11	Processo de entrega protocolo ICA, adaptado de (KANTER, 1998)	33
2.12	Arquitetura de softwares SPICE, adaptado de (WANG; LIANG, 2011)	34
2.13	Arquitetura de softwares Microsoft, adaptado de (WANG; LIANG, 2011)	35
2.14	Diferentes index de imagem virtual (TANG, 2011)	39
2.15	Modelos de otimizações COW e QCOW2	40
2.16	Modelo de imagem FDV, adaptado de (TANG, 2011)	42
2.17	Algoritmo Pré-Cópia, adaptado de (BARUCHI, 2010)	43
2.18	Algoritmo Pós-Cópia, adaptado de (BARUCHI, 2010)	44
3.1	Utilização de CPU por aplicação e quantidade de Vds, adaptado de (FALVO, 2014).	48
3.2	Operação de E/S por aplicação e quantidade de Vds, adaptado de (FALVO, 2014).	49

3.3	Utilização de memória por aplicação e quantidade de VDs, adaptado de (FALVO, 2014).	49
3.4	Taxa média de transmissão de dados entre cliente/servidor, adaptado de (FALVO, 2014).	50
3.5	Consumo de energia versus taxa de utilização de memória e processador, com Gentoo (ALBIERO, 2011).	51
4.1	Exemplificação dos custos do modelo estimativo, adaptado de VMware Inc. . .	54
4.2	Exemplificação C_{eet} do VD com diferentes custos de transferências	59
4.3	Exemplificação C_{eet} do VD com diferentes consumos médios do servidor . . .	59
4.4	Exemplificação C_{eet} do VD com diferentes taxas médias de transmissão	60
4.5	Exemplificação C_{eet} do VD com diferentes custos energético médio do cliente .	61
5.1	C_{eet} comparativo Desktop físico vs. Desktop Virtual em diferentes Datacenters	65
5.2	C_{eet} do VD executando em D1 e D2 com diferentes taxas de transmissão média	66
5.3	C_{eet} do VD, por aplicação, executando em diferentes Datacenters com diferentes custos por bit e PUE	68
5.4	C_{eet} do VD executando em Datacenters com diferentes custos energéticos médios por bit para transmissão	70
5.5	C_{eet} do VD executando em diferentes Datacenters com diferentes PUE	71
5.6	Comparativo C_{eet} do VD executando na máquina local c/ hypervisor x DC remoto, executando aplicação IE	76
5.7	Comparativo C_{eet} do VD executando na máquina local c/ hypervisor x DC remoto, executando aplicação MP	77
5.8	C_{eet} do VD com diferentes quantidades de arquivos, Execução remota x execução na máquina local c/ hypervisor LOC1	78
5.9	C_{eet} do VD com diferentes quantidades de arquivos, Execução remota x execução na máquina local c/ hypervisor LOC2	78

LISTA DE TABELAS

2.1	Provedores de serviço DaaS e seus respectivos sites de acesso	27
4.1	Configurações típicas utilizadas para o VD e para D1 na estimativa, valores com base em (BALIGA, 2011), (FALVO, 2014),(ALBIERO, 2011) e (LIU, 2013)	58
5.1	Configurações dos DCs para o cenário 1	63
5.2	Taxa de transmissão média por aplicação (FALVO, 2014)	64
5.3	Configurações dos DCs para o cenário 2	67
5.4	Configurações dos DCs para o cenário 3	72

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	12
1.1 Contexto e Motivação	12
1.2 Objetivos	13
1.3 Metodologia	14
1.4 Contribuições	15
1.5 Estrutura da monografia	15
CAPÍTULO 2 – FUNDAMENTOS TEÓRICOS E TECNOLÓGICOS	16
2.1 Virtualização de hardware	16
2.2 Desktops Virtuais	19
2.3 Computação em Nuvem	21
2.3.1 Provedores de Serviços	23
2.3.2 Infraestrutura Como Serviço (IaaS)	24
2.3.3 Plataforma como Serviço (PaaS)	25
2.3.4 Software Como serviço (SaaS)	25
2.3.5 Desktops como Serviço (DaaS)	26
2.4 Camada de Gerenciamento de Desktops Virtuais	27
2.4.1 Pilha de protocolos e software VMWare	27
2.4.2 Pilha de protocolos e software Citrix	28
2.4.3 Pilha de protocolos e software Red Hat	32

2.4.4	Pilha de protocolos e software Microsoft RDS	34
2.5	Técnicas de Otimização para gerenciamento de recursos virtuais	37
2.5.1	Caching e Compartilhamento de Blocos	37
2.5.2	Copy - on - Write	38
2.5.3	Copy - on - Read e Prefetching	40
2.5.4	Algoritmos de migração	41
2.6	Considerações finais do capítulo	44
 CAPÍTULO 3 – TRABALHOS RELACIONADOS		45
3.1	Trabalho de (BALIGA, 2011)	45
3.2	Trabalho de (FALVO, 2014)	47
3.3	Outros trabalhos	50
3.4	Considerações finais do capítulo	51
 CAPÍTULO 4 – UM MODELO PARA ESTIMATIVA DO CONSUMO DE ENERGIA DE DESKTOPS VIRTUAIS		52
4.1	Detalhamento do Modelo	53
4.2	Aplicação do modelo para estimativas energéticas	57
4.3	Considerações finais do capítulo	60
 CAPÍTULO 5 – APLICANDO O MODELO DE CONSUMO ENERGÉTICO		62
5.1	Cenário 1: Desktop físico vs. Desktop Virtual	63
5.1.1	Aplicando diferentes taxas de transmissão ao cenário 1	65
5.2	Cenário 2: Execução remota com diferentes custos de transmissão	66
5.2.1	Aplicando diferentes custos por bit transmitido e PUE ao cenário 2	69
5.3	Cenário 3: Execução do desktop virtual no datacenter vs. execução no hypervisor local (Horizon)	71
5.3.1	Aplicando diferentes quantidades de dados ao cenário 3	74

5.4	Considerações finais do capítulo	75
CAPÍTULO 6 – CONCLUSÕES		79
6.1	Trabalhos Futuros	81
REFERÊNCIAS		82
GLOSSÁRIO		86
CAPÍTULO 7 – RELATÓRIO DA DEFESA DA DISSERTAÇÃO		87

Capítulo 1

INTRODUÇÃO

1.1 Contexto e Motivação

Na última década, o gerenciamento do consumo energético pelas infraestruturas de Tecnologia de Informação e de Comunicação (TICs) nas empresas surgiu como um dos maiores desafios ambientais. Esse aumento pode ser observado no mundo todo e a tendência é crescer mais a cada ano, fazendo com que os preços energéticos aumentem, o que acarreta no aumento progressivo da regulação do setor por parte do governo. Para frear o aumento do consumo, tem havido pressão para que se melhore a eficiência energética de componentes como processadores, discos, dispositivos de rede, dispositivos móveis, entre outros. Outras medidas incluem busca por eficiência energética por meio de virtualização e consolidação de recursos, bem como o desenvolvimento de arquiteturas, serviços e processos mais eficientes.

Em 2007, um relatório da empresa Gartner's mostrou que a indústria de TICs produzia cerca de 2% das emissões globais de CO₂ (GARTNER, 2007), dos quais, 23% provinham dos centros de dados (*datacenters*). No mesmo ano, um relatório publicado pela Google (INC., 2014) apontou que os seus servidores apresentavam baixa utilização de recursos, entre 10% e 50%, enquanto que a energia consumida com resfriamento do centro de dados ficavam entre 40% e 70% dos seus custos operacionais.

O relatório anual do Greenpeace de 2012 (COOK, 2012) afirma que “na Era da Informação, os centros de dados são as fábricas do século 21”, podendo consumir mais energia elétrica do que 180.000 residências. Percebe-se que os centros de dados estão crescendo rapidamente, tanto em quantidade de recursos quanto no consumo de energia, principalmente por conta da redução dos preços dos recursos computacionais, da informatização dos processos de negócio e da popularização dos serviços pervasivos por meio da Internet e das redes sem fios. Estima-se

que os centros de dados, que executam as inúmeras aplicações disponíveis na Internet e nos dispositivos móveis, consomem cerca de 1.3% de toda a energia produzida no mundo (GAO, 2012) e que para 2020 essa quantidade deve aumentar aproximadamente de 8% (KOOMEY, 2011).

Nesse contexto, um modelo de computação que vem sendo muito utilizado por empresas, órgãos do governo e por instituições de ensino é o *Serviço de Desktops Virtuais* (VDS, acrônimo em inglês para *Virtual Desktop Service*), que se caracteriza por ser um conjunto de servidores conectados à Internet para prover a estação de trabalho do usuário de forma remota em um ambiente virtual de computação. Esse serviço possibilita que usuários possam acessar seus arquivos e executar suas aplicações em seus *desktops* virtuais (VDs, acrônimo em inglês para *Virtual Desktop*), que estarão disponíveis para acesso a qualquer momento por meio da Internet. Para realizar o acesso, são utilizados dispositivos como computadores comuns ou dispositivos móveis, como um notebook, *tablet* ou celular.

Dentre os benefícios da virtualização de *desktops*, podemos elencar a melhoria na utilização dos recursos de computacionais, tais como processador, memória, disco e rede. Além disso, reduz-se os custos para a administração dos recursos, já que esses são alocados de forma centralizada e mais eficiente. Nota-se também uma mobilidade maior para usuário, com a possibilidade de acesso remoto em qualquer lugar, desde que haja um cliente, conectado à rede, para estabelecer conexão com o servidor e assim acessar o VD. Muitas empresas já adotam o uso de VD no ambiente de trabalho dos seus funcionários e colaboradores. Já nas instituições de ensino entretanto, seu uso ainda é incipiente, porém, percebe-se uma demanda crescente (CAVALCANTI; OBELHEIRO; KOSLOVSKI, 2014).

Apesar de oferecer ganhos e facilidades para seus usuários, um VDS pode gerar um considerável consumo de energia, por isso este trabalho tem como objetivo estudar diferentes alternativas e estratégias que possam mitigar o consumo de energia e melhorar a eficiência energética de um VDS.

1.2 Objetivos

O objetivo desta dissertação é criar um modelo capaz de estimar o consumo energético de executar um VD em diferentes cenários típicos de uso. Espera-se que esse modelo possa ser utilizado para estimar os diferentes custos energéticos de instanciar e executar um VD em diferentes cenários, respondendo qual situação pode levar a um menor consumo em cada um dos cenários propostos. Toma-se como premissa que os VDs possuem qualidade de acesso razoável

durante suas execuções e que, independente da localização em que os VDs forem instanciados, os usuários podem executar normalmente suas atividades em seu VD, cumprindo os princípios de qualidade de serviço.

Para avaliar o modelo proposto, são considerados provedores de serviços que possuem servidores distribuídos em diferentes localizações e com diferentes configurações que possam executar os VDs. Tomando como decisão o consumo energético gerado pelo VD ao ser instanciado nesse servidor, pretende-se utilizar o modelo para estimar os custos energéticos nas diferentes localizações possíveis. A partir destas estimativas, é possível decidir por instanciar ou, caso já esteja instanciado, migrar o VD entre os servidores disponíveis, buscando a melhor eficiência energética.

Vale notar que a solução proposta também utiliza na decisão o custo energético necessário para efetuar a migração dos dados de memória, entre o servidor que está executando o VD e outro servidor para onde possa ser mais vantajoso migrá-lo.

1.3 Metodologia

Este trabalho foi iniciado com um estudo sobre virtualização de *desktops* em ambientes de nuvem, buscando identificar oportunidades de pesquisa e embasar uma proposta de solução. Durante a pesquisa bibliográfica foram observadas oportunidades relacionadas a consumo energético em ambientes virtualizados, consumo energético em execução de VDs e custo de transferência de dados pela rede. Estimar com precisão a quantidade de recursos que são utilizados no servidor e qual é o impacto destes recursos no consumo energético durante a execução de um VD, não é uma tarefa simples. Diferentes estudos já foram realizados para medir, calcular e estimar consumo energético de recursos virtualizados em diferentes situações. Alguns destes estudos são apresentados no capítulo 3 deste trabalho.

Tomando como base o levantamento bibliográfico inicial realizado, é proposto um modelo matemático para estimar o consumo energético médio de um VD, em Joules, durante sua execução em um servidor. O modelo utiliza para a estimativa do consumo energético uma somatória envolvendo diferentes custos energéticos. Entre esses custos estão o custo energético da execução do VD no servidor, o custo energético de transferência de seus arquivos entre os servidores, o custo energético de transmissão de quadros de tela entre o VD e o cliente de acesso, e, finalmente, o custo energético do cliente de acesso utilizado pelo usuário. Alguns cenários são propostos para avaliar o modelo proposto, utilizando para isso diferentes valores nas diferentes variáveis de cada custo energético.

1.4 Contribuições

A principal contribuição deste trabalho é um modelo que pode ser utilizado para estimar o consumo energético de executar um VD em um determinado servidor, independente de seu *hardware* e configurações. Esse modelo estimativo é uma alternativa a usuários que procuram eficiência energética porém não possuem informação disponível, e não possuem ou não desejam investir em equipamentos de medição energética capaz de trazer essa informação. Medidores de energia também não resolvem por completo o problema da falta de informação sobre consumo energético em VD, pois não poderiam ser instalados em todos os servidores disponíveis. Além disso, existem outros custos energéticos além apenas do consumo dos servidores. Os provedores de serviço também não disponibilizam muitas informações sobre consumo energético, logo estimativa energética é uma alternativa.

A partir dos custos energéticos estimados pelo modelo, é possível concluir qual entre diferentes servidores, em um determinado cenário, tem a melhor eficiência energética para instanciar e executar o VD requisitado. Com esse modelo será possível responder, por exemplo, quando é melhor ao usuário utilizar um VD remoto ou sua estação de trabalho convencional para executar uma determinada aplicação. É possível também decidir se compensa, no ponto de vista energético, realizar a migração do VD entre diferentes DCs em determinadas situações.

1.5 Estrutura da monografia

Este trabalho está organizado da seguinte forma: o capítulo 2 apresenta a base conceitual deste trabalho sobre virtualização, computação em nuvem e VDs. Ainda nesse capítulo, uma seção é dedicada aos *frameworks* de virtualização, seus protocolos e *software* de infraestrutura. Também são apresentadas técnicas de otimização de recursos a fim de melhorar a qualidade de experiência do usuário em utilização de VDS. Essas técnicas são importantes e são consideradas de forma implícita na avaliação do modelo proposto.

O capítulo 3 apresenta os trabalhos relacionados, que servem como base teórica e fornecem informações relacionadas a este trabalho e ao modelo proposto. O Capítulo 4 traz a proposta, com o modelo de estimativa energética para provisionamento dos VDs. O capítulo 5 apresenta os cenários de experimentos e resultados obtidos utilizando o modelo proposto. O capítulo 6 apresenta a conclusão e as considerações finais. Por fim, são apresentadas as referências bibliográficas utilizadas.

Capítulo 2

FUNDAMENTOS TEÓRICOS E TECNOLÓGICOS

Neste capítulo, são apresentados os principais fundamentos teóricos e tecnológicos sobre consumo de energia em computação e *desktops* virtuais. Esses fundamentos são essenciais para embasar e facilitar o entendimento do modelo estimativo de consumo energético em VD que é proposto no capítulo 4 deste trabalho.

2.1 Virtualização de hardware

A ideia de virtualização surgiu na década de 70, quando os *mainframe* disponibilizaram *softwares* que executavam, além de aplicações básicas, o ambiente operacional necessário para uma aplicação. O conceito de migração também surgiu nesta época, quando a IBM disponibilizou, nos *mainframes 370*, a possibilidade de migrar aplicações entre suas plataformas. Para tal, era necessário que existisse uma máquina virtual (VM) para execução da aplicação na plataforma destino da migração.

A virtualização pode ser considerada como um *framework* ou metodologia para compartilhar os recursos de um computador em múltiplos ambientes de execução. Conceitos como tempo compartilhado, simulação completa ou parcial da máquina, emulação e qualidade de serviços podem ser aplicados a virtualização (SINGH, 2004). Virtualização também pode ser descrita como uma camada de *software* inserida entre o *hardware* e as aplicações que executam tarefas para os usuários. O objetivo é tornar a utilização do *hardware* ao mesmo tempo mais eficiente e conveniente (SILBERCHATZ; GALVIN, 2001). Mais eficiente porque com maior número de máquinas virtuais e sistemas emulados sendo executados, o *hardware* pode atingir a capacidade máxima dos seus recursos e, assim, o retorno do investimento feito.

A virtualização pode ser vantajosa em ambientes que necessitam maior heterogeneidade de

software e sistemas operacionais. Afinal, com a virtualização, pode-se criar diferentes tipos de máquinas virtuais com diferentes sistemas operacionais, o que permite executar diferentes tipos de aplicações sem aumentar a quantidade de computadores. Em sua essência, a virtualização consiste em estender ou substituir um recurso, ou uma interface, existente por um outro, imitando seu comportamento (CARISSIMI, 2008). Existem, ainda, mecanismos de virtualização tratados no próprio *hardware*. A ideia básica, que surgiu na década de 70, é baseada em eficiência, controle de recursos e equivalência.

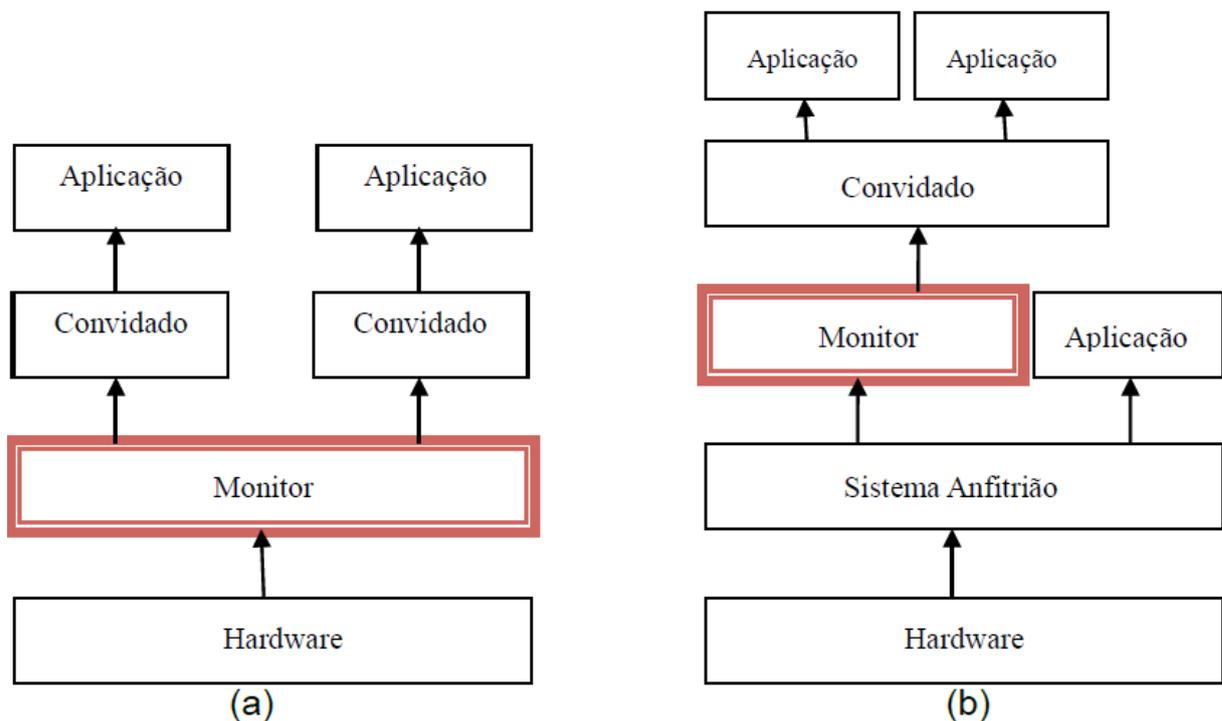


Figura 2.1: Diferentes tipos de máquinas virtuais (SMITH; NAIR, 2005)

A figura 2.1 apresenta dois tipos de máquinas virtuais. No modelo apresentado na figura 2.1 a), o monitor controla o *hardware* e sobre ele são instanciadas as máquinas virtuais. Esse comportamento é similar ao sistema real, onde sistemas operacionais próprios e aplicações são executadas. Esse comportamento pode ser encontrado nos *hypervisors* *VMWare Server* e *Xen*, que são apresentados nas próximas seções deste capítulo. Já no modelo da figura 2.1 b), as máquinas virtuais são instanciadas como processo comuns em um sistema operacional anfitrião. Assim, o sistema operacional anfitrião cria uma camada entre o monitor e o *hardware*. Como exemplos desta virtualização, podem ser citados *Virtualbox* e o *Bochs* (BARUCHI, 2010).

Existem outros tipos de virtualização em que o *hardware* subjacente é replicado para que o sistema operacional da VM possa executar como se estivesse diretamente sobre a máquina física. Neste tipo de virtualização, o sistema operacional utilizado na VM não precisa sofrer alterações e pode executar normalmente. Existe ainda a paravirtualização, que recompila o

sistema hóspede de forma a ser preparado para hospedar as VMs.

A virtualização é composta basicamente por três grupos de recursos físicos virtualizados, os processadores, os discos e a rede. Os processadores, quando trabalham com virtualização, passam a trabalhar de forma a compartilhar o tempo de execução entre as diferentes máquinas virtuais em execução. Esse compartilhamento do processador pode gerar problema de disputa por recurso e atraso no processamento. Já os discos podem ser virtualizados utilizando volumes lógicos, como por exemplo o LVM (*Logical Volume Manager*) que é um controlador de dispositivo fornecido pelo kernel do *Linux* para o mapeamento de blocos físicos em blocos virtuais (KAROVIČ, 2015). Esses blocos virtuais podem ser utilizados como discos das VMs que são instanciadas no recurso físico.

A virtualização de rede, por sua vez, é essencial para o acesso remoto às VMs. Para realizar esta virtualização a rede deve ser adaptada, as *interfaces* passam a trabalhar com multiplexação e demultiplexação para que as interfaces virtuais também possam utilizar a *interface* de rede física de forma compartilhada. As interfaces virtuais de redes criadas possuem, assim como as máquinas físicas, endereços MAC distintos, um endereço de IP distinto que é atribuído para cada interface de VM, suporte a transmissões *unicast*, *multicast*, *broadcast* e VLANs. Assim, cada VM é tratada como um sistema real, pelas conexões de rede, o que permite conexão a *switches* e roteadores como se fossem máquinas físicas distintas.

Equipamentos como *switches* e roteadores não são naturalmente integrados a máquinas virtuais, porém existe a possibilidade de emulação. Para esse tipo de emulação, existem a tecnologia TUN/TAP, que são adaptadores de redes virtuais disponíveis para sistemas operacionais atuais, como o MacOS, o Windows, Linux e Unixes, em geral (CARISSIMI, 2008). Esses dispositivos aparecem normalmente em redes privadas virtuais (VPN) e emulam o comportamento das camadas de rede e de enlace. Neste contexto, o TAP permite a criação de *bridges* enquanto o TUN é o responsável por realizar o roteamento. Utilizando esses *drivers*, aplicações em execução podem transmitir dados pela rede como se fossem dispositivos externos. Essa técnica é utilizada como base para comunicação em ambientes de rede virtualizados.

Existem, no entanto, alguns problemas que podem ser apresentados sobre virtualização. É praticamente impossível implantar uma VM que emule completamente o *hardware*. Réplicas de rede com um *daemon* em execução ou replicação de entrada e saída são virtualizações difíceis de serem configuradas. Outro problema pode ser a disputa por controle de recurso, como exemplo pode ser citada a implementação de paginação para memória virtual feita pelo sistema hospedeiro. Assim, as VMs, que devem respeitar esse controle, acabam concorrendo com o sistema real pelo recurso. Com isso, o desempenho é afetado conforme cresce o número de máquinas

virtuais instanciadas no mesmo *hardware*, afinal são sistemas operacionais executando sobre outro sistema operacional. Para tentar minimizar o *overhead* sobre o sistema primário, existem os *hypervisores*, que gerenciam os recursos e os sistemas virtuais em execução na máquina física.

2.2 Desktops Virtuais

O conceito de VDI (acrônimo em inglês para *virtual desktop infrastructure*) representa a virtualização da estação de trabalho do usuário, separando o ambiente *desktop* do servidor físico em que este é executado (KIBE; KOYAMA; UEHARA, 2012). VDI tem como objetivos a disponibilidade e confiabilidade da estação de trabalho do usuário, pois seus arquivos ficam armazenados em servidores que normalmente possuem backups redundantes dos arquivos, gerando maior segurança e proteção aos mesmos. Além disso, VD, podem ser replicados em diferentes servidores para que estejam disponíveis sempre que necessário, assim podem ser acessados de qualquer lugar, pela rede, por um simples navegador ou via um programa cliente de acesso (CARISSIMI, 2008), como pode ser visto na figura 2.2.

Os clientes utilizados para acesso ao VD são dispositivos simples como computadores, *tablets*, *smartphones*, entre outros, chamados *thin client*, aparelhos que transmitem quadros de tela e comandos de teclado e mouse entre o usuário e o servidor. Assim, o investimento financeiro feito pelo usuário para utilizar VDs é baixo, envolve apenas esse cliente de acesso e o custo financeiro sob os recursos que seu VD consumir do servidor.



Figura 2.2: VDs, adaptado de (BOSING; KAUFMANN, 2012) e commons.wikimedia.org

Os VDs podem também reduzir custos gerenciais aos provedores de serviço por serem mais fáceis e baratos de gerenciar, afinal é menos trabalhoso gerenciar vários VDs, que podem ser administrados de forma remota e centralizada, do que suportar várias máquinas físicas espalha-

das por diversas estações de trabalho. Com gerenciamento otimizado e utilização sob demanda, os VDs são uma boa alternativa contra o crescimento do consumo energético em computação, pois são utilizados sob demanda e não necessitam ficar ativos em tempo integral (RUMBLE, 2011). Existem ainda algumas técnicas como a consolidação de recursos, que podem ajudar a diminuir ainda mais o consumo energético dos VDs. Essa técnica visa alocar a máxima quantidade de recursos virtuais possíveis em um mesmo recurso físico, para que atinja o máximo de aproveitamento do *hardware* do servidor e a utilização máxima dos recursos, evitando ociosidade e economizando energia (SILBERCHATZ; GALVIN, 2001). Nesta técnica, os recursos virtuais (VDs/VMs) são migrados entre os servidores para que sejam consolidados na menor quantidade de servidores possível. Após a migração, os servidores que ficaram ociosos podem ser desativados.

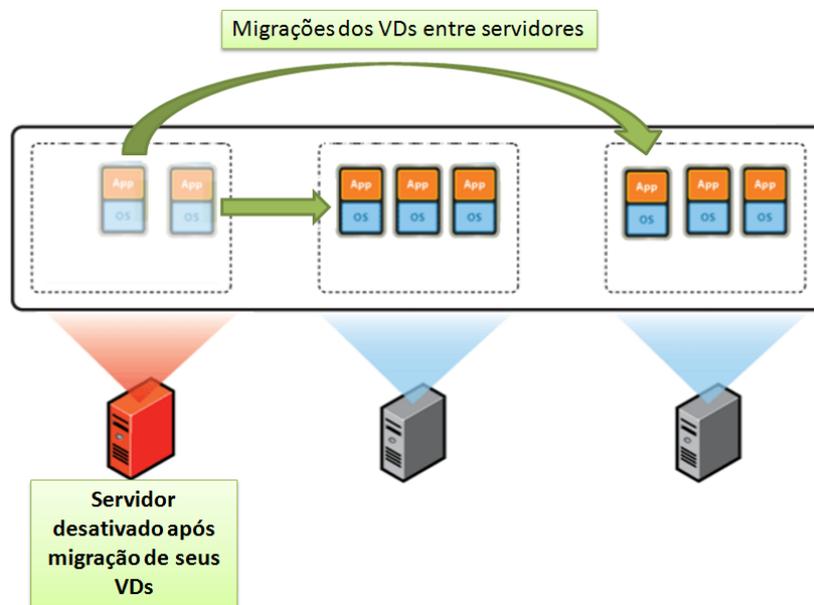


Figura 2.3: Consolidação e migração de VDs, adaptado de *mirantis.com*

Uma outra técnica utilizada pelos provedores de serviço para balancear a carga entre servidores, para consolidar recursos virtuais e para diminuir o consumo energético é a migração dos VDs, conforme ilustrado na imagem 2.3. Os VDs são compostos basicamente por seu sistema operacional, arquivos e as aplicações utilizadas pelo usuário, assim em uma migração esses dados todos precisam ser transferidos entre os servidores. Os provedores, normalmente, possuem diferentes servidores aptos a executar o VD, de modo que VDs em execução podem ser remanejados entre esses servidores. Utilizando migração, os provedores de serviço evitam sobrecarga de utilização dos servidores físicos, melhoram a execução dos VDs e consolidam recursos para economizar energia (RAZAVI; RAZOREA; KIELMANN, 2014).

A migração de um VD pode, no entanto, trazer problemas para o usuário e para o servidor se

isso não for planejado, pois durante a migração a execução do VD precisa ser interrompida por um determinado tempo, para completar a transferência da memória. Mesmo que esse tempo seja um tempo pequeno isso pode trazer prejuízos ao usuário, como a interrupção de uma execução de uma tarefa por exemplo. Existe também a possibilidade de que essa migração custe, devido à quantidade de arquivos a transferir, ainda mais energia do que manter o VD no servidor atual. Por isso, os provedores de serviço muitas vezes utilizam de servidores de cache, compactação e outras técnicas de otimização para realizar a migração dos dados dos VDs.

A virtualização de *hardware* junto com a computação em nuvem gera otimização ao ambiente de VDI, pois possibilita a criação de vários recursos e isso possibilita a criação de vários VDs em um mesmo recurso físico. Para instanciar esses VDs são necessários *hipervisores* que gerenciam o *hardware* para criar os recursos virtuais. Existem diferentes controladores disponíveis atualmente, como o *vSphere* da *VMware*, *XenServer* da *Citrix* e o *RD Virtualization Host* da Microsoft, que são detalhados nas próximas seções deste capítulo.

2.3 Computação em Nuvem

A Computação em nuvem tem como fundamento básico a disponibilidade de recursos e serviços acessíveis pela rede de qualquer lugar, a qualquer momento e independente de plataforma. Diferentes aplicações, serviços, máquinas e VDs são disponibilizados com acesso via Internet. Um dos objetivos do conceito de computação em nuvem é explorar o máximo dos recursos computacionais, provendo recursos sob demanda, evitando capacidade ociosa do *hardware* e consequentemente evitando desperdícios energéticos. Essa abordagem pode ajudar a diminuir o prejuízo ambiental causado pelo consumo energético em computação, como nos *datacenters* (DCs) e outras infraestruturas de recursos computacionais. A emissão de gás carbônico dos DCs já é responsável por cerca de 2% do CO₂ emitido no mundo (BELOGLAZOV; BUYYA, 2010). Evitar, então, a utilização ineficiente dos recursos computacionais é um dos objetivos dos provedores de serviços o que pode refletir também na redução de gastos desnecessários com refrigeração e energia elétrica.

A computação em nuvem possibilita o fornecimento de infraestrutura de Tecnologia da Informação (TI) como um serviço adquirido e pago sob demanda pelos clientes (COSTA, 2010). Assim, os recursos são alocados dinamicamente pelos provedores de serviço conforme a demanda dos usuários, que também pagam somente pelo que utilizam. Esses recursos distribuídos, fornecidos pela computação em nuvem, podem ser tão eficientes quanto super computadores, isto é, desde que as tarefas sejam distribuídas entre os diferentes recursos de forma balanceada

e ordenada. Esse tipo de processamento computacional também deverá ser mais barato para o usuário do que o investimento financeiro em uma única super máquina. Outro ponto importante da computação em nuvem é quanto à segurança, uma vez que muitos dados ficam armazenados em computadores remotos, exigindo grande proteção, já que uma falha causaria a exposição de dados de clientes e um prejuízo aos provedores.

Existem quatro diferentes implantações de infraestruturas de nuvem, a privada, a pública, o modelo de comunidade e híbrida. O modelo privado consiste em um conjunto de recursos que compõem a nuvem restrita a um grupo de usuários. O acesso é controlado por senha ou chaves de restrição de acesso. Este modelo de computação em nuvem é desenvolvido para utilização restrita, seja por empresas, instituições ou grupos particulares de usuários (CHAVES; URIARTE; WESTPHALL, 2010). Este modelo pode ser gerado a partir de recursos próprios da empresa ou instituição, no entanto, além de ser mais caro, exige configurações e suporte próprio, que podem gerar mais problemas que soluções.

Já o modelo de computação em nuvem público é composto por serviços que podem ser acessados por diferentes tipos de usuários de qualquer lugar a qualquer momento pela Internet. O acesso é restrito ao contrato estabelecido pelos serviços daquela nuvem. Como exemplo do modelo público de computação em nuvem cita-se as ferramentas da empresa Google, como o google Docs e g-mail. Estes serviços são gratuitos, porém ainda necessitam de identificação de usuário e senha para o acesso (SO, 2011).

Uma nuvem pública também pode ter parte dos recursos particulares e parte dos recursos contratados, o que caracteriza o modelo híbrido, como pode ser visto na figura 2.4. Neste modelo, um usuário qualquer devidamente cadastrado pode ter acesso a alguns recursos desta nuvem, porém parte dos serviços e recursos podem ser restritos a uma organização. O modelo híbrido também pode ser utilizado por empresas ou instituições como forma de *backup* ou servidores extras em caso de sobrecarga em seus servidores particulares. Assim, uma quantidade de recursos é contratada de provedores para somar-se aos recursos particulares existentes.

Por fim, o modelo comunitário refere-se a uma infraestrutura controlada e compartilhada, em termos de dados e aplicações, por um conjunto de Organizações com interesses comuns. Esses interesses podem ser, por exemplo, em requisitos específicos de segurança.

Já os serviços de computação em nuvem são providos de diferentes maneiras, como pode ser visto na figura 2.5, onde cada um possui uma finalidade e um grupo diferente de usuários. A pirâmide invertida é resultado da utilização de cada tipo de serviço, em que o SaaS tem, normalmente, mais usuários do que o IaaS.



Figura 2.4: Modelos de computação em Nuvem

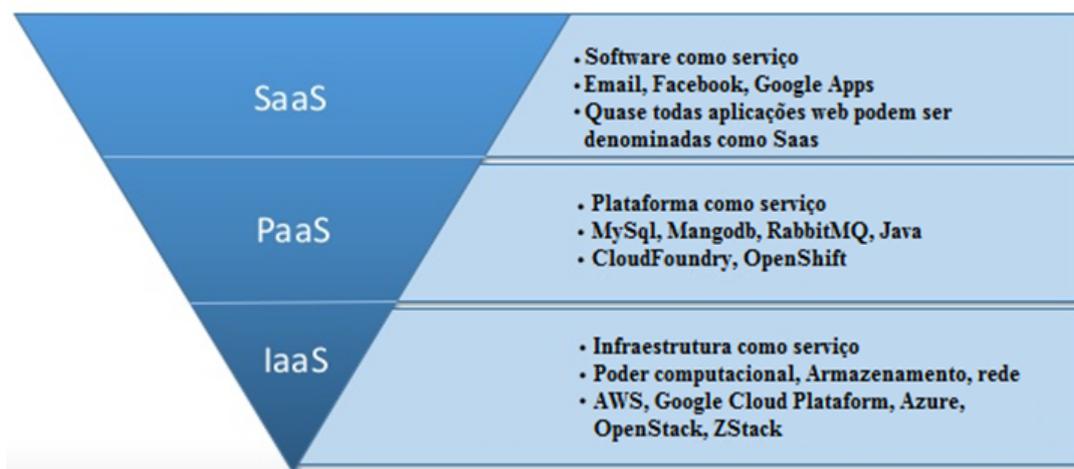


Figura 2.5: Tipos de serviço da computação em nuvem

2.3.1 Provedores de Serviços

Para gerenciar e prover serviços, como os apresentados na figura 2.5, existem os provedores de serviço. Esses provedores seguem os princípios de qualidade de serviço, visando a disponibilidade e confiabilidade do serviço. A Disponibilidade do serviço tem como objetivo garantir ao usuário que seu serviço estará, tanto quanto possível, disponível assim que solicitado. Já a confiabilidade do serviço, tem como objetivo garantir ao usuário que seus serviços não vão falhar durante utilização.

Amazon Web Services (AWS) é um importante exemplo desses provedores de IAAS, pois influenciou alguns softwares controladores de nuvem, como Eucalyptus e OpenStack. AWS é proprietária de dois serviços que são base para alguns softwares controladores de computação

em nuvem, o EC2 (*elastic compute cloud*, em uma tradução livre para português computação em nuvem elástica) e o S3 (*Simple Storage Service* - Serviço de armazenamento simples). EC2 é um serviço Web que fornece capacidade de computação redimensionável em nuvem. É projetado para tornar a escalabilidade computacional web mais simples para desenvolvedores. EC2 disponibiliza interfaces simples de serviço da Web, permitindo ao usuário configurar e gerenciar estruturas com mínimo esforço. O EC2 também gerencia recursos computacionais reduzindo o tempo para obter e inicializar novas instâncias de serviços. Já S3 disponibiliza o armazenamento via Internet, gerenciando e facilitando a utilização deste tipo de serviço para os usuários desenvolvedores. Assim como o EC2, o S3 fornece uma interface de serviço web simples, que pode ser utilizada para gerenciar arquivos referentes aos recursos virtuais do usuário (BRANTNER, 2008).

Existem outros provedores de serviços, como Rackspace, que é um provedor de infraestrutura com investimentos em centros de dados, servidores de hardware, equipe de TI e gestão de conectividade à Internet (HILLEY, 2009). Rackspace oferece serviços semelhantes com os providos pela Amazon, tais como hospedagem, máquinas virtuais e hospedagem compartilhada.

Esses provedores fornecem serviços de computação em nuvem como os apresentados na figura 2.5. Existe ainda um quarto tipo de serviço, chamado DaaS (Acrônimo em inglês para desktop como serviço) que é apresentado, junto aos outros serviços, nas próximas seções deste capítulo.

2.3.2 Infraestrutura Como Serviço (IaaS)

Infraestrutura como Serviço (IaaS, acrônimo inglês para *Infrastructure as a Service*) refere-se a uma combinação de hospedagem, hardware, armazenamento, provisionamento e serviços básicos necessários para executar um serviço de computação em nuvem. Desta forma, garante-se toda a estrutura básica para que o usuário possa executar suas aplicações de forma remota (PASTRAMA, 2011). Usuários podem adquirir e configurar recursos, como largura de banda, máquinas virtuais e armazenamento que são disponibilizados neste tipo de serviço (COSTA, 2011). Este modelo permite aos usuários maior flexibilidade e gestão sobre os recursos utilizados, porém exige um esforço para configuração, normalmente inexistente quando comparado com os serviços SaaS ou PaaS.

Como exemplo deste tipo de serviço cita-se Amazon Web Services, com serviços de armazenamento Simple Storage Solution (S3) e serviços de computação elástica em nuvem Amazon Elastic Compute Cloud (EC2) (BALDUZZI, 2012).

2.3.3 Plataforma como Serviço (PaaS)

O modelo Plataforma como serviço (PaaS, acrônimo inglês para *Platform as a Service*) é um conjunto de serviços disponibilizado pela nuvem com objetivo de fornecer uma plataforma de computação para desenvolvimento. Esse tipo de serviço são normalmente utilizados por usuários desenvolvedores de aplicações. O desenvolvedor possui ferramentas e bibliotecas disponibilizadas pelo provedor de serviços com a finalidade de desenvolver e instalar as aplicações que depois são disponibilizadas em modelo de SaaS. O provedor fornece conexões de rede, servidores e armazenamento (COSTA, 2011).

O serviço PaaS facilita a implantação de aplicações sem o custo e a complexidade de contratar e gerenciar o *hardware/software* necessários para o desenvolvimento, proporcionando assim facilidade e rapidez necessárias para todo o processo de desenvolvimento e entrega de softwares. Como exemplo deste tipo de serviço é possível citar Microsoft Azure ou Google App Engine (MARCHÃO, 2012).

2.3.4 Software Como serviço (SaaS)

Nesse modelo, aplicações completas são disponibilizadas para acesso via Internet de forma gratuita ou cobrada sob demanda de utilização. SaaS (acrônimo inglês para *Software as a Service*) também pode ser chamado de *software on-demand* (Inglês para *software* sob demanda), é um modelo em que os dados de *software* e suas dependências são hospedado na nuvem e que podem ser acessados por usuários por meio de cliente de acesso ou simples navegadores Web (COSTA, 2011). Normalmente, durante a utilização, não é necessária nenhuma instalação ou configuração por parte do usuário, trazendo assim facilidade e comodidade ao mesmo. Outra vantagem é quanto à compatibilidade das aplicações, que também é mais ampla do que a instalação tradicional das aplicações (GONG, 2010).

Atualizações nas aplicações deste tipo de serviço também são mais simples, uma vez que o processo é executado na infraestrutura controlada pelo provedor de serviços. Assim, atualizações e novas funcionalidades podem ser instaladas de forma centralizada e de modo a que o usuário não seja afetado. Existem muitos exemplos de aplicações nesta categoria da computação em nuvem. Podem ser citados o Google apps, emails, serviços de comunicação entre outros softwares que possam ser acessados de forma remota (MARCHÃO, 2012).

2.3.5 Desktops como Serviço (DaaS)

DaaS (acrônimo inglês para *Desktop as a Service*) deriva de certa forma do conceito SaaS, porém o usuário contrata uma estação de trabalho completa e não apenas uma aplicação específica. Desenvolvido com virtualização de *Desktops*, DaaS apresenta um modelo eficiente, escalável e confortável de serviço sob demanda. (SHU, 2012). No modelo DaaS, usuários utilizam e pagam normalmente pelo que utilizam, apesar de que alguns provedores já apresentam modelo de cobrança mensal. Como citado no começo deste capítulo, com o conceito de VD, o usuário tem economia financeira em *hardware* pois as estações de trabalho físicas tornam-se menores e o usuário só precisa de um simples cliente de acesso. O *hardware* e a configuração dos serviços são responsabilidades do provedor de serviços. Os principais desafios deste conceito DaaS estão na parte de transmissão de quadros de tela e comandos, visando a diminuição da latência e da transmissão de dados (KIBE; KOYAMA; UEHARA, 2012).

Amazon WorkSpaces é um dos serviços providos pela Amazon que fornece o serviço DaaS. A finalidade é prover um serviço que forneça estações de trabalho prontas para utilização aos usuários. O serviço provê gerenciamento da infraestrutura de computação de *desktops*, resultando em um baixo investimento inicial ao usuário. O armazenamento é feito de forma persistente na Nuvem utilizando os serviços AWS. Amazon Workspace utiliza o protocolo Remote Display Protocol (RDP) para comunicação entre o cliente utilizado pelo usuário e seu VD, além de serviços de criptografia como o AWS Key Management Service (KMS), com a capacidade de criptografar volumes de armazenamento do WorkSpaces. Esses serviços podem gerar uma economia inicial em *hardware* ao usuário, porém a longo prazo podem não ser tão vantajosos. De acordo com o site oficial da Amazon WorkSpaces, na cotação atual o preço da contratação deste serviço varia entre 25 dólares por mês, para o modelo mais básico, e 75 dólares por mês para o modelo mais avançado em uma determinada localização. Isso somente incluindo aplicações básicas, onde o contrato de aplicações avançadas podem adicionar um extra de 15 dólares mensais a este contrato.

Existem outros provedores VDI como o tuCloud, que utiliza de configurações baseadas no hypervisor Citrix, assim utilizam do protocolo HDX para comunicação entre o cliente do usuário e seus VDs. tuCloud possui clientes como NNSA (Acronimo para National Nuclear Security Administration) e a universidade UCLA (Acrônimo para University of California Los Angeles). tuCloud Apresenta diferentes modelos de venda do apresentado pela Amazon; vendem pacotes completo de nuvem DaaS, ou seja, pacotes com uma quantidade maior de VDs prontos que podem ser utilizados por organizações como os clientes citados. A venda tem preço por usuário, que pode chegar na cotação atual a 9.99 dólares por usuário.

Outros provedores com características semelhantes ao citado nesta seção podem ser encontrados no mercado, com diferenças que são normalmente encontradas no custo financeiro e planos de venda. Alguns provedores também são diferenciados por utilizarem diferentes hipervisores e protocolos de comunicação. Esses provedores são apresentados na tabela 2.1.

Provedor DaaS	Site
AWS	http://aws.amazon.com/pt/workspaces/
CludMyOffice	http://www.cloudmyoffice.com/
CyberlinkASP	http://www.cyberlinkasp.com/
RamNode	http://www.ramnode.com/
TuCloud	http://www.tucloud.com/DaaS.html

Tabela 2.1: Provedores de serviço DaaS e seus respectivos sites de acesso

2.4 Camada de Gerenciamento de Desktops Virtuais

Esta seção descreve alguns dos controladores de máquinas (VMs) e *desktops* virtuais (VDs) disponíveis no mercado. São apresentadas algumas de suas características, seus protocolos de comunicação, seus fluxos de ações e seus componentes. Esses controladores, assim como citado na seção anterior, estão presentes nos provedores de serviços e são utilizados como base para controlar e gerenciar o provisionamento dos recursos virtuais. São responsáveis pelo gerenciamento do *hardware* no servidor, pela alocação dos VDs/VMs, além de realizar a comunicação entre o cliente de acesso utilizado pelos usuários e seus respectivos VDs.

2.4.1 Pilha de protocolos e software VMWare

VMware server é a solução de infraestrutura da VMware para virtualização de *desktops*, constituída por gestão e automação voltados a gerenciar os sistemas de virtualização. Essa infraestrutura possui ferramentas que operam na alocação e no monitoramento dos recursos nas máquinas físicas e, por fim, na virtualização das plataformas, que é de fato onde ocorre a virtualização dos *desktops*. Existem também recursos de virtualização de rede, memória, armazenamento, gerenciamento dos arquivos, processadores e sistema de arquivos virtual (*VMFS virtual machine file system*) (SURKSUM, 2014).

VMware server utiliza o protocolo PC-over-ip e tem como uma de suas principais ferramentas o *VMware Workstation*, que, como o próprio nome diz, é a estação de trabalho que permite a criação de máquinas virtuais sobre o *hypervisor* com sistemas operacionais e softwares distintos. A arquitetura é constituída por vários componentes que podem ser resumidos a cinco peças

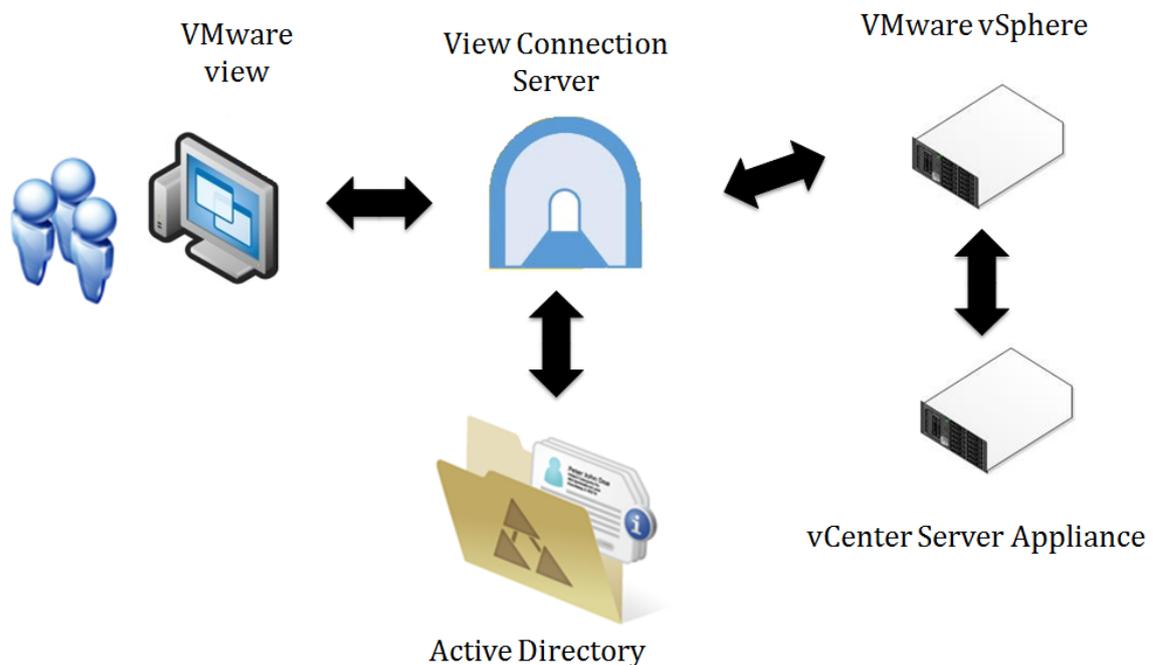


Figura 2.6: Arquitetura de softwares VMware, adaptado de (WANG; LIANG, 2011)

chaves, como apresentado na figura 2.6 (WANG; LIANG, 2011). O primeiro, *VMware view*, é o *software* de acesso utilizado pelos clientes para acessar suas VMs, utilizando computadores ou clientes portáteis como *tablets* e celulares. *View connection server* atua como *broker* para conexões com os clientes, realiza a autenticação com o diretório e busca a VM requisitada pelo usuário.

vSphere, por sua vez, é a plataforma de virtualização que disponibiliza recursos para as VMs. *vCenter server* controla junto ao *vSphere* a configuração, a implantação e o gerenciamento das máquinas e *desktops* virtuais. Por fim, *View agent* é o controlador instalado em cada VM. Este agente comunica-se com o cliente instalado na máquina física do usuário e provê o monitoramento de dispositivos, permitindo impressão de arquivos, conexões USB e outras funcionalidades por meio da rede.

2.4.2 Pilha de protocolos e software Citrix

Xen Desktop é a solução de *software* para virtualização oferecida pela empresa Citrix (WANG; LIANG, 2011). A comunicação entre as VMs e o usuário é feita utilizando o protocolo ICA (*Independent Computing Architecture*) (CARISSIMI, 2008). O *hypervisor* Xen é licenciado nos termos da GLP (*General Public Licence*). Ele é voltado a arquiteturas x86 e foi criado na Universidade de Cambridge, adquirido pela Citrix System em 2007. Xen difere do VMware pois pode, a partir da versão 3, implementar a paravirtualização e também a virtualização con-

vencional. Na paravirtualização, o *kernel* é recompilado, gerando um sistema operacional capaz de criar vários *domains-U* onde o *domain-0* é o *domain host* e os demais são capazes de receber as máquinas virtuais (CARISSIMI, 2008).

Os *domains* podem ser privilegiados ou não e é o *hypervisor* que controla os recursos que são disponibilizados pelo sistema físico para as máquinas virtuais. O sistema operacional base é executado no *domain-0*, como pode ser visto na figura 2.7, e tem todos os privilégios sobre o sistema. As máquinas virtuais, no entanto, são instanciadas nos *domains-U* que são os domínios sem privilégio e que só podem ser executados após o sistema 0 ser iniciado.

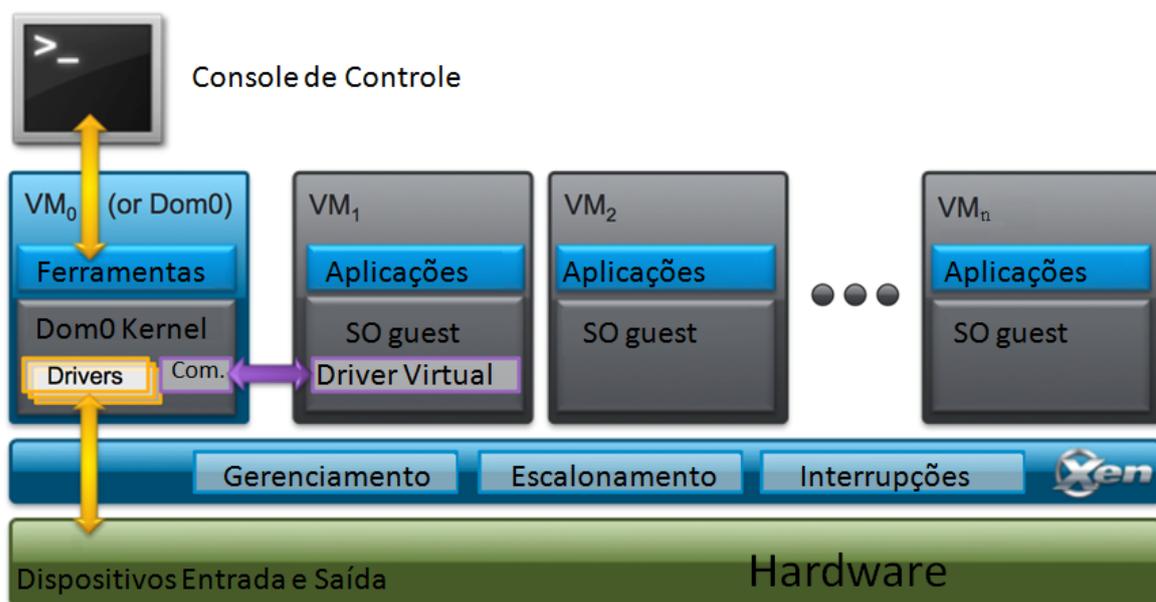


Figura 2.7: Domínios e Hypervisors, adaptado de (CARISSIMI, 2008)

Xen, a partir da versão 3 não é mais dependente de sistemas *Unix*. No entanto, para executar sistemas operacionais *Windows*, ainda depende que seja atendida a restrição do modelo do processador. Para ser flexível e atender tanto *paravirtualização*, que pode trazer um melhor desempenho em sistemas *unix*, e a virtualização completa, os domínios são diferenciados entre para-virtualizados (domínios U-PV) e virtualizados (domínios U-HVM).

A arquitetura Citrix-Xen é formada por vários componentes que, juntos, constituem a arquitetura principal com sete integrantes, como pode ser visto na figura 2.8. O primeiro citado é o *Desktop receiver*, que é o *software* utilizado pelo cliente de acesso para realizar a conexão remota com a VM utilizando o protocolo ICA. Esse software interage diretamente com o *Access gateway*, que utiliza padrões SSL para autenticar o usuário externo antes que possa acessar sua VM.

Já o componente *Desktop delivery controller* é responsável por gerenciar a conexão entre a

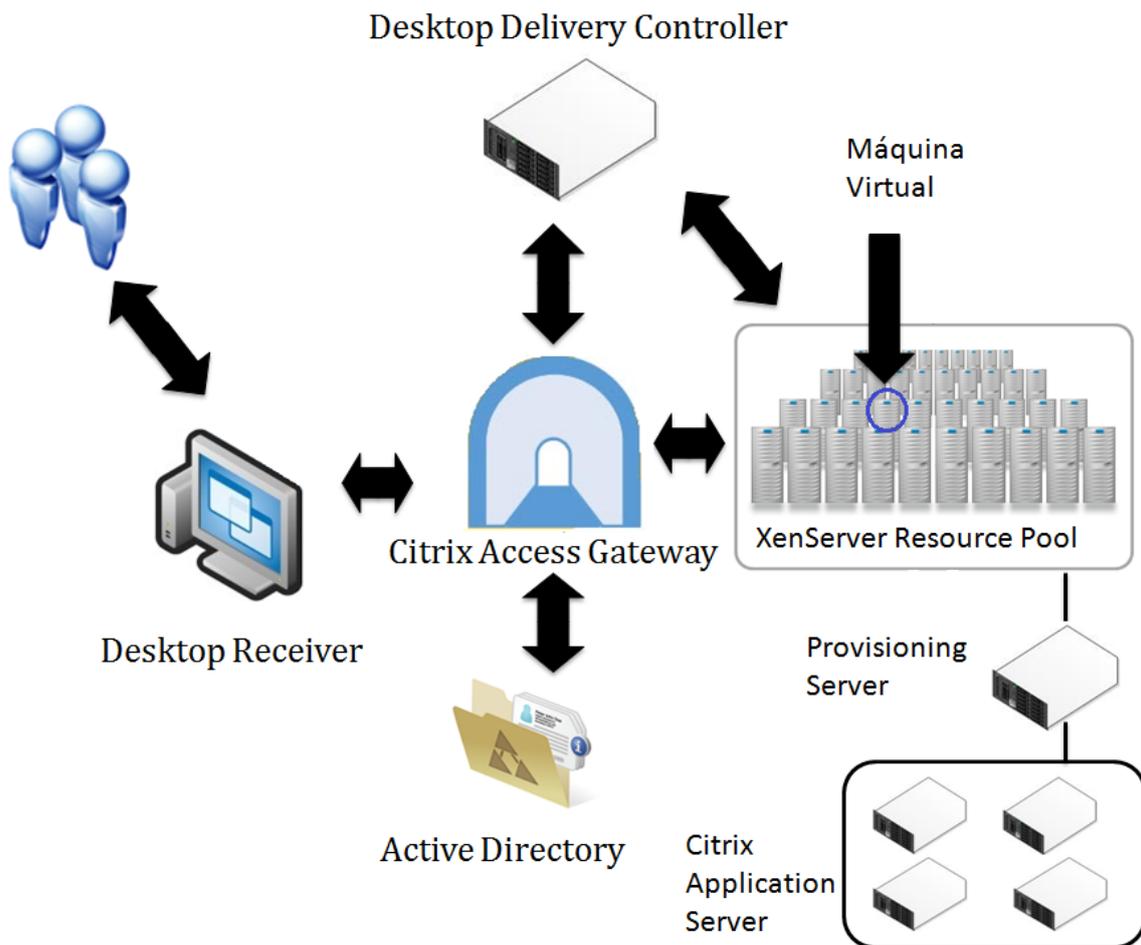


Figura 2.8: Arquitetura de softwares Citrix, adaptado de (WANG; LIANG, 2011)

VM e o cliente, assim que a VM é iniciada. Assim, quando o cliente é iniciado, é realizada uma verificação que encaminha a conexão específica até a VM requisitada pelo usuário. As VMs são então alocadas em recursos disponibilizados pelo *XenServer resource pool* que é a plataforma de virtualização Citrix, responsável por prover e gerenciar os recursos para virtualização. A instanciação da VM fica a cargo do *Provisioning server*; assim, logo que a VM é iniciada, este componente é responsável por prover as imagens dos sistemas operacionais pré-configuradas para determinados perfis de usuários.

Por conseguinte, após o *provisioning server* prover as imagens do sistema operacional, é a vez do *Application delivery controller* prover as aplicações do usuário requisitante. Existe ainda o *Virtual desktop agent*, que é executado dentro da VM para realizar a conexão com o cliente utilizando o protocolo ICA, e o *Active directory*, que realiza a autenticação do usuário. Assim, para que o usuário acesse sua VM, ele deve utilizar suas credenciais para acessar o *Access gateway*. O *gateway* então encaminha a conexão ao *Desktop delivery controller* que verifica as credenciais do usuário e recupera as VMs que o mesmo possui. Essa lista recuperada

é enviada ao cliente/navegador do usuário que escolhe qual VM deseja acessar. O servidor cria um arquivo ICA para a VM escolhida e envia ao *gateway* para que seja entregue ao cliente. Com o arquivo de acesso, o cliente envia a solicitação de conexão ao *gateway* novamente, que a encaminha até a VM.

O protocolo utilizado para estas transições é uma extensão HDX (*high definition experience*) do protocolo ICA. Essa extensão possui algumas vantagens, como transmissão gráfica, o *MediaStream HDX*, e vídeos em *flash* (KOURIL; LAMBERTOVA, 2010). O protocolo também ajusta a taxa de bits durante a transmissão conforme a conectividade, o que é excelente para redes heterogêneas (SCHLOSSER, 2010). O protocolo ICA (Independent Computing Architecture) foi lançado em 1995, com a ideia de executar todo o processamento do lado servidor. Devido a seus conceitos semelhantes ao protocolo xWindows utilizados pelo Unix, foi rapidamente integrado a alguns sistemas operacionais como Windows NT, 95, 3.x (KANTER, 1998).

O pacote do protocolo ICA é constituído por um byte de comando seguido por dados. Existe ainda a possibilidade de ser utilizado como negociação para gerenciar a conexão. O cabeçalho é detalhado nas figuras 2.9 e 2.10.

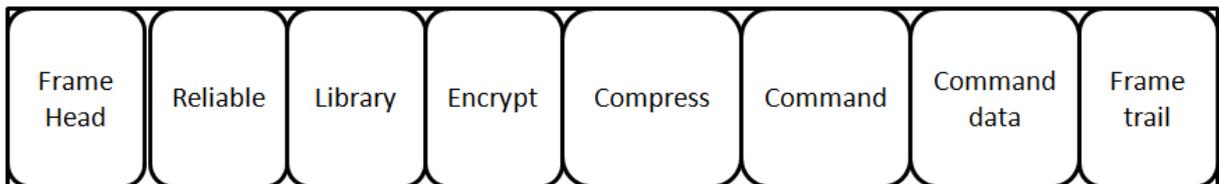


Figura 2.9: Cabeçalho protocolo ICA, adaptado de (KANTER, 1998)

O protocolo ICA pode ter diferentes comandos como controle, tela cheia, teclado e movimento para mouse. O comando de controle é utilizado para gerenciar, negociar e controlar conexão entre cliente e servidor, além de conexão de hardwares como teclado e mouse. Comandos de tela cheia são utilizados para controlar alterações na tela transmitida, evitando que toda a tela seja transmitida a todo instante entre cliente e o servidor, atualizando somente os pixels alterados no último comando. ICA controla também a escrita, o movimento de cursor e a rolagem de tela. Já o Controle de teclado transmite o grupo de coordenadas que representam as alterações a serem ajustadas na tela pelo usuário na estação cliente. Por fim, o Controle de mouse transmite as coordenadas do ponteiro ao servidor e retorna a posição do mouse visualizada pelo usuário na estação cliente.

Para garantia de pacotes, utilizando de um protocolo ICA, existe o procedimento a ser realizado como podemos ver na figura 2.11. O Procedimento começa com aplicação gráfica sendo exportada e encapsulada para um datagrama do protocolo, essa etapa é o protocolo lógico conhecido como *ThinWire*. Após o término desta etapa, os dados são entregues aos *drivers* de

QUADRO	NOME	STATUS	FUNÇÃO
Frame Head	cabeçalho	opcional	Preâmbulo para transporte de frames
Reliable	confirmação	opcional	Confirmação de entrega do cabeçalho. Funciona como detecção de erro.
Encrypt	criptografia	opcional	Gerenciamento de dados criptografados
Compress	compressão	opcional	Gerenciamento de dados comprimidos
Command	comando	exigido	Indica início do pacote
Command Data	dados	opcional	Comando associado aos dados. Poderá ser 0 (zero) e seu tamanho dependerá do tamanho dos dados transmitidos. Pode conter ainda pacotes do protocolo de canal virtual (<i>Virtual Channels protocol</i>) ¹⁶
Frame Controll	sincronia	opcional	Teste de sincronismo para conexões assíncronas.

Figura 2.10: Funções protocolo ICA, adaptado de (KANTER, 1998)

transporte ICA, que então envia de forma ordenada e sem erros até o receptor.

Outras vantagens do protocolo ICA é que possui conectividade *plug-and-play* para vários dispositivos multimídia, compressão de dados multimídias para melhorar a transmissão e priorização de serviços.

2.4.3 Pilha de protocolos e software Red Hat

O projeto SPICE da Redhat tem como objetivo criar uma solução para virtualização de *desktops* utilizando o protocolo SPICE (*Simple protocol for independent computing environment*), que possui funcionalidades semelhantes ao RDP da Microsoft para conexão remota. Também utiliza canais paralelos para estabelecer a comunicação entre o cliente e sua VM com chave RSA de 1024 bits para segurança da conexão. O grande desafio atual da comunidade que trabalha no desenvolvimento do SPICE está em *Network tunneling* e *USB sharing* por diferentes servidores.

A arquitetura de componentes Red Hat para virtualização possui basicamente quatro componentes principais, que recebem as requisições e gerenciam toda criação e comunicação entre o cliente e sua VM, como pode ser observado na figura 2.12. Para iniciar a comunicação, utilizam

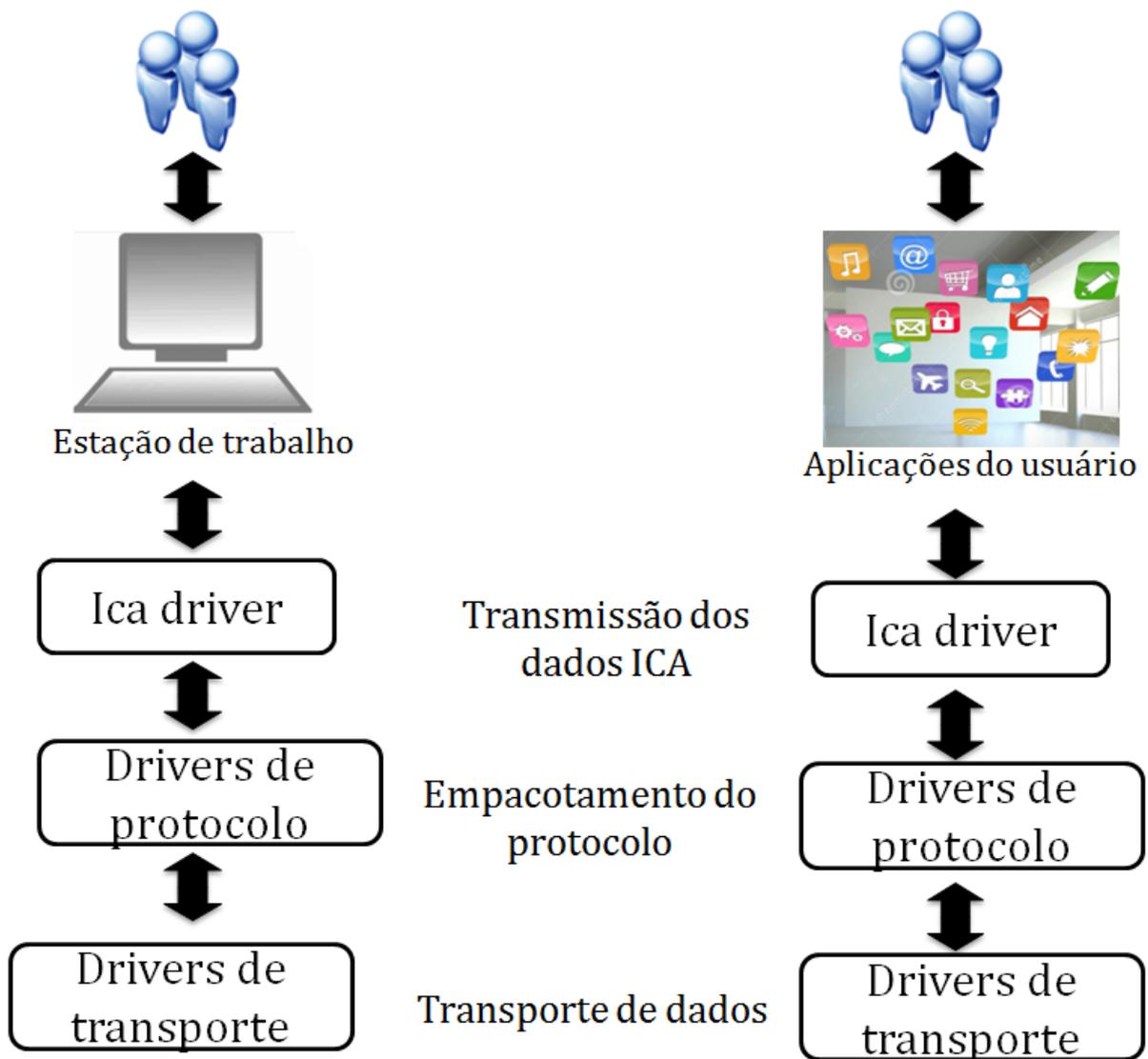


Figura 2.11: Processo de entrega protocolo ICA, adaptado de (KANTER, 1998)

o *SPICE client*, que é o *software* instalado no lado do cliente de acesso para conectar a VM, utilizando o protocolo SPICE. Este protocolo é um protocolo *open source* para comunicação entre o cliente de acesso e a VM (WANG; LIANG, 2011).

Um dos principais componentes desta arquitetura é o *Red Hat Enterprise Virtualization Hypervisor (RHEV-H)*, que é a plataforma de virtualização para prover hospedeiros para as VMs que, junto ao *Red Hat Enterprise Virtualization Manager (RHEV-M)*, instancia e gerencia todos os componentes RedHat para completar a solução de virtualização. Para garantir a segurança dos dados, é utilizado o *LDAP Directory Active Directory*, que é o componente responsável pela autenticação do usuário antes do acesso a VM.

Assim, para requisição de *login* em sua VM o usuário deve conectar-se ao *RHEVM login page* e informar suas credenciais. Após a validação, é informada ao usuário uma lista com

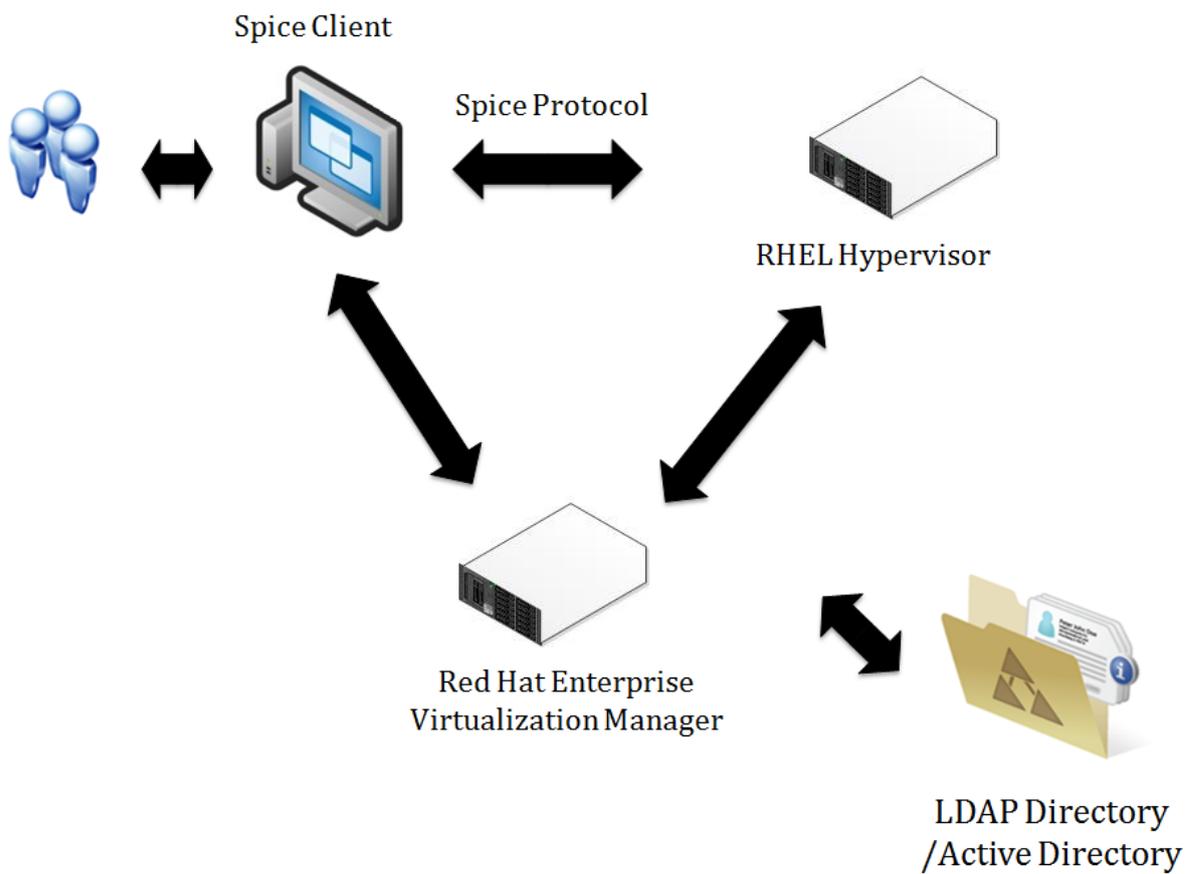


Figura 2.12: Arquitetura de softwares SPICE, adaptado de (WANG; LIANG, 2011)

todas suas VMs; após selecionar a VM que deseja conectar, o cliente cria a conexão utilizando o protocolo SPICE até o destino.

2.4.4 Pilha de protocolos e software Microsoft RDS

RDS (*Remote Desktop Service*) é o *Terminal Service* (Nome do serviço de ambiente de trabalho remoto) da Microsoft (WANG; LIANG, 2011). Microsoft tenta atender o conceito de virtualização em diferentes ramos e para tal disponibiliza diferentes soluções de virtualização, que são divididas por conceito, como virtualização de aplicações, apresentação, gerenciamento, *desktops* e servidores. A virtualização de *desktops* e servidores utiliza sistemas hospedeiros *windows* para criar e disponibilizar recursos virtuais. Já os outros tipos de virtualização disponibilizados pela Microsoft são tarefas gráficas e aplicações por demanda, por exemplo softwares como serviço como o *OneDrive* (serviço de *storage* na nuvem da Microsoft).

Assim como os outros controladores, o *software* de virtualização da Microsoft possui uma aplicação que deve ser instalada no cliente de acesso; essa aplicação é o RD Client, como pode ser visto na figura 2.13. RD client realiza a conexão RDP com as VMs remotas e para isso, deve

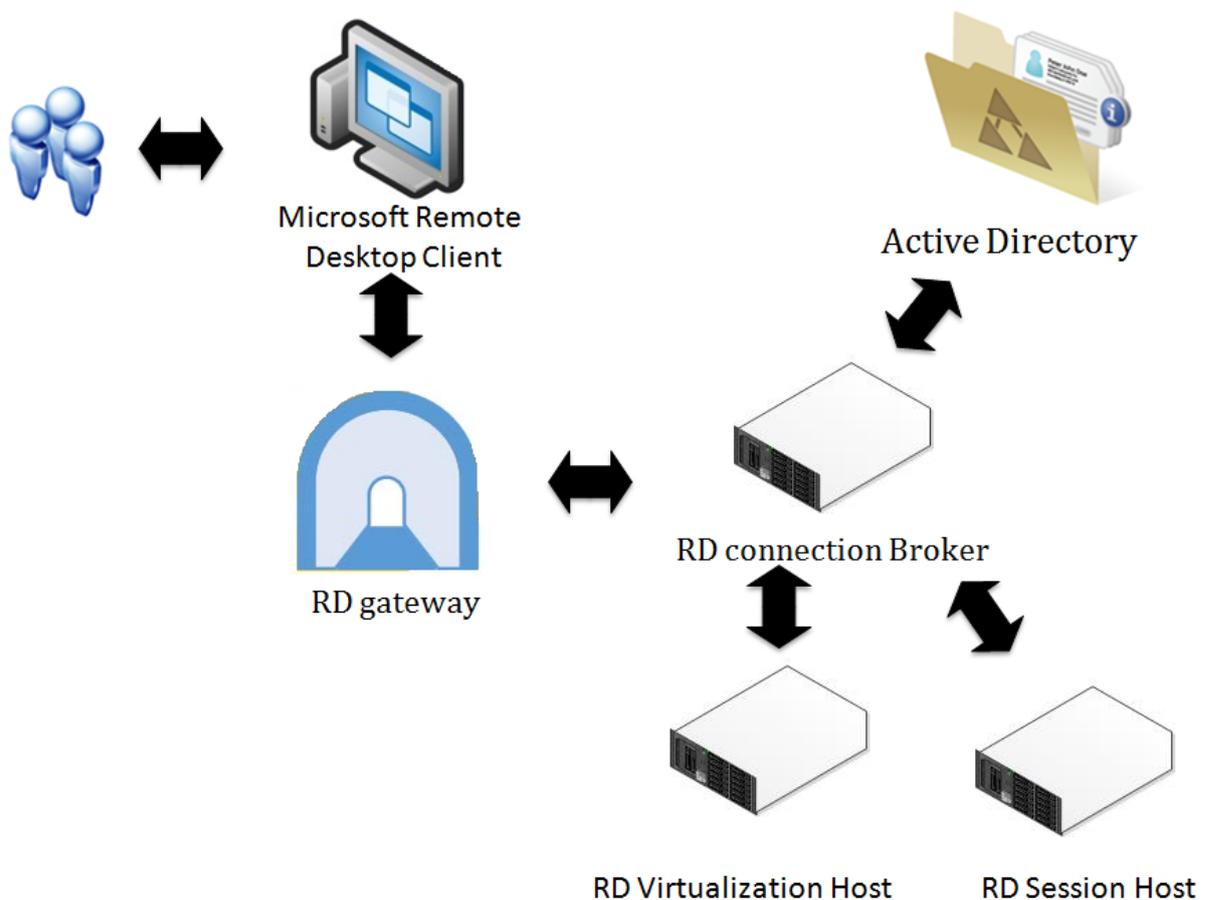


Figura 2.13: Arquitetura de softwares Microsoft, adaptado de (WANG; LIANG, 2011)

primeiro conectar-se ao *RD Gateway (RDG)* com objetivo de garantir autenticação e segurança das conexões remotas com seus respectivas VMs.

A conexão, após autenticação, é encaminhada ao *RD Connection Broker (RDCB)* responsável pela comunicação entre usuário e provedor. É este componente que cria a sessão entre o usuário e o provedor, além de alocar dinamicamente a VM do usuário. Assim, quando o usuário efetua o *login*, o *broker* seleciona um *host* para a VM daquele determinado usuário. O próximo componente acionado é o *Remote Desktop Session Host (RDSH)*. Este componente pode ser definido como o *host* e é nele que será de fato executado o cliente RD, com aplicações baseadas em *Windows* ou com o sistema *Windows* completo. O usuário pode se conectar a esse *host* para executar suas aplicações, verificar seus arquivos e utilizar conexões de rede no seu servidor.

O responsável por prover os recursos, utilizados como *host* para as VMs, é o *RD Virtualization Host (RDVH)*. Por fim, o *Active Directory*, assim como nas outras arquiteturas citadas nesta seção, é o responsável por autenticar e garantir a segurança das conexões entre clientes e

VMs. Para acessar sua VM o usuário solicita a sessão ao RD, que funciona como um redirecionador. Após o cliente se conectar ao RD e terminar a autenticação, a conexão é redirecionada ao *Connection Broker*, que busca pelas VMs do usuário requisitante. Após recuperar a VM, esta é então executado em um dos *hosts*. Caso esteja desativada, a VM será ativada e terá então suas informações redirecionadas ao cliente. Um tunelamento é então criado até o cliente de acesso, que agora pode conectar-se a sua VM (WANG; LIANG, 2011).

O protocolo utilizado nessa comunicação é o RDP. Esse protocolo provê conexão remota para VDIs, incluindo *display* e *inputs* que funcionam sobre conexões de rede para aplicações Windows (KOURIL; LAMBERTOVA, 2010). RDP é uma extensão do protocolo T-120 e herda as funcionalidades deste protocolo, como a capacidade de gerenciar várias conexões de aplicativos, usuários e de garantir segurança sobre diferente conexões de rede. RDP possui *multipoint data delivery* para distribuir dados em tempo real para diferentes conexões, sem criar fluxos individuais para cada uma.

RDP possui vários canais destinados a transmissão de dados, porém as transmissões normalmente só utilizam um único canal para teclado, mouse e dados. RDP também oferece suporte para topologias de rede como ISDN, POTS, e muitos protocolos de LAN, como IPX, NetBIOS e TCP/IP. Na versão 7, RDP executa apenas sobre TCP/IP e apresenta algumas novidades, como o suporte ao *Aero desktop environment*, *System Direct 2D e 3D*, grava e reproduz vídeos em alta resolução sem perder a sincronização entre vídeo e áudio. RDP utiliza serviço de canais para diminuir a perda de comunicação entre cliente e servidor, onde cada canal é utilizado para determinada aplicação como redirecionamento de impressoras, clipboard, etc. Podem existir até 31 canais estáticos por conexão e também existe a possibilidade de canais virtuais.

O protocolo RDP possui, ainda, serviço de segurança padrão e avançada. O serviço padrão realiza basicamente troca de chaves criptografadas com garantias. Já a opção avançada é feita por componentes externos, como protocolos TLS. Gerenciamentos para balanço de carga e economia de banda também são opções de serviços possíveis para esta arquitetura. RDP é conceitualmente semelhante ao protocolo ICA da Citrix, fato que pode ser fundamentado pelo fato de ter sido constituído após a implementação dos protocolos ICA nos sistemas operacionais da Microsoft. Ambos seguem o conjunto de padrões OSI (*Open Systems Interconnection*) para envio e recepção de dados em LAN (SOUZA FILHO, 2003). As funcionalidades de controle, tela cheia, teclado e movimento de mouse também estão presentes no protocolo RDP, conforme apresentados previamente no capítulo sobre ICA, assim como ambos seguem os mesmos princípios e funcionalidades não é necessário apresentar os diagramas de protocolo novamente.

2.5 Técnicas de Otimização para gerenciamento de recursos virtuais

A junção da virtualização com a computação em nuvem possibilita a criação de recursos computacionais que podem ser utilizados para instanciar os recursos virtuais (VR, acrônimo em inglês para virtual resources). No entanto, para que a máquina virtual seja instanciada neste novo servidor, é necessário a migração dos dados. A migração é utilizada para melhorar a alocação do máquina virtual, seja para melhorar desempenho ou economizar energia. Uma técnica interessante, proposta pela virtualização, é a consolidação de recursos, que utiliza da migração para alocar o máximo de máquinas virtuais possível em um mesmo *hardware* físico sem comprometer demasiadamente seu desempenho. Essa técnica permite que alguns servidores sejam desativados buscando diminuir o consumo energético e uma melhor utilização do *hardware*, evitando capacidade ociosa dos servidores.

Esta seção descreve então algumas das técnicas de otimização utilizadas por provedores de recursos para otimizar o provisionamento e o gerenciamento dos VRs, que são *desktops* virtuais ou máquinas virtuais. Essas técnicas realizam melhorias durante a inicialização, migração e alocação dos VRs, diminuindo a quantidade de arquivos em uma possível migração, diminuindo concorrência pela rede, diminuindo utilização de recursos físicos como disco e memória da máquina *host*.

As técnicas apresentadas nesta seção foram utilizadas de forma implícita durante os experimentos realizados para validação do modelo apresentado neste trabalho. Sem a utilização de nenhuma otimização, a quantidade de arquivos durante transmissão e transferência dos VRs seria muito maior do que a quantidade utilizada nos experimentos. Isso poderia gerar resultados imprecisos e que não são a realidade dos provedores de serviço, que normalmente utilizam técnicas como às apresentadas nesta seção. Esta seção é iniciada pelas técnicas básicas de otimização até as técnicas um pouco mais avançadas, apresentando também trabalhos realizados na área.

2.5.1 Caching e Compartilhamento de Blocos

Estes dois conceitos são normalmente encontrados nas técnicas de otimização em gerenciamento de VRs. Técnicas de otimizações como *prefetching*, *copy-on-write* e *copy-on-read* que são apresentadas nas próximas seções, utilizam caching e compartilhamento de blocos em suas estratégias. A técnica de *caching* está diretamente relacionada ao compartilhamento de blocos, pois os blocos compartilhados podem ser armazenados em diferentes tipos de *cache* para apri-

morar, ainda mais, essa otimização. Utilizando essas duas técnicas juntas, pode-se diminuir a sobrecarga no cache e diminuir o número de blocos armazenados. Outra vantagem é que não é necessário transmitir diversas vezes um mesmo bloco da imagem, que seja compartilhado, para diferentes VMs, diminuindo assim a utilização da rede, que já é concorrida por diferentes usuários e aplicações (JONES; ARPACI-DUSSEAU; ARPACI-DUSSEAU, 2006).

O *cache* pode ser configurado, pelo provedor de serviços, como um *storage* local que irá armazenar os arquivos referentes as diferentes VMs instanciadas no *datacenter*. Esse *storage* será compartilhado por uma rede interna privada e, com isso, aumenta-se a velocidade e evita-se sobrecarga da rede compartilhada durante o acesso aos arquivos das VMs que ali estão armazenados. Uma implementação da otimização de compartilhamento de blocos é o armazenamento local, onde parte da imagem base, utilizada pelas VMs desse provedor, é armazenada em um *cache* de memória da máquina *host* e então distribuída para as VMs instanciadas neste servidor, como é ilustrado na figura 2.15 que apresenta a diferença entre os modelos de imagem básico, COW, e otimizado, CQOW2. Essa implementação aumenta a velocidade na recuperação e diminui o tempo de acesso a imagem compartilhada. Outra técnica semelhante foi implementada utilizando *storages* SSD para armazenar partes semelhantes do sistema operacional e assim distribuir entre as VMs alocadas no *datacenter* (HE, 2013). Com essa técnica, foi possível melhorar em até 90%, no pico, e média de 50% na velocidade de distribuição.

2.5.2 Copy - on - Write

Copy-on-Write (Inglês para copiar ao escrever, COW) é uma técnica de otimização que utiliza imagens, que contenham partes semelhantes, para economizar espaço em disco durante sua execução. Isso quer dizer que, se múltiplas VMs forem instanciadas a partir de um modelo de imagem base, podem e devem conter partes semelhantes que podem ser compartilhadas, como o *kernel*, alguns *software* base e até mesmo algumas aplicações. Assim, nesta técnica, é criada uma imagem base que contém as informações semelhantes de diferentes instâncias de VMs, como pode ser visto na primeira parte da figura 2.14, chamada de *L1 Table*. Nessa mesma imagem, *L2 Table* representa as modificações realizadas em uma VM específica durante sua execução. Essas modificações são armazenadas em *snapshots* complementares que devem ser carregados junto com esta VM quando ela for instanciada. Com isso, a técnica economiza armazenamento em um primeiro momento, evitando que um *snapshot* completo, contendo toda a imagem, seja criado para cada VM. Além de que, caso seja necessário atualizar várias VMs basta alterar a imagem base, assim, a modificação é espalhada pelas diferentes VMs que apontam para essa imagem. Atualizações específicas necessitam de modificações nos *snapshots*

complementares de cada VM (CHEN; WO; LI, 2009).

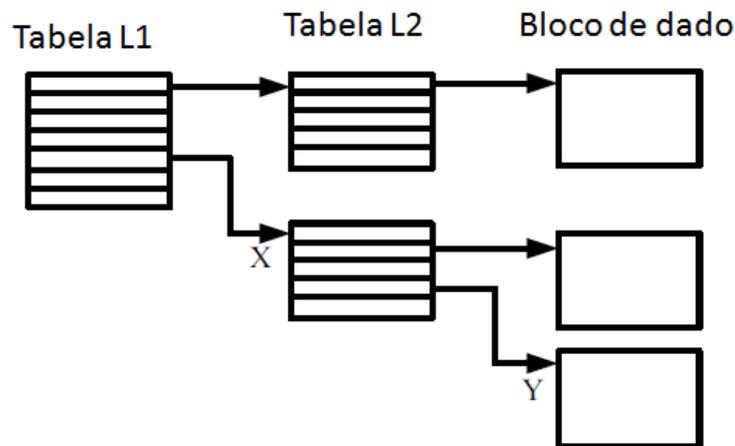


Figura 2.14: Diferentes index de imagem virtual (TANG, 2011)

Esta técnica torna-se ineficiente quando aumenta-se o número de VMs e *snapshots* no servidor, pois durante a instanciação da VM deve-se recuperar os vários *snapshots* complementares criados especificamente para esta VM que está sendo instanciada. Assim, com diversos *snapshots* de diferentes modificações, a técnica perde um pouco de eficiência de disco e rede, que também perde em desempenho devido à concorrência gerada durante a execução de várias VMs. Além de que, o tráfego gerado pelos recursos virtuais somado ao tráfego atual existente na rede, pode resultar em problemas como perda de pacotes e atraso na entrega de blocos.

Existem otimizações deste modelo como o *QCOW2* (*QEMU - copy-on-write2*), que atrasa ao máximo a migração da imagem para que seja copiada somente quando realmente for necessário. Durante a alocação às VMs são iniciadas nos servidores do provedor de serviço, seguindo o modelo *EC2* (*Elastic compute cloud - Amazon*) e possuem armazenamento *DAS* (*Directly Attached Storage*) sem custos adicionais (TANG, 2011). Esse tipo de armazenamento efêmero reduz a velocidade do processo de criação e migração de imagens, o que diminui os benefícios de nuvens elásticas, isso ocorre porque durante a instanciação ou migração uma cópia dos dados deve ser realizada para cada VM. Armazenamento persistente como *NAS* (*Network Attached Storage*) é mais caro, cobra pelo espaço e quantidade de acesso utilizado, porém é acessível por todos os *hosts* compartilhados na rede. Configurações avançadas em um servidor evitam que todas as VMs sejam instanciadas utilizando *NAS*, é preferível que a imagem esteja alocada em um cache local. O problema, porém, não é completamente solucionado, já que esta imagem cache necessita estar sempre atualizada, pois caso esteja desatualizada, as VMs vão ser instâncias desatualizadas.

Na imagem 2.15, são apresentados os dois modelos de otimizações. No modelo COW,

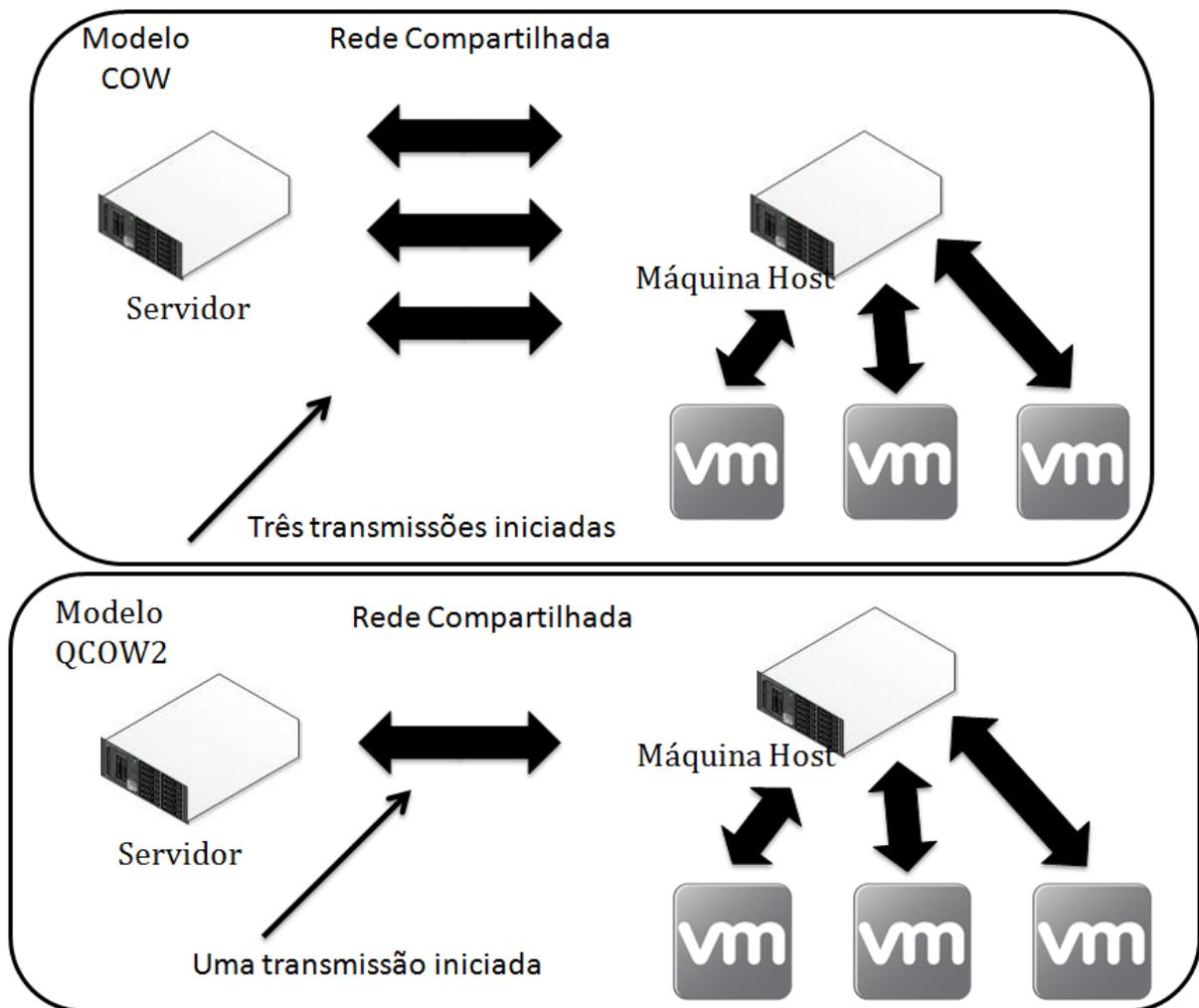


Figura 2.15: Modelos de otimizações COW e QCOW2

quando são iniciadas três VMs, há três tráfegos iniciados pela rede. Isso gera um tráfego intenso na rede, afinal, a rede é compartilhada também com outros usuários e aplicações. Já o modelo otimizado QCOW2 gera um único tráfego que é ligado com a imagem base utilizando NAS, criando uma imagem DAS local para cada VM, e isso permite que as VMs sejam iniciadas em um menor tempo, pois acessam somente o cache na máquina *host*.

2.5.3 Copy - on - Read e Prefecthing

Esta técnica de otimização tem como objetivo diminuir a utilização de rede, que pode ser sobrecarregada por técnicas como o *copy-on-write*. A ideia principal desta técnica é realizar transmissões de dados e arquivos pela rede em ciclos, de preferência em períodos ociosos da mesma, quando for instanciar ou migrar uma VM. O primeiro ciclo transfere apenas o necessário para iniciar a VM e, então, nos próximos ciclos, transfere progressivamente os demais arquivos conforme necessidade. A grande dificuldade desta técnica é identificar a prioridade

dos arquivos que devem ser transferidos.

Esta técnica possui otimizações semelhantes ao citado no *QCOW2*, como a cópia local da imagem alterada do modelo NAS para o modelo DAS, com intuito de criar *cache* local atualizado. Esta cópia local da imagem base é criada para que as solicitações dos VMs sejam atendidas de forma local e não seja gerada concorrência na rede compartilhada (LIGUORI; VAN HENBERGEN, 2008). Uma otimização importante desta técnica é o *prefetching* adaptativo que monitora a rede, observando momentos ociosos para realizar a transferência dos arquivos, durante a instanciação ou migração de VMs. Assim, a otimização, se estende também à rede, pois realiza transferências somente em momentos de pouca utilização e concorrência da rede. Durante transferência dos arquivos, caso a rede comece a ter concorrência, a técnica de *prefetching* interrompe a transferência, deixando a rede disponível para o tráfego concorrente, continuando a transferência ou quando realmente necessário ou em um novo momento ocioso da rede.

Na figura 2.16 é apresentado o novo formato de imagem de VM chamado FDV (TANG, 2011). Esse novo formato utiliza as vantagens combinadas de técnicas como *copy-on-write* e *copy-on-read* com *prefetching*. Assim, foi proposto um novo modelo de imagem que instancia a VM com a quantidade mínima de dados necessária para iniciar, e durante períodos ociosos da rede transfere o restante dos arquivos da VM. Como podemos ver na parte a) da figura 2.16 a VM só está pronta após um longo período de inicialização, devido ao tempo de transferência de dados pela rede. Já na parte b), utilizando o FDV, a VM está pronta após um primeiro ciclo e durante outros ciclos, intercalados com a concorrência da rede, transfere o restante dos arquivos para a VM. Esta técnica pode ter problemas de espera caso a VM solicite acesso a alguma página de memória que ainda não foi transferida, porém em muitos casos a abordagem é melhor do que aguardar a transferência completa dos dados.

2.5.4 Algoritmos de migração

Além dessas técnicas de otimizações apresentadas nesta seção, existem também os algoritmos para otimizar a migração de VMs entre servidores. Sem a utilização desses algoritmos de otimização, o tempo da migração de uma VM pode ser muito elevado e causar indisponibilidade para o usuário, o que é desagradável e também é contra a qualidade de serviço.

O primeiro algoritmo, apresentado na figura 2.17, é uma técnica de migração chamada pré-cópia, que tem como objetivo copiar toda a memória da VM para o servidor destino antes de começar sua execução. Essa técnica é executada em ciclos, onde a cópia da memória para o novo hospedeiro é efetuada no primeiro ciclo, enquanto os outros ciclos atualizam a memória

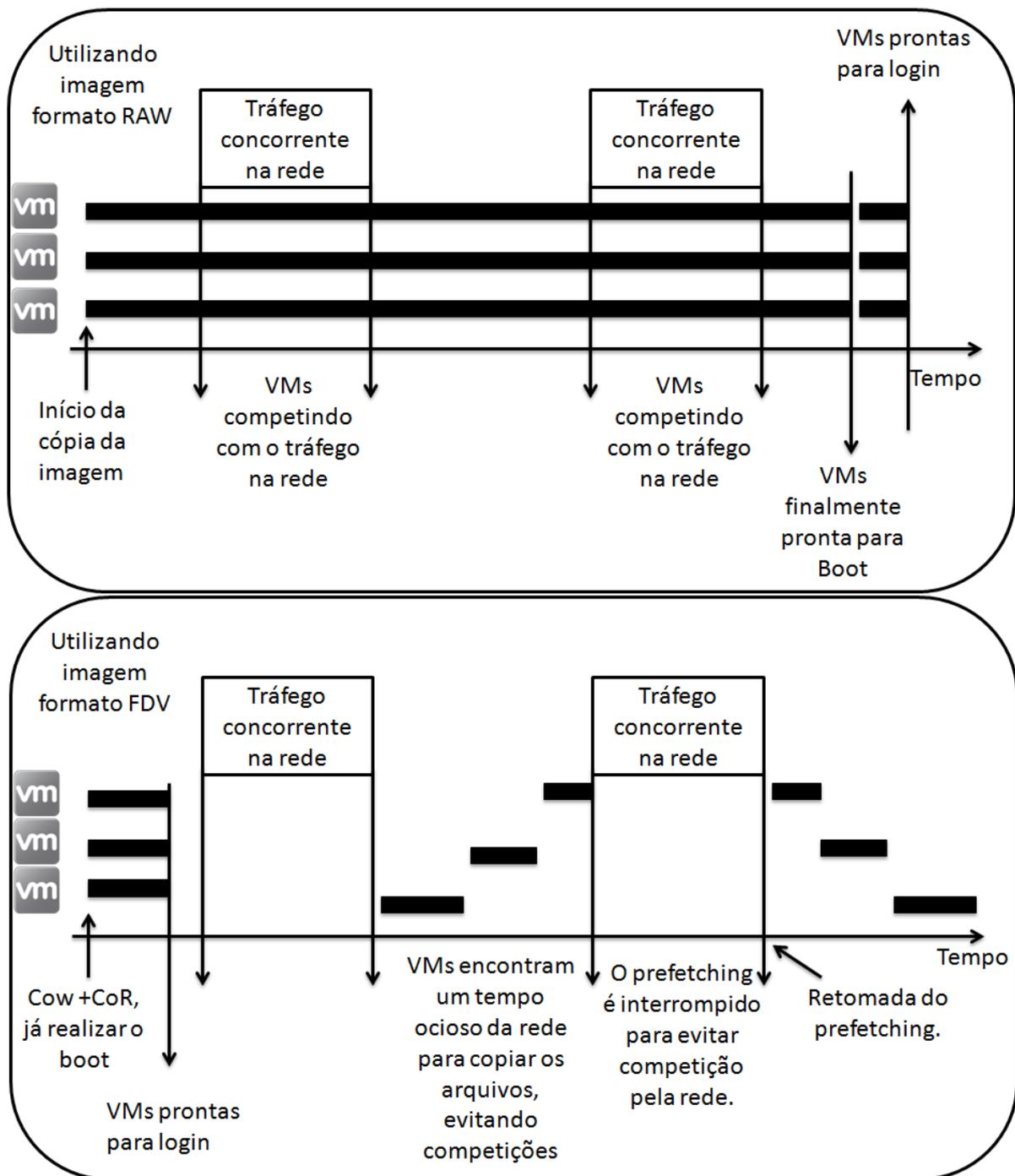


Figura 2.16: Modelo de imagem FDV, adaptado de (TANG, 2011)

na máquina destino até que esta esteja alinhada com a máquina fonte. Assim que terminados os ciclos de transferência, a máquina fonte é reiniciada no hospedeiro destino quase que imperceptivelmente ao usuário (MAGALHÃES; SOARES; GOMES, 2011). O problema é que se os ciclos de memória continuarem infinitamente a migração nunca termina; logo um limite para a migração é necessário. A otimização pode acontecer identificando os ciclos com maior número de alterações e copiá-los apenas no último ciclo, quando a máquina inicial já está parada (MA;

LIU; LIU, 2010).

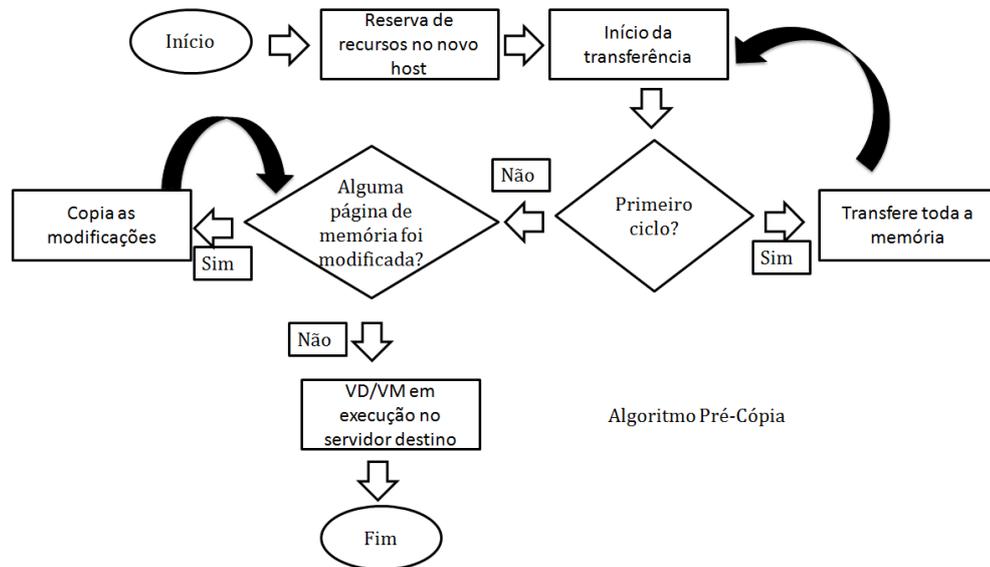


Figura 2.17: Algoritmo Pré-Cópia, adaptado de (BARUCHI, 2010)

O segundo algoritmo é o algoritmo de pós-cópia para migração, conforme figura 2.18. Este algoritmo tem como objetivo migrar e instanciar uma VM de maneira mais rápida possível até o servidor destino e depois, conforme necessidade, completar a transferência dos dados. O algoritmo funciona basicamente assim: i) em um primeiro momento, a VM está com a memória e os registradores em execução no servidor fonte; ii) quando é solicitada a migração, o controlador faz uma cópia, em ciclos, das informações necessárias para iniciar a VM no servidor destino; iii) após o primeiro ciclo de transferência, a VM então é iniciada no servidor destino do ponto em que foi parada no servidor fonte; IV) Após ser iniciada o servidor fonte continua as transferências dos demais dados, conforme necessidade, em diferentes ciclos (BARUCHI, 2010). Esse algoritmo base pode ser otimizado diminuindo a alocação de memória, antes de efetivamente parar a máquina. Para isso, uma das técnicas que podem ser utilizadas é o *Ballooning* (WALDSPURGER, 2002). Este mecanismo aumenta ou diminui a alocação de memória forçando a VM a utilizar algoritmos de paginação para descartar páginas de memória. Assim, logo que a VM atingir a alocação de memória esperada, a migração é iniciada, evitando assim a perda de memória durante a migração.

O *prefetching* também pode ser adicionado como otimização no algoritmo de pós-cópia. Assim, as páginas que têm grande chance de utilização futura são carregadas na memória do hospedeiro destino, mesmo sem estarem sob utilização no momento. Estas páginas são carregadas em momentos ociosos da rede quando o algoritmo que está monitorando o sistema percebe a chance de copiar. Com isso diminui-se o número de páginas em falta no hospedeiro destino quando a VM terminar a migração. Resumidamente, a técnica *copy-on-read* copia o *working set*

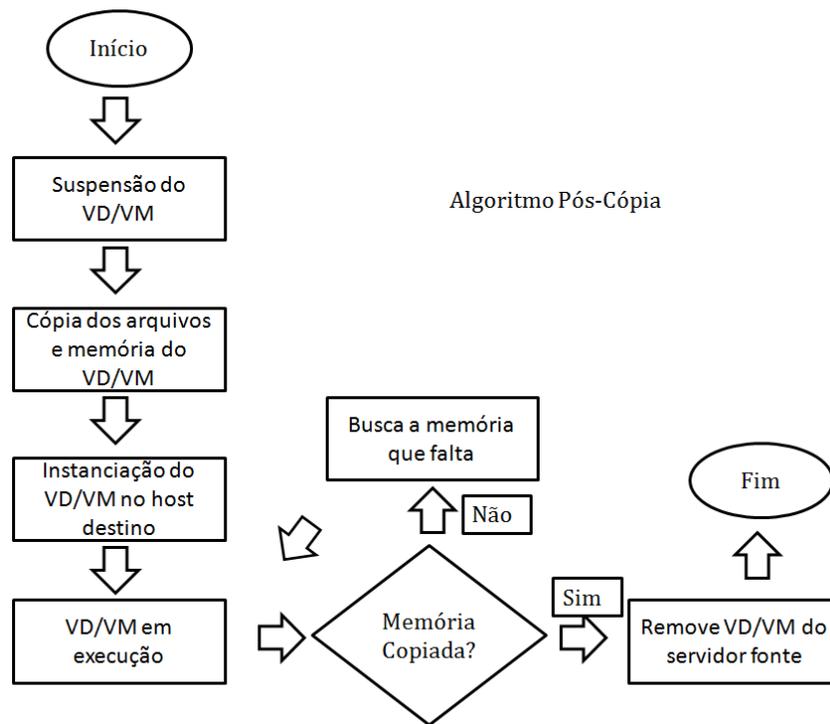


Figura 2.18: Algoritmo Pós-Cópia, adaptado de (BARUCHI, 2010)

mínimo para iniciar a VM, contendo as páginas mais acessadas da VM. Assim, o desafio maior dessa técnica é identificar com precisão as páginas mais utilizadas e ordená-las com prioridade para os ciclos de cópia.

2.6 Considerações finais do capítulo

Os fundamentos teóricos e tecnológicos, como os apresentados nas primeiras seções deste capítulo 2, são essenciais para o entendimento do modelo que será proposto. Nesta dissertação, muitos desses conceitos são utilizados de forma implícita e por isso é necessário que sejam apresentados. Alguns desses conceitos são importantes para o entendimento sobre a virtualização de *desktops*, a possibilidade de sua migração entre servidores e os benefícios que a migração pode trazer, como a eficiência energética.

Já as técnicas de otimizações, apresentam alternativas para os provedores realizarem a alocação e migração, quando necessária, de recursos virtuais de forma otimizada. Essas técnicas apresentam ideias que, quando utilizadas, vão resultar em menor utilização de recursos de armazenamento, memória e rede dos provedores de recursos.

Capítulo 3

TRABALHOS RELACIONADOS

Estimar com precisão o consumo energético durante a execução de um VD, não é uma tarefa simples. Diferentes estudos já foram feitos para medir, calcular e estimar, em diferentes situações, o consumo energético em computação. Alguns desses estudos estão relacionados ao objetivo deste trabalho, fornecendo dados e informações necessárias para que seja possível criar um modelo capaz de estimar o consumo de energia de um VD.

O consumo energético total de instanciar um VD em um servidor é composto por diferentes custos energéticos. Os custos energéticos são compostos desde a parcela de utilização dos recursos físicos no servidor em que o VD é instanciado, até o custo energético do cliente de acesso utilizado para comunicação entre cliente/servidor. Para realizar essa comunicação é necessária a transmissão de quadros de tela e eventos de teclado e mouse, que são transmitidos normalmente pela rede *WAN* (*Wide Area Network*), rede essa que possui um custo energético elevado por envolver mais *hardware* para transmissão do bit. Em algumas situações, os provedores de serviço, onde são executados os VDs, podem decidir migrar esses VDs entre seus diferentes servidores e isso gera um custo energético que pode ser ainda maior devido ao custo energético de transferência dos arquivos referentes a esse VD.

3.1 Trabalho de (BALIGA, 2011)

Em (BALIGA, 2011), foi desenvolvido um modelo para estimar o consumo energético de *software* executado como serviço. Como parte do modelo, também foi desenvolvida uma estimativa do consumo energético da transmissão por bit em diferentes tipos de redes, como *WAN* e *LAN* (*Local Area Network*). Esse valor é calculado em função do número de saltos (*hops*) necessários e da quantidade de dispositivos envolvidos para transmitir os bits. Nas estimativas de custos propostas são avaliadas as capacidades, redundâncias dos dados e a refrigeração dos

servidores.

$$P_{sf} = P_{sf,PC} + \frac{1.5P_{sf,SR}}{N_{sf,SR}} + 2B_d \frac{1.5P_{SD}}{B_{SD}} + AE_t \quad (3.1)$$

$$P_{st} = \frac{B_d D}{3600} \left(E_t + \frac{1.5P_{st,SR}}{C_{st,SR}} \right) + 2B_d \frac{1.5P_{SD}}{B_{SD}} \quad (3.2)$$

Na equação 3.1, retirada do trabalho de (BALIGA, 2011), P_{sf} refere-se ao consumo energético que cada usuário, utilizando *software* como serviço, gera em *Watts* ao servidor. A equação é composta pela somatória de fatores como o ($P_{sf,PC}$) que é o consumo energético do cliente utilizado pelo usuário para conectar ao software como serviço. Outro fator é a razão entre o consumo energético do servidor ($P_{sf,SR}$) e o número de usuários conectados a este servidor ($N_{sf,SR}$). Nesse segundo fator, ainda existe um ajuste com uma multiplicação de 1.5 para refrigeração e outros *overheads*. Outro fator adicionado a está somatória é a razão entre (P_{sd}), consumo energético dos discos que são utilizados para armazenamento de dados no servidor, pela capacidade dos mesmo (B_{sd}). Este terceiro fator ainda é multiplicado pelo tamanho do arquivo (B_d) e um fator de 2 para redundância de dados. O quarto fator adicionado nesta equação 3.1 é referente ao custo energético da transmissão de tela e comandos, que será feita entre o cliente que o usuário está utilizando para se conectar e o servidor destino. Este fator é a multiplicação do número de *frames* transmitidos por segundo, onde a transmissão da tela completa representa um *frame* completo, (A) e o consumo por bit transferido (E_t) que é o consumo energético gasto pela transferência de um bit pela rede.

Na segunda equação, também retirada do mesmo trabalho de (BALIGA, 2011), é apresentado um modelo que visa estimar o consumo energético referente ao armazenamento de dados em um servidor de armazenamento. Logo, temos na equação 3.2 o cálculo do P_{st} , que é o consumo energético do serviço de armazenamento. Essa equação é composta pela somatória de dois fatores, o primeiro é a multiplicação do tamanho do arquivo que será transferido entre os servidores (B_d) pelo número de *downloads* efetuado por hora (D), dividido por 3600 para ajuste em segundos. Essa divisão é então multiplicada a uma soma entre o consumo por bit transferido na rede (E_t) e a razão entre consumo do servidor do provedor ($P_{st,SR}$) e capacidade do servidor do provedor ($C_{st,SR}$). A equação é então multiplicada pelo fator de ajuste de 1.5 referente a refrigeração e *overhead*. O segundo fator somado é a multiplicação entre a divisão do consumo dos discos de armazenamento no servidor destino (P_{sd}) pela capacidade dos discos do servidor destino (B_{sd}) e o tamanho do arquivo a ser transferido. Nesse segundo fator ainda existe o ajuste de 2 para redundância de arquivos, onde os arquivos são replicados em servidores

diferentes gerando um *backup* para tolerância a falhas.

Uma das principais contribuições apresentada em (BALIGA, 2011), que é utilizada em alguns cenários da avaliação do modelo deste trabalho, é o custo energético para transferir um bit pela rede. Este fator impacta diretamente no resultado das equações e é estimado como podemos ver a seguir na equação 3.3.

$$E_i = 6 \left(\frac{3P_{es}}{C_{es}} + \frac{P_{bg}}{C_{bg}} + \frac{P_g}{C_g} + \frac{2P_{pe}}{C_{pe}} + \frac{2 * 9P_c}{C_c} + \frac{8P_w}{2C_w} \right) \quad (3.3)$$

Essa equação 3.3 estima o custo energético de transmissão por bit entre diferentes *hosts* pela rede de internet (WAN). Onde são considerados para isso os seguintes fatores P_{es} , P_{bg} , P_g , P_c , P_w que são respectivamente o consumo dos *Ethernet switches*, *broadband gateway routers*, *datacenter gateway routers*, *provider edge routers*, *core routers*, e *WDM transport equipment*. Esses fatores são então divididos pela suas capacidades que são C_{es} , C_{bg} , C_g , C_{pe} , C_c , C_w respectivamente. Os fatores de ajuste são para redundância, refrigeração e outros *overheads*. Esses fatores foram calculados com base em estimativas de saltos (*hops*) que um bit percorre entre o cliente e o servidor. Assim, pode-se estimar o consumo energético de um bit ao passar por cada um deles, e então utilizar esse fator para calcular o custo de transmitir vários bits que compõem um arquivo.

3.2 Trabalho de (FALVO, 2014)

Um estudo sobre a qualidade de experiência do usuário (QoE) de VDs foi apresentado em (FALVO, 2014). O objetivo desse trabalho é avaliar a satisfação do usuário de VD com várias aplicações interativas com diferentes requisitos. As aplicações utilizadas foram um navegador Web, um IDE de desenvolvimento de software, um reprodutor de vídeo, um software de cálculos matemáticos realizando uma operação 3D em tela e um editor de textos. O estudo apresenta um conjunto de medições do consumo de recursos (cpu, largura de banda, memória e IOPS), além do número médio de bits transmitidos, entre cliente e servidor, para quadro de telas e comandos por aplicação. Tais valores oferecem uma referência do consumo de recursos.

Nos experimentos apresentados em (FALVO, 2014), foi configurada uma arquitetura que possibilitava simular a qualidade da rede e do acesso remoto aos VDs. Foram realizados diferentes experimentos onde alguns VDs foram instanciados com diferentes aplicações em execução. Os experimentos foram executados com 1, 5, 10, 15, 20 e 25 VDs instanciados simultaneamente. Em um desses experimentos, foram executados testes reais com usuários, onde durante

o acesso ao seu VD eram propostas diferentes condições de qualidade de rede. Durante o experimento, a latência de rede aumentava e diminuía, assim foi possível medir qual o grau de satisfação do usuário durante diferentes latência de rede.

As figuras 3.1, 3.2 e 3.3 representam os resultados dos experimentos executados onde foram medidos o consumo de CPU, I/O e memória de acordo com tarefas executadas e quantidade de VDs instanciados no servidor. Para os experimentos foram utilizados 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.

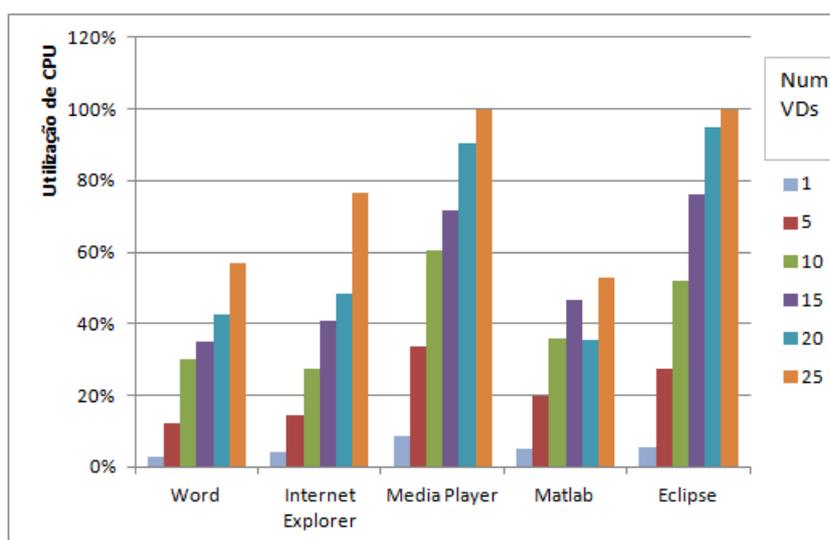


Figura 3.1: Utilização de CPU por aplicação e quantidade de Vds, adaptado de (FALVO, 2014).

Na figura 3.1 é possível observar que aplicações como o *Microsoft Word* e *Microsoft internet explorer* não alcançam a capacidade máxima de utilização de CPU do servidor, mesmo quando foram instanciados 25 VDs executando essas aplicações. A porcentagem máxima de utilização de CPU só foi atingida com a execução de aplicativos de mídia, utilizando o reprodutor *Windows Media Player* e aplicações de desenvolvimento utilizando o IDE Eclipse. Para operações de entrada e saída, apresentadas na figura 3.2, os valores crescem drasticamente quando aplicações como *Matlab* e o ide de desenvolvimento *Eclipse* são executados em 25 *desktops virtuais* instanciados simultaneamente. *Windows Media player* e *Internet explorer* têm um número médio alto, enquanto o *Microsoft Word* quase não sofre alterações em sua quantidade de E/S, mesmo com 25 VDs instanciados.

Em relação a memória, como apresentada na figura 3.3, os valores mais altos são atingidos utilizando o *Internet explorer*, aplicativo de operações *Matlab* e o ide de desenvolvimento *Eclipse*, para quando temos grandes quantidades de VDs. Assim, nas figuras 3.1, 3.2 e 3.3 ficam

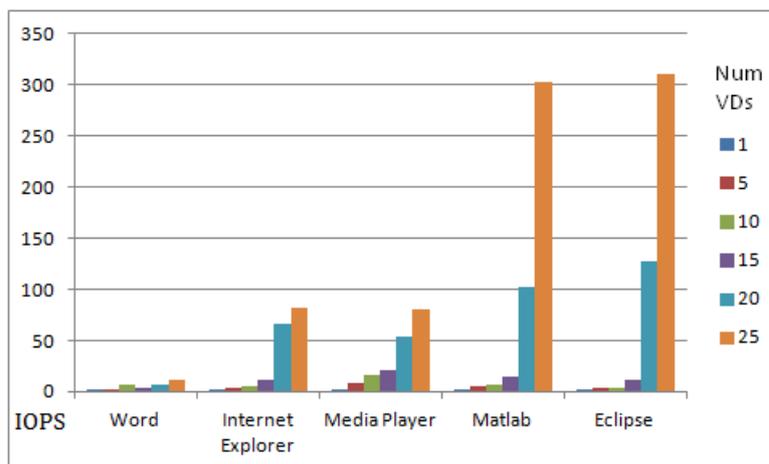


Figura 3.2: Operação de E/S por aplicação e quantidade de Vds, adaptado de (FALVO, 2014).

evidentes a quantidade de recursos do *hardware* utilizada executando diferentes aplicações. É possível concluir também que a memória ainda é o principal gargalo para as aplicações, uma vez que mesmo com 10 VDs em execução os valores começam a se aproximar dos máximos por quaisquer aplicações.

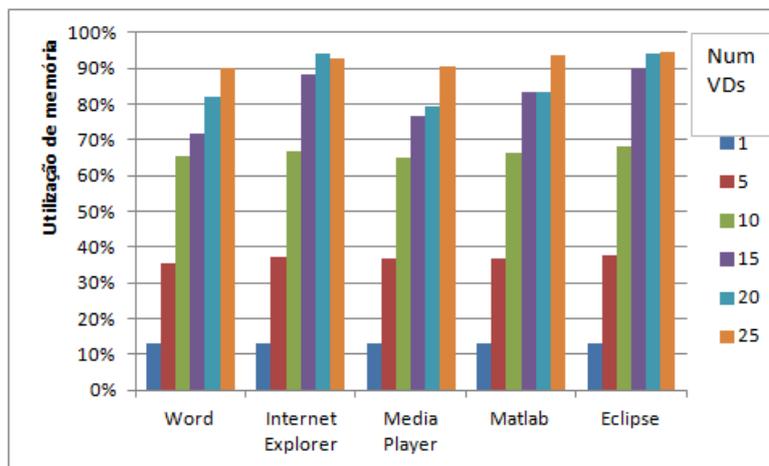


Figura 3.3: Utilização de memória por aplicação e quantidade de VDs, adaptado de (FALVO, 2014).

Por fim, na figura 3.4 é apresentada a taxa média de transmissão, entre cliente e servidor, de algumas aplicações. Nessas medições, realizadas em (FALVO, 2014), fica evidente o quão maior é a taxa de transmissão para aplicações de reprodução de mídia, como é exemplo do Media Player (MP). Essa aplicação MP transmite em média mais que o dobro de bits que a segunda aplicação que mais transmite desta imagem, que é o Internet Explorer. Algumas dessas taxas de transmissão são utilizadas em diferentes cenários do capítulo 5 deste trabalho para avaliar o modelo estimativo proposto.

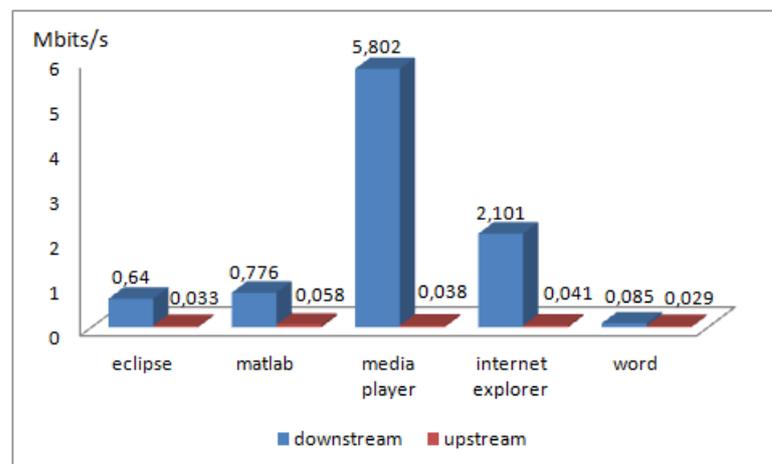


Figura 3.4: Taxa média de transmissão de dados entre cliente/servidor, adaptado de (FALVO, 2014).

3.3 Outros trabalhos

Existem outros trabalhos que também ajudam a embasar o modelo proposto no próximo capítulo, como (LIU, 2013) em que é apresentado um modelo até certo ponto semelhante a (BALIGA, 2011), pois ambos desenvolvem modelos de estimativas de consumo energético de servidores. O trabalho de (LIU, 2013), propõe também técnicas de otimização em migração de VDs, que podem reduzir o consumo energético em até 72%. Os experimentos apresentados em (LIU, 2013) mostram que durante a migração de um VD entre servidores houve uma elevação de 30W na potência consumida. Com base nos dados coletados, foi desenvolvido em (LIU, 2013) um modelo de previsão que possui como objetivo mostrar e melhorar o consumo energético em joules que a migração causa ao servidor. Alguns destes custos energéticos, como o custo do servidor fonte durante a migração, foram utilizados na composição do modelo proposto neste trabalho.

Outro trabalho, com abordagem diferente dos trabalhos anteriores, é apresentado em (ALBIERO, 2011) que utiliza diferentes aplicativos que geram carga de execução para sobrecarregar diferentes tipos de *hardware*. A utilização desses *hardware* passa por diferentes medições até atingir suas capacidades máximas, assim é possível extrair diferentes medições energéticas. Nos experimentos apresentados em (ALBIERO, 2011), esses aplicativos foram testados em diferentes máquinas com arquiteturas e sistemas operacionais diferentes, possibilitando analisar o comportamento de cada uma delas.

A tabela 3.5, retirada de (ALBIERO, 2011), apresenta alguns resultados e as diferentes variações conforme utilizações de número de núcleos, indicada por GO-X-YY, onde X é o número de núcleos utilizados em cada experimento e YY são os indicadores de medições de

memória (MA) ou padrão (P). O *hardware* analisado neste experimento foi um computador multiprocessado (SGI Altix XE310) com dois processadores Intel Quad Core de 2.0GHz, memória de 8GB, disco rígido SATA de 500GB e de 7200 RPM e fonte de alimentação ATX de 980W RMS. Quanto ao ambiente de software, foi utilizada a distribuição padrão do sistema operacional GNU/Linux Gentoo e o utilitário top.

Estado	Taxa de Uso do Processador	Taxa de Uso da Memória	Consumo de Energia	Consumo de Energia
G0_1_Mi	1%	20%	235 W	268 VA
G0_8_Ma	100%	100%	318 W	363 VA
G0_1_P	12,5%	20%	247 W	278 VA
G0_2_P	25%	20%	258 W	290 VA
G0_3_P	37,5%	20%	270 W	301 VA
G0_4_P	50%	20%	280 W	312 VA
G0_5_P	62,5%	20%	289 W	324 VA
G0_6_P	75%	20%	300 W	337 VA
G0_7_P	87,5%	20%	309 W	350 VA
G0_8_P	100%	20%	317 W	361 VA
G0_1_M	12,5%	100%	246 W	279 VA

Figura 3.5: Consumo de energia versus taxa de utilização de memória e processador, com Gentoo (ALBIERO, 2011).

Outros cenários de experimentos com diferentes tipos de *hardware* foram apresentados em (ALBIERO, 2011) e diferentes comparações foram executadas, com diferentes testes de medições realizados e resultados coletados. Assim, com esses dados apresentados, é possível estimar o consumo energético de diferentes aplicações que podem ser executadas em um VD. Relacionando essas diferentes informações apresentadas nos diferentes trabalhos deste capítulo, foi possível propor e avaliar cenários baseados em dados reais para validar o modelo estimativo proposto no próximo, apresentado capítulo desta dissertação.

3.4 Considerações finais do capítulo

Nesse capítulo foram apresentados trabalhos relacionados ao tema dessa dissertação. Esses trabalhos apresentam informações que foram utilizadas durante o desenvolvimento e avaliação do modelo, conforme será apresentado no capítulo 4. As variáveis do modelo são modificadas de acordo com as configurações do *datacenter* em que o VD é instanciado e, também, pelas características do próprio VD. Essas variáveis sofrem alteração mediante diferentes situações, resultando em custos energéticos diferentes e impacto no resultado final da estimativa energética. Alguns dos dados e informações que serão utilizados nos experimentos do capítulo 5, vieram de estudos relacionados, conforme apresentado neste capítulo 3.

Capítulo 4

UM MODELO PARA ESTIMATIVA DO CONSUMO DE ENERGIA DE DESKTOPS VIRTUAIS

Como apresentado no começo deste trabalho, centros de dados estão crescendo rapidamente, tanto em quantidade de recursos quanto no consumo de energia. Por isso o uso de um modelo estimativo é uma alternativa a usuários que procuram eficiência energética porém não possuem informação disponibilizadas pelos DCs e não possuem equipamentos de medição energética capazes de trazer essa informação. Medidores de energia também não resolvem por completo o problema do consumo energético em VD, pois não poderiam ser instalados em todos os servidores disponíveis e, além disso, existem outros custos energéticos além apenas do consumo dos servidores.

Outro problema que a proposta deste trabalho tentará evitar é referente ao consumo energético gerado por migração dos *desktops* virtuais (VDs) que estão em execução. A migração é utilizada por *datacenters* (DCs) para realizar o balanceamento de carga entre seus servidores ou consolidar recursos, com objetivo de melhorar desempenho dos mesmos. Consolidar e evitar sobrecarga de servidores normalmente ajuda a diminuir o consumo energético dos DCs, porém, se não for feito de maneira apropriada, pode trazer custos energéticos ainda maiores devido a transferência dos arquivos que representam esses VDs, além de perda de desempenho (DE LA VIGLIOTTI; MACEDO BATISTA, 2014). Logo, estimar previamente, com base no tempo de execução do VD, e então evitar custos energéticos com a transferência desnecessárias de arquivos pode gerar uma economia energética considerável a esses DCs. Afinal, a transferência de dados pela rede gera custo energético por bit transferidos, estudos mostram que esse custo pode chegar a $2.7\mu\text{j}$ por bit em redes públicas (WAN) e $0.46\mu\text{j}$ em redes privadas (LAN) (BALIGA, 2011).

Assim, no presente trabalho, o objetivo é estimar previamente o consumo energético da execução de um VD, em Joules, em diferentes servidores por um determinado tempo. A partir

dos custos energéticos estimados pelo modelo, concluir qual entre esses diferentes servidores tem a melhor eficiência energética para instanciar e executar o VD requisitado em um determinado cenário. Entre esses servidores pode estar também a estação de trabalho local do usuário, caso opte por executar localmente. Logo, com esse modelo é possível responder, por exemplo, quando é melhor ao usuário utilizar um VD remoto ou sua estação de trabalho convencional para executar uma determinada aplicação. É possível também decidir se compensa, no ponto de vista energético, realizar a migração do VD entre diferentes DCs em determinadas situações.

4.1 Detalhamento do Modelo

A Equação 4.1 é a equação geral do modelo proposto nesta dissertação. Esse modelo estima o consumo energético de instanciar e executar um VD em um determinado servidor. O servidor pode ser uma máquina em um *datacenter* ou um computador privado, quando o usuário resolver instanciar o VD em sua própria máquina.

Esse modelo se faz necessário uma vez que o crescimento da utilização desta tecnologia de VDs pode trazer impacto direto no custo energético dos DCs. No cenário atual, as estimativas de custos energéticos propostas, se resumem a modelos que estimam o custo energético de softwares ou armazenamento como forma de serviço.

Empresas e organizações também podem se beneficiar deste modelo, uma vez que possuam diferentes DCs que possam receber os VDs dos funcionários ou utilizem de serviço de desktops virtuais contratado em ambiente de nuvem. Utilizando esse modelo, considerando que a qualidade de serviço dos VDs pode ser atingida todos os DCs disponíveis, o consumo energético pode ser diminuído. Com isso, as empresas, além de criarem melhorias ambientais, conseguem uma economia financeira.

Assim, nessa equação 4.1, são considerados custos energéticos que compõem todo o ciclo de um VD, desde sua migração até o servidor em que será executado, até o cliente de acesso utilizado pelo usuário. A somatória desses custos energéticos em um determinado tempo de execução, resulta no C_{eet} (Consumo energético estimado total) de um VD em Joules.

$$C_{eet} = C_{transf} + C_{exec} + C_{trans} + C_{cliente} \quad (4.1)$$

Dos quatro custos energéticos da equação 4.1, três deles são multiplicados pelo tempo de execução (C_{exec} , C_{trans} e $C_{cliente}$). Esses três custos energéticos precisam ser multiplicadas pelo tempo pois permanecem sendo acrescentados durante o tempo de execução no servidor, assim

para alguns custos energéticos são utilizados valores médios na estimativa.

C_{transf} é o único custo energético que é adicionado somente quando a migração dos arquivos acontece, logo não precisa ser multiplicado pelo tempo de execução mas sim pelo número de transferências executadas. O único tempo utilizado no custo C_{transf} é o tempo da transferência, que é utilizado para estimar o custo energético no servidor que envia os arquivos (servidor fonte). Vale notar que o resultado final da Equação 4.1 é apresentado em Joules, logo todos os custos energéticos devem ser utilizados como Joules quando necessário. A figura 4.1 ilustra os custos energéticos do modelo proposto.

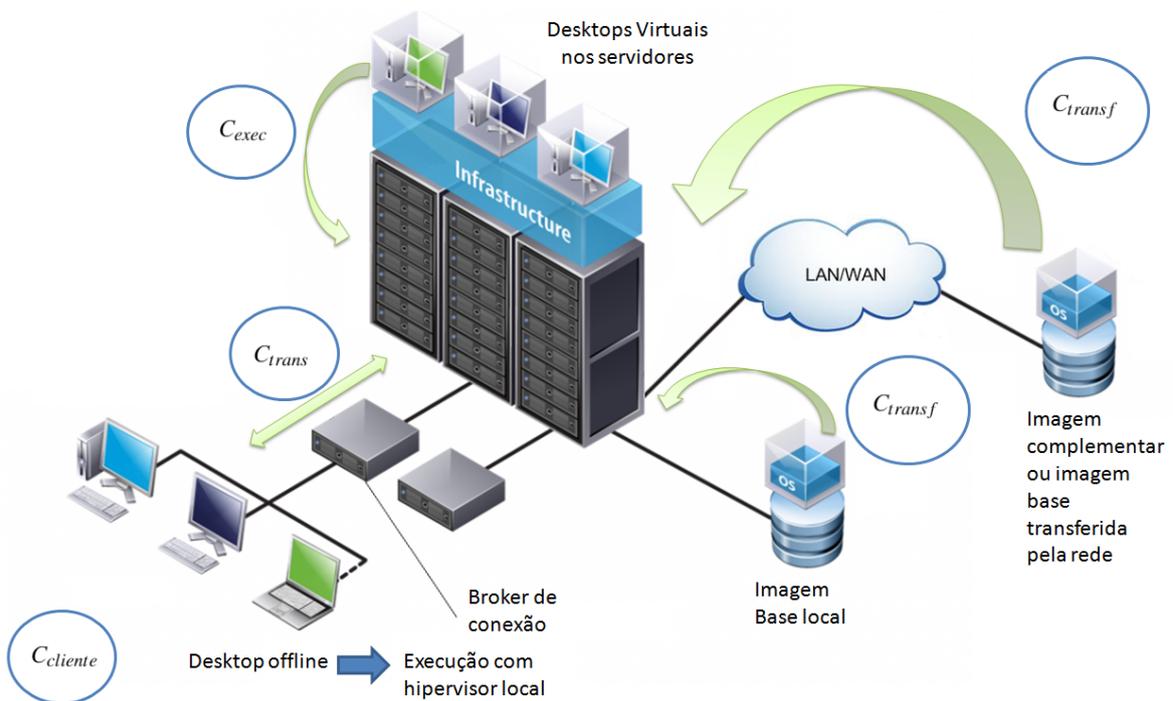


Figura 4.1: Exemplificação dos custos do modelo estimativo, adaptado de VMware Inc.

O primeiro fator, C_{transf} , representa o custo energético da migração de um VD entre servidores. Para isso, é considerado o custo energético de transferir os arquivos referentes ao VD entre o último servidor em que este foi instanciado, caso exista, e o servidor destino em que será instanciado desta vez. Os arquivos transferidos correspondem a todos os arquivos referentes ao VD que não existam no servidor em que este será instanciado. Esses arquivos podem ser compostos desde a imagem base, que contém o sistema operacional, softwares e outros arquivos padrões utilizados pelos VD, até os *snapshots* correspondentes ao VD específico do usuário, que contém seus arquivos e softwares próprios. Note que C_{transf} pode ser igual a zero, caso não seja necessário transferir arquivos para a execução do VD.

A composição do custo C_{transf} é detalhada nas equações 4.2 e 4.3. Na Equação 4.2, percebe-se que a quantidade de bits a serem transferidos (N_{bits}) é multiplicada pelo custo

energético médio de transmissão por bit (C_{Bit}) na rede. Em seguida, essa multiplicação é somada ao custo energético médio de execução do servidor fonte (C_{Fonte}), servidor que envia os dados durante o tempo da migração. O custo energético C_{Fonte} é detalhado na Equação 4.3. Nela, observa-se que o custo energético médio do servidor que envia os arquivos (C_{Serv}) é dividido pelo número de usuários ativos no servidor fonte (N_{Usur}), ou seja, o custo energético desse servidor é compartilhado pelo número de usuários que utilizam esse servidor no momento da migração. Este custo é então multiplicado pelo tempo de migração destes arquivos (T_{Transf}) até o novo servidor hospedeiro deste VD, como a migração normalmente leva minutos e não horas essa multiplicação normalmente será por um número menor que um. O custo do servidor que recebe os arquivos é considerado, de forma implícita, no fator C_{exec} , uma vez que a execução do VD será neste novo servidor.

$$C_{transf} = (N_{bits} * C_{bit}) + C_{fonte} \quad (4.2)$$

$$C_{Fonte} = (C_{Serv}/N_{Usur}) * T_{Transf} \quad (4.3)$$

Já o segundo componente energético da Equação 4.1 é o C_{exec} , que é o custo energético que a execução do *desktop* virtual e suas aplicações vão trazer ao DC em que for instanciado. O custo energético médio do servidor que irá instanciar o VD (C_{serv}), apontado na equação 4.4, depende muito das aplicações que já estão em execução neste servidor, as aplicações que vão executar no VD, do *hardware* que compõe o servidor e do número de instancias compartilhando o mesmo recurso. Por isso, durante os experimentos de validação que envolvem tempo de execução, são utilizados valores médios de consumo energético nesta variável.

Para estimar este custo energético do VD no *datacenter* (DC) durante sua execução, é necessário que o custo energético C_{serv} seja multiplicado pelo tempo de execução do VD e pelo *PUE* (Acrônimo em inglês para *Power usage effectiveness*) deste determinado DC, como apresentado na equação 4.4. Esta multiplicação é necessária para incluir os custos de equipamentos que não são de computação, como iluminação e refrigeração. O PUE é uma métrica calculada pela razão entre a quantidade de energia consumida por todos os equipamentos do DC, incluindo equipamentos de refrigeração e iluminação, somados aos equipamentos de computação e divididos pelo consumo energético dos equipamentos de computação. O valor PUE ideal é de 1, correspondendo a um DC onde toda a energia fornecida pela rede elétrica é dedicada para equipamentos de TI, não havendo gasto de energia com refrigeração e nem com distribuição de energia. Um PUE menor que 1 seria possível com a geração própria de energia, mas atu-

almente este modelo é comercialmente impraticável de se implementar (VERDI, 2010) (KHOSRAVI; GARG; BUYYA, 2013) (BELADY, 2007). Este custo energético (C_{serv}) é compartilhado entre o número de usuários conectados. Portanto, conforme definido na Equação 4.4, esse fator é dividido pelo número máximo de instâncias ativas no servidor (N_{inst}).

O número máximo de instâncias é utilizado por técnicas como a consolidação de recursos, que tem como objetivo utilizar a capacidade máxima do *hardware*. Para isso, a técnica de consolidação aloca o máximo de instâncias possíveis em um mesmo *hardware*, evitando que vários servidores permaneçam ativos com poucos VDs instanciados. Note que a memória não é considerada nesse modelo, porém é considerado que não existem *overhead* da mesma e que ela está sendo utilizada dentro dos limites do *hardware*. Portanto nesta equação, que tem como objetivo eficiência energética, será utilizado na estimativa o número máximo de instâncias possíveis ao servidor, que pode ser calculado conforme a equação 4.5. Nessa Equação 4.5 o número de processadores (N_{proc}) é multiplicado pelo número de núcleos ($N_{nucleos}$) e por fim multiplicado pelo número de instâncias por núcleo ($N_{instNucleo}$), apresentando o conceito de *overcommit*, lembrando que os limites de memória são respeitados. Esse conceito permite que a soma da memória dos VDs seja maior que quantidade de memória física disponível, utilizando compartilhamento de tempo para a execução dos VDs.

$$C_{exec} = C_{serv} * Pue * T_{exec} / N_{inst} \quad (4.4)$$

$$N_{inst} = N_{proc} * N_{nucleos} * N_{instNucleo} \quad (4.5)$$

O Custo de transmissão (C_{trans}) é o custo energético médio gerado pela transmissão de bits referentes à interação com o usuário, incluindo tela e eventos de comando de teclado e mouse. Esse custo é gerado após o VD ser instanciado e começar a transmissão referentes as atualizações de tela com o cliente em que o usuário está acessando seu VD. Esse custo energético reflete cada mudança que um comando executado pelo usuário gere na tela de seu cliente. Essa mudança pode ser completa, que modifica a tela toda, como uma ação de minimizar uma janela por exemplo, ou uma mudança mínima como um comando de digitar uma letra de uma palavra em um editor de texto. O protocolo utilizado pelo provedor de serviço também pode influenciar nos bits transmitidos e na forma em que eles são transmitidos. Portanto, a quantidade de quadros que devem ser transmitidos é alterada no decorrer do tempo, para mais e para menos, logo será utilizado nesta equação o valor médio de transmissão de bits.

Esse custo C_{trans} também é multiplicado pelo tempo, já que enquanto a execução do VD

continuar esse custo energético continuará existindo e sendo acrescentado no modelo. Esse fator é detalhado na equação 4.6, em que o número de bits médio por segundo (N_{bitS}) é multiplicado pelo número de segundos em uma hora (3600), multiplicado, por sua vez, pelo custo energético para transmissão de cada bit e, por fim, multiplicado pelo tempo de execução (T) em horas.

$$C_{trans} = N_{bitS} * 3600 * C_{Bit} * T \quad (4.6)$$

Por fim, há o custo energético médio do cliente ($C_{cliente}$) utilizado pelo usuário para o acesso ao VD. Esse custo também pode ser o custo do próprio servidor quando a execução das aplicações for em seu próprio *desktop* local. Esse valor é determinado pelo consumo energético médio do *hardware* utilizado para acesso ao VD ou execução local ($C_{aparelho}$), que pode ser desde um *desktop* convencional até um cliente simples como um *tablet*. Esse valor também precisa ser multiplicado pelo tempo de utilização, como podemos ver na equação 4.7.

$$C_{cliente} = C_{aparelho} * T \quad (4.7)$$

4.2 Aplicação do modelo para estimativas energéticas

Nesta seção o modelo proposto é utilizado para estimar o custo energético, em Joules, de um VD quando executado em um determinado DC, aqui chamado D1. Note que, para realizar essas estimativas exemplificativas, são considerados alguns valores típicos para as variáveis que compõem os fatores do modelo, esses valores utilizados são apresentados na tabela 4.1. Esses valores utilizados não são valores fixos e são alterados conforme configurações do DC ou servidor que instancia o VD. Alguns dos valores utilizados para essa estimativa são baseados em trabalhos apresentados no capítulo 3.

Nessas estimativas iniciais algumas variáveis, que compõem os fatores da equação 4.1, são modificadas para apresentar o impacto dessas variáveis no C_{et} do VD durante sua execução. Nas figuras 4.2, 4.3, 4.4 e 4.5, são apresentadas respectivamente variações na quantidade de arquivos a serem migrados, custo energético médio do servidor que executa o VD, taxa média de bits transmitidos e o custo médio do cliente de acesso. O tempo de execução foi alterado de 1h à 40h. Esse valor representa, normalmente, a jornada de trabalho semanal de um funcionário por exemplo. Vale notar que esses valores foram definidos para esses experimentos e não são fixos, assim, podem ser alterados conforme necessidade de quem utiliza o modelo estimativo.

Na figura 4.2 são apresentados os resultados dos C_{et} quando utilizadas diferentes quanti-

Configuração	Valor utilizado na estimativa
Quantidade de bits na migração	5 GB
Custo médio por bit transmitido	2.7 μ J
Consumo médio do Serv. que envia os arquivos	187,5 W/h
Número de instancias no servidor fonte	32
Tempo de migração aprox.	0,167 horas (10 minutos)
Consumo Médio do Serv. de execução	300W/h
Núm. de inst. no Serv. de execução	120
PUE	1,2
Taxa Trans. média	2,101 Mbits/s
Consumo médio do aparelho de acesso	90W/h

Tabela 4.1: Configurações típicas utilizadas para o VD e para D1 na estimativa, valores com base em (BALIGA, 2011), (FALVO, 2014),(ALBIERO, 2011) e (LIU, 2013)

dades de arquivos transferidos durante a instanciação ou migração do VD. Essa variável (quantidade de arquivos) compõe o fator C_{transf} da equação 4.1 e neste primeiro grupo de estimativas o valor inicial, que era de 5GB, é aumentado até 150GB. Assim, nesta figura 4.2, é possível verificar que o C_{eet} do VD aumenta conforme é aumentada a quantidade de arquivos que devem ser transferidos. A diferença, após 40h de execução, é de aproximadamente 3362,96KJ entre o D1 - 5GB e D1 - 150GB.

Já para gerar os resultados apresentados na segunda figura 4.3, a quantidade de arquivos transferidos em uma possível migração volta ao valor inicial (5GB) e o custo energético alterado dessa vez é o consumo médio do servidor de execução, variável que compõe o fator C_{exec} da equação 4.1. Este consumo médio do servidor, inicialmente igual a 100W, foi aumentado até 500W. Note que durante as estimativas esse valor é convertido de Watts para Joules. Nesta figura 4.3 é possível verificar que o custo do VD cresce mais devagar do que na figura 4.2, resultando em uma diferença máxima de 576KJ após 40h de execução. Esse tempo foi escolhido por representar, usualmente, uma jornada semanal de trabalho de um usuário. Essa diferença menor acontece porque existem, além do consumo energético do servidor, outros custos energéticos que compõem o fator C_{exec} do VD mas não estão sendo alterados nesta estimativa, como PUE e número de instâncias em execução. Tais valores impactam diretamente no resultado deste fator

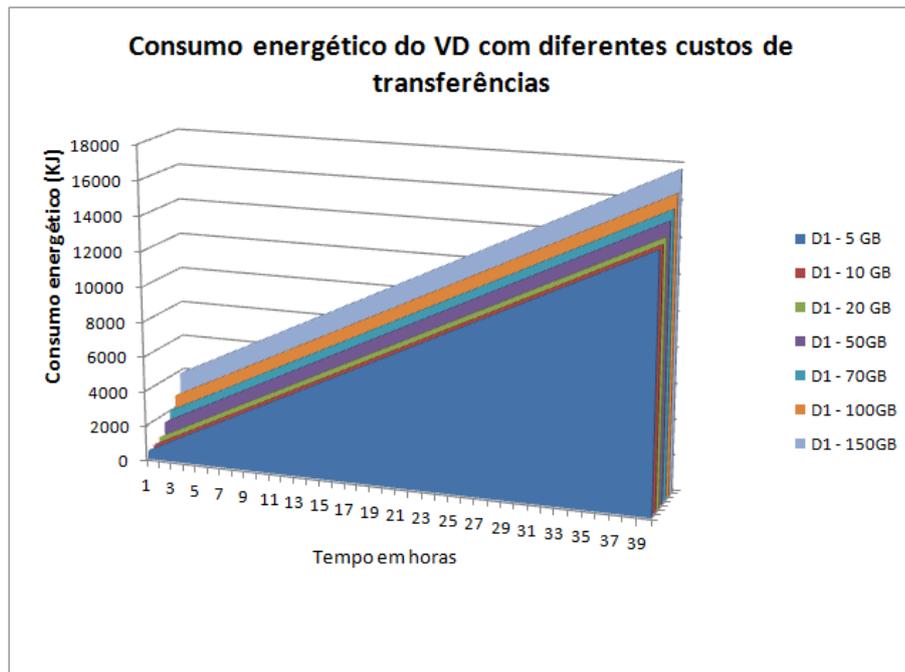


Figura 4.2: Exemplificação C_{et} do VD com diferentes custos de transferências

de consumo energético C_{exec} .

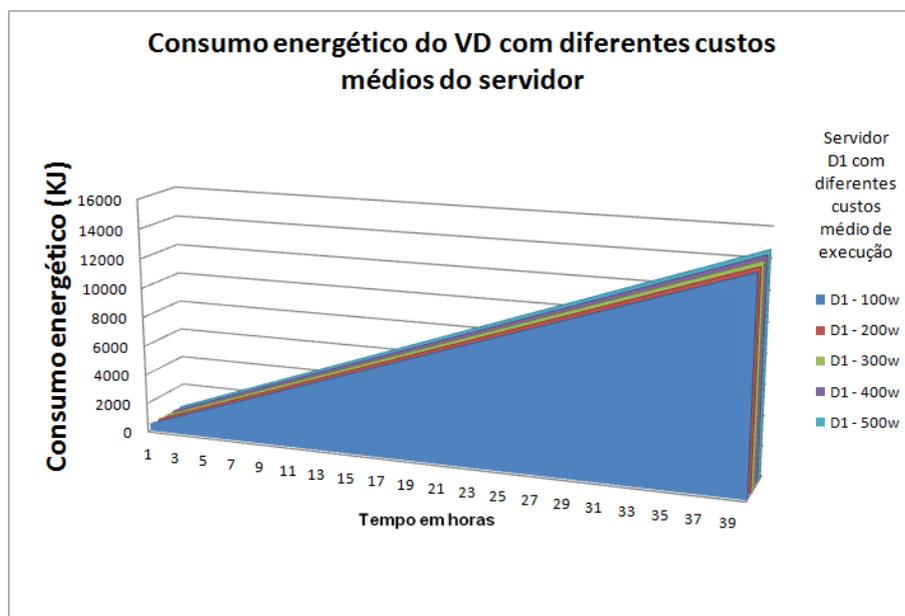


Figura 4.3: Exemplificação C_{et} do VD com diferentes consumos médios do servidor

Por sua vez, na figura 4.4, o único valor modificado para gerar esses resultados é a taxa de transmissão média de bits, as outras variáveis, alteradas previamente, voltam aos valores iniciais como apresentado na tabela 4.1. A taxa de transmissão média compõe o fator C_{trans} da equação 4.1 e é alterada conforme aplicação que o usuário está executando no VD. Neste exemplo, a taxa média de transmissão aumentou de 1 Mbits/seg até 8Mbits/s. Para tais valores toma-se

como base o trabalho de (FALVO, 2014), em que foram apresentadas algumas aplicações e suas taxas de transmissão médias. Foi possível verificar o quanto esse valor impacta no custo final do VD, onde a diferença máxima após 40h de execução foi de aproximadamente 2853,80KJ.

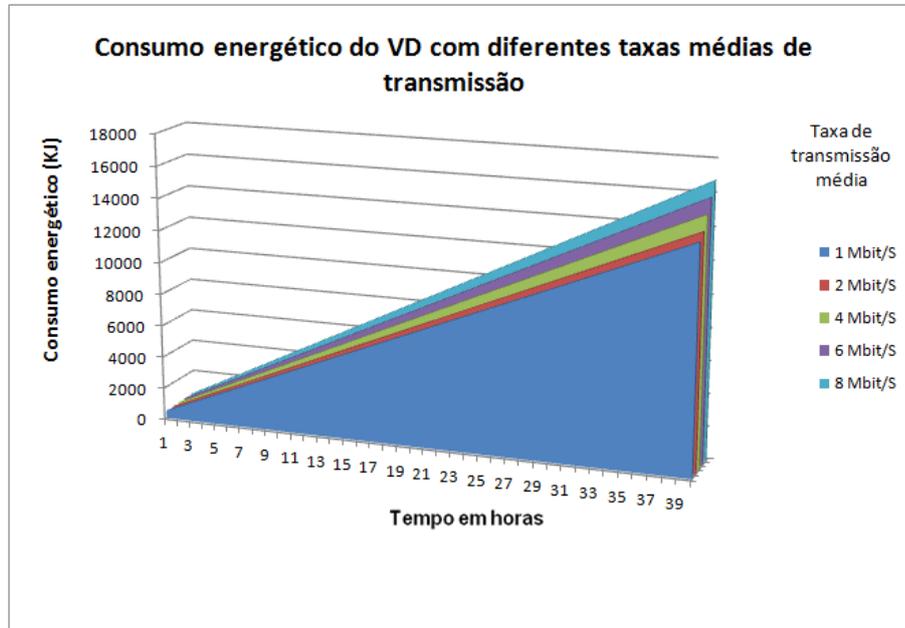


Figura 4.4: Exemplificação C_{eet} do VD com diferentes taxas médias de transmissão

Por fim, para gerar os resultados apresentados na figura 4.5, o único valor sendo modificado desta vez é o consumo médio do aparelho que o usuário utiliza para acessar seu VD. Essa variável, quando multiplicada pelo tempo, resulta no fator $C_{cliente}$ da equação 4.1 e impacta diretamente no C_{eet} desse VD. É possível verificar a diferença considerável entre um cliente que consome em média 60W para um que consome em média 110W. Esta diferença é de aproximadamente 7200KJ, após 40h.

4.3 Considerações finais do capítulo

Nesse capítulo foi apresentado o modelo energético para estimativa do consumo de um VD. Esse modelo é necessário para estimativas de consumo energético durante utilização de VD, uma vez que provedores de serviço não disponibilizam muitas informações sobre consumo energético, logo estimativa energética é uma alternativa. Alguns modelos são apresentados no capítulo 3, porém nenhum estima o consumo da execução de um VD. Com esse modelo é possível estimar previamente o consumo energético, em Joules, do VD do usuário em um determinado tempo. A partir dessa estimativa o usuário pode decidir qual o melhor servidor, baseada em eficiência energética, que pode instanciar este VD. Utilizando do modelo pode-se

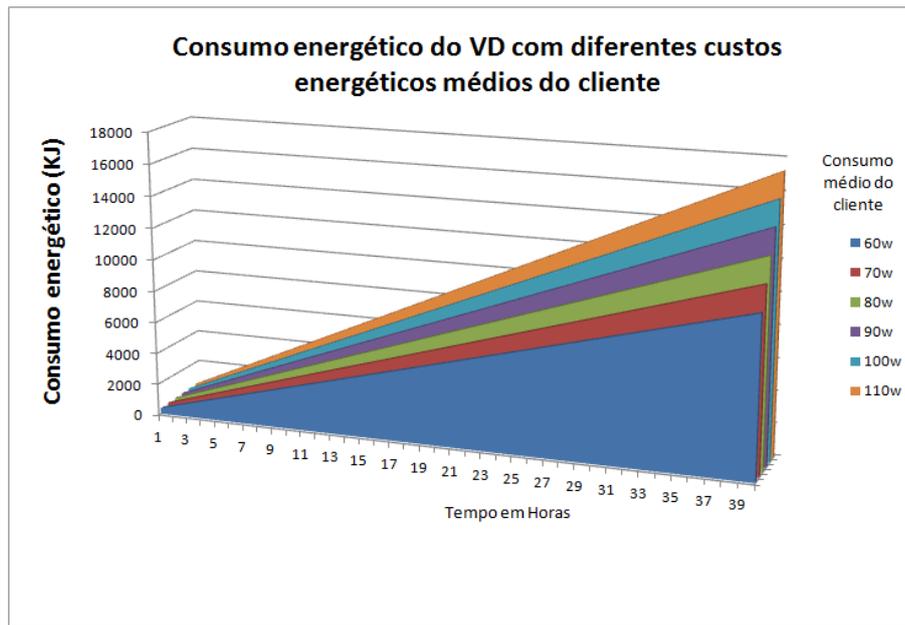


Figura 4.5: Exemplicação C_{et} do VD com diferentes custos energético médio do cliente

evitar, por exemplo, o consumo energético de uma migração desnecessária de um VD entre servidores ou a alocação do VD em um servidor que consuma mais energia.

Capítulo 5

APLICANDO O MODELO DE CONSUMO ENERGÉTICO

Esse capítulo descreve alguns cenários hipotéticos relacionados à execução dos *desktops* virtuais (VD) e seus respectivos consumos energéticos. O modelo proposto no capítulo anterior será aplicado a cada um desses cenários, com o objetivo de identificar as configurações com menor custo energético para decidir onde instanciar um novo VD.

Durante os experimentos, foram considerados possíveis *datacenters* (DC) onde o VD do usuário pode ser instanciado e também inclui, em alguns cenários, a própria estação de trabalho do usuário, caso seja melhor executar suas tarefas localmente ao invés de utilizar um servidor remoto. Os DCs utilizados nesses experimentos possuem diferentes configurações e diferentes *PUE*. Nota-se que, como o objetivo do modelo estimativo é o consumo energético total e não momentâneo, os custos energéticos utilizados foram convertidos de *Watts* para *Joules*. Também é importante notar que o modelo estima o consumo energético total; logo, quanto menor o C_{eet} , mais eficiente, no ponto de vista energético, a execução do VD neste servidor é.

Foi considerado também que os DCs, propostos nos experimentos, possuem os requisitos para entrega do serviço de *desktops* virtuais com qualidade, ou seja, não foram considerados nesses experimentos latência de rede elevada ou limitação de memória, por exemplo. O tempo de execução, dos experimentos, variou inicialmente de 1h à 25h, representando o tempo médio de 5 horas diárias, em uma jornada semanal de 5 dias úteis de trabalho. Depois, em uma continuação de cada um dos cenário, essa variação no tempo de execução foi expandida para 40h, que representa, normalmente, a jornada de trabalho semanal de um funcionário por exemplo. Vale notar que esses valores foram definidos para esses experimentos e não são fixos, assim, podem ser alterados conforme necessidade de quem utiliza o modelo estimativo.

5.1 Cenário 1: Desktop físico vs. Desktop Virtual

No primeiro cenário hipotético, serão comparadas duas situações. Na primeira situação, considera-se que o usuário utiliza um *desktop* convencional em sua máquina local. Nesse caso, todas as aplicações são executadas localmente. Na segunda situação, considera-se que o mesmo usuário irá utilizar o serviço de VD, que é oferecido por um DC remoto. Ou seja, a utilização irá gerar tráfego na Internet e considera-se que o tempo de utilização do *desktop* irá durar algumas horas. O objetivo é utilizar o modelo para indicar qual situação será mais econômica em termos do consumo de energia.

Para a realização dos experimentos, é necessário considerar alguns valores típicos. Por exemplo, foi considerado que o *desktop* local possui um consumo médio de 110Wh, que seria o consumo típico de um *laptop* executando com processador e memória em 100% de utilização (ALBIERO, 2011). Na segunda situação, considerou-se dois DCs arbitrários que podem instanciar o VD, chamados D1 e D2. Para esses DCs, neste cenário, as configurações são definidas conforme apresentado tabela 5.1.

	D1	D2
Eficiência Energetica (PUE)	1.8	1.8
Consumo energético médio p/ maq	300W/h	300W/h
Processadores / maq	2	2
Núcleos P/ processador	10	6
Overcommit	6	2
Instancia / maq	120	24

Tabela 5.1: Configurações dos DCs para o cenário 1

Nesse cenário são considerados dois tipos de aplicações diferentes que são executadas no VD, um reprodutor de vídeo (MP, acrônimo inglês para *media player*) e um navegador web (IE, acrônimo inglês para *Internet Explorer*). Por isso, o consumo de rede para a entrega do VD (isto é, transmissão da tela e eventos de teclado e mouse) terá dois valores médios para a estimativa de C_{trans} , conforme apresentado na tabela 5.2. O custo energético por bit transmitido pela rede de acesso à Internet é em média 2.7. Tais valores são utilizados conforme apresentado em estudos (FALVO, 2014) (BALIGA, 2011).

O cliente de acesso, por sua vez, consome 90Wh para a recepção e exibição da tela e interação com o usuário e o VD tem 5GB de dados. Vale notar que os DCs não são os mais eficientes em termos de consumo de energia e possuem um PUE de 1.8. Assim, se os experimentos demonstrarem ganhos em favor do VD, tais ganhos poderão ser ainda maiores se utilizado um

Aplicação em execução	Navegador Web	Reprodutor de Vídeo
Taxa Trans. aproximado	2,101 Mbits/s	5,802 Mbits/s

Tabela 5.2: Taxa de transmissão média por aplicação (FALVO, 2014)

servidor mais eficiente.

A execução local não tem custos energéticos como C_{exec} , C_{trans} e C_{transf} , porém aumenta o consumo do $C_{cliente}$ por ser o custo energético para execução local. O custo energético de $C_{cliente}$ não é dividido entre usuários, então o consumo energético é integral do usuário da máquina. Logo, aplicando o modelo proposto, a execução remota do VD será mais eficiente, no ponto de vista energético, que a execução local se uma das condições das equações 5.1 ou 5.2 forem atendidas.

$$C_{eetLocal} > C_{eetD1} \quad (5.1)$$

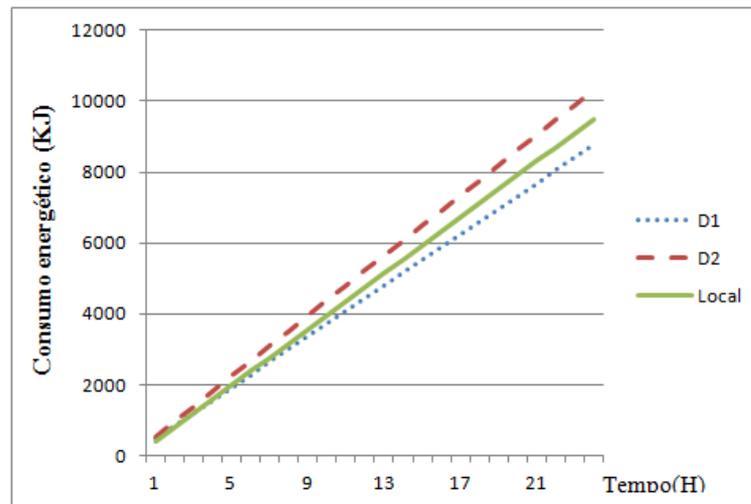
ou

$$C_{eetLocal} > C_{eetD2} \quad (5.2)$$

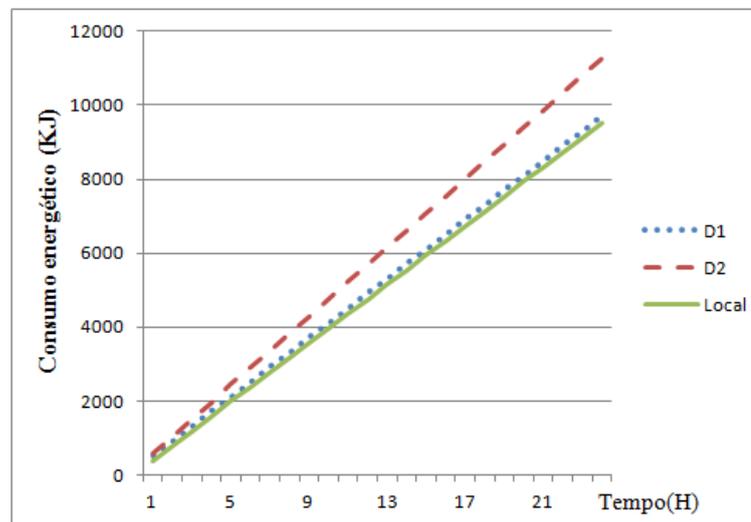
As figuras 5.1 a) e 5.1 b) apresentam os resultados dos experimentos realizados no Cenário 1. Na primeira situação desse cenário, o único custo energético para execução local é o custo do próprio cliente. Já o custo energético da execução remota, representado na segunda situação, é calculado conforme a equação 1, com o fator tempo variando de 1 a 25 horas de execução.

A partir das estimativas energéticas calculadas e apresentadas na figura 5.1 a), conclui-se que a execução de um VD só foi vantajosa em comparação a um *desktop* local após, no mínimo, 6 horas de uso, executando a aplicação Internet Explorer e no centro de dados D1, que possui melhor eficiência energética. O principal fator foi a capacidade dos servidores de cada *datacenter*. Destaca-se que eles oferecem capacidades de entrega de serviço diferenciados, mas consomem a mesma quantidade de energia.

Quando considerado no experimento a execução da aplicação Media player, apresentado na figura 5.1 b), para reprodução de vídeo, a execução local é, após 1h de execução, mais eficiente do que a execução remota, inclusive para D1. Esse resultado é reflexo do ao alto custo energético gerado pela transmissão de quadro de telas para execução do reprodutor de vídeo.



(a) Comparação com VD executando a aplicação Internet Explorer



(b) Comparação com VD executando a aplicação Media player

Figura 5.1: C_{eet} comparativo Desktop físico vs. Desktop Virtual em diferentes Datacenters

5.1.1 Aplicando diferentes taxas de transmissão ao cenário 1

Neste cenário 1, as aplicações escolhidas para realizar as estimativas energéticas foram Internet Explorer e Media Player. Os valores escolhidos para taxa de transmissão média dessas aplicações foram baseados no estudo de (FALVO, 2014). Nesta subseção, o modelo foi utilizado para estimar o C_{eet} do mesmo VD nos mesmos DCs, apresentados neste cenário 1, porém com diferentes taxas de transmissão média, que foram alteradas entre 1, 3 e 8Mbits para esse VD. Vale notar que nenhuma outra configuração foi alterada e, permanecendo como descritas no cenário 1.

Os resultados dessas novas estimativas são apresentados na figura 5.2, em que é possível observar que, mesmo alterando a taxa de transmissão, C_{eet} em D1 é sempre menor que em

D2, devido a diferença de capacidade dos servidores. D1 mostra-se eficiente inclusive quando comparado em diferentes faixas de taxa de transmissão média, por exemplo, execução do VD em D1 utilizando transmissão média de 8Mbits/s é quase tão eficiente quanto a execução em D2 utilizando taxa de transmissão média de 1Mbits/s. É possível observar também que quanto maior a taxa de transmissão média, menor a eficiência energética de utilizar um VD. Com taxa de transmissão média de 1Mbits/s, por exemplo, o C_{eet} de D2 foi quase tão eficiente quanto o C_{eet} da execução local, mesmo D2 sendo um DC com baixa capacidade. Já com taxa de transmissão média de 8Mbits/s para execução do VD em D1 ou D2, a execução local é mais eficiente que a execução em qualquer um desses DCs.

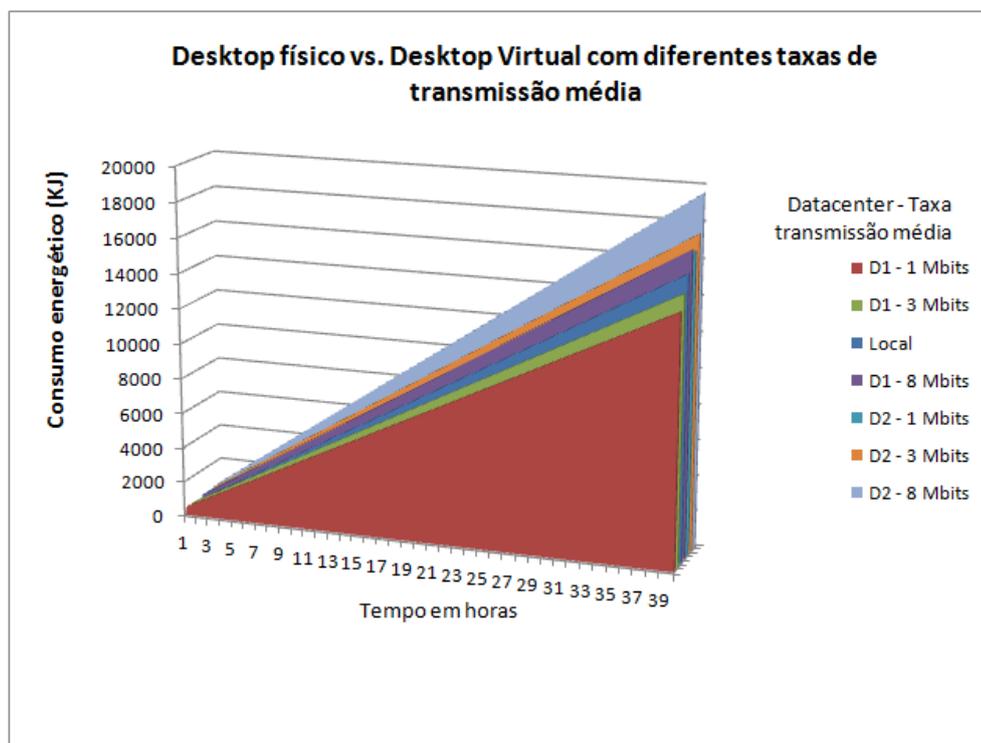


Figura 5.2: C_{eet} do VD executando em D1 e D2 com diferentes taxas de transmissão média

5.2 Cenário 2: Execução remota com diferentes custos de transmissão

Nesse segundo cenário é considerada uma situação hipotética de uma empresa de grande porte que possui mais de um DC para executar os VDs de seus funcionários. O cenário considera um funcionário que tem seu VD instanciado em D1, um DC próximo ao seu local de trabalho de origem, e fará uma viagem a outro local de trabalho distante. O local de destino fica próximo a outros dois DCs (D2 e D3) onde o VD poderá ser executado utilizando rede local. Caso a execução seja mantida em D1, o tráfego passará por rede de longa distância.

O objetivo desse cenário é descobrir se compensa ao usuário, no ponto de vista energético, continuar com seu VD instanciado em D1 com conexão WAN ou realizar a migração para um DC mais próximo de sua nova localização (D2 ou D3), onde a conexão voltará a ser LAN. Assim, alguns fatores da equação 4.1 são determinantes neste cenário, como o custo energético da transmissão de bits pela rede (C_{trans}) e o custo energético da transferência (C_{transf}). Logo, a execução em D2 ou D3 só será mais econômica, no ponto de vista energético, se:

$$C_{trans}D1 + C_{exec}D1 > C_{transf}D2 + C_{trans}D2 + C_{exec}D2 \quad (5.3)$$

ou

$$C_{trans}D1 + C_{exec}D1 > C_{transf}D3 + C_{trans}D3 + C_{exec}D3 \quad (5.4)$$

$C_{cliente}$ não é importante neste cenário, pois é igual (mesmo cliente de acesso) para os dois lados das equações 5.3 e 5.4. Já C_{transf} não é considerado para D1 pois é o servidor atual do VD. Entretanto, neste experimento, foram considerados diferentes valores para o custo médio por bit transferido (C_{Bit}), um dos fatores da equação 4.6 que calcula a estimativa de C_{trans} . É utilizado, neste cenário, um valor médio de $2.7\mu j$ por bit transmitido para redes WAN e $0.46\mu j$ para redes LAN, valores baseados em (BALIGA, 2011).

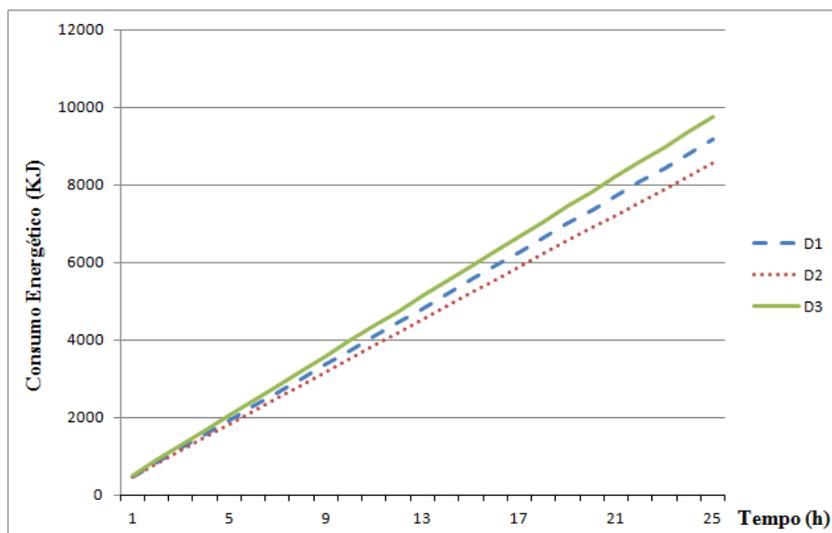
O cenário conta com três opções de DCs onde o VD do usuário poderá ser executado. A Tabela 5.3 apresenta o resumo das características de cada um deles. Note que foi considerado, para fins de experimento, uma diferença no PUE dos DCs D2 e D3. D2 possui PUE igual 1.2, muito eficiente, e D3 um PUE de 2.2, pouco eficiente. D1, DC atual do VD, possui um PUE razoável de 1.8.

	D1	D2	D3
Eficiencia Energetica (PUE)	1.8	1.2	2.2
Consumo energético médio p/ maq	300W	300W	350W
Processadores / maq	2	2	2
Núcleos p/ processador	10	10	6
Overcommit	6	6	4
Instancia p/ maq	120	120	48
Tipo de conexão	WAN	LAN	LAN
Quantidade de arquivos para migração do VD	0GB (DC atual)	5 GB	5GB

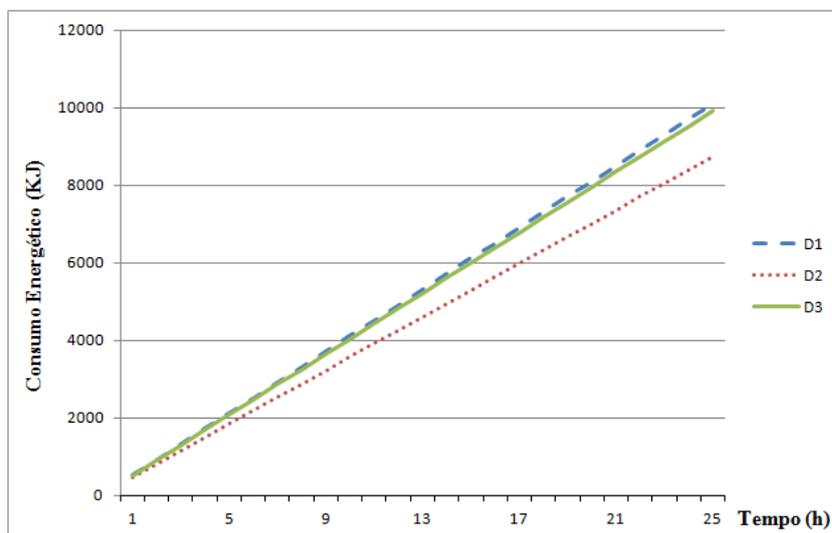
Tabela 5.3: Configurações dos DCs para o cenário 2

Neste cenário, também foram utilizadas duas diferentes aplicações executando neste VD.

Um reprodutor de vídeo (MP, acrônimo inglês para *media player*) e um navegador web (IE, acrônimo inglês para *Internet Explorer*). Essas aplicações foram escolhidas devido a diferença entre suas taxas de transmissão de quadros e comandos de teclado e mouse, que podem ser observadas na tabela 5.2 (FALVO, 2014). O cliente de acesso utilizado pelo usuário, para se conectar ao seu VD, consome em média 90W/h para transmissão de quadro de tela e comandos. Para cada experimento realizado neste cenário, o VD tem seu custo energético estimado executando em cada um dos DCs da tabela 5.3 e para cada aplicação da tabela 5.2.



(a) Comparação com VD executando a aplicação IE



(b) Comparação com VD executando a aplicação MP

Figura 5.3: C_{et} do VD, por aplicação, executando em diferentes Datacenters com diferentes custos por bit e PUE

As figuras 5.3 a) e 5.3 b) apresentam os resultados desse cenário 2. O experimento foi executado com o tempo incrementado de 1 a 25 horas de execução em que foi possível concluir a importância do PUE e de C_{Bit} (custo de transmissão por bit) da equação 4.6.

Na figura 5.3 a), em que é considerado um VD executando a aplicação IE, é possível observar que o C_{eet} em D3 é sempre maior que C_{eet} em D1, logo não é viável migrar o VD de D1 para D3. Isso é resultado do custo energético gerado pela migração (C_{transf}) somado ao custo da execução (C_{exec}) em um DC com poucas instâncias por máquina e um PUE elevado como D3. Assim, o custo de transferência (C_{transf}) do VD até D3 não foi compensado durante o tempo pelo menor custo de transmissão (C_{trans}) de D3 em relação a D1. Já a migração deste VD de D1 para D2 é viável a partir de 3 horas de execução, uma vez que o custo energético de transferência (C_{transf}) para D2 é compensado devido ao menor C_{exec} e menor C_{trans} em D2 do que em D1.

Já na figura 5.3 b), em que foi considerado o VD executando a aplicação MP, é possível observar que, devido a alta taxa de transmissão de quadro de tela e comandos gerada por essa aplicação, é viável a migração do VD de D1 para D2 e também para D3. Essa alta taxa de transmissão de dados gera, no decorrer do tempo, um custo energético elevado, pois cada bit transmitido terá um custo maior se transmitido pela rede WAN do que pela rede LAN. Assim, com menor C_{trans} em D3 do que em D1, a migração é viável mesmo com C_{exec} maior em D3. Por fim, por D2 possuir um menor custo energético de execução e também um menor custo energético de transmissão, se mostra muito mais eficiente no ponto de vista energético do que D1 ao executar o VD.

5.2.1 Aplicando diferentes custos por bit transmitido e PUE ao cenário 2

Neste cenário 2, os diferentes custos energéticos de transmissão gerados entre o cliente e o servidor em que o VD foi executado, compensaram, em alguns casos, os custos energéticos da migração do VD de um DC para outro. Porém, os custos energéticos médios por bits transmitidos sofrem alterações em seus valores conforme altera-se o tipo de conexão de rede utilizada para transmissão dos bits. O PUE também terá valores diferentes quando utilizado outros DCs, diferentes dos apontados no cenário 2. Assim, o C_{eet} do VD também é impactado.

Por isso, o que será apresentado nessa subseção é uma variação do que foi apresentado no cenário 2. No primeiro experimento, chamado EXP2.1, os custos energéticos médios por bit transmitido para os *datacenters* D2 e D3 são alterados, deixando de utilizar os valores apresentados no cenário 2 e agora utilizam os valores médios de $0.2\mu\text{j}$, $1.5\mu\text{j}$ e $3\mu\text{j}$ por bit. Nenhum outro custo energético foi alterado em EXP2.1 e portanto seguem as mesmas configurações apresentadas no cenário 2, inclusive o DC D1 que não sofreu alteração.

Já no segundo experimento, chamado EXP2.2, o custo energético médio por bit transmitido volta a seu valor inicial, conforme valores apresentados no cenário 2, na tabela 5.3. Em EXP2.2

o valor alterado, em relação ao cenário 2, foi o PUE dos DCs D2 e D3, que utilizaram neste novo experimento PUEs iguais a 1, 2 e 3. Nenhum outro valor foi alterado neste experimento, D1 continua com PUE igual a 1.8. Em ambos os novos experimentos (EXP2.1 e EXP2.2), a aplicação escolhida para execução no VD foi o navegador web IE por possuir uma taxa de transmissão média intermediária, com valor aproximado de 2,1 Mbits/s (FALVO, 2014).

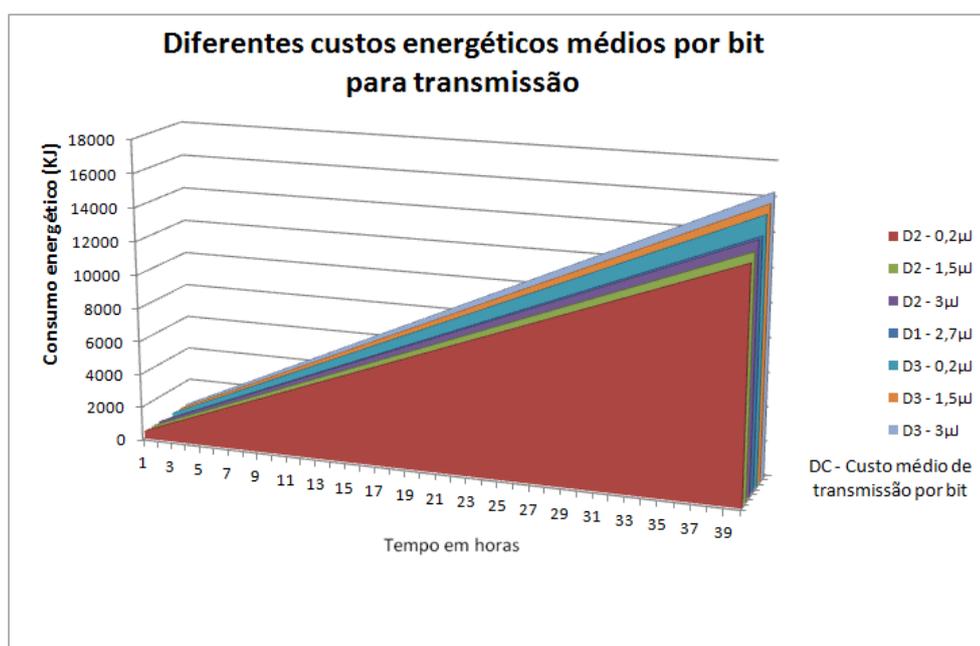


Figura 5.4: C_{eet} do VD executando em Datacenters com diferentes custos energéticos médios por bit para transmissão

Nos resultados de EXP2.1, apresentados na figura 5.4, é possível observar que D3 nunca é mais eficiente que D1, portanto a migração de D1 para D3 não compensa, do ponto de vista energético. Já em relação a D2, um DC eficiente, D1 só será mais eficiente quando o custo médio de transmissão por bit em D2 for maior que em D1. Isso acontece porque D2 tem um PUE menor e uma capacidade maior que D1 e D3, logo o C_{eet} de D2 é menor na maioria dos experimentos.

Já nos resultados de EXP2.2, apresentados na figura 5.5, é possível observar que quando o valor do PUE de D3 é igual a 1, um valor considerado PUE ideal, este DC tem praticamente o mesmo C_{eet} que D1, mesmo com a capacidade de D3 sendo menor que a de D1. Assim, a migração de D1 para D3 só compensa quando PUE de D3 igual a 1. Já D2, devido a sua capacidade, apresenta boa eficiência energética e, mesmo com um PUE elevado, o seu C_{eet} se manteve menor que os outros DCs.

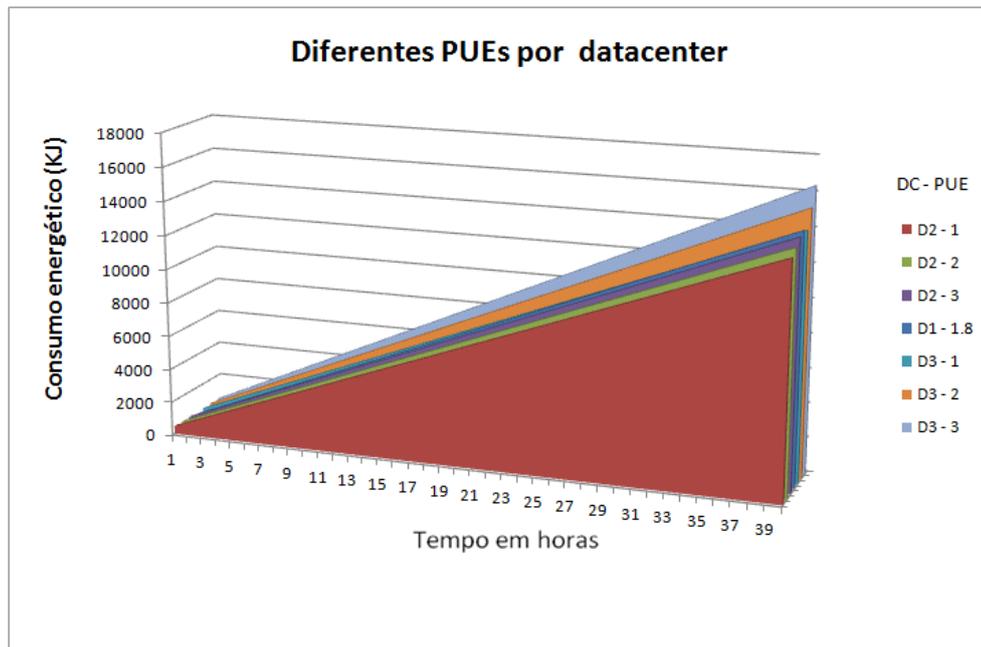


Figura 5.5: C_{eet} do VD executando em diferentes Datacenters com diferentes PUE

5.3 Cenário 3: Execução do desktop virtual no datacenter vs. execução no hypervisor local (Horizon)

Nesse terceiro cenário, considera-se um usuário que necessita desconectar-se temporariamente da rede, porém sem parar de executar seu VD. Para isso, é necessário utilizar um *hypervisor* que suporte tal funcionalidade, por exemplo o *Horizon*, *hypervisor* da *VMware* que tem como opção esta execução local. Neste tipo de execução local, o VD é migrado do DC para o *desktop* local, onde é instanciado temporariamente. Após a execução, logo que o usuário conecta-se à rede novamente, os arquivos referentes ao VD que foram atualizados durante a execução serão sincronizados com o servidor. O objetivo deste experimento é verificar qual tipo de execução é mais eficiente no ponto de vista energético, a execução remota tradicional de um VD ou sua execução local utilizando um *hypervisor* como o *Horizon*.

Para a realização dos experimentos, é necessário considerar alguns valores típicos. Por exemplo, são adotados dois diferentes custos energéticos médios para as máquinas locais, com *hypervisor*, durante o período temporário em que executarem o VD. O primeiro custo energético médio para execução local, em uma máquina chamada LOC1, é de 95W/h e o segundo valor, em outra máquina chamada LOC2, é de 110W/h. Esses valores de consumo energético médio são um pouco maiores do que o utilizado, durante os experimentos, para clientes de acesso. Vale notar que a execução local com *hypervisor*, em LOC1 ou LOC2, exige migração de arquivos e neste cenário, o valor utilizado é de 20GB. Esse valor de 20 GB foi escolhido justamente por

resultar em um custo energético não eficiente para a execução local com *hypervisor*. Imaginando empresas ou provedores com vários DCs, as imagens locais estão distribuídas e, então, esse valor de migração deve ser menor do que 20GB. Assim, mesmo em desvantagem, se os experimentos demonstrarem ganhos em favor da execução local, tais ganhos poderão ser ainda maiores se utilizado valor menor para migração de arquivos.

O custo energético estimado total (C_{eet}) das execuções locais, em LOC1 ou LOC2, são comparados ao C_{eet} da execução remota convencional de um VD, onde para isso são utilizados dois DCs arbitrários, chamados D1 e D2. A Tabela 5.4 contém o resumo das características de cada um deles. Nota-se que para a execução remota em D1 ou D2 é necessário um cliente de acesso e neste experimento, o consumo energético médio adotado para este cliente é de 90W/h.

	D1	D2
Eficiência Energetica (PUE)	1.2	2.2
Consumo energético médio p/ maq	300W	350W
Processadores / maq	2	2
Núcleos p/ processador	10	6
Overcommit	6	6
Instancias p/ máquina	120	72
Quantidade de arquivos para migração do VD	20GB	0GB (DC atual)

Tabela 5.4: Configurações dos DCs para o cenário 3

Neste cenário 3, também foram utilizadas duas diferentes aplicações executadas separadamente neste VD. Um reprodutor de vídeo (MP, acrônimo inglês para *media player*) e um navegador web (IE, acrônimo inglês para *Internet Explorer*). Suas taxas de transmissão são apresentadas na tabela 5.2 conforme valores apresentados em (FALVO, 2014). O custo médio por bit transmitido pela internet foi de $2.7\mu\text{j}$ (BALIGA, 2011). Esses valores foram utilizados para estimar o custo energético C_{trans} durante os experimentos deste cenário. Conforme citado no cenário anterior, essas aplicações foram escolhidas devido a diferença entre suas taxas de transmissão de quadros e comandos, gerando diferentes comportamento para o resultado dos experimentos. Neste cenário, o VD tem seu C_{eet} estimado para a sua execução em LOC1, LOC2, cada um dos DCs da tabela 5.4 e executando cada aplicação da tabela 5.2.

A execução local em LOC1 ou LOC2 tem vantagens como os custos energéticos C_{trans} e C_{exec} com valores iguais a zero, afinal esse tipo de execução não necessita transmitir quadros de tela e comandos pela rede, também não é necessário um servidor remoto para executar o VD. No entanto, a execução local aumenta custos energéticos de fatores como $C_{cliente}$, por

executar o VD em sua própria estação de trabalho, e C_{transf} , para migrar o VD até a máquina local. Assim, aplicações como o MP, que possuem alta taxa de transmissão de quadro de telas e comandos, normalmente são mais eficientes, no ponto de vista energético, durante a execução local. As equações 5.5 e 5.6 apresentam condições para que a execução local em LOC1 ou LOC2 seja mais eficiente, no ponto de vista energético, do que a execução remota. Note que para a execução em D2 não existe o custo energético C_{transf} , por ser o servidor atual do VD.

$$C_{transf}Local + C_{cliente}Local < C_{transf}D1 + C_{trans}D1 + C_{exec}D1 + C_{cliente}D1 \quad (5.5)$$

ou

$$C_{transf}Local + C_{cliente}Local < C_{trans}D2 + C_{exec}D2 + C_{cliente}D2 \quad (5.6)$$

A figura 5.6 apresenta os resultados da execução da aplicação IE no VD. Já na figura 5.7 são apresentados os resultados da execução da aplicação MP no VD. Note que D2 é considerado o DC atual em que o VD está instanciado, logo não precisa de migração para execução. Já D1 é um *datacenter* com maior eficiência energética que D2, porém para executar o VD considera-se que existe custo de migração (C_{transf}) deste até D1.

Na figura 5.6 a), executando o VD em LOC1 (custo energético médio de 95W), o C_{eet} da execução local foi menor do que o C_{eet} da execução remota em D1 após aproximadamente 6h de execução. Já a execução em D2, um DC com menos instâncias por máquina e maior PUE, apresenta nas primeiras horas menor C_{eet} do que D1 e LOC1, porém após 11h de execução o C_{eet} em D2 torna-se maior. Isso acontece devido à migração de arquivos que não é necessária em D2, porém, após esse tempo de execução, o alto custo C_{exec} em D2 faz com que a execução neste DC consuma mais energia que as outras opções.

Já na figura 5.6 b), executando o VD em LOC2 (custo energético médio de 110W), a execução remota do VD mostra-se mais eficiente, no ponto de vista energético. Esse resultado é possível devido ao custo energético médio elevado da máquina local LOC2, que é comparado ao C_{trans} razoável gerado pela aplicação IE durante a execução remota e pelo C_{exec} remoto amortizado pelo compartilhamento de recursos entre os usuários do servidor. Assim, como o custo energético da execução local é integral do usuário, a execução local consome sempre mais energia do que a execução remota neste experimento.

No segundo experimento deste cenário, apresentado na figura 5.7, executando a aplicação MP no VD, o C_{eet} da execução local é, na maioria das vezes, menor ou praticamente igual ao C_{eet} da execução remota. Isto é resultado do alto custo energético em C_{trans} gerado durante a execução da aplicação MP em um VD instanciado em D1 ou D2. Na figura 5.7 a) a execução do

VD em LOC1 é mais eficiente, no ponto de vista energético, do que em D1 e D2. Esse resultado é devido ao baixo custo energético médio da máquina local LOC1 (95W) e o custo energético C_{trans} igual a zero. Já na figura 5.7 b) o C_{eet} da execução local em LOC2, mesmo com um consumo energético maior (110W), é quase sempre menor ou igual o C_{eet} remoto. Somente após 21h de execução, o C_{eet} de LOC2 torna-se um pouco maior que o C_{eet} em D1, um DC eficiente, devido ao custo energético maior na máquina local (110W). Já comparando o C_{eet} da execução em LOC2 com o C_{eet} de D2, observa-se que após 17h ocorre a inversão e D2 passa a ter um C_{eet} maior que LOC2. Isso é resultado da execução da aplicação MP que gera um C_{trans} elevado para a execução remota, custo esse que não existe para o C_{eet} da execução local.

5.3.1 Aplicando diferentes quantidades de dados ao cenário 3

Neste cenário 3 a quantidade de dados em uma possível migração do VD foi estabelecida em 20GB, mas este valor não é um valor fixo e irá variar entre diferentes VDs. Para apresentar o impacto do C_{transf} no custo energético de um VD, executando em um determinado DC, foram realizadas modificações no tamanho do VD durante uma possível migração. As novas quantidade de dados do VD durante o experimento realizado nesta subseção são 5GB, 50GB e 100GB. Vale notar que nenhuma outra configuração foi modificada em relação ao cenário 3. A aplicação escolhida para execução no VD durante este novo experimento foi o navegador web IE, por, segundo estudos, possuir uma taxa de transmissão média intermediária, com valor aproximado de 2,1 Mbits/s (FALVO, 2014).

Os resultados desse experimento são apresentados nas figuras 5.8 e 5.9. A única diferença entre as figuras é a máquina local utilizada para comparação com a execução remota. Na figura 5.8 a máquina utilizada é LOC1 que consome em média 95W e na figura 5.9 a máquina utilizada é LOC2 que consome em média 110W. Nessas figuras, fica evidente que com o aumento da quantidade de arquivos, em uma possível migração do VD, aumenta-se também o C_{eet} do VD. É possível observar também que, com o aumento do tamanho do VD, mais tempo levará para que o C_{eet} de D1 seja menor que o C_{eet} de D2. Para o tamanho de 50GB, por exemplo, D1 tem um C_{eet} inicial muito maior do que D2, porém após 40h de execução o valor dos C_{eet} é praticamente igual em D1 e D2. Isso acontece porque o C_{Transf} de D2 é zero, por ser o servidor atual, mas o C_{exec} de D1 é menor que o C_{exec} de D2. Assim, D2 acaba sendo mais eficiente, no ponto de vista energético, pelas primeiras horas de execução do VD, porém no decorrer do tempo D1 será mais eficiente.

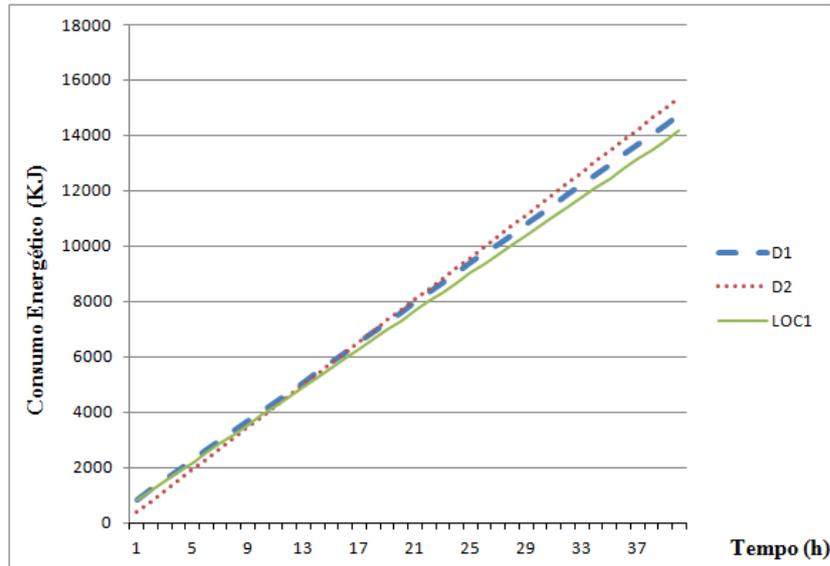
A execução local em LOC1, comparada com a execução remota na figura 5.8, é mais eficiente que a execução remota em D1 e D2 na maioria das comparações. D2 tem um C_{eet} menor

pelas primeiras horas de execução, por não necessitar de migração, porém no decorrer do tempo de execução C_{eet} em LOC1 será menor que em D2. Por exemplo, é possível observar na figura 5.8 que, para quantidade de arquivos igual a 50GB, após 27h de execução o C_{eet} de LOC1 torna-se menor que em D2. Já quando comparado a D1, LOC1 tem sempre um C_{eet} menor. Isso é resultado do custo médio de execução igual a 95W da máquina LOC1 e C_{trans} não existir para execução local.

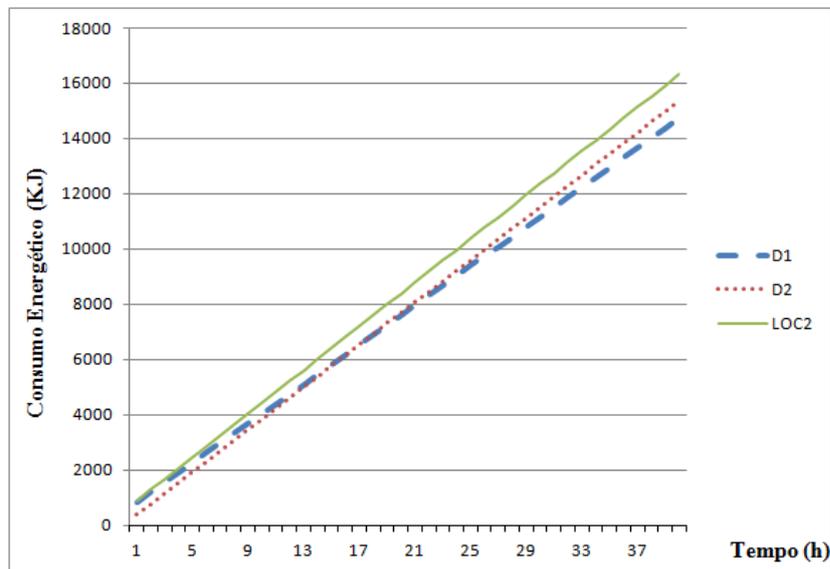
Já a execução local em LOC2, comparada com a execução remota na figura 5.9, é sempre menos eficiente do que a execução remota no experimento desta subseção. Isso é resultado do custo energético médio elevado na máquina local LOC2 (110W) e a taxa média de transmissão de dados pequena da execução da aplicação IE no VD. Soma-se a isso o fato de que o C_{exec} da execução remota nos servidores é amortizado pelos usuários, enquanto que local o custo é integral do usuário. Além de que o aumento do custo em C_{transf} não favorece a execução local em LOC2, pois esse também necessita realizar a migração do VD.

5.4 Considerações finais do capítulo

Neste capítulo o modelo proposto foi aplicado a cenários de execução utilizados normalmente por usuários e servidores. Nesses experimentos, realizados neste capítulo 5, foram utilizadas informações como consumos energéticos médios dos servidores, número de instâncias máximas por servidores, consumo energético médio da transmissão de quadros e comandos pela rede, quantidade média de bits transmitidos durante a execução de algumas aplicações, além de outras informações, com base no que foi apresentado, no capítulo 3, em (BALIGA, 2011), (ALBIERO, 2011), (LIU, 2013) e (FALVO, 2014). Alguns desses custos energéticos e informações utilizadas tiveram ainda seus valores expandidos, durante os cenários, para apresentar o impacto desses fatores energéticos no resultado final do C_{eet} do VD. Vale notar que os cenários apresentados não possuem restrições de *hardware* ou *software*.

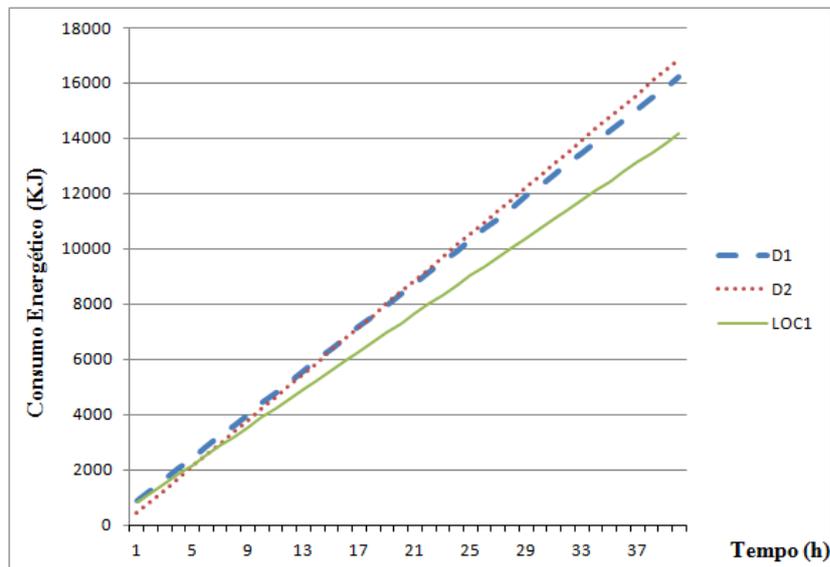


(a) Máquina local LOC1 com custo médio de execução em 95w

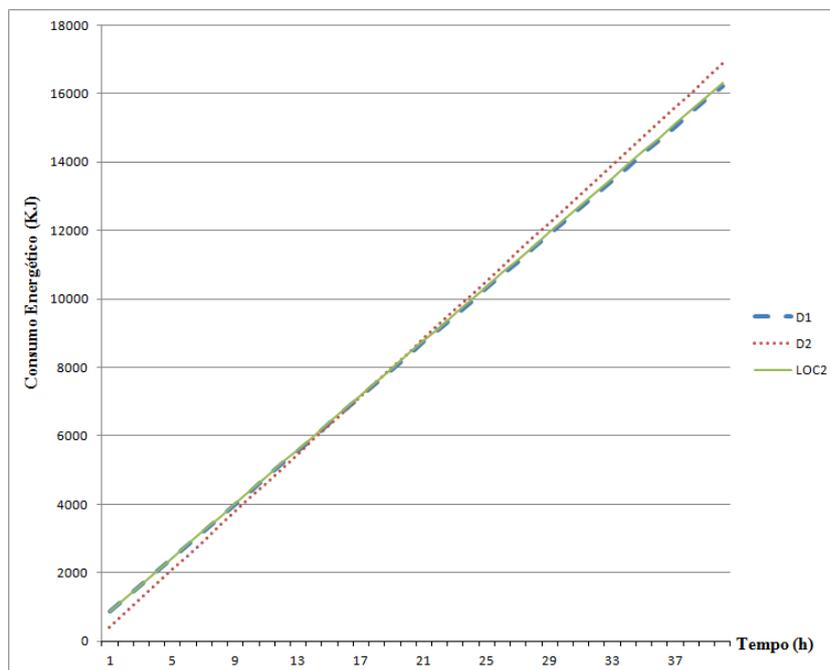


(b) Máquina local LOC2 com custo médio de execução em 110w

Figura 5.6: Comparativo C_{et} do VD executando na máquina local c/ hypervisor x DC remoto, executando aplicação IE



(a) Máquina local LOC1 com custo médio de execução em 95w



(b) Máquina local LOC2 com custo médio de execução em 110w

Figura 5.7: Comparativo C_{et} do VD executando na máquina local c/ hypervisor x DC remoto, executando aplicação MP

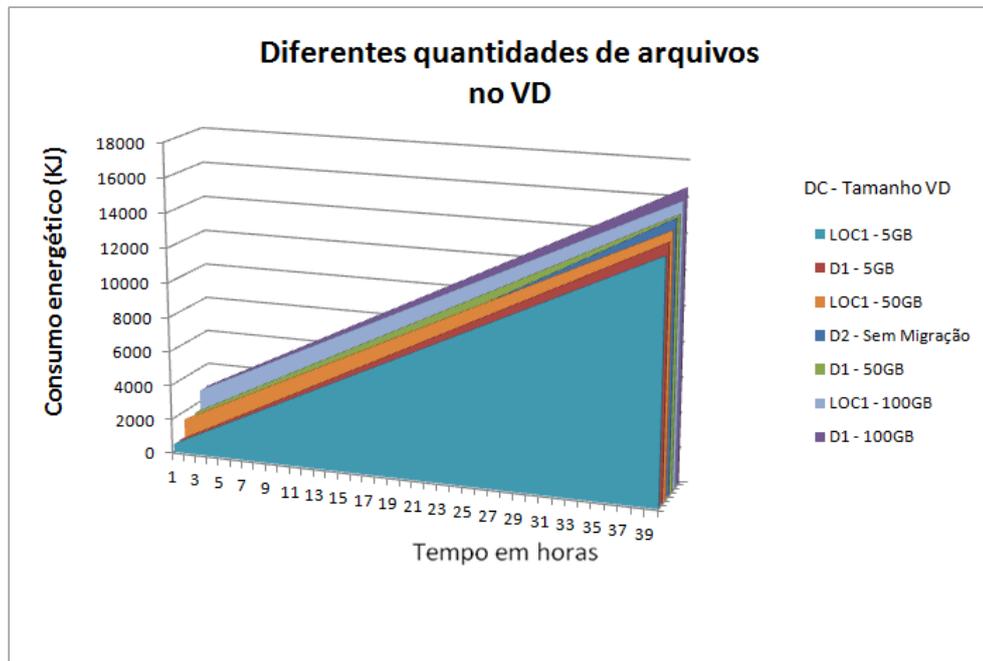


Figura 5.8: C_{et} do VD com diferentes quantidades de arquivos, Execução remota x execução na máquina local c/ hypervisor LOC1

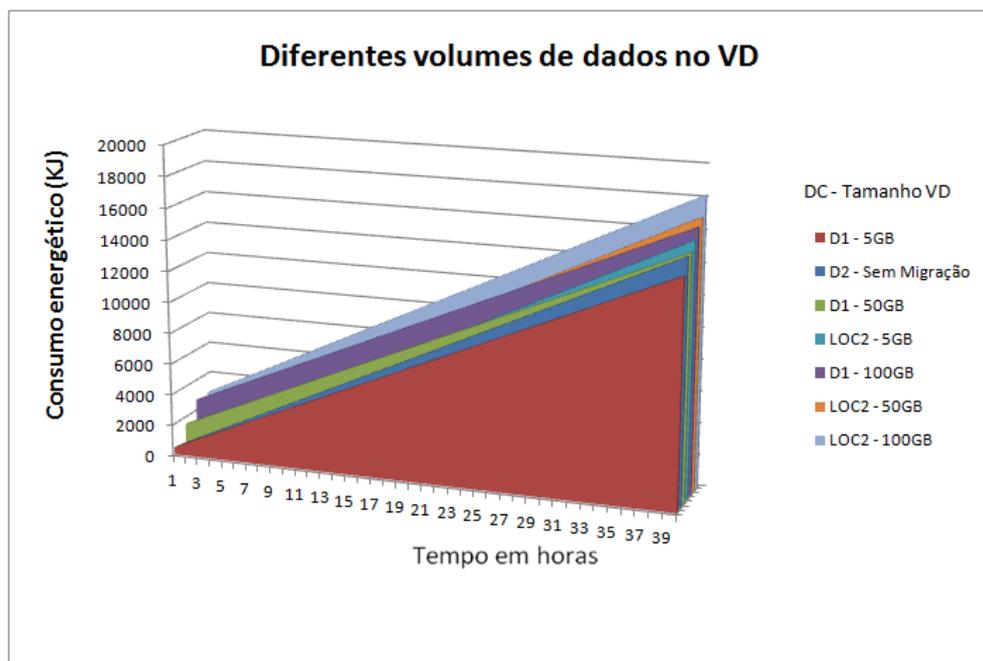


Figura 5.9: C_{et} do VD com diferentes quantidades de arquivos, Execução remota x execução na máquina local c/ hypervisor LOC2

Capítulo 6

CONCLUSÕES

Nesta dissertação foi desenvolvido um modelo que tem como objetivo estimar o consumo energético total da execução de um VD quando instanciado em um determinado servidor, local ou remoto (DC), durante um determinado tempo de execução. Isso quer dizer que o modelo proposto não é dependente de uma determinada arquitetura, nem de uma determinada configuração de servidor ou cenário de execução específico, o que tornou seu desenvolvimento ainda mais complexo.

Ao aplicar esse modelo estimativo aos cenários propostos, no capítulo 5, foi possível concluir que para aplicações com taxa de transmissão, de quadros e comandos, baixa ou regular é melhor instanciar um VD do que executar um *desktop* local. Esse cenário pode ser observado no primeiro experimento do capítulo 5. Isso acontece porque aplicações com alta taxa de transmissão média, por exemplo um reproduzidor de vídeo (MP), podem fazer com que o custo energético C_{trans} seja muito elevado. C_{trans} é resultado do número de bits transmitidos, multiplicado pelo custo de transmissão por bit e o tempo, logo, quanto mais bits transmitidos maior será o C_{trans} . Assim, como apresentado na figura 5.1 b), o C_{eet} da execução remota pode ser, no decorrer do tempo, maior do que a execução local, até mesmo para DCs eficientes como D1 neste cenário um.

Durante os experimentos, foi possível concluir também que a melhor eficiência energética obtida foi resultado de instanciar um VD em um DC com acesso à rede local (LAN), em que o consumo energético de transmissão por bit é menor (BALIGA, 2011). Essa situação pode ser observada no cenário de número 2, apresentado no capítulo 5, em que o custo energético médio por bit transmitido na rede LAN é tão pequeno que compensou, inclusive, o custo energético da migração do VD entre os DCs. Para aplicações com alta taxa de transmissão como o MP, o pequeno C_{trans} da rede local (LAN) levou apenas 5h para compensar a migração do VD. Neste caso a migração compensa até para o DC D3, um servidor com menor poder computacional

e maior PUE do que D1, como pode ser observado na figura 5.3 b). O impacto do fator PUE também foi observado na continuação do cenário 2. Foi possível concluir que quando o PUE do DC é pequeno, pode compensar inclusive o custo energético elevado de servidores mais antigos e com baixa capacidade computacional, como pode ser observado para D3 com PUE igual a 1 na figura 5.5.

O *Horizon* da *VMware*, utilizado como *hypervisor* durante o cenário 3 do capítulo 5, tem eficiência energética satisfatória e como pode ser observado foi melhor que a execução remota em alguns experimentos. Vale notar que seu consumo energético será menos eficiente do que a execução remota convencional, somente quando a máquina local não for eficiente ou para aplicações com pequena taxa de transmissão média de quadros e comandos, como o IE apresentado na figura 5.6. Para a execução da aplicação MP, com alta taxa de transmissão média de quadros, a execução local em LOC1 e LOC2 é mais ou tão eficiente quanto a execução remota, inclusive quando comparado a D1, um DC eficiente, como pode ser observado na figura 5.7. Esse resultado é possível devido ao custo C_{trans} igual a zero encontrado na execução local. Já na execução remota o custo C_{trans} gera um consumo energético elevado, agravado ainda mais por aplicações com taxa de transmissão média elevada, como exemplo o MP.

O tamanho da transferência do VD, durante uma possível migração, foi avaliado com diferentes tamanhos durante a continuação do cenário 3. Nestes experimentos confirmou-se que quanto maior o tamanho do VD maior será o consumo energético para realizar sua migração entre DCs (C_{transf}). Assim, quanto maior o C_{transf} , maior será o tempo necessário para que o servidor eficiente do novo DC compense esse custo da migração. Isso quer dizer que, conforme aumenta-se o tamanho do VD, mais tempo será necessário para que o C_{exec} em D1, um servidor com mais capacidade que D2, compense o C_{transf} do VD até D1. Já o consumo energético em D2, o DC atual e que não precisa de migração neste cenário, tem menor C_{eet} do que LOC1, LOC2 e D1 nas primeiras horas de execução, porém no decorrer do tempo o C_{eet} em D2 torna-se maior que o C_{eet} das outras opções. O tempo em que D2 terá o C_{eet} menor é proporcional ao aumento do tamanho do VD, logo, quanto maior o VD maior será o tempo que D2 é mais eficiente que as outras opções.

Assim, conclui-se deste trabalho que a execução de um VD é, na maioria das vezes, mais eficiente, no ponto de vista energético, do que executar um *desktop* em uma máquina local. Isso sem contar as vantagens apresentadas nos primeiros capítulos deste trabalho como mobilidade, facilidade no gerenciamento, disponibilidade e utilização máxima dos recursos. Porém, conclui-se também que pequenas alterações em configurações dos DCs ou custos energéticos dos VDs trazem impacto direto ao C_{eet} destes VDs. Como exemplo cita-se o resultado apresentado na

figura 4.5, em que a diferença no consumo energético médio do cliente resulta em diferentes C_{et} do VD. Logo, utilizar o modelo proposto para estimar previamente o consumo energético total do VD, pode ser uma boa alternativa para evitar consumos energéticos e migrações desnecessários deste VD.

6.1 **Trabalhos Futuros**

Como possíveis trabalhos futuros podem-se propor melhorias ao modelo que foi apresentado neste trabalho. Tais melhorias poderiam estimar em tempo real o número de instâncias ativas no servidor destino ou estimar em tempo real os diferentes custos energéticos utilizados na equação, como taxa de transmissão por aplicação e custo de transmissão por bit. Outro tipo de melhoria seria desenvolver mecanismos que utilizassem o modelo em tempo de escalonamento, para decidir qual servidor teria o menor custo energético para instanciar um determinado VD.

Seriam possíveis também melhorias referentes a outras vertentes de tomada de decisão, além de eficiência energética. Como exemplo, poderiam ser criados outros modelos que verificariam qual o servidor possui o tempo de inicialização menor ou custo financeiro mínimo para instanciar um determinado VD. Com esses outros modelos, criar um modelo geral que ranqueia esses custos de acordo com a preferência do usuário e assim tome a melhor decisão para onde executar o VD do mesmo.

REFERÊNCIAS

- ALBIERO, F. W. Monitoramento e avaliação do consumo energético em função do poder computacional. *Relatório técnico, Graduação em Informática, UFSM. Disponível em: <http://www-usr.inf.ufsm.br/~weber/monografia/tg-fabio.pdf>. Acesso em: junho de, 2011.*
- BALDUZZI, M. et al. A security analysis of amazon's elastic compute cloud service. In: ACM. *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. [S.l.], 2012. p. 1427–1434.
- BALIGA, J. et al. Green cloud computing: Balancing energy in processing, storage, and transport. *Proceedings of the IEEE, IEEE*, v. 99, n. 1, p. 149–167, 2011.
- BARUCHI, A. *Memory Dispatcher: uma contribuição para a gerência de recursos em ambientes virtualizados*. Tese (Doutorado) — Universidade de São Paulo, 2010.
- BELADY, C. L. In the data center, power and cooling costs more than the it equipment it supports. *Electronics cooling, FLOMERICS LIMITED*, v. 13, n. 1, p. 24, 2007.
- BELOGLAZOV, A.; BUYYA, R. Energy efficient resource management in virtualized cloud data centers. In: IEEE COMPUTER SOCIETY. *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. [S.l.], 2010. p. 826–831.
- BOSING, A.; KAUFMANN, E. R. Virtualização de servidores e desktops. *Unoesc & Ciência-ACET*, v. 3, n. 1, p. 47–64, 2012.
- BRANTNER, M. et al. Building a database on s3. In: ACM. *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. [S.l.], 2008. p. 251–264.
- CARISSIMI, A. Virtualização: da teoria a soluções. *Minicursos do Simpósio Brasileiro de Redes de Computadores–SBRC*, v. 2008, p. 173–207, 2008.
- CAVALCANTI, G.; OBELHEIRO, R. R.; KOSLOVSKI, G. P. Alocação ótima de recursos para infraestruturas virtuais confiáveis. *Proc. XXXII-SBRC*, 2014.
- CHAVES, S. A. de; URIARTE, R. B.; WESTPHALL, C. B. Implantando e monitorando uma nuvem privada. In: *VIII Workshop em Clouds, Grids e Aplicações*. [S.l.: s.n.], 2010.
- CHEN, Y.; WO, T.; LI, J. An efficient resource management system for on-line virtual cluster provision. In: IEEE. *Cloud Computing, 2009. CLOUD'09. IEEE International Conference on*. [S.l.], 2009. p. 72–79.
- COOK, G. *How clear is your cloud? catalysing an energy revolution*. Amsterdam, The Netherlands, abr. 2012. Disponível em: <<http://goo.gl/yd1FAS>>.

- COSTA, R. et al. Sobre a amplitude da elasticidade dos provedores atuais de computação na nuvem. *Anais do XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC2011)*. Sociedade Brasileira de Computação (SBC), 2011.
- COSTA, R. et al. Just in time clouds: Enabling highly-elastic public clouds over low scale amortized resources. *Federal University of Campina Grande/Federal University of Paraíba, Brazil, Tech. Rep. TR-3*, 2010.
- DE LA VIGLIOTTI, A. P.; MACEDO BATISTA, D. Energy-efficient virtual machines placement. In: IEEE. *Computer Networks and Distributed Systems (SBRC), 2014 Brazilian Symposium on*. [S.l.], 2014. p. 1–8.
- FALVO, M. R. An experimental evaluation of the use of virtual desktops. *Universidade Federal de São Carlos*, 2014.
- GAO, P. X. et al. It's not easy being green. In: *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. New York, NY, USA: ACM, 2012. (SIGCOMM '12), p. 211–222. ISBN 978-1-4503-1419-0.
- GARTNER. *Gartner Energy & Utilities IT Summit 2007: an invitation only event*. Dallas, EUA, 2007. Disponível em: <<http://www.gartner.com/it/page.jsp?id=502432>>.
- GONG, C. et al. The characteristics of cloud computing. In: IEEE. *Parallel Processing Workshops (ICPPW), 2010 39th International Conference on*. [S.l.], 2010. p. 275–279.
- HE, W. et al. Adaptive data reduction scheme for ssd-based host-side caches in vdi storage. *Advanced Science and Technology Letters - International Conference on Electrics, Electronics, and Computer Science (EEC 2013)*, 2013.
- HILLEY, D. Cloud computing: A taxonomy of platform and infrastructure-level offerings. *Georgia Institute of Technology, Tech. Rep*, 2009.
- INC., G. *green*. [S.l.], out. 2014. Disponível em: <<http://www.google.com/green/>>.
- JONES, S. T.; ARPACI-DUSSEAU, A. C.; ARPACI-DUSSEAU, R. H. Geiger: monitoring the buffer cache in a virtual machine environment. In: ACM. *ACM SIGOPS Operating Systems Review*. [S.l.], 2006. v. 40, p. 14–24.
- KANTER, J. P. *Understanding thin-client/server computing*. [S.l.]: Microsoft Press Redmond, Washington, 1998.
- KAROVIČ, M. D.-V. Cloud and virtualization in linux environment. *CER Comparative European Research 2015*, p. 130, 2015.
- KHOSRAVI, A.; GARG, S. K.; BUYYA, R. Energy and carbon-efficient placement of virtual machines in distributed cloud data centers. In: *Euro-Par 2013 Parallel Processing*. [S.l.]: Springer, 2013. p. 317–328.
- KIBE, S.; KOYAMA, T.; UEHARA, M. The evaluations of desktop as a service in an educational cloud. In: IEEE. *Network-Based Information Systems (NBIS), 2012 15th International Conference on*. [S.l.], 2012. p. 621–626.

- KOOMEY, J. *Growth in data center electricity use 2005 to 2010*. Oakland, CA: Analytics Press, 2011. 2010 p.
- KOURIL, J.; LAMBERTOVA, P. Performance analysis and comparison of virtualization protocols, rdp and pcoip. *Proc. of ICCOMP*, 2010.
- LIGUORI, A.; VAN HENSBERGEN, E. Experiences with content addressable storage and virtual disks. In: *Workshop on I/O Virtualization*. [S.l.: s.n.], 2008.
- LIU, H. et al. Performance and energy modeling for live migration of virtual machines. *Cluster computing*, Springer, v. 16, n. 2, p. 249–264, 2013.
- MA, F.; LIU, F.; LIU, Z. Live virtual machine migration based on improved pre-copy approach. In: IEEE. *Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on*. [S.l.], 2010. p. 230–233.
- MAGALHÃES, D. V.; SOARES, J. M.; GOMES, D. G. Análise do impacto de migração de máquinas virtuais em ambiente computacional virtualizado. *Anais do XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 235–248, 2011.
- MARCHÃO, J. M. O. A. L. *Os serviços em cloud na ótica de utilização empresarial: um estudo de viabilidade*. Tese (Doutorado) — Escola Superior de Ciências Empresariais, 2012.
- PASTRAMA, A. Infrastructure as a service between the mantychore and gsn projects. *Telecommunication Systems Laboratory*, Set 2011.
- RAZAVI, K.; RAZOREA, L. M.; KIELMANN, T. Reducing vm startup time and storage costs by vm image content consolidation. In: SPRINGER. *Euro-Par 2013: Parallel Processing Workshops*. [S.l.], 2014. p. 75–84.
- RUMBLE, S. M. et al. It's time for low latency. In: USENIX ASSOCIATION. *Proceedings of the 13th USENIX conference on Hot topics in operating systems*. [S.l.], 2011. p. 11.
- SCHLOSSER, D. et al. Improving the qoe of citrix thin client users. In: IEEE. *Communications (ICC), 2010 IEEE International Conference on*. [S.l.], 2010. p. 1–6.
- SHU, S. et al. Prototyping efficient desktop-as-a-service for fpga based cloud computing architecture. In: IEEE. *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. [S.l.], 2012. p. 702–709.
- SILBERCHATZ, A.; GALVIN, P. *Sistemas operacionais*. (1a edição). Campus, Rio de Janeiro, 2001.
- SINGH, A. An introduction to virtualization. ISSN <http://www.kernelthread.com/publications/virtualization/>. URL <http://www.kernelthread.com/publications/virtualization/>, 2004.
- SMITH, J.; NAIR, R. *Virtual machines: Versatile platforms for systems and processes* (the morgan kaufmann series in computer architecture and design). Morgan Kaufmann Publishers Inc., 2005.
- SO, K. Cloud computing security issues and challenges. *International Journal of Computer Networks*, v. 3, n. 5, 2011.

SOUZA FILHO, J. d. Análise comparativa entre ambientes computacionais baseados nas tecnologias de processamento distribuído e de thin client. *Universidade regional de Blumenau - Centro de Ciências tecnológicas e Engenharia de telecomunicações*, Nov 2003.

SURKSUM, K. van. Release: VMware mirage 5.2. virtualization. info, 2014.

TANG, C. Fvd: A high-performance virtual machine image format for cloud. In: *USENIX Annual Technical Conference*. [S.l.: s.n.], 2011.

VERDI, F. L. et al. Novas arquiteturas de data center para cloud computing. *Minicursos do XXVIII SBRC*, p. 103–152, 2010.

WALDSPURGER, C. A. Memory resource management in vmware esx server. *ACM SIGOPS Operating Systems Review*, ACM, v. 36, n. SI, p. 181–194, 2002.

WANG, J.; LIANG, L. Survey of virtual desktop infrastructure system. ietf.org, Nov 2011. Disponível em: <<https://tools.ietf.org/html/draft-ma-appsawg-vdi-survey-00>>.

GLOSSÁRIO

$C_{cliente}$ – *Custo do cliente de acesso*

C_{eet} – *Consumo energético estimado total*

C_{exec} – *Custo energético da execução do desktop virtual*

C_{transf} – *Custo energético da migração do VD*

C_{trans} – *Custo da transmissão de quadros de tela e comandos*

DC – *Datacenter*

VDS – *Acrônimo em inglês para Virtual Desktop Service ou Serviço de desktops virtuais*

VD – *Desktop Virtual*

VM – *Máquina virtual*

VR – *Acrônimo em inglês para virtual resources, que podem ser máquinas virtuais ou desktops virtuais*

Capítulo 7

RELATÓRIO DA DEFESA DA DISSERTAÇÃO



UNIVERSIDADE FEDERAL DE SÃO CARLOS
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a defesa de dissertação de mestrado do candidato Pedro Wilson Bignatto Junior, realizada em 07/03/2016.

Prof. Dr. Hermes Senger
(UFSCar)

Prof. Dr. Hélio Crestana Guardia
(UFSCar)

Prof. Dr. Mário Dantas
(UFSC)

Certifico que a sessão de defesa foi realizada com a participação à distância do membro Prof. Dr. Mário Dantas e, depois das arguições e deliberações realizadas, o participante à distância está de acordo com o conteúdo do parecer da comissão examinadora redigido no relatório de defesa do aluno Pedro Wilson Bignatto Júnior.

Prof. Dr. Hermes Senger
Presidente da Comissão Examinadora
(UFSCar)



UNIVERSIDADE FEDERAL DE SÃO CARLOS
 Centro de Ciências Exatas e de Tecnologia
 Programa de Pós-Graduação em Ciência da Computação

Relatório de Defesa de Dissertação
 Candidato: Pedro Wilson Bignatto Júnior

Aos 07/03/2016, às 9 h, realizou-se na Universidade Federal de São Carlos, nas formas e termos do Regimento Interno do Programa de Pós-Graduação em Ciência da Computação, a defesa de dissertação de mestrado sob o título: *Alocação de Recursos ciente de energia para entrega de Desktops Virtuais em um Ambiente de Computação em Nuvem*, apresentada pelo candidato Pedro Wilson Bignatto Júnior. Ao final dos trabalhos, a banca examinadora reuniu-se em sessão reservada para o julgamento, tendo os membros chegado ao seguinte resultado:

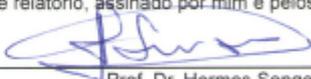
Participantes da Banca	Função	Instituição	Resultado
Prof. Dr. Hermes Senger	Presidente	(UFSCar)	<u>Aprovado</u>
Prof. Dr. Hélio Crestana Guardia	Titular	(UFSCar)	<u>Aprovado</u>
Prof. Dr. Mário Dantas	Titular	(UFSC)	<u>Aprovado</u>

Resultado Final: Aprovado

Parecer da Comissão Julgadora*:

A Comissão examinadora considerou que o trabalho satisfaz plenamente as exigências do programa de mestrado. O candidato fez boa apresentação e mostrou domínio do assunto tratado. A comissão solicita, no entanto, que o candidato efetue as modificações apontadas, ficando o orientador responsável pela verificação final.

Encerrada a sessão reservada, o presidente informou ao público presente o resultado. Nada mais havendo a tratar, a sessão foi encerrada e, para constar, eu, Maria Cristina Carreira Trevelin, representante do Programa de Pós-Graduação em Ciência da Computação, lavrei o presente relatório, assinado por mim e pelos membros da banca examinadora.


 Prof. Dr. Hermes Senger (presidente)


 Prof. Dr. Hélio Crestana Guardia

 Prof. Dr. Mário Dantas

 Representante do PPG: Maria Cristina Carreira Trevelin

Certifico que a defesa realizou-se com a participação à distância do membro Prof. Dr. Mário Dantas e, depois das arguições e deliberações realizadas, o participante à distância está de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.


 Prof. Dr. Hermes Senger (presidente)

() Não houve alteração no título da dissertação Houve. O novo título passa a ser:

Um modelo para estimativa do consumo de energia de desktops virtuais.

Observações:
 a) Se o candidato for reprovado por algum dos membros, o preenchimento do parecer é obrigatório.
 b) Para gozar dos direitos do título de Mestre em Ciência da Computação o candidato ainda precisa ter sua dissertação homologada pelo Conselho de Pós-Graduação da UFSCar.
 c) Nesta ata não são necessárias as assinaturas dos componentes da banca.
 d) Homologação da dissertação aprovada na XXª reunião da CPG de DD/MM/AAAA.