



Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

Implementação de consolidação de servidores utilizando software livre

Allysson Chagas Carapeços

CASCADEL
2011

ALLYSSON CHAGAS CARAPEÇOS

**IMPLEMENTAÇÃO DE CONSOLIDAÇÃO DE SERVIDORES
UTILIZANDO SOFTWARE LIVRE**

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientador: Prof. MSc. André Luiz Brun

CASCADEL
2011

ALLYSSON CHAGAS CARAPEÇOS

**IMPLEMENTAÇÃO DE CONSOLIDAÇÃO DE SERVIDORES
UTILIZANDO SOFTWARE LIVRE**

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em
Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel,
aprovada pela Comissão formada pelos professores:

Prof. MSc. André Luiz Brun (Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Cláudio Ernesto Pertille Ramos (Co-Orientador)
Analista de Sistema Sênior II,
ITAIPU BINACIONAL

Prof. MSc. Anibal Mantovani Diniz
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Dr. Marcio Seiji Oyamada
Colegiado de Ciência da Computação,
UNIOESTE

Cascavel, 04 de novembro de 2011

DEDICATÓRIA

Dedico este trabalho a todas as pessoas que participaram direta e indiretamente da minha formação acadêmica. Em especial dedico a Deus, por ter me proporcionado a vida, a Nossa Senhora Aparecida por atender meus pedidos, aos meus pais Airton Pereira Carapeços e Helena de Chagas Rodrigues, aos meus irmãos Deise Ane de Chagas Rodrigues e Elyzandro Carapeços, à minha namorada Tatiana Peiter, pela paciência, compreensão e apoio incondicional que recebi de cada um deles. Aos meus colegas de faculdade por todos os momentos e conquistas que tivemos, em especial ao amigo Odair Moreira de Souza.

“Seja você quem for, seja qual for a posição social que você tenha na vida, a mais alta ou a mais baixa, tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá. De alguma maneira você chega lá.”

Ayrton Senna

AGRADECIMENTOS

Agradeço a Deus e a Nossa Senhora Aparecida por terem proporcionado saúde para viver tudo que vivi, propiciando novos conhecimentos e oportunidades, fazendo novos colegas, levando amigos para vida toda.

Impossível deixar de agradecer meus pais, Airton Pereira Carapeços e Helena de Chagas Rodrigues, e meus irmãos, Deise Ane de Chagas Rodrigues e Elyzandro Carapeços, pelo carinho, amor, dedicação, palavras de apoio e tudo que fizeram em prol de minha educação, o maior bem que eles podem me deixar.

À minha amiga e namorada, Tatiana Peiter, a qual foi um dos presentes que conquistei durante a minha vida acadêmica, dividindo momentos inesquecíveis para mim, sendo sempre carinhosa, atenciosa e compreensiva, entendendo também cada um dos dias que estava cansando, devido horas, dias e semanas estudando e tentando solucionar problemas durante a faculdade.

Ao meus amigos e colegas que me apoiaram na realização deste trabalho e contribuíram para meu crescimento pessoal e profissional, em especial ao Odair Moreira de Souza e o Professor Carlos José Maria Olguín.

Tenho que agradecer a três pessoas que apostaram em minha dedicação e competência para realizar este trabalho, dois professores, Guilherme Galante e André Luiz Brun, e meu colega durante a realização do meu estágio na Itaipu Binacional, Cláudio Ernesto Pertille Ramos, que mesmo após meu desligamento da empresa, aceitou manter sua co-orientação.

Ao Colegiado de Ciência da Computação, muito obrigado pela viabilização de todos os equipamentos utilizados na montagem da infraestrutura deste trabalho.

Lista de Figuras

1.1	Norma de infraestrutura de telecomunicações para <i>datacenters</i> (adaptado de [1]).	2
2.1	Exemplo de empresa com <i>datacenter</i> consolidado.	13
2.2	Benefícios da implantação da <i>Green IT</i> (adaptado de [2]).	14
2.3	Termômetro verde da Greenpeace [3].	18
3.1	Arquitetura do <i>hypervisor</i> KVM (adaptado de [4]).	21
3.2	Técnica da emulação de <i>hardware</i> (adaptado de [5]).	22
3.3	Técnica da virtualização completa (adaptado de [6]).	23
3.4	Técnica da paravirtualização (adaptado de [6]).	24
3.5	Técnica da virtualização em nível de sistema (adaptado de [6]).	25
3.6	Modelo da arquitetura do VMware Server [7].	27
3.7	Modelo da arquitetura do VMware Server ESX [8].	28
3.8	Base da arquitetura KVM (adaptado de [9]).	31
4.1	Modelo da infraestrutura de simulação.	39
4.2	Interface da ferramenta para consolidação de servidores.	42
5.1	Geração de carga contínua nas máquinas virtuais.	47
5.2	Comportamento das máquinas físicas com carga contínua.	48
5.3	Geração de carga dinâmica nas máquinas virtuais.	49
5.4	Comportamento das máquinas físicas com carga dinâmica.	51
5.5	Migração de máquinas virtuais ociosas.	52
5.6	Migração de máquinas virtuais com carga simulada de 10 clientes.	53
5.7	Migração de máquinas virtuais com carga simulada de 50 clientes.	54

Lista de Tabelas

5.1	Tempo de migração das máquinas virtuais com carga contínua (em segundos). .	47
5.2	Tempo de migração das máquinas virtuais com carga dinâmica (em segundos).	50
5.3	Comparação entre migrações.	54

Lista de Abreviaturas e Siglas

TI	Tecnologia da Informação
IDC	International Data Corporation
ANSI	American National Standards Institute
TIA	Telecommunications Industry Association
EIA	Energy Information Administration
ASHRAE	American Society of Heating, Refrigerating and Air Conditioning Engineers
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
TCO	Total Cost of Ownership ou Custo Total de Posse
KVM	Kernel-based Virtual Machine
MIT	Massachusetts Institute of Technology
VMM	Monitor de Máquina Virtual
Intel VT	Intel Virtualization Technology
AMD-V	Advanced Micro Devices Virtualization
CIO	Chief Information Officer ou Diretor de Informática
GNU GPL	GNU General Public License ou GNU Licença Pública Geral
USB	Universal Serial Bus
VDI	Virtual Disk Image
iSCSI	Internet Small Computer System Interface
RDP	Microsoft Remote Desktop Protocol
IBM	International Business Machines
NUMA	Non-Uniform Memory Access
VPS	Virtual Private Servers
CPU	Central Processing Unit ou Unidade Central de Processamento
ROI	Return On Investment ou Retorno Sobre Investimento
SAN	Storage Area Network ou Área de Armazenamento em Rede
CO ₂	Dióxido de Carbono
RAM	Random Access Memory ou Memória de Acesso Aleatório
LCD	Display de Cristal Líquido
CRT	Tubo de Raios Catódicos
SSH	Secure Shell
NFS	Network File System
NRPE	Nagios Remote Plugin Executor
XML	Extensible Markup Language
DHCP	Dynamic Host Configuration Protocol
NAT	Network Address Translation
IP	Internet Protocol

Sumário

Lista de Figuras	vii
Lista de Tabelas	viii
Lista de Abreviaturas e Siglas	ix
Sumário	x
Resumo	xiii
1 Introdução	1
1.1 Contextualização	1
1.2 Relevância da pesquisa	4
1.3 Objetivos	5
1.4 Estrutura do trabalho	6
2 <i>Green IT</i>	7
2.1 Contextualização de <i>Green IT</i>	8
2.2 Técnicas de <i>Green IT</i>	9
2.2.1 Desligar os equipamentos ociosos	9
2.2.2 Modos para economizar energia	10
2.2.3 Eliminar a carga fantasma	11
2.2.4 Aumentar o ciclo de vida	11
2.2.5 Consolidação de servidores	12
2.3 Benefícios de <i>Green IT</i>	14
2.4 Adesão	16
3 Virtualização	19
3.1 Conceito	19
3.2 História da virtualização	19

3.3	<i>Hypervisor</i>	20
3.4	Técnicas de virtualização	21
3.4.1	Emulação de <i>hardware</i>	21
3.4.2	Virtualização completa	23
3.4.3	Paravirtualização	24
3.4.4	Virtualização em nível de sistema	25
3.5	Ferramentas para virtualização	25
3.5.1	VMware	26
3.5.2	Windows Virtual PC	28
3.5.3	VirtualBox	29
3.5.4	Xen	30
3.5.5	KVM	30
3.5.6	OpenVZ	32
3.6	Benefícios da virtualização	32
3.6.1	Consolidação e otimização de recursos	33
3.6.2	Aumento da flexibilidade	34
3.6.3	Melhoria no custo total de propriedade	34
3.6.4	Facilidade no gerenciamento	35
3.6.5	Aumento do tempo de atividade e continuidade do serviço	35
3.6.6	Redução de custos e energia	36
4	Implementação	37
4.1	Primeira fase: montagem da infraestrutura	38
4.2	Segunda fase: criação e definição das máquinas virtuais	40
4.3	Terceira fase: implementação da ferramenta	41
5	Testes e Resultados	45
5.1	Geração de carga contínua	46
5.2	Geração de carga dinâmica	48
5.3	Resposta da rede e tempo de migração	51
5.4	Estudo de aplicação em caso real	54

6 Conclusão	56
6.1 Trabalhos futuros	57
A Instalação do Servidor Nagios	58
B Instalação do Cliente Remoto Nagios	70
C Instalação de Servidor e Cliente NFS	75
D Instalação do KVM	77
E Configurando chaves de autenticação para SSH	82
Glossário	84
Referências Bibliográficas	85

Resumo

O grande aumento na quantidade de equipamentos eletrônicos têm levado a questionamentos em relação aos danos causado ao meio ambiente. Empresas que fabricam estes equipamentos estão buscando maneiras para reduzir este impacto e reciclar o máximo possível seus componentes. A área de *Green IT* tem papel fundamental no estudo e desenvolvimento de técnicas para uso pessoal, empresarial e em *datacenters*, para redução dos prejuízos ao ambiente. Como uso pessoal e empresarial existem técnicas para reduzir o gasto de energia dos computadores e equipamentos, aumento do ciclo de vida e eliminação do efeito fantasma. Em *datacenters* uma das soluções é a consolidação de servidores, que em conjunto com a virtualização, atua como um gerenciador de recursos deixando o mínimo possível de máquinas físicas em atividade, reduzindo assim a quantidade de máquinas ociosas e os gastos com energia elétrica.

As ferramentas existentes que proporcionam a consolidação de servidores requerem a aquisição de licença, já que todos são softwares proprietários. Este trabalho teve como objetivo a implementação de uma ferramenta para consolidação de servidores em software livre. Nesta implementação foi utilizado o sistema operacional GNU/Linux e como *hypervisor* o KVM.

Na realização dos testes foi necessário a utilização de carga gerada, monitorando o tempo de migração das máquinas virtuais e o tempo de resposta das mesmas durante a migração.

Palavras-chave: Virtualização, Consolidação de servidores, *Green IT*, *Green Computing*, KVM, software livre.

Capítulo 1

Introdução

1.1 Contextualização

Devido a problemas com o aquecimento global e esgotamento dos recursos naturais, as pessoas estão cada vez mais preocupadas com a questão ambiental. Todos os setores da economia buscam de forma constante prolongar ao máximo o fim dos recursos, e a estagnação ou diminuição do aquecimento.

Com a diminuição do valor para aquisição de computadores, *laptops*, *smartphones* e outros aparelhos eletrônicos, estes equipamentos tornaram-se mais populares. Este aumento do número de eletrônicos obriga as grandes empresas de telecomunicações a reestruturar sua infraestrutura, provocando um maior investimento em centros de processamento e armazenamento de dados e de redes de alta velocidade.

Projetistas da área de TI (Tecnologia da Informação) começaram a se preocupar com o meio ambiente, fazendo com que, por exemplo, um processador diminua a sua frequência quando estiver baixa carga de trabalho para economia de energia. Para um simples computador ou *laptop*, a economia pode ser considerada pequena, porém, em grande escala, esta diminuição é muito atrativa. Este é o caso dos centros de processamento de dados ou *datacenter*.

Um erro comum, é pensar que apenas provedores de serviços de telecomunicações possuem *datacenters*. Em grandes países, empresas de médio e grande porte possuem centro de processamento de dados próprio. Em pesquisa feita em nos Estados Unidos pelo IDC (*International Data Corporation*), em 2008 já existiam cerca de 2,5 milhões de *datacenters* [10]. Estudos revelam que 1% da energia elétrica gerada no planeta é utilizada para manter os *datacenters*. Esta quantidade de energia tende a crescer de forma vertiginosa, já que o número de servidores

em uso no mundo aumentou de 6 milhões para 28 milhões entre os anos 2000 e 2007.

Desde a construção até a finalização do projeto de um *datacenter*, deve-se seguir um padrão internacional, a norma ANSI/TIA/EIA-942, desenvolvida em conjunto por ANSI (*American National Standards Institute*), TIA (*Telecommunications Industry Association*) e EIA (*Energy Information Administration*). Em conjunto com esta norma, ainda existem a ASHRAE (*American Society of Heating, Refrigerating and Air Conditioning Engineers*) e IEEE (Instituto de Engenheiros de Eletricistas e Eletrônicos). A ASHRAE é uma sociedade técnica internacional, que tem o objetivo de promover a sustentabilidade no que diz respeito a ventilação, refrigeração e ar condicionado. Todo este esquema é representado na figura 1.1 [1].

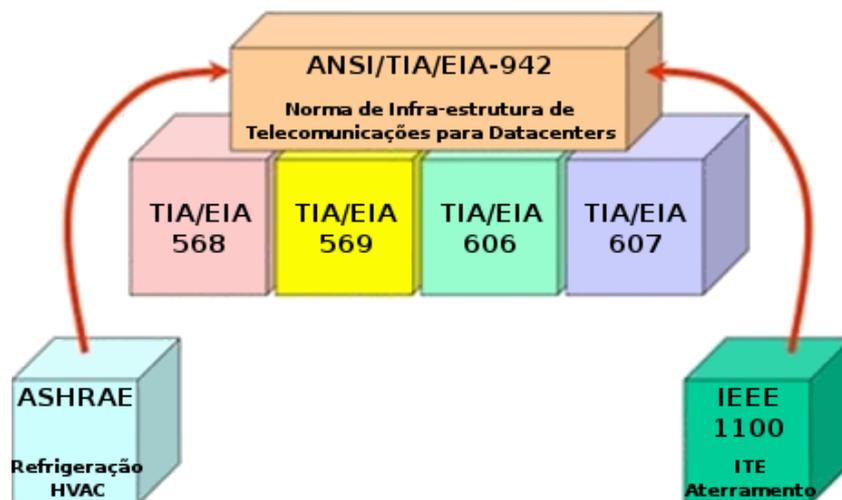


Figura 1.1: Norma de infraestrutura de telecomunicações para *datacenters* (adaptado de [1]).

O investimento em infraestrutura para construção de um *datacenter* gira em torno de milhões de reais, podendo chegar a bilhões, dependendo da capacidade e estrutura. Para evitar o superaquecimento dos servidores, evitando o desgaste acelerado do *hardware*, a ASHRAE controla um dos principais consumos elétricos, a refrigeração. Estima-se que refrigeração é culpada por até 50% dos gastos para manutenção do *datacenter* [11].

Outro problema com relação ao gasto com energia para processamento de dados, podendo ser considerado um desperdício, é com relação ao percentual médio de utilização dos servidores. Todo computador possui um poder de processamento, porém nem sempre um servidor irá operar utilizando toda esta capacidade. Este é um dos motivos do desenvolvimento da virtualização. Com a virtualização é possível que uma máquina física otimize o uso de seu processamento,

rodando várias máquinas virtuais em apenas uma máquina física. Esta máquina física então terá seu processamento dividido entre N máquinas virtuais, sendo ativadas apenas quando necessárias.

Além do benefício de facilitar o gerenciamento do processamento, temos a diminuição da quantidade de máquinas físicas, já que o ideal é manter um serviço para cada servidor, garantindo maior segurança e disponibilidade.

Com vários servidores virtuais rodando em uma máquina física, tem-se como efeito colateral, uma possível parada de todos os serviços se houver algum problema na máquina física. Este problema pode ser evitado com técnicas de alta disponibilidade, tal como redundância de servidores.

As preocupações em relação ao gasto de energia e o mal gerenciamento da mesma, incentivaram ainda mais o desenvolvimento da *Green Computing* ou *Green IT (Information Technology)*. A *Green IT* é a prática do uso de recursos computacionais de forma eficiente, tendo como objetivos, maximizar a eficiência da utilização de energia durante a vida útil do produto e promover a reciclagem ou biodegradabilidade de produtos em desuso [12]. Outra definição aposta para o desenvolvimento responsável, redução de custos e melhor gerenciamento no uso dos recursos ambientais. Para isto investe-se na implementação de computadores e periféricos com gasto eficiente de energia, bem como o consumo reduzido de recursos e destinação adequada de lixo eletrônico [13].

Segundo estudo patrocinado pela Symantec, no ano de 2007 através de discussões, empresas começaram a considerar *Green IT* [14], adotando-os recentemente. Somente com a utilização de alternativas para refrigeração, a redução dos gastos pode chegar em até 40%. No ano de 2010, UOL e Yahoo!, duas grandes empresas provedoras de internet, inauguraram seus *datacenters* verdes e, no caso do Yahoo!, além da economia de aproximadamente 40% com gastos em refrigeração, teve também a diminuição de 95% do uso de água [15].

Dentro deste contexto a *Green IT* tem grande importância através de suas técnicas, sempre motivadas pela sustentabilidade. Uma destas técnicas é a consolidação de servidores, que tem papel fundamental no desenvolvimento de *green datacenter*, atuando através do seu principal componente, que é virtualização dos servidores.

Na consolidação de servidores utiliza-se uma máquina física hospedando N máquinas virtu-

ais, sendo uma para cada servidor, reduzindo o número de máquinas físicas. Ao consolidar os servidores é possível fazer um gerenciamento da quantidade de máquinas físicas em operação e a energia consumida pela mesma. O gerenciamento causa uma redução de energia, pois uma máquina física entra em estado ativo apenas se há a necessidade.

As principais ferramentas que proporcionam a consolidação de servidores são proprietárias, e com base neste problema, o foco principal desta pesquisa é o desenvolvimento da consolidação de servidores, implementando uma ferramenta, utilizando software livre.

Para demonstrar melhor o funcionamento de consolidação de servidores, como exemplo, podemos pegar uma empresa que trabalha apenas em horário comercial, durante a noite e madrugada não há necessidade de todas as máquinas físicas ligadas, já que o processamento é reduzido consideravelmente. Com esta redução as máquinas virtuais são migradas para apenas uma máquina física em operação, deixando as demais em estado “dormindo”. Durante o período comercial, conforme a exigência de processamento, as máquinas físicas acordam e os servidores virtualizados migram para a quantidade de máquinas físicas necessárias que corresponda a quantidade de processamento exigido. Ao final do expediente, retorna o ciclo para evitar que as máquinas físicas fiquem ociosas. Com este esquema tem-se um gerenciamento melhor sobre recursos do *hardware* e redução no uso de recursos energéticos em caso de inatividade ou uso parcial do processamento do servidor.

1.2 Relevância da pesquisa

Devido ao fato da preocupação com o excessivo gasto de recursos energéticos dos *datacenters*, os estudos para redução destes custos tem se tornado mais relevantes nos últimos anos. A utilização da virtualização e consolidação de servidores é um dos principais métodos para melhor aproveitamento dos recursos, seja ele de *hardware* ou energético. O *datacenter*, que possui servidores consolidados, tem a garantia da redução considerável nos gastos com energia elétrica e uma série de benefícios, por exemplo, aumento na otimização de recursos, melhor flexibilidade, melhor gerenciamento dos servidores, redução de custos, diminuição da parada de serviço, etc. [16].

Existem muitas ferramentas que dão suporte para virtualização, entre elas temos Virtual-Box, KVM (*Kernel-based Virtual Machine*), VMware, Xen etc., que distinguem entre softwares

proprietários e softwares livres. Dentre estas ferramentas, algumas possuem o sistema de controle para consolidação de servidores, por exemplo VMware. Outras opções que não possuem o controle, existem alternativas em empresas que desenvolvem ferramentas para consolidação e virtualização como a Uptime Software¹. As duas soluções apresentadas para consolidação, possuem a peculiaridade de serem proprietárias e requererem o pagamento de licença.

Nem todas as empresas e organizações estão dispostas a desembolsar grandes valores em licenças, para isto, o desenvolvimento deste trabalho surge como uma alternativa bastante interessante. O estudo e desenvolvimento do trabalho está baseado no uso de tecnologias livres, uma alternativa que impõe a necessidade do estudo das ferramentas existentes para virtualização e monitoramento, para posterior desenvolvimento de uma ferramenta similar em software livre.

1.3 Objetivos

Considerando as questões relacionadas ao gasto de energia elétrica e gerenciamento dos recursos, conforme seção 1.1, e levando em conta as questões apresentadas sobre as ferramentas na seção 1.2, a proposta do trabalho objetivou principalmente a implementação de um controle para a consolidação de servidores, utilizando software livre. Este objetivo pode ser dividido nos seguintes pontos:

- Conceituar e aprofundar o conhecimento a respeito de consolidação de servidores, bem como estudar casos implementados;
- Estudar as ferramentas de virtualização e migração entre máquinas virtuais, principalmente o software KVM;
- Monitorar o uso dos recursos de hardware das máquinas físicas;
- Implementar um sistema de controle para consolidação de servidores;
- Testar os impactos gerados com o uso de migração e da implementação da consolidação, analisando as questões de *downtime*, inatividade, custo etc.;

¹<http://www.uptimesoftware.com/server-virtualization.php>

- Utilizar software livre durante o processo de estudo e desenvolvimento do trabalho, desde o sistema operacional utilizado na máquina física, até mesmo a ferramenta que será implementada.

1.4 Estrutura do trabalho

O trabalho está organizado estruturalmente em 6 capítulos, divididos da seguinte maneira:

- Capítulo 2 - *Green IT*: inicialmente será apresentado a fundamentação teórica de TI, bem como a importância que ela possui dentro das empresas independente do tamanho. Após será apresentado o conceito de *Green IT*, os principais motivos que estão levando as empresas a utilizá-la, seus benefícios e quais são as empresas que estão utilizando massivamente seus conceitos.
- Capítulo 3 - Virtualização: apresenta-se um breve histórico e o conceito da virtualização, benefícios que um *datacenter* virtualizado tem sobre um *datacenter* comum, as principais ferramentas disponíveis no mercado e suas funcionalidades e as técnicas mais utilizadas.
- Capítulo 4 - Implementação: descreve-se o processo de montagem, instalação e configuração da estrutura de simulação de uma infraestrutura de TI virtualizada, e também a construção da ferramenta de consolidação de servidores.
- Capítulo 5 - Testes e Resultados: especifica-se os casos de testes realizados e os resultados obtidos.
- Capítulo 6 - Conclusão: para finalizar apresenta-se as principais considerações do desenvolvimento deste trabalho e os possíveis trabalhos futuros.

Além destes 6 capítulos, descreve-se nos apêndices todos os passos para instação e configuração das ferramentas utilizadas neste trabalho.

Capítulo 2

Green IT

Devido a grande popularização dos computadores e com o surgimento de novas tecnologias, houve o aumento considerado no volume de informações processadas e enviadas. Esta informação pode variar entre tarefas simples, como enviar e-mails ou navegar pela internet, chegando a patamares com alto nível de processamento de dados confidenciais, exigindo segurança da informação transmitida e armazenada, por exemplo, instituições financeiras e órgãos do governo.

Segundo Aguilar [17] podemos definir a tecnologia da informação como “...um grupo de todas as atividades e soluções que utilizam recursos de computação. As aplicações são tantas e estão ligadas às mais diversas áreas, tanto é que existem várias definições e nenhuma consegue defini-la por completo”.

Também segundo Keen apud Laurindo et al. [18] “O conceito de tecnologia da informação é mais abrangente do que os de processamento de dados, sistemas de informação, engenharia de software, informática ou o conjunto de hardware e software, pois também envolvem aspectos humanos, administrativos e organizacionais”. Esta diversificação das áreas proporcionam, a empresas e organizações, rapidez, competitividade, redução de custos, produtividade e integração [19].

Com todos esses benefícios, a TI deixou de ser coadjuvante, tornando-se responsável por muitas melhorias, integrando setores, ganhando produtividade e importância na tomada de decisão, auxiliando no alcance de metas.

O grande aumento no número de servidores, e um dos principais componentes da infraestrutura de TI, o *datacenter*, começou a ter cada vez mais relevância. Isto ocorreu devido a necessidade de equipamentos mais eficientes, porém como efeito colateral, houve o aumento da quantidade consumida de energia elétrica. Este fato está ligado a grande quantidade de

calor gerado, com isso, para resfriar uma quantidade maior de servidores, é necessário uma quantidade maior de condicionadores de ar.

Outra preocupação está relacionada à redução e extinção de produtos químicos que causam danos ao meio ambiente, para fabricação de equipamentos e produtos eletrônicos, bem como o destino que os mesmos levam ao se tornarem obsoletos. Por estes e outros motivos, a *Green IT* surge com o propósito de melhor gerenciamento dos recursos, para diminuição dos danos causados a natureza.

2.1 Contextualização de *Green IT*

Uma das principais preocupações nas últimas décadas, é o patamar assustador no qual a poluição e a destruição do meio ambiente se encontram. Nos últimos 30 anos um terço dos recursos naturais foram consumidos [20]. Diante destes problemas, governos buscam incessantemente maneiras para retardar estes danos. A nível mundial existem tratados, como por exemplo o Protocolo de Kyoto, assinado por 175 países, muito conhecido e difundido pelas idéias de redução dos gases que aumentam o efeito estufa e por promover o uso de fontes energéticas renováveis.

Estas mudanças na natureza tiveram a participação massiva do ser humano, que a partir do século XVIII, com a revolução industrial na Inglaterra, iniciou-se o processo de mecanização da mão de obra do homem, com isso alguns recursos naturais, não utilizados até então, começaram a ter grande importância e necessidade, por exemplo, combustíveis fósseis. Iniciava-se então a era das máquinas, trazendo como efeito colateral o aumento excessivo da liberação de CO₂ (dióxido de carbono), principal causador do efeito estufa que degrada o meio ambiente. Segundo o Ministério da Ciência de Tecnologia do Brasil [21], “*A Revolução Industrial mudou para sempre a relação entre o homem e a natureza*”.

Após a revolução industrial, iniciou-se a era das inovações tecnológicas, no século XX, trazendo facilidades a vida cotidiana. Com os benefícios proporcionados pela tecnologia e o aumento populacional, nos últimos anos o número de aparelhos eletrônicos em geral está em constante avanço, aprimorando o desenvolvimento da área de TI.

A área de TI contribui de diversas maneiras na degradação do meio ambiente. Segundo estudos da consultoria Forrester Research, a área de TI contribui em média com 10% das emissões

de dióxido de carbono e 10% do consumo de energia das empresas. Em estimativa apresentada pela consultoria Gartner, aproximadamente 2% do dióxido de carbono emitido na atmosfera, tem como responsável os computadores [22]. Neste aspecto surge a *Green IT*, uma área da tecnologia da informação que busca, através de estudos, o desenvolvimento de técnicas para aumentar a eficiência energética, reduzir o lixo e resíduos emitido por equipamentos eletrônicos, desde a fabricação até tornarem-se obsoletos [23, 24]. Na seção 2.2 tem-se detalhes a respeito destas técnicas desenvolvidas.

2.2 Técnicas de *Green IT*

Diante deste contexto, muitas técnicas foram desenvolvidas para auxiliar a área de TI tornar-se *green*. Nas seções 2.2.1, 2.2.2, 2.2.3 e 2.2.4 têm-se técnicas que podem ser utilizadas no cotidiano de empresas e em casa, e na seção 2.2.5 a técnica de consolidação de servidores utilizada em *datacenters* virtualizados.

2.2.1 Desligar os equipamentos ociosos

Desligar os equipamentos é a maneira mais simples, eficaz e óbvia para reduzir o consumo de energia elétrica. Muitas pessoas mantêm seus equipamentos ligados diariamente, evitando desligá-los mesmo quando eles estão ociosos.

Este hábito pode estar relacionado a alguns os mitos citados por Marcais [25]. O primeiro mito diz respeito a diminuição da vida útil, o que não procede, pois todos os dispositivos que formam o computador foram desenvolvidos e preparados para serem desligados. Já o segundo está relacionado ao aumento de carga necessária no momento de desligar o computador, o que é verdade, porém se o computador ficar ligado durante 10 minutos, ele irá consumir mais energia do que a gasta para desligar e religar o equipamento. Por último temos as atualizações, onde usuários deixam seus equipamentos ligados durante a madrugada para atualizar seus sistemas operacionais, porém durante a madrugada é onde ocorrem a maior quantidade de ataques de *hackers*.

2.2.2 Modos para economizar energia

Os sistemas operacionais possuem algumas maneiras para auxiliar na economia de energia, porém alguns desses métodos não são utilizados pois usuários não gostam de esperar que um computador “adormecido” volte ao seu estado normal.

- Protetor de tela

O primeiro método tem o objetivo de economizar energia e evitar o efeito de queima dos *pixels*, porém em relação à economia de energia ele tem um efeito contrário ao proposto [26]. A utilização de proteção de tela durante um ano é equivalente a deixar uma lâmpada de 100 *watts* ligada durante o mesmo período [25].

- Monitor no modo “adormecer”

Este método é muito útil e deve ser utilizado em todos os computadores. Um monitor que possua um gasto de 55 *watts* no modo normal, tem seu consumo abaixado para 3 *watts* no modo “adormecer” [26].

- Disco rígido no modo “adormecer”

Um disco rígido possui um gasto de aproximadamente 10 *watts* em computadores de mesa, e um consumo um pouco inferior em *laptops* [26]. Com o disco rígido “adormecido” o consumo de energia é diminuído, e o serviço volta ao estado normal segundos após acionado.

- Sistema em espera

O sistema em espera, ou *standby*, é um dos métodos mais eficientes para economia de energia elétrica. Um computador que possui o gasto de 100 *watts*, no estado normal, tem seu consumo reduzido para 5 *watts* [26]. Este método mantém a memória RAM (*Random Access Memory* ou Memória de Acesso Aleatório) ativa, deixando os demais dispositivos em modo “adormecido”, com isso assim que acionado o sistema operacional volta para seu estado de execução antes do estado adormecer.

- Modo hibernar

Método mais eficiente que o sistema em espera, já que um computador de mesa no modo hibernar tem um gasto de 3 *watts* contra os 5 *watts* do sistema em espera [26]. A eficiência

deste método possui como efeito colateral um consumo maior de tempo para retorno ao estado normal, este fato deve-se ao tempo para salvar e restaurar da memória e do disco [27].

2.2.3 Eliminar a carga fantasma

Muitos aparelhos eletrônicos apresentam um problema chamado carga fantasma. A carga fantasma é a taxa mínima que um equipamento utiliza mesmo após desligado. Isto pode ser notado em monitores, por exemplo, no modo “adormecer”, sistema em espera e modo hibernar. Em apenas uma máquina o problema não é muito relevante, porém em uma empresa que possua 50 computadores, e estes forem colocados no modo hibernar ao final de um ano teremos um gasto considerável de energia que é considerado um desperdício.

Para eliminar a carga fantasma é aconselhável retirar o cabo de força da tomada ou utilizar um equipamento que corte a corrente, por exemplo, estabilizador ou filtro de linha. Se o computador e todos os periféricos utilizados pelo mesmo estiverem ligados em um filtro de linha, assim que desligar o computador basta colocar o chaveador do filtro de linha no modo desligado e todos os equipamentos estarão eliminando a carga fantasma [28, 29].

2.2.4 Aumentar o ciclo de vida

A aquisição de novos equipamentos é inevitável, já que a área de TI muda constantemente com novas tecnologias. Todos os equipamentos do mercado devem ser estudados, buscando otimizar o ciclo de vida, para evitar a troca constante em um período de tempo inferior ao esperado, pois isso eleva a quantidade de lixo eletrônico. Também deve-se verificar a quantidade de energia que o mesmo gasta, optando sempre pelo mais econômico [27].

É interessante como o mercado tem ido exatamente contra isso, sempre lançando dispositivos que tem vida útil menor, placas de vídeo que demandam maior consumo de energia, tecnologias que requerem cada vez *hardwares* mais eficientes e tornando obsoletos os equipamentos mais antigos.

2.2.5 Consolidação de servidores

O *datacenter*, um dos principais gastos com energia elétrica das empresas, não poderia ficar de fora destes estudos. Um fator preocupante é que empresas estão necessitando mais servidores com poder computacional cada vez maiores, porém, recursos como espaço e energia, estão se tornando escassos.

Para otimizar o uso do poder computacional e os gastos com energia elétrica, um termo bastante conhecido é a utilização de virtualização. Na virtualização, uma máquina física tem sua capacidade de processamento dividida em várias máquinas, chamadas de máquinas virtuais. As máquinas virtuais têm a quantidade de processamento, memória e armazenamento que serão utilizadas de maneira especificadas em sua definição. Com a utilização de máquinas virtuais, a relação 1 para 1, onde cada máquina física possui um servidor, é substituída por N para 1, com isso o poder computacional é melhor gerenciado. A diminuição das máquinas físicas proporciona a redução do gasto com refrigeração, isto se deve ao fato da diminuição do número de máquinas físicas, e conseqüentemente o calor gerado por elas. No capítulo 3 será detalhado as principais informações a respeito de virtualização.

Todas as ferramentas mais populares que possibilitam a virtualização, como por exemplo VMware, Xen, VirtualBox, KVM etc.; possuem a tecnologia *live migration*, que possibilita a migração de uma máquina virtual entre duas máquinas físicas, sem a necessidade de parada de serviço. Um fator muito relevante que pode ser agregado à virtualização e à *live migration*, é a consolidação de servidores.

A consolidação de servidores pode ser considerada um gerenciamento dos recursos de *hardware* tornando-os otimizados. Com sua utilização é possível ter um controle da infraestrutura sendo possível gerenciar o uso das máquinas virtuais, e conforme a demanda de processamento, e migrá-las para uma quantidade menor de máquinas físicas, deixando as demais desligadas, sem consumo de energia elétrica. Esta redução no consumo energético se torna irrelevante em uma infraestrutura que possui poucas máquinas físicas, porém em uma quantidade maior, a redução é satisfatória.

Segundo Kazandjieva [30], “*Consolidação de servidores, combinado com migração de máquinas virtuais e um gerente de datacenter, podem ser usados não só para balancear as máquinas virtuais entre máquinas físicas, mas também para diminuir e desligar máquinas físicas*

desnecessárias”. Entre vários outros benefícios, a consolidação é vista como uma forma eficaz de reduzir a emissão de dióxido de carbono.

Para se ter noção do benefício proporcionado pela consolidação, a figura 2.1, apresenta o resultado de uma simulação de consolidação de servidores. Como exemplo, cria-se uma empresa fictícia que possui seu horário de trabalho das 8h às 12h e das 13h30 até 17h30, contendo cinco máquinas físicas com servidores virtualizados em sua infraestrutura de TI.

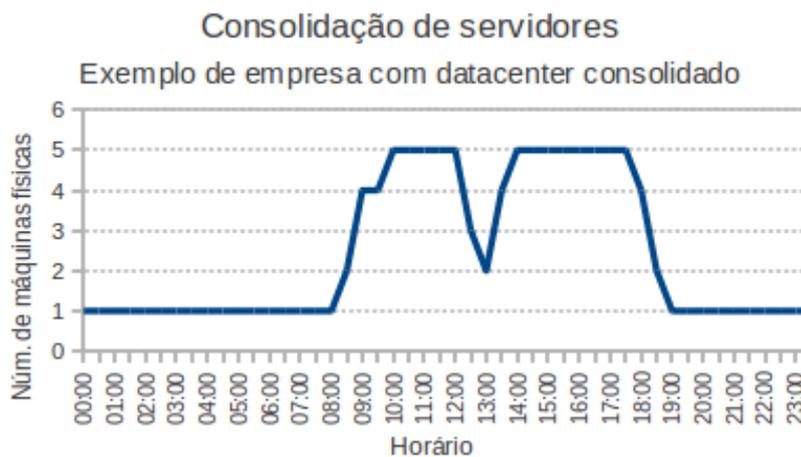


Figura 2.1: Exemplo de empresa com *datacenter* consolidado.

Todos os servidores virtuais estão ocupando apenas uma máquina física, neste caso a utilização de servidores de e-mail, impressão etc., possuem um aumento durante o horário que os funcionários estão trabalhando, então a exigência em relação ao poder computacional aumenta, obrigando desta maneira, os servidores buscarem máquinas físicas que proporcionem o poder computacional requerido. Com isso, a máquina física só entra em estado ativo se houver a necessidade do uso da mesma. Ao final do expediente, onde o uso dos servidores é reduzido, os servidores migram-se para uma quantidade menor de máquinas físicas, deixando as demais desligadas. Este gerenciamento que a consolidação de servidores oferece, causa uma redução considerável no consumo de energia elétrica.

Ao tornar o *datacenter* verde, os benefícios em relação ao gerenciamento computacional e a redução de energia elétrica são muito grandes. Outros benefícios são vistos na seção 2.3.

2.3 Benefícios de *Green IT*

Os motivos e benefícios que levam as empresas a investir em *green IT* são muitos, dentre os principais pode-se citar três: (1) as alterações climáticas, onde pesquisas concluem que um dos responsáveis pelos danos ambientais é a emissão de CO₂; (2) economia, com o passar do tempo, ocorrerão reduções de gastos com resfriamento e melhor gerenciamento dos recursos computacionais, conseqüentemente, diminuição do gasto com energia elétrica, causando um grande impacto positivo na economia da empresa; (3) e a valorização da energia, isso se deve ao fato de que empresas estão utilizando meios de gerar sua própria energia elétrica, desta maneira, valorizando este recurso que poderá atingir um ponto crítico, caso não for lhe dado a devida atenção [31]. Na figura 2.2 existe uma relação entre o antes e depois de uma empresa que investiu em *Green IT*, destacando-se como benefício, a melhor imagem da empresa e um orçamento que possibilita o crescimento.

Antes		→	Depois
Financeiro 	Aumento do custo de energia elétrica Diminuição dos orçamentos de TI Restrições sobre o crescimento de TI	  	Visualização da linha base de uso de energia Redução dos custos e eficiência energética Maior orçamento possibilitando o crescimento
Operacional 	Servidor de sistemas densos Custos com energia e refrigeração Envelhecimento dos datacenters	  	Maior desempenho por quilowatt Redução da energia de operação e refrigeração Vida útil prolongada das instalações existentes
Ambiental 	Responsabilidade social das empresas Falta imagem pública Melhorar a moral do	  	Significativa economia de energia e diminuição das emissões de dióxido de carbono Melhor imagem pública Contribuição positiva para o movimento Green criando um bom lugar para trabalhar

Figura 2.2: Benefícios da implantação da *Green IT* (adaptado de [2]).

Segundo Ebbers et al [2], pode-se dividir em quatro itens as principais preocupações dos CIOs para estruturação de um *datacenter*. Estes itens devem ser levados em conta para estruturar bem um *datacenter*, prevendo modificações ou expansões e principalmente direcionado para *green IT*.

- Gerenciar o uso da energia elétrica - uma das principais preocupações das organizações é

o custo do quilowatt/hora de energia elétrica, que está em constante crescimento. Estudos feitos pelo Uptime Institute, revelam que os custos para manter os servidores e a refrigeração durante três anos, é equivalente a uma vez e meia o custo de aquisição do hardware dos servidores. Com a necessidade de servidores mais potentes e de alta performance, um fator a se preocupar é o orçamento gasto para manter os servidores.

- Esgotamento da capacidade energética - em algumas organizações, já existe a limitação de energia elétrica disponível para implantação de novos servidores. Isto acontece principalmente em áreas urbanas muito povoadas, onde empresas fornecedoras de energia não possuem disponibilidade para alimentar quantidades excessivas de servidores. Para fazer com que os limites aumentem, seria necessário um investimento muito alto.
- Esgotamento da capacidade de refrigeração - existem muitas salas que abrigam *datacenters* que foram projetadas há 15 anos atrás, e apesar de haver um projeto visando a manutenção dos servidores para manter a boa qualidade de serviço, as instalações de refrigeração possuem uma capacidade limitada tendo em vista os equipamentos atuais. Os métodos de refrigeração comum necessitam até 3kW por *rack*, porém os equipamentos atuais necessitam entre 20 e 30kW por *rack*.
- Necessidade de espaço - apesar da tecnologia estar muito avançada e possibilitar uma quantidade maior de processamento em menor espaço físico, no caso de equipamentos para processamento e armazenamento de informações em grande escala, esta redução nem sempre é notada, por exemplo, em um *datacenter* que necessita uma unidade de armazenamento nova. Se a sala onde os equipamentos estão não possui mais espaço, não haverá um lugar ideal disponível para o *storage*. Uma solução seria a construção de outra sala, porém isso é muito caro, pois deve-se aplicar as normas de segurança para a estrutura.

Com todos os fatores descritos, muitas empresas já aderiram a *green IT*. Na seção 2.4, vamos citar as principais grandes empresas que tem uma atenção especial em relação a diminuição de danos causados a natureza.

2.4 Adesão

Nos últimos anos, o número de empresas que se preocupam com o meio ambiente está em constante desenvolvimento. Todas as empresas que investem em *green IT*, possuem um grande crescimento e destaque em relação as demais. Com um melhor gerenciamento do poder computacional, redução dos desperdícios e gastos com energia elétrica, e a otimização de processos, tem aumentado o lucro destas empresas.

Entre as grandes empresas de TI que estão investindo pesado na questão ambiental, podemos citar provedores de internet, grandes marcas de celular e fabricantes de equipamentos para informática. Também existem empresas em outros setores da economia, como por exemplo, no setor bancário. Abaixo vamos descrever sucintamente algumas destas empresas e qual a relação que elas tem com a preservação do meio ambiente.

- Google: a empresa não investe somente na questão de *green IT*. Foi a pioneira ao instalar placas de células fotovoltaicas para geração de energia. Ela firmou um acordo com grandes organizações como Microsoft, IBM, Lenovo, Dell, Intel, HP e outras 27 empresas, firmando o acordo conhecido como *Climate Savers Computing*, tendo como objetivo definir metas para produção de equipamentos e componentes, visando a eficiência energética [32].
- Yahoo: tem um dos projetos mais ambiciosos, pois tem o objetivo de neutralizar a emissão de dióxido de carbono. Para cumprir com este objetivo, a preocupação se estende em toda a empresa, que em seus *datacenters* utiliza virtualização. Outro fator são os *datacenters* que estão sempre de acordo com as normas ambientais [32].
- Apple: se preocupa de uma maneira geral com a questão ambiental. Em seus produtos pode-se destacar a utilização de materiais duráveis e recicláveis. Os monitores utilizados em *laptops* não utilizam mercúrio em sua composição [32].
- Nokia: não produz mais celulares com embalagens de PVC. Está preocupada com a reciclagem e o descarte das baterias de maneira correta. De todos os componentes que constituem o celular, entre 60% e 85% são recicláveis [17].

- Motorola: também tem uma preocupação muito grande com o descarte das baterias. Tem como objetivo eliminar a utilização de metais pesados da composição do celular. Possui um modelo 100% ecologicamente correto, o Moto W233 Eco, onde todos os componentes do celular são recicláveis e o plástico é feito de garrafas recicladas [32].
- Banco do Brasil: trocou vários monitores CRT (Tubo de Raios Catódicos) para LCD (Display de Cristal Líquido), que utilizam menos energia. Utiliza água da chuva no *datacenter* e a refrigeração é feita com gases que não afetam a camada de ozônio. Também utiliza geradores de energia que utilizam biodiesel, que limita a fuligem [17].
- Itaú Unibanco: o grupo tem preocupação de um modo geral com o meio ambiente desde 2004. No ano de 2008 foi criado um comitê de *green IT*, que divide-se em eficiência energética, lixo eletrônico e *green workplace* (ações para tornar sustentável o ambiente de trabalho). Em 2009 o primeiro comitê, conseguiu a redução de 2 Megawatts por hora. Segundo João Antonio Dantas Bezerra [22], “*Como exemplo de ações que têm contribuído para esses números, o diretor cita a consolidação de datacenters, a virtualização de servidores, a modernização dos sistemas de refrigeração e a troca de monitores CRT por LCD*”. Apenas a modernização do *datacenter*, causou uma redução de 40% de energia elétrica. O segundo comitê conseguiu, apenas no ano de 2009, a reciclagem de 125 toneladas de equipamentos. Por fim, o terceiro comitê trabalhou reduzindo a impressão, aumentando o tempo de vida útil dos equipamentos e evitando o deslocamento dos profissionais.

Além das empresas citadas acima, existem inúmeras que têm ações em prol do meio ambiente. A organização global Greenpeace, que tem como foco defender o ambiente e promover a paz, disponibiliza em seu site o “Termômetro verde”, onde é possível acompanhar o andamento das ações das empresas de eletrônicos a favor do meio ambiente. A figura 2.3 mostra o resultado do estudo feito em outubro de 2010, onde a Nokia desbanca como a empresa que possui melhores atitudes em relação ao meio ambiente. O documento completo que descreve o “Termômetro Verde” é o “Guia de Eletrônicos Verde”, que pode ser obtido no site do Greenpeace¹.

Segundo o Greenpeace [3] “O “*Guia de Eletrônicos Verde*” pretende limpar o setor de

¹<http://www.greenpeace.org/international/en/campaigns/toxics/electronics/Guide-to-Greener-Electronics/>

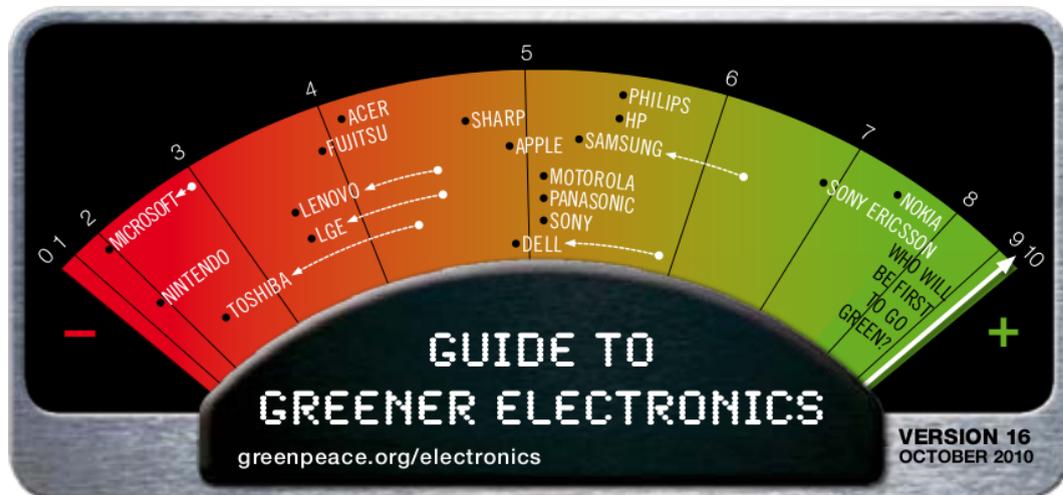


Figura 2.3: Termômetro verde da Greenpeace [3].

eletrônica e obrigar os fabricantes a assumir a responsabilidade pelo ciclo de vida de seus produtos, incluindo o lixo eletrônico que seus produtos geram.”.

Os critérios de classificação do “Termômetro Verde” são baseados em três quesitos, (1) como as empresas estão limpando os seus produtos, eliminando substâncias que causam danos ao meio ambiente; (2) se a retomada e reciclagem de seus produtos é feita de forma responsável; (3) e como está a redução dos impactos climáticos das suas operações e seus produtos. Mais informações e o estudo completo sobre as empresas envolvidas estão disponíveis no site do Greenpeace [3].

Capítulo 3

Virtualização

3.1 Conceito

A virtualização de servidores é uma das maiores tendências atuais da indústria de TI. Segundo Nanda e Chuieh [33] “A virtualização é uma tecnologia que combina ou divide os recursos computacionais para apresentar um ou vários ambientes operacionais como a utilização de metodologias de hardware e particionamento ou agregação de software, máquina de simulação parcial ou total, emulação, time-sharing, e muitos outros”.

Ao contrário do que muitas pessoas pensam, o conceito de virtualizar servidores não foi criado nos últimos anos, ele apenas difundiu-se em grande escala. O início da virtualização aconteceu na década de 1960, porém seu uso e difusão aconteceram no final da década de 1990, chegando a tornar-se de grande importância para toda e qualquer organização. Antes de entrar em detalhes técnicos, na seção 3.2 tem-se um breve histórico de sua evolução.

3.2 História da virtualização

Na década de 1960, surgiu o termo *time-sharing*, que significa compartilhamento de tempo, e hoje é conhecido como a virtualização. Na época, o primeiro professor de computação da Universidade de Oxford, Christopher Strachey, dedicou-se à pesquisa em um trabalho conhecido como *Time Sharing in Large Fast Computers* (Compartilhamento de tempo em grandes computadores), o qual fazia referência a multiprogramação. O objetivo dele era possibilitar que enquanto uma pessoa criasse um programa no console, outra pessoa depurasse o mesmo, evitando a espera para uso do periférico [34].

Nesta época surgiu o Projeto Atlas, ou simplesmente Atlas, que foi o primeiro supercom-

putador da década de 60, onde utilizava-se o conceito de multiprogramação, compartilhamento de tempo e periférico. Este projeto contou com uma parceria entre o Departamento de Energia Elétrica da Universidade de Manchester, o MIT (*Massachusetts Institute of Technology*) e a Ferranti Limited, e criou o computador mais rápido da década [35]. Com os conceitos utilizados aqui é considerado o nascimento do *hypervisor*, ou VMM (*Virtual Machine Monitor* ou Monitor de Máquina Virtual) [34].

Durante os anos seguintes, com o baixo uso de *mainframes*, equipamentos de grande porte com alta capacidade de processamento, os foram projetos descontinuados e perderam forma.

No final da década de 1990, a virtualização começou a alavancar e, desta vez, sendo um dos temas mais discutidos e estudados ano após ano. Grandes empresas visualizaram um mercado promissor e iniciaram o estudo e desenvolvimento de ferramentas *hypervisor*.

3.3 *Hypervisor*

As ferramentas monitoras de máquinas virtuais, ou *hypervisor*, são parte fundamental na criação de um ambiente virtualizado, a função dele é criar uma camada de *hardware* virtualizada e comunicar-se em tempo real com o *hardware* físico e o sistema operacional virtualizado. Com o *hypervisor*, é possível executar múltiplos sistemas operacionais simultaneamente em apenas um hospedeiro.

Entre os serviços mais simples prestados pelo *hypervisor*, está a possibilidade do uso de uma máquina virtual, podendo de forma fácil, conceder, criar, editar e remover as mesmas. Algumas ferramentas possibilitam a visualização da performance e que o usuário tenha acesso a estatística da utilização dos recursos.

Uma das ferramentas que será descrita posteriormente, está representada no esquema de funcionamento do *hypervisor* da figura 3.1. Na área indicada pelo cor verde, a parte inferior faz referência ao *kernel* do GNU/Linux, que possibilita o gerenciamento da virtualização do hardware de tecnologias Intel VT (Intel *Virtualization Technology*) ou AMD-V (*Advanced Micro Devices Virtualization*) e exporta o resultado no arquivo `/dev/kvm`, para que sistemas virtualizados possam mapear os recursos. Na outra parte, temos o libvirt e qemu, que trabalham para fornecer a parte da emulação do *hardware*.

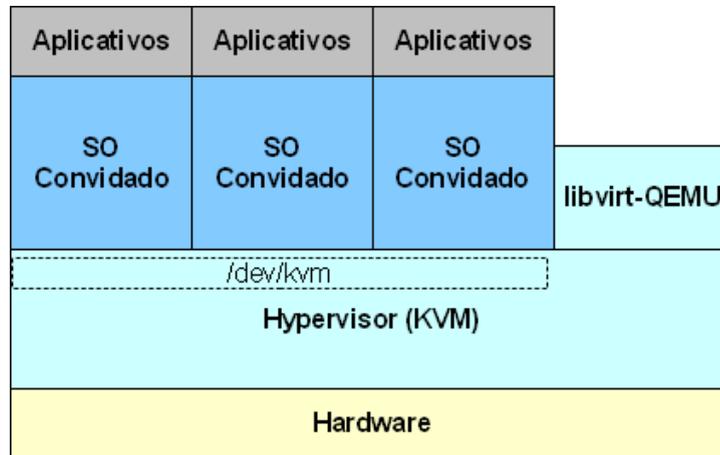


Figura 3.1: Arquitetura do *hypervisor* KVM (adaptado de [4]).

3.4 Técnicas de virtualização

Há disponíveis muitas técnicas de virtualização, porém, os autores Tim Jones [36] e Shannon Meier [16], dividem a taxonomia em quatro, que possuem maior relevância e popularidade. (1) A emulação de *hardware*, considerado o método mais complexo; (2) a virtualização completa, onde mediação é a palavra-chave; (3) a paravirtualização, uma técnica parecida com a virtualização completa, porém exigindo modificação do sistema operacional convidado; (4) e por fim, a virtualização em nível de sistema, que mostra uma maneira totalmente diferente em relação as outras. Nas seções 3.4.1, 3.4.2, 3.4.3 e 3.4.4, descreve-se detalhadamente cada uma.

3.4.1 Emulação de *hardware*

Na emulação de *hardware*, o *hypervisor* trabalha sobre uma camada de *hardware* emulado, como o próprio nome sugere. Todo este *hardware* emulado é criado via software no sistema operacional hospedeiro, geralmente feito em assembly, até mesmo o processador da máquina virtual é criado [34].

Este método é considerado o mais complexo das virtualizações. O *hardware* emulado é normalmente referido como um *hypervisor*. Na figura 3.2 é mostrado que cada sistema operacional convidado executa sobre o *hardware* emulado desejado [5].

O *hypervisor* fornece uma padronização do *hardware*, onde o sistema operacional convidado reside e interage. Esta consistência possibilita a migração de uma máquina para outra, mesmo

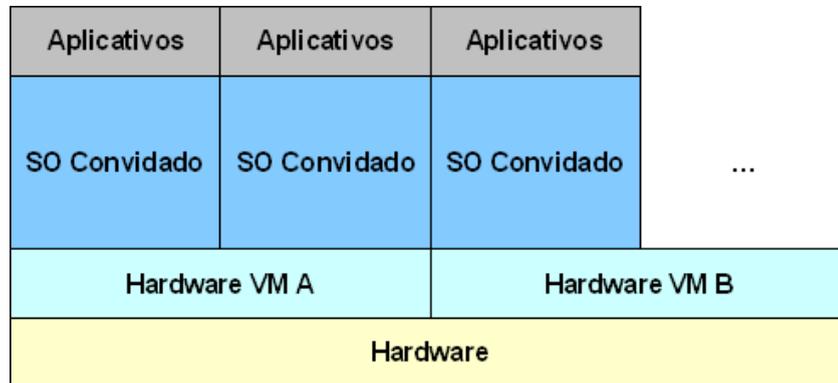


Figura 3.2: Técnica da emulação de *hardware* (adaptado de [5]).

que as máquinas físicas sejam diferentes, pois o *hypervisor* traduz as chamadas do *hardware* emulado para recursos específicos da máquina física [37].

Entre as aplicações comuns da emulação de *hardware*, temos o desenvolvimento de um *firmware* de *hardware*, com isso, pode-se validar muitos aspectos de um *hardware* emulado, simulando outro que ainda não esteja disponível [5].

A grande vantagem da emulação de *hardware* é justamente executar um sistema operacional sem necessidade de modificação do mesmo.

Existem algumas desvantagens na emulação de *hardware*, entre elas, podemos citar o desempenho inferior, levando em consideração que a virtualização prejudica o desempenho, ou seja, aplicações geralmente têm um desempenho superior em sistemas não virtualizados. Outra problemática é em relação à instalação de um *hardware* novo. Como o *hypervisor* faz a comunicação com a máquina física e recursos reais da máquina, ao instalar um dispositivo novo seria necessário o *driver* para uso do mesmo, porém, no *hypervisor*, não existe suporte à instalação de novos *drivers*. Isso pode causar problemas em organizações que querem usufruir de novas tecnologias de *hardware* [37].

Duas grandes empresas fornecem suporte à virtualização utilizando emulação de *hardware*: a VMware, com os seus produtos VMware Server e ESX Server; e a Microsoft, que oferece um produto chamado Virtual Server.

3.4.2 Virtualização completa

A virtualização completa (*Full virtualization*), ou virtualização nativa, como também é conhecida, é um modelo onde o *hypervisor* executa sobre o *hardware* puro. Este *hypervisor* intermedia a comunicação entre o sistema operacional convidado da máquina virtual a ser instalada e o *hardware* da máquina física, traduzindo todas as instruções do sistema operacional em tempo real. Mediação é a palavra-chave deste modelo [35]. O esquema de funcionamento deste modelo é mostrado na figura 3.3, onde temos vários sistemas operacionais convidados executando em um *hypervisor*.

Em relação ao desempenho, a virtualização completa se sobressai se comparada à emulação do *hardware*, porém é inferior ao sistema operacional executando nativamente, isto se deve ao fato da mediação do *hypervisor* [6].

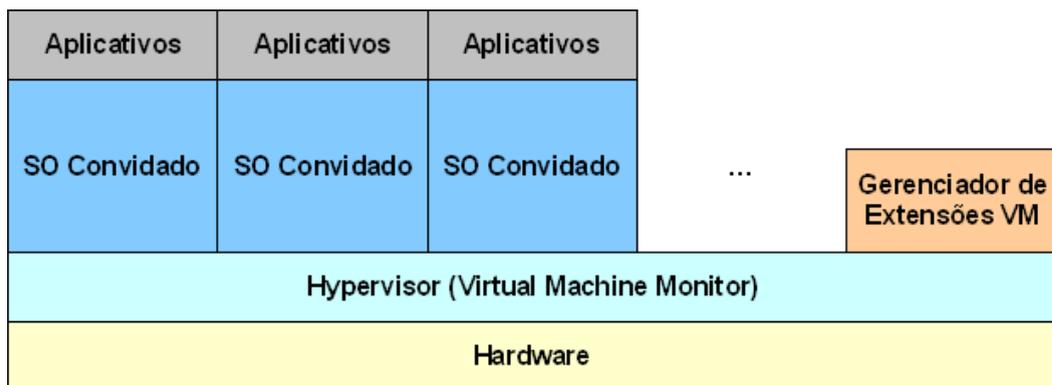


Figura 3.3: Técnica da virtualização completa (adaptado de [6]).

Entre as vantagens do modelo, podemos destacar a flexibilidade, que possibilita execução de diferentes sistemas operacionais de diferentes fornecedores, melhor isolamento, segurança para as máquinas virtuais, simplificação da migração e portabilidade, e principalmente o fato do sistema operacional poder executar sem modificações. Sua principal desvantagem diz respeito a uma mesma máquina virtual não executar em arquiteturas diferentes [7].

Alguns obstáculos surgiram devido a problemas gerados com alguns tipos de *hardwares*. Os problemas ocorriam na criação da camada de interpretação de instruções dos processadores da plataforma x86, onde 17 instruções específicas criaram problemas quando virtualizadas, fazendo o sistema operacional convidado gerar erros durante a execução, podendo até finalizar a aplicação de forma inesperada. Para solucionar estes problemas, a VMware foi a pioneira ao

desenvolver uma técnica que converte as instruções quando geradas e as converte em instruções seguras, possibilitando a execução sem intervenções [38].

Intel e AMD (*Advanced Micro Devices*) oferecem suporte para virtualização através de processadores com tecnologia Intel VT e AMD-V.

3.4.3 Paravirtualização

Uma técnica similar a virtualização completa e bastante popular é a paravirtualização. O método utiliza um *hypervisor* compartilhado que fica logo acima da camada de *hardware* puro, porém, fazendo com que o sistema operacional convidado tenha conhecimento do código de virtualização [16]. Basicamente esquematizado na figura 3.4, a paravirtualização exige que todos os sistemas operacionais convidados sejam modificados, caso isso não ocorra, não será possível a execução do sistema operacional sobre o *hypervisor*.

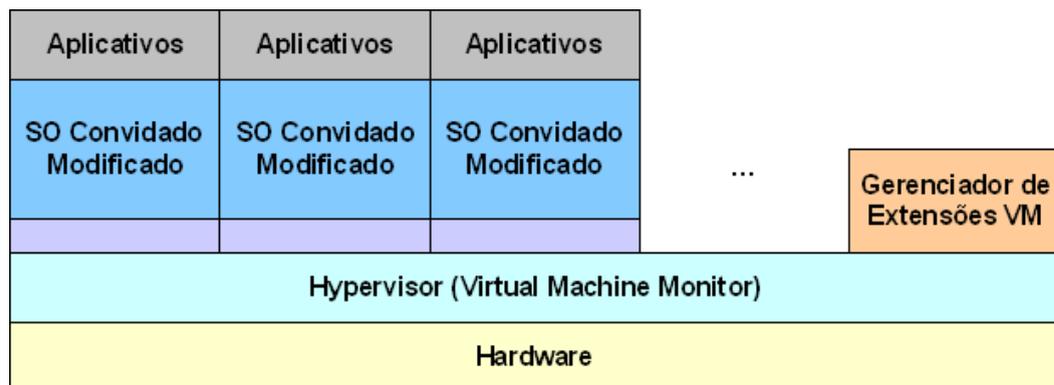


Figura 3.4: Técnica da paravirtualização (adaptado de [6]).

Com a paravirtualização, é possível que diferentes sistemas operacionais rodem simultaneamente no mesmo *hypervisor*. Por exemplo, pode-se instalar um sistema operacional convidado GNU/Linux e outro Microsoft Windows 7.

A principal desvantagem é justamente a exigência da modificação do sistema operacional convidado, já que alguns destes sistemas não incorporam essas mudanças.

Seu desempenho é bastante semelhante à virtualização completa, ficando próximo de um sistema operacional não virtualizado.

Um modelo de arquitetura que utiliza este método é a arquitetura Power, onde foram feitas mudanças específicas para o AIX, SUSE Linux Enterprise Server e Red Hat Enterprise Linux.

No que diz respeito ao GNU/Linux, foi aprimorado o código paravirt-ops, ou pv-ops, que foi criado inicialmente pela Xen. O aprimoramento foi incluído no *kernel* padrão do GNU/Linux a partir da versão 2.6.23, proporcionando o uso do sistema operacional em sistemas paravirtualizados [39].

3.4.4 Virtualização em nível de sistema

Esta última técnica apresenta uma grande diferença dentre as demais. A virtualização em nível de sistema possui um sistema operacional logo acima da camada de *hardware*, e os sistemas virtualizados ficam acima deste sistema operacional, como esquematizado na figura 3.5. Este método possibilita que apenas um único sistema operacional disponibilize suporte a todos os sistemas convidados [6].



Figura 3.5: Técnica da virtualização em nível de sistema (adaptado de [6]).

Em relação ao desempenho, ele possui um ótimo resultado, já que é considerado nativo. Assim como a técnica de paravirtualização, a desvantagem da virtualização em nível de sistema é a exigência do sistema operacional modificado, porém, existe um agravante de haver apenas um sistema operacional entre o *hardware* e os sistemas virtualizados [6]. Este último é o principal problema, já que todos sistemas virtualizados estão executando sobre apenas um sistema operacional, caso houver algum problema durante a execução do sistema operacional, todos os sistemas virtualizados deixarão de funcionar.

3.5 Ferramentas para virtualização

Muitos foram os motivos para inspirar grandes empresas desenvolvedoras de TI a estudarem e desenvolverem a virtualização, dentre eles podemos destacar o melhor gerenciamento do

datacenter e a diminuição de máquinas físicas. Desde o princípio, na década 1960, o termo de virtualização chamou a atenção, porém, somente nos últimos anos que realmente os CIOs (*Chief Information Officer* ou Diretor de Informática) estão utilizando efetivamente esta tecnologia.

O mercado possui diversas soluções e ferramentas para virtualização, que diferenciam entre proprietárias e livres. Nas subseções a seguir, descreve-se sucintamente as principais, dentre elas o KVM, que será o *hypervisor* utilizado no desenvolvimento deste trabalho. Na seção 3.6, serão apresentados as motivações para o uso da virtualização e os inúmeros benefícios proporcionados pela mesma.

3.5.1 VMware

A empresa VMware Inc. desenvolve o mais popular *hypervisor* que existe no mercado. Seus produtos diferenciam-se entre pagos e gratuitos e possibilitam a virtualização de um simples *desktop* até grandes *datacenters*. Além disso ela foi a desenvolvedora do primeiro sistema operacional para computação na nuvem, o VMware vSphere, que estende os limites para expansão de máquinas virtuais e servidores. A seguir tem-se a descrição dos principais produtos que possibilitam a virtualização.

- VMware Workstation

Seu uso é direcionado a *desktops* e necessita um sistema operacional hospedeiro para funcionamento. Para utilizá-la, é necessário adquirir uma licença, e a última versão lançada, a 7.1, suporta até 8 processadores virtuais e discos virtuais de 2 TB. Também proporcionou uma melhora nos recursos gráficos, utilizando OpenGL 2.1 e DirectX 9.0, o que faz esta versão ser até duas vezes mais rápida que a versão anterior na instalação dos sistemas operacionais Windows Vista e Windows 7. Tem a performance otimizada através dos processadores Intel Core i3, i5 e i7 [40].

- VMware Player

É a versão gratuita para executar máquinas virtuais já criadas, direcionada para *desktops*. Essa ferramenta também necessita de um sistema operacional hospedeiro. Como não é possível criar máquinas virtuais nela, existe a possibilidade de utilizar a criação através de ferramentas não oficiais, por exemplo através do site EasyMX ¹ [41].

¹<http://www.easyvmx.com/>

- VMware Fusion

Uma solução para a virtualização em sistemas operacionais são os hospedeiros Mac, que utilizam o processador Intel. A última versão deste produto é a 3.1, que apresenta um desempenho 35% superior a versão anterior. Com ela, pode-se instalar uma máquina virtual com o sistema operacional convidado Windows [42].

- VMware Server

Seu uso é voltado para servidores e necessita de um sistema operacional hospedeiro. Não é necessário adquirir licença, pois ele tornou-se gratuito a partir do ano de 2006. Possui recursos do VMware Workstation, porém utiliza recursos voltados para servidores [43]. Na figura 3.6 é mostrada a arquitetura do VMware Server utilizando um sistema operacional hospedeiro (GNU/Linux ou Windows) e a utilização de várias máquinas virtuais com sistemas operacionais convidados, tudo isso em apenas uma máquina física.



Figura 3.6: Modelo da arquitetura do VMware Server [7].

- VMware ESX Server

Com a primeira versão disponibilizada em 2001, é considerado um produto robusto, seguro e completo, voltado para servidores de grande porte. Para sua utilização é necessário o pagamento de licença. Na figura 3.7 podemos ver a arquitetura do VMware ESX Server, e que o mesmo não necessita um sistema hospedeiro, pois ele mesmo comunica-se com o *hardware*, conseguindo desta maneira, altos níveis de desempenho, robustez e escalabilidade [44].

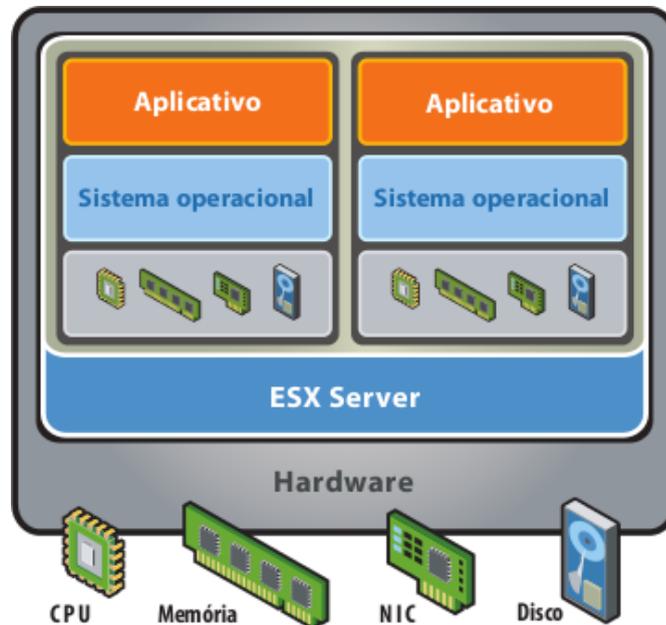


Figura 3.7: Modelo da arquitetura do VMware Server ESX [8].

- VMware ESXi Server

Ferramenta que irá substituir o VMware ESX Server, segundo muitos especialistas, devido ao fato do VMware vSphere dar suporte apenas a arquitetura ESXi, após a versão 4.1. Através de uma visão superficial, as funcionalidades continuam as mesmas da arquitetura ESX, porém na ESXi o console de serviço, um sistema operacional GNU/Linux que era utilizado para executar localmente o gerenciamento de tarefas, foi retirado da arquitetura, diminuindo o espaço do código básico utilizado no ESX de 2 GB para apenas 100 MB da ESXi. Para utilizar essa ferramenta é necessário o pagamento da licença, assim como VMware ESX Server [45].

Além destes produtos, a VMware Inc. possui diversos produtos que auxiliam no gerenciamento, por exemplo, para várias máquinas virtuais poderem acessar o mesmo meio de compartilhamento e efetuar migrações em tempo de execução. Esta última é feita através do VMware vMotion.

3.5.2 Windows Virtual PC

O Windows Virtual PC é um *hypervisor* bastante simples. Ele está disponível apenas para sistemas operacionais Windows 7 e é sucessor das ferramentas Microsoft Virtual PC. Apesar

da maioria das ferramentas disponibilizadas pela Microsoft terem custo, esta pode ser utilizada sem nenhum desembolso com licenças.

Esta nova versão possui alguns avanços, como a possibilidade de utilizar altas resoluções, integração entre os sistemas operacionais instalados (hospedeiro e convidado), suporte a *multi threads* e suporte a USB (*Universal Serial Bus*) [46].

3.5.3 VirtualBox

Criada inicialmente pela empresa Innotek, possuía uma licença proprietária. No ano de 2007 o VirtualBox tornou-se uma solução disponível sobre os termos da GNU GPL (GNU *General Public License* ou GNU Licença Pública Geral), e no ano seguinte foi adquirida pela Sun Microsystems. A Oracle é a atual proprietária desta ferramenta, pois em abril de 2009 adquiriu a Sun Microsystems [47]. Apesar de ser mais utilizado em *desktops*, o VirtualBox também pode ser utilizado em servidores.

O VirtualBox utiliza a técnica de virtualização completa, eliminando a necessidade de modificação dos sistemas operacionais convidados. Uma vantagem desta ferramenta, é executar em sistemas hospedeiros Windows, GNU/Linux, Solaris e Mac OS X, suportando também todos esses sistemas operacionais como convidados.

Todos os discos rígidos virtuais são emulados através do VDI (*Virtual Disk Image*), e o *hypervisor* possui a disponibilidade de utilizar discos virtuais através da tecnologia iSCSI (*Internet Small Computer System Interface*). A memória do adaptador gráfico pode ser ajustada conforme a necessidade do usuário, e existe a possibilidade de instalação de *drivers* para sistemas operacionais convidados Windows e GNU/Linux. Também disponibiliza a virtualização de dispositivos de som e USB [48].

Algumas características são próprias desta ferramenta, entre elas, permitir que dentro de um sistema operacional convidado seja instalado o VirtualBox e também outro sistema convidado, controle total através de linha de comando, suporte às tecnologias Intel VT e AMD-V e um servidor RDP (Microsoft *Remote Desktop Protocol*) [49].

3.5.4 Xen

O Xen é um *hypervisor* de código aberto, desenvolvido para processadores de 32 e 64 bits. Ele utiliza a técnica de paravirtualização e possibilita executar vários sistemas operacionais de forma segura e eficiente, com desempenho próximo ao nativo. Estes sistemas operacionais devem obrigatoriamente ser modificados para suportar a virtualização. Como exemplo temos o Red Hat e o Debian.

Sua história iniciou nos laboratórios de computação da Universidade de Cambridge, com o objetivo de prover uma infraestrutura pública para computação distribuída. O Projeto XenServers, como foi denominado na época, foi liderado pelo pesquisador Ian Pratt, o qual, posteriormente, fundou a empresa XenSource Inc. que mantém o projeto com o nome Xen [49]. No ano de 2007, a XenSource Inc. foi adquirida pela Citrix Systems Inc. que mantém versões *open source* e empresarial. Dentre os parceiros que mais apoiaram o desenvolvimento dessa ferramenta, estão a IBM (*International Business Machines*) e a Intel [34].

Entre as características deste *hypervisor*, estão o desempenho próximo do nativo, principalmente quando o processador possui a tecnologia de virtualização, suporte a todos os *hardwares* com *drivers* disponíveis para GNU/Linux, migração em tempo de execução com zero *downtime* e suporte aos processadores Intel VT e AMD-V [34].

Com as ferramentas empresariais, é possível obter o máximo do *hypervisor* Xen, como migrar as máquinas virtuais entre servidores ou *datacenters* e obter informações sobre o desempenho dos mesmos.

3.5.5 KVM

O *hypervisor* KVM é uma ferramenta que utiliza a técnica de virtualização completa e possibilita a virtualização, tendo como sistema operacional hospedeiro o GNU/Linux. Sua primeira versão foi disponibilizada no *kernel* do GNU/Linux a partir da versão 2.6.20, de fevereiro de 2007, o que torna-o um módulo carregável do *kernel* [50].

Uma alternativa bastante interessante para quem busca ferramentas de virtualização *open source*, o KVM tem como principal objetivo criar um *hypervisor* moderno que leve em consideração, principalmente, a experiência de outras ferramentas e as tecnologias de *hardware* atuais [51]. Para utilizar de maneira otimizada os recursos da virtualização e evitar a modificação do

sistema operacional convidado, são necessários processadores que possuam as tecnologias de virtualização acopladas, como Intel VT e AMD-V.

Com a inclusão da ferramenta no kernel do GNU/Linux, também é possível a utilização de todas as implementações e benefícios disponíveis nele. Um exemplo geral é o gerenciador de memória do GNU/Linux, que é considerado consolidado e possibilita suporte NUMA (*Non-Uniform Memory Access*), utilizado em sistemas multiprocessados. Este é um dos objetivos da equipe de desenvolvimento do KVM: sempre fazer uso da tática de reusabilidade, evitando ao máximo recriar algo. O *hypervisor* Xen não utiliza deste pensamento e constrói para si um gerenciador de memória a partir do zero [51]. Os avanços foram reconhecidos, obtendo apoio de grandes empresas como AMD, HP (*Hewlett-Packard*), IBM, Intel, Novell e Red Hat.

Existem dois modos de execução padrões no GNU/Linux, o usuário e o *kernel*. A utilização do KVM acrescenta um modo de execução chamado modo visitante. Seu funcionamento baseia-se na não execução de entradas/saídas do código convidado. No modo *kernel* muda para o modo convidado e gerencia todas suas entradas/saídas e instruções especiais. Por fim o modo usuário executa as entradas/saídas em nome do hospedeiro [9].

Na estrutura do KVM, existem dois componentes, um *driver* para gerenciar o *hardware* de virtualização e um componente especial do usuário para emular o *hardware* hospedeiro. Na figura 3.8, onde é apresentada a arquitetura base do KVM no *kernel*, o primeiro componente, o *driver*, expõe a capacidade através de um dispositivo localizado em `/dev/kvm`. Já o segundo, conta com o auxílio do *qemu*, um emulador de *hardware*, virtualizador genérico e *open source*.

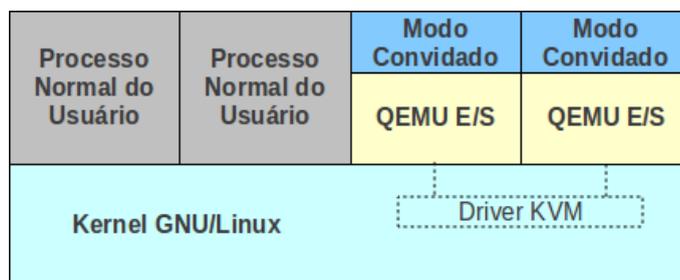


Figura 3.8: Base da arquitetura KVM (adaptado de [9]).

A tecnologia de migração de máquinas virtuais está disponível tanto em modo *live*, em tempo de execução, como também *offline*. Após o processo de migração entre duas máquinas físicas, a máquina virtual continua sendo executada em seu novo hospedeiro, sem grandes danos

em relação ao tempo de parada de serviço.

Esta será a ferramenta utilizada como *hypervisor* para desenvolvimento do trabalho. Entre alguns exemplos de sistemas operacionais convidados que o KVM suporta desenvolvidos pela Microsoft: Windows 2008, Windows 7, Windows Vista, Windows XP etc.; nas distribuições GNU/Linux: CentOS 5.5, Fedora 13, Red Hat Linux, Debian, Ubuntu, SUSE Linux Enterprise Server 11, Android 2.2, openSUSE, Mandriva etc. [52].

3.5.6 OpenVZ

OpenVZ é uma ferramenta de virtualização desenvolvida pela Swsoft, que se baseia em uma versão modificada do kernel do GNU/Linux. Ela utiliza a técnica de virtualização em nível do sistema. Um servidor físico com OpenVZ possibilita a execução de vários VPS (*Virtual Private Servers*). Esta ferramenta de virtualização possui a desvantagem de obrigar que o sistema operacional convidado seja GNU/Linux [53].

Em relação as vantagens da ferramenta, está a promessa de uma performance semelhante ao nativo e a compatibilidade com Parallels Virtuozzo Containers, um produto produzido em larga escala pela Parallels Inc, voltado para virtualização em nível do sistema [54].

Uma característica que o OpenVZ possui, é a migração em tempo de execução. Nela, o servidor virtual é “congelado” e salvo em um arquivo do disco, podendo ser migrado para qualquer outra máquina física que possua o OpenVZ, onde o servidor virtual muda o estado para “descongelado”. O tempo, desde o início até o final da migração, dura alguns segundos e o servidor restabelece o serviço normalmente [55].

3.6 Benefícios da virtualização

A virtualização proporciona muitos benefícios, levando principalmente em consideração, a diminuição dos custos. Organizações de todos os tamanhos buscam também, através da virtualização, obter um controle maior sobre a infraestrutura, podendo desta maneira, gerenciar os recursos de modo flexível e aumentar a independência da localização em uma máquina física.

Com a utilização da virtualização, pode-se unir ferramentas e mesclar tecnologias, permitindo uma gestão dos recursos distribuídos heterogeneamente. Esta liberdade proporciona um aumento na flexibilidade, permitindo a administração em conjunto, evitando o gerenciamento

de cada computador como um recurso separado. Um *datacenter*, contendo armazenamento e servidores virtualizados, possibilita o funcionamento de forma segura através do *hardware* compartilhado [16].

No interior de São Paulo, a Univen Petróleo tem se beneficiado com a virtualização da sua infraestrutura de TI. Diógenes Novaes, gerente de TI da Univen, revela que a infraestrutura era composta por 20 servidores, os quais suportavam todas as operações da empresa, e após a virtualização todas as operações foram mantidas em apenas 3 servidores físicos. “*O projeto de virtualização proporcionou não somente a otimização do espaço físico que os servidores ocupavam mas também a economia de 60% em energia elétrica despendida nas máquinas e no sistema de refrigeração de todo os sistema corporativo de TI da companhia*”, revela Novaes [56].

Segundo a empresa Peak 10 Data Center Solutions [57], os benefícios que a virtualização de servidores pode oferecer está dividida em seis principais grupos. Abaixo vamos descrever cada um dos grupos.

3.6.1 Consolidação e otimização de recursos

Um dos principais benefícios da virtualização é a consolidação dos servidores, que foi contextualizado na seção 2.2.5. Alguns relatórios expõem que a maioria dos servidores x86 são utilizados em menos de 10% em relação ao uso de CPU (*Central Processing Unit* ou Unidade Central de Processamento) ou processador [57]. Com a utilização da consolidação, o gerenciamento dos recursos de *hardware* tornam-se otimizados, elevando a utilização do processador de uma máquina física.

Segundo Meier [16], “*Consolidando a infraestrutura pode-se aumentar as taxas de utilização entre 5% e 10% para mais de 70%, contribuindo na melhora do ROI (Return On Investment ou Retorno Sobre Investimento)*”. O aumento da taxa de utilização em servidores virtualizados demonstram o quanto os recursos são otimizados com o uso da consolidação, desta maneira, diminuindo consideravelmente os custos.

3.6.2 Aumento da flexibilidade

Nenhuma solução torna um servidor tão independente do *hardware* da máquina física quanto a virtualização. Esta independência é um aspecto importante quando é necessário efetuar a expansão do servidor, que é um dos principais inimigos dos administradores de TI. Com a independência do *hardware* da máquina física, tem-se uma infraestrutura simplificada, o que ajuda também, na redução dos custos com o gerenciamento.

O nível de serviço tem uma melhora através da virtualização, pois com ela, agrupamos em apenas uma máquina vários servidores com diferentes sistemas operacionais. Máquinas agrupadas facilitam correção e atualização de *firmware* dos servidores, e também permitem uma transição suave para novas gerações de *hardwares* [57].

Em *datacenters* virtualizados, os servidores que irão sofrer um *upgrade*, que é a atualização para nova geração de *hardware*, tornam-se mais rápidos e simples efetuar o *backup* ou restauração dos servidores. Para que o *upgrade* faça uso destes benefícios, o *hardware* novo deve ser compatível com o antigo. Este processo demorava dias antes da virtualização. Isso deve-se ao fato de que qualquer mudança de *hardware*, por mais simples que fosse, o servidor sofreria atualização de *drivers* ou até mesmo a reinstalação do sistema operacional [57].

Outra ferramenta bastante utilizada na virtualização, é a clonagem de um servidor. Com ela, podemos fazer cópias de um servidor enquanto o mesmo está em execução, sem nenhuma necessidade de desligamento ou parada de serviço, de forma bastante flexível.

3.6.3 Melhoria no custo total de propriedade

Com um melhor gerenciamento da infraestrutura, o ROI se torna mais rápido. Uma infraestrutura compartilhada, como a virtualização proporciona, diminui desperdícios de processamento e conseqüentemente o TCO (*Total Cost of Ownership* ou Custo Total de Posse) [58].

O aumento de flexibilidade está diretamente ligado à virtualização dos servidores, e através desta virtualização, o número de máquinas físicas é reduzido consideravelmente. A redução no número de máquinas físicas tem um efeito primordial na diminuição de custos, pois os gastos com resfriamento para os servidores se tornam menores, reduzindo também os custos com energia.

3.6.4 Facilidade no gerenciamento

Os servidores de um *datacenter* virtualizado possuem o acesso e administração a todos os servidores de modo facilitado. Todos os operadores dos servidores ou consultores técnicos podem, de uma forma cômoda, efetuar todo o gerenciamento através de uma única visão, por exemplo, utilizando apenas um monitor, teclado e mouse.

A eficiência proporcionada pela virtualização é notada quando existe a necessidade de aplicação de *patches*, que são correções de erros, atualizações no sistema ou ajustes na política de segurança. Uma mudança na política de segurança poderia distribuir de forma muito mais eficiente para todo o *datacenter* [57].

Com a facilidade e flexibilidade oferecida através da virtualização, o tempo gasto com gerenciamento dos servidores fica na relação de dias por minutos. Os desenvolvedores conseguem aumentar a velocidade de aplicação de testes e implantação, melhorando assim a infraestrutura.

3.6.5 Aumento do tempo de atividade e continuidade do serviço

Em muitas organizações, um *downtime* ou parada de serviço, seria catastrófico. Por exemplo, em uma instituição bancária, qualquer quantidade de tempo que um servidor fique fora do ar pode parar muitos processos. Nesta ideia, a virtualização tem uma grande importância, pois com ela consegue-se diminuir as paradas de serviço, migrando o servidor para uma máquina física alternativa, em caso de uma falha de *hardware*, por exemplo.

Métodos de recuperação de problemas e continuidade do serviço acabam tendo um tempo e custo muito elevados. Com o uso da virtualização, a migração de uma máquina virtual pode ocorrer em um espaço de tempo muito curto, principalmente se a máquina virtual possui apenas processamento das informações. Neste caso, o armazenamento das informações seria baseado na utilização de uma SAN (*Storage Area Network* ou Área de Armazenamento em Rede).

Outra forma de armazenamento, seria utilizar a máquina virtual, porém o tempo de migração acaba ficando maior do que o modelo anterior. Por exemplo, em um *datacenter* que possui uma máquina física que está com um problema específico em um *hardware* que necessita a parada para troca ou manutenção. Neste caso, as máquinas virtuais que estão executando sobre essa máquina física são migradas para outro servidor físico, e a continuidade do serviço se torna transparente para o usuário final, ao término da manutenção é feita nova migração, na qual a

máquina virtual volta para sua posição de origem.

3.6.6 Redução de custos e energia

Com a utilização da virtualização, os benefícios tanto são da parte técnica quanto da financeira. Com o uso da virtualização e a consolidação dos servidores, um *datacenter* pode ser considerado mais “verde”. Este fato se deve a grande redução no número de máquinas físicas e arrefecimento. A redução de custo de manutenção pela metade é considerada normal, podendo chegar até 80%, o que não é inaudito [57].

Por exemplo, em um *datacenter* com 120 servidores, se não houver a virtualização, são necessárias 120 máquinas físicas. Com a virtualização, estes 120 servidores poderiam estar em 20 máquinas físicas. As 20 máquinas físicas teriam um processamento mais ativo que as 120 iniciais, e mantendo a mesma qualidade de serviço, com isso há uma redução considerável em gastos com energia elétrica.

Capítulo 4

Implementação

Para implementar a ferramenta de consolidação é necessário que muitas características sejam definidas. Todo o processo foi estudado buscando as melhores ferramentas disponíveis para tornar o ambiente simulado o mais real possível. Em todo o desenvolvimento sempre optou-se pela utilização de software livre, desde a instalação do sistema operacional GNU/Linux, até o programa responsável para geração de carga nas máquinas virtuais.

O primeiro passo para a preparação do ambiente é a instalação de um sistema operacional com suporte a SSH (*Secure Shell*), e em seguida obter o uso do processador da máquina física. Duas ferramentas surgiram como opção para obtenção destes dados, o Ganglia¹ e o Nagios². Ambas são desenvolvidas em software livre e utilizadas em grandes organizações para monitoramento da infraestrutura.

Após alguns testes e estudos sobre cada ferramenta, optou-se pelo Nagios, devido a grande quantidade de *plugins* e extensões já existentes, tornando a resolução do trabalho mais objetiva. Todas as informações a respeito da configuração do Nagios estão descritas no apêndice A para servidores e no apêndice B para clientes remotos.

Basicamente toda implementação baseia-se em três fases: a infraestrutura, que se preocupa com a montagem da infraestrutura, com três computadores utilizados para simular uma aplicação real; criação e definição das máquinas virtuais, onde seis servidores são criados através de máquinas virtuais; e implementação da consolidação de servidores, que faz o gerenciamento das seis máquinas virtuais em duas máquinas físicas, levando em consideração o cenário. Nas seções 4.1, 4.2 e 4.3, estão descritos todos os procedimentos para implementação do trabalho.

¹<http://ganglia.sourceforge.net/>

²<http://www.nagios.org/>

4.1 Primeira fase: montagem da infraestrutura

Para a montagem da infraestrutura levou-se em conta alguns critérios que seriam necessários para seu desenvolvimento. Inicialmente foram disponibilizadas três máquinas físicas, onde duas tinham o papel de servidores de máquinas virtuais e uma como servidor de armazenamento (*storage*). Estas máquinas serviram apenas para testes, sem tecnologias de virtualização, como AMD-V e Intel VT, e com baixo poder de processamento e memória, até o providenciamento de máquinas mais robustas para manipulação das máquinas virtuais.

O primeiro passo para preparar a infraestrutura de simulação do ambiente real, foi a instalação do sistema operacional, num primeiro momento optado pelo GNU/Linux Ubuntu 10.10 versão 32bits.

Após as três máquinas estarem com seus sistemas operacionais instalados, começou-se a instalação do protocolo NFS (*Network File System*), um sistema de arquivos distribuídos. O NFS é instalado e configurado no servidor de armazenamento (*storage*), onde define-se um diretório que será compartilhado, criando uma unidade de armazenamento na rede para os servidores de máquina virtual. Esta unidade de armazenamento na rede para o servidor de máquinas virtuais se torna transparente, pois é muito semelhante às unidades de armazenamento físico, exceto na questão de desempenho para entrada/saída. O apêndice C contém todo o processo de instalação do servidor e dos clientes remotos do NFS.

Com o ambiente entre os servidores já criado, foi necessária a instalação e configuração da ferramenta que monitora os recursos de hardware utilizados. Este monitoramento poderia ser feito através do comando “top” do GNU/Linux, porém desde o início buscou-se a simulação mais próxima da realidade de grandes organizações, e ferramentas como Nagios, Ganglia, Cacti, Zabbix, Zenoss, etc; são indispensáveis na infraestrutura. A instalação da ferramenta escolhida, o Nagios (versão 3.2.3), foi feita em conjunto com o Nagios *Plugins* (versão 1.4.15) e um *plugin* conhecido como “check_cpu”, desenvolvido em Shell por Mike Adolphs³. Atuando como um agente nos servidores de máquina virtual foi instalado o Nagios NRPE (Nagios *Remote Plugin Executor*) (versão 2.12), o qual tem a função de retornar ao servidor onde está instalado o Nagios, as informações a respeito da máquina.

Todos os dados coletados através do Nagios, são armazenados em uma base de dados

³<http://www.matejunkie.com/cpu-check-plugin-for-nagios/>

MySQL através do complemento NDOUtils (versão 1.4b9). Com as informações armazenadas em uma base de dados, torna-se mais fácil a recuperação de dados para a implementação. Na figura 4.1 temos a infraestrutura de simulação detalhada, formada por três máquinas físicas. O servidor de armazenamento também hospeda o servidor Nagios de monitoramento, a base de dados MySQL e a implementação desenvolvida em Java e Shell Script. Ambos os servidores de máquinas virtuais possuem ligação com a Internet, e as máquinas virtuais utilizam a mesma interface para ligação com a internet e também com o *storage*. A ligação com a internet é feita através de um *switch* com velocidade máxima de transferência de dados de 100 Mbit/s.

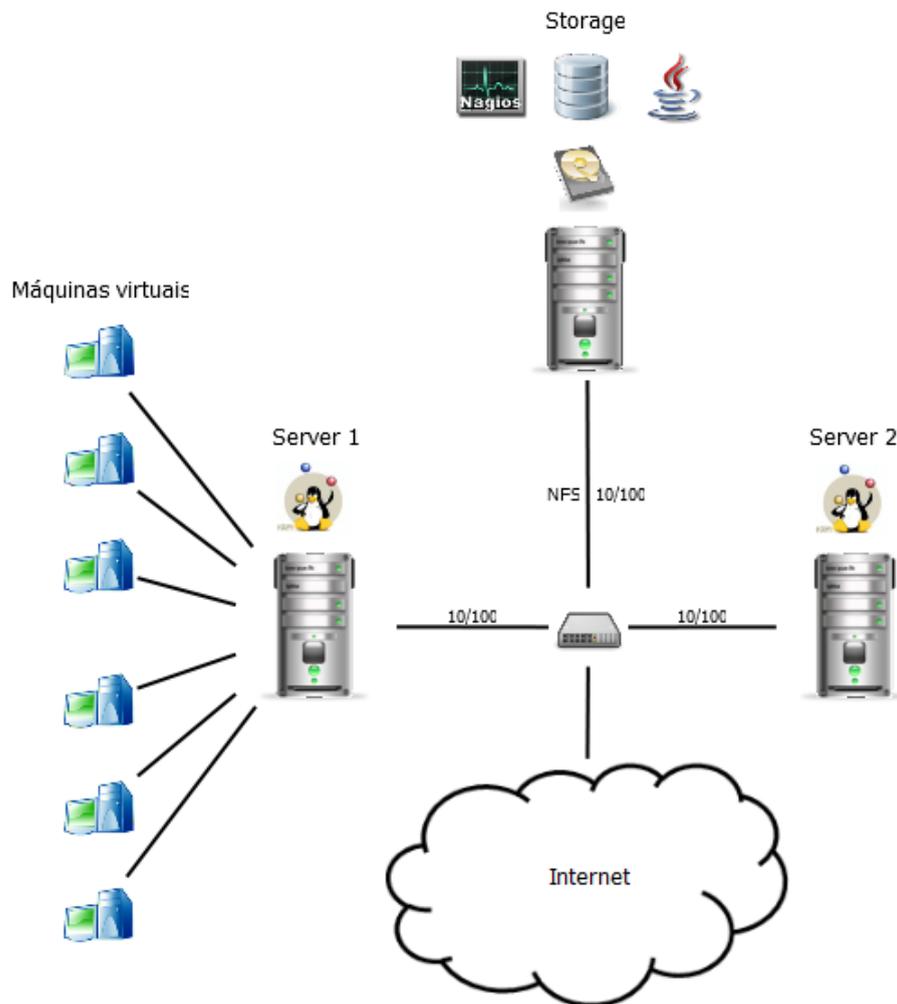


Figura 4.1: Modelo da infraestrutura de simulação.

Posteriormente, novas máquinas físicas foram conseguidas para a sequência do trabalho acarretando na necessidade de preparação dos novos computadores. Quando houve a reinsta-

lação da infraestrutura, optou-se pelo sistema operacional GNU/Linux Debian Squeeze 64bits (versão 6.0.2), pois o mesmo é utilizado em grande parte dos servidores de organizações, por ser um sistema operacional estável e com bom desempenho. As três máquinas preparadas para a continuidade do trabalho possuem a mesma configuração, consistindo de um processador AMD Athlon 64 X2 5000+, com 2GB de memória RAM e um disco rígido Samsung HD161GJ 160G SATAII.

4.2 Segunda fase: criação e definição das máquinas virtuais

A infraestrutura básica foi montada, com dois servidores de máquinas virtuais e um servidor de armazenamento, ligados através do protocolo NFS e sendo monitorado constantemente com o Nagios.

Na segunda fase, onde ocorre a criação e definição de máquinas virtuais, há necessidade da instalação e configuração do KVM. Esta instalação foi feita em ambos os servidores de máquinas virtuais, com todos os pacotes necessários para manipulação das mesmas.

Todas as VMs ficam no servidor de armazenamento, e nos servidores de máquinas virtuais são feitos apenas o processamento e armazenamento da memória RAM. As motivo no qual as VMs estarem armazenadas no servidor de armazenamento, é necessidade de fazer migrações destas máquinas virtuais.

Para criação e definição das máquinas virtuais, utilizou-se um código em XML (*Extensible Markup Language*), onde foram configuradas todas as características de cada uma delas de forma organizada, o que facilita sua compreensão e configuração. Neste arquivo XML, pôde-se escolher o nome, arquitetura, os discos rígidos e removíveis, a quantidade de memória RAM necessária, a interface de rede, e outras preferências.

Por padrão do sistema operacional, as configurações de rede são definidas para eth0 e lo, responsáveis pela interface de rede cabeada “Ethernet” e “localhost”. Os computadores que possuem uma interface de rede sem fio também possuem wlan0. Todas as configurações de rede cabeada e sem fio obtém IP automaticamente através do DHCP (*Dynamic Host Configuration Protocol*).

Ao total foram criadas seis máquinas virtuais com arquitetura 64 bits, 256 MB de memória RAM e 10 GB de armazenamento em disco virtual. Todas elas estão com o sistema operacional

Debian Squeeze 64bits (versão 6.0.2), com o Apache instalado para proporcionar a utilização das mesmas como um servidor web, e são identificadas com os nomes VM1, VM2, VM3, VM4, VM5 e VM6.

Na definição das máquinas virtuais, existe a criação de uma nova interface padrão, a *virbr0*, que fornece o endereço de rede para elas através de uma rede NAT (*Network Address Translation*). Como os servidores virtuais devem possuir um endereço de rede próprio, houve a necessidade de instalar o pacote “*bridge-utils*”, que possibilita a criação de uma interface de rede *bridge*, definida com o nome *br0*, onde cada máquina virtual recebe um endereço IP (*Internet Protocol*) real na rede. Esta definição de uma rede *bridge* é feita através de configuração do arquivo */etc/network/interfaces*.

Todas as máquinas foram criadas e definidas em um servidor de aplicação, e todos os passos necessários para realização deste processo, estão documentados no apêndice D, incluindo também as configurações na interface de rede.

4.3 Terceira fase: implementação da ferramenta

A última fase consistiu na implementação da ferramenta. Nesta fase, foram coletados todos os dados necessários e definido os cenários para execução do *software*.

Os parâmetros que o usuário pode definir são: horários de entrada e saída nos períodos da manhã e da tarde, o tempo de verificação ou períodos, que pode ser de 5, 10, 15, 20, 25 e 30 minutos, e a porcentagem do uso do processador considerada crítica. Após escolher o tempo é possível também selecionar o número de verificações ou *checkpoints*, que será um divisor do número de períodos estipulado. Por exemplo, se o usuário selecionar uma verificação a cada 5 minutos, o *checkpoint* pode ser 5 (uma a cada minuto) ou 1 (a cada cinco minutos), porém se o usuário selecionar uma verificação a cada 10 minutos, o *checkpoint* pode ser 10 (uma a cada minuto), 5 (uma a cada dois minutos), 2 (uma a cada 5 minutos) e 1 (uma a cada 10 minutos).

Como a ferramenta possui um cenário definido, as migrações só ocorrem em um sentido por vez. Inicialmente todas as máquinas virtuais devem estar ligadas e rodando no *Server 1*. Durante o horário de início do trabalho pela manhã até o horário de saída do trabalho a tarde, as máquinas virtuais saem do *Server 1* e vão para o *Server 2*. No período de tempo restante, ou seja entre o horário de saída do trabalho a tarde até o horário de entrada pela manhã, as

máquinas virtuais fazem o caminho inverso (se houver possibilidade), saindo do *Server 2* e indo para o *Server 1*. Após a execução, a ferramenta de consolidação decide com base nos dados que o usuário define, se a máquina deve ser migrada ou não.

A interface da ferramenta desenvolvida é apresentada na figura 4.2, onde os dados do cenário são especificados à esquerda. A direita pode-se ver onde as máquinas virtuais estão sendo executadas, e abaixo têm-se o log da execução.



Figura 4.2: Interface da ferramenta para consolidação de servidores.

Toda a implementação da ferramenta foi feita na linguagem de programação Java, utilizando uma *thread* que é executada conforme o período e o *checkpoint* estipulado, chamando outros métodos para coletar os dados necessários. Para algumas operações específicas foi utilizado Shell Script em conjunto com SSH, por exemplo, verificar o uso do processador de cada máquina virtual e solicitar a migração da máquina virtual. Os comandos utilizados no Shell Script e SSH precisam do acesso super usuário (root) no computador a ser acessado, para isto é necessário o compartilhamento de chaves de autenticação, que é configurado conforme o apêndice

E.

No algoritmo 1 têm-se o pseudo-código da *thread* que fica em execução. Durante sua execução é verificado o horário atual, com o período de trabalho da empresa estipulado pelo usuário, então a ferramenta determina o sentido da migração, que na estrutura de simulação existente pode ser do *Server 1* para o *Server 2* ou do *Server 2* para o *Server 1*.

Algorithm 1 Pseudo-código da *thread*.

Thread *Consolidacao*(*cenario*: *Cenario*);

variáveis

int maiorCPU = 0, *menorCPU* = 100, *minimoVer* = 0;

início

enquanto *Cenario.getFlagParar* <> *true* **faça**

maiorCPU ← *validaVMMaiorCPU*();

menorCPU ← *validaVMMenorCPU*();

horaA ← *horario*();

minimoVer ← *minimoVer* + 1;

se *horaA* > *Cenario.getHoraEManha* **and** *horaA* < *Cenario.getHoraSTarde* **and**
 minimoVer > *Cenario.getnumVerificacao* **and** *flagUsoCPUServer1* == *true* **en-**
 tão

se *Cenario.getFlagMaq2Ligada* == *false* **então**

WakeOnLan.acordaComputador(*ipBroadcast*, *mac*);

Thread.sleep(1000 * 2 * 60);

fim se

Shell.ExecuteCommand("sh migraVMMaiorCPUdoServer1paraServer2.sh");

senão

se (*usoCPUServer1* + *menorCPU*) < *Cenario.getLimiteCritico* **então**

Shell.ExecuteCommand("sh migraVMmenorCPUdoServer2paraServer1.sh");

fim se

se *existeVMServer2* == *false* **então**

Shell.ExecuteCommand("sh nomeDoScriptDesligar.sh");

fim se

fim se

Thread.sleep(*Cenario.getTempoVer*/*Cenario.getNumVer*);

fim enquanto

fim

Se o horário atual estiver dentro do período de trabalho, as migrações devem ocorrer no sentido *Server 1* para *Server 2*, porém para isto deve-se verificar se o uso do processador do *Server 1* está acima do limite crítico, pois com isso confirma-se a necessidade de migração para outro servidor. Na estrutura simulada existe apenas o *Server 2*, então certifica-se que o mesmo

não está ligado, se não estiver é enviado um comando em sua placa de rede para ser ligado⁴.

Com a máquina ligada, através do comando “ps -e -o pcpu,pid,args –sort pcpu | grep (NO-MEVM).img” obtem-se o número do processo (pid) de cada máquina virtual, e em seguida utilizando o comando “top -b -p (pid) -n 1” executado em Shell Script e SSH nos servidores de máquinas virtuais, verifica-se qual é a quantidade de processamento de cada máquina virtual. Esta verificação é possível pois cada máquina virtual KVM é executada como um processo do sistema operacional. Ao final é confirmado se o uso do processador no *Server 1* está maior do que o uso do processador no *Server 2*, e então é solicitada a migração da máquina virtual com maior uso de processador para o *Server 2*.

Caso o horário atual não estiver dentro do período de trabalho, as migrações devem ocorrer no sentido *Server 2* para *Server 1*. Os critérios de migração incluem escolher a máquina virtual com menor uso de processador, a verificação se alguma máquina virtual poderá migrar para o *Server 1* sem ultrapassar o limite crítico, e se o uso de processador do *Server 1* está menor que o *Server 2*. Ao final, se esses critérios são atendidos, a máquina virtual é migrada durante sua execução para o *Server 1*. Também é verificado se existe alguma máquina virtual no *Server 2*, caso não haja a máquina é desligada.

⁴<http://www.jibble.org/wake-on-lan/>

Capítulo 5

Testes e Resultados

Para validação da ferramenta para consolidação em software livre, são necessários testes para verificação de seu comportamento e eficiência. Durante a implementação da ferramenta, a infraestrutura para testes criada foi básica, com apenas duas máquinas físicas, devido a disponibilidade das mesmas, porém sempre houve a preocupação em tornar o ambiente mais realista possível.

São apresentados dois testes diferentes com duração de duas horas, utilizando geração de carga e migrações em *live*. O primeiro é baseado na geração de carga contínua, já o segundo com uma carga dinâmica. Também verifica-se a disponibilidade do serviço durante a migração e como foi o tempo de resposta dos servidores web das máquinas virtuais.

Para geração de carga nos servidores web, foi utilizado o OpenWebLoad¹, uma ferramenta que fornece as medições de desempenho em tempo real, e que possui distribuições para GNU/Linux e Microsoft Windows. Para aplicar o OpenWebLoad utilizando os parâmetros padrões, bastou executar a ferramenta e escolher o endereço que foi testado e o número de clientes simulados para acessar o endereço. Por exemplo, com o comando “./openload 10.81.112.200 20” a ferramenta gera carga no endereço 10.81.112.200, simulando o acesso de 20 clientes, durante um período indeterminado.

Na geração da carga contínua foi utilizado apenas o parâmetro para saída de arquivo, porém para carga dinâmica além da saída de arquivo, foi necessária a utilização de outro parâmetro para que a geração da carga ocorra em um período especificado.

Como a geração de carga dinâmica ocorre apenas durante um período especificado, foi necessário efetuar o controle desta carga gerada, disponibilizando um período onde não há geração

¹<http://openwebload.sourceforge.net/>

de carga. Cada período que a carga foi gerada é conhecido como instância. Para controlar as instâncias foi criado uma *thread* em Java que solicita a execução da ferramenta durante o período desejado.

Os comandos “./openload 10.81.112.200 20 > saidac.txt” e “./openload -l 120 10.81.112.200 20 > saidad.txt”, são exemplos para testes com carga contínua com 20 clientes e saída para o arquivo “saidac.txt”, e dinâmica durante 120 segundos, para 20 clientes e saída para o arquivo “saidad.txt”, respectivamente.

Na realização dos testes foi necessário o auxílio de 2 máquinas físicas, onde cada uma era responsável por gerar carga em três máquinas virtuais. A primeira máquina física cliente, que será identificada como “Clientes 1”, possui um processador Intel Core 2 Duo 1.66 GHz e 4 GB de memória RAM DDR2 Dual Channel, já a segunda, identificada como “Clientes 2”, contém um processador Intel Pentium D 3.00 GHz e 1 GB de memória RAM DDR Single Channel. Assim como em um ambiente real, as máquinas dos clientes não possuem uma padronização de configuração, portanto as duas máquinas físicas possuem configurações distintas.

5.1 Geração de carga contínua

O primeiro teste consistiu na geração de carga nas seis máquinas virtuais existentes, durante o período de uma hora. Para execução do teste que iniciou às 20h40 e terminou às 22h40, o cenário ficou definido com o horário de início do trabalho às 21h10 e término às 22h10, o tempo de verificações de cinco minutos, efetuando uma verificação por minuto, e intervalo crítico de 70%.

A máquina “Clientes 1” ficou responsável por gerar carga para VM1 de 10 clientes, para VM2 de 15 clientes e para VM4 de 10 clientes, enquanto o “Clientes 2” gera carga para VM3 de 5 clientes, para VM5 de 10 clientes e para VM6 de 15 clientes. Estes números de clientes foram gerados aleatoriamente.

A figura 5.1 apresenta o uso total de processador das máquinas virtuais em cada instante do teste. Durante a primeira meia hora, as máquinas virtuais não receberam nenhuma carga, com isso o uso do processador ficou sempre ocioso e abaixo de 5%. Após 21h08 foi iniciada a geração de carga em todas máquinas virtuais, porém após das 22h10 apenas a VM2 e a VM6 continuaram recebendo carga. No instante de 22h16 apenas a VM2 continuava a receber carga,

Tabela 5.1: Tempo de migração das máquinas virtuais com carga contínua (em segundos).

Tempo	VM1	VM2
Migração Ida	34	30
Migração Volta	31	30

que teve fim apenas às 22h22.

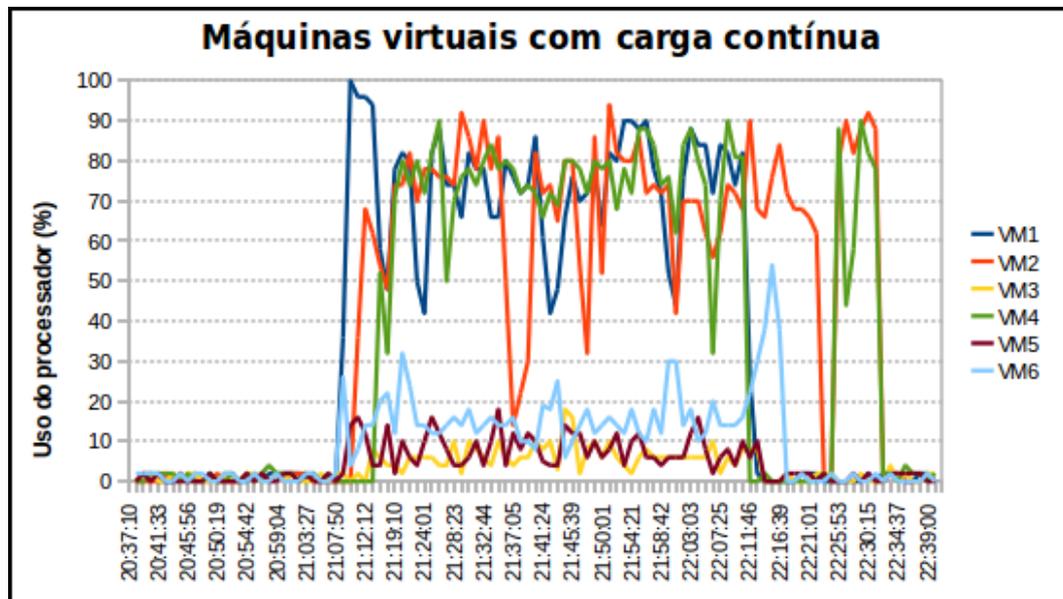


Figura 5.1: Geração de carga contínua nas máquinas virtuais.

Para validar que o programa não efetua migrações fora do horário de trabalho, após às 22h25 foi gerado carga nas VM2 e VM4 durante seis minutos. Como é demonstrado na figura 5.2, o *Server 2* não foi acionado neste período.

Em relação ao tempo que as máquinas virtuais gastaram para efetuar a migração, na tabela 5.1 está representado que a VM1 necessitou um tempo superior em relação a VM2, em ambos os sentidos da migração. A diferença não teve um impacto considerável, pois o tempo gasto é muito curto.

Na figura 5.2 pode-se verificar o comportamento das máquinas físicas durante a geração de carga nas máquinas virtuais, na qual pode-se observar que após às 21h08 o uso do processador no *Server 1* sobe chegando ao máximo de 92,16%. Com as cinco verificações acima do limite crítico, às 21h15 é enviado um comando para ligar o *Server 2*, e em seguida a VM1 é migrada, pois seu uso de processador é de 78%. Na próxima verificação nota-se que o uso de processador

do *Server 1* ainda não está abaixo do limite crítico, então migra-se a VM2, que estava com 74% do uso de processador. Após estas duas migrações, o sistema entra em equilíbrio até às 22h15, quando o uso do processador do *Server 1* fica abaixo dos 10%, e então as máquinas virtuais que estavam no *Server 2*, retornam a sua máquina de origem. Como no *Server 2* não existem mais máquinas virtuais, não existe a necessidade de mantê-la ligada, então é solicitado seu desligamento.

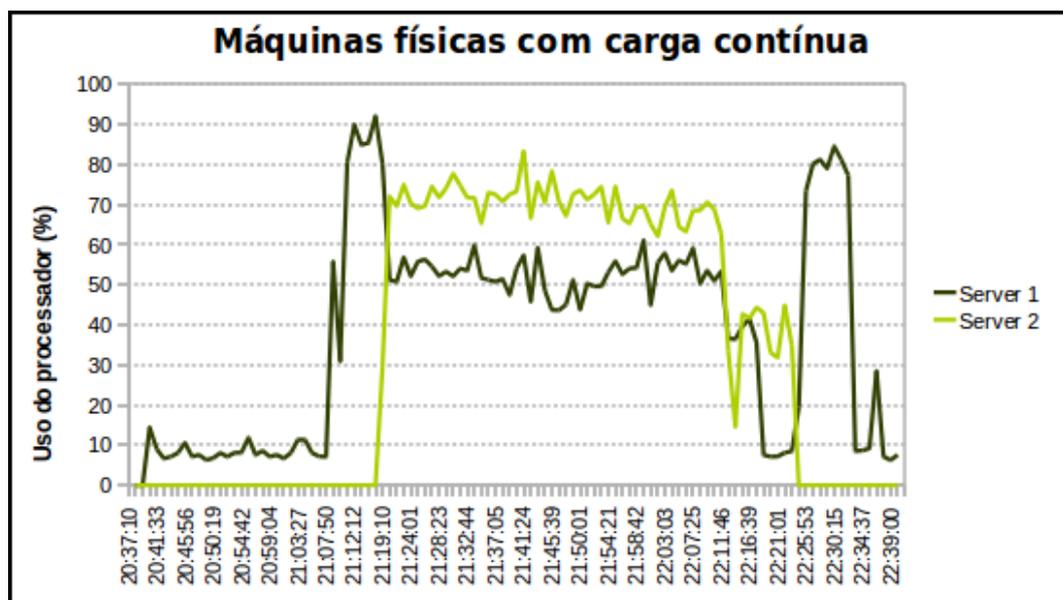


Figura 5.2: Comportamento das máquinas físicas com carga contínua.

5.2 Geração de carga dinâmica

O segundo teste consiste na geração de carga nas seis máquinas virtuais existentes, porém diferentemente do anterior, este teste teve uma carga gerada durante seis minutos, com intervalo de seis minutos em estado ocioso.

Para execução do teste que iniciou às 19h44 e terminou às 21h49, o cenário ficou definido com o horário de início do trabalho às 20h20 e término às 21h20, o tempo de verificações de cinco minutos, efetuando uma verificação por minuto, e intervalo crítico de 70%.

Assim como no primeiro teste, a máquina “Clientes 1” ficou responsável por gerar carga para VM1 de 10 clientes, para VM2 de 15 clientes e para VM4 de 10 clientes, enquanto o

“Clientes 2” gera carga para VM3 de 5 clientes, para VM5 de 10 clientes e para VM6 de 15 clientes,

A figura 5.3 apresenta o uso total de processador das máquinas virtuais, em cada instante do teste. Durante a primeira meia hora, as máquinas virtuais não receberam nenhuma carga, com isso o uso do processador ficou sempre ocioso e abaixo de 5%. Após 20h19 foi iniciada a geração de carga em todas máquinas virtuais durante seis minutos. Como foram detectadas durante as cinco verificações dentro do horário de trabalho, o uso de processador maior que o limite crítico, houve a necessidade de migração da máquina virtual que fazia uso de maior quantidade de processamento, a VM4 com 58%.

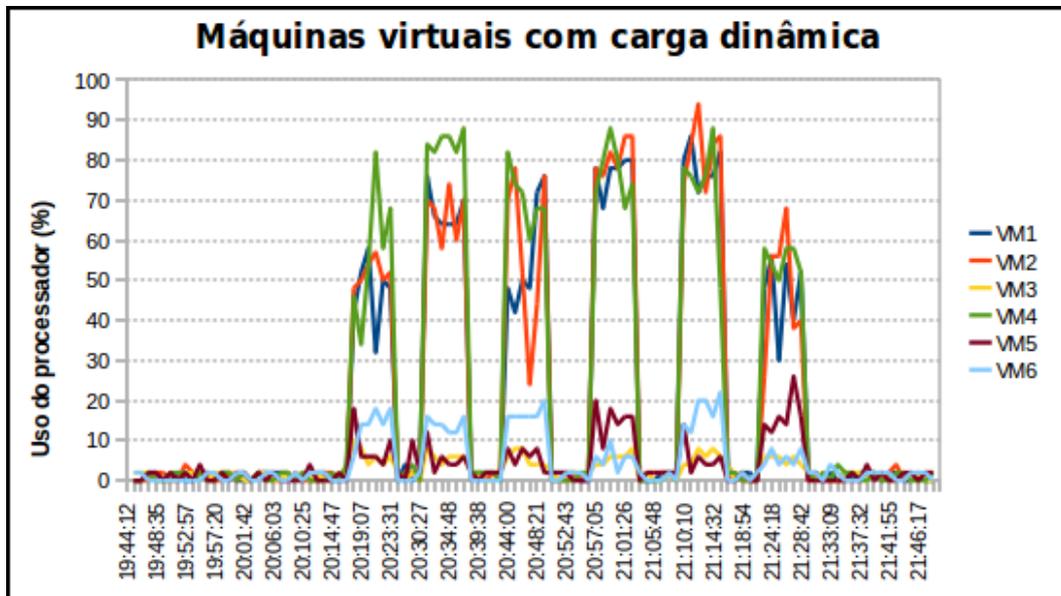


Figura 5.3: Geração de carga dinâmica nas máquinas virtuais.

Como as máquinas virtuais deixaram de receber carga durante seis minutos, só na próxima instância de geração de carga, entre às 20h31 e 20h37, que verificou-se a necessidade de migração de mais uma máquina virtual, pois com a migração de apenas uma, o uso de processador no *Server 1* ainda estava acima do limite crítico. A segunda máquina virtual migrada foi a VM2, com 70% de utilização do processador.

Durante todas as outras instâncias, não houve necessidade de migração, pois o sistema ficou em equilíbrio até chegar no horário de saída do trabalho, após às 21h20. Assim que o horário de saída do trabalho foi alcançado, haviam cinco verificações indicando que poderia retornar

Tabela 5.2: Tempo de migração das máquinas virtuais com carga dinâmica (em segundos).

Tempo	VM2	VM4
Migração Ida	28	33
Migração Volta	29	30

uma máquina virtual, pois seu uso de processamento somado com o uso total do processador do *Server 1*, era menor que o limite crítico. No instante 20h20 houve a migração da máquina virtual com menor utilização de processamento, a VM4. Logo após verificou-se que poderia ser migrada outra máquina virtual, pelo mesmo motivo da anterior, então migrou-se a VM2, e então foi solicitado o desligamento do *Server 2* que não seria mais utilizado.

Na figura 5.4 pode-se validar que o programa fora da hora de trabalho não irá efetuar migrações, após às 21h23 foi gerado carga em todas as máquinas virtuais durante seis minutos, e o *Server 2* não foi acionado neste período.

Pode-se verificar o comportamento das máquinas físicas durante a geração de carga nas máquinas virtuais, na figura 5.4, onde após as 20h18 o uso do processador no *Server 1* sobe chegando ao máximo de 94,28%. Com as cinco verificações acima do limite crítico, às 20h23 é enviado um pulso para ligar o *Server 2*, e em seguida migrada a VM4, pois seu uso de processador é de 68%. Após o intervalo de seis minutos entre as cargas geradas, na segunda instância foi verificado que o uso de processador do *Server 1* estava acima do limite crítico, então é migrada a VM2, que estava com 70% do uso de processador. Após estas duas migrações, o sistema entra em equilíbrio até às 21h20, quando o uso do processador do *Server 1* fica abaixo dos 15% durante as cinco verificações, e então as máquinas virtuais que estavam no *Server 2*, retornam a sua máquina de origem. Como no *Server 2* não abriga mais máquinas virtuais, não existe a necessidade de mantê-la ligada, então é solicitado seu desligamento.

Com os testes efetuando a geração de carga dinâmica, o tempo que as máquinas virtuais gastaram para efetuar a migração estão representados na tabela 5.2. A VM4 necessitou um tempo superior em relação a VM2, em ambos os sentidos da migração. Assim como no teste com carga contínua, apesar de haver diferença no tempo, esta é bastante pequena.

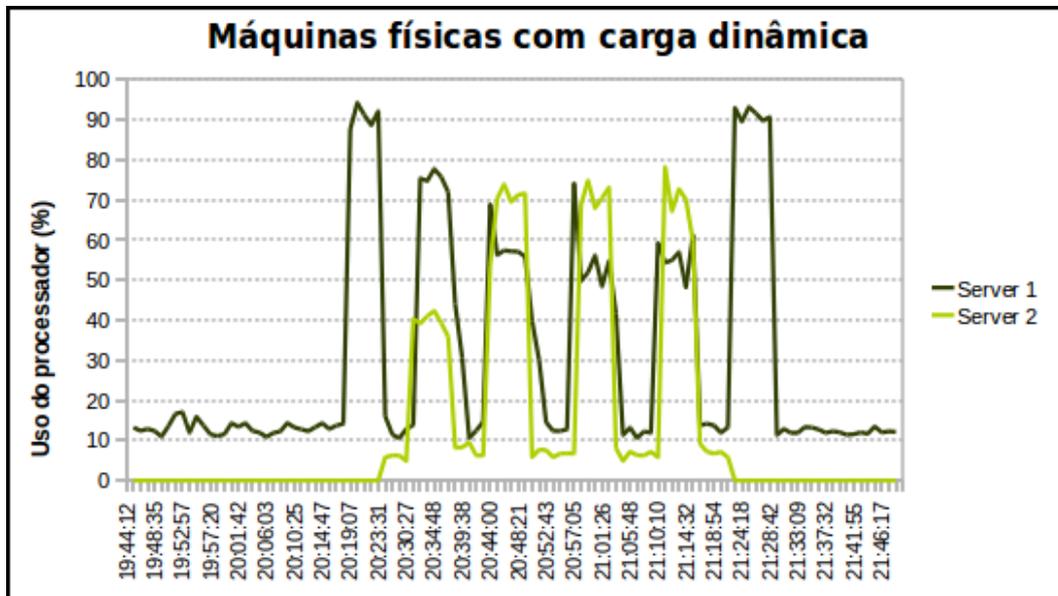


Figura 5.4: Comportamento das máquinas físicas com carga dinâmica.

5.3 Resposta da rede e tempo de migração

Para realização de testes em relação ao desempenho das máquinas virtuais durante a migração, utilizou-se da ferramenta “ping” com a saída em um arquivo. O comando “ping” tem como objetivo enviar um pacote de requisição ICMP (*Internet Control Message Protocol*) e aguardar sua resposta, ao final é calculado o seu tempo de resposta.

Todos os testes foram feitos a partir do “Clientes 1”, enviando pacotes de 56 bytes, tamanho padrão da ferramenta. Estes testes foram feitos com duas máquinas virtuais, a VM1 e VM2, obtendo os seus respectivos tempos de resposta. Não foi levado em consideração nenhum critério para balanceamento nos servidores físicos, pois este teste tem o objetivo de comparar o desempenho das máquinas virtuais durante a migração. Como já descrito no capítulo 4, a rede que interliga as máquinas físicas é *Fast Ethernet*, ou seja, sua velocidade máxima é 100 Mbit/s.

Na figura 5.5, têm-se as migrações das máquinas virtuais em estado ocioso entre o *Server 1* e o *Server 2*.

Durante as 19 primeiras verificações as máquinas virtuais estavam no *Server 1*, e logo após foi solicitada a migração para o *Server 2*. A VM1 demorou 19 segundos até finalizar a migração, tendo um tempo de resposta máximo de 139 ms. O tempo de migração da VM2 foi de 21 segundos, e seu tempo de resposta máximo de 128 ms. As máquinas virtuais ficaram durante

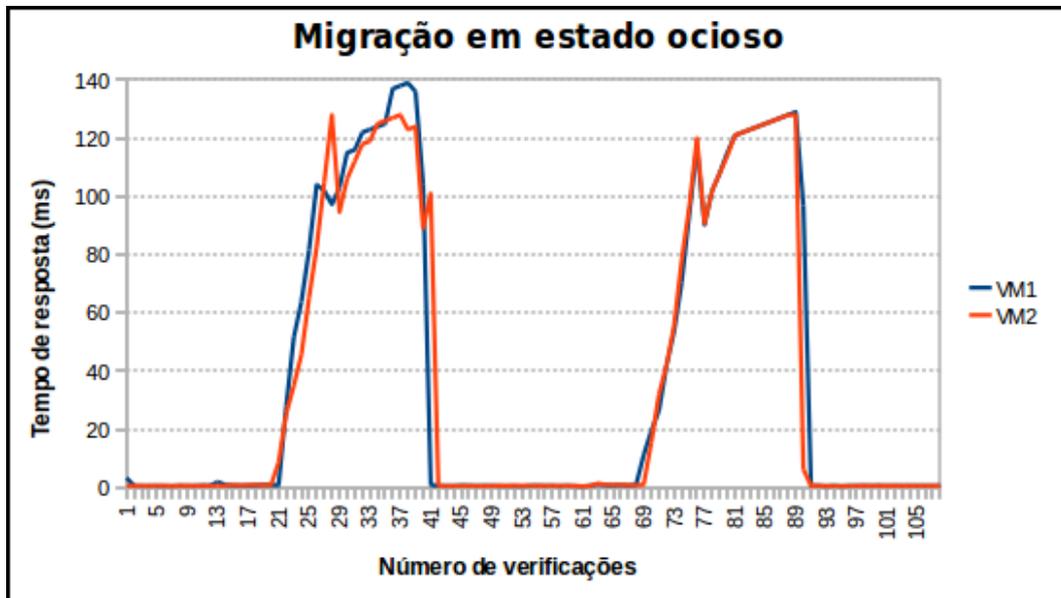


Figura 5.5: Migração de máquinas virtuais ociosas.

25 segundos no *Server 2*, e então solicitou-se a migração para o *Server 1*. Durante o retorno, a VM1 precisou de 22 segundos para finalizar a migração, e um tempo de resposta máximo de 129 ms. Já a VM2 utilizou-se de 20 segundos, e um tempo máximo de resposta de 128 ms. Todas as requisições obtiveram resposta, com 0% de pacotes perdidos.

O primeiro teste com geração de carga está representado na figura 5.6, onde ocorreram migrações das máquinas virtuais entre o *Server 1* e o *Server 2*. Cada máquina virtual foi testada individualmente, e receberam a carga gerada de 10 clientes através do OpenWebLoad.

As primeiras 25 verificações foram feitas com as máquinas virtuais no *Server 1*, e logo após foi solicitada a migração para o *Server 2*. A VM1 demorou 20 segundos até finalizar a migração, tendo um tempo de resposta máximo de 125 ms. O tempo de migração da VM2 foi de 27 segundos, e seu tempo de resposta máximo de 126 ms. As máquinas virtuais ficaram durante 27 segundos no *Server 2*, e então solicitou-se a migração para o *Server 1*. Durante o retorno, a VM1 precisou de 29 segundos para finalizar a migração, e um tempo de resposta máximo de 123 ms. Por sua vez, a VM2 utilizou 19 segundos para migração e apresentou tempo máximo de resposta de 124 ms. Assim como no teste com os servidores ociosos, todas as requisições obtiveram resposta.

Seguindo os procedimentos de execução dos testes anteriores, o terceiro cenário foi o que

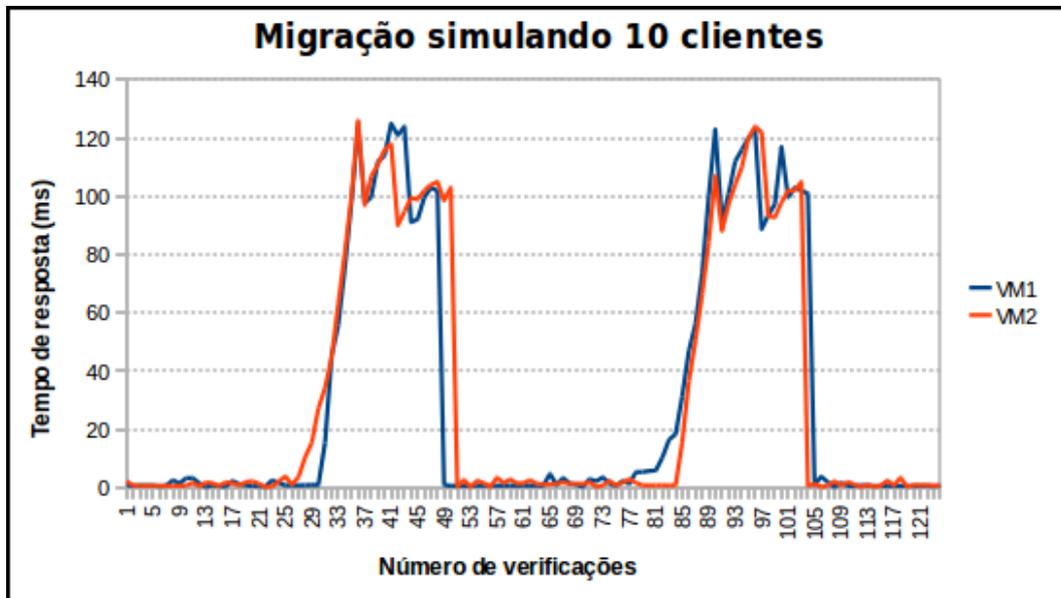


Figura 5.6: Migração de máquinas virtuais com carga simulada de 10 clientes.

mais exigiu em relação ao número de carga gerada em uma única máquina virtual. Na figura 5.7, têm-se a representação dos tempo de resposta e de migração, onde cada máquina virtual foi testada individualmente, recebendo a carga gerada de 50 clientes.

Durante as primeiras 44 verificações as máquinas virtuais estavam no *Server 1*, e logo após foi solicitado a migração para o *Server 2*. A VM1 demorou 66 segundos até finalizar a migração, tendo um tempo de resposta máximo de 135 ms. O tempo de migração da VM2 foi bem superior ao da VM1, durando 153 segundos, e tendo o tempo máximo de resposta em 130 ms. Durante o retorno, a VM2 teve o tempo total de migração inferior em relação a VM1, invertendo os resultados apresentados na migração de ida. No retorno da VM1 para o *Server 1*, demorou 145 segundos e teve o tempo máximo de resposta de 117 ms, enquanto a VM2 demorou 66 segundos com o tempo máximo de resposta em 123 ms. Todas as requisições obtiveram resposta, com 0% de pacotes perdidos, assim como em ambos os testes anteriores.

Para facilitar a comparação dos tempos de migração e resposta dos três testes, na tabela 5.3, têm-se a representação dos dados. O tempo de migração tem um aumento considerável quando a máquina virtual recebe uma carga maior, este fato se deve ao uso do processador, pois conforme mostrado na tabela o tempo de resposta não sofre nenhuma alteração avultada.

O tempo máximo de resposta está relacionado com a migração em ambos os sentidos, e

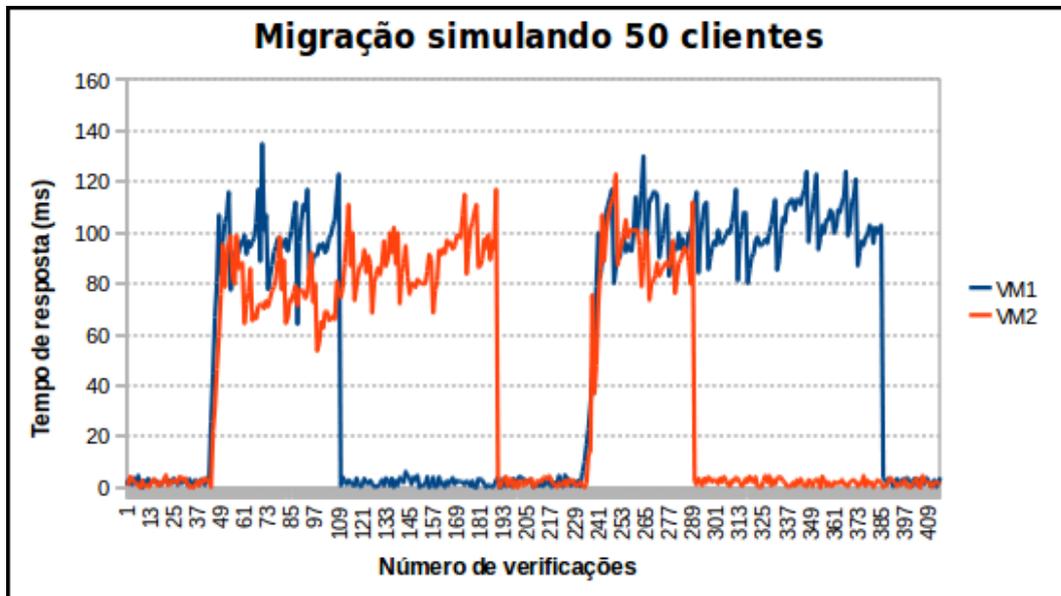


Figura 5.7: Migração de máquinas virtuais com carga simulada de 50 clientes.

Tabela 5.3: Comparação entre migrações.

Tempo	Ocioso		10 clientes		50 clientes	
	VM1	VM2	VM1	VM2	VM1	VM2
Migração Ida (s)	19	21	20	27	66	145
Migração Volta (s)	22	20	29	19	153	55
Resposta máxima (ms)	139	128	125	126	135	123

mesmo com o aumento de carga o valor mantém-se entre 80 e 140 ms. Este tempo de migração e de resposta máxima poderia ser inferior se a rede utilizada possuísse uma taxa de transferência maior, por exemplo a rede *Gigabit Ethernet*, onde a velocidade máxima poder chegar em 1 Gbit/s.

5.4 Estudo de aplicação em caso real

Utilizando o Simulador de consumo de energia elétrica² desenvolvido pela COPEL (Companhia Paranaense de Energia Elétrica), é possível verificar o real consumo de qualquer equipamento, levando em consideração a potência do mesmo, e a quantidade de tempo que ele fica em utilização.

Para verificar o consumo, foi utilizado o servidor em *rack* Dell PowerEdge 2950, e suas

²<http://www.copel.com/hpcopel/simulador/>

especificações técnicas descritas pelo fabricante. Segundo a Dell, este servidor possui fontes redundantes de 750W de potência.

Ao selecionar no simulador um equipamento com 750W de potência e o período de tempo que o mesmo fica ligado, a própria ferramenta calcula o total de kWh (quilowatt-hora).

Primeiramente selecionando um período de tempo de 24h por dia, durante 5 dias por semana, em todas as semanas do ano, o simulador fornece que o total de 395kWh serão gastos durante um mês. Este total de quilowatt-hora, custa mensalmente R\$189,45.

O segundo cálculo foi feito baseado na utilização da ferramenta de consolidação em software livre desenvolvida neste trabalho, com um cenário de 8h trabalhadas por dia em uma empresa, somando mais 2h diárias de intercalação, totalizando 10h com necessidade do servidor estar ligado e com uso de processador. Selecionamos então a mesma potência de 750W, com um período de 10h por dia, durante 5 dias por semana, em todas as semanas do ano, o simulador fornece que o total de 165kWh serão gastos durante um mês, totalizando um custo mensal de R\$79,19.

A utilização da ferramenta de consolidação, pode gerar uma economia de energia elétrica de aproximadamente 2790kWh, totalizando R\$1323,12. Este valor foi calculado baseado em apenas um servidor, porém em grandes organizações existem N servidores, com especificações técnicas diferentes, alguns deles fazendo maior uso de energia elétrica e outros menor.

Capítulo 6

Conclusão

A ferramenta de consolidação de servidores desenvolvida demonstrou que existe viabilidade na utilização da técnica e em sua implantação, através de testes e da análise dos mesmos. Independente do tamanho da empresa, se ela enquadrar-se no esquema do cenário proposto, a mesma terá uma economia imediata com relação ao consumo de energia, sem sofrer danos na qualidade de serviço.

Mesmo a ferramenta limitando-se à especificação de um cenário, toda a infraestrutura criada possui basicamente um controle sobre a necessidade ou não da migração das máquinas virtuais. Caso a empresa deseje qualquer alteração em relação às variáveis utilizadas na definição da migração, as alterações se tornam bem simples por serem pontuais.

Levando-se em conta que uma máquina física precisa de um tempo para ser ligada, e a migração ter gasto em média pouco mais que meio minuto utilizando carga contínua e dinâmica, o tempo de migração pode ser considerado pequeno.

O principal diferencial da ferramenta de consolidação de servidores desenvolvida, é que a mesma está baseada totalmente na utilização de software livre, tornando-se uma opção para quem deseja efetuar a consolidação de sua infraestrutura sem a necessidade de pagamento de licenças com softwares proprietários.

Em relação aos resultados e os testes efetuados, alguns fatores poderiam ser modificados para tornar os mesmos mais realistas. Na montagem da infraestrutura foi utilizada uma máquina física comum como servidor de armazenamento, o que tem influência direta em relação ao desempenho proporcionado por ela. Em grandes empresas e organizações, onde há necessidade de alta disponibilidade, é recomendado utilizar uma unidade NAS (*Network-Attached Storage*) ou SAN, que são dispositivos especiais para armazenamento em uma rede.

Outro fator que interfere diretamente no desempenho, é a velocidade da rede que interliga os servidores. A velocidade da rede onde foram realizados os testes, estava limitada à velocidade máxima de 100 Mbit/s, devido a tecnologia utilizada na interface de rede e nos cabeamentos.

6.1 Trabalhos futuros

Para a preparação de toda a infraestrutura de simulação, foram gastos meses de estudos e testes. Um trabalho que traria uma facilidade grande para que este ambiente possa ser reproduzido, seria a criação de um pacote de instalação que automatize a instalação e configuração, tanto em servidores de máquinas virtuais, quanto no servidor de armazenamento.

A ferramenta desenvolvida proporciona a base para que um cenário definido pelo usuário possa funcionar, limitado conforme as variáveis, tempo de trabalho, tempo de verificação, número de verificações e limite crítico. Para transformar o trabalho em um estado mais genérico ou adaptável, seria necessária a criação de novas variáveis e a aplicação de técnicas robustas de balanceamento de carga, possibilitando por exemplo, a exclusão da escolha do horário de trabalho, para que este não seja um fator decisivo no sentido da migração.

Ainda em relação à generalização, seria interessante a criação de uma base de dados, onde poderia ser feito o cadastro, alteração e exclusão de servidores de máquinas virtuais e de máquinas virtuais. Com este cadastro é possível ampliar, reduzir e modificar o tamanho da infraestrutura utilizada, conforme a necessidade do usuário.

Em relação à interface do programa, um recurso que poderia ser agregado seria a demonstração de gráficos em tempo real do uso de processador das máquinas físicas e das máquinas virtuais. Este recurso poderia ser feito também com o desenvolvimento de uma interface web em conjunto com o RRDTool¹, uma ferramenta *open source* que auxilia na criação de gráficos para monitoramento.

¹<http://oss.oetiker.ch/rrdtool/>

Apêndice A

Instalação do Servidor Nagios

Este apêndice foi baseado em documentações e tutoriais já existentes [59, 60, 61], e tem o objetivo de criar um tutorial que demonstra a instalação do *software* de monitoramento Nagios (versão 3.2.3), Nagios *Plugins* (versão 1.4.15), Nagios NRPE (Nagios Remote Plugin Executor) (versão 2.12) e NDOUtils (versão 1.4b9), no GNU/Linux Debian Squeeze (versão 6.0.2).

Toda e qualquer modificação deve ser feita utilizando o super usuário ou root. Para não haver problemas com o Nagios e seus *plugins*, siga até o Passo 14 deste tutorial. A partir do Passo 15 a instalação da ferramenta NDOUtils é opcional.

• Passo 1 – Instalação dos requisitos

Inicialmente é necessário efetuar a instalação e configuração de pacotes que são pré-requisitos. Os pacotes que são pré-requisitos são.

- Apache
- GD Development Libraries
- GCC compiler and development libraries

O Apache e o GR Development Libraries, podem ser instalados do seguinte comando no terminal. Já o GCC vem instalado junto com o SO.

```
# apt-get install apache2 libgd2-xpm-dev
```

• Passo 2 – Criação de usuário e grupos

É necessária a criação de um usuário e dois grupos antes de iniciar a instalação do Nagios.

Para criar o usuário “nagios” digite os seguinte comando no terminal.

```
# /usr/sbin/useradd nagios
```

Após a criação do usuário é necessário definir uma senha para o mesmo. A definição é feita com este comando.

```
# /passwd nagios
```

Com o usuário criado, agora cria-se de um grupo denominado “nagios”.

```
# /usr/sbin/groupadd nagios
```

Em seguida é necessário definir o grupo “nagios” ao qual o usuário “nagios” pertence.

```
# /usr/sbin/usermod -G nagios nagios
```

Agora é preciso criar um grupo chamado de “nagcmd” para permitir que comandos externos sejam passados pela interface web.

```
# /usr/sbin/groupadd nagcmd
```

```
# /usr/sbin/usermod -G nagcmd nagios
```

```
# /usr/sbin/usermod -G nagcmd www-data
```

● Passo 3 – Criação de pastas

Antes de iniciar a instalação do *software* Nagios, por questão de organização, é recomendado a criação de um diretório de armazenamento para os downloads. Neste tutorial o diretório deverá ser criado no caminho `/home/usuário/nagiosd`, onde o *usuário* é o nome do usuário logado no Debian.

Em nosso tutorial o nome do usuário é “allysson”.

```
# mkdir /home/allysson/nagiosd
```

```
# cd /home/allysson/nagiosd
```

● Passo 4 – Download do Nagios e do Nagios *Plugins*

Agora para estar com a versão mais atualizada do Nagios, acesse o site do *soft-*

ware¹. Através do comando abaixo efetue o *download* do Nagios e do Nagios *Plugins*.

```
# wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.3.tar.gz
# wget http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.15.tar.gz
```

• Passo 5 – Instalação do Nagios

Se você já efetuou o *download* do Nagios, para iniciar a instalação é necessário descompactar e compilar. Para descompactar o pacote e compilar segue os comandos abaixo.

```
# cd /home/allysson/nagiosd
# tar xzvf nagios-3.2.3.tar.gz
# cd nagios-3.2.3
# ./configure --with-command-group=nagcmd
# make all
```

Com os binários gerados, siga os comandos para instalá-los, configurar os script de inicialização e permissões para comandos externos.

```
# make install
# make install-init
# make install-config
# make install-commandmode
```

Para finalizar a instalação do Nagios, é necessário instalar e configurar o ambiente web. Inicialmente é feita a instalação do ambiente e depois criar um usuário e senha para autenticar no ambiente.

```
# make install-webconf
# htpasswd -c /usr/local/nagios/etc/htpasswd.users
nagiosadmin
```

Agora o Nagios já está instalado porém é necessário reiniciar o Apache.

¹<http://www.nagios.org/download>

Lembrando que ainda não é para iniciar o mesmo, pois antes é necessária a instalação dos *plugins*.

```
# /etc/init.d/apache2 reload
```

- **Passo 6 – Instalação dos *plugins* do Nagios**

Terminada a instalação do *core* do Nagios e já efetuado o *download* do Nagios *plugins* o primeiro passo é a instalação dos *plugins* do Nagios. Com os *plugins* é possível monitorar serviços de *hardware*.

Para descompactar o pacote e compilar segue os comandos abaixo.

```
# cd /home/allysson/nagiosd
# tar xzvf nagios-plugins-1.4.15.tar.gz
# cd nagios-plugins-1.4.15
# ./configure --with-nagios-user=nagios
--with-nagios-group=nagios
# make
# make install
```

Após a instalação dos *plugins* tem que configurar o Nagios para iniciar com o sistema. Isso pode ser feito com o seguinte comando.

```
# ln -s /etc/init.d/nagios /etc/rcS.d/S99nagios
```

- **Passo 7 – Validando os arquivos de configuração do Nagios**

A verificação de erros nos arquivos de configurações do Nagios é feita da seguinte maneira.

```
# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

- **Passo 8 – Iniciando o Nagios**

Após a instalação do Nagios e seus *plugins*, pode-se iniciar o *software* através do seguinte comando.

```
# /etc/init.d/nagios start
```

Para acessar o ambiente web, é necessário do navegador de internet. Por padrão o Nagios

já deve estar monitorando a máquina local (*localhost*).

```
# http://ip_do_servidor_nagios/nagios/  
# http://localhost/nagios/
```

Logo após ele solicitará o usuário e a senha do “nagiosadmin”, setado anteriormente.

- **Passo 9 – Download do Nagios NRPE**

O NRPE tem como função atuar como uma espécie de um agente, que executa o Nagios em uma máquina remota. Através do comando abaixo efetue o *download* do Nagios NRPE.

```
# cd /home/allysson/nagiosd  
# wget http://downloads.sourceforge.net/project/nagios/nr  
pe-2.x/nrpe-2.12/nrpe-2.12.tar.gz
```

- **Passo 10 – Instalando o Nagios NRPE**

Se você já efetuou o *download* do Nagios NRPE, para iniciar a instalação é necessário descompactar e compilar. Para descompactar o pacote e compilar segue os comandos abaixo.

```
# tar xzf nrpe-2.12.tar.gz  
# cd nrpe-2.12  
# ./configure  
# make all  
# make install-plugin  
# make install-daemon  
# make install-daemon-config
```

Em alguns casos, dependendo da versão do sistema operacional, exige-se a instalação de dois pacotes com módulos SSL. A instalação destes pacotes pode ser feita com o comando abaixo.

```
# apt-get install libssl-dev openssl
```

- **Passo 11 – Validando o Nagios NRPE**

Com o comando abaixo, valida-se a instalação do NRPE. A saída do comando deve apresentar apenas “NRPE v2.12”.

```
# /usr/local/nagios/libexec/check_nrpe -H localhost
```

- **Passo 12 – Configurando o Nagios NRPE**

Agora é necessário definir o comando “check_nrpe”. A definição é feita editando o arquivo “commands.cfg”.

```
# nano /usr/local/nagios/etc/objects/commands.cfg
```

Então ao final do arquivo inclui-se as seguintes linhas:

```
define command {  
    command_name check_nrpe  
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c  
    $ARG1$  
}
```

- **Passo 13 – Definindo um template para o Nagios NRPE**

Para organizar o monitoramento, recomenda-se a criação de um modelo para um grupo de máquinas. Abra o arquivo “template.cfg” com o comando abaixo.

```
# nano /usr/local/nagios/etc/objects/templates.cfg
```

E insira as seguintes linhas:

```
define host {  
    name linux-nrpe ; Nome do modelo  
    use generic-host ; Herda valores padrões de generic-host  
    check_period 24x7  
    check_interval 1  
    retry_interval 1  
    max_check_attempts 10  
    check_command check-host-alive  
    notification_period 24x7  
    notification_interval 30  
    notification_options d,r
```

```
contact_groups admins
register 0

}
```

- **Passo 14 – Definindo uma máquina remota para monitoramento**

A definição de máquinas físicas será feita na pasta “/usr/local/nagios/etc/meuservers/”.

Com o comando abaixo, um arquivo “server1.cfg” é criado.

```
# nano /usr/local/nagios/etc/meuservers/server1.cfg
```

Abaixo têm-se um modelo de um servidor definido:

```
define host {
    use linux-nrpe ; Herda valores padrões de linux-nrpe
    host_name Server1 ; Nome do host
    alias Debian 6 ; Um nome longo para o host
    address 10.81.118.59 ; Endereço IP do host
    check_period 24x7
    check_interval 1
    retry_interval 1
    max_check_attempts 10
    normal_check_interval 1
    retry_check_interval 1
}

define service {
    use generic-service ; Herda valores genéricos
    host_name Server1
    service_description Usuarios Ativos Atualmente
    check_command check_nrpe!check_users
}

define service {
    use generic-service ; Herda valores genéricos
```

```

    host_name Server1
    service_description Espaco livre em /dev/sda1
    check_command check_nrpe!check_sda1
}

define service {
    use generic-service ; Herda valores genéricos
    host_name Server1
    service_description Numero de Processos
    check_command check_nrpe!check_total_procs
}

define service {
    use generic-service ; Herda valores genéricos
    host_name Server1
    service_description Processos Zombie
    check_command check_nrpe!check_zombie_procs
}

define service {
    use generic-service ; Herda valores genéricos
    host_name Server1
    service_description Carga da CPU
    check_interval 1
    retry_interval 1
    normal_check_interval 1
    retry_check_interval 1
    check_command check_cpu!20!80
}

```

Obs.: Na definição deste servidor foi utilizado o plugin “check_cpu”. Para instalar este plugin siga as informações abaixo, caso não deseje este plugin pule para o Passo 15.

Instalação do plugin “check_cpu”

O plugin “check_cpu” retorna o valor real do uso do processador da máquina. Este plugin está disponível sob a Licença Pública Geral, no site de seu desenvolvedor². Faça o download do plugin e copie para pasta onde localizam-se os plugins do Nagios.

```
# cd /home/allysson/nagiosd
# wget https://www.monitoringexchange.org/attachment/download/Check-Plugins/Operating-Systems/check_cpu-sh/check_cpu.sh
# cp check_cpu.sh /usr/local/nagios/libexec/
```

Agora é necessário definir o comando “check_cpu”. A definição é feita editando o arquivo “commands.cfg”.

```
# nano /usr/local/nagios/etc/objects/commands.cfg
```

Então ao final do arquivo inclui-se as seguintes linhas:

```
define command {
    command_name check_cpu
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c
    check_cpu -t 30 -a $ARG1$ $ARG2$
}
```

• Passo 15 – Instalação dos requisitos

O NDOUtils tem como função exportar as informações obtidas pelo Nagios, para uma base de dados MySQL. A instalação desta ferramenta é opcional.

Inicialmente é necessário efetuar a instalação e configuração de pacotes que são pré-requisitos. Os pacotes que são pré-requisitos são.

- mysql-server
- libmysqlclient-dev

²<http://www.matejunkie.com/cpu-check-plugin-for-nagios/>

– phpmyadmin

O mysql-server e o libmysqlclient-dev, podem ser instalados do seguinte comando no terminal.

```
# apt-get install mysql-server libmysqlclient-dev
```

Durante a instalação será solicitado uma palavra passe.

Em seguida inicia-se a instalação e configuração do phpmyadmin.

```
# apt-get install phpmyadmin
```

Durante a instalação será solicitado qual o servidor Web utilizado (selecione Apache2), se você deseja configurar a base (selecione Sim), qual é a palavra passe (digite uma senha para acesso a interface do phpMyAdmin) e a senha do MySQL (digite a mesma escolhida durante a instalação do MySQL).

Com o banco de dados instalado, podemos acessar através do navegador no endereço <http://localhost/phpmyadmin/>. Será solicitado um usuário e uma senha (usuário é root, senha é a escolhida para palavra passe).

O próximo passo é a criação de uma base de dados e um usuário que tenha permissão para SELECT, INSERT, DELETE e UPDATE. Selecione a guia “Privileges”, em seguida “Add new User”, escolha o nome do usuário e senha, selecione a opção “Create database with same name and grant all privileges”, marque as caixas SELECT, INSERT, DELETE e UPDATE, e por fim clique em “Go”.

Pronto agora deve existir uma base com o mesmo nome do usuário. Durante este tutorial, o nome do usuário e base de dados criadas foram “nagios”.

● **Passo 16 – Download do NDOUtils**

. Através do comando abaixo efetue o *download* do NDOUtils.

```
# cd /home/allysson/nagiosd  
# http://downloads.sourceforge.net/project/nagios/ndoutils  
-1.x/ndoutils-1.4b9/ndoutils-1.4b9.tar.gz
```

● **Passo 17 – Instalando o NDOUtils**

Se você já efetuou o *download* do NDOUtils, para iniciar a instalação é necessário

descompactar e compilar. Para descompactar o pacote e compilar segue os comandos abaixo.

```
# tar -xzf ndoutils-1.4b9.tar.gz
# cd ndoutils-1.4b9
# ./configure --prefix=/usr/local/nagios/ --enable-mysql
--disable-pgsql --with-ndo2db-user=nagios
--with-ndo2db-group=nagios
# make
# make install
# make install-init
# cd db/
# ./installdb -u root -p nagios -h localhost -d nagios
```

• Passo 18 – Configurando o NDOUtils

Agora é necessário editar o arquivo de configuração do Nagios.

```
# nano /usr/local/nagios/etc/nagios.cfg
```

Verifique se o “event_broker_options” está -1, caso contrário edite deixe como abaixo.

```
event_broker_options=-1
```

Então ao final do arquivo incluí-se as seguintes linhas:

```
broker_module=/usr/local/nagios/bin/ndomod.o
config_file=/usr/local/nagios/etc/ndomod.cfg
```

Agora é necessário instalar os arquivos de configuração de exemplo.

```
# cd ..
# make install-config
```

Os arquivos de configuração de exemplo devem ser renomeados para seu funcionamento. Edite os arquivos de configuração ndomod.cfg e ndo2db.cfg, especificando o nome do banco de dados, prefixo, usuário e senha.

```
# mv /usr/local/nagios/etc/ndomod.cfg-sample
```

```
/usr/local/nagios/etc/ndomod.cfg
# mv /usr/local/nagios/etc/ndo2db.cfg-sample
/usr/local/nagios/etc/ndo2db.cfg
```

Configure o NDOUtils para que inicie junto com o SO.

```
# update-rc.d ndo2db defaults
# update-rc.d nagios defaults
```

- **Passo 19 – Iniciando e validando o NDOUtils**

Com o comando abaixo, inicia-se o NDOUtils.

```
# /etc/init.d/ndo2db start
```

Para validar se o NDOUtils iniciou, entre com o seguinte comando.

```
# cat /usr/local/nagios/var/nagios.log | grep sink
```

Se o NDOUtils iniciou de forma correta, a saída do comando acima deve conter uma linha no final com a seguinte informação: “ndomod: Successfully flushed N queued items to data sink”. Caso retorne uma linha dizendo que houve erro ao iniciar o NDOUtils, revise a configuração feita e ao final reinicie o NDOUtils com o comando abaixo.

```
# /etc/init.d/ndo2db restart
```

Apêndice B

Instalação do Cliente Remoto Nagios

Este apêndice foi baseado em documentações e tutoriais já existentes [59, 60], e tem o objetivo de criar um tutorial que demonstra a instalação do *software* de monitoramento Nagios *Plugins* (versão 1.4.15) e Nagios NRPE (Nagios Remote Plugin Executor) (versão 2.12), no GNU/Linux Debian Squeeze (versão 6.0.2).

Toda e qualquer modificação deve ser feita utilizando o super usuário ou root. Para não haver problemas na instalação e configuração siga todos os passos deste tutorial.

• Passo 1 – Instalação dos requisitos

Inicialmente é necessário efetuar a instalação e configuração de pacotes que são pré-requisitos. Os pacotes que são pré-requisitos são.

- build-essential
- libssl-dev
- xinetd

Estes pacotes podem ser instalados do através do seguinte comando no terminal.

```
# apt-get build-essential libssl-dev install xinetd
```

• Passo 2 – Criação de usuário e grupos

É necessária a criação de um usuário e dois grupos antes de iniciar a instalação do Nagios.

Para criar o usuário “nagios” digite os seguinte comando no terminal.

```
# /usr/sbin/useradd nagios
```

Após a criação do usuário é necessário definir uma senha para o mesmo. A definição é feita com este comando.

```
# /passwd nagios
```

● Passo 3 – Criação de pastas

Antes de iniciar a instalação do *software* Nagios, por questão de organização, é recomendado a criação de um diretório de armazenamento para os downloads. Neste tutorial o diretório deverá ser criado no caminho `/home/usuário/nagiosd`, onde o *usuário* é o nome do usuário logado no Debian.

Em nosso tutorial o nome do usuário é “allysson”.

```
# mkdir /home/allysson/nagiosd
# cd /home/allysson/nagiosd
```

● Passo 4 – Download do Nagios e do Nagios *Plugins*

Agora para estar com a versão mais atualizada do Nagios, acesse o site do *software*¹. Através do comando abaixo efetue o *download* do Nagios *Plugins*.

```
# wget http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.15.tar.gz
```

● Passo 5 – Instalação dos *plugins* do Nagios

Após o *download* dos *plugins* do Nagios, o primeiro passo é a instalação dos mesmos.

Para descompactar o pacote e compilar segue os comandos abaixo.

```
# cd /home/allysson/nagiosd
# tar xzvf nagios-plugins-1.4.15.tar.gz
# cd nagios-plugins-1.4.15
# ./configure
# make
# make install
```

¹<http://www.nagios.org/download>

Agora é necessário dar as permissões necessárias ao usuário nos diretórios dos plugins.

```
# chown nagios.nagios /usr/local/nagios
# chown -R nagios.nagios /usr/local/nagios/libexec
```

- **Passo 6 – Download do Nagios NRPE**

O NRPE tem como função atuar como uma espécie de um agente, que executa o Nagios em uma máquina remota. Através do comando abaixo efetue o *download* do Nagios NRPE.

```
# cd /home/allysson/nagiosd
# wget http://downloads.sourceforge.net/project/nagios/nrpe-2.x/nrpe-2.12/nrpe-2.12.tar.gz
```

- **Passo 7 – Instalando o Nagios NRPE**

Se você já efetuou o *download* do Nagios NRPE, para iniciar a instalação é necessário descompactar e compilar. Para descompactar o pacote e compilar segue os comandos abaixo.

```
# tar xzf nrpe-2.12.tar.gz
# cd nrpe-2.12
# ./configure
# make all
# make install-plugin
# make install-daemon
# make install-daemon-config
# make install-xinetd
```

- **Passo 8 – Validando o Nagios NRPE**

Com o comando abaixo, valida-se a instalação do NRPE. A saída do comando deve apresentar apenas “NRPE v2.12”.

```
# /usr/local/nagios/libexec/check_nrpe -H localhost
```

- **Passo 9 – Instalando o Nagios NRPE no Xittd**

Agora é necessário saber o IP do servidor Nagios. Com o comando abaixo iremos confi-

gurar o Nagios NRPE no Xinetd.

```
# nano /etc/xinetd.d/nrpe
```

Então na diretiva “only_from”:

```
only_from = 127.0.0.1 <ip_do_nagios_server>
```

```
only_from = 127.0.0.1 10.81.117.144
```

Edite o arquivo “/etc/services”:

```
# nano /etc/services
```

Adicione a seguinte entrada:

```
nrpe 5666/tcp # NRPE
```

Reinicie o servido do Xinetd:

```
# /etc/init.d/xinetd restart
```

- **Passo 10 – Validando o Nagios NRPE sob Xinetd**

Com o comando abaixo, valida-se a configuração do NRPE rodando sob Xinetd. A saída do comando deve apresentar “tcp 0 0 *:nrpe *: LISTEN”.

```
# netstat -at | grep nrpe
```

- **Passo 11 – Adicionando plugins customizados no NRPE**

Para adicionar plugins customizado no Nagios NRPE, é necessário defini-lo em “nrpe.cfg”.

```
# nano /usr/local/nagios/etc/nrpe.cfg
```

Obs.: Na definição deste cliente foi utilizado o plugin “check_cpu”. Para instalar este plugin siga as informações abaixo.

Instalação do plugin “check_cpu”

O plugin “check_cpu” retorna o valor real do uso do processador da máquina. Este plugins está disponível sob a Licença Pública Geral, no site de seu desenvolvedor².

²<http://www.matejunkie.com/cpu-check-plugin-for-nagios/>

Faça o download do plugin e copie para pasta onde localizam-se os plugins do Nagios.

```
# cd /home/allysson/nagiosd
# wget https://www.monitoringexchange.org/attachment/download/Check-Plugins/Operating-Systems/check_cpu-sh/check_cpu.sh
# cp check_cpu.sh /usr/local/nagios/libexec/
```

Agora é necessário definir o comando “check_cpu”. A definição é feita editando o arquivo “nrpe.cfg”.

```
# nano /usr/local/nagios/etc/nrpe.cfg
```

Então ao final do arquivo inclui-se as seguintes linhas:

```
command[check_cpu]=/usr/lib/nagios/plugins/check_cpu.sh
$ARG1$ $ARG2$
```

Apêndice C

Instalação de Servidor e Cliente NFS

Este apêndice foi baseado em um tutorial já existente [62], e tem o objetivo de criar um tutorial que demonstra a instalação do protocolo NFS (*Network File System*) no GNU/Linux Debian Squeeze (versão 6.0.2).

Toda e qualquer modificação deve ser feita utilizando o super usuário ou root. Para não haver problemas na instalação e configuração siga todos os passos deste tutorial.

Instalação e configuração de Servidor NFS

- **Passo 1 – Instalação dos pacotes**

Inicialmente é necessário efetuar a instalação e configuração de pacotes para o servidor NFS.

```
# apt-get install nfs-kernel-server nfs-common
```

- **Passo 2 – Configuração do serviço**

Em seguida é necessário criar um diretório e alterar a permissão de acesso para a mesma. No exemplo o diretório criado é o /dados.

```
# mkdir /dados
# chmod 777 /dados
```

Para definir o diretório a ser compartilhando basta editar o arquivo /etc/exports. Lembrando que o diretório no exemplo é /dados, concedendo acesso a toda rede 10.81.X.X, com opções de gravação e fornecer o usuário como se fosse local.

```
# nano /etc/exports
```

```
# chmod 777 /dados
```

Exemplo: `/dados 10.81.117.144/255.255.0.0 (rw,no_root_squash)`

Pressione “Ctrl + O” para salvar, confirme o armazenamento das informações, e em seguida pressione “Ctrl + X” para sair.

Agora basta digitar o comando para exportar o diretório compartilhado.

```
# exportfs -a
```

ou

```
# /etc/init.d/nfs-kernel-server restart
```

Instalação e configuração de Cliente NFS

- **Passo 1 – Instalação dos pacotes**

Inicialmente é necessário efetuar a instalação e configuração de pacotes para o cliente NFS.

```
# apt-get install portmap nfs-common
```

- **Passo 2 – Configuração do serviço**

Para conseguir o acesso a pasta compartilhada pelo servidor NFS basta montar uma unidade. No exemplo iremos montar o diretório compartilhado no diretório local `/media/storage`, como no exemplo abaixo.

```
# mount -t nfs 10.81.117.144:/dados /media/storage
```

Caso haja a necessidade de montagem automática do diretório compartilhado, basta editar o arquivo `/etc/fstab`.

```
# nano /etc/fstab
```

E adicionar o código abaixo.

```
10.81.117.144:/dados /media/storage nfs rw, sync, auto 0 0
```

Pressione “Ctrl + O” para salvar, confirme o armazenamento das informações, e em seguida pressione “Ctrl + X” para sair.

Apêndice D

Instalação do KVM

Este apêndice foi baseado em documentações e tutoriais já existentes [63, 64, 65], e tem o objetivo de criar um tutorial que demonstra a instalação e configuração do KVM e de uma máquina virtual como exemplo, no GNU/Linux Debian Squeeze (versão 6.0.2).

Toda e qualquer modificação deve ser feita utilizando o super usuário ou root. Para não haver problemas na instalação e configuração siga todos os passos deste tutorial.

Instalação e configuração do KVM

- **Passo 1 – Verificação dos requisitos**

Inicialmente é necessário uma verificação para saber se o processador da máquina física suporta alguma tecnologia de virtualização, seja ela AMD-V ou Intel VT. Caso haja suporte as palavras “vmx” ou “smv” estarão destacadas.

```
# egrep '(vmx|svm)' -color=always /proc/cpuinfo
```

- **Passo 2 – Instalação dos pacotes**

Se o processador possui uma das tecnologias, inicia-se o processo de instalação dos pacotes “kvm”, “qemu-kvm”, “libvirt-bin”, “virtinst”, “virt-manager” e “bridge-utils”. O pacote “bridge-utils” é necessário para instalação da rede no modo *bridge*, ou seja, as máquinas virtuais irão pegar um IP (*Internet Protocol*) real na rede.

```
# apt-get install kvm qemu-kvm libvirt-bin virtinst  
virt-manager bridge-utils
```

- **Passo 3 – Adicionando usuário aos grupos**

Após a instalação é necessário adicionar o usuário do SO nos grupos. Abaixo temos como nome do usuário “allysson”.

```
# adduser allysson libvirt
# adduser allysson kvm
```

- **Passo 4 – Verificação da instalação**

Para verificar a instalação foi realizada corretamente, utiliza-se o seguinte comando.

```
# virsh -c qemu:///system list
```

- **Passo 5 – Configuração da rede *bridge***

Com todos os pacotes já instalados é necessário configurar a rede no modo *bridge*. Para efetuar a configuração é necessário editar as propriedades da rede.

```
# nano /etc/network/interfaces
```

Mantenha apenas o descrito abaixo neste arquivo. auto lo iface lo inet loopback

```
iface br0 inet dhcp
    bridge_ports eth0
```

```
auto br0
```

Pressione “Ctrl + O” para salvar, confirme o armazenamento das informações, e em seguida pressione “Ctrl + X” para sair.

- **Passo 6 – Reiniciando o serviço de rede**

Agora é necessário reiniciar as interfaces de rede.

```
# /etc/init.d/networkig restart
```

Para verificar se a rede está em modo *bridge* basta executar o comando “ifconfig”. O comando deve retornar as interfaces lo, eth0 e br0.

Criação e definição de uma máquina virtual

- **Passo 1 – Criando um disco rígido virtual**

Como exemplo, tem-se a máquina virtual com nome VM1. Inicialmente é necessário criar um disco rígido virtual, que pode ser criada através de um volume lógico ou utilizando o comando dd. Com o comando abaixo especifica-se um disco rígido virtual no diretório /media/storage/VM1/ com 10Gb (count=10240).

```
# dd if=/dev/zero of=/media/storage/VM1/VM1.img bs=1M
count=10240
```

- **Passo 2 – Criando uma máquina virtual**

A maneira mais simples, para definir uma máquina virtual, é utilizar um arquivo XML. Abaixo temos um exemplo de como uma máquina virtual pode ser definida, modificando o nome, uuid, memória RAM, o tipo de arquitetura utilizada, onde estão localizados o disco rígido virtual e a imagem para instalação do SO, e por fim definir a interface como bridge e br0.

Digite o comando abaixo para criar um arquivo “.xml”.

```
# nano /media/storage/VM1/VM1.xml
```

Agora insira as linhas abaixo modificadas conforme sua necessidade:

```
<domain type='kvm' >
  <name>VM1</name>
  <uuid>ca96c170-cb73-11e0-9572-0800200c9a66</uuid>
  <memory>512000</memory>
  <currentMemory>1677721</currentMemory>
  <vcpu>1</vcpu>
  <os>
  <type arch='x86_64' machine='pc'>hvm</type>
  <boot dev='cdrom' />
</os>
<features>
```

```

<acpi/>
<apic/>
<pae/>
</features>
<clock offset='localtime' />
<on_poweroff>destroy</on_poweroff>
<on_reboot>destroy</on_reboot>
<on_crash>destroy</on_crash>
<devices>
<emulator>/usr/bin/qemu-system-x86_64</emulator>
<disk type='file' device='disk'>
<source file='/media/storage/VM1/VM1.img' />
<target dev='hda' bus='ide' />
</disk>
<disk type='file' device='cdrom'>
<source file='/media/storage/ImagensDebian/debian-6.0.2.1-amd64-DV
<target dev='hdb' bus='ide' />
<readonly/>
</disk>
<interface type='bridge'>
<source bridge='br0' />
</interface>
<input type='mouse' bus='ps2' />
<graphics type='vnc' port='-1' autoport='yes'
keymap='pt-br' />
</devices>

</domain>

```

- **Passo 3 – Definindo uma máquina virtual**

Com todas as especificações da máquina virtual já escolhidas, agora basta definir a máquina virtual e iniciar a mesma.

```
# virsh define /media/storage/VM1/VM1.xml
```

- **Passo 4 – Iniciando uma máquina virtual**

Com a definição da máquina virtual já realizada, pode-se iniciar a execução da mesma.

```
# virsh start VM1
```

Com a máquina já definida e iniciada, pode-se utilizar o virt-manager para visualizar máquina virtual em operação. Como já instalamos o virt-manager para acessar ele basta ir no menu “Aplicativos > Sistema > Gerenciador de Máquinas Virtuais”.

Apêndice E

Configurando chaves de autenticação para SSH

Este apêndice foi baseado em um tutorial já existente [66], e tem o objetivo de criar um tutorial que demonstra a criação e configuração para uso de chaves de autenticação para SSH no GNU/Linux Debian Squeeze (versão 6.0.2).

Utilizando chaves de autenticação é possível aumentar a segurança através do SSH (Secure Shell). Isto se deve à aplicação de um par de chaves, uma chave pública, que será instalada nos servidores, e uma chave privada, que fica protegida por uma passphrase.

No exemplo abaixo têm-se 2 computadores: o cliente, que inicia a conexão com o servidor, e o servidor que recebe a conexão do cliente. A geração de chaves deve ser feita com o usuário que terá acesso via SSH, em nosso caso com o usuário root. Para não haver problemas na configuração siga todos os passos deste tutorial.

- **Passo 1 – Geração das chaves**

Para iniciar a configuração, no computador cliente é necessário gerar as chaves, que ficarão no diretório `/.ssh/`. Ele irá solicitar para entrar com a passphrase e os locais onde serão salvadas as chaves. Se o campo passphrase ficar em branco, não existirá passphrase. O arquivo `id_rsa`, contém a chave privada, e o `id_rsa.pub` a chave pública.

```
# ssh-keygen -t rsa
```

- **Passo 2 – Alterando permissões para acesso a chave privada**

Como a chave privada deve ser secreta, a permissão de acesso deve ser alterada.

```
# chmod 600 /root/.ssh/id_rsa
```

- **Passo 3 – Enviando chave pública para o servidor**

Após gerar as chaves e alterar a permissão, o próximo passo é enviar a chave pública para o servidor (10.81.118.60).

```
# ssh-copy-id -i /root/.ssh/id_rsa.pub 10.81.118.60
```

- **Passo 4 – Reiniciando o serviço**

Agora basta reiniciar o serviço SSH.

```
# /etc/init.d/ssh restart
```

- **Passo 5 – Validando as configurações**

Para testar a conexão basta efetuar uma conexão do cliente para o servidor.

```
# ssh 10.81.118.60
```

Para verificar se a chave pública foi enviada para o servidor, basta entrar com o comando abaixo no servidor, e em seguida verificar se existe a informação do usuário e IP do cliente.

```
# cat /.ssh/authorized_keys
```

Ao final deve conter informações a respeito do usuário e host que tem acesso autorizado no servidor.

Glossário

Inaudito Algo que nunca se ouviu dizer, de que não há exemplo
Rack Equipamento que armazena servidores

Referências Bibliográficas

- [1] VERAS, M. *Classificação dos DATACENTERS (TIA 942)*. 2009. Consultado na Internet em: 01/03/2011. Disponível em: <<http://datacenter10.blogspot.com/2009/01/classificao-dos-datacenters-tia-942.html>>.
- [2] EBBERS, M. et al. *The Green Data Center: Steps for the Journey*. 1. ed. Estados Unidos: Redpaper, 2008.
- [3] GREENPEACE. *Guide to Greener Eletronics*. Consultado na Internet em: 05/07/2011. Disponível em: <<http://www.greenpeace.org/international/en/campaigns/toxics/electronics/how-the-companies-line-up/>>.
- [4] PEDRINI, F. *Introduzione alla virtualizzazione con KVM*. Consultado na Internet em: 19/05/2011. Disponível em: <<http://www.miamammauslinux.org/2010/07/introduzione-alla-virtualizzazione-con-kvm/>>.
- [5] JONES, M. T. *Virtual Linux*. Consultado na Internet em: 09/05/2011. Disponível em: <<http://www.ibm.com/developerworks/library/l-linuxvirt/>>.
- [6] MEIER, S. *Ibm systems virtualization: Servers, storage, and software*. In: _____. 1. ed. Estados Unidos: Redpaper, 2008. cap. : Server virtualization techniques, p. 27–30.
- [7] VMWARE. *Virtualização grátis para servidores Windows e Linux*. Consultado na Internet em: 19/05/2011. Disponível em: <http://hotsite.cntbrasil.com.br/files/datasheet/server_datasheet_br.pdf>.

- [8] VMWARE. *Plataforma para virtualização de servidores, sistemas de armazenamento e rede*. Consultado na Internet em: 19/05/2011. Disponível em: <http://hotsite.cntbrasil.com.br/files/data-sheet/esx_datasheet_br.pdf>.
- [9] QUMRANET. *KVM: Kernel-based Virtualization Driver*. Consultado na Internet em: 16/05/2011. Disponível em: <http://www.linuxinsight.com/files/kvm_whitepaper.pdf>.
- [10] VERAS, M. *Datacenter: Componente central da infraestrutura de TI*. 1. ed. Rio de Janeiro: Brasport, 2009.
- [11] HARA, F. *Alternativas em refrigeração de datacenters: água gelada natural*. 2011. Consultado na Internet em: 22/01/2011. Disponível em: <<http://www.fabiohara.com.br/2011/01/04/alternativas-em-refrigeracao-de-datacenters-agua-gelada-natural/>>.
- [12] ROY, S.; BAG, M. Green computing - new horizon of energy efficiency and e-waste minimization - world perspective vis-à-vis indian scenario. In: *Emerging Technologies in E-Government*. Nova Deli: GIFT Publishing, 2008. p. 64–69.
- [13] ITCOMPANY. *Green IT*. 2011. Consultado na Internet em: 08/03/2011. Disponível em: <<http://www.itcompany.hu/greenit.html>>.
- [14] NODA, T. S. *Data center verde: redução de custos segue como principal motivador*. 2009. Consultado na Internet em: 27/11/2010. Disponível em: <<http://cio.uol.com.br/tecnologia/2009/03/09/data-center-verde-reducao-de-custos-segue-como-principal-motivador/>>.
- [15] GONSALVES, A. *Data center "verde" faz Yahoo economizar 40% com energia*. 2010. Consultado na Internet em: 27/11/2010. Disponível em: <<http://www.itweb.com.br/noticias/index.asp?cod=71980>>.
- [16] MEIER, S. *IBM Systems Virtualization: Servers, Storage, and Software*. 1. ed. Estados Unidos: Redpaper, 2008.
- [17] AGUILAR, F. P. de. *Tecnologia da Informação Verde: Uma Abordagem Sobre Investimentos e Atitudes das Empresas para Tornar Socialmente Sustentável o Meio Ambiente*.

- Dissertação (Monografia) — FATEC-ZL - Faculdade de Tecnologia Zona Leste, São Paulo - SP, 2009.
- [18] LAURINDO, F. J. B. et al. O papel da tecnologia da informação (ti) na estratégia das organizações. In: *Gestão & Produção*. Nova Iorque: [s.n.], 2001. v. 8, n. 2, p. 160–179.
- [19] LUCAS, T. dos S. *TI Verde: A Sustentabilidade na área tecnológica*. Dissertação (Monografia) — FATEC-ZL - Faculdade de Tecnologia Zona Leste, São Paulo - SP, 2010.
- [20] SCHIRNDING, Y. von; MULHOLLAND, C. Health in the context of sustainable development. In: *World Health Organization Meeting: Planning the Health Agenda for the WSSD*. [S.l.: s.n.], 2002.
- [21] MCT. *Ciência da Mudança do Clima*. Consultado na Internet em: 26/05/2011. Disponível em: <<http://www.mct.gov.br/index.php/content/view/3883.html>>.
- [22] AMERICANO, T. *Empresas colhem benefícios dos projetos de TI verde*. 2010. Consultado na Internet em: 21/06/2011. Disponível em: <<http://computerworld.uol.com.br/gestao/2010/06/22/empresascolhembeneficiosdos-projetosdetiverde/>>.
- [23] HORVATH, A.; MASANET, E. Enterprise strategies for reducing the life-cycle energy use and greenhouse gas emissions of personal computers. In: *Proceedings of the 2006 IEEE International Symposium on Electronics and the Environment*. San Francisco - California: [s.n.], 2006. p. 21–26.
- [24] PROTHERO, A.; FITCHETT, J. A. Greening capitalism: Opportunities for a green commodity. *Journal of Macromarketing*, Estados Unidos, v. 20, n. 1, p. 46–55, Junho 2000.
- [25] MARCAIS, T. Green computing – it is easy being green. In: *ASCUE*. North Myrtle Beach - Estados Unidos: [s.n.], 2008. p. 122–127.
- [26] TALEBI, M. *Computer Power Consumption Benchmarking for Green Computing*. Dissertação (Dissertação) — Villanova University, Villanova - Estados Unidos, 2008.

- [27] TALEBI, M.; WAY, T. Methods, metrics and motivation for a green computer science program. In: *ACM SIGCSE Bulletin - SIGCSE '09*. Chattanooga - Estados Unidos: [s.n.], 2009. p. 362–366.
- [28] DUNN, C. “Smart” Power Strips: Helping to Stop Idle Current Now! 2005. Consultado na Internet em: 12/07/2011. Disponível em: <http://www.treehugger.com/files/2005/12/smart_power_str.php>.
- [29] KEMP, W. H. *The Renewable Energy Handbook: A Guide to Rural Energy Independence, Off-Grid and Sustainable Living*. 1. ed. Tamworth - Canadá: Aztext Press, 2006.
- [30] KAZANDJIEVA, M. et al. *Software or Hardware: The Future of Green Enterprise Computing*. Palo Alto - Estados Unidos, Julho 2011.
- [31] CARINHAS, P. *Green Computing Guide*. [S.l.]: Fortuitous Technologies. Apostila.
- [32] SILVA, T. K. da. *T.I. Verde - Um Estudo de Aplicações e Impactos*. Dissertação (Monografia) — FATEC-ZL - Faculdade de Tecnologia Zona Leste, São Paulo - SP, 2009.
- [33] NANDA, S.; CHIUEH, T. *A Survey on Virtualization Technologies*. Consultado na Internet em: 10/05/2011. Disponível em: <<http://www.ecsl.cs.sunysb.edu/tr/TR179.pdf>>.
- [34] WILLIAMS, D.; GARCIA, J. *Virtualization with Xen: Including XenEnterprise, XenServer and XenExpress*. 3. ed. Burlington - USA: Syngress Publishing Inc, 2007.
- [35] SILVA, D. E. da et al. *Virtualização Como Alternativa para Ambiente de Servidores*. Dissertação (Monografia) — FATESG - Faculdade de Tecnologia Senai de Desenvolvimento Gerencial, Goiânia - GO, 2008.
- [36] JONES, M. T. *GNU/Linux Application Programming*. 2. ed. Boston - USA: Course Technology, 2008.
- [37] GOLDEN, B.; SCHEFFY, C. *Virtualization for Dummies*. 1. ed. Indianapolis - USA: Wiley Publishing Inc, 2008.
- [38] VMWARE. *Desafios e obstáculos da virtualização de x86*. Consultado na Internet em: 11/05/2011. Disponível em: <<http://www.vmware.com/br/technology/history.html>>.

- [39] ERIKSSON, J. *Virtualization, Isolation and Emulation in a Linux Environment*. Dissertação (Monografia) — UMU - Umea University, Suíça, 2009.
- [40] VMWARE. *VMware Workstation: Run Multiple OS Including Linux & Windows7, on Virtual Machines*. Consultado na Internet em: 03/05/2011. Disponível em: <<http://www.vmware.com/products/workstation/overview.html>>.
- [41] VMWARE. *VMware Player: Run Windows 7, Chrome OS - Free Download for a Virtual PC*. Consultado na Internet em: 03/05/2011. Disponível em: <<http://www.vmware.com/products/player/overview.html>>.
- [42] VMWARE. *VMware Fusion: execute o Windows 7 e o Chrome OS no Mac para a virtualização de desktops*. Consultado na Internet em: 03/05/2011. Disponível em: <<http://www.vmware.com/br/products/fusion/overview.html>>.
- [43] VMWARE. *VMware Server; Free Virtualization Download for Virtual Server Consolidation*. Consultado na Internet em: 03/05/2011. Disponível em: <<http://www.vmware.com/products/server/overview.html>>.
- [44] VMWARE. *VMware ESX Server*. Consultado na Internet em: 03/05/2011. Disponível em: <http://hotsite.cntbrasil.com.br/files/datasheet/esx_datasheet_br.pdf>.
- [45] VMWARE. *VMware ESXi and ESX Info Center: Migrate to ESXi Hypervisor*. Consultado na Internet em: 18/05/2011. Disponível em: <<http://www.vmware.com/products/vsphere/esxiandesx/overview.html>>.
- [46] MICROSOFT. *Compare Windows Virtual PC features*. Consultado na Internet em: 04/05/2011. Disponível em: <<http://www.microsoft.com/windows/virtual-pc/features/compare.aspx>>.
- [47] MURPHY, N. *An investigation into the development of an application for the remote administration of virtual machines using a mobile phone*. Dissertação (Monografia) — DIT - Dublin Institute of Technology, Dublin - Irlanda, 2008.
- [48] VIRTUALBOX. *Technical documentation*. Consultado na Internet em: 09/05/2011. Disponível em: <http://www.virtualbox.org/wiki/Technical_documentation>.

- [49] SILVA, R. F. da. *Virtualização de Sistemas Operacionais*. Dissertação (Monografia) — ISTCC - Instituto Superior de Tecnologia em Ciências da Computação, Petrópolis - RJ, 2007.
- [50] SANTOS, R. C. M. dos. Arquiteturas virtualizadas, comparação entre xen e kvm. Trabalho apresentado aos Profs. Andrea Schwertner Charão e Benhur de Oliveira Stein, como requisito parcial da disciplina de Arquiteturas Paralelas de Alto Desempenho do Curso de Mestrado em Ciência da Computação da UFSM. Julho 2008.
- [51] REDHAT. *KVM - Kernel Based Virtual Machine*. Consultado na Internet em: 16/05/2011. Disponível em: <<http://www.redhat.com/f/pdf/rhev/DOCKVM.pdf>>.
- [52] KVM. *Guest Support Status*. Consultado na Internet em: 16/05/2011. Disponível em: <http://www.linux-kvm.org/page/Guest_Support_Status>.
- [53] SWSOFT. *OpenVZ User's Guide Version 2.7.08*. Consultado na Internet em: 12/05/2011. Disponível em: <<http://ftp.openvz.org/doc/OpenVZUsersGuide.pdf>>.
- [54] COMPUTERWORLD. *Cinco tecnologias de virtualização com código aberto*. Consultado na Internet em: 12/05/2011. Disponível em: <<http://computerworld.uol.com.br/tecnologia/2011/01/20/cincotecnologiasdevirtualizacao-comcodigoaberto/>>.
- [55] KOLYSHKIN, K. *Virtualization in Linux*. Consultado na Internet em: 12/05/2011. Disponível em: <<http://ftp.openvz.org/doc/openvzintro.pdf>>.
- [56] BOTTINI, C. Iniciativa pioneira da univen petróleo virtualiza plataforma sap. *Asugnews*, São Paulo - SP, n. 50, p. 8–9, Janeiro-Fevereiro 2011.
- [57] PEAK10. *Benefits of a Virtualized Data Center*. Consultado na Internet em: 11/05/2011. Disponível em: <<http://www.peak10.com/resources/files/Peak10Benefitsofavirtualizeddatacenter.pdf>>.
- [58] CACIATO, L. E. *Virtualização e Consolidação dos Servidores do Datacenter*. 2011. Consultado na Internet em: 08/03/2011. Disponível em: <http://www.ccuec.unicamp.br:9000/CCUEC/database/images/Artigo_Virtualizacao_Datacenter.pdf>.

- [59] CARNEIRO, E. *Manual completo de instalação e configuração do Nagios 3 no Linux Debian 4.0 Etch*. 2008. Consultado na Internet em: 22/03/2011. Disponível em: <<http://www.vivaolinux.com.br/artigo/Manual-completo-de-instalacao-e-configuracao-do-Nagios-3-no-Linux-Debian-4.0-Etch/>>.
- [60] GALSTAD, E. *NRPE Documentation*. [S.l.], Maio 2007.
- [61] DRIZZO, T. *Nagios + NagiosQL + PNP4Nagios*. 2010. Consultado na Internet em: 05/04/2011. Disponível em: <<http://www.vivaolinux.com.br/artigo/Nagios+-NagiosQL+-PNP4Nagios?pagina=6>>.
- [62] BARBOSA, V. L. R. *NFS fácil no Debian Etch*. 2008. Consultado na Internet em: 19/05/2011. Disponível em: <<http://www.vivaolinux.com.br/dica/NFS-facil-no-Debian-Etch>>.
- [63] SOUSA, R. F. de. *KVM*. Consultado na Internet em: 14/06/2011. Disponível em: <<http://wiki.debian.org/KVM>>.
- [64] SOUSA, R. F. de. *Virtualização com KVM no Debian*. 2011. Consultado na Internet em: 05/08/2011. Disponível em: <<http://virtualizacao.ribafs.org/kvm/36-virtualizacao-com-kvm-em-debian>>.
- [65] API libvirt virtualization. *Documentation*. Consultado na Internet em: 24/08/2011. Disponível em: <<http://libvirt.org/docs.html>>.
- [66] MORIMOTO, C. E. *Dominando o SSH*. 2006. Consultado na Internet em: 16/09/2011. Disponível em: <<http://www.hardware.com.br/tutoriais/dominando-ssh/pagina5.html>>.