

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

**Analisando a utilização da virtualização em laboratório
de ensino**

Alessandro Rodrigo Franco

CASCABEL

2010

Alessandro Rodrigo Franco

Analisando a utilização da virtualização em laboratório de ensino

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. Marcio Seiji Oyamada

CASCADEL

2010

Alessandro Rodrigo Franco

Analisando a utilização da virtualização em laboratório de ensino

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. Marcio Seiji Oyamada(Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Aníbal Mantovani Diniz(Co-
Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Ademar Cezar Feil
Diretoria de Informática, UNIOESTE

Cascavel, 29 de novembro de 2010.

DEDICATÓRIA

Dedico este trabalho a todas as pessoas que me apoiaram no decorrer de minha vida.

Lista de Figuras

2.1 – Virtual Machine Monitor.....	06
2.2 – Tipo I.....	07
2.3 – Tipo II.....	08
2.4 – Tipo III.....	08
2.5 – Níveis de privilégio dos Processadores x86.....	09
2.6 – Diagrama de representação da virtualização total.....	11
2.7 – Exemplo do uso de memória na virtualização total.....	12
2.8 – Diagrama de representação da paravirtualização.....	14
2.9 – Diagrama de representação da paravirtualização.	15
3.1 – Diagrama de representação do VMWare Workstation.....	18
3.2 - Diagrama de representação do VMWare ESX Server.....	19
3.3 – Diagrama de representação do Virtual Server.....	20
3.4 – Diagrama de representação do Hyper-V.....	21
3.5 – Diagrama de representação do XenServer.....	22
3.6 – Resultado de Adams e Agesen.....	23
4.1 – Ambiente da UNIOESTE.....	30
4.2 – Arquitetura implantada.....	31
4.3 – Gráfico dos processadores com cinco usuários.....	32
4.4 – Gráfico dos processadores com dez usuários.....	32
4.5 – Gráfico da rede com cinco usuários.....	33
4.6 – Gráfico da rede com dez usuários.....	33
4.7 – Gráfico de memória RAM com cinco usuários.....	34
4.8 – Gráfico de memória RAM com dez usuários.....	34

Lista de Tabelas

2.1 – Comparação entre Paravirtualização e Virtualização total em relação aos requisitos de Popek e Goldberg.....	15
3.1 – Recursos disponíveis nas ferramentas de virtualização nas suas versões pagas ou gratuitas.	24
3.2 – Resultados obtidos dos testes feitos pelo Rick Vanover.....	25

Lista de Abreviaturas e Siglas

VM	Máquina Virtual
MMV	Monitor de Máquina Virtual
SO	Sistema Operacional
HVM	Suporte de máquina virtual em hardware
CPU	Processador
MMU	Unidade de gerencia de memória

Sumário

1. INTRODUÇÃO.....	1
1.1 HISTÓRIA DA VIRTUALIZAÇÃO	1
1.2 BENEFÍCIOS DA VIRTUALIZAÇÃO	2
1.3 OBJETIVO.....	3
1.4 ESTRUTURA DO TEXTO.....	3
2. VIRTUALIZAÇÃO	5
2.1 CONCEITOS	5
2.1.1 <i>Máquina Virtual (VM)</i>	5
2.1.2 <i>Monitor de Máquinas Virtuais (VMM)</i>	6
2.2 TIPOS DE MONITOR DE MÁQUINAS VIRTUAIS	7
2.2.1 <i>Tipo I (Nativo)</i>	7
2.2.2 <i>Tipo II (Hospedeiro)</i>	7
2.2.3 <i>Tipo III (Híbrido)</i>	8
2.3 ARQUITETURA X86	9
2.3.1 <i>Suporte em hardware para virtualização</i>	10
2.4 TIPOS DE VIRTUALIZAÇÃO	11
2.4.1 <i>Virtualização Total</i>	11
2.4.2 <i>Paravirtualização</i>	13
2.4.3 <i>Virtualização no nível do sistema operacional</i>	14
2.4.4 <i>Discussão</i>	15
3. LEVANTAMENTO DE SOLUÇÕES	17
3.1 VMWARE.....	17
3.1.1 <i>VMWare Workstation</i>	17
3.1.2 <i>VMWare Server ESX</i>	18
3.2 MICROSOFT	19
3.2.1 <i>Microsoft Virtual Server</i>	19
3.2.2 <i>Microsoft Hyper-V</i>	20

3.3 CITRIX XENSERVER.....	22
3.4 DISCUSSÃO	23
3.4.1 Funcionalidades.....	23
3.4.2 Desempenho	24
4. ESTUDO DE CASO	28
4.1 DEFINIÇÃO DO PROBLEMA	28
4.2 AMBIENTE	29
4.3 SOLUÇÃO	29
4.4 RESULTADOS	31
5. CONCLUSÃO.....	36
SCRIPTS EM VISUAL BASIC PARA CRIAÇÃO, COMPARTILHAMENTO E	
EDIÇÃO PERMISSÕES DE PASTAS DOS USUÁRIOS.....	37
SCRIPT PARA CRIAR AS PASTAS DOS ALUNOS:.....	37
SCRIPT PARA CRIAR AS PASTAS DOS PROFESSORES:	38
SCRIPT PARA COMPARTILHAR E EDITAR AS PERMISSÕES DE CADA PASTA CRIADA:	38
SCRIPT EM SHELL PARA OBTER MÉTRICAS DO XENSERVER.....	39
PERFLOG.SH	39
GLOSSÁRIO.....	41
REFERÊNCIAS BIBLIOGRÁFICAS.....	42

Resumo

O conceito de virtualização não é novo, surgiu inicialmente na década de 60 com a tentativa de melhorar o uso dos recursos nos *mainframes*. Devido ao grande aumento do poder computacional nas últimas décadas este conceito está ressurgindo, em novas arquiteturas e com novos recursos e desafios. A virtualização abrange vários domínios distintos seja em servidores, serviços, armazenamento e entre outros. O presente trabalho apenas aborda a tecnologia de virtualização no domínio dos servidores, mais precisamente servidores da arquitetura x86 ou compatíveis. O objetivo principal deste trabalho foi testar a eficiência do uso de virtualização em servidores da arquitetura x86 nos ambientes educacionais. Um caso especial, dos laboratórios de ensino do curso de Ciência da Computação da UNIOESTE foi descrito, e uma solução com o uso de virtualização para o mesmo foi avaliada. A solução proposta foi exposta a duas baterias de testes quantificando o uso de virtualização e acesso via terminal remoto ao servidor. Os resultados obtidos nestas baterias demonstraram que o uso de memória e de processadores ficou abaixo da capacidade máxima do servidor, além disto, foram avaliados os impactos indiretos benéficos desta solução no decorrer do ano de 2010. Em síntese, a solução proposta obteve um desempenho aceitável no uso dos recursos e encontraram-se alguns impactos benéficos não quantificáveis principalmente na gerência das máquinas.

Palavras-chave: Virtualização, Paravirtualização, Virtualização Total, Sistema Operacional, Máquina Virtual, Monitor de máquinas virtuais, Hypervisor.

Capítulo 1

1. Introdução

A virtualização é a tecnologia que está ganhando cada vez mais popularidade entre os profissionais da indústria de tecnologia de informação (TI) e entre seus executivos. A indústria concentra um grande esforço em segurança, em gerenciamento de informações e dispositivos para a produção de sistemas mais flexíveis, de modo a torna-los disponíveis aos usuários a qualquer tempo, em qualquer lugar. O conceito de virtualização abre espaço para a utilização real do poder computacional dos futuros computadores de uma forma flexível e confiável.

1.1 História da virtualização

O conceito de virtualização não é novo, iniciou-se na década de 60 com o conceito de tempo compartilhado (*time sharing*), e fortificou-se com o lançamento da arquitetura VM/370 da IBM (ROSENBLUM, GARFINKEL, 2005). Com esta arquitetura a IBM lançou pela primeira vez na história o conceito de virtual machine monitor (VMM), o qual era responsável em gerenciar a distribuição dos recursos do *hardware* entre as máquinas virtuais que estavam em execução. A grande expansão da virtualização aconteceu na década de 90, onde, muitas ferramentas comerciais foram lançadas. Em 1998, a empresa VMWare criou o primeiro produto de virtualização para plataformas x86, após seu sucesso estendeu seu produto entrando no mercado de virtualização de servidores. Outras empresas que se destacaram foram Virtual PC da Microsoft e VirtualBox da Sun.

Em meados de 2000, o laboratório de computadores da Universidade de Washington propôs um outro conceito de virtualização, com o seu projeto de monitor de máquinas virtuais para arquitetura x86 chamado de *Denali*. Em 2003, o laboratório de computadores da Universidade de Cambridge aderiu a este novo conceito com o *Xen*, seu projeto de monitor de máquinas virtuais. Entre 2005 e 2006, época que houve uma grande demanda por soluções de virtualização eficientes, ainda não era possível obter de forma significativa nas plataformas

x86. A Intel e a AMD, em paralelo, estenderam suas arquitetura no intuito de fornecer suporte a virtualização no nível de *hardware*. A Intel lançou a *Virtualization Technology*, conhecida como *Intel-VT*, e a AMD lançou *AMD-V*, formalmente conhecida como *Pacifica*. Este suporte no *hardware* é frequentemente chamado de *Hardware Virtual Machine* (HVM).

Em 2007, o *Xen* possuía uma fatia enorme do mercado de virtualização chamando a atenção da empresa Citrix, famosa por suas soluções de terminais e acessos remotos a servidores *Windows*, que acabou comprando-a e aperfeiçoando-a, criou-se então o *XenServer*, inicialmente tornando-a uma ferramenta comercial paga. Com o passar do tempo houve a necessidade de expandir o mundo da virtualização do *XenServer* para todos os campos de TI, tornando-a gratuita novamente. Em julho de 2008, a Microsoft deu seu grande passo no mundo da virtualização com o lançamento da versão final do seu produto conhecido agora como *Hyper-V*.

Durante todos esses anos de história, nota-se o surgimento de uma vicissitude de técnicas avançadas de abstração e de emulação, todas buscando manter os requisitos de segurança, desempenho e confiabilidade, primordiais para sua aceitação no mercado.

1.2 Benefícios da virtualização

A grande maioria das aplicações possui uma dependência no nível do sistema operacional, e as empresas/pessoas possuem uma dependência a inúmeras aplicações, o que implica na necessidade da utilização de vários sistemas operacionais. Outro grande problema é o antigo paradigma, onde, apenas um sistema operacional é executado sobre o mesmo computador. Estes dois problemas foram os combustíveis principais para o surgimento de uma nova tecnologia, a virtualização. Esta tecnologia apresenta um novo paradigma, onde, vários sistemas operacionais trabalham simultaneamente e isoladamente em um mesmo *hardware*, isoladamente pelo fato de um dos sistemas operacionais com algum comprometimento, seja em nível de aplicação quanto em nível de segurança, não implicar no comprometimento do todo.

Outro aspecto interessante de ser levado em consideração é a Lei de Moore, a qual informa que o número de transistores em um microprocessador dobra a cada 18 meses (Moore, 1965). Está lei é válida até hoje e possui como principal implicação o rápido crescimento do poder computacional tornando várias máquinas obsoletas em um período curto de tempo, surgindo à necessidade de mais gastos em poder computacional. A tecnologia de virtualização possibilita

a “reciclagem” destas máquinas obsoletas, economizando investimento em poder computacional. Segue um exemplo para melhor compreensão deste benefício, ao invés da empresa ter que renovar toda sua frota de computadores obsoletos poderia investir em apenas um de grande porte (servidor), no qual através da virtualização faria todo o trabalho, e utilizaria os computadores obsoletos para se comunicar com o servidor.

As empresas possuem inúmeros servidores, os quais executam apenas tarefas simples desperdiçando seu real poder computacional. A tecnologia de virtualização permite que estas inúmeras tarefas simples sejam executadas em um único servidor físico. Adotando esta solução com um bom planejamento a empresa reduz drasticamente seus custos com espaço físico, energia, resfriamento, entre outros. Esta economia de custos é o principal fator de a tecnologia estar ganhando cada vez mais força e mais investimento dos setores de tecnologia de informação (TI).

1.3 Objetivo

O objetivo deste trabalho é avaliar a eficiência do uso de virtualização em servidores da arquitetura x86 ou compatíveis nos ambientes educacionais, em especial em laboratórios de ensino de cursos de ciência da computação, onde para cada disciplina um conjunto de ferramentas e aplicações são necessárias.

Os aspectos mensuráveis que serão levantados neste trabalho são uso de CPU, uso de memória e uso de entrada e saída pelas máquinas virtuais. Outros aspectos, de forma mais subjetivas, levados em consideração serão facilidade, manutenção, organização, entre outros.

1.4 Estrutura do texto

O presente trabalho está dividido da seguinte forma:

- **Capítulo 2:** Serão levantados alguns conceitos da tecnologia de virtualização, suporte da arquitetura x86, tipos de máquinas virtuais e os tipos de virtualização, desempenho obtidos em outros trabalhos.
- **Capítulo 3:** Serão levantado alguns produtos comerciais de virtualização para a plataforma x86, como VMWare, Xen e outros. E alguns testes realizados para avaliar o desempenho dos citados.

- **Capítulo 4:** Será abordado o estudo do caso dos laboratórios de ensino da Universidade Estadual do Oeste do Paraná (UNIOESTE). Resolução de seu problema, e análise dos resultados obtidos.
- **Capítulo 5:** Será apresentado a conclusão referente a este trabalho.

Capítulo 2

2. Virtualização

2.1 Conceitos

O conceito de virtualização surgiu da necessidade das organizações terem de obter múltiplos sistemas computacionais para executarem as suas tarefas de negócio. Estas aquisições requeriam grandes investimentos e nem todas as empresas eram capazes de suprir esta necessidade. Com a virtualização esta necessidade pode ser suprida, permitindo que um sistema computacional execute isoladamente diversas cópias virtuais de si mesmo, conhecidas como máquinas virtuais.

Atualmente o domínio de atuação da virtualização não se limitava somente aos grandes servidores, expandiu-se de uma forma que abrangeu os computadores portáteis e os computadores pessoais. Os principais fatores que contribuíram para o crescimento do domínio foram o aumento do poder computacional, a queda dos preços e as pesquisas para inovar a tecnologia de virtualização.

Pode-se dizer que a tecnologia de virtualização é a simulação isolada de múltiplos ambientes virtuais (iguais ou não) em um mesmo ambiente físico, onde os mesmos disputam entre si a utilização dos recursos do ambiente físico (JACOMO, 2009). Uma melhor compreensão do conceito de virtualização pode ser vista traçando-se um paralelo entre virtualização e emulação. Emulação é a técnica de um sistema se passar por outro sistema, onde há uma tradução do código de um sistema para o código do outro sistema, enquanto que a virtualização é a técnica de um sistema se passar por dois ou mais sistemas iguais, ou seja, há uma replicação do sistema físico.

2.1.1 Máquina Virtual (VM)

O conceito de máquina virtual foi introduzido pela primeira vez pela IBM, na década de 60, definindo que uma máquina virtual é uma cópia totalmente protegida e isolada de um sistema físico (CREASY, 1980). Notamos que de uma maneira simplificada o termo genérico intitulado de máquina virtual é apenas um computador fictício criado por um sistema de simulação, ou seja, as máquinas virtuais são uma simulação física do sistema real. Laureano define que, *“A funcionalidade e o nível de abstração de uma máquina virtual encontram-se*

em uma posição intermediária entre uma máquina real e um emulador, de forma que os recursos de hardware e de controle são abstraídos e usados pelas aplicações”.

2.1.2 Monitor de Máquinas Virtuais (VMM)

O monitor de máquinas virtuais (*hypervisor* como é conhecido) consiste em uma camada adicional entre o *hardware* e as máquinas virtuais, com o intuito de prover e coordenar a multiplexação de recursos do sistema real entre as mesmas (máquinas virtuais). A ideia genérica do monitor de máquinas virtuais consiste em fornecer um processador virtual e versões virtuais dos dispositivos de entrada e saída, armazenamento, memória, entre outros para as máquinas virtuais e controlar o uso e o acesso a estes recursos (ROSE, 2004). A figura 2.1 mostra o retrato de uma arquitetura contendo um monitor de máquina virtual, que no caso está sendo chamado de *Virtual Machine Monitor*.

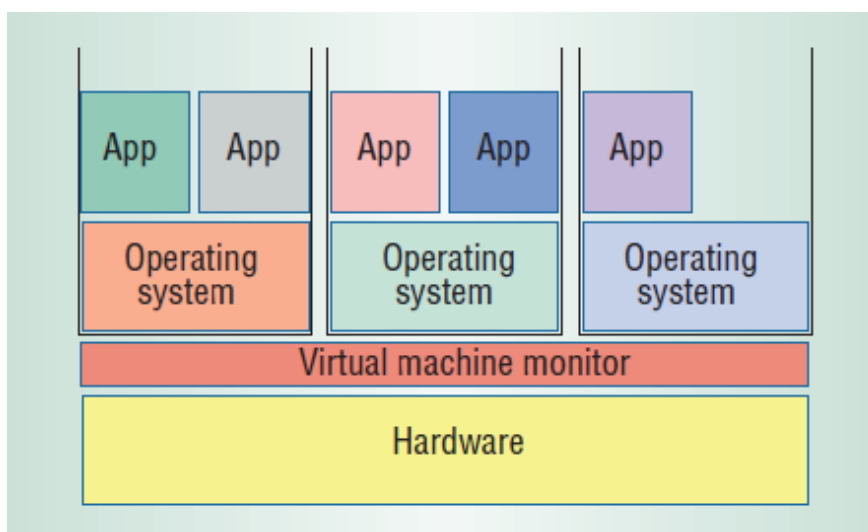


Figura 2.1 - Virtual Machine Monitor

Fonte: Virtual Machine Monitor: Current Technology and Futures Trends (ROSENBLUM, 2005)

Segundo Popek e Goldberg, “Um monitor de máquinas virtuais pode ser construído se o conjunto de instruções sensíveis para aquela arquitetura de computadores é um subconjunto das instruções privilegiadas da mesma”. Ou seja, toda vez que uma máquina virtual for executar uma instrução privilegiada a CPU deve parar e repassar o controle para o monitor, deste modo o mesmo decidirá se a CPU deverá ou não executá-la. Em 1974, Popek e Goldberg, generalizaram os requisitos que devem ser atendidos por um monitor de máquinas virtuais.

- **Equivalência:** Um sistema, executando sobre um monitor, deverá se comportar da mesma maneira que se comportaria se estivesse sendo executado diretamente no hardware.

- **Controle de recurso:** Durante todo o tempo, o monitor, deverá ter completo controle dos recursos que estão sendo virtualizados pelas máquinas virtuais.
- **Eficiência:** Um grande número de instruções deverá executar sem a intervenção do monitor.

2.2 Tipos de Monitor de Máquinas Virtuais

Em sua essência existem apenas dois tipos monitores de máquinas virtuais, as outras são derivações das mesmas.

2.2.1 Tipo I (Nativo)

A camada do monitor de máquinas virtuais se localiza entre o *hardware* e as máquinas virtuais, ocupando a posição de maior privilégio do sistema. Neste caso estas máquinas virtuais se comportam com uma máquina física completa, porém com menor privilégio e são frequentemente chamadas de *guests system* (sistema convidado). A principal vantagem deste tipo é que os *guests* podem acessar diretamente o hardware. As principais ferramentas que utilizam esta abordagem são o Microsoft Hyper-V e o XenServer da Citrix.

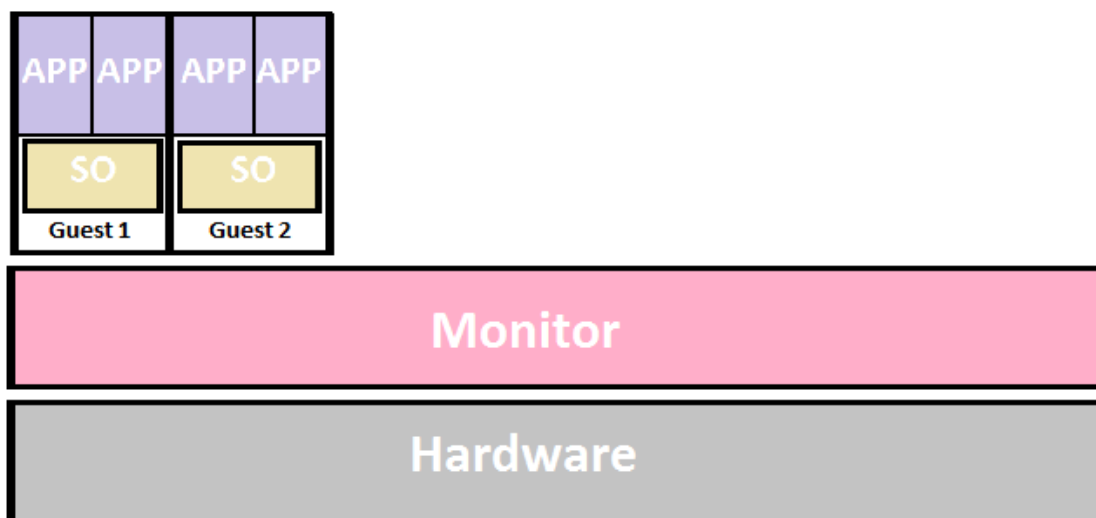


Figura 2.2 – Tipo I

2.2.2 Tipo II (Hospedeiro)

A camada do monitor de máquinas virtuais é um processo do sistema operacional adjacente, chamado de *host system*. O monitor possui o mesmo privilégio de uma aplicação comum no *host system* e os recursos oferecidos pelo monitor são fornecidos pelo *host*. Em

geral, utiliza-se um monitor diferente para cada nova máquina virtual. Exemplos de ferramentas que utilizam em suas implementações este tipo são o Microsoft Virtual Server e o VMWare Workstation.

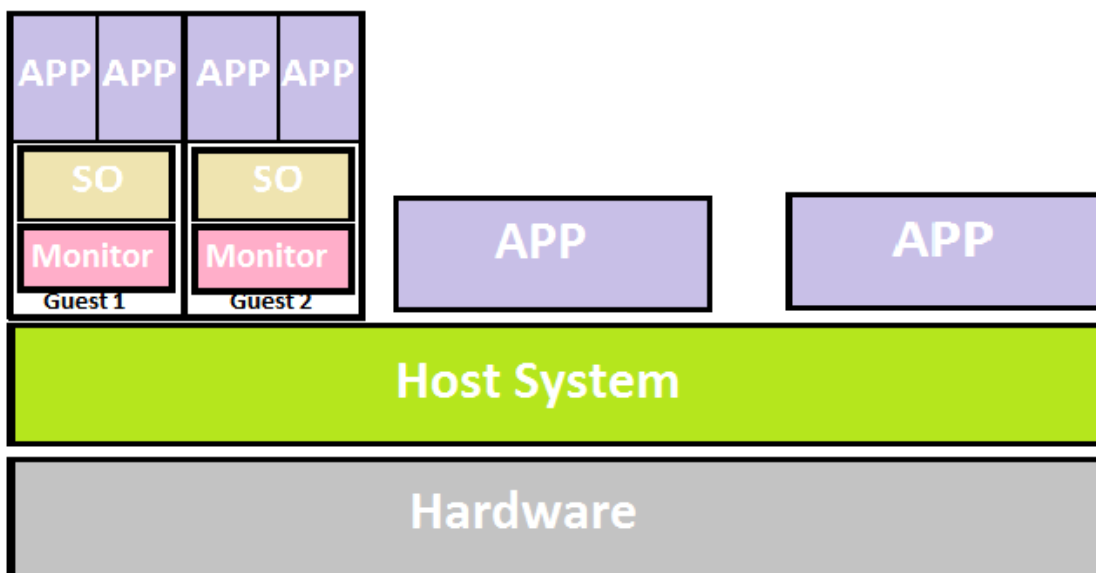


Figura 2.3 – Tipo II

2.2.3 Tipo III (Híbrido)

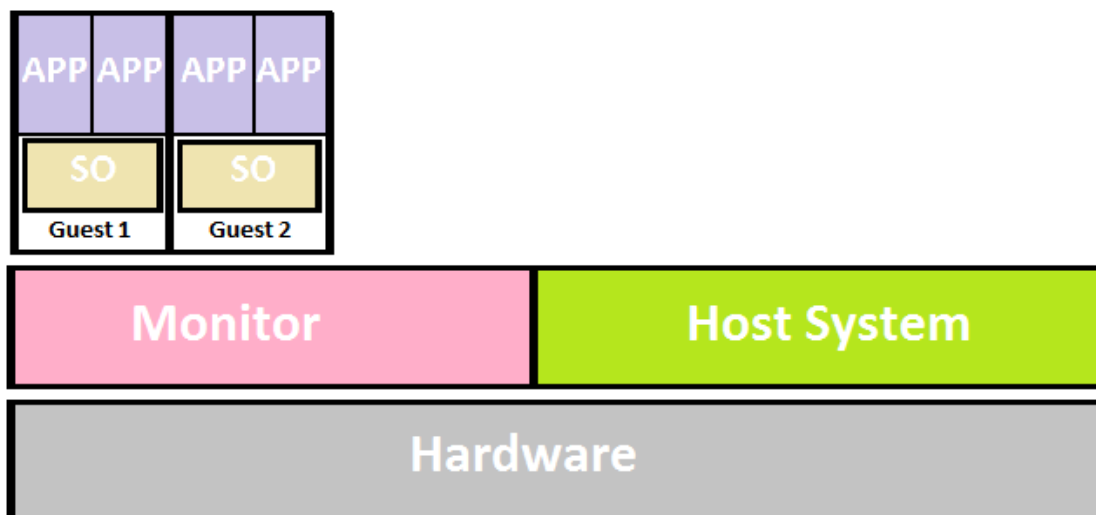


Figura 2.4 – Tipo III

As tecnologias de virtualização geralmente não implementam o tipo I e o tipo II no seu conceito, elas buscam um equilíbrio e algumas modificações entre os mesmos com o intuito de alcançarem um melhor desempenho. Geralmente a camada do monitor de máquinas virtuais é

executada com o sistema operacional *host* que ajuda a criar as máquinas virtuais (JACOMO, 2009). Um exemplo de ferramenta que utiliza este tipo de monitor é o VMWare ESX Server.

2.3 Arquitetura x86

Em 1978, a Intel lança seu microprocessador 8086, dando início a série da arquitetura x86. O que levou a seu sucesso foi o fato de processadores novos executarem as mesmas instruções dos antigos, ou seja, a troca de um processador antigo por um mais recente que seja não implica na reescrita das aplicações. Atualmente esta arquitetura e suas arquiteturas compatíveis são as que dominam o mercado desde computadores pessoais até os servidores.

Nesta arquitetura existe um pequeno conjunto de instruções, não privilegiada, que podem desestabilizar o sistema, este conjunto não é um problema para um único sistema operacional, entretanto ao utilizarmos a virtualização este conjunto se torna um problema gravíssimo e deve se ter maneiras de evita-lo (ROSE, 2004). Nota-se com isto que esta arquitetura não foi projetada para ser virtualizada e possui aspectos importantíssimos há serem levados em consideração na hora de projetar um sistema para virtualização.

Um exemplo de instrução extremamente sensível que deve ser simulada pelo monitor é a *popf*. Executada no modo privilegiado altera as *flags* da unidade lógica-aritmética (ALU) e as *flags* do sistema e executada no modo não privilegiado, a mesma (*popf*), apenas altera a *flag IF* do sistema. Nota-se que o comando possui duas interpretações distintas para o processador e em nenhum dos casos gera uma interrupção, desta maneira o monitor não consegue contornar este problema e o sistema poderá ser executado de uma forma inesperada.

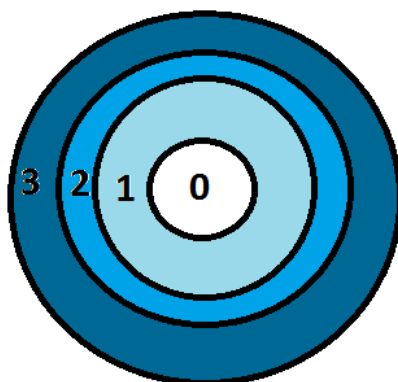


Figura 2.5 – Níveis de privilégio dos Processadores x86

Uma característica importante que deve se notar é a forma de proteção para a execução de um processo no modo protegido do processador. Neste modo, o processador, possui quatro

níveis de privilégios nomeados, na literatura, como *Ring 0* à *Ring 3*. Desta forma o processador sabe quais operações e quais acessos de memória a aplicação pode efetuar, gerando uma interrupção caso algo seja violado.

O *Ring-0* é o nível de maior privilégio do processador, usado pelo sistema operacional, e o de menor privilégio é o *Ring-3*, usado para executar as aplicações do usuário, os *Ring 1* e *Ring 2*, em geral, não são utilizados.

Robin e Irvine, definiram algumas características que devem ser utilizadas com intuito de implementar a virtualização nesta arquitetura.

- Instruções não sensíveis e não privilegiadas devem ser executadas diretamente pelo processador.
- Instruções sensíveis e privilegiadas devem ser emuladas pelo monitor de máquinas virtuais.
- Instruções sensíveis e não privilegiadas devem ser monitoradas pelo monitor de máquinas virtuais.

2.3.1 Suporte em hardware para virtualização

Os principais fabricantes da arquitetura x86 notaram a importância da virtualização em termos de competitividade e desenvolveram o suporte em *hardware* para a virtualização. Em 2006, tanto a Intel quanto a AMD introduziram no mercado as suas famílias de processadores com suporte a virtualização em *hardware* (HVM).

Visto que as duas foram desenvolvidas em paralelo, suas novas instruções não são iguais em termos sintáticos, mas são semanticamente parecidas. As duas adicionaram um *Ring -1*, de privilégio máximo, para o monitor de máquinas virtuais. Com esta adição tornou-se possível para o monitor, fiscalizar com maior eficiência as instruções sensíveis que antes ele não tinha controle.

Os sistemas operacionais, como visto anteriormente, são projetados para executar no *Ring 0*, e era necessário a modificação dos mesmos ou uma outra técnica complexa para serem virtualizados com confiabilidade. Com a adição deste novo *ring*, tornou-se possível a execução de sistemas não modificados, satisfazendo o requisito de equivalência de Popek e Goldberg.

2.4 Tipos de virtualização

Atualmente existem diversas propostas de virtualização, por exemplo: virtualização de servidores, virtualização de serviços, virtualização de *storage*, entre outras. Para cada propósito nós temos tipos distintos de virtualização, este trabalho se restringe a virtualização de servidores da arquitetura x86 ou compatíveis.

Muitos anos de pesquisa e desenvolvimento da tecnologia de virtualização se passaram neste domínio, surgindo quatro tipos diferentes de abordagem, são elas: virtualização total, paravirtualização, virtualização no nível do sistema operacional e virtualização nativa (WILLIAM, 2007).

A virtualização nativa é uma mistura entre virtualização total ou paravirtualização com técnicas de aceleração (WILLIAM, 2007). Esta combinação é utilizada pelas grandes ferramentas de virtualização do mercado que serão abordadas no Capítulo três.

2.4.1 Virtualização Total

É a técnica de virtualização que provê virtualmente todo o *hardware* para um *guest system*. Visto que este tipo providencia um sistema de hardware completo, um sistema projetado para executar nesta arquitetura de *hardware* implicará na execução do mesmo (sistema) no *hardware* virtualizado, sem que haja necessidade de haver algum tipo de modificação.

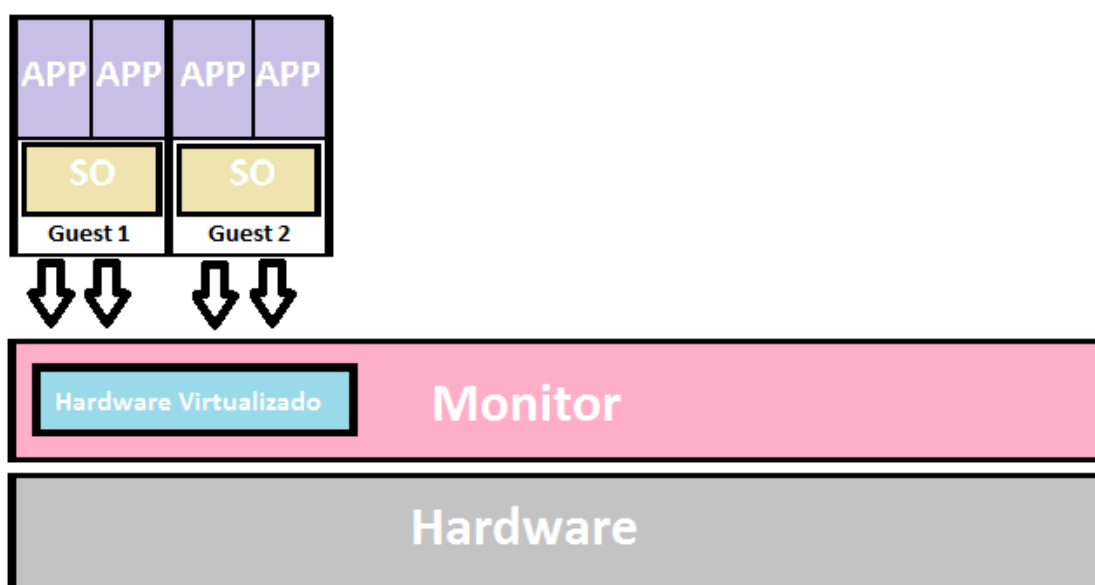


Figura 2.6 – Diagrama de representação da virtualização total

Antigamente este tipo de virtualização era um grande desafio na arquitetura x86, tendo em vista o problema do não suporte a virtualização, explicado anteriormente. Em 1998, a

VMWare conseguiu contornar este problema com a técnica tradução binária dinâmica (*dynamic binary translation*). Possibilitando o monitor avaliar, reorganizar e traduzir as sequências de instruções do *guest system* para uma nova sequência, evitando que algum *guest* corrompesse o requisito de isolamento entre máquinas virtuais. O principal defeito desta técnica é a perda de desempenho.

Com o surgimento das tecnologias de virtualização de hardware para a plataforma x86, o processador executa as instruções dos *guests* e no caso de executar uma instrução sensível o mesmo passa o controle para o monitor, responsável por emular esta instrução. Este recurso de hardware diminui o *overhead* de software causado pelo monitor, provendo um ganho significativo no quesito de desempenho do sistema pelo menos em teoria.

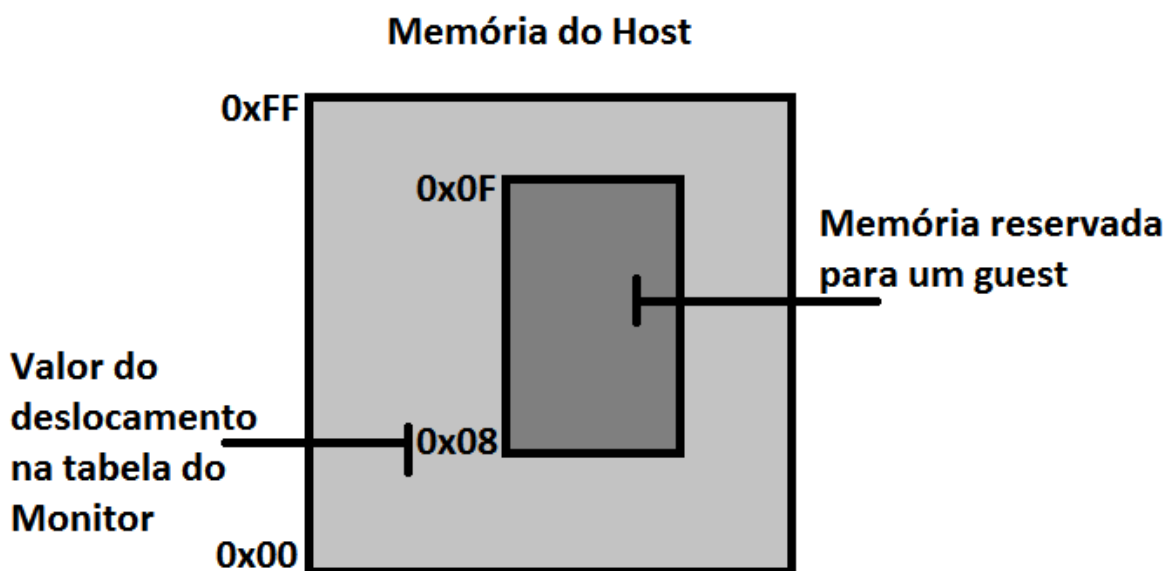


Figura 2.7 – Exemplo do uso de memória na virtualização total

Um fato de se preocupar é em relação à memória e ao uso do disco rígido. Os monitores que implementam este tipo de virtualização, devem pensar em uma maneira dos sistemas operacionais não modificados continuarem a acreditar que estão iniciados na posição 0 da memória física. A solução abordada pela grande maioria dos monitores é a mesma que os sistemas operacionais utilizam para enganar os processos na tradução de endereços lógicos para físicos, onde há uma tabela com endereços bases (deslocamento) de cada *guest* e os endereços de memória dentro do *guest* são posições relativas. Quando um *guest* for acessar uma posição da memória, o monitor consulta em sua tabela de deslocamento a quantidade de *bytes* que será deslocado para o mesmo acessar a posição efetiva na memória. A mesma

abordagem citada anteriormente segue-se para o disco rígido, como mostrado na figura a baixo.

Este tipo de solução tem um grande impacto no desempenho do sistema, devido ao fato de em geral causar ao menos duas falha de pagina (*page fault*) de memória. Os próximos passos do suporte em hardware para virtualização será no desenvolvimento de estratégias com o intuito de melhorar este desempenho.

Os *guest system*, em geral estão executando no *Ring 3*, onde não possuem privilégios para o acesso direto ao hardware. A tentativa de acesso a estes dispositivos gera uma interrupção, repassando o controle ao monitor, o qual a interpreta e simula o *hardware* real de acordo com a necessidade do *guest*. O grande problema é em relação ao desempenho do sistema, como visto, há simulação do *hardware*, ou seja, a um *overhead* de processamento.

Esta simulação pode ser contornada com as estratégias avançadas fornecidas pelo novo suporte de virtualização em *hardware*, um exemplo disto pode ser visto no caso de algum *guest* receber à liberdade para acessar diretamente o *hardware* de algum dispositivo sem a necessidade do monitor ter que emula-lo. Sem a necessidade de o monitor emular o *hardware* ocasiona um grande e positivo impacto no desempenho do sistema.

2.4.2 Paravirtualização

O fato de inúmeras empresas enfrentarem problemas, relacionados ao desempenho, na hora de implementar a virtualização total para a arquiteturas x86 e a comodidade no desenvolvimento de suporte a virtualização dos desenvolvedores desta arquitetura, foram o principal combustível para que outra solução fosse desenvolvida. O termo paravirtualização foi introduzido primeiramente pela Universidade de Washington, em meados de 2000, no seu monitor de máquinas virtuais chamado de Denali, logo após outros pesquisadores utilizaram o mesmo termo em seus projetos, como o Xen e VMware.

Esta técnica consiste, em sua essência teórica, de uma camada de software provendo uma interface otimizada entre o *hardware* e os sistemas *guests*, além disto, a emulação do *hardware* somente é feita nas partes mais críticas, enquanto que, nas outras a camada provê condições de acesso direto ao *hardware*, diminuindo a intervenção do monitor de máquinas virtuais significativamente. O resultado é um monitor bem menos complexo, se comparado aos implementados pela virtualização total.

Uma característica importante de se frisar é o fato dos sistemas operacionais dos *guests* devem estar modificados de forma a usarem esta interface com o monitor, mas isto não é um

grande problema se levarmos em conta o fato de ser muito mais fácil modificar um sistema operacional existente do que criar um grande e eficiente método de tradução binária.

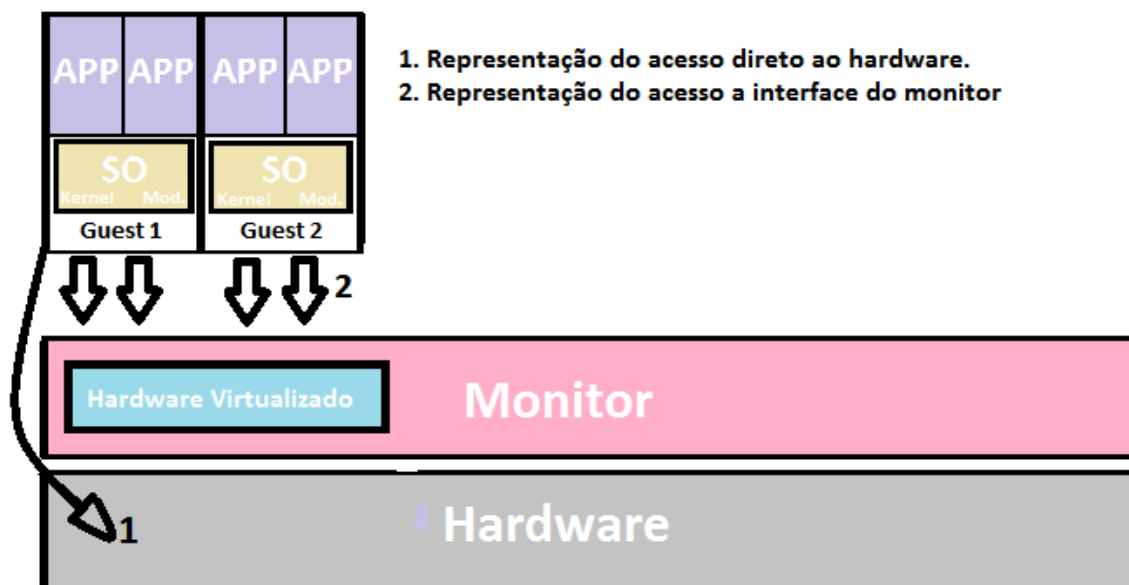


Figura 2.8 – Diagrama de representação da paravirtualização.

Este acesso direto ao *hardware*, feito pelos sistemas operacionais dos *guests* é baseado no fato dos sistemas operacionais já fazerem um rigoroso controle do acesso a estes. Com destaque o caso do controle de acesso à memória, um exemplo de modificação simples que pode ser feita neste caso é definir no início do *guest* as posições de memória que o mesmo pode acessar, deste modo fica a cargo do *guest* o controle de acesso à memória, que já está implementado nele.

Outro fato interessante de se notar é nos casos de acessos aos dispositivos menos perigosos como o mouse e o teclado, este acesso pode ser feito direto, sem a intervenção do monitor proporcionando grandes ganhos no desempenho.

2.4.3 Virtualização no nível do sistema operacional

Este conceito é um dos mais simples, é baseado em um *kernel* que permite várias instâncias do mesmo, sendo cada uma delas executadas em paralelo no mesmo *host*. A idéia principal deste conceito é simplesmente gerar uma cópia do sistema operacional do *host*, com o intuito de prover alguma espécie de isolamento. É importante frisar que é permitido apenas instância de um mesmo sistema operacional, tirando de fato o quesito de flexibilidade da virtualização.

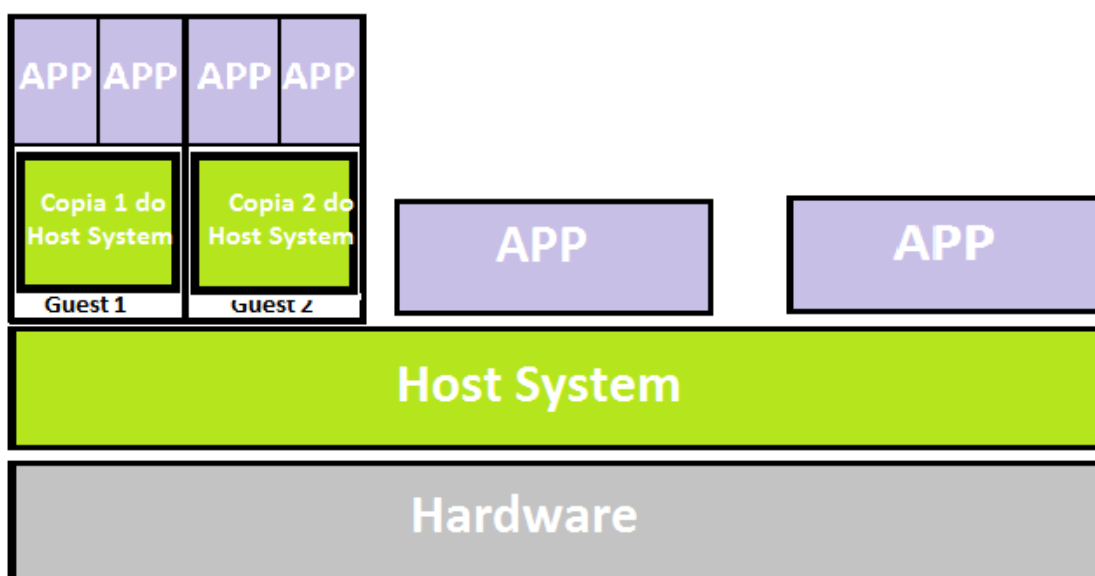


Figura 2.9 – Diagrama de representação da paravirtualização.

A principal diferença que se pode notar dos outros tipos de virtualização é que este tipo é um sistema operacional completo enquanto que os outros são implementados na forma de um *micro-kernel*.

2.4.4 Discussão

Nesta seção serão abordadas apenas a virtualização total e a paravirtualização, que são as essências das principais ferramentas de virtualizações fornecidas no mercado. É importante frisar, para não haver dúvidas, que a discussão será baseada nas duas formas de virtualização implementadas na plataforma x86 com suporte a máquinas virtuais.

Para iniciarmos nossa discussão, primeiramente devemos fazer uma análise e avaliar perante a satisfação dos requisitos levantados por Popek e Goldberg. No quadro abaixo, o X representa a satisfação do requisito.

Tabela 2.1 – Comparação entre Paravirtualização e Virtualização total em relação aos requisitos de Popek e Goldberg.

	Eficiência	Equivalência	Controle de Recursos
Paravirtualização	+	-	+
Virtualização Total	+-	+	+

Devo salientar que na paravirtualização o requisito de equivalência não foi satisfeito pelo motivo de prover uma melhor eficiência dos *guests*, no caso são modificados para usufruir da interface otimizada do monitor.

No caso da virtualização total o requisito de equivalência é satisfeito, mas notamos problemas no requisito eficiência. No futuro a tendência é suprir este problema de eficiência com adição de novas funcionalidades em hardware pelos desenvolvedores dos processadores.

Notamos o fato que os dois tipos possuem um pleno controle dos recursos, ou seja, sabem quando e quanto um recurso foi utilizado por um *guest*.

Capítulo 3

3. Levantamento de Soluções

Uma infinidade de produtos foi lançada durante todos esses anos, muitos destes foram totalmente esmagados pelos concorrentes e caíram no esquecimento outros sobrevivem até hoje. Esta sobrevivência se deve principalmente ao fato da escolha do tipo de virtualização e investimentos no intuito de melhorar a eficiência de seus produtos. Este capítulo faz um levantamento das características dos produtos das principais empresas de mercado da virtualização na plataforma x86 e apresenta resultados da comparação destes produtos levantados por outros autores.

3.1 VMWare

Atualmente a empresa VMWare, fundada em 1998, possui mais de 170.000 clientes espalhados pelo mundo. Obteve sucesso com o lançamento de seu primeiro produto comercial para virtualização na arquitetura x86. A empresa possui diversos produtos, gratuitos ou não, este presente trabalho destaca apenas os principais (VMWare Workstation e o VMWare ESX Server).

3.1.1 VMWare Workstation

Este produto utiliza a abordagem de um monitor de máquinas virtuais do tipo II (hospedeiro), podendo ser executado em qualquer ambiente (empresas ou computadores pessoais). A adoção deste conceito é fundamentada pelo fato da arquitetura x86 ser do tipo aberta para os periféricos, ou seja, existem diversos tipos de produtos para cada periférico e ficaria muito caro fornecer suporte a todos eles.

Devemos também considerar a característica de generalização dos sistemas operacionais atuais, os quais apresentam uma interface genérica não dependente de *hardware* para os programas utilizarem. O VMWare Workstation utiliza esta interface para disponibilizar os recursos de *hardware* do computador sem precisar se preocupar em desenvolver seus próprios drivers(SUGERMAN et al, 2001).

Um fato interessante de se notar é que este produto utiliza a virtualização total, em sua essência, mesmo em processadores que não suportam a virtualização em *hardware*. A empresa conseguiu solucionar este problema, introduzindo um interpretador que utiliza a técnica de tradução binária dinâmica.

Basicamente o interpretador separa o estado virtual do estado real do processador possibilitando a execução de instruções sensíveis ou privilegiados que não geram interrupções como, por exemplo, a *popf* citada na seção 2.3. Em síntese, a ideia é separar em blocos o código do *guest* em execução (denominada *guest on-demand*) e analisá-lo em busca de instruções que afetam o estado do sistema, caso não encontre o bloco é executado diretamente na CPU. No caso de haver uma instrução o monitor deve simular a execução da mesma.

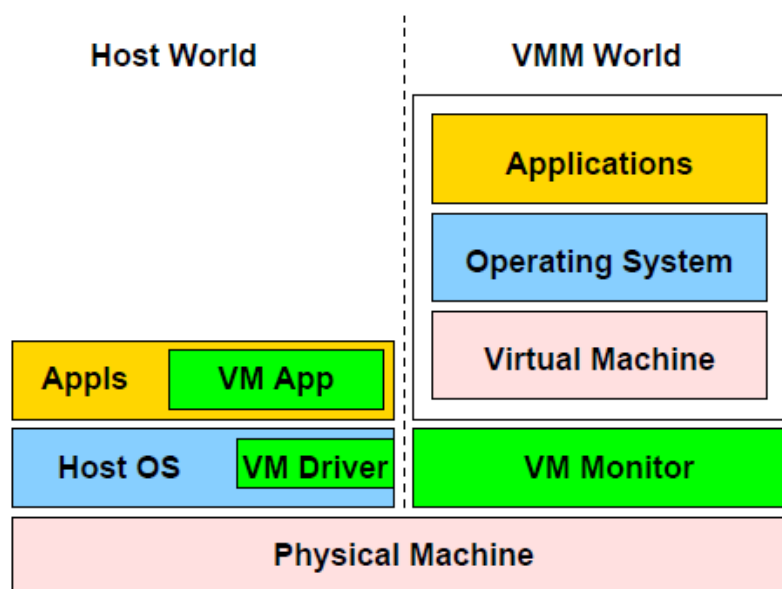


Figura 3.1 – Diagrama de representação do VMWare Workstation

Fonte: Virtualization I/O Devices on VMWare Workstation (SURGERMAN, 2001)

3.1.2 VMWare Server ESX

Diferente do VMWare Workstation, este produto utiliza a abordagem de um monitor de máquinas virtuais do tipo III (híbrido). Este produto ao invés de necessitar do sistema operacional do *host* conta com um *kernel* customizado (VMkernel).

Apesar de haver muita confusão o VMkernel não é o monitor de máquinas virtuais pelo simples fato de não executa diretamente as máquinas virtuais. Para cada máquina virtual iniciada o VMKernel inicia um monitor de máquinas virtuais, ou seja, existe um monitor diferente para cada máquina iniciada (AGESEN, 2009).

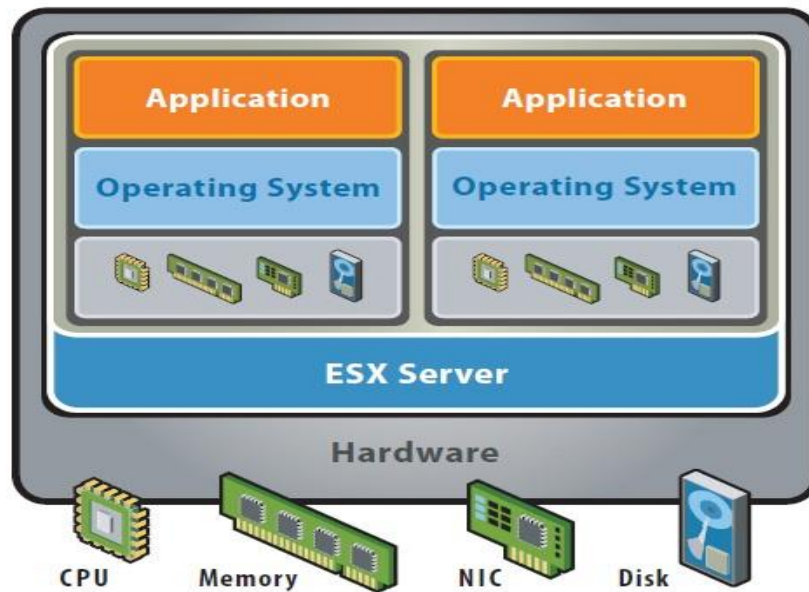


Figura 3.2 - Diagrama de representação do VMWare ESX Server

Fonte: Understanding Full Virtualization, Paravirtualization, and Hardware Assist (VMWare, 2007)

O produto trabalha com a virtualização total e a paravirtualização, ficando a cargo do administrador a escolha da abordagem que se comportará melhor em seu problema. O principal foco de atuação deste produto são os servidores.

3.2 Microsoft

A Microsoft, fundada em 1975, é uma multinacional da tecnologia da informação com produtos de sistemas operacionais, jogos, ambientes de desenvolvimento de software, entre outros que fogem do escopo deste trabalho. Neste texto destaca-se dois produtos em especial para o contexto do presente trabalho que são o Virtual Server e o Hyper-V.

3.2.1 Microsoft Virtual Server

Esta ferramenta é uma evolução do Virtual PC, a qual virtualizava o sistema operacional Windows na arquitetura PowerPC, para servidores. O Virtual Server é um serviço do Windows responsabilizado em iniciar uma *thread* para cada máquina virtual e seu monitor de máquinas virtuais gerencia e interage com as mesmas (máquinas virtuais).

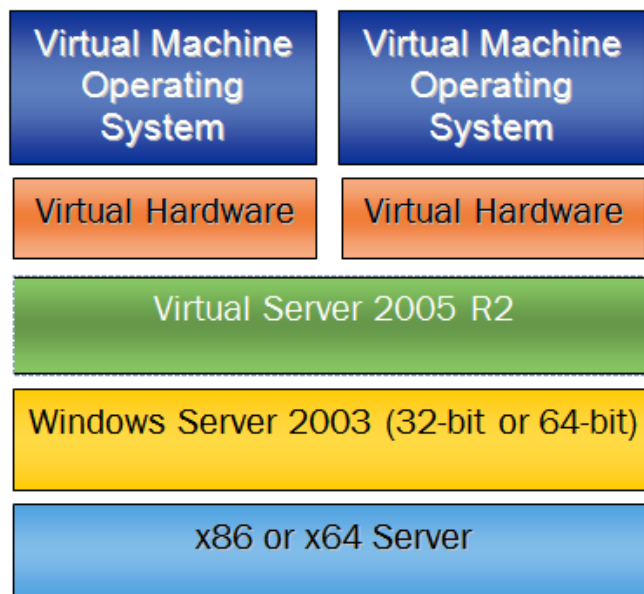


Figura 3.4 – Diagrama de representação do Virtual Server

Fonte: Microsoft Virtual Server 2005 R2 Technical Overview (MICROSOFT, 2005)

O Virtual Server não é o responsável em coordenar a utilização do processador e nem em prover acesso ao *hardware* para os *guests*. Estas tarefas ficam a cargo do sistema operacional do *host*, que no caso deve ser uma versão do sistema operacional Windows Server. Esta ferramenta utiliza a virtualização total com o tipo II de monitor de máquinas virtuais (hospedeiro).

3.2.2 Microsoft Hyper-V

Devido ao fato do Virtual Server possuir um baixo desempenho em relação às ferramentas de seus concorrentes a Microsoft resolveu reformulá-lo. Este baixo desempenho é causado principalmente pela arquitetura hospedeira, de fato ela possui um grande *overhead* de processamento. A solução foi mudar esta arquitetura e criar uma nova ferramenta chamada de Hyper-V.

Esta nova ferramenta utiliza o tipo III (Híbrido) de monitor de máquinas virtuais e deve conter uma máquina virtual chamada de *parent partition* com um sistema operacional da família Windows Server 2008 ou mais recentes. O Hyper-V pode usufruir de duas abordagens de virtualização que são a paravirtualização e a virtualização total, a escolha é automática baseada no tipo da *child partition*, como será mostrado adiante.

A *parent partition* é responsável pelo gerenciamento das outras máquinas virtuais (*child partition*) e o fornecimento do *hardware*, ou seja, os *drivers* dos periféricos que estão instalados na mesma podem ser acessados e usados pelas *child partition*. É importante frisar

que o escalonamento do processador e a memória física serão feitos pelo monitor e não pela *parent partition*.

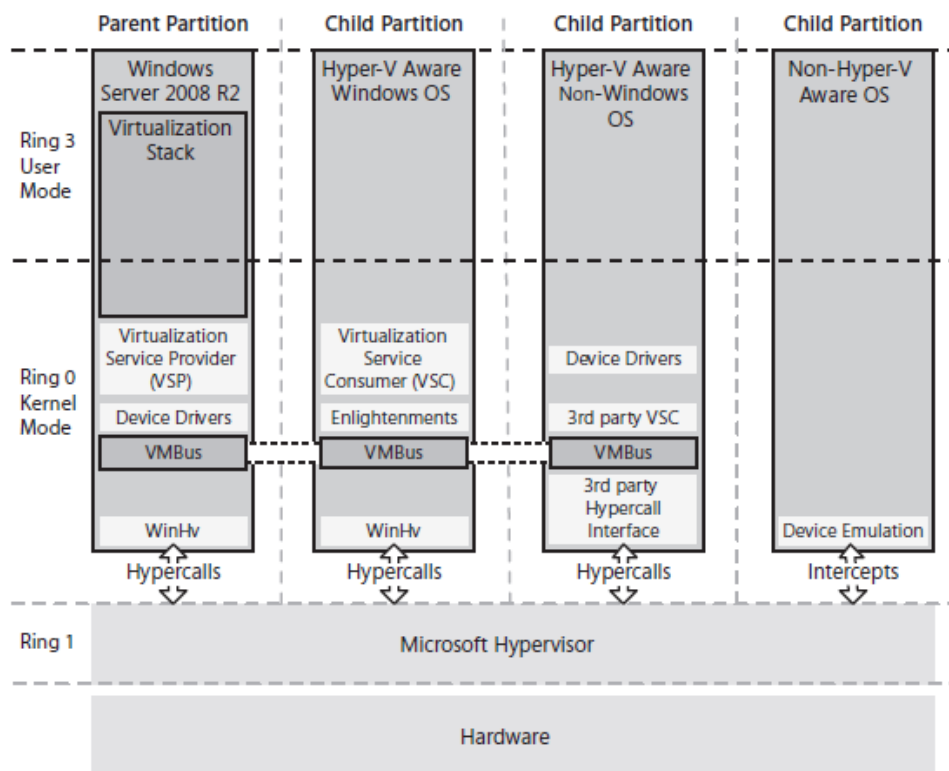


Figura 3.5 – Diagrama de representação do Hyper-V

Fonte: Understanding Microsoft Virtualization Solution (TULLOCH, 2010)

Esta ferramenta possibilita o suporte a três tipos distintos de *child partition*, abordados a seguir:

- *Child Partitions Hosting a Hyper-V-Aware Windows Operation System:* Neste tipo de máquina virtual utiliza-se drivers otimizados para execução no ambiente do Hyper-V.
- *Child Partitions Hosting a Hyper-V-Aware Non-Windows Operation System:* Neste tipo de máquina virtual também são utilizados espécies de *drivers* otimizados para execução no ambiente do Hyper-V, ou seja, utiliza-se a abordagem de paravirtualização.
- *Child Partitions Hosting a Non-Hyper-V-Aware Operation System:* Neste tipo de máquina virtual utiliza-se a emulação de *hardware*, perdendo drasticamente a performance perante as outras máquinas virtuais.

3.3 Citrix XenServer

O mundo da tecnologia de informação conheceu o Xen em 2003, desenvolvido pela Universidade de Cambridge e adquirido pela empresa Citrix em 2007. A Citrix distribui sua solução principalmente para servidores, chamada de XenServer.

O monitor de máquinas virtuais no XenServer é do tipo III (Híbrido), ou seja, é uma camada fina entre o *hardware* e os sistemas operacionais. No Xen cada máquina virtual é chamada de *DomainN*, onde N deve ser um número inteiro positivo maior que zero e único. A sua ideia básica é a simplicidade, retirando a responsabilidade do monitor de possuir *drivers* para os periféricos do computador e repassando-a para um domínio especial chamado de *Domain0*.

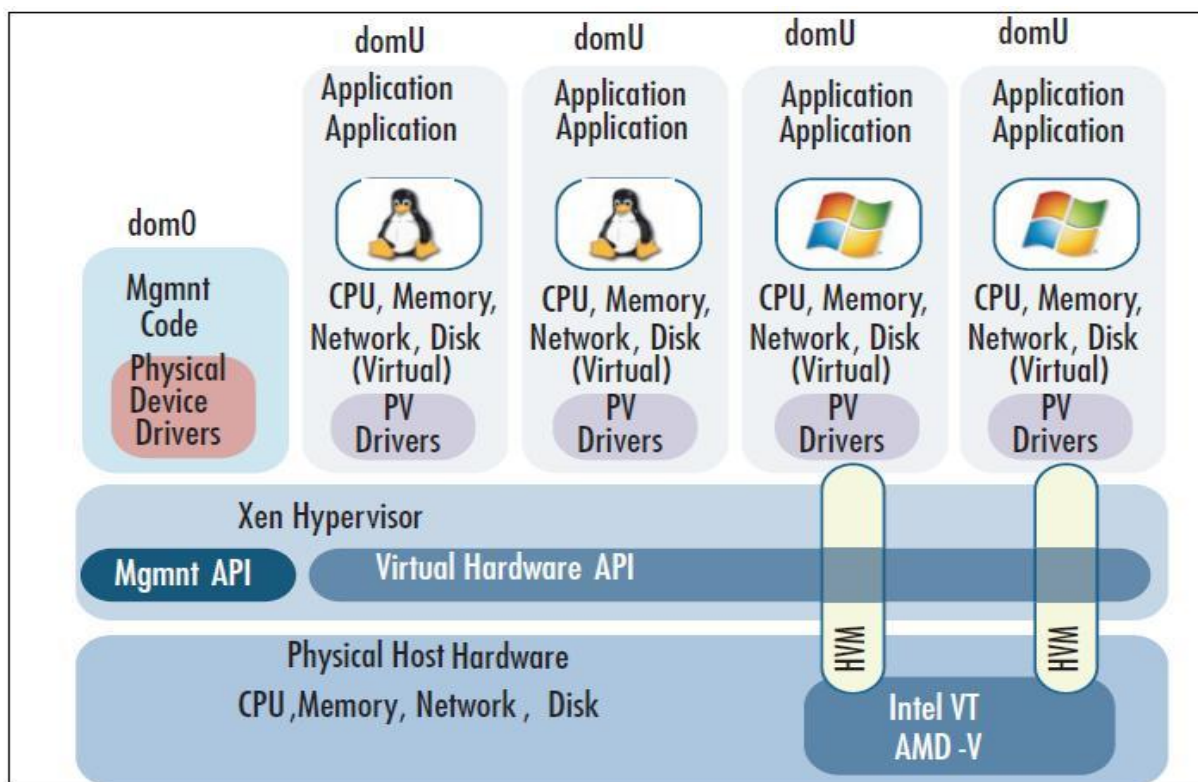


Figura 3.6 – Diagrama de representação do XenServer

Fonte: Virtualization With Xen (WILLIAM, 2007)

Assim como no Microsoft Hyper-V o XenServer deve possuir uma máquina virtual capaz de fornecer um gerenciamento dos demais *domains* e fornecer um maneira de acesso aos periféricos para os mesmos (*domains*), esta máquina é intitulada de *domain0*. Este *domain* é o sistema de maior privilégio do computador possuindo acesso direto a todos os recursos de *hardware* e fornecendo ferramentas administrativas para manipulação dos outros *domains*

(BARHAM *et al.*, 2003). O XenServer já possui um *kernel* customizado para estes objetivos, diferente do seu antepassado que era necessário a instalação de um *kernel* compatível (Linux, FreeBSD, etc...).

O XenServer suporta dois tipos de virtualização a paravirtualização e a virtualização total. A paravirtualização somente é possível em sistemas operacionais modificados para usufruir dos benefícios fornecidos pela API do XenServer. A virtualização total permite a execução de sistemas operacionais não modificados, este caso somente é usado em processadores com suporte de máquinas virtuais em *hardware*.

3.4 Discussão

Nesta seção será apresentado um comparativo das funcionalidades das diversas versões das principais ferramentas abordadas neste trabalho: VMWare ESX; Microsoft Hyper-V e Citrix XenServer. Além disto, serão apresentados alguns resultados obtidos por outros pesquisadores.

3.4.1 Funcionalidades

Em geral, a tecnologia de virtualização diminuiu drasticamente o acoplamento entre a camada dos sistemas operacionais com a camada de hardware. Este baixo acoplamento foi fundamental para a funcionalidade de *Live Motion*, a qual permite migrar uma máquina virtual em execução de um servidor a outro sem a necessidade de interrompê-la. Todas as ferramentas comerciais levantadas neste trabalho, nas suas versões gratuitas, possuem esta funcionalidade, entretanto é importante salientar que as máquinas virtuais devem ser instaladas com algumas características que variam de acordo com a ferramenta adotada. Por exemplo, no XenServer é necessário que máquina esteja num *storage* compartilhado no *pool* de servidores.

A empresa VMWare disponibiliza apenas nas suas versões pagas, o serviço de *Storage Motion*, o qual permite a migração de *storage* entre os *datas centers*. Desta forma, consegue-se migrar uma máquina virtual entre os diversos *data centers* espalhados pelo mundo, sem a necessidade de interrompê-la.

A funcionalidade *Live Motion* serviu de base para um serviço oferecido nas versões pagas destas empresas, chamada de Alta Disponibilidade. Este serviço permite a continuidade de execução das máquinas virtuais após a queda de um servidor. Em síntese, os servidores são

monitorados constantemente de forma que se algum venha a falhar, as máquinas serão migradas para outro servidor disponível.

Outro recurso que utiliza *Live Motion* é o balanceamento de carga, o qual tende a evitar que um servidor seja sobrecarregado de serviços. Este recurso implementado de forma automática está disponível apenas nas versões pagas das empresas Citrix e VMware, a ferramenta da Microsoft permite a distribuição dos recursos de forma manual.

A recuperação/*backup* é outro serviço disponível nas versões gratuitas das ferramentas citadas no texto. Todas permitem salvar o estado de uma máquina virtual, desta forma a máquina poderá ser restaurada no caso de alguma falha, é importante frisar que a restauração é feita a partir do estado salvo e todas as alterações feitas depois deste momento são perdidas.

Tabela 3.1 – Recursos disponíveis nas ferramentas de virtualização nas suas versões pagas ou gratuitas.

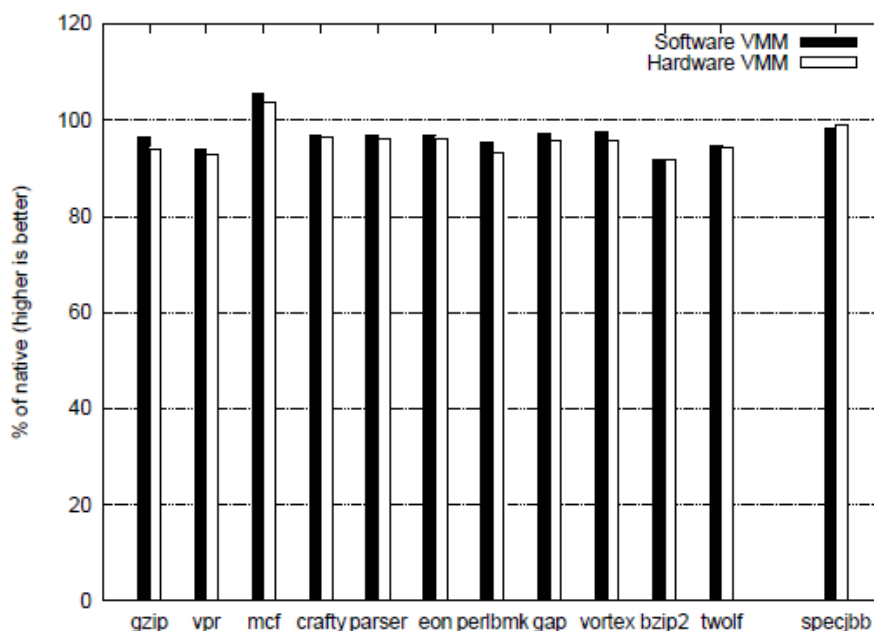
	Citrix XenServer		Microsoft Hyper-V		VMWare ESX	
	Gratuita	Paga	Gratuita	Paga	Gratuita	Paga
Live Motion	X	X	X	X	X	X
Storage Motion	-	-	-	-	-	X
Alta Disponibilidade	-	X	-	X	-	X
Balanceamento de carga(Automático)	-	X	-	-	-	X
Recuperação/Backup	X	X	X	X	X	X

De forma geral as empresas disponibilizam praticamente os mesmos serviços em suas versões gratuitas. Nota-se que as verdadeiras diferenças nos recursos disponíveis aparecem em suas versões pagas. A empresa VMware é a que se destacou perante as demais devido ao fato de oferecer uma gama muito maior de serviços que as outras ainda não implementaram.

3.4.2 Desempenho

O artigo “*A comparison of software and Hardware Techniques for x86 Virtualization*” de Adams e Agesen, ambos membros da empresa VMware, teve com objetivo testar a diferença das duas formas de implementar a virtualização total, implementada em software com a implementada utilizando as facilidades oferecidas pelo suporte em *hardware* para virtualização. A conclusão deste trabalho foi que a virtualização total em software, apesar de todo o *overhead* de processamento, ainda é mais eficiente que a em *hardware*. Concluíram que isto ocorreu por dois motivos principais, o primeiro é o fato de não oferecer um suporte a

virtualização da unidade de gerenciamento de memória (MMU) e o segundo foi de não ser adequado para a utilização de técnicas existentes de virtualização da MMU. A figura 3.6 demonstra os resultados obtidos.



SPECint 2000 and SPECjbb 2005.

Figura 3.9 – Resultado de Adams e Agesen.

Fonte: A comparison of software and Hardware Techniques for x86 Virtualization (ADAMS, AGESEN, 2006)

Rick Vanover (Consulte a referência), em 2009, testou e comparou as três principais ferramentas do mercado (VMWare ESX Server, Microsoft Hyper-V, Citrix XenServer). Os testes foram feitos em um Dell PowerEdge 2950 2x2 @ 3.0Ghz, 16GB RAM, 360GB armazenamento local. A instalação foi a padrão das ferramentas com os *guest* rodando Microsoft Windows Server 2003 com SQL Server 2005. A ferramenta PassMark foi utilizada para o levantamento das médias das operações de CPU, RAM e no caso dos Jobs foi usado o tempo de início e o tempo de finalização de um job. Os testes foram divididos em três baterias distintas, a primeira com seis *guests* com intenso trabalho, a segunda com doze *guests* com intenso trabalho e a terceira com doze *guests* e baixo trabalho. Os resultados obtidos das baterias 1, 2 e 3 estão expressos na tabela a seguir.

Tabela 3.2 – Resultados obtidos dos testes feitos pelo Rick Vanover.

Bateria 1			
	Hyper-V	XenServer 5	Server ESX 3.5

Média das Operações de CPU por <i>guest</i>	29 bilhões por hora	36,5 bilhões por hora	29,5 bilhões por hora
Média das operações de memória RAM por <i>guest</i>	108,83 bilhões por hora	112 bilhões por hora	88,5 bilhões por hora
Média dos tempos de completar um SQL Job por <i>guest</i>	4 minutos 40 segundos	4 minutos 26 segundos	5 minutos 8 segundos
Bateria 2			
	Hyper-V	XenServer 5	Server ESX 3.5
Média das Operações de CPU por <i>guest</i>	8,67 bilhões por hora	8,75 bilhões por hora	3,67 bilhões por hora
Média das operações de memória RAM por <i>guest</i>	31,75 bilhões por hora	26,67 bilhões por hora	10 bilhões por hora
Média das operações de Disco por <i>guest</i>	416 milhões por hora	583 milhões por hora	667 milhões por hora
Média dos tempos de completar um SQL Job por <i>guest</i>	6 minutos	8 minutos 3 segundos	11 minutos 28 segundos
Bateria 3			
	Hyper-V	XenServer 5	Server ESX 3.5
Média das Operações de CPU por <i>guest</i>	5 bilhões por hora	3,75 bilhões por hora	7,08 bilhões por hora
Média das operações de memória RAM por <i>guest</i>	1,08 bilhões por hora	1,25 bilhões por hora	1,25 bilhões por hora
Média das operações de Disco por <i>guest</i>	167 milhões por hora	187 milhões por hora	187 milhões por hora
Média dos tempos de completar um SQL Job por <i>guest</i>	4 minutos 43 segundos	5 minutos 34 segundos	5 minutos 34 segundos

O autor (Rick Vanover) conclui que em ambientes com intenso uso de memória e processador as ferramentas da Citrix e da Microsoft são duas ótimas soluções, e nos casos de muitas máquinas virtuais com poucos uso de recursos a ferramenta da VMWare é uma ótima escolha. Outro aspecto que o autor deixa claro é que não se deve subjugar a VMWare pelo fato da bateria de testes não levar em conta outras facilidades da ferramenta como a VMotion, recursos de gerenciamento, entre outras.

Capítulo 4

4. Estudo de caso

Neste Capítulo será descrito os experimentos realizados nos laboratórios de ensino do curso de Ciência da Computação da UNIOESTE. O intuito do estudo de caso é avaliar o impacto da utilização de virtualização na resolução de alguns problemas destes laboratórios.

4.1 Definição do Problema

A hora de preparar as máquinas dos laboratórios para uma aula específica sempre foi a causa de muito transtorno para equipe da coordenação dos laboratórios. Destaca-se o caso dos mesmos terem que preparar um *kernel* específico para a execução de testes em alguma disciplina. Por mais eficiente que seja a equipe, estima-se que este trabalho levaria pelo menos dois dias para esta preparação. Logo após a realização dos testes citados as máquinas deverão voltar a sua rotina normal de trabalho, ou seja, a equipe deverá restaurar as máquinas para o antigo padrão, levando alguns dias para realização deste processo.

Outro problema levantado é na manutenção das máquinas, como o caso da instalação e da preparação dos sistemas para o uso diário. Por exemplo, um laboratório com vinte máquinas levaria mais de uma semana de serviço. Uma situação que piora ainda mais este caso é o fato de que cada professor tem preferência a um sistema específico para sua disciplina implicando no desagrado de alguns professores caso os sistemas não estejam de acordo com sua preferência.

Outro fator que deve ser levado em consideração é a grande necessidade de processamento por parte das novas tecnologias, tendo em vista que a instituição é pública e a troca anual de equipamentos é praticamente impossível principalmente por razões burocráticas e econômicas.

A virtualização tende a diminuir estes problemas significativamente principalmente pelas razões levantadas nos capítulos anteriores a respeito desta tecnologia. Este presente trabalho tem como o objetivo proporcionar uma maneira de empregar a virtualização neste contexto e tentar quantificar os benefícios gerados diretamente e indiretamente.

4.2 Ambiente

O ambiente, como dito anteriormente, será os laboratórios de ensino do curso de Ciência da Computação na UNIOESTE. O equipamento a ser utilizado é um servidor HP (2 Processadores Intel Xeon E5405, 16GB RAM, 320 GB disco rígido) . Cada processador contém dois núcleos, e cada núcleo contém duas threads, totalizando oito threads disponíveis no sistema.

A UNIOESTE, por razões de segurança, possui um sistema de *login* de usuário centralizado e as máquinas virtuais devem estar de acordo com esta política. Em síntese, os *guests* não possuem cadastro local de usuários e o controle de acesso às máquinas é feito por um servidor.

Outro ponto interessante a destacar é a possibilidade dos usuários possuírem um diretório para armazenamento de seus arquivos. Este diretório poderá ser acessado por qualquer *guest*, ou seja, o aluno terá acesso a sua pasta não importando qual sistema operacional o mesmo está utilizando no momento.

4.3 Solução

No final do ano 2009 e início do ano de 2010 foi feito um levantamento das principais ferramentas disponíveis no mercado para virtualização da arquitetura x86, destacando-se as ferramentas da Microsoft, da VMWare e da Citrix. Naquele tempo a única ferramenta para servidor distribuída gratuitamente era o XenServer da Citrix, o que acabou tendo um grande peso em nossa escolha.

Primeiramente o XenServer foi instalado e configurado no servidor HP. Foi feito um teste de eficiência com a instalação de um *guest* com o sistema operacional Windows Xp Service Pack 2 configurado para ser acessado remotamente por inúmeros usuários. No teste notava-se que para cada usuário a área física da memória ocupada por uma instância da aplicação do NetBeans era a mesma, aproximadamente 400MB cada uma. Desta forma sobrecarregava a memória com partes da aplicação que poderiam ser compartilhadas tornando o resultado insatisfatório. É importante frisar que este problema ocorria não pela virtualização e sim pela maneira de gerência de memória e processamento multiusuário no próprio sistema operacional.

O mesmo teste foi feito com um *guest* com o sistema operacional Windows Server 2008 R2. O resultado foi que somente na primeira vez que o sistema executava a aplicação do

NetBeans consumia aproximadamente os 400MB, para os outros usuários a aplicação consumia aproximadamente 80MB adicional. Este teste foi satisfatório no aspecto de gerência de memória multiusuário, suprimindo o problema citado no parágrafo acima. Com base neste experimento adotamos o sistema operacional Windows Server da Microsoft como máquina *guest* do servidor.

Nota-se o fato de o servidor possuir pouco espaço físico de armazenamento em relação ao número de usuários X máquinas virtuais. Deste fato surgiu a necessidade da criação de um *Storage* para que os usuários tenham a possibilidade de armazenar suas informações sem sobrecarregar o servidor. O *Storage* foi configurado em outra máquina com a distribuição Fedora 10 compartilhando seu segundo HD de 250GB para o XenServer através da tecnologia de *ISCSI*.

Em geral as preferências dos professores em sistemas operacionais baseados no UNIX são as distribuições do Fedora e Debian. Baseado neste aspecto foram instalados mais duas máquinas *guests* uma com o Fedora 10 e a outro com o Debian 4. Feito isto, foi necessário configurá-las para realizarem o *login* no servidor da UNIOESTE, retirando o cadastro de usuários locais para cada *guest*. Esta escolha satisfaz o problema de gerenciamento de contas dos usuários, centralizando-o e tornando-o independente das máquinas virtuais.

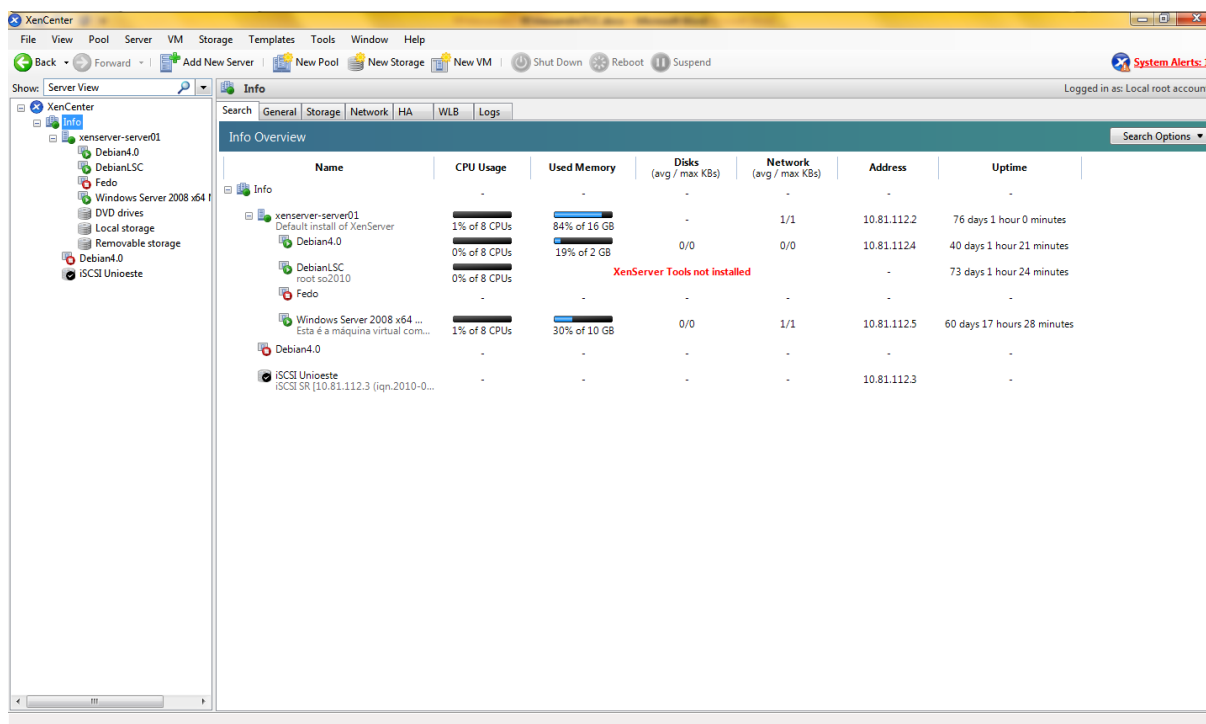


Figura 4.1 – Ambiente da UNIOESTE.

As pastas dos usuários tiveram um planejamento mais apurado, neste contexto foram levantadas algumas possíveis soluções. A solução que se apresentou satisfatória, perante as demais, foi a de criar e compartilhar no *guest* do Windows Server 2008 uma partição com todas as pastas de alunos e professores do curso, e cada pasta somente poderá ser acessada pelo proprietário e pelos administradores. Os outros *guests* foram configurados a montar a pasta do usuário que está efetuando seu *login* através da sua pasta compartilhada citada acima. Desta forma, o requisito para o usuário ter a flexibilidade de acessar sua pasta não importando em qual *guest* o mesmo se encontrar está satisfeito. Para automatizar este processo de criação e definição de permissões de acesso foram criados três *scripts* em Visual Basic Script, apresentados no Apêndice A.

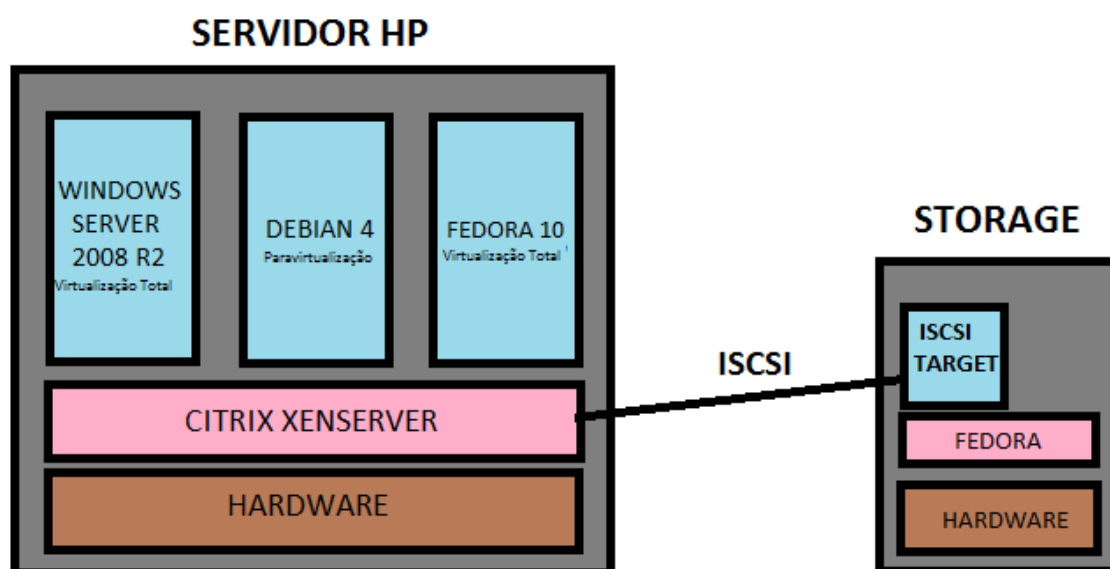


Figura 4.2 – Arquitetura implantada nos laboratórios de ensino do curso de Ciência da Computação na UNIOESTE.

4.4 Resultados

Para obter os resultados foi criado um ambiente sintético simulando uma atividade real nestes laboratórios e a utilização dos recursos do servidor estava sendo monitorados através do script disponível no Apêndice B. É importante salientar que os testes foram feitos para testar a virtualização com a utilização do serviço de acesso remoto oferecidos por um sistema operacional e não se avaliou o desempenho da virtualização versus servidor real.

Esta simulação teve o auxílio de uma ferramenta de gravação de macro para o *guest* do Windows Server 2008 R2, a ferramenta simula um usuário conectando remotamente nesta máquina virtual e inicializando a aplicação do NetBeans 6.8. Além disto, o usuário cria um projeto em *Java* e codifica-o. Este teste apresenta características de utilização muito próxima as reais, com os usuários virtuais compilando, criando classes e codificando seu projeto.

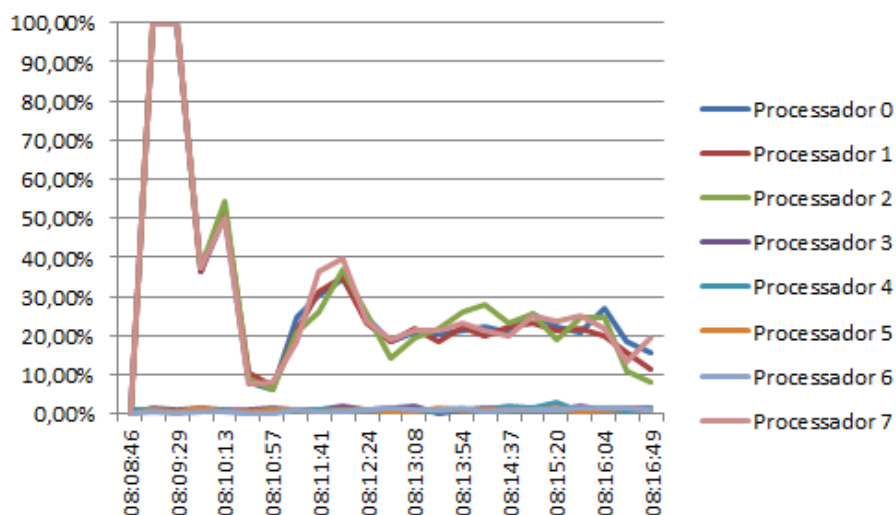


Figura 4.3 – Gráfico do uso dos processadores com cinco usuários.

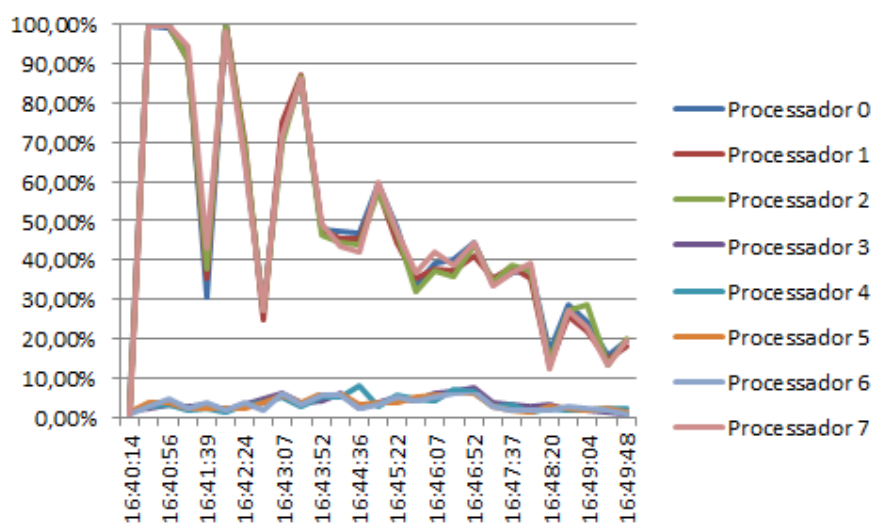


Figura 4.4 – Gráfico do uso dos processadores com dez usuários.

Para a realização dos testes duas baterias foram criadas, a primeira com a simulação de cinco computadores utilizando remotamente este *guest* e a segunda com dez computadores com o mesmo padrão de acesso.

Através da análise dos gráficos das figuras 4.3 e 4.4, relativos à utilização dos processadores, conclui-se que apesar do número de usuários dobrarem o número de processadores principais utilizados permaneceu o mesmo. Nota-se que com dez usuários o pico referente à abertura do NetBeans prolongou-se em relação aos de cinco, este detalhe foi ocasionado devido a quantidade de processamento necessário para o mesmo ser carregado em todos os dez usuários. Fica evidente o fato que o número de processadores disponíveis no servidor para uma aula de laboratório (dez a vinte alunos) é o suficiente para obter um bom desempenho.

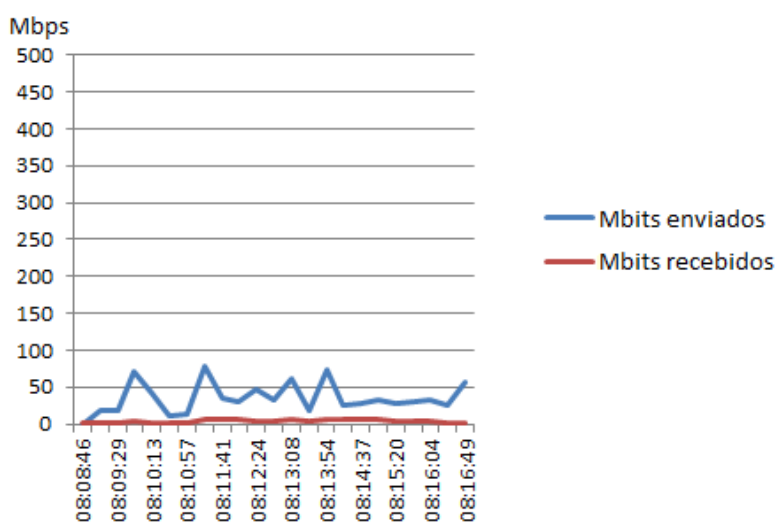


Figura 4.5 – Gráfico da rede com cinco usuários.

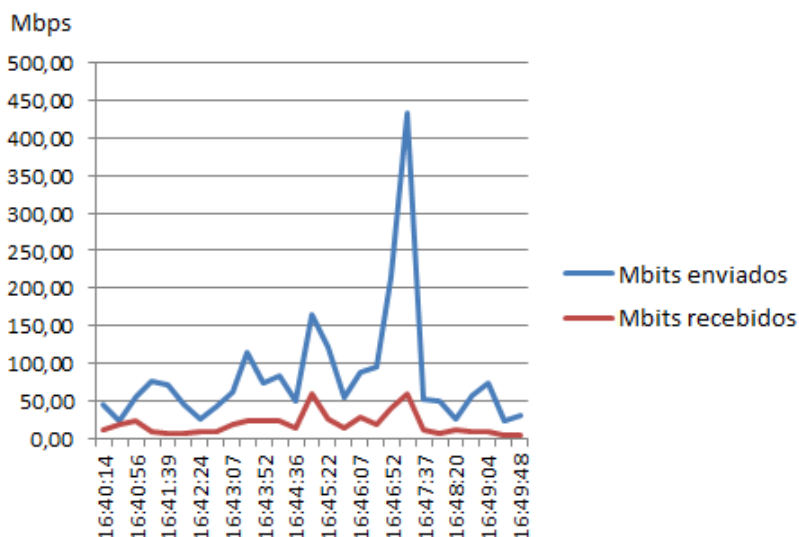


Figura 4.6 – Gráfico da rede com dez usuários.

Analisando os gráficos do tráfego de rede apresentados nas figuras 4.5 e 4.6, nota-se que houve um aumento significativo no gráfico com dez usuários em relação aos de cinco. Este

aumento ocorre tanto nos bits enviados como nos bits recebidos, apesar disto ainda podemos perceber que a placa ethernet de um 1Gbit do servidor não está sendo usada em sua total capacidade, por meio deste fato estima-se que esta placa ainda é suficiente para ser utilizada em uma aula de laboratório.

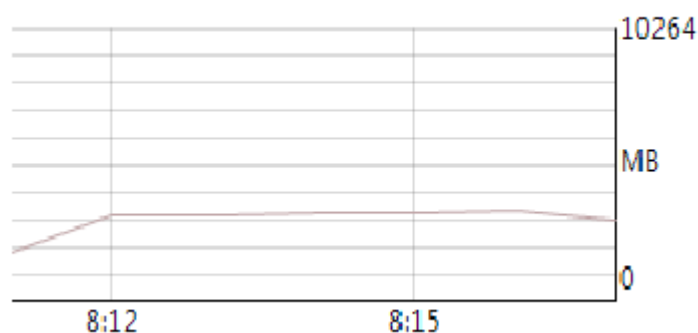


Figura 4.7 – Gráfico de uso de memória RAM com cinco usuários, obtidos do programa XenCenter.

É importante salientar que a máquina virtual testada possui 10GB de memória RAM e a eficiência no gerenciamento de memória está a cargo do sistema operacional Windows Server 2008 R2. Analisando as figuras 4.7 e 4.8, identifica-se um aumento no uso de memória para cada usuário. No entanto, este aumento por usuário é pequeno se comparada ao sistema operacional Windows XP, explicado no item 4.3 deste capítulo. Nota-se que com dez usuários a memória ultrapassa a metade de sua capacidade total, deixando evidente que em uma aula com 20 usuários a mesma pode chegar a seu limite.

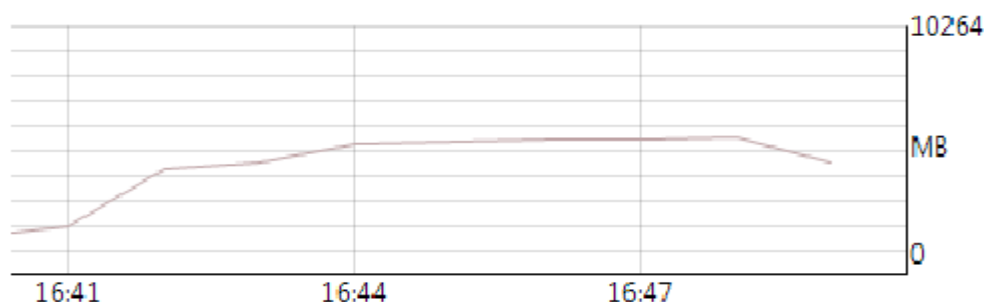


Figura 4.8 – Gráfico de uso de memória RAM com dez usuários, obtido do programa XenCenter.

Durante o ano letivo de 2010 algumas disciplinas usufruíram das máquinas virtuais nestes laboratórios, encontrando apenas um inconveniente à medida que o número de usuários crescia a rede transbordava e encerrava a conexão dos usuários do laboratório. Analisando este caso notou-se o fato do problema de saturação não se encontrava na placa ethernet do

servidor, mas na própria estrutura física da rede sem fio dos laboratórios, visto que muitos alunos utilizavam seus notebooks conectados a esta rede.

Além dos levantamentos de carga do servidor, alguns pontos subjetivos com a utilização desta tecnologia foram avaliados. Principalmente na parte da gerência, destaca-se a facilidade que a mesma proporciona para a criação, edição, restauração e remoção de máquinas virtuais. Nota-se por meio do caso da criação de uma máquina virtual com uma enorme lista de aplicativos a serem instalados, levaria dias para instalar tudo em um laboratório com 20 máquinas. Com a tecnologia do XenServer, uma máquina virtual seria criada com toda a lista de processos e seu estado salvo, possibilitando a clonagem da mesma. Desta forma, o tempo necessário para a equipe preparar as 20 máquinas diminui drasticamente.

Capítulo 5

5. Conclusão

Este trabalho apresenta uma proposta para a utilização de tecnologias de virtualização em laboratórios de ensino, principalmente nos laboratórios de cursos como o de ciência da computação. O intuito principal do trabalho é analisar o impacto, benéfico ou não, na utilização desta tecnologia.

Muitos trabalhos foram feitos com o foco de provar os benefícios da virtualização de servidores, dos quais, a maioria destes feitos em ambientes comerciais. Estes ambientes possuem características diferentes se comparado com o ambiente educacional proposto a ser avaliado, justificando e motivando este trabalho.

Foram apresentados os diversos tipos de virtualização da arquitetura x86 e as principais ferramentas comerciais disponíveis atualmente. Um estudo de caso nos laboratórios de ensino do curso de Ciência da Computação da Universidade Estadual do Oeste do Paraná (UNIOESTE) foi realizado para avaliar o impacto da virtualização neste ambiente.

O principal resultado obtido na simulação de uma aula prática nos laboratórios de ensino do curso de Ciência da Computação, apresentados no Capítulo 4, a exigência por processador sempre esteve abaixo da metade da capacidade do servidor. Nota-se também que a utilização da virtualização em ambientes desta categoria é uma tecnologia viável, entretanto deve-se ter cautela em relação à rede sem fio dos laboratórios, a qual por razões físicas e do projeto atual não fornece suporte ao tráfego de rede necessário, implicando na queda da mesma.

Destaca-se também as vantagens da virtualização não quantificadas neste presente trabalho, como economia de energia, economia de espaço, flexibilidade na escolha de sistemas, facilidade de gerência e de manutenção.

Apêndice A

Scripts em Visual Basic para criação, compartilhamento e edição permissões de pastas dos usuários.

Neste apêndice está disponibilizado três scripts em visual basic script para serem executados na máquina virtual do Windows server. O primeiro e o segundo *script* precisa necessariamente ser executado por um usuário pertencente o domínio, este usuário não precisa ter privilégios especiais no domínio. O terceiro *script* precisa ser executado como Administrador, caso contrário não concluirá com sucesso.

Os *scripts* utilizam o disco rígido nomeado como “E:” para criar as pasta dos usuários, cada usuário terá sua pasta com seu nome de *login* no domínio “WinNT://UNIOESTE-CAC/”. A ordem de execução deve ser a seguinte, primeiro os *scripts* de criação de pastas e por ultimo o *script* de compartilhamento e edição de permissões de acesso.

Script para criar as pastas dos alunos:

```
Dim strhomedir
Set objFindUser = GetObject("WinNT://UNIOESTE-CAC/AlunosInfo")
Set fso = CreateObject("Scripting.FileSystemObject")
Set listMember = objFindUser.Members
Set selecionar_objeto = CreateObject("Scripting.FileSystemObject")
For Each Member In listMember
    strhomedir = "e:\" & Member.Name
    If (fso.FolderExists(strhomedir)) Then
        'faz nada
    Else
        Set criar_pasta = selecionar_objeto.CreateFolder(strhomedir)
        CreateFolder = criar_pasta.Path
    End If
next
```

Script para criar as pastas dos professores:

```
Dim strhomedir
Set objFindUser = GetObject("WinNT://UNIOESTE-CAC/AlunosInfo - Prof")
Set fso = CreateObject("Scripting.FileSystemObject")
Set listMember = objFindUser.Members
Set selecionar_objeto = CreateObject("Scripting.FileSystemObject")
For Each Member In listMember
    strhomedir = "e:\" & Member.Name
    If (fso.FolderExists(strhomedir)) Then
        'faz nada
    Else
        Set criar_pasta = selecionar_objeto.CreateFolder(strhomedir)
        CreateFolder = criar_pasta.Path
    End If
next
```

Script para compartilhar e editar as permissões de cada pasta criada:

```
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objShell = CreateObject("WScript.Shell")
Set objFolder = objFSO.GetFolder("E:")
For Each Subfolder in objFolder.SubFolders
    exec= "%comspec% /c cacls " & Subfolder.Path & " /E /C /G UNIOESTE-CAC\" &
Subfolder.Name & ":f ""Domain Admins"":f system:f Administradores:f"
    objShell.Run(exec)
    exec2 = "%comspec% /c net share " & Subfolder.Name & "=" & Subfolder.Path & "
/GRANT:UNIOESTE-CAC\" & Subfolder.Name & ",FULL
/GRANT:Administradores,FULL"
    exec2 = exec2
    objShell.Run(exec2)
Next
```


Apêndice B

Script em shell para obter métricas do XenServer.

Segue abaixo o script utilizado para obter as métricas de uso de processador e uso da placa de rede, imprimindo os dados em tela e salvando os mesmos dados no arquivo de nome PerfLog.csv. Para executa-lo digite no console “sh PerfLog.sh”.

PerfLog.sh

```
echo "Tempo","Processor 0","Processor 1","Processor 2","Processor3","Processor 4","Processor 5","Processor 6","Processor7","Memory Bytes Used","Eth0 Bytes Sent","Eth0 Bytes Received" > PerfLog.csv
```

```
echo "Tempo","Processor 0","Processor 1","Processor 2","Processor3","Processor 4","Processor 5","Processor 6","Processor7","Memory Bytes Used","Eth0 Bytes Sent","Eth0 Bytes Received"
```

```
echo ""
count=0
#entra em loop infinito
while [ $count -lt 2 ]
do
#calcula data e hora
dateVar="$(date +%H:%M:%S)"
#calcula utilização dos processadores
procUtil0="$(xe host-cpu-param-get uuid=90854f10-5418-3c47-072c-e1207dd69fda param-name=utilisation)"
procUtil1="$(xe host-cpu-param-get uuid=50230805-91f0-8480-e2ff-87a44fc4d8ce param-name=utilisation)"
procUtil2="$(xe host-cpu-param-get uuid=4672ffaf-bcf5-2387-1ab0-46c7a167f2de param-name=utilisation)"
procUtil3="$(xe host-cpu-param-get uuid=3c1884b9-6aa9-c66e-23a5-3e02261ab4f3 param-name=utilisation)"
```

```

procUtil4="$(xe host-cpu-param-get uuid=6f3bfc5e-d075-0571-a9b6-28a0c08b83e2
param-name=utilisation)"
procUtil5="$(xe host-cpu-param-get uuid=da81654e-0d82-93ad-e080-d9a6c708e859
param-name=utilisation)"
procUtil6="$(xe host-cpu-param-get uuid=73b19ac7-f9d3-488b-19f1-822809fd42e6
param-name=utilisation)"
procUtil7="$(xe host-cpu-param-get uuid=ac435f54-e8fd-6b74-eaba-f5ff650822ba
param-name=utilisation)"

```

```
#calcula as métrica da memória
```

```
memFree="$(xe host-list params=memory-free --minimal)"
```

```
memTotal="$(xe host-list params=memory-total --minimal)"
```

```
memUsed=$((memTotal - memFree))
```

```
#calcula as métrica da interface
```

```
pifEth0Sent="$(xe host-data-source-query data-source=pif_eth0_tx)"
```

```
pifEth0Received="$(xe host-data-source-query data-source=pif_eth0_rx)"
```

```
#imprime em tela
```

```
echo
```

```
""$dateVar"",""$procUtil0"",""$procUtil1"",""$procUtil2"",""$procUtil3"",""$procUtil4"",""$procU
til5"",""$procUtil6"",""$procUtil7"",""$memUsed"",""$pifEth0Sent"",""$pifEth0Received""
```

```
#salva no arquivo
```

```
echo ""$dateVar"",""$procUtil0"",""$procUtil1"",""$procUtil2"",""$procUtil3"",
```

```
""$procUtil4"",""$procUtil5"",""$procUtil6"",""$procUtil7"",""$memUsed"",""$pifEth0Sent"",""
```

```
$pifEth0Received"" >> \um.csv
```

```
echo ""$dateVar"",""$memUsedWindows"" >> memUm.csv
```

```
sleep 20
```

```
#count=$((count + 1))
```

```
done
```

Glossário

Kernel	Núcleo completo de um sistema operacional
Micro-Kernel	Núcleo básico de um sistema operacional
Flags	Representa sim ou não para um estado específico de um sistema
Overhead	Excesso
Falha de página	No contexto de memória virtual, este termo significa que a referência de memória feita por um processo não está na memória.
Otimizada	Alteração que melhora o desempenho
Implementar	Programar
On-Demand	Algo realizado sob demanda
RAM	Memória de acesso randômico.
Storage	Local de armazenamento
Datas Centers	Local onde são concentrados os equipamentos de processamento e armazenamento de dados de uma empresa ou organização.

Referências Bibliográficas

ADAMNS, Keith; AGESEN, Olen. A comparasion of software and hardware techniques for x86 virtualization. In *Proceedings of the 12th international Conference on Architectural Support For Programming Languages and Operating Systems*. California: San Jose, 2006.

BARHAM, Paul; et al. Xen and the art of virtualization. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*. New York: 2003.

CREASY, R. J. The Origin of the VM/370 Time-sharing System. **IBM J. Res. Develop.**, Disponível em: <<http://www.research.ibm.com/journal/rd/255/ibmrd2505M.pdf>>. Acessado em março de 2010.

CHAGANTI, Prabhakar. *Xen Virtualization A practical handbook*. Olton Birmingham: Packt Publishing, 2007.

CHISNALL, David. *The definitive guide to the Xen Hypervisor*. Suite 900, Boston: Pearson Education Inc, 2008.

Citrix – XenServer. Disponível em: <<http://www.citrix.com/>>. Acesso em março de 2010.

Conceitos-Chave. Disponível em: <http://www.gta.ufrj.br/grad/09_1/versao-final/virtualizacao/conceitos%20chave.html>. Acesso julho de 2010.

JACOMO, C. Maximiliano. Virtualização – Uma realidade no T.I. In: *Palestra para os alunos da faculdade de tecnologia INED*. Belo Horizonte: 2009.

Lab Experiment: Hypervisor. Disponível em: <<http://virtualizationreview.com/articles/2009/03/02/lab-experiment-hypervisors.aspx>>. Acesso em março de 2010.

LAUREANO, Marcos Aurelio Pchek; *Máquinas virtuais e emuladores*. ISBN: 978-85-7669-191-4. Porto Alegre: Editora SBC, 2008.

Microsoft Corporation - Virtual PC. Disponível em: <<http://www.microsoft.com/windowsserversystem/virtualserver/>>. Acesso em março de 2010.

Microsoft Corporation – Hyper-V. Disponível em: <<http://www.microsoft.com/hyper-v-server/en/us/>>

POPEK, Gerald J.; GOLDBERG, Robert P. Formal Requirements for virtualizable third generation architectures. In: *Fourth ACM Symposium on Operation System Principles*. New York: Yorktown Heights, 1973.

ROBIN, S. John; IRVINE, E. Cynthia. Analysis of the intel Pentium ability to support a secure virtual machine monitor. In: *Proceedings of the 9th Conference on USENIX Security Symposium - Volume 9*. Colorado: Denver, 2000.

ROSE, Robert. *Survey of System Virtualization Techniques*. [S.L]: Disponível em <<http://www.robertwrose.com/vita/rose-virtualization.pdf>>. Acesso julho de 2010.

ROSENBLUM, M.; GARFINKEL, T. *Virtual Machine Monitors: Current Technology and Future Trends*. [S.L]: IEEE Computer, vol. 38, 2005.

SURGERMAN, Jeremy; VMWare, Inc. *Virtualizing I/O Devices on VMWare Workstation's Hosted Virtual Machine Monitor*. Inc. 3145 Porter Drive Building F Palo Alto.

Sun – VirtualBox. Disponível em: <<http://www.sun.com/>>. Acesso em março de 2010.

The Standard Performance Evaluation Corporation (SPEC). Disponível em: <<http://www.spec.org/specvirtualization/>>. Acesso em março de 2010.

TULLOCH, Mitch. *Understanding Microsoft Virtualization Solutions, From Desktop to Datacenter*. Washington: Microsoft Press, 2010.

VMWare. Disponível em: <<http://www.vmware.com>>. Acesso em março de 2010.

VMWare. *Understanding Full Virtualization, Paravirtualization, and Hardware Assist*. Inc. 3401 Hillview Ave.

VMWare. *Introduction VMWare Virtual Platform*. Inc. 3145 Porter Drive Building F Palo Alto.

WILLIAM, D.E. *Virtualization with Xen*. Burlington. Syngress Publishing, 2007.