

**FACULDADE DE ADMINISTRAÇÃO E NEGÓCIOS DE
SERGIPE – FANESE
NÚCLEO DE PÓS GRADUAÇÃO E EXTENSÃO - NPGE
CURSO DE PÓS-GRADUAÇÃO “LATO SENSU”
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES**

MARCELO RICARDO DE ARAGÃO BATISTA

VMWARE SERVER

**Aracaju – SE
2009**

MARCELO RICARDO DE ARAGÃO BATISTA

VMWARE SERVER

**Trabalho de Conclusão de Curso
apresentado ao Núcleo de Pós-
Graduação e Extensão da FANESE,
como requisito para obtenção do título
de Especialista em Redes de
Computadores**

**Aracaju – SE
2009**

MARCELO RICARDO DE ARAGÃO BATISTA

VMWARE SERVER

Trabalho de Conclusão de Curso apresentado ao Núcleo de Pós-Graduação e Extensão da FANESE, como requisito para obtenção do título de Especialista em Redes de Computadores

Sérgio Andrade Galvão

Mário Vasconcelos Andrade

Marcelo Ricardo de Aragão Batista

Marcelo Ricardo de Aragão Batista

Aprovado (a) com média: _____

Aracaju (SE), ____ de _____ de 2009.

RESUMO

A tecnologia de máquinas virtuais está rapidamente emergindo como um importante componente dos sistemas do futuro. Ela permite que um único computador execute múltiplos sistemas operacionais simultaneamente. As empresas podem utilizar um único servidor para múltiplas tarefas que normalmente teriam que executar em vários servidores, cada um rodando um sistema operacional diferente. Nos últimos anos, vários sistemas de virtualização foram desenvolvidos com o objetivo de subdividir os amplos recursos de um computador moderno. Alguns necessitam de *hardware* especializado, outros objetivam atingir 100% de compatibilidade em detrimento da performance. Alguns sistemas sacrificam a segurança ou funcionalidade para obter velocidade. Poucos oferecem isolamento de recursos ou garantia de performance. Esse artigo apresenta o *VMware Server*, uma máquina virtual x86 [1] que permite a múltiplos sistemas operacionais compartilharem hardware de um modo seguro e gerenciando recursos sem sacrificar performance ou funcionalidade. Isso é atingido fornecendo uma máquina virtual abstrata na qual sistemas operacionais como *Linux* e *Windows XP* podem ser portados com o mínimo de esforço.

Palavras-chave: Virtualização. *VMware Server*. Máquina virtual. Sistemas Operacionais. Recursos.

ABSTRACT

The virtual machine technology is quickly emerging as an important component of the systems of the future. It allows that a single computer executes multiples operating systems simultaneously. The companies can use a single server for multiple tasks that normally would have that to execute in various servers, each one executing a different operating system. In recent years, various virtualization systems had been developed with the objective to subdivide the ample resources of a modern computer. Some need specialized hardware, others objectify to reach 100% of compatibility in detriment of the performance. Some systems sacrifice the security or functionality to get speed. Few offer isolation of resources or guarantee of performance. This article presents the VMware Server, a virtual machine x86 [1] that allows multiple operating systems to share the hardware in a safe way and managing resources without sacrificing performance or functionality. This is reached supplying an abstract virtual machine in which operating systems as Linux and Windows XP can be carried with the minimum of effort.

Keywords: Virtualization. *VMware Server*. Virtual Machine. Operating Systems. Resources.

LISTA DE TABELAS

Tabela 1 Resultados do benchmark do Apache da máquina real.....	27
Tabela 2 Resultados do benchmark do Apache da máquina virtual.	28

LISTA DE GRÁFICOS

Figura 1 Instrução trap para o hypervisor	14
Figura 2 Exemplo de Microkernel	17
Figura 3 Virtualização e Paravirtualização.....	18
Figura 4 Arquitetura do VMware Server	20
Figura 5 Gráfico do benchmark do Apache da máquina real.....	27
Figura 6 Gráfico do benchmark do Apache da máquina virtual.	28

SUMÁRIO

RESUMO	_____
ABSTRACT	_____
LISTA DE TABELAS	_____
LISTA DE GRÁFICOS	_____
1 INTRODUÇÃO	6
2 VIRTUALIZAÇÃO	9
3 REQUISITOS PARA VIRTUALIZAÇÃO	12
4 HYPERVISOR TIPO 1	13
5 HYPERVISOR TIPO 2: VMWARE SERVER	15
6 PARAVIRTUALIZAÇÃO	17
7 VMWARE SERVER	19
7.1 Funcionamento do VMware Server	19
7.2 Vantagens do VMware Server	20
7.3 Especificações do VMware Server	21
7.4 Sistemas Operacionais Convidados Suportados pelo VMware Server	23
8 ANÁLISE DOS RESULTADOS	26
9 CONSIDERAÇÕES FINAIS	29
REFERÊNCIAS	30
CREDENCIAIS DO AUTOR	31

Existem várias maneiras de construir um sistema para hospedar múltiplas aplicações e servidores em uma máquina compartilhada. Provavelmente o jeito mais simples é implantar um ou mais *hosts* executando um sistema operacional padrão como *Linux* ou *Windows*, e então permitir aos usuários a possibilidade de instalar arquivos e iniciar processos – a proteção entre aplicações seria providenciada por técnicas convencionais de Sistemas Operacionais. [4]

VMware Server funciona como um computador dentro de um computador, o que permite iniciar um sistema operacional completo e executar quaisquer programas para aquele S.O., enquanto seu sistema operacional original fica intacto (e usável) [5]. Nenhuma mudança é necessária na máquina física a não ser a instalação do *VMware Server*. A máquina virtual não requer particionamento do disco rígido ou um disco rígido separado. Pelo contrário, a máquina virtual cria um disco rígido virtual para ser usado. Esse disco rígido virtual é um arquivo localizado no S.O. nativo [13]. Por exemplo, um usuário *Linux* que precise usar um programa *Windows* que não existe para *Linux* pode executar o *VMware Server* em uma janela, iniciar o *Windows* nela, executar o software e então desligar o *Windows*. O oposto também pode ser feito – um usuário *Windows* precisando usar aplicações *Linux* pode executá-las em uma máquina virtual. [5]

O objetivo desse trabalho é estudar as técnicas de virtualização mais utilizadas pelos *softwares* de virtualização existentes e comparar a performance de servidores virtuais com servidores nativos para verificar o custo benefício da virtualização de servidores. Através do uso de ferramentas de *benchmark*, podemos comparar a performance dos dois ambientes sem precisarmos estar em um ambiente de produção.

Embora a tecnologia de virtualização já exista há muitas décadas, somente agora as empresas vislumbram sua utilização na redução da complexidade do ambiente de informática e na redução dos gastos com energia elétrica. A justificativa desse trabalho é justamente mostrar os benefícios dessa tecnologia em um estudo de caso da utilização da virtualização no servidor web mais utilizado do mundo pelas empresas, o *Apache*.

Como metodologia, será efetuada uma pesquisa bibliográfica em livros e artigos sobre o tema em questão, elaborando a fundamentação teórica necessária, abordando os conceitos de Virtualização de Sistemas Operacionais.

Com os conceitos em mente, o próximo passo será um estudo detalhado das características do *VMware Server*, finalizando com um estudo de caso comparando em laboratório a performance do servidor web *Apache* em um ambiente virtualizado com o mesmo servidor em um ambiente nativo sem virtualização, utilizando a ferramenta de *benchmark Autobench* [12].

2 VIRTUALIZAÇÃO

Atualmente, a maioria das empresas possui vários servidores, um para cada serviço: um servidor de *e-mail*, um servidor *WEB*, um servidor de *FTP* entre outros. Todos esses servidores estão interligados por uma rede de alta velocidade, ou seja, um multicomputador [6]. Geralmente esses servidores são colocados em máquinas diferentes por questões de segurança, confiabilidade e desempenho – se uma máquina falhar, os outros serviços continuam ativos. Porém, toda essa tolerância a falhas possui um preço: será necessário um grande investimento em *hardware*. [6]

Para solucionar esse problema, utiliza-se uma técnica já conhecida há aproximadamente 40 anos: **virtualização**. Essa tecnologia permite a um único computador hospedar múltiplas máquinas virtuais, cada uma executando um sistema operacional diferente [6]. Uma máquina virtual (Virtual Machine - VM) é uma camada de abstração ou ambiente entre os componentes de *hardware* e o usuário final. Máquinas virtuais executam sistemas operacionais e são algumas vezes chamadas de servidores virtuais. Um sistema operacional nativo pode executar várias máquinas virtuais e compartilhar os componentes de *hardware* como a CPU, controladores, disco, memória, e E/S entre os servidores virtuais. [7]

A vantagem dessa abordagem é que uma falha em uma máquina virtual não afeta as outras. Num sistema virtualizado, diferentes servidores podem rodar em diferentes máquinas virtuais, mantendo a tolerância a falhas da abordagem do multicomputador, mas com um custo muito menor e uma melhor capacidade de gerenciamento e manutenibilidade. [6]

A virtualização possui um problema óbvio: como todas as máquinas virtuais estão em um único servidor real, se esse servidor parar todas as máquinas serão desativadas. Mas como a maioria dos problemas com servidores não estão

relacionados com falhas de *hardware* e sim com sobrecarga e *bugs* de sistemas, a virtualização continua sendo uma abordagem atrativa em termos de custo. [6]

Executar software em máquinas virtuais tem outras vantagens além do forte isolamento. Dentre estas vantagens, ter menos máquinas reais diminui custos em *hardware* e energia elétrica e toma menos espaço físico. [6]

Outra vantagem é que migrar máquinas virtuais é muito mais fácil do que migrar processos que estão executando em um S.O. tradicional. Para migrar os processos, uma quantidade grande de informação de estado crítica sobre cada processo é mantida em tabelas do S.O., incluindo informações sobre arquivos abertos, alarmes, sinais de controle etc [6]. Já para migrar uma máquina virtual, tudo que precisa ser movido é a imagem da memória, uma vez que todas as tabelas são movidas em conjunto. [6]

As máquinas virtuais também podem executar aplicações legadas em S.O. que não possuem mais suporte ou que não funcionam em *hardware* atual. [6]

Desenvolvimento de software é outro uso importante de máquinas virtuais. Um programador que quer ter certeza que seu programa funcionará no *Windows 98*, *Windows 2000*, *Windows XP*, *Windows Vista*, *Linux* etc. não precisa de vários computadores com vários sistemas operacionais diferentes. Mesmo particionando o disco rígido, ele teria que reiniciar o computador para poder usar cada sistema operacional de uma vez, enquanto que as máquinas virtuais podem estar rodando ao mesmo tempo. [6]

Existem três categorias básicas de virtualização [14]:

- **Virtualização de Armazenamento**, a qual cria um armazenamento físico da união de múltiplos dispositivos de armazenamento de rede para que eles pareçam como um único dispositivo de armazenamento;
- **Virtualização de Rede**, a qual combina recursos de computação em uma rede dividindo a banda disponível em canais independentes que podem ser designados para um servidor particular ou dispositivo em tempo real;
- **Virtualização de Servidores**, a qual esconde a natureza física dos recursos dos servidores, incluindo o número e identidade de servidores individuais, processadores e sistemas operacionais e do software executando neles.

A última categoria é a aplicação mais utilizada da tecnologia de virtualização atualmente. Quando as pessoas usam o termo “virtualização”, elas geralmente se referem à virtualização de servidores. [14]

3 REQUISITOS PARA VIRTUALIZAÇÃO

Existem duas formas de se fazer virtualização: utilizando um *hypervisor* tipo 1 (ou monitor de máquina virtual) ou um *hypervisor* tipo 2 [6]. O *hypervisor* tipo 1 funciona como um sistema operacional que possui vários outros sistemas operacionais rodando em cima dele. Já o tipo 2 é uma aplicação que roda em um S.O. e que tem a habilidade de gerenciar várias máquinas virtuais. Os sistemas operacionais que rodam em cima do *hypervisor* são chamados de sistemas operacionais convidados. No caso do tipo 2, o sistema que roda a aplicação é chamado de sistema operacional hospedeiro. [6]

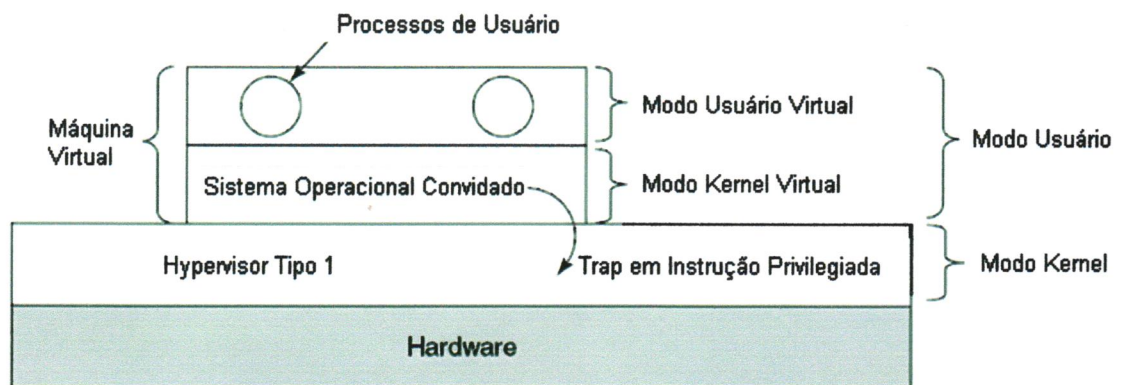
Em ambos os casos, as máquinas virtuais devem agir como *hardware* real, ou seja, deve poder iniciar um sistema operacional, instalar um sistema operacional etc. [6]

A razão da existência desses dois tipos de *hypervisores* foi um defeito na arquitetura *Intel 386* que continuou ao longo de vinte anos para manter a compatibilidade com os programas antigos.

4 HYPERVISOR TIPO 1

Os *hypervisores* tipo 1 rodam diretamente sobre o hardware. A máquina virtual executa como um processo de usuário em modo usuário e por isso não pode executar instruções sensíveis [6]. Instruções sensíveis (ou privilegiadas) são usadas exclusivamente pelo sistema operacional (para manipular interrupções, leitura e gravação em dispositivos, e memória virtual) e formam a classe dominante de instruções requeridas para a virtualização. Por definição, um programa de usuário não pode executar essas instruções. Uma técnica que força a emulação dessas instruções é executar todo o código de uma máquina virtual, incluindo o S.O. sendo virtualizado, como código de usuário. O *hypervisor* então manipula a exceção produzida pela tentativa de executar uma instrução privilegiada e executa a ação desejada [8]. A máquina virtual executa um S.O. convidado que pensa que está no modo *kernel*, embora na verdade esteja no modo usuário. Esse modo é chamado de **modo *kernel* virtual**. [6]

Quando o sistema operacional convidado executa uma instrução sensível, uma instrução *trap* [6] ocorre (figura 1). O *hypervisor* pode então inspecionar a instrução para verificar se ela foi disparada pelo sistema operacional convidado na máquina virtual ou por um programa de usuário na máquina virtual. No primeiro caso, ele executa a instrução diretamente no modo *kernel*; no outro caso, ele emula o que o verdadeiro *hardware* faria quando se encontra uma instrução sensível executada no modo usuário. [6]



Fonte: TANENBAUM [6]

Figura 1 Instrução trap para o hypervisor

5 HYPERVISOR TIPO 2: VMWARE SERVER

O primeiro *hypervisor* tipo 2 foi o *VMware Server*. Ele executa como um programa de usuário comum em um S.O. hospedeiro como *Windows* ou *Linux*. Quando ele inicia pela primeira vez, ele age como um computador que acabou de ligar e espera encontrar um *CD-ROM* contendo um S.O. no drive de *CD-ROM*. Ele então instala o S.O. em um *disco virtual* (um arquivo no *Windows* ou *Linux*) executando o programa encontrado no *CD-ROM*. Uma vez instalado o S.O. convidado no disco virtual, ele pode ser iniciado. [6]

Quando ele executa um programa binário de um *Pentium*, obtido do *CD-ROM* de instalação ou do disco virtual, o *VMware Server* pesquisa no código procurando por **blocos básicos**, isto é, execução de instruções que terminam em um *jump*, *call*, *trap* ou outras instruções que mudam o fluxo de controle. Por definição, nenhum bloco básico contém instruções que modificam o contador de programa exceto o último. O bloco básico é inspecionado para verificar se ele contém alguma instrução sensível. Se existir, cada uma é trocada por uma chamada a um procedimento do *VMware Server* que cuida delas. A instrução final também é trocada com uma chamada para o *VMware Server*. [6]

Uma vez que esses passos são executados, o bloco básico é copiado para o *VMware Server* e executado. Um bloco básico que não contém qualquer instrução sensível será executado tão rápido no *VMware Server* como se estivesse na máquina real – porque elas executam na máquina real. Instruções sensíveis são capturadas e emuladas. Essa técnica é conhecida como **tradução binária**. [6]

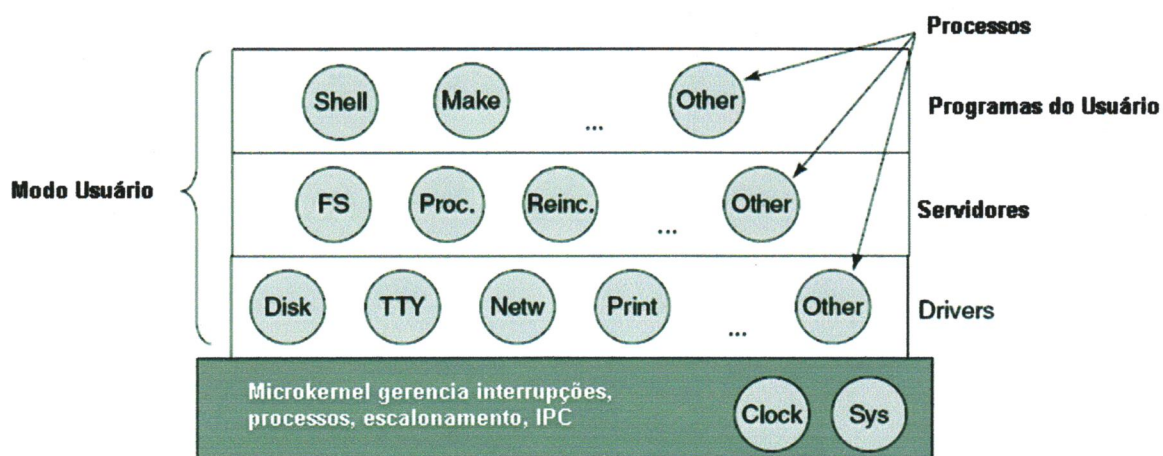
Depois que o bloco básico completou sua execução, o controle retorna ao *VMware Server*, que localiza o próximo bloco. Se o próximo bloco já foi traduzido, ele pode ser executado imediatamente. Senão, ele é então traduzido, copiado e executado. Eventualmente, a maior parte do programa será copiada e executada

próximo da velocidade total. Várias otimizações são utilizadas, por exemplo, se um bloco básico termina pulando para outro bloco, a instrução final pode ser trocada por uma instrução *jump* ou *call* diretamente para o bloco básico traduzido, eliminando todo o processamento adicional associado com a busca do próximo bloco. Também, não há necessidade de trocar instruções sensíveis em programas do usuário – o *hardware* irá ignorá-las. [6]

6 PARAVIRTUALIZAÇÃO

Os *hypervisores* do tipo 1 e 2 trabalham com S.O.s convidados sem modificações, mas precisam de alguns artifícios vistos anteriormente para terem uma performance aceitável. Uma abordagem diferente é modificar o código fonte do S.O. convidado para que, ao invés de executar instruções sensíveis, ele faça **chamadas ao hypervisor**. Isso faria com que o S.O. convidado funcionasse como um programa de usuário fazendo chamadas ao sistema (*system calls* [6]) para o S.O. (*hypervisor*). Quando esse caminho é usado, o *hypervisor* deve definir uma interface consistindo de um conjunto de chamadas de procedimentos (*procedure calls*) que o S.O. convidado pode usar. Esse conjunto de chamadas forma uma **API** [9] (**Application Programming Interface**). [6]

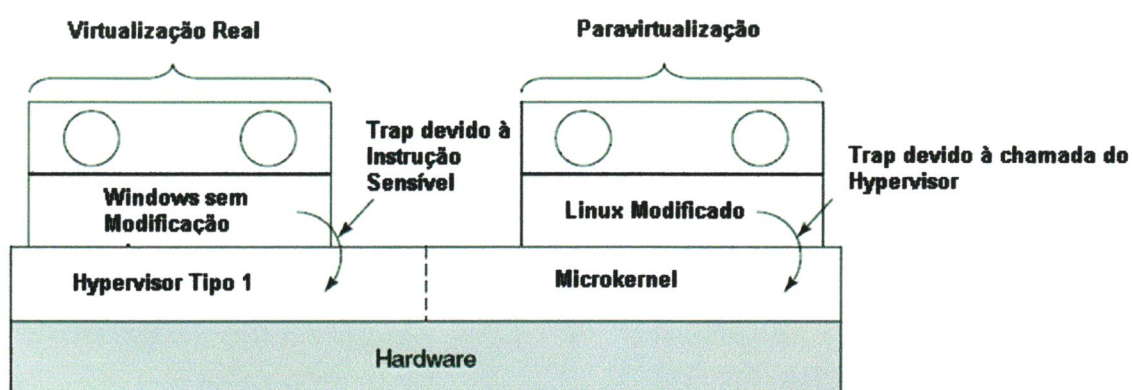
Removendo todas as instruções sensíveis do S.O. e fazendo com que ele faça chamadas ao *hypervisor* para conseguir serviços de sistema como E/S, o *hypervisor* se transforma em um *microkernel* (figura 2). Um S.O. convidado que tem suas instruções sensíveis removidas é chamado de **paravirtualizado**.



Fonte: TANENBAUM [6]

Figura 2 Exemplo de Microkernel

A diferença entre a verdadeira virtualização e a paravirtualização é ilustrada na figura 3. Nela vemos duas máquinas virtuais. Na esquerda está uma versão do *Windows* sem modificações como S.O. convidado. Quando uma instrução sensível é executada, o *hardware* causa uma instrução *trap* para o *hypervisor*, o qual emula a instrução e retorna. Na direita está uma versão modificada do *Linux* que não possui instruções sensíveis. Quando ele precisa fazer alguma operação, como E/S, ele faz uma chamada ao *hypervisor*, como uma aplicação normal fazendo uma chamada ao sistema no *Linux* padrão. [6]



Fonte: TANENBAUM [6]

Figura 3 Virtualização e Paravirtualização

Paravirtualização provou ser uma abordagem de virtualização eficiente com baixo processamento da camada de virtualização. Porém, a desvantagem é que ela necessita de profundas modificações no *kernel* do S.O., limitando seu uso no mercado. Modificação do *kernel* significa que o S.O. deve ser de código aberto ou liberado pela empresa desenvolvedora do S.O. [15]

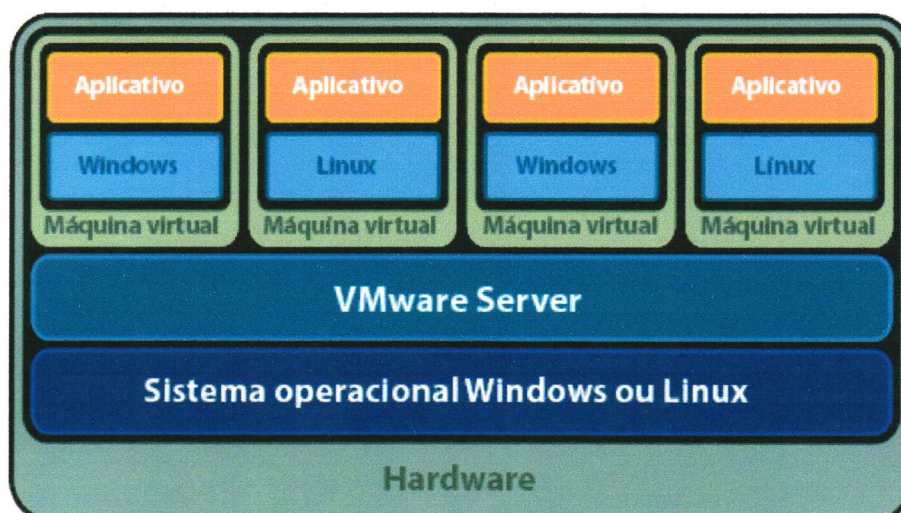
7 VMWARE SERVER

O *VMware Server* é um programa gratuito de virtualização para servidores *Windows* e *Linux* com suporte de nível corporativo. Ele permite que as empresas particionem um servidor físico em várias máquinas virtuais para que elas possam aproveitar todos os benefícios da virtualização. O *VMware Server* é poderoso e, ao mesmo tempo, de fácil utilização por usuários sem experiência com a tecnologia de virtualização de servidores. [10]

7.1 Funcionamento do VMware Server

O *VMware Server* é instalado e executado como um aplicativo em um sistema operacional de *host Windows* ou *Linux*. Uma fina camada de virtualização particiona o servidor físico para que várias máquinas virtuais possam ser executadas simultaneamente em um único servidor (figura 4). [10]

Os recursos de computação do servidor físico são tratados como um pool uniforme de recursos que podem ser alocados às máquinas virtuais de forma controlada. [10]



Fonte: VMWARE [10]

Figura 4 Arquitetura do VMware Server

O *VMware Server* isola cada máquina virtual de seu host e das outras máquinas virtuais, impedindo que sejam afetadas se alguma máquina virtual falhar. Os dados não vazam entre as máquinas virtuais e os aplicativos só podem se comunicar por conexões de rede configuradas. O *VMware Server* encapsula o ambiente da máquina virtual como um conjunto de arquivos fáceis de fazer backup, mover e copiar.

7.2 Vantagens do VMware Server

Ao criar e executar máquinas virtuais com o *VMware Server*, os usuários podem:

- Colocar em produção servidores adicionais em minutos, sem investir em novo *hardware*;
- Executar S.O. e aplicativos *Windows*, *Linux*, *Solaris* e *Netware* no mesmo servidor físico;
- Aumentar a utilização da *CPU* de um servidor físico;
- Mover máquinas virtuais de um servidor físico para outro, sem necessidade de reconfiguração;
- Capturar o estado completo de uma máquina virtual e recuperar aquela configuração com o clique de um botão.

- Simplificar o desenvolvimento e os testes de software, permitindo que os desenvolvedores criem vários ambientes com diferentes S.O.s no mesmo servidor;
- Simplificar os testes de *patches*, novos aplicativos e sistemas operacionais pelo departamento de TI, permitindo que os administradores de sistemas façam testes em um ambiente seguro de máquina virtual e que possam retornar a um estado “limpo” com o recurso de *snapshot*;
- Simplificar a colocação de servidores em produção, desenvolvendo uma máquina virtual uma vez e implantando-a várias vezes;
- Avaliar *software* em máquinas virtuais prontas para uso, sem necessidade de instalação e configuração;
- Realocar S.O.s legados como *Windows NT Server 4.0* e *Windows 2000 Server* em uma máquina virtual executada em *hardware* e S.O.s novos;
- Aproveitar dispositivos virtuais pré-desenvolvidos, prontos para execução, incluindo *hardware*, S.O.s e ambientes de aplicativos virtuais.

7.3 Especificações do VMware Server

O *VMware Server* proporciona um *hardware* virtual acima do processador e memória reais do computador hospedeiro. Abaixo seguem as especificações do *hardware* da máquina virtual: [11]

- Processador:
 - Mesmo processador do computador hospedeiro;
 - Um processador virtual no sistema hospedeiro com um ou mais processadores lógicos;
- Chip Set:
 - Intel 440BX;
 - NS338 SIO;
 - 82093AA IOAPIC;

- BIOS:
 - PhoenixBIOS 4.0 Release 6 com VESA BIOS;
- Memória:
 - Até 8 GB, dependendo da memória do computador hospedeiro, da versão do *hardware* da máquina virtual e do suporte do S.O. convidado;
 - Memória total disponível para todas as máquina virtuais é limitada somente pela quantidade de memória do computador hospedeiro;
- Gráficos:
 - VGA;
 - SVGA;
- Drives IDE:
 - Até 4 dispositivos. Qualquer um deles pode ser um disco rígido virtual ou um drive de CD/DVD;
 - Discos virtuais IDE até 950 GB;
 - Drive de CD/DVD pode ser um dispositivo físico no sistema hospedeiro ou cliente, ou um arquivo de imagem ISSO;
- Dispositivos SCSI:
 - Até 60 dispositivos. Da mesma forma que os drives IDE, podem ser um disco rígido virtual ou um drive de CD/DVD;
 - Discos virtuais SCSI até 950 GB;
 - Controlador LSI Logic LSI53C10xx Ultra320 SCSI I/O;
 - Adaptador de barramento Mylex (BusLogic) BT-958;
- Slots PCI:
 - Seis slots PCI virtuais podem ser divididos entre os controladores SCSI, placas virtuais Ethernet, adaptadores virtuais de vídeo e adaptadores virtuais de som;
- Drives de Disquete:
 - Até dois drives de disquete de 1,44 MB;
 - Drives físicos arquivos de imagens de disquetes;
- Portas Seriais:
 - Até quatro portas seriais;

- Portas Paralelas:
 - Até três portas bidirecionais paralelas;
- Portas USB:
 - Suporte a USB 2.0;
- Teclados:
 - Teclados de 104 teclas;
- Mouse:
 - Mouse PS/2;
- Placa Ethernet:
 - Até dez placas virtuais Ethernet;
 - Compatível com AMD PCnet-PCI II;
- Redes Virtuais:
 - Suporta até dez switches de redes virtuais em sistemas operacionais hospedeiros *Windows* ou 255 se for *Linux*;
 - Suporta os protocolos TCP/IP, Samba, Netware e NFS;
- Som:
 - Entrada e saída de som somente no sistema hospedeiro;
 - Emula Creative Labs Sound Blaster AudioPCI. Entrada MIDI, controladores de jogos não são suportados, exceto se forem USB.

7.4 Sistemas Operacionais Convidados Suportados pelo VMware Server

O *VMware Server* suporta um grande número de S.O.s convidados.

Abaixo estão relacionados os mais importantes: [11]

- Sistemas operacionais *Windows* de 64 bits:
 - Windows Server 2008 x64 Standard Edition;
 - Windows Server 2008 x64 Enterprise Edition;
 - Windows Vista x64 Business Edition;
 - Windows Vista x64 Ultimate Edition;
 - Windows XP Professional x64;
 - Windows Server 2003 x64 Standard Edition;
 - Windows Server 2003 x64 Web Edition;

- Windows Server 2003 x64 Enterprise Edition;
- **Sistemas operacionais *Windows* de 32 bits:**
 - Windows Server 2008 Standard Edition;
 - Windows Server 2008 Enterprise Edition;
 - Windows Vista Business Edition;
 - Windows Vista Ultimate Edition;
 - Windows XP Professional;
 - Windows Server 2003 Standard Edition;
 - Windows Server 2003 Web Edition;
 - Windows Server 2003 Enterprise Edition;
 - Windows Small Business Server 2003 Standard Edition;
 - Windows Small Business Server 2003 Premium Edition;
 - Windows 2000 Server;
 - Windows 2000 Advanced Server;
- **Sistemas operacionais *Linux* de 64 bits:**
 - Mandrake Linux;
 - Mandriva Linux;
 - Red Hat Enterprise Linux;
 - SUSE Linux;
 - SUSE Linux Enterprise Server;
 - openSUSE Linux;
 - Open Enterprise Server (OES);
 - Ubuntu Linux;
- **Sistemas operacionais *Linux* de 32 bits:**
 - Mandrake Linux;
 - Mandriva Linux;
 - Red Hat Enterprise Linux;
 - SUSE Linux;
 - SUSE Linux Enterprise Server;
 - openSUSE Linux;
 - Open Enterprise Server (OES);
 - Ubuntu Linux;
- **Sistemas operacionais Sun Solaris de 64 bits:**
 - Solaris x86;

- **Sistemas operacionais Sun Solaris de 32 bits:**
 - Solaris x86;
- **Sistemas operacionais Novell NetWare de 32 bits:**
 - NetWare;

8 ANÁLISE DOS RESULTADOS

Para testar a eficiência da máquina virtual *VMware* utilizamos o servidor *web Apache*, versão 2.2, o mais utilizado no mundo, em uma máquina real *Windows XP SP3* e uma máquina virtual *Windows XP SP3*, com a mesma configuração de *hardware* – processador *Intel Core 2 Duo E6750 2,66GHz* e memória RAM de 512MB. Os resultados demonstram que com o *VMware*, a performance do servidor virtual ficou praticamente idêntica ao do servidor nativo.

Para capturar os dados do teste de performance foi utilizado o software de *benchmark Autobench*. Ele permite retirar estatísticas de servidores *web* como o *Apache*, bem como comparar o desempenho de dois servidores *web* diferentes.

A tabela 1 mostra os dados capturados do *Apache* versão 2.2 da máquina real e a figura 5 o gráfico derivado desses dados, enquanto que na tabela 2 e na figura 6 visualizamos os dados do *Apache* versão 2.2 da máquina virtual. Podemos observar nas tabelas e nas figuras que o **tempo de resposta**, o índice mais importante para a avaliação da performance do *Apache* versão 2.2, em uma máquina virtual é quase idêntico ao do ambiente nativo.

Requisições por segundo	Conexões por segundo	Mínimo de respostas por segundo	Média de respostas por segundo	Máximo de respostas por segundo	Tempo de resposta (segundos)	KB por segundo
200.0	20.0	199.9	200.0	200.0	0.7	69.7
400.0	40.0	398.4	400.0	401.6	1.0	139.5
600.0	60.0	598.4	600.0	601.6	1.3	209.2
800.0	80.0	799.2	800.0	801.3	1.5	278.9
1000.1	100.0	999.5	999.9	1000.5	2.1	348.7
1195.1	119.5	1161.9	1200.0	1238.7	3.6	416.7
1399.9	140.0	1302.0	1400.0	1451.4	5.2	488.0
1587.2	158.7	1520.1	1599.9	1680.3	7.7	553.4
1799.8	180.0	1798.9	1799.8	1800.5	5.7	627.5
1967.0	196.7	1841.9	1968.2	2007.5	4.5	685.8

Tabela 1 Resultados do benchmark do Apache da máquina real.

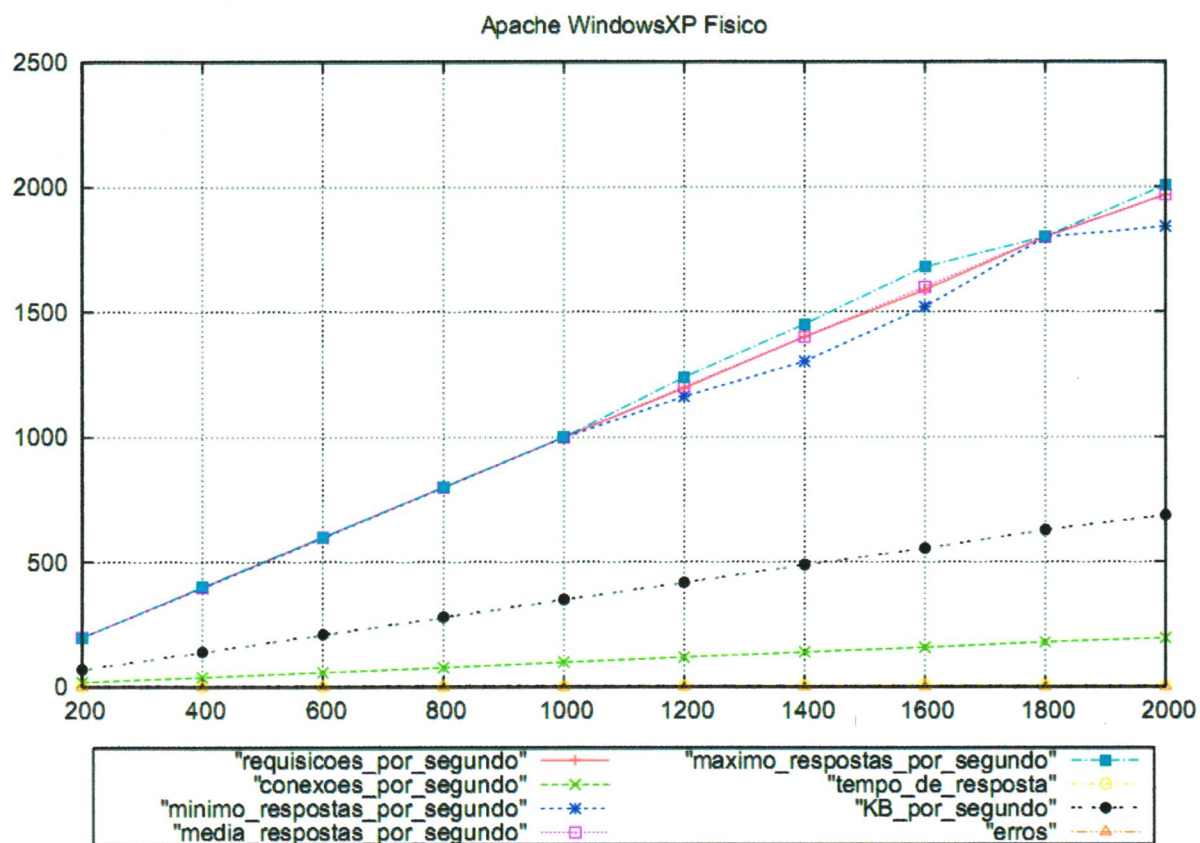


Figura 5 Gráfico do benchmark do Apache da máquina real.

Requisições por segundo	Conexões por segundo	Mínimo de respostas por segundo	Média de respostas por segundo	Máximo de respostas por segundo	Tempo de resposta (segundos)	KB por segundo
200.0	20.0	199.0	200.0	202.0	0.9	69.7
400.0	40.0	388.2	400.0	411.8	1.2	139.5
600.0	60.0	597.7	600.0	601.9	1.5	209.2
799.8	80.0	798.1	800.0	802.4	1.5	278.8
655.5	90.0	0.0	714.1	1003.4	12.8	227.7
1114.8	120.0	0.0	1110.4	1494.7	25.7	388.6
1399.3	139.9	1397.0	1399.9	1402.8	2.6	487.8
1597.0	159.7	1559.1	1594.8	1621.1	7.2	556.8
1465.3	169.4	1347.8	1471.7	1553.0	127.0	510.8
1024.7	173.6	758.8	1083.7	1433.5	204.0	357.3

Tabela 2 Resultados do benchmark do Apache da máquina virtual.

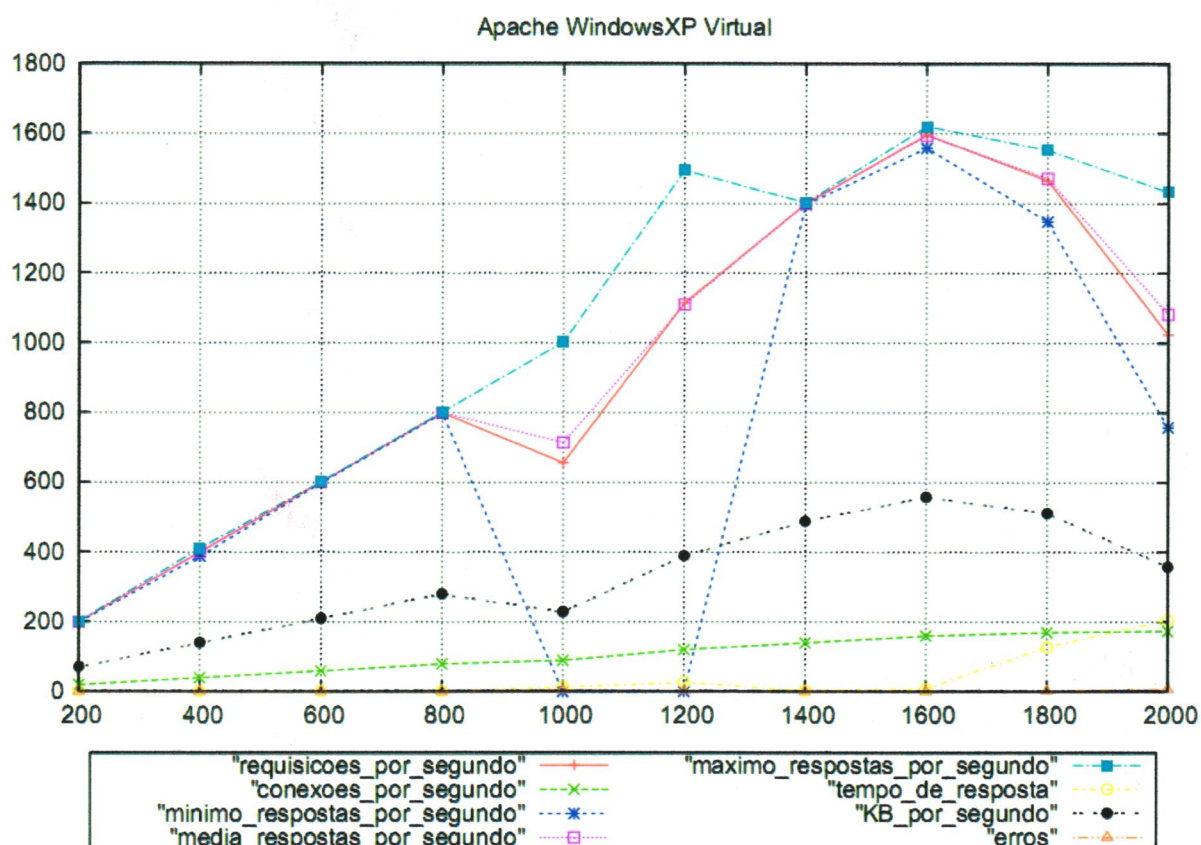


Figura 6 Gráfico do benchmark do Apache da máquina virtual.

9 CONSIDERAÇÕES FINAIS

A capacidade de rodar mais *software* num mesmo *hardware*, combinando aplicações virtualizadas que apresentam demandas de processamento diferentes, dá às empresas a chance de aproveitar melhor os recursos dos servidores. Não faltam aplicações com pico de utilização num determinado horário ou dia do mês, que deixam o *hardware* ocioso por um bom tempo. A virtualização é uma das técnicas de consolidação de servidores que permite uma melhor utilização dos recursos dos servidores das empresas.

Como toda a tecnologia, a virtualização tem prós e contras. Não estão descartadas incompatibilidades e perda de performance com certas aplicações. Como vimos no exemplo do servidor *web Apache*, a performance é um pouco inferior no ambiente virtualizado devido ao processamento adicional ocasionado pela camada de virtualização.

Entre as vantagens, temos a redução do consumo de energia com a redução do número de servidores devido à virtualização de serviços, facilidade para aumentar a capacidade de processamento à medida que demanda cresce, testes mais seguros de novas aplicações em novos ambientes entre outras.

Neste artigo apresentamos o *VMware Server*, um produto da empresa *VMware*, líder de mercado na área de virtualização. Como vimos na análise de resultados, a performance de um servidor *web* virtual e um servidor *web* nativo é praticamente equivalente, o que torna a virtualização com sua utilização uma proposta vantajosa para as empresas.

REFERÊNCIAS

- [1] TANENBAUM, Andrew S. **Organização Estruturada de Computadores**. São Paulo: Prentice Hall, 2007.
- [2] OLIVEIRA, Gisele de. **Estudo comparativo de metodologias de Consolidação de servidores para a elaboração de uma proposta de implementação**. Joinville, 2004.
- [3] **Web Services Activity Statement**.
Disponível em: <<http://www.w3.org/2002/ws/Activity>>. Acesso em: 03 abr. 2009.
- [4] BARHAM, Paul et al. **Xen and the Art of Virtualization**. Cambridge, 2003.
- [5] WARD, Brian. **The Book of VMware – The Complete Guide to VMware Workstation**. Toronto, 2002.
- [6] TANENBAUM, Andrew S. **Modern Operating Systems**. New Jersey: Prentice Hall, 2008.
- [7] DANIELS, J. **Server Virtualization Architecture and Implementation**. Crossroads, New York, v. 16, n. 1, p. 8-12, 2009.
- [8] CROSBY, S. et al. **The Virtualization Reality**. Acm Queue, New York, p. 34-41, 2007.
- [9] **API**
Disponível em:
<http://www.pcmag.com/encyclopedia_term/0,2542,t=application+programming+interface&i=37856,00.asp>. Acesso em: 16 jul. 2009.
- [10] BRASIL, VMware. **VMware Server**. São Paulo, 2005.
- [11] VMWARE. **VMware Server User's Guide**. Palo Alto, 2008.
- [12] MIDGLEY Julian T. **Autobench**. Disponível em:
<<http://www.xenoclast.org/autobench/>>. Acesso em: 28 out. 2009.
- [13] NECAISE, Rance D. et al. **Using VMware for Dual Operating Systems**. Virginia, 2001.
- [14] RAY, E. et al. **Virtualization Security**. Orange, 2009.
- [15] Chen, W. et al. **A Novel Hardware Assisted Full Virtualization Technique**. Hunan, 2008.

CREDENCIAIS DO AUTOR

Bacharel em Ciência da Computação.

Analista Judiciário - Analista de Sistemas do Tribunal de Justiça de Sergipe.

E-mail: mrdab23@gmail.com, marcelo.ricardo@tjse.jus.br.

Telefone: 3217-8293 / 8801-4753.