# Vantagens de se Executar Serviços de Rede Multicamadas em Ambientes Virtualizados com Xen

Wallace Rodrigues de Santana[1]

**Resumo:**

Este artigo discute as vantagens de se hospedar um sistema multicamadas em um ambiente virtualizado, ao invés de um ambiente formado por máquinas *stand-alone*. São apresentadas as técnicas usadas pelo Xen para que ele possa extrair o máximo de desempenho. Este artigo oferece também uma visão geral de como a tecnologia se desenvolveu, e o impacto na economia de recursos e na redução da complexidade.

**Palavras-chave**: Sistemas Multicamadas; Virtualização; Xen.

**Abstract:**

*This paper discusses the advantages of hosting a multi-tier system in a virtualized environment instead a stand-alone environment. The techniques used by Xen for extract the maximum performance are presented. It also offers an overview how the technology has developed, and the impact on the economy of resources and the reduction of complexity.*

***Keywords**: Multi-tier Systems; Virtualization; Xen.*

## 1. Introdução

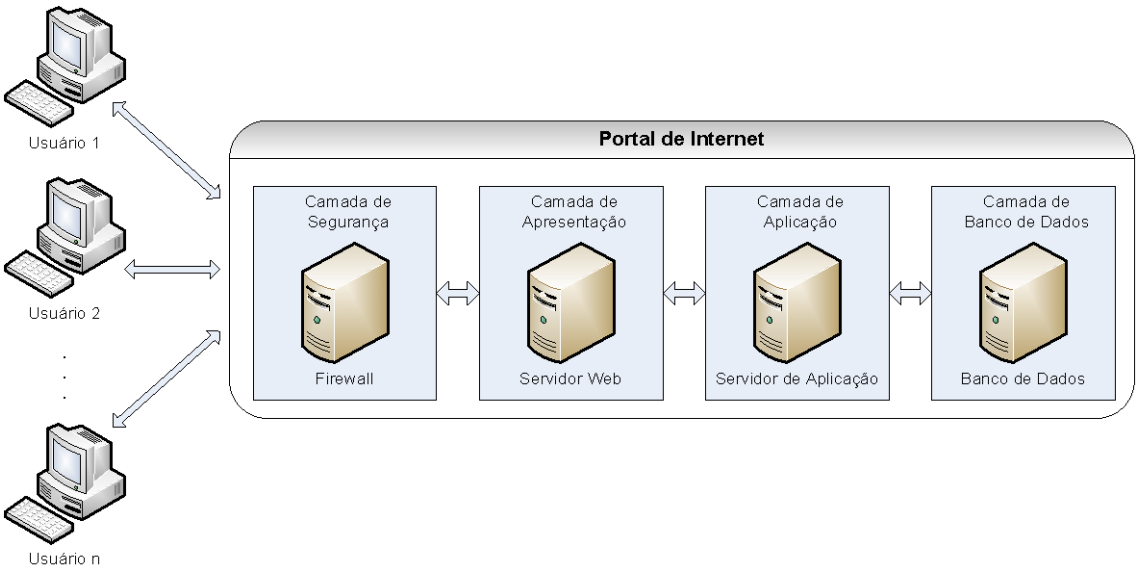
A seção 2 deste trabalho apresenta uma visão geral do que são sistemas multicamadas e suas implicações no projeto e dimensionamento de uma infra-estrutura de rede e servidores. A seção 3 explica de modo geral o que é virtualização e quais são suas vantagens, e oferece um breve histórico do desenvolvimento das tecnologias que proporcionaram o desenvolvimento dos sistemas virtualizados, bem como alguns aspectos de arquitetura de computadores que permitiram ser possível o advento da virtualização. Na seção 4 é apresentado um exemplo de como criar um sistema multicamadas em um ambiente virtualizado. Já na seção 5 é apresentado o que é alta disponibilidade e tolerância a falhas, e como isso pode impactar nos custos de implantação e manutenção de uma infra-estrutura de rede. Na seção 6 são explicados os componentes da arquitetura do Xen. Na seção 7 são apresentadas as vantagens do Xen e na seção 8 as características futuras que os mantenedores do Xen pretendem desenvolver. Por fim, na seção 9 são discutidas as considerações finais deste trabalho.

[1] Mestre em Engenharia de Informação pela UFABC; Consultor Técnico em Informática da Câmara Municipal de São Paulo.

## 2. Sistemas multicamadas

Sistemas multicamadas são sistemas compostos por mais de um subsistema de *software* que é, geralmente, de tipos diferentes. Por exemplo, um portal de Internet pode compor-se de um *firewall*, um servidor de páginas web, um servidor de aplicação e um servidor de banco de dados. Geralmente estes subsistemas estão em máquinas distintas porque cada uma delas requer necessidades específicas de *hardware*, como por exemplo, especificações diferentes de processador, memória e disco. A Figura 01 mostra como estes subsistemas podem ser interligar.

Figura 01. Arquitetura de um sistema multicamadas




No exemplo da Figura 01, um usuário abre um navegador de Internet em sua estação de trabalho e conecta-se à página do portal, tendo como primeira porta de entrada o *firewall*. Se o *firewall* atuar apenas como um filtro de pacotes, a estação cliente poderá estabelecer uma sessão direta com o servidor web. Mas no caso do *firewall* atuar como um *proxy*, é ele quem transmitirá a requisição de página para o servidor web, que irá montá-la de acordo com as regras de negócio do servidor de aplicação, que por sua vez consultará as informações constantes do banco de dados. Finalmente, o servidor web devolve a requisição para o *firewall* que a retransmite para a estação do usuário.

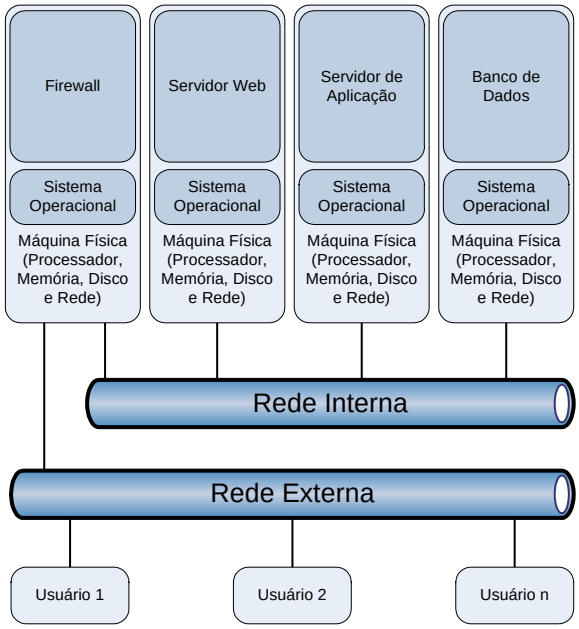
Outro aspecto importante desta arquitetura é quanto à comunicação entre os diferentes servidores: para se obter a máxima taxa de transferência possível e uma maior confiabilidade, faz-se necessário instalar um ambiente de rede com altas taxas de

velocidade e com tolerância a falhas. Estas interconexões por sua vez formarão os diversos pontos de falha do sistema, o que implicará a redução da confiabilidade.

**2.1. Um típico sistema multicamadas**

Um sistema multicamadas é formado por outros subsistemas separados, porque cada um destes necessita de *hardwares* diferentes. Se considerarmos um portal de Internet, como o da Figura 02, por exemplo, o *firewall* deve ser capaz de receber e encaminhar pacotes de rede em alta velocidade e possuir mais de uma placa de rede para isolar o tráfego entre as redes interna e externa. Os servidores web e de aplicação terão de manipular uma grande quantidade de requisições simultâneas, o que exige boa capacidade de processamento e quantidade suficiente de memória para manter o estado de cada sessão ou requisição. Já o servidor de banco de dados precisa ter um sistema de disco rápido o suficiente de forma a dar conta das requisições de consulta e/ou gravação nas tabelas do banco de dados.

Figura 02. Exemplo de um sistema multicamadas




No exemplo da Figura 02, pode-se notar que o *firewall* possui duas conexões de rede, para poder isolar o tráfego entre as redes interna e externa e prover uma camada de segurança ao sistema. Para aumentar o nível de confiabilidade e disponibilidade do sistema de modo geral, é necessária uma arquitetura de rede tolerante a falhas e que seja capaz de suportar altas taxas de transmissão entre os servidores.

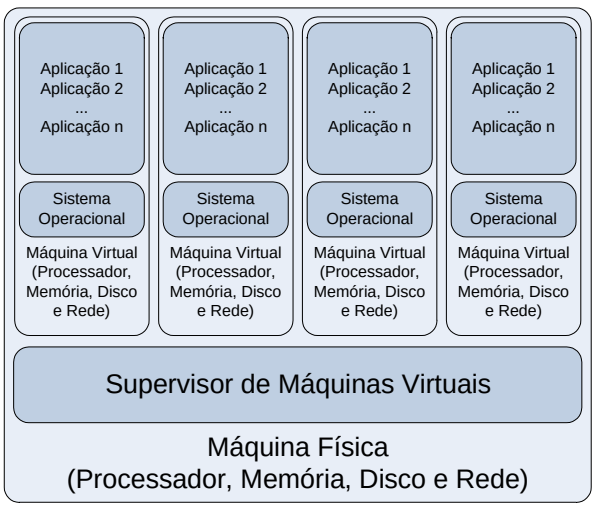
## 3. Ambiente virtualizado

A virtualização pode ser definida como uma metodologia de divisão de recursos de *hardware* de um computador em múltiplos ambientes de execução, por meio da aplicação de um ou mais conceitos ou tecnologias como particionamento de *hardware* e *software*, compartilhamento de tempo de máquina, simulação completa ou parcial, emulação, qualidade de serviço, e muitos outros (WILLIAMS, 2007). Um ambiente virtualizado nada mais é do que um servidor robusto em que é simulado por *software* o funcionamento de diversos outros servidores *stand-alone*, que podem estar ou não interconectadas entre si.

A tecnologia de virtualização consiste em fazer com que um único servidor físico seja particionado em vários servidores virtuais. Na prática a virtualização é tão transparente que o usuário não nota se está lidando com uma máquina real ou virtual.

Cada máquina virtual usa uma fração dos recursos da máquina física, como disco, memória, processador e rede (BARHAM, 2006). É o *software* de virtualização, também conhecido como Supervisor de Máquinas Virtuais, quem controla o acesso das máquinas virtuais a estes recursos. A Figura 03 mostra uma máquina física particionada em quatro máquinas virtuais. Ao invés de se instalar um sistema operacional comum na máquina física, instala-se um sistema operacional modificado que inclui um Supervisor de Máquinas Virtuais.

Figura 03. Arquitetura de máquinas virtuais




As vantagens de se colocar diversas máquinas virtuais dentro de uma mesma máquina física podem ser observadas a seguir (WILLIAMS, 2007):

- menor custo com energia e refrigeração de ambientes de processamento de dados;
- menor custo com aquisição de novos equipamentos;
- menor custo com aquisição de licenças de *software* que são cobradas por unidade de processador;
- aumento da utilização dos recursos da máquina e conseqüente redução da ociosidade;
- simplificação de migração de *software* legado;
- facilidade para realocar recursos entre partições existentes.

Além das vantagens descritas acima, outro aspecto interessante da virtualização pode ser observado. Caso a máquina virtual necessite comunicar-se com o mundo exterior, ela deve compartilhar o recurso de rede com a máquina física. Mas se esta máquina virtual precisar comunicar-se diretamente com outra dentro do mesmo ambiente virtualizado, o Supervisor de Máquinas Virtuais pode criar uma rede virtualizada entre as duas máquinas para permitir a comunicação. Essa rede nada mais é do que um esquema de compartilhamento de memória, que é muito mais rápido do que qualquer implementação de rede de alta velocidade, já que a comunicação é feita usando recursos de *software* e não de *hardware* (MENON, 2006) (MENON, 2005).

### 3.1. Funcionamento de um sistema virtualizado

A virtualização já existe há mais de três décadas, quando era usada apenas em dispendiosos *mainframes* e exigia *software* caro. Hoje, com o advento de novas tecnologias e com a redução do custo de *hardware*, já é possível implementar projetos de consolidação de servidores usando o método de virtualização a um custo bem acessível.

O supercomputador Atlas, construído no início dos anos 60 pelo Departamento de Engenharia Elétrica da Universidade de Manchester, se valeu de recursos como compartilhamento de tempo, multiprogramação e controle de periféricos compartilhados, e se tornou o precursor dos sistemas virtualizados. O Atlas separava os processos do sistema operacional em dois componentes: um chamado de supervisor e

outro responsável pela execução dos programas. O supervisor gerenciava os recursos chave, como tempo de processamento, por exemplo, e disponibilizava um ambiente de execução para os programas. Nascia aí o que se conhece hoje como monitor de ambientes ou máquinas virtuais (WILLIAMS, 2007). O Atlas introduziu também os conceitos de memória virtual, chamado de armazenamento de um nível, e técnicas de paginação para sistemas de memória. A memória do sistema operacional era logicamente separada da memória reservada aos programas, apesar de serem uma só fisicamente. A técnica usada para essa separação lógica abriu caminho para a criação de camadas de abstração, tão comuns hoje entre as diversas tecnologias de virtualização (WILLIAMS, 2007).

Atualmente, o que o *software* de virtualização faz é adicionar uma camada de abstração entre o *hardware* e as aplicações. Quanto maior afinidade houver entre o *hardware* e o *software* de virtualização, melhor será o aproveitamento de recursos e maior será o desempenho geral do sistema.

Essa relação entre o *software* e o *hardware* é tão importante que o Xen foi desenvolvido levando-se em conta as particularidades de uma das plataformas mais usadas nas empresas: a plataforma baseada na especificação Intel x86.

Máquinas baseadas na arquitetura x86 possuem níveis de privilégio diferentes para execução de instruções. Os atuais sistemas operacionais consideram dois níveis de instruções: um chamado de modo supervisor e outro de modo usuário.

O modo supervisor é um modo de execução que permite ao programa executar todas as instruções de acesso privilegiado, como acesso a dispositivos de entrada e saída e operações de gerenciamento de memória (WILLIAMS, 2007). Já no modo usuário, o programa só pode acessar as instruções de acesso privilegiado por meio de chamadas ao modo supervisor, que serve como intermediário, e também só pode acessar o espaço de memória previamente alocado. Desta forma, um programa "mal comportado" não interferiria na execução de outro programa, mantendo assim o sistema estável.

Considerando esses detalhes da arquitetura x86, o gerenciador de máquinas virtuais operaria no modo supervisor, e cada máquina virtual seria instanciada no modo usuário, de forma que a indisponibilidade de uma determinada máquina virtual não afetasse a estabilidade das demais.

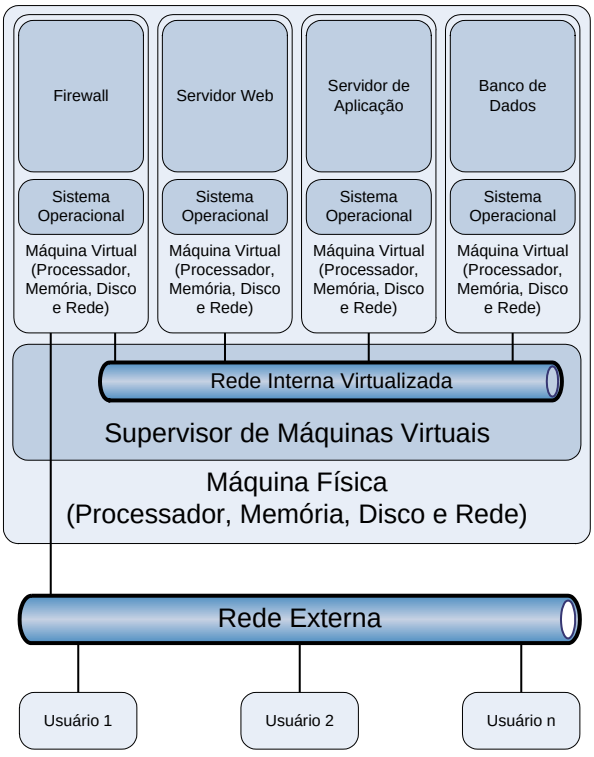
## 4. Sistema multicamadas em ambiente virtualizado

No exemplo da Figura 04, uma única máquina física está particionada em quatro máquinas virtuais. O *firewall* possui duas interfaces de rede: uma para comunicar-se com a rede externa e outra com a rede interna. A primeira interface é compartilhada com a placa de rede física do servidor. A segunda interface é uma placa de rede lógica conectada numa rede virtualizada, em que estão conectados os outros servidores.

Para implementar os processadores virtuais, cada máquina virtual usará uma fatia de tempo do processador físico. Se a máquina física tiver mais de um processador, eles podem ser alocados individualmente para cada máquina virtual. De forma análoga se dará a alocação de memória, ou seja, cada máquina virtual irá usar uma fatia da memória física.

E para disponibilizar os discos virtuais, cada máquina virtual alocará uma fatia do disco rígido da máquina física. Como o servidor de banco de dados tem por característica executar operações que exijam uma alta taxa de escrita e gravação, se a máquina física possuir mais de um disco rígido, este pode ser alocado exclusivamente para o servidor de banco de dados, de modo a aumentar o desempenho.

Figura 04. Exemplo de sistema multicamadas em ambiente virtualizado



## 5. Alta disponibilidade e tolerância a falhas

Há muitos ambientes de Tecnologia da Informação que suportam aplicações de missão crítica e que precisam estar acessíveis e funcionando ininterruptamente. Ambientes de TI que atendem a estes requisitos são chamados de ambientes de alta disponibilidade. Como exemplos de ambientes que atendam a esses critérios, pode-se destacar: sistemas de controle e gerenciamento de usinas de geração de eletricidade, sistemas de suporte à vida em hospitais, sistemas de controle de tráfego aéreo, entre outros. A implementação de sistemas de alta disponibilidade se faz por meio da instalação de componentes sobressalentes e até mesmo com o espelhamento completo de servidores. Implementações deste tipo são comumente conhecidas como *clusters*.

Já o termo tolerância a falhas refere-se à capacidade de elementos de *hardware* ou *software* continuarem funcionando mesmo depois de detectada uma falha que possa comprometer todo o sistema. Um exemplo típico de tolerância a falhas são os esquemas de correção de erros encontradas em bancos de memória, como o CRC (*Cyclic Redundancy Check* ou Checagem Cíclica de Redundância) e o ECC (*Error Correcting Code*, ou Código de Correção de Erro). Outro exemplo bastante comum é o uso de esquemas de discos que conseguem permanecer em funcionamento mesmo após um de seus discos falharem. Este esquema de disco é conhecido como RAID (*Redundant Arrays of Independent Disks* ou Matriz Redundante de Discos Independentes).

### 5.1. Implementando alta disponibilidade e tolerância a falhas por meio do Xen

O objetivo principal de se implementar alta disponibilidade e tolerância a falhas em ambientes de Tecnologia da Informação é garantir a continuidade dos negócios e a recuperação de desastres em casos de panes.

Um dos problemas de se usar sistemas multicamadas em máquinas *stand-alone* é a implementação da rede que conectará as diversas máquinas do sistema, pois cada conexão de rede deve ser entendida como um ponto potencial de falha.

Outra questão importante é o alto custo com *hardware*. Para implantar sistemas em regime de alta disponibilidade, é necessário lançar mão de implementações em *cluster*, que compõem-se de sistemas de armazenamento externo (conhecidos como *storage*), servidores com componentes redundantes e esquemas de discos espelhados e interconexões de rede baseadas em fibra ótica, com *switches* que comportem conexões

óticas. O alto custo de implantação e manutenção, combinado com os custos dos equipamentos de energia e refrigeração, fazem com que o uso de ambientes virtualizados seja um forte aliado na busca por racionalização de recursos, aumento de desempenho e redução da complexidade de infra-estrutura de ambientes de TI.

Em ambientes virtualizados, os custos com infra-estrutura são menores, e a complexidade de *hardware* também. Logo, gera-se uma grande economia na aquisição, manutenção e operacionalização dos equipamentos de informática necessários para as operações de uma empresa.

De fato, usando-se menos componentes de *hardware*, consequentemente os pontos de falhas diminuirão. Se considerarmos apenas os componentes de rede, haverá uma diminuição drástica do número de conexões, além de um aumento significativo no desempenho do tráfego de rede, devido ao fato de a comunicação entre máquinas virtuais ser feita usando recursos de *software* e não de *hardware* (MENON, 2006).

As vantagens da virtualização, quanto à confiabilidade, podem ser vistas a seguir (WILLIAMS, 2007):

- isolamento lógico de cada máquina virtual;
- criação de partições dedicadas para esquemas de tolerância a falhas do tipo ativo-passivo, em que a partição (passiva) assume no caso de uma partição em produção (ativa) falhar.

Portanto, pode-se notar que a virtualização é um aliado importante no desafio de consolidação de servidores e na redução da complexidade, que se reflete em menores custos e melhor eficiência do conjunto da infra-estrutura de tecnologia da informação, além de garantir uma maior confiabilidade do ambiente instalado.

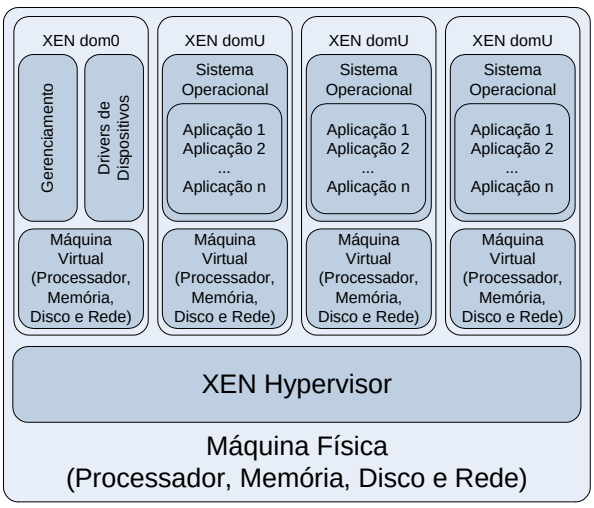
## 6. Arquitetura do Xen

O Xen é um gerenciador de máquinas virtuais que funciona tanto em máquinas de arquitetura de 32 bits ou 64 bits. Implementa uma camada logo acima do *hardware* físico e permite que várias máquinas virtuais possam compartilham os recursos (WILLIAMS, 2007).

Há muitas formas de se compartilhar recursos em uma mesma máquina. Uma delas se baseia no método de abrigar arquivos e processos para múltiplas aplicações em uma única instância do sistema operacional. Isso permite que se alcance um razoável nível de proteção entre os diferentes processos usando-se técnicas convencionais existentes nos sistemas operacionais (WILLIAMS, 2007). No entanto, uma falha num processo isolado pode diminuir o desempenho do sistema, mesmo que não o comprometa.

Para contornar isso, o Xen usa o esquema de domínios, conforme mostrado na Figura 05. O supervisor de máquinas virtuais, chamado de Xen Hypervisor, implementa uma camada de abstração que contém API´s de gerenciamento e de *hardware* virtual, incluindo uma interface de controle que permite às máquinas virtuais acessarem e compartilharem o *hardware* físico. Quando o sistema é iniciado, um domínio especial é instanciado, chamado de dom0. Este domínio contém o *software* de gerenciamento do modo usuário e os *drivers* de dispositivos, e é responsável por iniciar e parar outros domínios hospedeiros com menos privilégios, chamados de domU. Para garantir um alto desempenho, a comunicação entre os domínios é feita de maneira assíncrona, ou seja, uma máquina virtual pode continuar a executar seus processos sem ter de esperar pelo término das requisições de outras máquinas (WILLIAMS, 2007).

Figura 05. Arquitetura do Xen



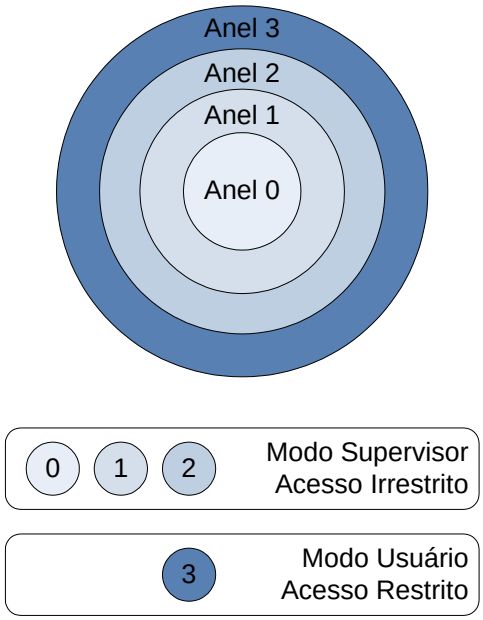

Fonte: (WILLIAMS, 2007)

## 6.1. Virtualização de processador

Virtualizar o processador é um grande desafio na arquitetura x86. Nos processadores x86, há quatro níveis distintos de camadas de privilégio, chamados de anéis. Eles vão do

anel 0 (mais privilegiado) até o anel 3 (menos privilegiado). O sistema operacional base geralmente é executado no nível 0, porque este é o único nível capaz de executar instruções privilegiadas. Já os níveis 1 e 2 não são usados desde o sistema operacional OS/2 da IBM, já obsoleto. Assim, qualquer sistema operacional que rode em máquina virtual pode ser portado para o Xen, desde que modificado de forma a ser executado no nível 1, impedindo que conflite diretamente com as instruções privilegiadas no nível 0 (BARHAM, 2006) (WILLIAMS, 2007).

Figura 06. Níveis de privilégios da arquitetura x86




Fonte: (WILLIAMS, 2007)

No caso das chamadas IRQ (*Interruption Request*, ou Requisição de Interrupção), cada máquina virtual, ou domU, mantém uma tabela única para manipular as chamadas IRQ. Quando um domU dispara uma requisição de interrupção, o dom0 captura essa requisição e executa as ações necessárias (BARHAM, 2006) (WILLIAMS, 2007).

### 6.2. Virtualização de memória

O Xen deve lidar com o gerenciamento da alocação de memória física entre os domínios e certificar-se de que os segmentos estarão isolados, para que não haja conflitos (BARHAM, 2006) (WILLIAMS, 2007).

Quando uma máquina virtual é inicializada, é alocada a ela uma quantidade inicial de memória, não necessariamente a quantidade total de memória pré-configurada. Toda

vez que a máquina virtual requisitar uma página de memória, ela irá reservá-la a partir da porção alocada inicialmente e que deverá ser validada pelo Xen. Todas as alocações posteriores também deverão ser validadas, de modo que o supervisor possa manter controle do ambiente (BARHAM, 2006).

### 6.3. Virtualização de dispositivos de entrada e saída

No Xen, a virtualização de dispositivos de entrada e saída segue esquemas similares à virtualização de CPU e de memória. Neste caso específico, a máquina virtual deve abstrair o *hardware* e também implementar os *drivers* de dispositivos para os diversos tipos de sistemas operacionais que podem ser emulados (WILLIAMS, 2007).

Os dados são transferidos entre os domínios usando esquemas de compartilhamento de memória e anéis descritores assíncronos, alcançando assim altas taxas de transferência (WILLIAMS, 2007).

Os principais dispositivos que podem ser emulados são: discos, portas seriais, portas paralelas, portas USB, placas de rede, placas de som, placas de vídeo e dispositivos SCSI (*Small Computer System Interface*), usados para conectar discos externos.

## 7. Vantagens do Xen

O Xen não é o único *software* de virtualização disponível, mas é um dos que tem apresentado melhores vantagens em relação a outros *softwares* de virtualização, como o Microsoft Hyper-V e o VMware (WILLIAMS, 2007) (MENON, 2005):

- máquinas virtuais com desempenho muito próximo ao de máquinas reais;
- suporte completo a máquinas x86 de 32 bits e x86 com extensões de 64 bits;
- suporte a quase todos os *drivers* de dispositivos disponíveis para Linux;
- suporte até 32 processadores em cada máquina virtual, desde que a máquina física suporte a mesma quantidade de processadores;
- alocação dinâmica de recursos;
- tempo de parada quase zero para migração de máquinas virtuais em execução entre duas máquinas físicas virtualizadas;
- suporte a processadores com características de virtualização por *hardware*.

Outra vantagem importante do Xen é quanto ao fato de seu licenciamento se dar por meio da GPL (*General Public License*, ou Licença Pública Geral), o que lhe confere o *status* de *software* livre ou *software* de código aberto. Isso faz com que o custo de aquisição seja baixo e o número de desenvolvedores ao redor do mundo seja crescente, o que permite a constante melhoria do produto e a correção rápida dos *bugs*.

## 8. Características futuras do Xen

Dentre as características futuras do Xen, nas quais o time de desenvolvimento vem trabalhando, podemos destacar as seguintes (WILLIAMS, 2007):

- melhoria do desempenho e aumento da escalabilidade;
- suporte a novos sistemas operacionais e novas arquiteturas de *hardware* e *software*;
- extensão do suporte para além da arquitetura x86 de processadores;
- melhoramentos gerais na arquitetura do produto.

O time de desenvolvimento do Xen ainda prevê o suporte à computação paralela de alto desempenho, fazendo com que o Xen seja compatível com a arquitetura NUMA (*Non Uniform Memory Architecture*, ou Arquitetura de Memória não Uniforme). A arquitetura NUMA é baseada no conceito de que cada CPU pode acessar a memória local mais rapidamente do que uma memória não local ou remota. A arquitetura NUMA suporta o escalamento de CPU´s em esquemas de multiprocessamento simétrico (SMP) desde que cada CPU mantenha sua própria memória local e um controlador lógico (WILLIAMS, 2007).

Outra área que os desenvolvedores do Xen querem melhorar é o suporte a processadores com vários núcleos (WILLIAMS, 2007).

## 9. Considerações finais

Os ambientes de Tecnologia da Informação que implementam serviços de redes em multicamadas geralmente têm de lidar com os altos custos de *hardware* e de infra-estrutura, sem contar a necessidade de ter de contar com uma equipe técnica

especializada. E estes problemas se tornam mais críticos à medida que o ambiente necessite estar no ar 24 horas por dia, 7 dias por semana.

Recentemente as empresas têm buscado soluções de consolidação de servidores, para diminuir os custos de manutenção dos *datacenters*, e também para aproveitar a ociosidade dos servidores. Há várias formas de se consolidar servidores, seja colocando serviços idênticos em uma mesma máquina, adquirindo equipamentos montáveis em *racks*, ou usando o conceito de máquinas virtuais (virtualização).

Ao utilizar sistemas virtualizados, dois objetivos importantes podem ser alcançados: primeiro, os custos gerais com infra-estrutura, incluindo custos com manutenção, refrigeração de ambientes, energia, etc. podem ser reduzidos de maneira bastante significativa; e em segundo lugar, a implementação de alta disponibilidade e tolerância a falhas podem ser conseguidas de forma mais simples e com custo reduzido.

Sendo assim, virtualizar ambientes de TI tem sido um aliado importante no aumento da eficiência do ambiente e na redução do TCO (*Total Cost Ownership*, ou Custo Total de Propriedade), que nada mais é do que a estimativa de custos diretos e indiretos associados com a aquisição, manutenção e atualização do parque de *hardware* e *software*, bem como o treinamento dos recursos humanos envolvidos.

**Referências Bibliográficas**

BARHAM, Paul; DRAGOVIC, Boris; FRASER, Keir; HAND, Steven; HARRIS, Tim; HO, Alex; NEUGEBAUER, Rolf; PRATT, Ian; WARFIELDB, Andrew. *Xen and the Art of Virtualization*. Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP), Outubro de 2003.

MENON, Aravind; COX, Alan L.; ZWAENEPOEL, Willy. *Optimizing Network Virtualization in Xen*. Proceedings of the 2006 USENIX Annual Technical Conference, Junho de 2006.

MENON, Aravind; SANTOS, Jose Renato; TURNER, Yoshio; JANAKIRAMAN, G. John; ZWAENEPOEL, Willy. *Diagnosing Performance Overheads in the Xen Virtual Machine Environment*. First ACM/USENIX Conference on Virtual Execution Environments (VEE'05), Junho de 2005.

WILLIAMS, David E.; GARCIA, Juan. *Virtualization with Xen(tm):* Including Xenenterprise, Xenserver, and Xenexpress. Syngress Publishing, Inc. Burlington, 2007.