Capítulo
4

Virtualização: da teoria a soluções

Alexandre Carissimi
Instituto de Informática
Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15064 CEP 91501-970 – Porto Alegre – RS - Brasil

Abstract

Virtualization is a technique which allows a computer system to be partitioned on multiples isolated execution environments similar to a single physical computer. Such environments are called virtual machines (VM). Each VM can be configured on an independent way having its own operating system, applications, Internet services and network parameters. It is possible to interconnect virtual machines as if they were a physical one. Some virtualization tools offer virtual network support like switches and routers. Firewalls and VPN can be also used among VMs. This work presents virtualizations techniques, its implementations issues and its use on IT infrastructures. We discuss the different virtualizations approaches, hardware support, tools, advantages and disadvantages on using virtualization as a business strategy.

Resumo

Virtualização é a técnica que permite particionar um único sistema computacional em vários outros denominados de máquinas virtuais. Cada máquina virtual oferece um ambiente completo muito similar a uma máquina física. Com isso, cada máquina virtual pode ter seu próprio sistema operacional, aplicativos e serviços de rede (Internet). É possível ainda interconectar (virtualmente) cada uma dessas máquinas através de interfaces de redes, switches, roteadores e firewalls virtuais, além do uso já bastante difundido de VPN (Virtual Private Networks). É objetivo deste minicurso fornecer os conceitos básicos para compreender no que consiste a virtualização, suas formas de implementação, compromissos e vantagens para uma infra-estrutura de TI.

4.1. Introdução

A virtualização é um assunto que tem sido atualmente destaque no mundo da tecnologia da informação (TI), apesar de não ser exatamente uma novidade. A introdução da linguagem de programação Java trouxe consigo a noção de máquina virtual, mas a origem das máquinas virtuais remonta ao início dos anos 70. Nessa época, era comum que cada computador (*mainframe*), mesmo de um único fabricante, tivesse seu próprio sistema operacional, e isso se tornou uma das principais razões para o aparecimento das máquinas virtuais: permitir que software legado executasse nos caros *mainframes*. Na prática, o software não incluía apenas a aplicação, mas sim todo o ambiente operacional sobre o qual ele executava. Essa abordagem foi usada com sucesso pela IBM que, na linha de *mainframes* 370 e seus sucessores, oferecia uma máquina virtual, portada para várias de suas plataformas, sobre a qual as aplicações executavam. Dessa forma era possível executar, ou migrar, uma aplicação de uma plataforma para outra desde que houvesse uma versão de máquina virtual para a plataforma alvo. Uma máquina virtual nada mais é que uma camada de software que oferece um ambiente completo muito similar a uma máquina física. Com isso, cada máquina virtual pode ter seu próprio sistema operacional, bibliotecas e aplicativos.

À medida que os computadores começaram a se tornar mais comuns, a quantidade de sistemas operacionais convergiu para algumas poucas famílias (Unix, Macintosh e Microsoft), cada uma com um público-alvo e um conjunto de aplicativos. Nesse contexto, a virtualização deixava de ser um problema importante. No entanto, o aumento do poder computacional dos atuais processadores, a disseminação de sistemas distribuídos e a onipresença das redes de computadores causaram, por várias razões, o ressurgimento da virtualização.

Hoje em dia é muito difícil imaginar um sistema computacional que não seja conectado em rede. Na prática, essa conectividade faz com que os administradores de sistemas sejam responsáveis por manter um conjunto grande e heterogêneo de servidores, cada um executando uma aplicação diferente, que podem ser acessadas por clientes também heterogêneos. É comum encontrarmos em infra-estruturas de rede uma filosofia "um servidor por serviço" por razões que variam desde suporte a heterogeneidade dos clientes à segurança. Normalmente, nesse contexto, a carga de processamento de um servidor não explora todo o potencial disponibilizado pelo processador. Há um desperdício de ciclos de processamento e, por conseqüência, de investimento. A virtualização surge como uma opção para contornar esse problema.

Inicialmente, a virtualização pode auxiliar a se trabalhar em um ambiente onde haja uma diversidade de plataformas de software (sistemas operacionais) sem ter um aumento no número de plataformas de hardware (máquinas físicas). Assim, cada aplicação pode executar em uma máquina virtual própria, possivelmente incluindo suas bibliotecas e seu sistema operacional que, por sua vez, executam em uma plataforma de hardware comum. Em outras palavras, isso não deixar de ser um retorno à situação de executar software herdado em um sistema diferente daquele para o qual ele foi projetado. Assim, a virtualização proporciona um alto grau de portabilidade e de flexibilidade permitindo que várias aplicações, de sistemas operacionais diferentes, executem em um mesmo hardware. Ao se executar múltiplas instâncias de máquinas

virtuais em um mesmo hardware, também se está proporcionando um uso eficiente de seu poder de processamento. Essa situação é comumente denominada de consolidação de servidores e é especialmente interessante em *data centers* devido a heterogeneidade de plataformas inerente ao próprio negócio. Além disso, em *data centers*, a diminuição de máquinas físicas implica na redução de custos de infra-estrutura física como espaço, energia elétrica, cabeamento, refrigeração, suporte e manutenção a vários sistemas.

A flexibilidade e a portabilidade das máquinas virtuais também tornam interessante o uso da virtualização em *desktops*. É possível imaginar, por exemplo, o desenvolvimento de produtos de software destinados a vários sistemas operacionais sem ter a necessidade de uma plataforma física para desenvolver e testar cada um deles. Assim, as máquinas virtuais em *desktops* podem ser usadas para se definir ambientes experimentais sem comprometer o sistema operacional original da máquina, ou ainda, para compor plataformas distribuídas como clusters e grades computacionais.

Nos últimos anos, dada a importância e a gama de aplicações em que a virtualização pode ser empregada, houve um investimento maciço nesta tecnologia por parte de fabricantes de processadores e no desenvolvimento de produtos de software. Os processadores mais recentes da Intel e da AMD contam no seu projeto com mecanismos e soluções de hardware especialmente destinados a dar suporte a virtualização.

As máquinas virtuais, por emularem um ambiente computacional sobre outro impõem algumas restrições de implementação e de desempenho. É aqui que entra o desenvolvimento dos produtos de software para a virtualização. Basicamente, as máquinas virtuais podem ser implementadas como uma aplicação de um sistema operacional e executarem em modo usuário, ou serem uma camada de software posicionada entre o hardware da máquina e o sistema operacional. A primeira opção é o que se denomina de máquina virtual de processo e a segunda de monitor de máquina virtual ou *hypervisor* [Smith e Nair, 2005]. Ainda, um monitor de máquina virtual pode prover virtualização total ou para-virtualização, dependendo se, respectivamente, o sistema operacional que executa na máquina virtual deve ou não ser adaptado para tal.

A partir do momento que se define máquinas virtuais surge quase que imediatamente a necessidade de conectá-las em rede. Todas as máquinas virtuais existentes provêm interfaces de redes virtuais idênticas a suas similares reais, isso é, possuem endereços MAC e podem ser interligadas em equipamentos de interconexão de redes, como *switches* e roteadores. Tipicamente, isso é feito colocando a interface de rede física em modo promíscuo e multiplexando seu uso entre as diferentes interfaces virtuais. Um passo adiante é fornecer toda uma infra-estrutura virtual de rede, criando processos (*daemons*) que emulam *switches* e roteadores, ou incluindo esse tipo de suporte na própria infra-estrutura de virtualização.

Com base no que foi mencionado até o momento é possível imaginar que a virtualização oferece uma gama bastante grande de aplicações. O objetivo deste trabalho é apresentar os principais conceitos da virtualização, discutir aspectos relacionados com sua utilização e mostrar as principais ferramentas existentes para sua implantação. Para atingir esse objetivo este trabalho é organizado em oito seções incluindo esta introdução.

Inicialmente, a seção 4.2, fornece uma visão geral no que consiste a virtualização e faz uma revisão de conceitos básicos de sistemas operacionais necessários para uma melhor compreensão da virtualização e de sua implementação. A seção 4.3, por sua vez, apresenta a virtualização sob o ponto de vista de redes de computadores abordando seu uso na implantação de serviços Internet e na definição da infra-estrutura de rede. Atualmente existem várias ferramentas que oferecem suporte a virtualização, tanto soluções proprietárias quanto em software livre, as mais comuns, VMware, Xen e Microsoft, são apresentadas na seção 4.4. A seção 4.5 ilustra o emprego da virtualização através de estudos de casos, para isso foram selecionadas três aplicações significativas: consolidação de servidores, virtualização em *desktops* e a definição de *honeypots*. A adoção da virtualização em uma infra-estrutura de TI deve ser feita considerando uma série de aspectos, pois ela afeta sobremaneira como novos serviços podem ser agregados à infra-estrutura de rede, na aquisição de novos equipamentos e muda a filosofia de administração de sistemas. O objetivo da seção 4.6 é discutir alguns desses aspectos de forma a orientar o leitor sobre os principais pontos a serem avaliados quando da análise sobre adotar a virtualização. A seção 4.7 complementa essa análise abordando aspectos como segurança, gerenciamento e desempenho de máquinas virtuais. Por fim, a seção 4.8, que apresenta a conclusão deste trabalho.

4.2. Virtualização

Qualquer pessoa que atualmente use um computador sabe que existe algo denominado de sistema operacional que, de alguma forma, controla os diversos dispositivos que o compõe. A definição clássica para sistema operacional, encontrada em vários livros, é a de uma camada de software inserida entre o hardware e as aplicações que executam tarefas para os usuários e cujo objetivo é tornar a utilização do computador, ao mesmo tempo, mais eficiente e conveniente [Silberschatz, 2001].

A utilização mais eficiente busca um maior retorno no investimento feito no hardware. Maior eficiência significa mais trabalho obtido pelo mesmo hardware. Isso é obtido através da distribuição de seus recursos (espaço em memória principal, processador, espaço em disco, etc) entre diferentes programas. Cada programa tem a ilusão de estar executando sozinho no computador quando na realidade ele está compartilhando com os demais. Uma utilização mais conveniente do computador é obtida, escondendo-se do usuário detalhes de hardware, em especial, dos periféricos de entrada e saída. Tipicamente, isso é feito através da criação de recursos de mais alto nível oferecido através de interfaces gráficas. Por exemplo, os usuários usam espaço em disco através do conceito de arquivos. Arquivos não existem no hardware. Eles formam um recurso criado a partir do que o hardware oferece. Genericamente, isso é denominado de virtualização de recursos.

Um conceito importante em sistemas operacionais é de processo. Um processo é uma abstração que representa um programa em execução. Cada processo é um ambiente de execução isolado dos demais processos que executa sobre um processador lógico, isto é, um processador virtual, vinculado a si no momento da criação do processo. Cabe ao núcleo do sistema operacional, através de seu escalonador, alternar os diferentes processadores lógicos (virtuais) sobre um processador físico. A ilusão de paralelismo é

criada pelo chaveamento rápido entre os processos. Na realidade, os sistemas operacionais atuais possuem duas abstrações para unidade de execução: processos e threads. Entretanto, continua válida a noção de um processador virtual por unidade de execução. Um estudo desses conceitos extrapola o escopo deste trabalho e o leitor mais interessado pode obter mais detalhes em [Silberschatz, 2001][Oliveira, Carissimi e Toscani, 2004].

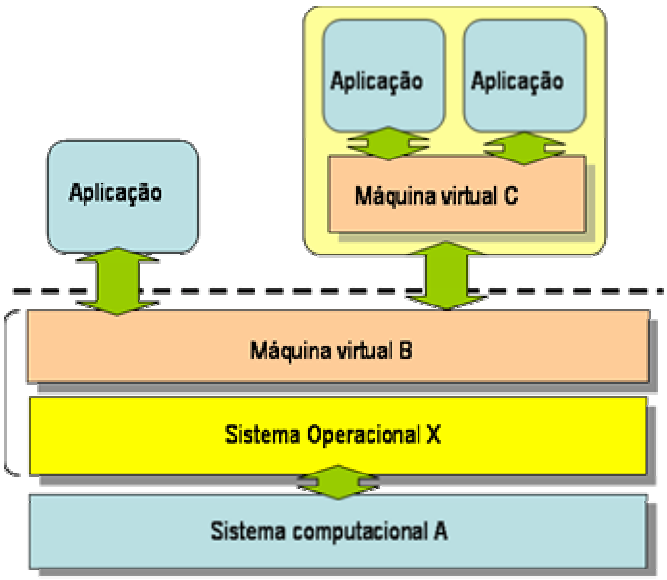

Figura 4.1 – Princípio básico de máquinas virtuais

Em sua essência, a virtualização consiste em estender ou substituir um recurso, ou uma interface, existente por um outro, de modo a imitar um comportamento. Isso é ilustrado genericamente na figura 4.1. Por exemplo, sobre o hardware do sistema computacional A é posto uma camada de software, o sistema operacional X, que fornece a ilusão de uma máquina virtual B para as aplicações do sistema operacional X. No entanto, uma dessas aplicações pode ser a implementação de uma máquina virtual C que, por sua vez, oferece um ambiente de execução para aplicações escritas para a máquina virtual C. Um exemplo prático disso, é a máquina virtual java (JVM – Java Virtual Machine) que permite que aplicações Java executem um ambiente virtual a JVM implementado para o sistema operacional GNU/Linux ou para o Windows.

4.2.1 Conceitos básicos

Para entender o que é uma máquina virtual, é interessante relembrar alguns aspectos fundamentais de sistemas operacionais. Um conceito importante em sistemas operacionais é o de processo. Um processo é uma abstração que representa um programa em execução. O processo é representado por um espaço de endereçamento lógico composto por regiões de texto, dados, pilha e *heap* (figura 4.2). A região de texto é onde residem as instruções do programa a serem executadas. A região de dados mantém todas as variáveis globais, inicializadas ou não.A pilha serve para armazenar o endereço de retorno de chamadas de função, para a passagem de parâmetros, além de ser também a área onde são armazenadas as variáveis locais de uma função. Por fim, a região de *heap* que serve para a alocação dinâmica de porções de memória.

A execução de um processo é acompanhada por dois registradores lógicos (virtuais): contador de programa (*Program Counter* - PC) e o apontador de pilha (*Stack Pointer* – SP). O contador de programa indica a instrução a ser executada e o apontador de pilha onde devem ser armazenados o endereço de retorno de uma chamada de função, seus parâmetros e suas variáveis locais. Cabe ao sistema operacional, através de seu escalonador e do *dispatcher*, mapear os registradores lógicos PC e SP para os registradores físicos PC e SP – e únicos – do processador. Isso é o que se denomina de chaveamento de contexto[1]. A alternância entre os diversos PC e SP lógicos, dos diferentes processos, nos registradores PC e SP físicos fornece a ilusão de que vários processos estão executando simultaneamente (figura 4.2). Portanto, um processo nada mais é que um tipo de máquina virtual que executa um único programa.

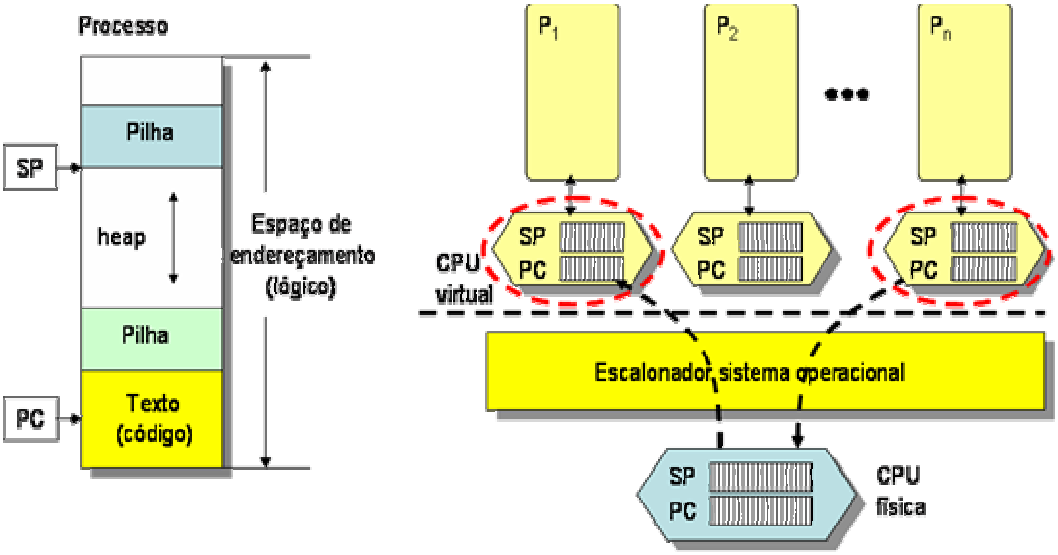

Figura 4.2 – A abstração de processo em um sistema operacional

Outro princípio importante em sistemas operacionais é a sua estruturação em camadas hierárquicas com diferentes níveis de abstrações e interfaces. Um computador é um sistema de computação relativamente complexo devido à variedade de componentes de hardware e de software que o constitui e a sua interação. Entretanto, essa complexidade nunca foi um empecilho para o crescimento e a evolução dos computadores em geral devido ao seu projeto de forma hierárquica, com diferentes níveis de abstração e com interfaces bem definidas entre esses níveis. O uso de níveis de abstração e interfaces, tanto para os componentes de software, como para os de hardware, permitiu que cada componente fosse visto como um subsistema independente oferecendo serviços para os demais. Os detalhes internos de implementação, de cada um deles, não precisam ser conhecidos: basta conhecer as interfaces e os serviços

[1] O chaveamento de contexto é mais complexo do que simplesmente agir sobre o PC e o SP, pois envolve o salvamento de estado de todos os registradores físicos do processador e a atualização de uma série de tabelas internas do sistema operacional, mas para esta discussão, essa visão simplificada é suficiente.

oferecidos. Tipicamente, um sistema de computação oferece três tipos de interfaces: instruções de máquina (privilegiadas); instruções de máquina (não privilegiadas) e chamadas de sistema; e interface aplicativa de programação. A figura 4.3 ilustra essas interfaces.

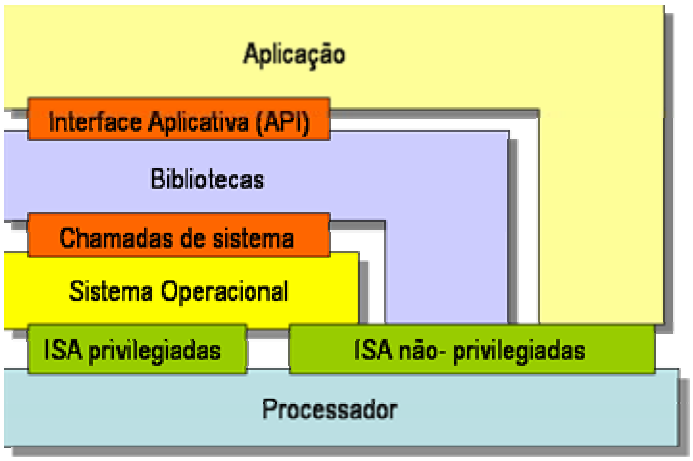

Figura 4.3 – Interfaces genéricas de um sistema de computação

O elemento central de um computador é seu processador. Cada processador tem um conjunto de instruções de máquina (ISA – *Instruction Set Architecture*) que pode seguir um determinado padrão. Por exemplo, Intel e AMD, fabricam processadores que implementam um mesmo padrão ISA, o Intel IA-32 (x86). Os projetistas de software compilam seu programas para obter códigos binários para um determinado ISA. Portanto, o conjunto de instruções (ISA) é uma interface entre o hardware e o software. Na realidade, as instruções de máquina são ainda divididas em dois grupos: privilegiadas e não-privilegiadas. Na prática, isso cria uma divisão nesse nível de interface que permite que apenas alguns programas, os com privilégios especiais, como o sistema operacional, possam executar todas as instruções, entre elas, as de manipulação de recursos de hardware, como entrada e saída e interrupções. Os programas de usuários executam apenas as instruções não-privilegiadas.

A segunda interface é composta pelas instruções de máquina não-privilegiadas e pelas chamadas de sistema, como as oferecidas por um sistema operacional. Essa interface possibilita que um programa de usuário execute instruções não-privilegiadas diretamente no processador, mas não permite o acesso aos recursos de hardware (instruções privilegiadas). As chamadas de sistema são uma forma de permitir que os programas de usuários acessem de forma indireta, e controlada, os recursos de hardware. Através delas, os programas de usuário executam, após ter sido garantido a autenticidade e a validade da operação, operações de acesso a recursos de hardware (tipicamente E/S).

A terceira interface consiste nas chamadas a funções de biblioteca que, em geral, são denominadas de interface aplicativa de programação (API – *Application Program Interface*). É comum que as chamadas de sistema sejam ocultadas por uma API.

Considerando essas interfaces, a implementação de máquinas virtuais pode ser feita de dois modos. Primeiro, é possível fazer um programa de aplicação que forneça um ambiente de execução para outras aplicações. Esse ambiente pode possuir um conjunto de instruções abstratas que são interpretadas para gerar as instruções de máquinas não-privilegiadas, as chamadas de sistema e de API de bibliotecas que correspondem à ação abstrata desejada. É o caso da máquina virtual java (JVM). É possível ainda que esse programa de aplicação emule chamadas de sistemas de outro sistema operacional, como ocorre quando se executa Linux em sistemas Windows com a ferramenta VMware *player*. Esse tipo de virtualização é o que se denomina de máquina virtual de processo (figura 4.4a) [Smith e Nair, 2005].

Uma abordagem alternativa é fornecer uma camada de software entre o hardware e o sistema operacional protegendo o acesso direto deste aos recursos físicos da máquina. Essa camada oferece como interface ao sistema operacional um conjunto de instruções de máquina que pode ser o mesmo do processador físico, ou um outro. O ponto importante é que essa interface deve estar disponível sempre que o computador estiver ligado e que ela possa ser usada simultaneamente por diferentes programas. O resultado final é que possível ter diversos sistemas operacionais (programas) executando independentemente na mesma plataforma. Genericamente, essa máquina virtual é referenciada como monitor de máquina virtual (*Virtual Machine Monitor* – VMM), também conhecido como *hypervisor*, ou ainda, máquina virtual de sistema (figura 4.4b) [Smith e Nair, 2005].

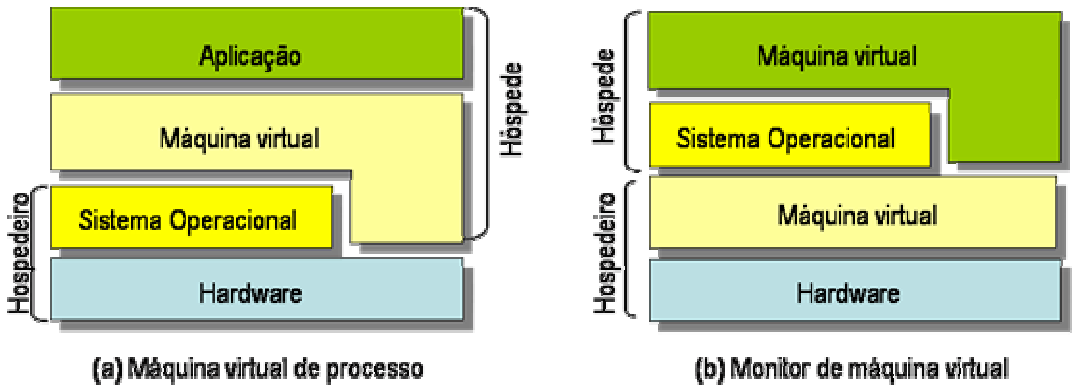

Figura 4.4 – Tipos de máquinas virtuais e sua relação com sistemas hóspede e hospedeiro.

Um ponto importante a destacar é que um processo é uma entidade efêmera, ou seja, ele existe apenas quando um programa está em execução. Portanto, uma máquina virtual de processo só existe enquanto este estiver executando. Já o monitor de máquina virtual (*hypervisor*) está presente sempre que o computador estiver ligado. O processo ou sistema que executa sobre uma máquina virtual é denominado de hóspede, enquanto o ambiente sobre o qual ele executa é chamado de hospedeiro. A figura 4.4 resume, de forma esquemática, as diferenças entre o uma máquina virtual de processo e monitor de máquina virtual e suas relações com sistema hóspede e sistema hospedeiro.

Na figura 4.4a, uma aplicação executa sobre um programa que implementa uma máquina virtual de processo (às vezes também chamado de *runtime* ou executivo). Nesse caso, a máquina virtual utiliza as funcionalidades providas pelo sistema operacional – como chamadas de sistemas e funções de biblioteca – e pelo próprio processador através de instruções não-privilegiadas. A aplicação emprega apenas a interface provida pelo ambiente (máquina) virtual. No outro caso, figura 4.4b, a aplicação executa usando as chamadas de sistemas e funções de bibliotecas providas por um sistema operacional, assim como as instruções não-privilegiadas oferecidas pelo monitor de máquina virtual. Ressalta-se que, nesse caso, o próprio sistema operacional executa as instruções oferecidas pelo monitor de máquina virtual, as quais podem ser as mesmas ou não do hardware (processador) subjacente.

4.2.2 Suporte de hardware para virtualização

Na seção anterior foi apresentado, sob o ponto de vista de software, no que consiste a idéia da virtualização. No entanto, da mesma forma que ocorre na implementação de sistemas operacionais convencionais, é necessário que o hardware do processador proveja mecanismos básicos que auxiliem o software na execução de tarefas consideradas essenciais.

Em 1974, Popek e Goldberg [Popek e Goldberg, 1974], introduziram três propriedades necessárias para que um sistema computacional oferecesse de forma eficiente suporte a virtualização:

- Eficiência: todas instruções de máquina que não comprometem o funcionamento do sistema devem ser executadas diretamente no hardware sem intervenção da máquina virtual.
- Controle de recursos: uma máquina virtual deve ter controle completo sobre os recursos virtualizados sendo estritamente proibido que um programa executando sobre a máquina virtual os acesse diretamente.
- Equivalência: um programa executando sobre uma máquina virtual deve exibir um comportamento idêntico àquele apresentado caso a máquina virtual não existisse e o programa acessasse diretamente uma máquina física equivalente. Duas exceções são consideradas. Primeira, eventualmente, algumas instruções podem ter seu tempo de execução aumentado. Segunda, pode haver problemas de conflito de acesso a recursos, os quais devem ser resolvidos de forma apropriada.

Essas propriedades se traduziram na classificação do conjunto de instruções de máquina de um processador (ISA) em três grupos e em dois teoremas. As instruções de máquina são divididas em: privilegiadas, que se executadas por um programa em modo usuário causam exceções (*trap*); sensíveis de controle, que permitem a alteração de recursos do sistema; e sensíveis comportamentais, cujo resultado ou comportamento dependem da configuração de recursos como, por exemplo, conteúdo de registradores internos ou modos de execução do processador.

O primeiro teorema diz que um monitor de máquina virtual (VMM) pode ser implementado se as instruções sensíveis de controle e comportamentais forem um subconjunto das instruções privilegiadas. Isso se traduz no fato que qualquer instrução que possa afetar o funcionamento da VMM deve passar por seu controle. O segundo teorema diz respeito a possibilidade de uma VMM executar uma instância de si mesma. Para melhor compreender os problemas relacionados com o processador e a virtualização é interessante analisar a arquitetura x86.

A arquitetura x86 provê quatro modos de operação para o processador, denominados de anéis de proteção (*rings*) ou CPL (*Current Privilege Level*), identificados de 0 a 3. Nos sistemas operacionais convencionais (Microsoft Windows e UNIXes) para esse tipo de arquitetura apenas dois modos são usados. O *ring* 0, que detém os maiores privilégios de execução, é usado pelo sistema operacional, e o *ring* 3, de menor privilégio é empregado pelos processos de usuário. Se um processo de usuário tentar executar uma instrução privilegiada ocorrerá uma exceção (*trap*) que deverá ser tratada adequadamente. Entretanto, a arquitetura x86, em especial o Pentium, possui dezessete instruções não privilegiadas [Robin e Irvine, 2000] que são sensíveis, ou seja, o teorema 1 é violado.

Na prática, a condição do teorema 1 é suficiente, mas não necessária, pois é possível implementar máquinas virtuais para arquiteturas que não o respeitam pagando-se o custo em desempenho. A virtualização nessas arquiteturas é feita tratando de forma apropriada as instruções consideradas como sensíveis. Duas técnicas são normalmente usadas. Primeira, as instruções sensíveis são identificadas em tempo de execução e geram um desvio para a VMM tratá-las adequadamente. Segunda, o programa a ser executado na VMM é modificado para que as instruções sensíveis sejam substituídas para chamadas a VMM. Essa técnica é conhecida como para-virtualização (seção 4.2.3).

Entretanto, os fabricantes de processadores, AMD e Intel, desenvolveram extensões para a arquitetura x86 para suportarem a virtualização. As extensões da AMD, denominada de AMD-V (AMD-*Virtualization*), codinome *Pacífica*, se aplica às arquiteturas x86 de 64 bits como o Athon, Turion, Phenom e as linhas mais recentes. A Intel apresenta suas extensões para as arquiteturas x86 de 32 e 64 bits. Essas extensões são conhecidas por IVT (*Intel Virtualization Technology*) ou pelo seu codinome, *Vanderpool*. As soluções da AMD e da Intel foram desenvolvidas independentemente uma da outra e são incompatíveis, embora sirvam para o mesmo propósito. A AMD implementa funções especiais no processador que são executadas por um *hypervisor* e que podem controlar, em seu nome, se determinados acessos de um sistema hóspede são permitidos. A Intel introduziu mecanismos similares (*virtual machines extensions*) que complementam a idéia do conceito de anéis de proteção com dois novos modos: *root* e não-*root*. Esses modos são controlados pelo *hypervisor* (que executa em modo *root*) e que pode transferir a execução de um sistema operacional hóspede para o modo não-*root* no qual instruções do anel zero são executadas sem risco para o sistema.

Por fim, uma conseqüência importante dessa discussão de suporte de hardware é que os softwares que implementam técnicas de virtualização podem apresentar problemas de compatibilidade, ou de desempenho, quando executados em determinados processadores. É importante verificar se a ferramenta de virtualização a ser empregada possui algum tipo de restrição quanto ao processador.

4.2.3 Virtualização total e paravirtualização

A implementação de máquinas virtuais de sistema ou monitores de máquinas virtuais (VMM) pode ser obtida através de duas técnicas: virtualização total ou completa e a para-virtualização.

A virtualização total (figura 4.5) consiste em prover uma réplica (virtual) do hadware subjacente de tal forma que o sistema operacional e as aplicações podem executar como se tivessem executando diretamente sobre o hardware original. A grande vantagem dessa abordagem é que o sistema operacional hóspede não precisa ser modificado para executar sobre a VMM. No entanto, essa abordagem tem alguns inconvenientes.

Primeiro, dada a diversidade de dispositivos existentes que compõem um computador, é muito difícil implementar uma máquina virtual que imite o comportamento exato de cada tipo de dispositivo. A solução consiste em prover na VMM suporte a um conjunto genérico de dispositivos. Tipicamente, cada VMM possui um teclado e mouse do tipo PS/2, unidades de floppy, controladores IDE, cd-rom ATAPI, portas seriais e paralelas, uma placa gráfica padrão e as placas de redes mais comuns em ambientes PC. Sendo assim, pode-se ter uma subutilização de um recurso de hardware real. Segundo, por não ser modificado, as instruções executadas pelo sistema hóspede devem ser testadas na VMM para saber se elas são sensíveis ou não, o que representa um custo de processamento. Terceiro, a implementação de VMM com virtualização total deve contornar alguns problemas técnicos devido a forma os sistemas operacionais são implementados. Por exemplo, um sistema operacional convencional (Linux ou Windows) implementa memória virtual através de paginação. Há toda uma gerência de alocação, liberação e controle de acesso às páginas que devem respeitadas. Grosso modo, é necessário "converter" o espaço de endereçamento do sistema hóspede para um real, disputando recursos com outro sistema hóspede. Tecnicamente, não há maiores empecilhos em se fazer isso, porém, esse tratamento também representa uma queda de desempenho.

A para-virtualização (figura 4.6) é uma abordagem alternativa que surge como forma de contornar as desvantagens da virtualização total. Nessa abordagem, o sistema hóspede é modificado para chamar[2] a VMM sempre que for executada uma instrução ou ação considerada sensível. Dessa forma, o teste por instrução não é mais necessário. Além disso, na para-virtualização os dispositivos de hardware são acessados por *drivers* da própria VMM.

[2] O termo normalmente empregado para tal é *hypercalll*, ou seja, a substituição da chamada de uma instrução sensível pela chamada a um tratador de interrupção de software (*trap*) com uma parametrização adequada de registradores.

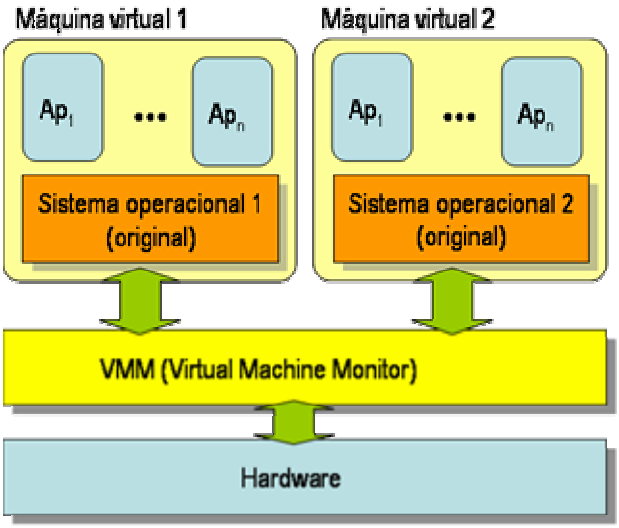

Figura 4.5 – Virtualização total

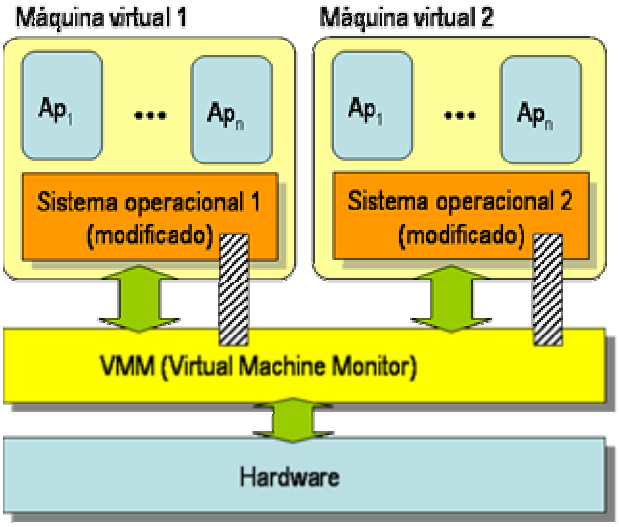

Figura 4.6 – Para-virtualização

Para concluir, convém ressaltar alguns pontos importantes desta discussão. A virtualização total permite que um sistema hóspede execute em uma VMM sem necessitar nenhuma alteração, ao passo que com a para-virtualização o sistema hóspede precisa ser modificado para "enxergar" a VMM. Por outro lado, a para-virtualização explora de maneira apropriada os recursos disponíveis pelo hardware real da máquina e apresenta um melhor desempenho que a virtualização total. Entretanto, face o atual suporte de hardware à virtualização presente nos processadores Intel e AMD, a virtualização total e a para-virtualização têm apresentado desempenhos semelhantes [XenSource 2008a]. A análise apresentada em [XenSource 2008a] consistiu em executar *benchmarks* como SPECcpu2000, Passmark e Netperf, que juntos simulam cargas de trabalho similares as apresentadas por *datacenters*.

4.2.4 Emulação, simulação e virtualização

Ao iniciar o estudo sobre virtualização, há dois conceitos que normalmente aparecem: simulação e emulação. Simulação, de acordo com dicionário Houaiss da língua portuguesa, é o ato de *representar com semelhanças certos aspectos de; imitar ou fingir*. Sob o ponto de vista computacional, um sistema físico, ou abstrato, pode ser descrito através de um modelo matemático que representa certas características chaves para o entendimento de seu comportamento ou funcionamento. Um simulador é um programa que implementa um modelo de sistema recebendo parâmetros de entrada e condições iniciais de contorno e serve para auxiliar na predição e análise de comportamento do sistema simulado. A simulação computacional tem sido uma técnica bastante importante em várias áreas como física, química, geociências, biologia, entre outras. Um tipo especial de simulação é a interativa, onde um agente externo humano participa do laço de simulação introduzindo eventos e reações que servem como parâmetros de entrada para um modelo. Esse é o caso de simuladores de vôo, de pilotagem e da maioria dos jogos de computadores.

Já emulação, também segundo o Houaiss, em um de seus sentidos, *é o esforço para imitar ou tentar seguir o exemplo de alguém*. Em termos computacionais, entende-se a capacidade de um programa de computador, ou de um dispositivo eletrônico, imitar outro programa ou dispositivo. Em si, um emulador é um programa que cria uma camada de software entre uma plataforma[3] hóspede e a plataforma hospedeira a ser imitada. Dessa forma, um emulador é capaz de executar os mesmos programas do sistema original (hóspede) sobre outro sistema (hospedeiro), reproduzindo os mesmos resultados para as mesmas entradas. É interessante salientar que não importa como o emulador faz isso, ou seja, seus estados internos, mas sim seu comportamento externo. Entretanto, R. Jain [Jain, 1991] define emulação como "*a simulation using hardware or firmware is called emulation*" e A. Singh [Singh, 2008] diz "*a simulator can be informally thought of as an accurate emulator*".

Da mesma forma, não existe uma definição consensual sobre o que é virtualização, uma usualmente empregada é aquela dada em [Singh, 2008]: virtualização é um *framework* ou metodologia para dividir os recursos de um computador em múltiplos ambientes de execução, aplicando um, ou mais conceitos, e tecnologias como particionamento de software ou hardware, tempo compartilhado, simulação completa ou parcial da máquina, emulação e qualidade de serviços. De maneira pragmática, virtualização pode ser definida como uma técnica que permite particionar um único sistema computacional em vários outros sistemas totalmente separados. Cada um desses sistemas é capaz de oferecer um ambiente, denominado de máquina virtual, que estende ou substitui uma interface existente de modo a imitar o comportamento de um outro sistema.

[3] Por plataforma se entende o conjunto composto pelo hardware e software de um sistema computacional, incluindo seu sistema operacional.

4.2.5 Outras técnicas de virtualização

Um desafio para a implementação de máquinas virtuais de processo é quando o sistema hóspede possui um conjunto de instruções diferente do sistema hospedeiro. A forma mais direta de resolver esse problema é através de interpretação. Um interpretador busca, decodifica, analisa, instrução por instrução, e traduz cada uma delas por seu equivalente no sistema hospedeiro. Apesar de viável, o interpretador introduz um custo de processamento que impacta no desempenho. Para contornar esse problema surge a técnica de tradução binária dinâmica [Ung, 2000] a qual converte as instruções do sistema hóspede para o sistema hospedeiro em blocos e as armazena em uma cache para reaproveitamento futuro.

É importante salientar que a técnica de tradução binária dinâmica é usada mesmo quando o sistema hóspede e o sistema hospedeiro possuem o mesmo conjunto de instruções. Nesse caso, a tradução ocorre no sentido de "ações", convertendo uma "ação" do sistema hóspede por sua ação equivalente no hospedeiro. Um exemplo é a possibilidade executar Linux sobre Microsoft Windows e vice-versa.

Outra técnica de virtualização que privilegia a portabilidade é a de máquinas virtuais de linguagem de alto nível. Nesse caso, a máquina virtual é definida e implementada levando em conta um ambiente de desenvolvimento composto por uma linguagem de alto nível e seu compilador. O código compilado não corresponde a nenhuma arquitetura real existente, mas sim a máquina virtual definida. Para cada arquitetura real deve haver uma implementação específica dessa máquina virtual, assim o código compilado para esse sistema pode executar sobre qualquer sistema. Essa é a abordagem da máquina virtual Java e da Microsoft *Commom Language Infrastructure*, base do .Net.

4.3. Virtualização em ambientes de rede

Até agora, a virtualização tem sido apresentada principalmente como uma técnica que permite executar múltiplos sistemas operacionais e suas aplicações em máquinas virtuais sobre uma única máquina física. Entretanto, o conceito de virtualização é mais amplo. Segundo a EMA (*Enterprise Management Association*), virtualização é a técnica que "mascara" as características físicas de um recurso computacional dos sistemas, aplicações ou usuários que o utilizam [EMA, 2008]. Nesse contexto, encontramos o uso da virtualização na implementação de *desktops* remotos, de discos virtuais, na computação em *cluster* e mesmo de dados como, por exemplo, através do uso de XML, SQL, JMS, entre outros.

Em ambientes de rede, na forma como estamos acostumados a pensar, a técnica de virtualização encontra emprego na implantação de serviços Internet e na própria infra-estrutura de rede. Esses aspectos são detalhados a seguir.

4.3.1 Virtualização de serviços Internet

Os serviços Internet, em geral, foram, e ainda são, concebidos usando uma arquitetura multi-camadas (*multi-tier system architecture*). Nesse tipo de solução, um serviço é dividido em várias unidades funcionais e as mesmas são distribuídas em várias máquinas objetivando uma série de benefícios, entre outros, balanceamento de carga, tolerância a falhas, reaproveitamento de servidores e serviços, etc. Um exemplo disso são os servidores web que acessam um banco de dados. O banco de dados não precisa estar na mesma máquina que o servidor web. Havendo um servidor de banco de dados único, tanto o servidor web como um outro serviço qualquer pode usufruí-lo. Assim não se replica o serviço de banco de dados e se evita que o servidor web atue também como um servidor de banco de dados.

No entanto, essa abordagem trouxe como contra-partida a filosofia "um servidor por serviço". Rapidamente, os responsáveis pelas áreas de TI se deram conta do problema (e custo) em gerenciar diferentes máquinas físicas, mesmo que tivessem o mesmo sistema operacional. Além disso, há problemas relacionados com consumo de energia elétrica, refrigeração, espaço físico, segurança física, etc. Nesse contexto, a virtualização surge como uma possibilidade de agregar os benefícios da componetização de sofware com a redução dos custos de manutenção de hardware e software. Assim, é possível manter a idéia de um "um servidor por serviço" sem ter um hardware específico.

Essa abordagem é reforçada pela lei de Zipf [Adamic, 2008] que pode ser sintetizada da seguinte forma: a freqüência de um evento é proporcional a $x^{-\alpha}$, onde x é um *ranking* de comparação de um evento a outro. Alguns estudos [Breslau, Cao, Fan, Philips e Shenker, 2008] mostraram que a freqüência de acesso a servidores web e outros serviços Internet seguem uma distribuição Zipfian, o que, na prática, se traduz pelo fato de que a maioria dos acessos a serviços Internet é para uma minoria deles. Portanto, conclui-se que uma minoria de serviços está ativa enquanto a maioria está bloqueada a espera de requisições, o que, claramente, representa um desperdício de recursos. Para exemplificar, imagine, entre outros, o uso dos servidores de autenticação, DHCP, impressão, arquivos, e-mail, web e DNS em sua rede corporativa.

Então por que não pensar na possibilidade de pôr vários serviços em um único servidor? Apesar de possível, e muitas vezes, ser feito dessa forma, essa solução traz inconvenientes. Primeiro, não há como fazer isso se os serviços executam sobre sistemas operacionais diferentes. Segundo, quanto mais serviços se têm em um mesmo sistema, mais se aumenta à chance de um deles apresentar uma vulnerabilidade. Se essa vulnerabilidade for explorada, pode-se comprometer todo o conjunto de serviços ao invés de apenas àquele que apresenta o problema. Nesse contexto, a virtualização surge como uma solução interessante. É possível se instalar uma máquina virtual por serviço, com todas essas máquinas virtuais executando sobre o mesmo hardware físico. Assim se mantêm os serviços isolados entre si, se reduz o custo de manter computadores adicionais específicos e se evita a subutilização do hardware. Essa abordagem de pôr vários servidores virtuais em uma mesma máquina física é denominada de consolidação de servidores.

4.3.2 Virtualização da infra-estrutura de rede

Além dos serviços Internet e outros que constituem o que os usuários prcebem como sendo "a rede", há a infra-estrutura física composta por equipamentos de interconexão e sua interligação. Uma forma de enxergar esses equipamentos é vê-los como máquinas com um sistema operacional específico (*runtime*) dedicado a execução de uma tarefa como roteamento, encaminhamento de pacotes (*switches*) e *firewalls*. Inclusive, muitos fabricantes desses equipamentos usam soluções baseadas em linux embarcado. Portanto, por que não estender a idéia de máquinas virtuais a esses equipamentos?

A virtualização de equipamentos de interconexão de redes inicia no suporte que as próprias máquinas virtuais oferecem para as interfaces de rede. As interfaces de redes virtuais se comportam exatamente como interfaces reais com endereços MAC distintos, suporte a *unicast*, *multicast*, *broadcast* e VLANs. Além disso, cada máquina virtual tem seu próprio endereço IP. Portanto, uma máquina virtual se comporta, sob o ponto de vista de interconexão à rede, como um sistema real, podendo ser interligado a *switches* e roteadores como se fossem máquinas físicas distintas. Tipicamente, a placa de rede física é programada para operar em modo promíscuo e o *driver* de rede é modificado para fazer a multiplexação e demultiplexação de seu uso pelas diferentes interfaces virtuais.

É comum se encontrar referência ao par TUN/TAP. Eles nada mais são do que *drivers* de dispositivos virtuais disponíveis para os sistemas operacionais atuais como o MacOs, o Windows 2000/XP/Vista, Linux e Unixes, em geral. Normalmente, eles são aparecem no contexto de redes privativas virtuais (VPN – *Virtual Private Network*) ou em conjunto com o OpenSSH. Na realidade, TUN/TAP emulam, respectivamente, o comportamento da camada de rede e de enlace. O TAP permite a criação de *bridges* enquanto o TUN executa roteamento. Uma aplicação pode usar o *driver* TUN/TAP para enviar e receber dados. No envio, os dados são encaminhados para a pilha de protocolos de rede como se eles fossem oriundos de uma fonte externa. A recepção é similar. Assim, com o uso dos *drivers* TUN/TAP, qualquer par de aplicações podem enviar e receber dados como se eles estivem tratando com um disposito externo. Essa é a base usada para prover uma comunicação em rede virtual.

Já os equipamentos de interconexão de redes, como *switches* e roteadores, não fazem parte naturalmente das máquinas virtuais, mas podem ser emulados da mesma forma. Atualmente existem duas formas de prover essa emulação. A primeira consiste em oferecer um produto completo que disponibilize, além da máquina virtual, suporte para equipamentos de interconexão de rede virtuais. Essa é a solução adotada pela VMware (*www.vmware.com*), pela Microsoft (*www.microsoft.com*) e pela Citrix (*www.citrix.com*) em seus produtos destinados a clientes coorporativos. A segunda é prover máquinas virtuais dedicadas a essa finalidade, como é, por exemplo, o caso da Vyatta (*www.vyatta.com*) que oferece um produto compatível com os *hypervisors* mais comuns no mercado. A solução da Vyatta agrega uma camada de software suplementar que simula equipamentos de interconexão de redes. Na realidade, há uma terceira possibilidade que é oferecer hardware específico com suporte a virtualização de equipamentos de interconexão, como é o caso de alguns produtos da linha Catalyst da Cisco (*www.cisco.com*).

Portanto, é possível construir uma infra-estrutura de rede completa totalmente virtualizada. A arquitetura exata e as opções de projeto, como em uma rede real, dependem, caso a caso, em função de requisitos da organização. O importante a salientar é que existe tecnologia para tal.

4.4. Ferramentas de virtualização

A virtualização tem se tornado a grande revolução da área de TI nesses últimos anos, basta ver o crescimento do volume de investimento das empresas nesse sentido e o crescimento das empresas que oferecem soluções de virtualização. Atualmente, existem disponíveis várias soluções de virtualização. Para ter uma idéia, consulte a *wikipedia* ou o *google* fazendo uma busca por "*virtualizaition*". Basicamente, existem soluções comerciais, gratuitas, em software livre, integradas a sistemas operacionais, etc. Seria inviável, e fora do escopo deste trabalho, tecer comentários sobre todas elas, por isso optou-se por apresentar aquelas que estão atualmente dominando o mercado da virtualização: VMware e o Xen. Além delas, há a resposta da Microsoft ao movimento mundial da virtualização, que dado o parque de máquinas instaladas com esse sistema também foi escolhida para uma discussão mais detalhada.

4.4.1 VMware

O VMware [VMware, 2008c] é na realidade uma infra-estrutura de virtualização completa com produtos abrangendo desde *desktops* a *data centers* organizados em três categorias: gestão e automatização, infra-estrutura virtual e virtualização de plataformas. Cada categoria possui um conjunto de produtos específicos. Os produtos de gestão e automatização têm por objetivo principal, como seu próprio nome induz, a permitir de uma forma automatizada e centralizada a gerência de todos os recursos da infra-estrutura virtual permitindo a monitoração do sistema, auxiliando na conversão de sistemas físicos em virtuais, na recuperação de desastres, entre outros.

Os produtos de infra-estrutura virtual auxiliam a monitoração e alocação de recursos entre as máquinas virtuais de forma a atender requisitos e regras de negócios. Eles fornecem soluções para alta-disponibilidade, *backup*, migração de máquinas virtuais e atualização de versões de softwares.

Por fim, os produtos de virtualização de plataformas, ou seja, aqueles destinados a criar máquinas virtuais como apresentado na seção 4.2.1. Essa categoria é composta por oito produtos:

- VMware ESX *Server* 3: é a base para a criação de *datacenters* virtuais. O ESX *server* é um *hypervisor* que virtualiza os recursos de hardware do tipo processador, memória, armazenamento e rede. Dessa forma, o ESX *Server* permite que um servidor físico seja particionado em várias máquinas virtuais e que cada uma seja vista como uma máquina física em uma infra-estrutura de rede convencional.
- VMware ESX *Server* 3i: possui as mesmas características e funcionalidades descritas anteriormente para o ESX *Server* 3. A diferença consiste na sua arquitetura interna e na forma como alguns procedimentos de gerenciamento são executados.

- VMware Virtual SMP: permite que uma única máquina virtual empregue mais de um processador físico simultaneamente. Na atual versão, uma máquina virtual pode empregar até quatro processadores físicos.
- VMware VMFS: é um sistema de arquivos que permite que várias máquinas virtuais acessem concorrentemente, para leitura e escrita, um mesmo meio de armazenamento. Além disso, oferece uma série de facilidades como adicionar e remover dinamicamente ESX *servers* da estrutura de um sistema de arquivos, adaptar tamanho de bloco para volume e para arquivos e recuperação de falhas.
- VMware *Server*: é a versão gratuita dos produtos ESX *Server*. Seu objetivo essencial é permitir que usuários testem o produto antes de adquiri-lo. Assim como as versões ESX, o VMware *Server* oferece a virtualização de processador, memória, armazenamento e infra-estrutura de rede que são configurados através de ferramenta própria não disponível na versão gratuita. Para contornar o problema de configuração, ou seja, a criação do ambiente hóspede, a VMware oferece uma série de imagens de ambientes predefinidas em seu site. Essas imagens são denominadas de *appliances* e contemplam os serviços de rede mais comuns (web, arquivos, impressão, DNS, etc).
- VMware *Workstation*: é o ambiente que permite a criação de máquinas virtuais sobre o *hypervisor*. Isso significa que é possível carregar um sistema operacional qualquer nessa máquina virtual e executar as suas aplicações. A configuração das máquinas virtuais para um determinado sistema operacional é feita através de ferramenta específica que é parte integrante desse produto.
- VMware Fusion: é a solução VMware *Workstation* equivalente para o sistema operacional MacOS X.
- VMware *Player*: é a versão gratuita do produto VMware *Workstation*. Assim como ocorria na versão *server*, o objetivo é permitir que usuários testem o uso da virtualização e não é possível definir (criar) o sistema hóspede na máquina virtual a partir do zero. Novamente, em seu site, a VMware distribui uma série de *appliances* (imagens de sistemas hóspedes) que contemplam diferentes distribuições de linux e Windows *Server* 2003.

A figura 4.7 mostra de forma esquemática a aplicação desses produtos na definição de uma infra-estrutura virtual completa (máquinas e rede). Os dois componentes essenciais são o produto ESX *server* que oferece a camada necessária para a virtualização dos recursos de hardware (*hypervisor*) e as máquinas virtuais definidas sobre ele através do produto VMware *workstation*. Cabe ressaltar que a versão ESX *Server* oferece *switches* virtuais (com suporte a VLANs) para interligar as máquinas virtuais.

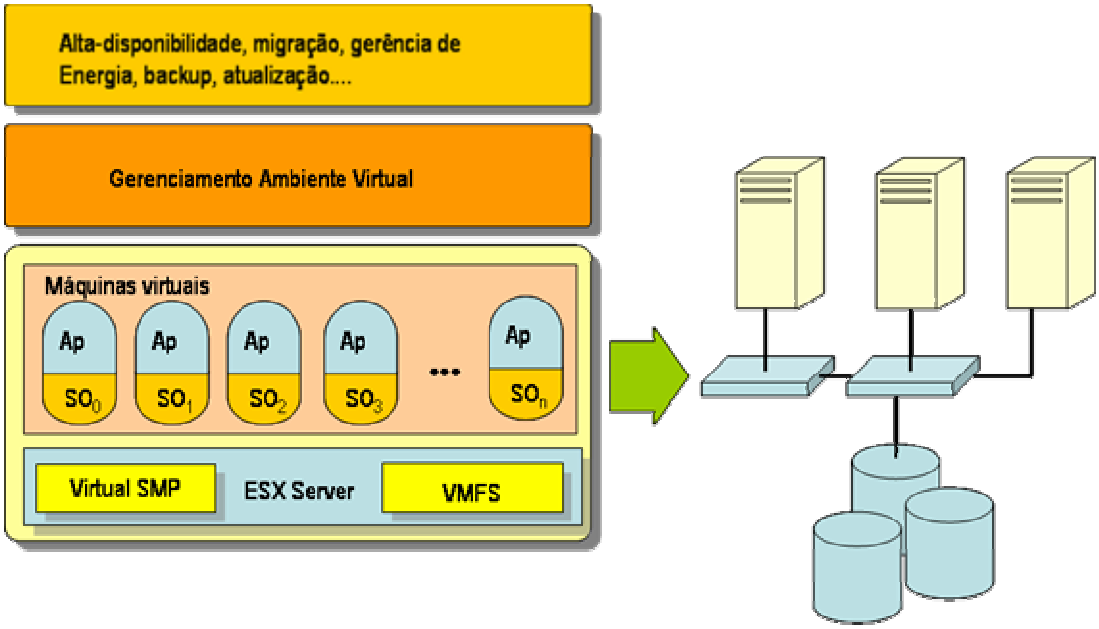

Figura 4.7 – infra-estrutura virtual VMware

Para concluir a apresentação do VMware, alguns comentários finais sobre o funcionamento do VMware *Player* e do VMware *Workstation* por serem bastante populares para as plataformas Intel 32-bits. A abordagem escolhida pelo VMware é de executar como uma aplicação do sistema hospedeiro, ou seja, funcionam também como máquina virtual de processo. Dessa forma, o suporte para os diferentes dispositivos de entrada e saída é fornecido pelo próprio sistema operacional hóspede. Essa solução é denominada de *Hosted Virtual Machine Architecture* (HVMA). Na realidade, por questões de desempenho, o VMware não executa completamente em espaço de usuário, pois é instalado um *driver* de dispositivo específico, (*VMDriver*) que permite que as máquinas virtuais acessem os *drivers* de dispositivo do sistema hóspede. O VMDriver põe a placa de rede em modo promíscuo e cria uma *bridge* ethernet virtual que recebe todos os quadros ethernet e os reencaminha para o sistema hóspede ou para a máquina virtual especificada. Essa implementação também oferece NAT (*Network Address Translation*), de tal forma que cada interface virtual tenha o seu próprio endereço IP.

4.4.2 Xen

O Xen [XenSource, 2008b] é um monitor de máquina virtual (*hypervisor* ou VMM), em software livre, licenciado nos termos da GNU General Public Licence (GPL), para arquiteturas x86, que permite vários sistemas operacionais hóspedes serem executados em um mesmo sistema hospedeiro. O Xen é originário de um projeto de pesquisa da universidade de Cambridge, que resultou em um empresa, a XenSource inc, adquirida pela Citrix System em outubro 2007.

O Xen implementa a virtualização de uma forma um pouco diferente daquela apresentada na seção 4.2. Os dois principais conceitos do Xen são domínios e *hypervisor*. Os domínios são as máquinas virtuais do Xen e são de dois tipos: privilegiada (domínio 0) ou não-privilegiada (domínio U). O *hypervisor* tem por função

controlar os recursos de comunicação, de memória e de processamento das máquinas virtuais, e não possui *drivers* de dispositivos. O *hypervisor* Xen, considerando suas características, não é capaz de suportar nenhum tipo de interação com sistemas hóspedes. Por isso, é necessário que exista um sistema inicial para ser invocado pelo *hypervisor*. Esse sistema inicial é o domínio 0. As outras máquinas virtuais só podem ser executadas depois que ele for iniciado. As máquinas virtuais de domínio U são criadas, iniciadas e terminadas através do domínio 0.

O domínio 0 é uma máquina virtual única que executa um núcleo linux modificado e que possuí privilégios especiais para acessar os recursos físicos de entrada e saída e interagir com as demais máquinas virtuais (domínios U). O domínio 0, por ser um sistema operacional modificado, possui os *drivers* de dispositivos da máquina física e dois *drivers* especiais para tratar as requisições de acesso a rede e ao disco efetuados pelas máquinas virtuais dos domínios U. A figura 4.8 mostra o relacionamento entre o *hypervisor*, o domínio 0 e as demais máquinas virtuais.

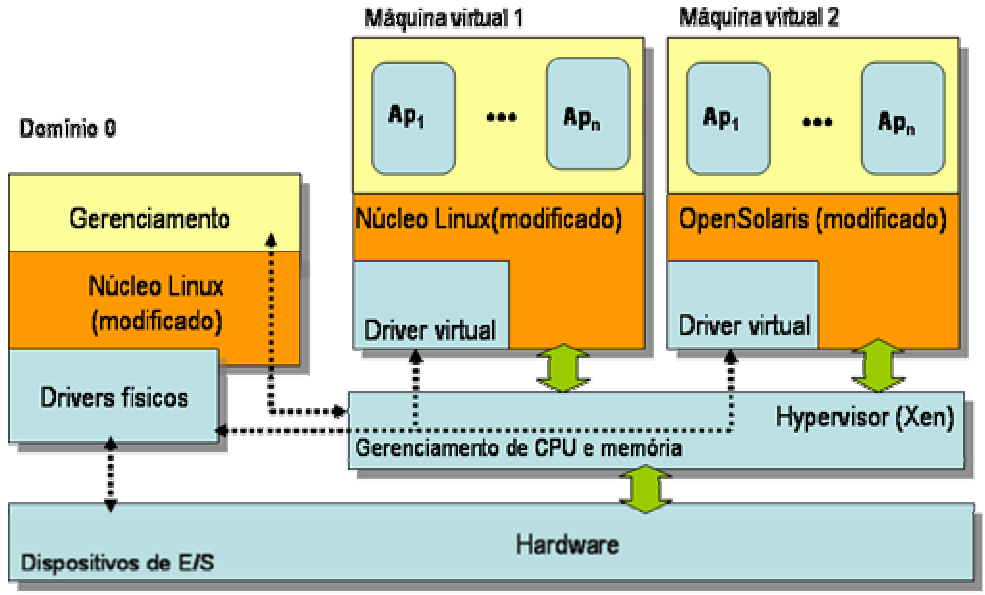

Figura 4.8 – Componentes do Xen: *hypervisor* e domínios

A primeira versão do Xen é de outubro de 2003 e, originalmente, o *hypervisor* foi implementado usando a técnica da para-virtualização, ou seja, era necessário modificar os sistemas operacionais hóspedes (domínios U) para torná-los conscientes do *hypervisor*. Essa decisão se justificava por questões de desempenho, mas limitou o emprego do Xen aos sistemas Unix, principalmente aqueles com filosofia de código aberto. A partir da versão 3, o Xen passou a oferecer virtualização completa, ou seja, permite o uso de sistemas operacionais não modificados, como os da família Microsoft Windows. Entretanto, isso só é possível se o processador oferecer suporte de hardware (Intel VT ou AMD-V).

Para oferecer suporte tanto para a para-virtualização como para a virtualização completa, o Xen distingue os domínios U entre para-virtualizados (domínios U-PV) e virtualizados (domínios U-HVM, de *Hosted Virtual Machines*). Os domínios U-PV têm consciência de que não tem acesso direto ao hardware e reconhecem a existência de outras máquinas virtuais. Os domínios U-HVM não têm essa consciência, nem reconhecem a existência de outras máquinas virtuais. Na prática, isso se traduz no fato de que os domínios U-PV possuem *drivers* específicos para acesso a rede e a disco para interagirem com as suas contra-partidas no domínio 0. Já as máquinas dos domínios U-HVM não possuem esses *drivers* (não foram modificados) e iniciam como um sistema convencional procurando executar a BIOS. O *Xen virtual firmware* simula a existência da BIOS executando todos os procedimentos esperados durante o *boot* normal de um ambiente PC compatível. O compartilhamento do disco e as requisições de rede de um domínio U-HVM são feitos através de um *daemon* Qemu vinculado a cada instância U-HMV (O QEMU é um emulador em software de código livre). O hardware disponível para as máquinas virtuais do domínio U-HVM são aquelas oferecidas pelo QEMU.

4.4.3 Virtualização e Microsoft

Atenta ao movimento da virtualização, a Microsoft oferece uma gama de produtos para esse tipo de tecnologia. Esses produtos exploram o conceito da virtualização, na sua forma mais ampla, para oferecer soluções que sejam integradas e apropriadas à infra-estrutura de TI que se encontra hoje em dia. Basicamente:

- Virtualização de aplicações: também denominada de *SoftGrid*, cujo objetivo é fornecer aplicações por demanda. Isso implica que se um determinado *desktop* necessita executar uma aplicação e a mesma não está disponível nele, o sistema executará automaticamente a busca, instalação e configuração da aplicação.
- Virtualização de apresentação: essa ferramenta separa e isola as tarefas de tratamento gráfico (visualização) e de E/S, permitindo que uma determinada aplicação seja executada em uma máquina, mas utilize recursos gráficos e de E/S de outra.
- Gerenciamento da virtualização: *System Center Virtual Machine Manager* é um ambiente de gerenciamento que facilita as tarefas de configuração e de monitoração de um ambiente virtual. É através dele que se realiza a administração das contas de usuários e seus privilégios.
- Virtualização de *desktops* (Virtual PC): permite a criação de máquinas virtuais em um sistema hospedeiro Microsoft Windows, cada uma com seu próprio sistema operacional. Basicamente destina-se àquelas aplicações onde é necessário executar software legado, criar ambientes de testes, treinamento, etc.
- Virtualização de servidores: é a solução que permite criar máquinas virtuais em servidores. Nessas máquinas, questões ligadas à segurança, tolerância a falhas, confiabilidade, disponibilidade se tornam importantes. Portanto, a solução de virtualização de servidores, denominada de Hyper-V (ou *viridian*) foi projetada para endereçar esses requisitos.

Desses produtos, dois têm especial interesse para o escopo deste trabalho, o virtual PC 2007 e o Windows *Server* 2008 Hyper-V, que serão comentados a seguir.

O Virtual PC 2007 é uma máquina virtual para família Windows que pode ser configurada para executar qualquer outro sistema operacional. Segundo a Microsoft, o principal objetivo do Virtual PC é o desenvolvimento e teste de software para múltiplas plataformas. Dentro desse princípio, o Virtual PC oferece mecanismos para interconectar logicamente as diferentes máquinas virtuais. Cada máquina virtual tem seu próprio endereço MAC e endereço IP. Além disso, o Virtual PC oferece um servidor de DHCP, um servidor NAT e *switches* virtuais. Dessa forma, é possível construir cenários de rede usando máquinas virtuais. O virtual PC 2007 é disponível para *download*, assim como um *white paper* que ensina a configurar as máquinas virtuais e um ambiente de rede. Um ponto interessante a comentar em relação à gratuidade do Virtual PC é que, na FAQ do Virtual PC, a Microsoft alega que o que tem valor agregado não é a máquina virtual em si, mas sim os ambientes de gerenciamento.

A Microsoft já vinha atuando no segmento de virtualização de servidores com o Microsoft Virtual *Server* 2005. A proposta do Windows 2008 *Server* Hyper-V é ser a evolução desse produto respondendo a novas demandas e explorando eficientemente as arquiteturas de 64 bits, processadores *multicore* e meios de armazenamento.O Windows 2008 é o componente-chave da estratégia da Microsoft para atuar no segmento da virtualização, no que ela denomina de *datacenter-to-destktop virtualization strategy*. Dentro dessa estratégia a Microsoft oferece todo um ambiente integrado de gerenciamento da virtualização (monitoração, automatização de procedimentos, migração, recuperação de desastres etc).

Entre as principais vantagens do Windows 2008 Server Hyper-V estão várias ferramentas para automatizar o processo de virtualização. Uma delas é o *Manager Physical-to-virtual* (P2V) que auxilia na conversão de servidores físicos para virtuais. Há também o Volume *Shadow Copy Services* que realiza automaticamente procedimentos de *backup* e de disponibilidade de forma que o sistema, como um todo, opere de forma homogênea independente de falhas e de "picos" de carga. Isso é feito por técnicas de migração de máquinas virtuais. Um ponto que recebeu especial atenção foi a segurança. Para isso, o Hyper-V usa mecanismos em hardware existentes nos atuais processadores, como o "*execute-disable-bit*" que reduz o sucesso de ataque de vários tipos de vírus e vermes. O Hyper-V reforça o aspecto de segurança através de um restrito controle de regras de permissões integrado com o *Active Directory* e com políticas de grupo. Além disso, o Hyper-V permite que as máquinas virtuais usem, sem restrição alguma, as técnicas e ferramentas de segurança tradicionalmente empregadas nas máquinas físicas (*firewalls*, quarentena, anti-virus, entre outros).

4.5. Casos de uso

A virtualização traz benefícios em uma série de situações. Seu emprego clássico é na consolidação de servidores, ou seja, executar vários serviços em uma mesma máquina física, mas cada um em uma máquina virtual separada. Além de permitir que o hardware físico seja melhor aproveitado, isso reduz investimento na aquisição e na manutenção de uma infra-estrutura física de rede (refrigeração, cabeamento, espaço físico, consumo de energia etc). Uma vantagem importante é o fato de que uma máquina virtual pode ser facilmente migrada de uma máquina física a outra, o que possibilita uma rápida recuperação em casos de falhas e agiliza a manutenção de parques de máquinas. Ainda, como cada máquina virtual é um sistema totalmente isolado dos demais, a falha, ou comprometimento, de um serviço não afeta os demais.

O desenvolvimento de software é uma outra situação em que a virtualização traz vantagens. É possível testar um produto em desenvolvimento sobre vários sistemas operacionais, e suas distintas versões, sem a necessidade de ter uma máquina física instalada com cada sistema operacional. Outro uso interessante da virtualização é a possibilidade de construir ambientes de testes, onde se pode instalar, configurar e avaliar pacotes de software antes de pô-lo em produção, isso sem comprometer um sistema computacional físico. Essa mesma característica permite a construção de *honeypots*. Nesse caso, é possível criar um ambiente virtual completo, com vários servidores, cada um com um sistema operacional distinto e/ou serviços, e assim monitorar atividades de tentativas de invasão sem, novamente, afetar um sistema físico.

Na prática, existem muitas aplicações onde a virtualização pode ser interessante, mas, tipicamente, elas se encaixam nas situações descritas anteriormente. Nesta seção optou-se por elaborar um pouco mais essas três situações: virtualização de servidores, virtualização de *desktops* e uso de *honeypots*.

4.5.1 Virtualização de servidores

De acordo com uma pesquisa realizada pela Network World (*www.networkworld.com*), a virtualização de servidores é motivada, em ordem de importância, pelas seguintes razões : aumentar a taxa de utilização de servidores, reduzir os custos operacionais de *datacenters*, melhorar os procedimentos de recuperação de desastres e de *backup*, criar ambientes mais flexíveis para desenvolvimento e teste de software e reduzir custos de administração de TI. O princípio básico é o melhor aproveitamento de recursos: invés de haver n servidores com percentual de utilização de x é possível ter um único servidor com um percentual de uso de aproximadamente $n.x$ ($n.x < 100$).

Atualmente, uma organização corporativa, por menor que seja, apresenta uma série de serviços Internet considerados essenciais, como servidores web, e-mail, etc. Muitas dessas organizações, por questões que variam desde investimentos à segurança, optam por hospedar seus servidores em *data centers*. Porém, isso não inválida dois pontos importantes. Primeiro, o problema de manter servidores, que passa a ser dos *data centers*. Segundo, apesar de "terceirizar" alguns serviços Internet, é comum manter na rede local da organização alguns serviços como DHCP, impressão, autenticação,

arquivos e aqueles de aplicações relacionados com o negócio da empresa (softwares de ERP, por exemplo).

Inicialmente, vamos discutir o caso da pequena e média organização e seus servidores. Nesse contexto, é comum os servidores serem máquinas ultrapassadas e "intocáveis". Afinal, se os serviços executados não excedem a capacidade dessas máquinas, por que investir em equipamentos novos? Além disso, atualizar máquinas implica em mexer em sistemas que, bem ou mal, estão funcionando. Entretanto, essa situação traz um risco inerente: um hardware está sujeito a falhas, principalmente discos, e, muitas vezes, para equipamentos ultrapassados, os contratos de manutenção não são mais válidos e as peças de reposição simplesmente inexistem. É uma questão de tempo ter um problema (grave) de indisponibilidade de serviços.

Para esses casos, uma solução possível é adquirir apenas um único equipamento novo e instalar nele tantas máquinas virtuais quanto serviços houver. É claro que se deve dimensionar a capacidade total de processamento desse novo servidor em relação à carga de serviços que ele receberá. A vantagem é trocar um parque de máquinas ultrapassadas e, eventualmente, subutilizadas, por uma máquina mais nova e bem utilizada. Os serviços que executam nas máquinas mais antigas podem ser migrados, um a um, para uma máquina virtual na nova máquina, facilitando o processo de transição. Dependendo da solução de virtualização escolhida, há ferramentas que auxiliam nessa migração. A notar, ainda, que cada máquina virtual tem seu próprio sistema operacional, portanto, é possíveis pôr no novo equipamento os serviços, ou aplicações, que executam em diferentes sistemas operacionais como, por exemplo, Microsoft Windows e Linux. Futuramente, se for necessário uma nova atualização de hardware, basta parar as máquinas virtuais, fazer uma imagem delas e as inicializar novamente, sem necessitar reinstalar e reconfigurar softwares específicos.

Para o caso de *data center*, as vantagens discutidas anteriormente tomam uma envergadura maior. Em um *data center*, pela sua própria natureza, é comum haver diferentes sistemas operacionais e, eventualmente, mais de uma versão de um mesmo sistema operacional devido a necessidades especificas de produtos que seus clientes executam. Sem virtualização, cada sistema operacional (e versão) precisaria de um hardware dedicado. Isso representa custo de instalação, manutenção, refrigeração, energia e suporte técnico para administrá-los. Já com a virtualização, um único hardware, dimensionado adequadamente, pode manter vários sistemas operacionais de forma mais econômica.

Os procedimentos de migração e de tolerância a falhas são importantes em *data center*. A migração pode ser usada no momento de atualizar parques de máquinas movendo um serviço (cliente) para outro enquanto a troca é feita. A migração ainda é útil como mecanismo de balanceamento de carga, movendo serviços de uma máquina a outra até homogeneizar a carga do sistema. Isso evita que certos sistemas fiquem sobrecarregados enquanto outros estão ociosos. Novamente, a noção de imagem de uma máquina virtual auxilia nessas tarefas.

Um ponto a ressaltar é que no caso de *data center*, a virtualizaçao não é a única alternativa. Muitos fabricantes oferecem linhas de servidores de alto-desempenho que

oferecem vantagens similares às da virtualização. Por exemplo, existe disponível no mercado máquinas NUMA de até 72 processadores que podem ser particionados via hardware em até 18 domínios de 4 processadores. Assim, uma única máquina física pode ser configurada para operar como se fosse 18 máquinas físicas distintas, cada uma com seu próprio sistema operacional. Há também a opção de servidores *blade*. Nesses casos, a queda de um nó (*blade*) pode ser sanada com a instalação física de outro *blade*. Os sistemas de gerenciamento de servidores *blade* oferecem software de gerenciamento para facilitar o *backup*, instalação e reconfiguração de *blades*. Decidir qual solução é função do gerente de TI ponderando vantagens e desvantagens em função do investimento e retorno esperados.

4.5.2 Virtualização de *desktops*

A consolidação de servidores é o uso clássico de virtualização, mas ela também traz uma série de benefícios quando empregada em *desktops*. Inicialmente, a virtualização oferece uma forma simples para testar novas configurações ou executar programas que foram feitos para sistemas operacionais diferentes do nativo. Dessa forma, um usuário que deseja testar um software, ou abrir um programa, que não foi desenvolvido para seu sistema operacional pode lançar mão desse recurso. Há uma série de ferramentas para o uso de virtualização em *desktops*, onde entre elas, se destacam os produtos da VMmare e o Virtual PC por serem de fácil instalação e uso.

Talvez a situação recém descrita não pareça um caso comum, mas há um tipo de usuário que pode se beneficiar de executar programas hóspedes em diferentes sistemas operacionais hospedeiros: projetistas de software. As máquinas virtuais permitem isso de maneira bastante simples, o que é interessante em fases de desenvolvimento e depuração de software. Um outro ponto, vinculado a depuração de sistemas, é a facilidade que as máquinas virtuais têm de quando finalizadas registrar uma espécie de instantâneo (*snapshot*) de seu estado. Isso permite que uma análise seja parada e retomada mais tarde a partir do mesmo ponto.

Em organizações de um determinado porte é comum o tempo de processamento dos *desktops* ser subutilizado. Por exemplo, o horário comercial é de cerca de 9 horas diárias, ou seja, 45 horas por semana, mas uma semana tem um total de 168 horas (24 x 7), portanto, a taxa de utilização dos equipamentos é de aproximadamente 27%. Uma forma de recuperar esse processamento é associar ao descanso de tela uma máquina virtual, dessa forma, quando o descanso de tela é ativado ele pode registrar uma máquina virtual em um servidor central e assim formar um cluster (agregado) de computadores virtuais. Esse uso é derivado de programas *peer-to-peer*, como *seti@home*. Existem vários projetos desenvolvidos nesse sentido. Normalmente, a técnica emprega um horário a partir do qual o registro é feito e validado, por exemplo, às 20 horas, quando se espera não haver mais nenhum colaborador na organização, e, no início da manhã (6 horas) é feito um salvamento do estado da computação (*checkpoint*) da máquina virtual e, em seguida, a máquina é liberada para seu uso convencional.

Outra utilização interessante da virtualização é em laboratórios de treinamento e de ensino como aqueles encontrados em universidades. Para algumas disciplinas é

necessário ter ambientes com diferentes sistemas operacionais. Apesar de ser possível, e comum, se configurar um PC compatível com múltiplos *boots*, essa solução nem sempre é a mais adequada. Primeiro, dependendo dos sistemas operacionais, questões práticas afetam a ordem de instalação e de configuração dos diversos sistemas. Segundo, e talvez mais importante, é comum haver cursos e treinamentos onde o participante necessita ter acessos administrativos. Nesses casos, mesmo mudando as senhas de administrador (*root*) a cada sessão, há a possibilidade de se corromper o sistema operacional e instalar *rootkits, backdoors, sniffers* de teclado, e todo um conjunto de *malwares*, além de ser possível, por exemplo, montar a partição dedicada a um outro sistema operacional e provocar danos. As máquinas virtuais de processo são as mais indicadas para essas aplicações. O participante de uma disciplina pode usar um ambiente virtual para fazer todos os experimentos e, por rodar de forma confinada em um processo, suas ações como administrador não afetam o sistema nativo. Ao finalizar o processo (máquina virtual) não há nenhum vestígio permanente do que foi feito. Um detalhe, havendo necessidade de preservar arquivos de uma aula a outra, as máquinas virtuais oferecem formas de comunicação (*ftp, telnet, scp* etc) e armazenamento (*usb*) que possibilitam a cópia e o salvamento de arquivos.

Um comentário para concluir esta seção. É importante lembrar que as máquinas virtuais são imagens que são facilmente instaladas, o que permite restaurar uma máquina para um sistema padrão de forma bastante simples e rápida. Para se ter uma ordem de grandeza, nos atuais PC, o tempo de inicialização de uma máquina virtual VMware, com um sistema hóspede Linux, em ambientes hospedeiros Microsoft Windows, é de cerca de 70 segundos.

4.5.3 *Honeypots*

De uma forma simplista, *honeypot* consiste em se colocar intencionalmente máquinas na Internet de forma que elas sejam atacadas por *crackers*. O intuito é monitorar as atividades desses, se precaver de ataques e tornar mais fácil a investigação de incidências de ataques e sua recuperação. O problema com *honeypots* é que dependendo do tipo de ação sofrida pode haver o comprometimento da máquina (sistema operacional). Nesses casos, a solução passa por reinstalar o sistema. Outra característica do uso de *honeypots* é que eles normalmente são compostos por máquinas destinadas a essa finalidade que são postas em segmentos de redes específicos, o que aumenta os investimentos tanto em nível de hardware como de suporte de TI.

A virtualização surge como uma opção interessante para se implementar *honeypots* por várias razões. Primeira, o comprometimento de um sistema operacional é resolvido apenas com a remoção da máquina virtual e a instanciação de uma nova. Segunda, em um mesmo hardware físico é possível se instalar várias máquinas virtuais, cada uma com um sistema operacional diferente, o que permite aumentar o número de máquinas "iscas". Por fim, com o uso de softwares de emulação de equipamentos de rede se pode definir uma infra-estrutura de rede virtual, com *firewalls*, *proxies*, roteadores etc, em uma única máquina. Note que isso não é imprescindível, pois as ferramentas de virtualização oferecem normalmente suporte a interfaces de redes

virtuais, com endereços MAC e IP distintos, e a VLANs, o que permite interconectar as máquinas virtuais em equipamentos de interconexão físicos sem maiores problemas.

Uma outra situação, associada com a idéia de *honeypot*, é a possibilidade de se executar softwares de origens "não confiáveis" em um ambiente onde os prejuízos são minimizados. Isso é interessante para abrir arquivos *attachment* suspeitos ou verificar *malwares*.

Vale a pena salientar que as ferramentas tradicionais existentes para *honeypots* podem ser utilizadas sem restrições alguma em ambientes virtualizados, pois elas são aplicações para um sistema operacional. Uma boa referência para ferramentas de *honeypot* é *www.honeyclient.org*.

4.6. Planejando soluções com virtualização

Na seção anterior foram apresentadas situações onde a virtualização pode ser empregada para trazer benefícios, porém o sucesso de sua implantação depende de uma análise mais criteriosa quanto ao seu emprego. É importante ter consciência que a virtualização é uma decisão sobre a arquitetura da infra-estrutura de TI e não apenas mais um projeto de implantação de um novo sistema operacional ou de um aplicativo. Portanto, a adoção da virtualização deve ser vista como um projeto de longo termo e não como modismo passageiro, pois, para obter o melhor resultado, é necessário avaliar e investir em uma boa solução e na capacitação da equipe de TI. Além disso, é importante salientar que com o tipo de mudança que a virtualização traz é possível se encontrar resistências culturais e desconfianças quanto a sua implantação. O objetivo desta seção é levantar uma série de aspectos para auxiliar e orientar na implantação de soluções baseadas em virtualização.

Inicialmente, não existe um manual de "*Como implementar virtualização em minha empresa*", mas muitos procedimentos gerenciais usuais podem ser aplicados com sucesso. Uma analogia interessante é a da segurança da informação à luz da norma ISO 17799. Os vários livros, manuais e artigos que tratam da segurança da informação fazem referência a dois pontos fundamentais: apoio e comprometimento dos níveis gerenciais e "pensar" a solução como um todo. Depois vêm, mas não menos importantes, uma série de aspectos como identificação de ativos, proprietários de dados, análise de riscos etc. Todos esses pontos são válidos para a virtualização.

A virtualização é um projeto que implica em uma mudança de paradigma, e como tal, é importante contar com o apoio de níveis hierárquicos superiores. Seu emprego modificará a forma pela qual os equipamentos serão comprados e como novos sistemas serão instalados. Uma recomendação comum é planejar a virtualização como um todo, mas implantá-la passo a passo. Isso permite que as pessoas adquiram confiança na solução e possibilita que se tenha, em curto prazo (cerca de seis meses), um ROI (*Returns on Investiment*) mensurável. Uma forma de fazer isso é iniciar consolidando servidores que não sejam fundamentais para o negócio. O fato de demonstrar que servidores como, por exemplo, DHCP, DNS, impressão, executam em uma única máquina física, sem comprometer desempenho e segurança, e que, eventualmente, cada um executa um sistema operacional diferente é algo que se torna

um cartão de visita para a solução. Esse "cartão de visitas" pode auxiliar na conquista dos níveis hierárquicos superiores.

Mas, ao menos existe um *road map* para a implantação da virtualização? A resposta é sim e, novamente, nada mais é do que uma adaptação de técnicas já conhecidas. É possível identificar quatro fases: avaliação do sistema; planejamento e projeto; execução; e, gerenciamento e manutenção.

A fase de avaliação do sistema consiste em um inventário de todos os sistemas, hardware e software, existentes na infra-estrutura a ser virtualizada. O objetivo é descobrir todos os serviços, os equipamentos e sistemas operacionais empregadosno dia-a-dia. É importante identificar a percepção que as pessoas têm da infra-estrutura de rede, a carga de trabalho dos servidores e o desempenho de cada um deles.

A visão que os usuários têm da rede não deve ser subestimada, pois são eles os clientes finais, os "proprietários" dos dados e os que executam tarefas diárias usando a infra-estrutura de rede. A melhor forma de se obter essa visão é através de entrevistas com os responsáveis das diversas áreas. O objetivo é entender o fluxo de informação e procedimentos feitos quotidianamente e como a virtualização pode afetá-los. A seguir, é preciso levantar a taxa de utilização dos recursos físicos como, processador, memória RAM, armazenamento e tráfego de rede. Isso permitirá dimensionar equipamentos e dividir adequadamente serviços de forma a utilizar eficazmente os recursos e a manter um equilíbrio de carga.

A etapa de planejamento e projeto consiste em mapear a infra-estrutura física em uma virtual com base nas informações levantados na avaliação do sistema. Fazem parte do planejamento e projeto identificar qual a melhor tecnologia de virtualização e o produto a ser empregado para atingir os objetivos propostos. Entre outros, são critérios a serem adotados: solução em software livre ou proprietário? Qual a sua robustez? Avaliação de continuidade da empresa ou da comunidade que desenvolve a desenvolve? Suporte técnico? Atualização? Curva de aprendizado? Ferramentas de gerenciamento? Na seqüência, é necessário dividir os serviços em máquinas considerando quanto cada um deles consome de recursos de processamento e de rede. É importante definir planos de contingência e a sistematização de procedimentos como configurações padrões, *backups* de dados e das máquinas virtuais, registro de incidentes de segurança, atualizações dos sistemas operacionais hóspedes e hospedeiros, assim como da própria solução de virtualização. Como toda etapa de planejamento, aqui deve ser estabelecido um cronograma físico de tarefas.

A terceira etapa consiste na execução. A recomendação óbvia é implementar um serviço por vez iniciando pelos menos sensíveis. Várias das soluções de virtualização possuem assistentes (*wizards*) denominadas de P2V (*physical to Virtual*) para auxiliar nessa etapa. Para cada máquina virtual (serviço) instalado é interessante fazer uma avaliação de seu comportamento e desempenho comparando-o com aquele apresentando na máquina física. Os dados obtidos nessa nova avaliação podem, eventualmente, alterar o projeto inicial. Esse ciclo de refinamento é normal.

Por fim, a última etapa, gerenciamento e manutenção. Uma vez implantada a infra-estrutura virtual é necessário garantir seu funcionamento de forma adequada. Para tal, o sistema como um todo deve ser monitorado, atualizado, sofrer procedimentos de salvamento e restauração de cópias de segurança. O funcionamento em produção do sistema, sem dúvida, trará lições importantes que realimentarão atividades futuras como planejamento de novas capacidades, planos de investimento, treinamento, entre outros.

De forma mais pragmática, que tipos de aspectos e recomendações gerais se deve cuidar quando da implantação da virtualização? Bom, cada caso é um caso, mas há alguns pontos comuns que podem ser comentados:

- Ao consolidar os servidores a tendência é deixá-los em uma mesma rede já que estão em uma mesma máquina física. Isso é considerado um erro. Mesmo que as máquinas virtuais estejam em uma mesma máquina física é recomendável manter a segmentação da rede criando diferentes redes de perímetro. O próprio *hypervisor* pode ser um problema para a segurança e convém lembrar disso ao colocar fisicamente a máquina na rede.
- A consolidação de servidores não é a única forma de virtualização. Os *desktops* podem usufruir dessa tecnologia para desenvolvimento e teste de software, instalação de laboratórios de treinamentos, teste de software de terceiros, definição de clusters virtuais, etc. Além disso, existem produtos de interconexão de rede que oferecem *switches* virtuais, ou seja, a rede em si pode usufruir da virtualização.
- Existem gamas de servidores que foram projetados visando um alto desempenho de processamento, com um armazenamento de grande quantidade de dados e com acesso a dados em alta vazão. Por exemplo, há várias opções de servidores empregando tecnologia *fiber channel* interconectar processadores com módulos de E/S. Lembrar que virtualizar nem sempre é a melhor (ou única) solução.
- Adquirir novos equipamentos tendo em vista a virtualização. Isso implica em verificar o uso de tecnologias de processadores como IVT (Intel) ou AMD-V (AMD) para obter um melhor desempenho e segurança dos *hypervisors*.
- Se a infra-estrutura de rede também for virtualizada, não esquecer de incluir sua arquitetura e seus parâmetros de configuração no *backup*.
- Estabelecer planos de contingência para o caso de recuperação de desastres. Isso inclui listar os vários sistemas operacionais envolvidos e as aplicações fundamentais para o negócio. Não negligenciar procedimentos de *backups* e testes regulares de seu conteúdo. Afinal um m *backup* mal feito, ou corrompido, é pior do que não ter, pois passa a falsa noção de segurança.

Por fim, um ponto muito importante. Como a virtualização permite que em uma mesma máquina física executem diferentes sistemas operacionais é necessário que equipes que estão "evangelizadas" em diferentes sistemas operacionais sejam integradas. Como sempre, o fator humano não deve ser negligenciado sob pena de se comprometer à solução como um todo. É importante investir em formação, integração e "quebrar" preconceitos que existem entre defensores de um e outro sistema operacional.

4.7. Sistemas virtuais, problemas reais

Até o presente momento a virtualização foi colocada como uma panacéia para vários problemas. Entretanto, existem aspectos a serem avaliados antes de se optar pela virtualização. Basicamente, são três pontos a serem melhor elaborados: segurança, gerenciamento e desempenho.

Inicialmente, o *hypervisor* é uma camada de software e, como tal, está sujeito a vulnerabilidades. Segundo Neil MacDonald, especialista de segurança da Gartner, hoje em dia, as máquinas virtuais são menos seguras que as máquinas físicas justamente por causa do *hypervisor*. Mas há muita controvérsia nesse campo. Além disso, há a questão da disponibilidade de serviços que ocorre com o comprometimento, logicamente ou fisicamente, da máquina física que hospeda vários servidores virtuais, já que todos seriam afetados simultaneamente. É verdade que as soluções corporativas, como o ESX *Server* e o Citrix Enterprise, permitem toda uma monitoração dos sistemas, replicação, migração e *backup* dos sistemas virtualizados, que diminuem o tempo de recuperação em caso de problemas, mas é algo a ser avaliado. Um outro argumento freqüentemente empregado em favor da disponibilidade é que um *pool* de máquinas virtuais é similar a um *rack* com vários servidores físicos, com a vantagem de ter uma flexibilidade e uma portabilidade maior que estes últimos.

O segundo aspecto diz respeito ao gerenciamento das máquinas virtuais. Os ambientes virtuais necessitam ser instanciados, monitorados, configurados, mantidos e salvos. Novamente, existem produtos, como os apresentados na seção 4.4, que integram essas soluções. A Network World (*www.networkworld.com*), no mês de setembro de 2007, realizou uma avaliação das plataformas de gerenciamento oferecidas pela VMware, Microsoft e XenSource (Citrix) e concluiu que a plataforma VMware é a mais flexível e de fácil utilização, porém ainda há espaço para melhorias. Além disso, há um aspecto de gerenciamento relacionado com a segurança que é falho em todas essas plataformas: a correlação de eventos e ações feitas como *root* em ambientes virtuais. Ainda, segundo o Gartner, existem cerca de 50 companhias (setembro 2007) que oferecem produtos de gerenciamento para máquinas virtuais.

Por fim, mas não menos importante, a questão desempenho que pode ser formulada de duas formas. Primeira, qual o custo de processamento introduzido pela camada de virtualização? Segunda, quantas máquinas virtuais são possíveis sem comprometer uma qualidade de serviço (escalabilidade)? Para responder esses questionamentos, vários estudos foram feitos e se encontram disponíveis na Internet, porém, muitos deles são específicos a uma determinada situação e solução. Atento a essa necessidade, e como forma de sistematizar e fornecer uma metodologia padrão para avaliar o desempenho de máquinas virtuais, foi criado o SPEC *Virtualization Commitee* (*www.spec.org/specVirtualization*). O objetivo desse comité é desenvolver *benchmarks* para avaliar o desempenho de servidores em *data centers*. No momento da redação deste trabalho[4], a liberação para primeira versão de *benchmark* está prevista para o segundo semestre de 2008.

[4] Março de 2008.

Enquanto não há um *benchmark* específico para avaliar a virtualização, os estudos feitos até agora costumam empregar *benchmarks* existentes que simulam cargas de processamento, de entrada e saída, e de tráfego na rede. A partir de uma avaliação feita pela VMware [VMware 2008a] e posteriormente questionado e reconduzido pela Xen Source [XenSource2008a], se popularizaram os seguintes *benchmarks*: SPECcpu2000, focado em aplicações computações intensivas; o Passmark, que gera uma carga de trabalho para testar os principais subsistemas que compõem um sistema operacional; NetPerf, para avaliar o desempenho no envio e recepção de dados via rede; SPECjbb2005 que representa um servidor e sua carga; e a compilação do pacote SPECcpu2000 INT. A tabela 4.1 sumariza os resultados obtidos pela VMware e pela Xen Source feitos com esses *benchmarks*:

Tabela 4.1 – Resumo das conclusões sobre análise de desempenho (VMware *versus* XenSource)

Benchmark	Estudo VMware	Estudo Xen Source
SPECcpu2000	Dependendo da carga, o ESX *Server* tem um sobre-custo entre 0 a 6% contra 1 a 12% do Xen 3.0.3 em relação ao sistema nativo.	O ESX *Server* tem um sobre-custo de ≈2% e o Xen 3.2 de ≈ 3% em relação ao sistema nativo.
Passmark	Em função do teste, o ESX *Server* gera de 4 a 18% de sobre-custo contra 6 a 41% do Xen 3.0.3 em relação ao sistema nativo.	Apresentam desempenho similar com uma variação de um em relação ao outro na faixa de 0.5% e, em função do teste, ambos tem um desempenho entre 80 a 95% do apresentado pelo sistema nativo.
Netperf	O ESX *Server* fornece desempenho próximo ao sistema nativo (≈98%), ao passo que o Xen 3.0.3 atinge apenas 2 a 3% dessa capacidade.	Desempenho similares e próximos do sistema nativo (≈98-99%). No teste com mais de um cliente há diferenças em favor do Xen (≈90%)
SPECjbb2005	O ESX *Server* corresponde a 91% do sistema nativo. Não foi possível fazer esse experimento com o Xen 3.0.3 por ele não ter suporte a máquinas SMP.	Desempenho similares e próximos ao sistema nativo (≈96-98%). Obs.: A versão 3.2 do Xen oferece suporte a máquina SMP.
Compilação	O ESX Server oferece um desempenho de ≈90% contra ≈68% do oferecido pelo Xen 3.0.3, em comparação ao sistema nativo.	ESX *server* tem um desempenho de cerca de 92% em relação ao sistema nativo, enquanto o do Xen é ≈86%

Abstraindo o fato de que os resultados da tabela 4.1 foram obtidos com equipamentos e softwares de versões diferentes e de terem sido feitos por entidades que tinham o intuito de "vender seu produto", o importante a ressaltar é o impacto da virtualização em relação a um sistema nativo. Os números fornecidos dão uma idéia do sobre-custo introduzido pela virtualização. A conclusão é que, em geral, esse sobre-custo não é muito significativo, portanto, seu uso não tem um impacto importante sobre a realização de um serviço.

A nosso conhecimento, não foram feitos *benchmark*s para avaliar a escalabilidade de *hypervisors*. Vale lembrar ainda que os *benchmarks* empregados foram desenvolvidos para um contexto e empregados em outro. Em todo caso, há uma movimentação geral entre os fornecedores de soluções para virtualização em reduzir, otimizar e melhorar o desempenho de seus produtos.

8. Conclusões

A virtualização é a nova onda de revolução na área de TI. Isso é confirmado por análises econômicas conduzidas pela Gartner, pela criação e pela evolução de associações como a EMA (*Enterprise Management Association*) e pelo crescimento exponencial de empresas como VMware e a Citrix, entre outras. Ainda, o IDC prevê que o crescimento do mercado e dos investimentos em virtualização subirá de U$6.5 bilhões para U$15 bilhões em 2011. Não é por nada que a Citrix desembolsou, em outubro de 2007, U$500 milhões para adquirir a XenSource.

A virtualização é uma técnica que permite que um sistema computacional (o hospedeiro) execute vários outros denominados de máquinas virtuais (hóspedes). Cada máquina virtual, por si só, é um ambiente de execução isolado e independente das demais. Com isso, cada máquina virtual pode ter seu próprio sistema operacional, aplicativos e serviços de rede (Internet). O sistema operacional do hóspede pode ser diferente daquele utilizado pelo hospedeiro. A virtualização não é um conceito recente, remonta da década de 60, e existem várias técnicas para implementá-la, onde se destacam as máquinas virtuais de processo e o monitor de máquina virtual.

Uma máquina virtual de processo nada mais é que uma aplicação que executa sobre um sistema operacional *A* e que emula o comportamento de um sistema operacional *B*. Assim, aplicações desenvolvidas para o sistema *B* podem executar sobre um sistema *A*. É importante salientar que essa técnica de implementação permite que binários de um processador sejam interpretados e substituídos por código equivalente de outro processador. Portanto, além de emular sistema operacional é possível emular processadores. As desvantagens dessa técnica são basicamente duas: desempenho e desperdício de capacidades do hardware físico. O desempenho é sacrificado já que há uma tradução de um sistema a outro, além de executarem em modo de usuário. O desperdício de capacidades físicas do hardware vem do fato que as máquinas virtuais de processo oferecem dispositivos de E/S genéricos e simples. O exemplo típico são placas de vídeo. Por executarem como uma aplicação, ao terminar o "processo máquina virtual", nenhum rastro permanece no sistema.

Os monitores de máquinas virtuais, mais conhecidos pelo acrônimo VMM (*Virtual Monitor Machine*) ou *hypervisor*, surgem para resolver as desvantagens das máquinas virtuais de processos. Eles são implementados como uma camada de software entre o hardware e o sistema operacional, oferecendo uma máquina virtual para este. Dessa forma eles conhecem e exploram eficientemente os dispositivos físicos de E/S. O desempenho tende a ser melhor por não executarem em modo usuário, o que evita chaveamentos de contexto. Existem duas técnicas usadas nos *hypervisors*: virtualização total e para-virtualização. A diferença essencial é se o sistema operacional hóspede precisa ser modificado (para-virtualização) ou não (virtualização total) para executar sobre o *hypervisor*.

Assim como já aconteceu no passado com a multiprogramação, o projeto dos processadores mais recentes tem considerado mecanismos em hardware para auxiliar na virtualização. É o caso dos fabricantes AMD e Intel que desenvolveram extensões para a arquitetura x86 suportar virtualização, respectivamente, AMD-Virtualization (AMD-V, codinome *Pacífica*) e Intel *Virtualization Technology* (IVT, condinome *Vanderpool*).

O uso típico da virtualização é na consolidação de servidores, isso é, permitir que vários servidores executem simultaneamente em um único hardware físico, mas cada um em sua própria máquina virtual. Como cada máquina virtual é um ambiente isolado, completo, os servidores podem ser de sistemas operacionais diferentes e, um eventual comprometimento de um não afeta os demais. Além disso, a consolidação de servidores reduz custos com aquisição de equipamentos, e com infra-estrutura física como espaço, energia, cabeamento, refrigeração e custos de gerenciamento e manutenção. Isso é especialmente interessante em *data centers*.

Entretanto a virtualização pode ser empregada com sucesso em várias outras situações como ambientes de desenvolvimento e teste de produtos, laboratórios de treinamento de cursos de redes e de sistemas operacionais, criação de clusters ou grades computacionais virtuais e servir de base para implantação de mecanismos de segurança (*honeypots*).

Apesar de muitas vezes se associar virtualização a sistemas operacionais hóspedes e a consolidação de servidores, ela é algo mais amplo. Segundo a EMA [EMA, 2008] virtualizar é "mascarar características físicas" de recursos computacionais dos sistemas operacionais, aplicações e usuários. Nesse ponto entra a possibilidade "mascarar" a infra-estrutura de rede. Há muito tempo, os sistemas operacionais já oferecem suporte a definição de interfaces de redes virtuais, permitindo que uma mesma placa de rede funcione como se fosse várias. De forma natural, cada máquina virtual possui sua própria placa de rede (virtual) com seu próprio endereço IP e endereço MAC. O próximo passo foi permitir que essas máquinas virtuais fossem interconectadas entre si usando equipamentos de interconexão também virtuais. Surgem assim *switches* e roteadores virtuais, permitindo que se faça (virtualmente) a definição de redes de perímetro e segmentação de redes.

Como a virtualização consiste basicamente em se pôr uma camada de software a mais em um sistema computacional a questão sobre quanto isso afeta o desempenho final é imediata. Estudos feitos pela VMware e pela XenSource apontam para um queda de desempenho, em geral, entre 2% e 10%, com algumas situações impondo perdas maiores. Cabe ressaltar que esses resultados foram obtidos usando *benchmarks* genéricos. Porém, atento a esse aspecto, os fabricantes estão investindo muitos esforços no sentido de reduzir a queda de desempenho e para que haja uma forma padrão, e isenta, de avaliar, há um comitê específico (SPEC *Virtualization Comitee*) trabalhando na definição de uma suíte de *benchmark* para a virtualização. A primeira versão dessa suíte é esperada para a segunda metade de 2008.

Para finalizar, um ponto interessante. Segundo a Microsoft (em seu site *technet*) o valor de mercado da virtualização não está na pilha necessária para implementá-la, isso é, no conjunto sistema operacional mais *hypervisor*, mas sim no desenvolvimento de sistemas de gerenciamento para ambientes virtuais. Essa opinião é reforçada por um outro estudo do Gartner que aponta um crescimento em *start-ups* oferecendo soluções

de gerenciamento para ambientes virtuais. Fazem parte do gerenciamento, as soluções de segurança. Um tópico, entre outros, que está em aberto é como correlacionar as atividades de *root* em um *hypervisor* com as diferentes máquinas virtuais. Questões relacionadas com desempenho, migração de máquinas virtuais, facilidades de *backup*, recuperação de falhas, configuração automática, também não estão totalmente fechadas.

Referências

Adamic, A. L. "Zipf, Power-laws and Pareto – a ranking tutorial". Information Dynamics Lab, HP Labs. (http://www.hpl.hp.com/research/idl/papers/ranking), acesso fevereiro 2008.

Adams, K.; Agesen, O. "A comparaison of software and hardware techniques for x86 virtualization", ASPLOS'06, San Jose, California, USA. 2006.

Baratz, A.; "Virtual Machines shootout: VMWare vs. Virtual PC", Ars Technica. August, 2004 (http://arstechnica.com/reviews/apps/vm.ars) Acesso novembro 2007.

Barham, P.; Dragovic, B.; Fraser, K.; Hand, S.; Harris, T.; Ho, A.; Neugebauer, R.; Pratt, I.; Warfield, A. "Xen and the Art of Visualization". In. Proc. 19th ACM Symp. On Operating System Principles (SOSP'03), 2003.

Breslau, L.; Cao, P.; Fan, L. ; Philips, G. And Shenker, S. "Web Caching and Zipf-like Distributions: Evidence and Implications". INFOCOM, 1999, pp. 126-134.

EMA, Entreprise Management Solutions. (http://www.emscorporation.com). Acesso março de 2008.

Goldberg, R.P. "Survey of virtual machine research". IEEE Computer, pp. 34-35, june 1974.

Jain, R. "Art of Computer Systems Performance Analysis". John Willey. 1991.

Henessy, J.L.; Patterson, D.A. "Computer Architecture: A quantitative approach". 4th edition. Morgan Kauffman Publishers, 2007.

LinuxServer (http://linux-vserver.org). Acesso em março de 2008.

Microsoft. "Microsoft Virtual PC" (http://www.microsoft.com/Windows/virtualpc), acesso novembro 2007.

Microsoft. "Microsoft Virtual *Server*" (http://www.microsoft.com/windowsserversystem/virtualserver), acesso novembro 2007.

Neiger, G.; Santoni, A. et alli "Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization", Intel Technology Journal, v. 10, pp. 166-179, 2006.

Oliveira, R..; Carissimi, A.; Toscani, A.; *Sistemas Operacionais*. Editora Sagra-Luzzato, 3ª edição, 2004.

Popek, G.; Goldberg, R. "Formal requirements for virtualizable 3rd generation architectures". Communications of the ACM, v.17, n.7, pp. 412-421, 1974.

Robin, J.S.; Irvine, C.E. “Analysis of the Intel Pentium’s Ability to Support a Secure Virtual Machine Monitor”. Proc. 9th USENIX Security Simposium, 2000.

Rose, R. “Survey of Virtualization Techniques”. (http://www.robertwrose.com). Acesso em fevereiro 2008.

Silberchatz, A.; Galvin, P; *Sistemas Operacionais*. (1a edição). Campus, Rio de Janeiro, 2001.

Singh, A.“An introduction to virtualization”(http://www.kernelthread.com/publications) Acesso fevereiro 2008.

Smith, J.E, Nair, R. “The architecture of virtual machines”. IEEE Computer, v.38, n.5, pp. 32-38, 2005.

Uhgli, R.; Neiger, G. et alli “Intel Virtualization Technology”. Computer Journal, v.38, pp. 48-56. 2005.

Ung, D.; Cifuentes, C. Machine-adaptable dynamic binary translation. ACM SIGPLAN Notices, ACM Press New York, NY, USA, vol. 35, 7, july 2000, pp 41-51.

VMWare (2008a) “A Performance Comparison of Hypervisors” (http://www.vmware.com). Acesso janeiro 2008.

VMWare (2008b) “VMWare virtual networking concepts” (http://www.vmware.com) Acesso fevereiro 2008.

VMWare (2008c), VMWare (http://www.vmware.com)

Wikipedia. “Comparaison of virtual machines” Wikipedia, The Free Encyclopedia, Acesso novembro 2007.

Xen Source (2008a), “A Performance Comparaison of Commercial Hypervisors” (http:/blogs.xensources.com/rogerk/wp-content/uploads/2007/03/*hypervisor*_performance_comparaison_1_0_5_with_esx-data.pdf). Acesso fevereiro 2008.

Xen Source (2008b), Xen Source (http://www.xensource.com). Acesso fevereiro 2008.