

Introdução a Virtualização

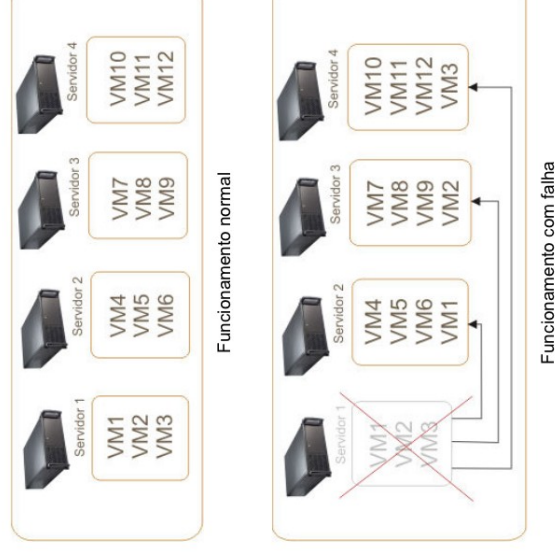
Sergio Roberto Charpinel Junior
Profa. Roberta Lima Gomes

Por que virtualizar?

- Descentralização de recursos computacionais
 - Cloud computing
- Plena utilização de recursos físicos
 - “Do more with less”
 - Reaproveitamento de recursos
- Diferentes SOs no mesmo hardware
 - Isolamento de aplicações
 - Segurança
- Redução no número de máquinas físicas
 - Economia de energia, espaço, dinheiro

Por que virtualizar?

- Facilidade de gerenciamento
- Treinamentos e ambientes de ensino
- Facilidade para restauração e recuperação de serviços
 - Maior disponibilidade
 - **Tolerância a falhas**
- Etc...

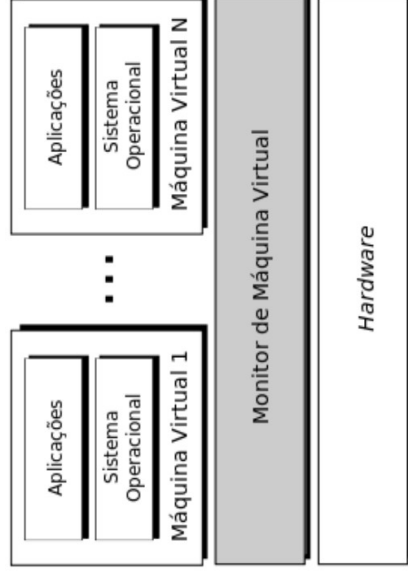


Contexto histórico

- Surgiu na década de 60 na IBM
 - Soluções combinadas em hardware e software (desempenhol)
 - Dividir logicamente o mainframe
 - Recurso caro, necessário utilização completa
- Popularização do x86 no final da década de 80
 - Desktops
 - Virtualização ficou de lado
- Popularização da Internet a partir da década de 90
 - Alta disponibilidade
 - Necessidade de economia de recursos
 - Virtualização ganha espaço novamente

Conceitos - MMV

- Monitor de Máquina Virtual (MMV) ou Virtual Monitor Machine (VMM)
 - Camada de software que abstrai os recursos físicos para utilização das máquinas virtuais
 - Hipervisor ou Hypervisor



Conceitos

- “Uma máquina virtual é uma cópia eficiente e isolada da máquina real” (POPEK; GOLD-BERG, 1974).
- Existem duas categorias de máquinas virtuais:
 - Máquinas virtuais de processo (ex. Java)
 - Execução de programas
 - Máquinas virtuais de sistemas (ex. Xen)
 - Execução de SO completo



Conceitos - MMV

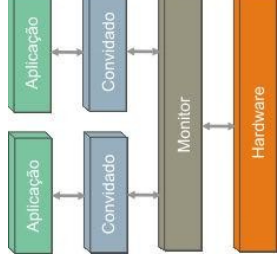
- Definições de Popek e Goldberg:
 - O MMV deve fornecer aos programas um ambiente idêntico ao da máquina original.
 - Os programas nesse ambiente devem apresentar como perda apenas uma diminuição de sua velocidade de execução.
 - O MMV deve possuir controle completo sobre os recursos do sistema
- O MMV deve interpretar e emular o conjunto de instruções entre as máquinas virtuais e a máquina real

Conceitos - MMV

- Principais funções do MMV:
 - Definir o ambiente de máquinas virtuais.
 - Alterar o modo de execução do sistema operacional convidado de privilegiado para não privilegiado, e vice-versa.
 - Emular as instruções e escalonar o uso da CPU para as máquinas virtuais
 - Muitas instruções do processador virtual devem ser executadas diretamente pelo processador real, sem que haja intervenção do monitor (eficiência!)
 - Gerenciar acesso aos blocos de memória e disco destinados ao funcionamento das máquinas virtuais.
 - Intermediar as chamadas de sistema e controlar acesso a outros dispositivos como CD-ROM, drives de disquete, dispositivos de rede, dispositivos USB.

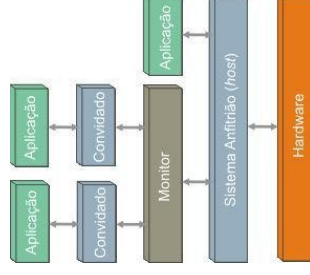
Tipos de Máquinas Virtuais

- Máquinas virtuais clássicas ou de Tipo I
 - O monitor é implementado entre o hardware e os sistemas convidados
 - Executa com a maior prioridade sobre os sistemas convidados
 - Pode interceptar e emular todas as operações que acessam ou manipulam os recursos de hardware
 - VMware ESXi Server,
 - Microsoft Hyper-V,
 - Citrix/Xen Server



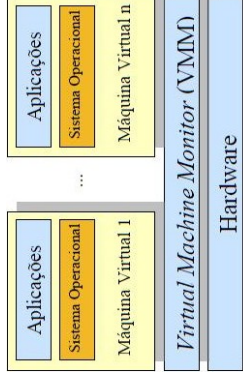
Tipos de Máquinas Virtuais

- Máquinas virtuais Hospedadas ou de Tipo II
 - O monitor é implementado como um processo de um sistema operacional "real"
 - O monitor simula todas as operações que o sistema anfitrião controlaria
 - VMware Workstation
 - Microsoft Virtual PC
 - Oracle Virtual Box



Virtualização Total

- Provê uma completa simulação da subcamada de hardware para os sistemas convidados
 - Todos os SOs que são capazes de executar diretamente em um hardware também podem executar em uma máquina virtual
 - Não há necessidade de modificações nos sistemas operacionais convidados
- O Monitor roda em modo kernel, e os sistemas convidados em modo usuário
 - Todas as instruções são "testadas"
 - As **instruções sensíveis** são capturadas e emuladas na VM
 - trap-and-emulate



Virtualização Total

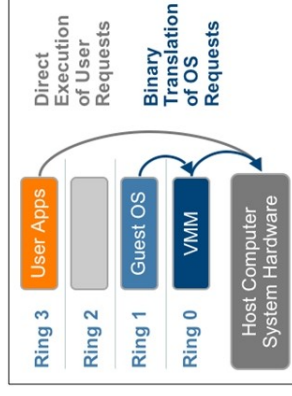
- Desvantagens
 - O número de dispositivos a serem suportados pelo Monitor é extremamente elevado
 - Para resolver... uso de dispositivos genéricos
 - Instruções executadas pelo SO visitante devem ser testadas pelo Monitor
 - Ter que contornar alguns problemas gerados pela implementação dos SOs
 - SOs foram projetados para serem executados como instância única nas máquinas físicas
 - Ex: Paginação → Disputa SOs → queda de desempenho

Virtualização Total

- Instruções sensíveis
 - Podem ser executadas somente em modo kernel
- Instruções privilegiadas
 - Geram trap quando executadas em modo usuário
- Segundo Popok e Goldberg, uma máquina é virtualizável se:
 - Instruções sensíveis formarem um subconjunto de instruções privilegiadas
- IBM/370 é virtualizável
- Intel 386 não é estritamente virtualizável
 - Algumas instruções têm comportamentos diferentes no modo usuário e no modo supervisor
 - Ex: Instruções de leitura de estado privilegiado e instruções que alteram tabela de páginas não geram trap.
- Resolvido em 2005 com Intel VT e AMD SVM
 - Implementam o modo hypervisor (entre o HW e o modo kernel)
 - Instruções sensíveis geram trap

Virtualização no x86 (Hypervisor 1)

- Solução Trap-and-emulate
 - Nas CPUs VT e SVM instruções sensíveis geram trap
 - MMV é alocado no nível 0
 - SO virtualizado no nível 1 (ou 3)
- Tradução binária...



Virtualização no x86 (Hypervisor 1)

- Tradução binária dinâmica
 - O monitor analisa, reorganiza e traduz as sequências de instruções emitidas pelo sistema convidado em novas sequências de instruções *on-the-fly*
 - O código é dividido em blocos e estes são então verificados
 - Visa-se com isso
 - (a) Adaptar as instruções geradas pelo sistema convidado à interface ISA do sistema real, caso não sejam idênticas;
 - (b) Detectar e tratar instruções sensíveis não-privilegiadas (que não geram traps ao serem invocadas pelo sistema convidado); ou
 - (c) analisar, reorganizar e otimizar as sequências de instruções geradas pelo sistema convidado, para melhorar o desempenho .
 - Blocos de instruções muito frequentes podem ter suas traduções mantidas em cache
 - Instruções privilegiadas são substituídas por chamadas de rotina do MMV
 - MMV emula instruções

Virtualização no x86 (Hypervisor 2)

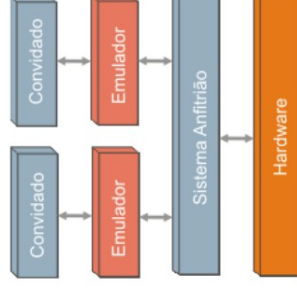
- MMV é um programa de usuário
- Tradução binária dinâmica
 - Instruções privilegiadas são substituídas por chamadas de rotina do MMV
 - MMV emula instruções

Paravirtualização

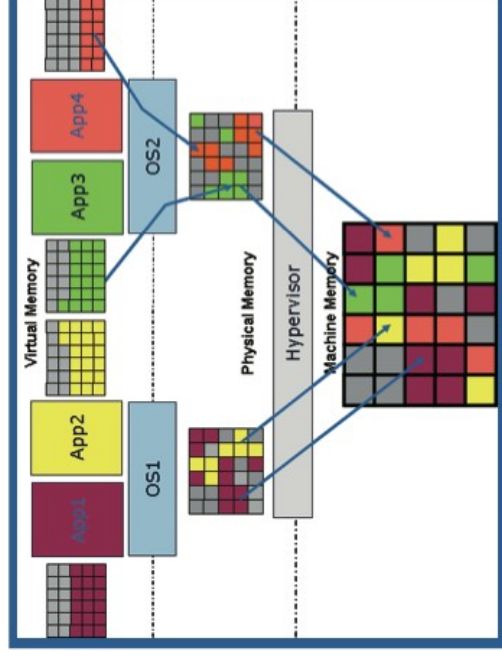
- Virtualização total
 - Desacoplamento maior da máquina física
 - Ex.: Hypervisor 1 e Hypervisor 2
- Paravirtualização
 - MMV fornece uma API para MVs
 - SO virtual é modificado
 - Instruções sensíveis são substituídos por chamadas ao MMV
 - Ganho de desempenho
 - Paravirt ops – API da MMV padronizada

Outras técnicas

- Emulação
 - simula o hardware do sistema para a execução do sistema convidado
 - “Traduz” instruções do sistema convidado para equivalentes no sistema anfitrião e vice-versa
 - Ex: QEMU

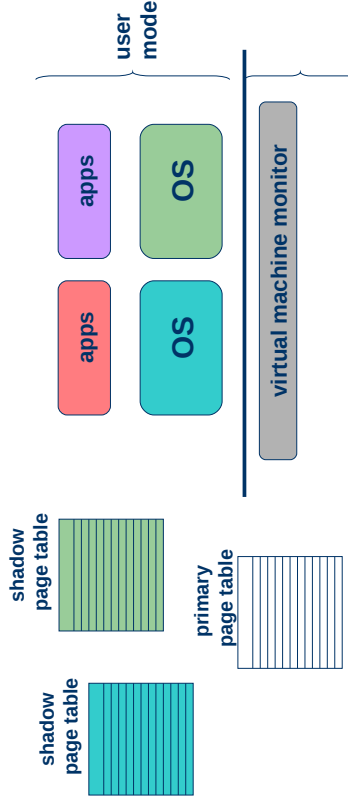


Virtualização de memória

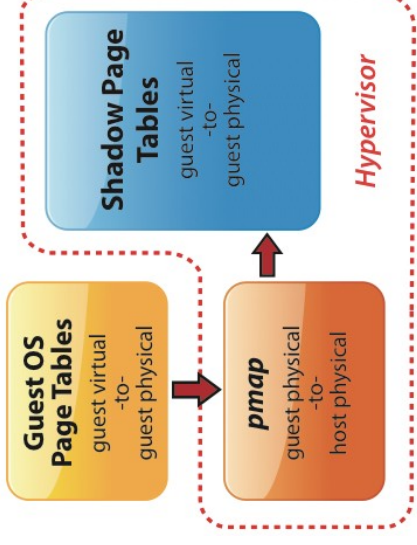


Virtualização de memória

- Tabela de páginas de sombra (Shadow page tables)
 - VMs exigem MMU emulada pelo Monitor
 - Monitor captura page fault e converte endereços virtuais do guest em endereços físicos do host



Virtualização de memória

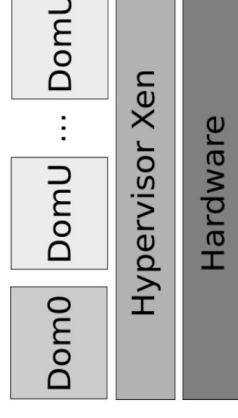


Virtualização de E/S

- Virtualização de disco
 - Arquivos, partições LVM, partições físicas, etc.
 - Pode expor disco diferente do real para MV e traduzir chamadas
- DMA
 - MMV pode/deve traduzir endereços
 - Hardwares atuais possuem MMU para E/S
- Solução Xen:
 - Uma MV (dom0) executa SO padrão e demais MVs (domUs) direcionam suas chamadas de E/S para ela

Xen

- Gratuito e de código aberto
- Desenvolvido por grandes empresas:
 - Citrix, AMD, HP, IBM, Intel, etc.
- Customizável



Xen

- DomU PV
 - Paravirtualizada
 - E/S com memória compartilhada
- HVM
 - Virtualização completa
 - E/S emulada pelo QEMU
- HVM-PV
 - HVM com drivers PV
- Suporte a live migration
- Discos compartilhados

VMware

- Desktop
 - VMware Workstation
 - VMware Fusion
 - VMware Player (free)
- Server
 - VMware ESX e VMware ESXi (free)
- Cloud
 - VMware vCloud
- Gerenciamento
 - VMware vCenter