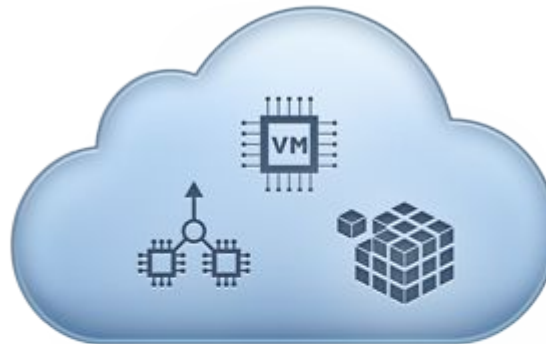


Computação em nuvem

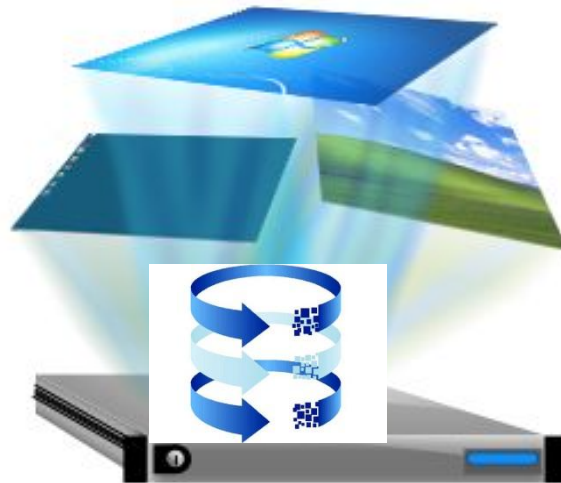
Tecnologias de Suporte à Computação em Nuvem



Prof. Dr. Marcos A. Simplicio Jr.
Laboratório de Arquitetura e Redes de Computadores
Departamento de Engenharia de Computação e
Sistemas Digitais
Escola Politécnica da Universidade de São Paulo

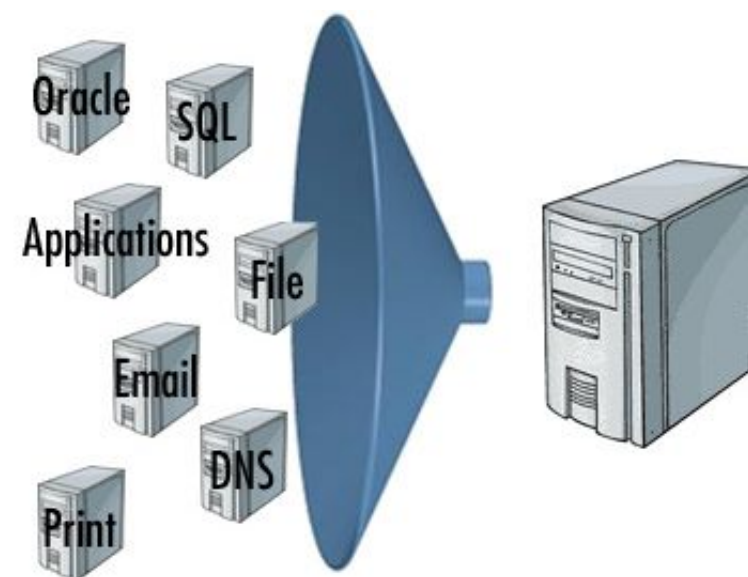
Objetivos – Aula 16

- Entender o conceito de virtualização e suas diversas formas

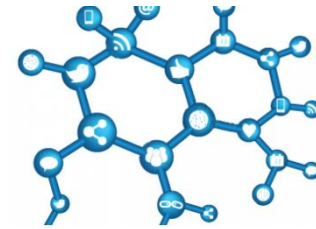


Virtualização

- Recursos computacionais em **Data Centers**
 - **Dimensionados** para **picos**
 - Máquinas **subutilizam processador**: boa parte do tempo ficam na espera de I/O (ex.: rede, disco)
 - Aplicações não utilizam o processador o tempo todo
- **Virtualização:**
 - Várias “**máquinas virtuais**” compartilhando uma infraestrutura



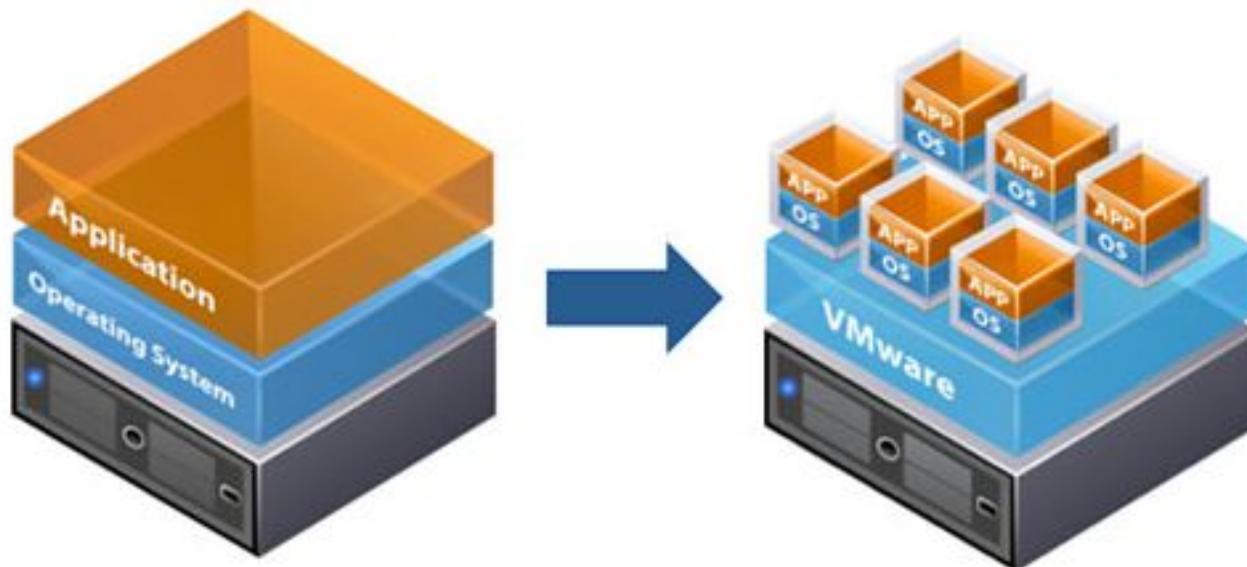
Virtualização: Conceitos



- ❑ **Conceito amplo** de virtualização
 - **Abstração** de um conjunto **de recursos** (computacionais e/ou de comunicação)
 - Abstração representa recursos com características diferentes
- ❑ Alguns exemplos já devem ser bem conhecidos
 - **VLANs** (Virtual LANs): abstração segmento **rede**
 - **VPNs** (Virtual Private Networks): abstração de **enlace**
 - **Java Virtual Machine**: abstração da **plataforma de execução** da aplicação
 - **RAID**: abstração de **armazenamento**

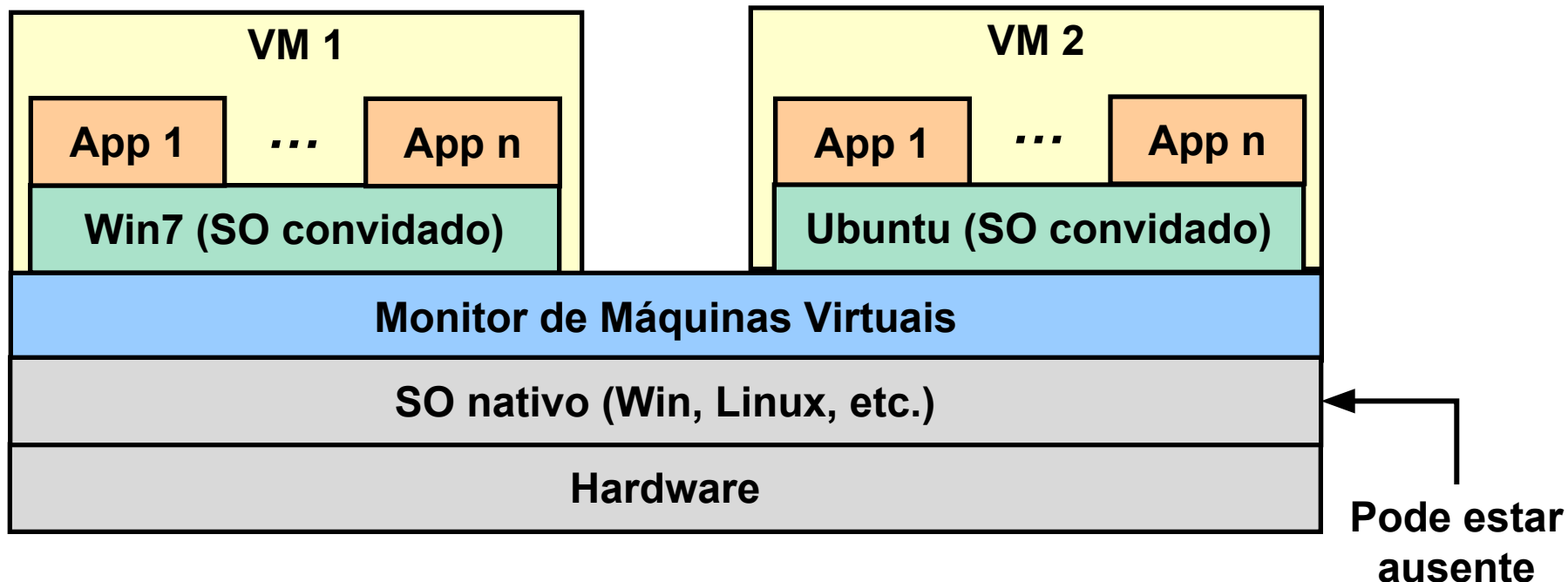
Virtualização: Máquinas Virtuais

- ❑ Máquina Virtual (*Virtual Machine* – VM)
 - Hardware físico é representado por um **hardware virtual**
 - Várias VMs podem ser executadas na mesma máquina física
 - VMs gerenciadas pelo **Monitor de Máquinas Virtuais**
 - Função essencial: **isolamento**



Virtualização: Monitor de Máquinas Virtuais

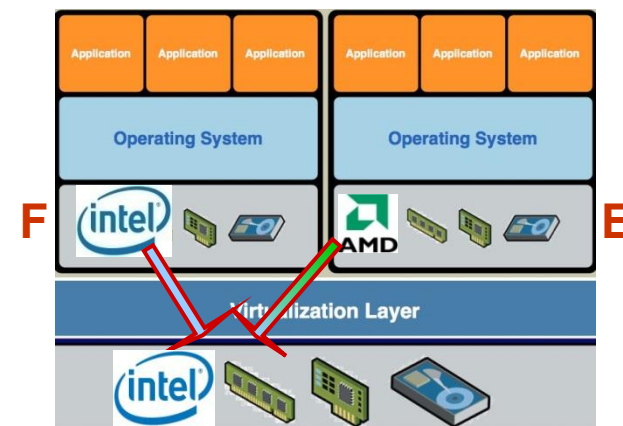
- ❑ Monitor de Máquinas Virtuais (MMV) ou Hypervisor
 - Executado sobre um sistema operacional ou diretamente sobre o hardware



Virtualização: tipos

□ Emulação (E)

- Máquina virtual **simula um hardware** com **CPU diferente** do hardware físico
- **MMV** precisa **converter instruções**
- **SO** funciona **sem modificações**
- Ex.: uma máquina virtual x86 em um computador com processador PowerPC (Virtual PC for Power Mac)



□ Virtualização Nativa/ Full Virtualization (F)

- Máquina virtual **simula um hardware** com **mesma CPU** do hardware físico
- **MMV** precisa controlar instruções, **sem convertê-las**
- **SO** funciona **sem modificações**
- Ex.: Intel x86 sobre Intel x86 (VMWare, Parallels)

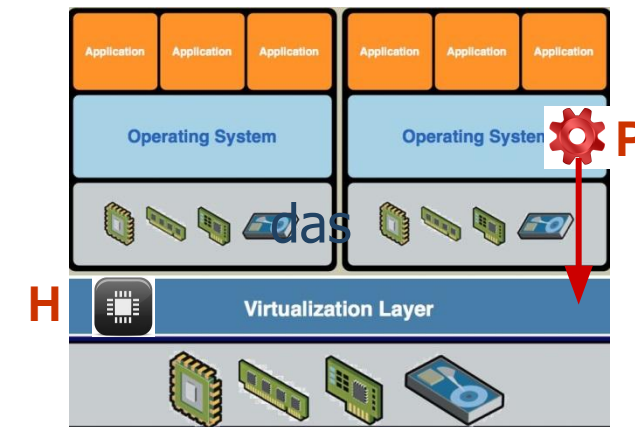
Virtualização: tipos

□ Assistida por Hardware (**H**)

- Hardware **auxilia no isolamento** máquinas virtuais
- Hardware se encarrega de **partes ineficientes e/ou complexas** da virtualização via software
- Ex.: processadores Intel com tecnologia VT + Linux XEN

□ Paravirtualização (**P**)

- **VM não simula completamente o hardware**
- **SO é modificado** para efetuar chamadas relacionadas ao monitor de máquinas virtuais
- Ex.: Linux XEN, VMWare VMTools



Virtualização: usos

- ❑ Consolidação de servidores físicos



- **Menos máquinas físicas** para gerenciar
- **Menor \$\$** de infraestrutura

- ❑ Recuperação de desastres / Manutenção



- Mais fácil **migrar** uma máquina virtual que reinstalar um SO

- ❑ Ambiente de testes



- Servidor físico pode ser recriado em **ambiente lógico para teste** de novas aplicações

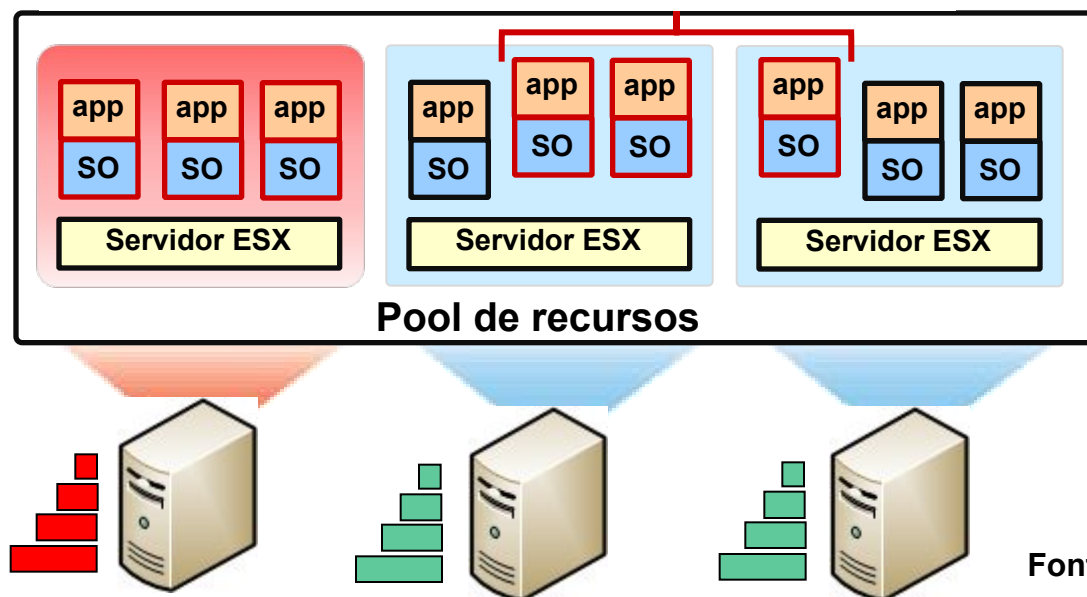
- ❑ Virtualização de serviços

- Serviços executados no servidor por **thin clients**



Virtualização: gerenciamento

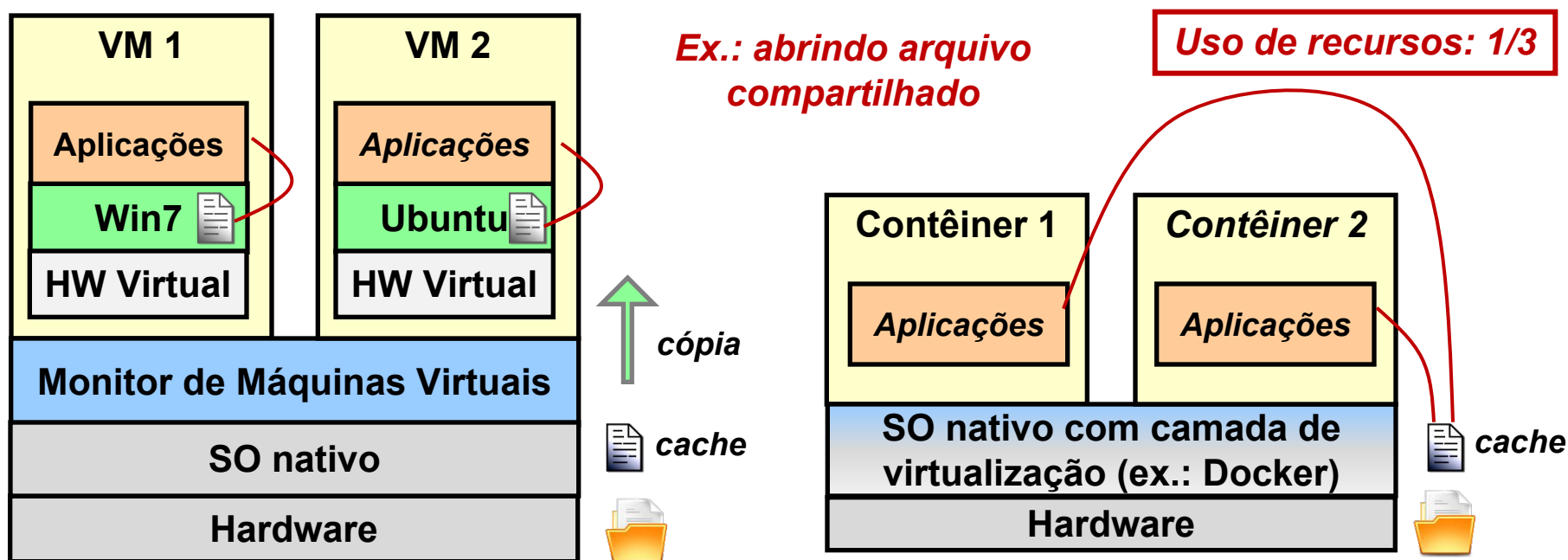
- ❑ Máquinas virtuais + Gerenciamento: permitem obter **utility computing**
 - Hardware virtual é padronizado
 - **Migração automática** de VMs padrão entre máquinas físicas
 - Máquina física fora do ar ou sobrecarregada



Fonte: VMware Virtual Infrastructure 3 (VI3)

Virtualização: Contêineres

- **Contêiner:** virtualização no nível de SO
 - **Menos “camadas”** para chegar ao hardware
 - **Melhor gerenciamento** de recursos (compartilhamento)



Virtualização: Contêineres



- ❑ História: conceito existe desde 2001 (Linux)
 - E Google usa contêineres em seus sistemas (ex.: GAE)
 - Maior visibilidade: 2013, com **Docker p/ gerenciamento**

- ❑ Vantagens:

- **Densidade:** mais leves → $\# \text{contêineres/host} > \# \text{VMs/host}$
- **Elasticidade:** criação e destruição rápidas (secs, não mins)
 - Especialmente para serviços pequenos (ex.: web)



- ❑ Limitações:

- Não suporta múltiplos kernels sobre mesmo kernel
 - Cenário ideal p/ contêineres: sistemas **homogêneos**
 - Nota: há soluções para suporte a contêineres de **distros Linux**
- **Segurança:** menos testada do que VMs



Resumo

- ❑ Entender o conceito e virtualização e suas diversas formas
 - **Virtualização**: abstração de um conjunto de recursos (ex.: máquina virtual → hardware subjacente)
 - **Tipos de virtualização**: emulação, nativa, assistida por hardware e paravirtualização
 - **Usos**: redução de custos, maior facilidade de manutenção, segurança, utility computing
 - **Contêineres**: virtualização por SO, na forma de processos

- ❑ Próxima aula: SDN

