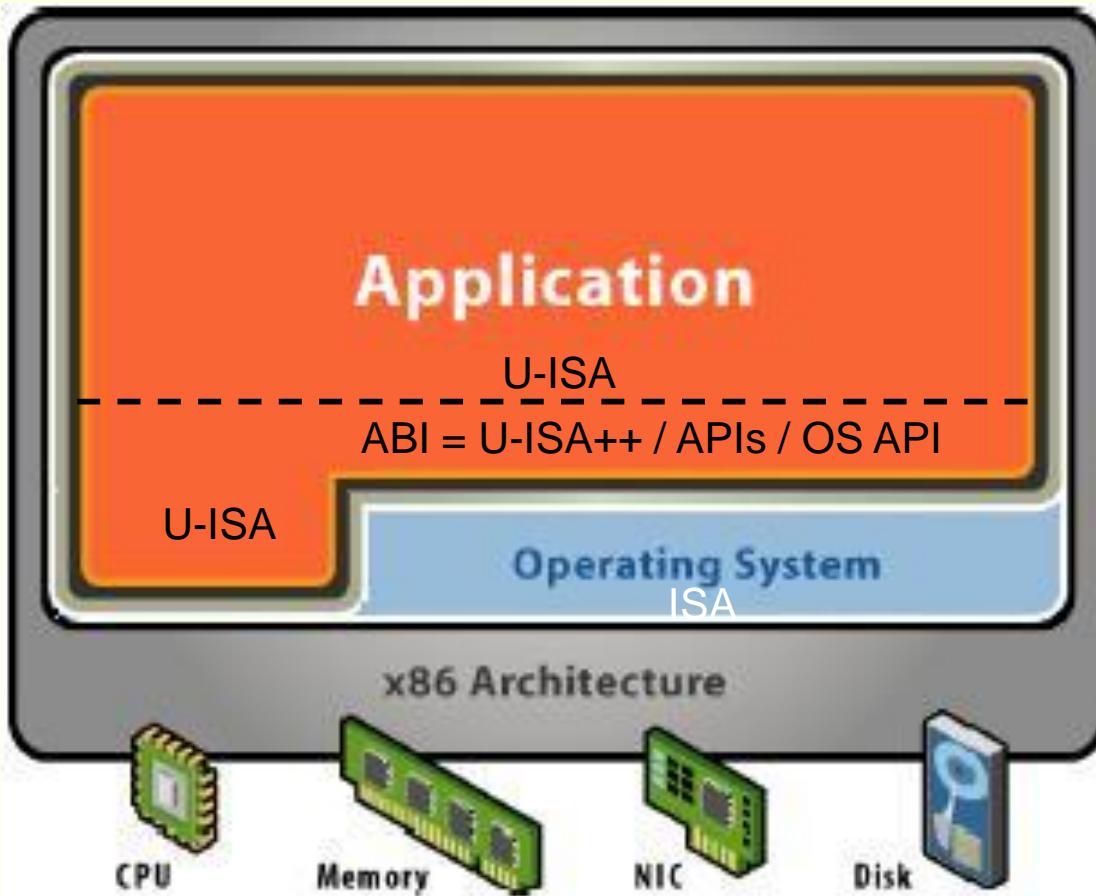


Fundamentos de Sistemas de Operação

Unix Windows NT Netware Mac OS DOS/V/VS Vax/VMS
Linux Solaris HP/UX AIX Mach Chorus

A idade do ágil:
2. Virtualização e VMs: Parte I

Aplicação, SO e Máquina (1)



ISA: Instruction Set Architecture

U-ISA: ISA disponível p/o utilizador

U-ISA++: U-ISA mais a “syscall”/trap

APIs: Linguagens, bibliotecas e APIs do SO

ABI: Application Binary Interface

Aplicação, SO e Máquina (2)

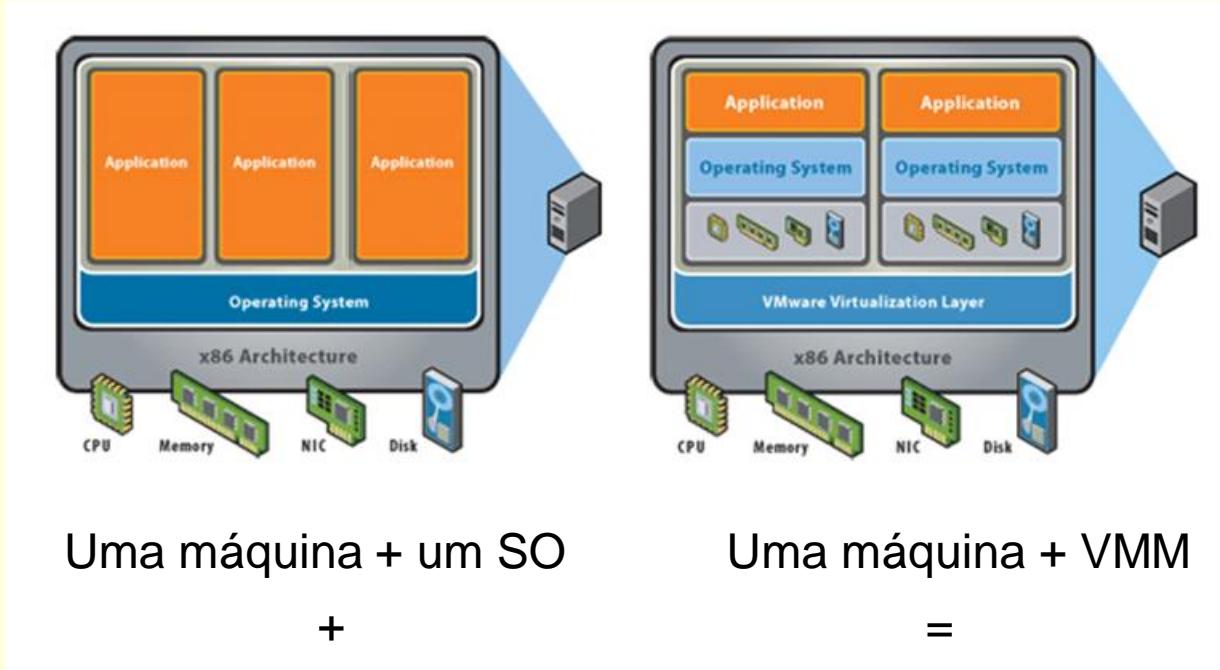
□ A aplicação,

- Executa instruções U-ISA (`mov`, `push`, `call`, `jmp`, `add`, etc.) que não requerem intervenção do SO para se completarem (desde que, naturalmente, não “façam” nada de ilegal)
- Executa a instrução `trap` / `int` para solicitar serviços ao SO
- Usa (em runtime) as APIs oferecidas pelas linguagens bibliotecas adicionais (e.g., sockets), e a API do próprio SO
- Se a aplicação obedece ao standard ABI, pode até ser copiada (em formato binário/executável) para outro SO distinto e executada lá

□ The OS

- Pode usar o ISA completo para executar operações sobre o hardware

O que é virtualização? (1)



Uma máquina + um SO

+

"montes" de aplicações

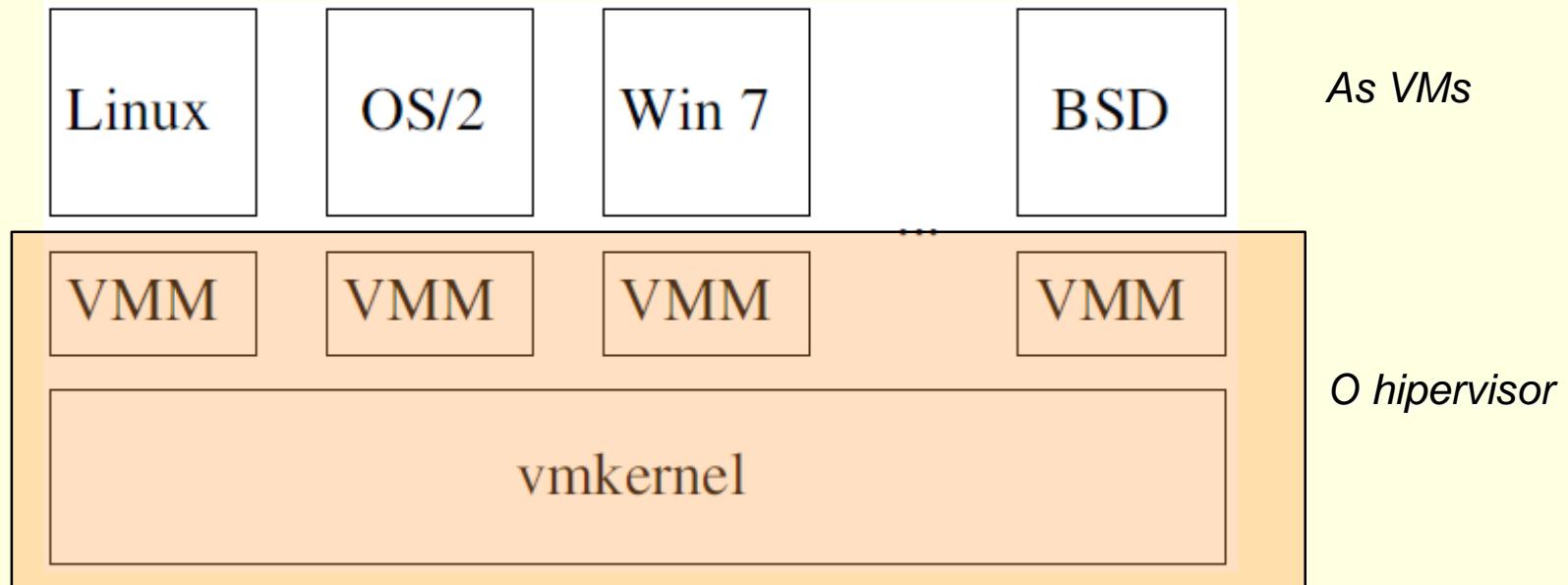
Uma máquina + VMM

=

"montes" de máquinas virtuais
cada uma com o seu SO e
aplicações

O que é um hipervisor? (1)

Unix Windows NT Netware Macos DOS/V/ Vax/VMS
Linux Solaris HP/UX AIX Mach Chorus



From: **The Evolution of an x86 Virtual Machine Monitor**
ACM OSR Volume 44, Number 4, December 2010

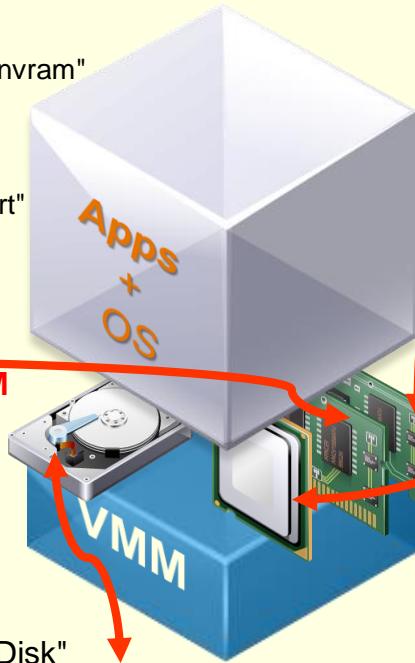
O que é um hipervisor? (2)

- A máquina virtual
 - É um “parente próximo” da “máquina física”, com o seu “hardware virtual” especificado num ficheiro (de texto) criado/gerido pelo VMM
 - Podem existir várias instâncias de VMs em execução, e elas
 - Comportam-se como máquinas distintas, isoladas
 - Correm as aplicações “tal-qual”, sem serem modificadas, e eficientemente
- O Virtual Machine Monitor (VMM)
 - Implementa o “hardware virtual” sobre o qual corre a VM
 - Um par (VM+VMM) constitui um “processo” do hipervisor.
- O hipervisor
 - É um SO (especializado) que executa processos (pares VM/VMM)
 - O VMM está tão intimamente ligado ao hipervisor que alguns autores usam os dois termos como sendo o mesmo, por isso... se quiser os detalhes, leia o artigo ☺.

O ficheiro de especificação da VM

Caso de estudo: VMware ESXi VM

```
...  
virtualHW.version = "10"  
nvram = "CentOS7-Minimal64-1708.nvram"  
pciBridge0.present = "TRUE"  
svga.present = "TRUE"  
pciBridge4.present = "TRUE"  
pciBridge4.virtualDev = "pcieRootPort"  
pciBridge4.functions = "8"  
  
...  
floppy0.present = "FALSE"  
svga.vramSize = "8388608"  
memSize = "1024" 1GB RAM  
sched.cpu.units = "mhz"  
sched.cpu.affinity = "all"  
sched.mem.affinity = "all"  
  
...  
scsi0.virtualDev = "lsilogic"  
scsi0.present = "TRUE"  
scsi0:0.deviceType = "scsi-hardDisk"  
scsi0:0.fileName = "CentOS7-Minimal64-1708-000002.vmdk"
```



```
...  
ethernet0.virtualDev = "vmxnet3"  
ethernet0.networkName = "PAULO"  
ethernet0.generatedAddress = "00:50:56:aa:aa:aa"  
  
...  
displayName = "CentOS7-Minimal64-1708"  
guestOS = "centos-64"  
  
...  
sched.cpu.min = "0"  
sched.cpu.shares = "normal"  
sched.mem.min = "0"  
sched.mem.minSize = "0"  
sched.mem.shares = "normal"  
numa.autosize  
cpu.maxPerVirtualNode = "1"
```

O VMM “cria” o hardware virtual a partir da “specfile” e apresenta-o ao Sistema de Operação hospedado

Uma infraestrutura de virtualização

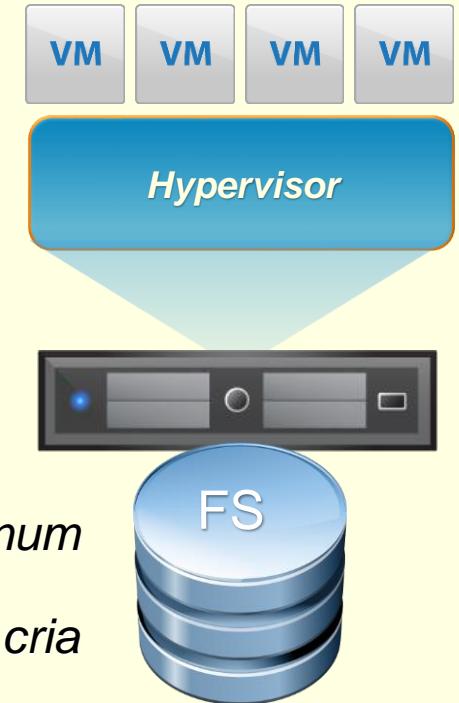
Com um só nó (servidor)...

O servidor tem um volume (disco) que armazena “tudo”, sob a forma de ficheiros

1. O software (hipervisor)
2. O catálogo de VMs existentes
3. A “specfile” e disco (virtual) de cada VM

Para executar uma VM,

1. Executa-se um comando ou clica-se no rato num GUI, apontando para a VM
2. O hipervisor lança um processo VMM; o VMM cria os recursos virtuais listados no “specfile”
3. A VM arranca executando a BIOS e depois “boota” do disco (o HDD é o ficheiro especificado na variável **scsi0:0.fileName** – slide anterior)



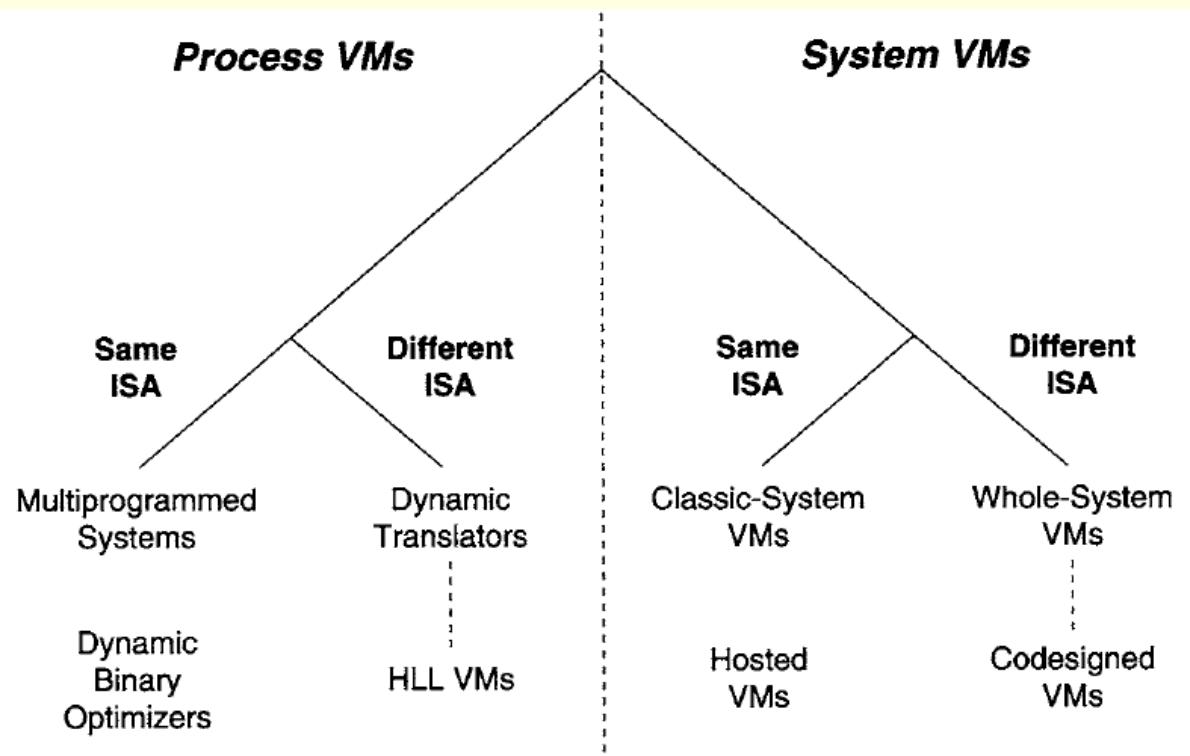
Virtualização e Armazenamento

- O disco de uma VM pode ser representado por um ficheiro, então, num SF moderno,
 - Um vDisk pode ser **thin provisioned**, i.e., começar por gastar pouco espaço e ir crescendo (recorda os ficheiros esparsos, disponíveis nos SFs modernos - ext3/4, xfs, NTFS, etc.)
 - Clonar um disco para ser usado noutra VM é tão simples como copiar o ficheiro
 - Podem usar-se técnicas de COW (recordar logging em SF) para criar snapshots que preservam o estado da VM num dado instante.
 - Usando snapshots podem criar-se linked-clones (o disco do clone aponta para um snapshot do original) e poupar imenso espaço...

Outras formas de virtualização

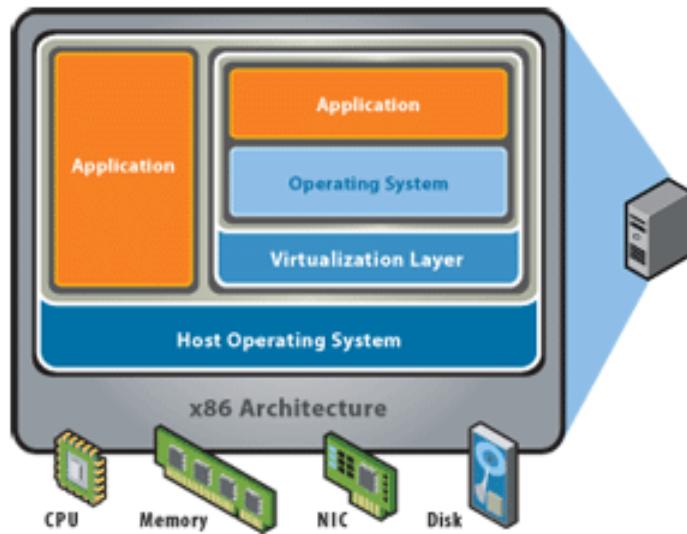
- É esta a única forma de virtualização?
 - Não, há outras...
 - Um SO é muitas vezes descrito como capaz de criar uma máquina virtual capaz de executar processos
 - A JVM é uma máquina virtual
 - Hoje, os containers são um “tópico quente” e estão relacionados com o tema “virtualização”
 - ...
- A definição que fizemos anteriormente (slide 6) corresponde à forma mais relevante no âmbito dos Centros de Dados – Classic-System VMs ou **Virtualização do Sistema Completo** (*full system virtualisation* - ver próximos slides)

Uma Taxonomia de VMs

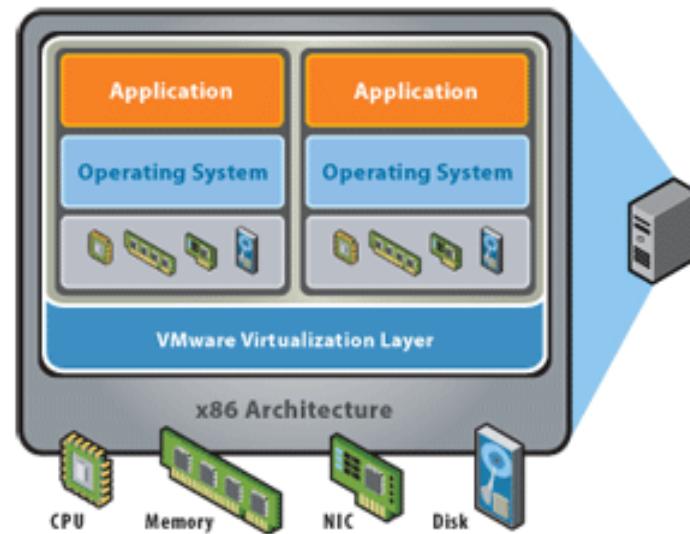


VMs de Sistema: Virtualização completa

Usado em: laptops, PCs



Usado em: servidores



VMMs hosted ou tipo II

VMware Workstation, Fusion,
Oracle VirtualBox

vs. VMMs nativas, bare metal, ou tipo I

VMware ESXi, Citrix Xen, KVM
Microsoft Hyper-V, IBM z/VM

Fundamentos de Sistemas de Operação

Unix *Windows NT* *Netware* *MacOS* *DOS/VIS* *Vax/VMS*
Linux *Solaris* *HP/UX* *AIX* *Mach*
Chorus

Demo: *DIvDC...*