MC714

Sistemas Distribuídos

1° semestre, 2017

Processos

Processos

- Processo: um programa em execução no sistema operacional.
  - Importante: gerenciamento e escalonamento.
- Em sistemas distribuídos:
  - Cliente-servidor: conveniente usar multi-threading.
  - Permitem clientes e servidores construídos de modo que comunicação e processamento se sobreponham.
- Virtualização / máquinas virtuais.
- Migração de processos / migração de código.

Processos e threads

- Granularidade oferecida pelos SOs
  - Processos
- Pode ser refinada
  - Threads.

- Facilita construção de aplicações e melhora no desempenho.

Processos e threads

- SO cria vários processadores virtuais
  - Cada um executa um programa
- Controle através de tabela de processos
  - Valores de registradores de CPU
  - Mapas de memória
  - Arquivos abertos
  - Privilégios
  - Etc.
- Processo: um programa em execução → programa sendo executado em um dos processadores virtuais do SO

Processos e threads

- SO toma conta para não haver interferência, intencional ou não, entre processos.
  - Compartilhamento concorrente de CPU (e outros recursos) é transparente.
  - SO requer suporte de hardware para essa separação.
- Criar processos
  - Criar espaço de endereço independente.
  - Inicializar segmentos de memória (zerar, copiar dados).
  - Pilha para dados temporários.
- Chaveamento entre processos:
  - Salvar contexto de CPU (registradores, contador de programa, ponteiro de pilha).
  - Modificar registradores do gerenciamento de memória.
  - Invalidar caches da TLB.

Processos e threads

- Thread: sua própria porção de código, independente de outras threads.
- Sem esforço para oferecer transparência de concorrência.
  - Controle guarda informação mínima necessária ao compartilhamento de CPU.
  - Contexto de thread = contexto de CPU + algumas informações para gerência de threads.
  - Ex.: monitorar exclusão mútua para não dar tempo de CPU a uma thread bloqueada.
- Proteger dados contra acesso inadequado é tarefa do desenvolvedor da aplicação multithread.

Processos e threads

- Monothread: chamada que bloqueia, bloqueia qualquer ação do processo.
- Multithread: pode-se separar o processo em threads que podem executar de forma independente.
  - Torna possível explorar paralelismo ao executar em multiprocessador; cada thread em uma CPU.
  - Pode executar também em um sistema monoprocessado, talvez leve mais tempo.
  - Cada vez mais importante com processadores multicore baratos.
- Aplicações grandes multiprocesso:
  - Comunicação entre processos demanda intervenção do núcleo em chamadas de sistema e chaveamento de contexto entre processos.
  - Com threads, minimiza overhead já que tudo acontece no espaço do usuário.

Threads

- Duas abordagens
  - Modo de usuário.
  - Núcleo ciente, com escalonamento.
- Modo de usuário (user thread)
  - Criar e terminar threads é mais barato
  - Alocar/liberar memória para pilha de threads
  - Chaveamento de contexto com poucas instruções
    - Troca de valores de registradores de CPU
    - Não precisa mudar mapas de memória, limpar TLB, etc.
    - Necessário para sincronização
- Núcleo (kernel thread)
  - Perde vantagens da thread, oferece melhor escalonamento quando bloqueios em threads acontecem.

Threads

- Solução intermediária: Lightweight processes (LWP).
- Processos que compartilham recursos
  - Espaço de endereçamento, páginas de memória física, sinalização e manipuladores de arquivos).
  - Evita troca de contexto.
- Pode ser visto como uma CPU virtual disponível para executar código ou chamadas de sistema.
- LWP roda sobre uma kernel thread.
- Fig. 54

LWP

- LWP + threads no espaço do usuário
- Criar, destruir e sincronizar threads é barato, sem intervenção do núcleo.
- Se processo tem LWPs suficientes, chamada bloqueante não suspenderá processo inteiro.
- Independentes de aplicação.
- Facilidades para multiprocessamento (LWP → processador).

Threads em SDs

- Threads atraentes
  - Manipular diversas comunicações lógicas
  - Permitir progresso da aplicação
- Cliente
  - Esconder atrasos de comunicação.
  - Iniciar comunicação e prosseguir com processamento independente.
  - Ex. browsers web.

Threads em SDs

- Servidor
  - Simplifica código.
  - Facilita desenvolvimento paralelo.

- Ex: servidor de arquivos
  - Com thread vs. sem threads
- Fig. 55

Virtualização

Virtualização

- Threads/processos:
  - Modo de fazer mais coisas ao mesmo tempo.
  - Concorrência - impressão de execução paralela em computador monoprocessado.
- “fingir maior capacidade”
  - Pode ser estendida a outros tipos de recursos.
  - Virtualização de recursos.

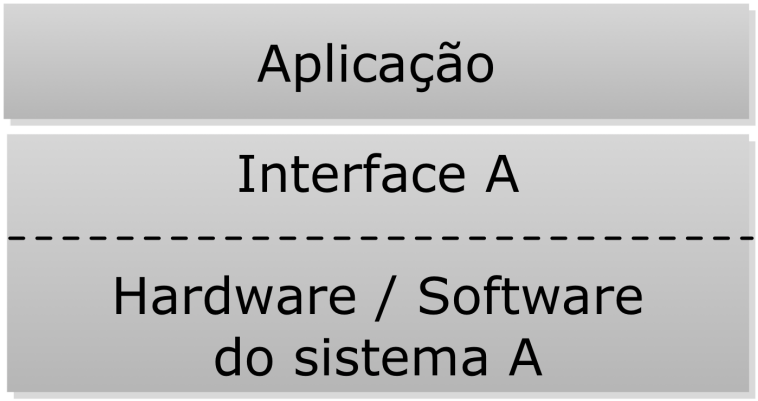
Virtualização

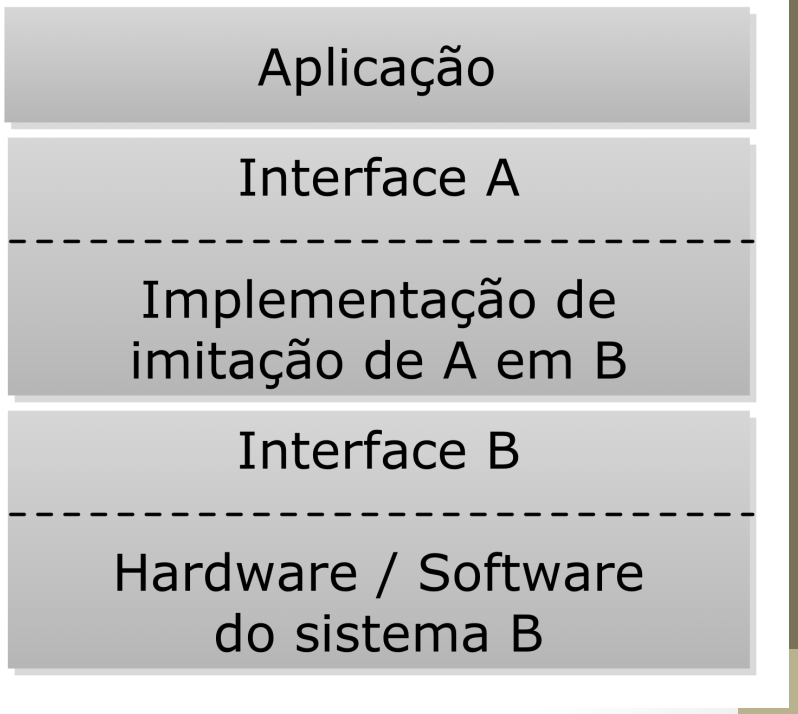
- Virtualização de recursos
- Utilizada há muito tempo.
- Interesse renovado
  - Sistemas com maior capacidade.
  - Sistemas distribuídos tornaram-se mais comuns e complexos.

Virtualização

- Sistema de computadores
  - Interface de programação para software de alto nível.
- Diferentes interfaces
  - Conjunto básico de instruções oferecido por uma CPU.
  - Conjunto de interfaces de programação de middlewares.
- Virtualização: estender ou substituir uma interface de modo a imitar o comportamento de um outro sistema.

Virtualização

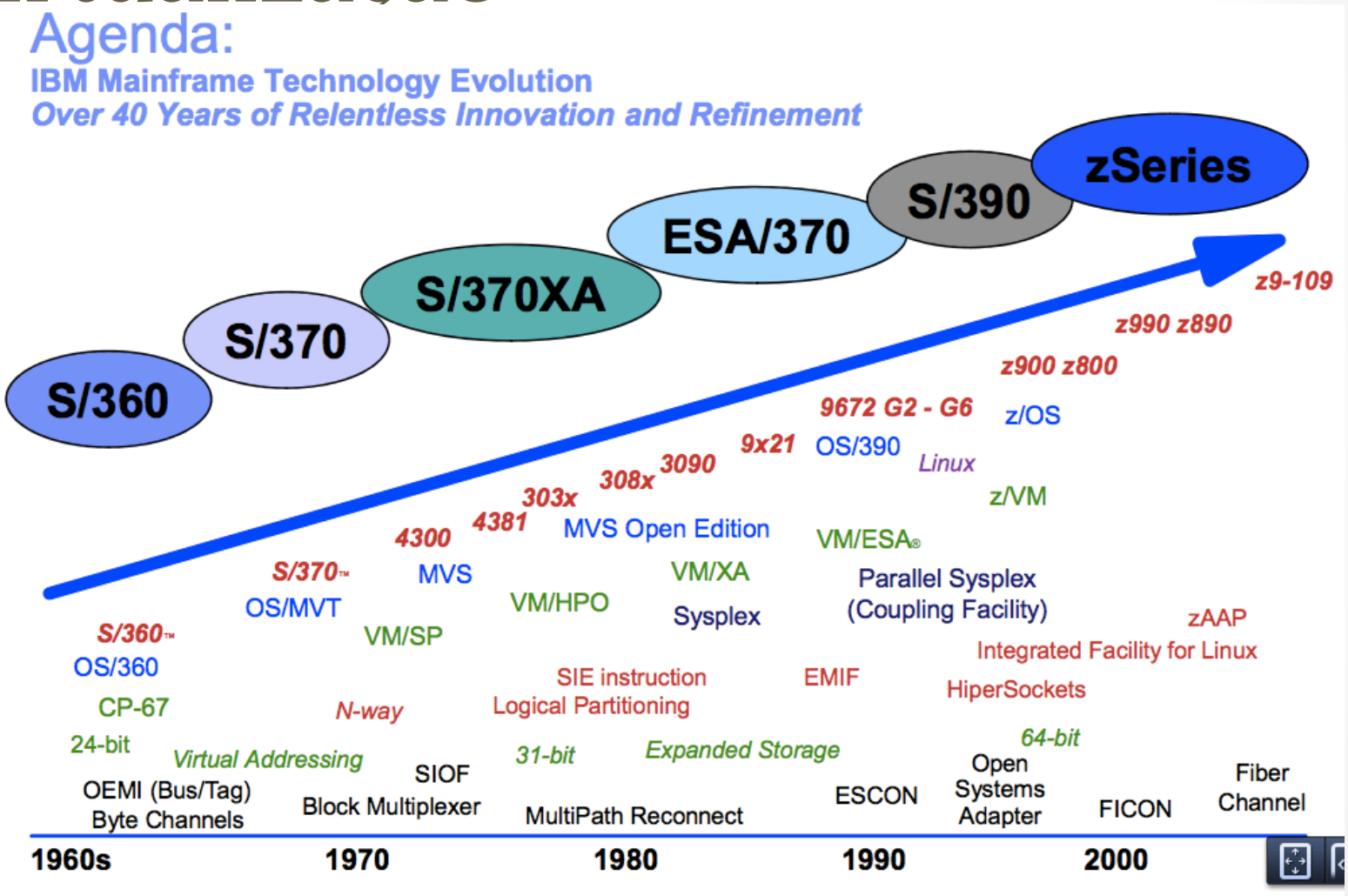




Virtualização

- Virtualização na década de 70:
  - Softwares em hardwares caros de mainframes.
- Onde software pode ser:
  - Aplicações.
  - Sistemas operacionais (SOs).
  - Aplicações + SO para os quais havia sido desenvolvido.
- Mainframes IBM 370 e sucessores.
  - Ofereciam máquina virtual para diferentes SOs.

Virtualizacão



Erich Amrehn, Jim Elliot. 45 (40)Years of Mainframe Virtualization: CP-67/CMS and VM/370 to z/VM. 2012.

Virtualização

- Hardware mais barato.
- Computadores mais potentes.
- Menor quantidade de SOs.
- Virtualização deixava de ser importante.

- Final da década de 90
  - Virtualização voltou a se tornar importante.

Virtualização

- Atualmente
  - Servidores mais baratos
  - Maior capacidade computacional.
- Custo total da posse inclui outras variáveis
  - Manutenção
  - Suporte e administração
  - Custos associados a brechas de segurança e falhas.
- Consolidação de servidores
  - Motivador importante para uso de virtualização.

Virtualização

- Máquina virtual em diferentes níveis de abstração
  - Processos individuais
  - Sistemas completos
- Máquinas virtuais
  - Uso flexível de hardware e isolamento de software.
  - Tradução de conjuntos de instruções.
- Variedade de arquiteturas de máquinas virtuais.
- Máquinas virtuais de processo e de sistema.

Virtualização - motivação

- Reduzir a quantidade de plataformas de hardware
  - Atende softwares com necessidades diferentes.
  - Cada aplicação executa em sua própria máquina virtual.
    - Incluindo bibliotecas e o sistema operacional.
- Portabilidade
- Flexibilidade
- Gerenciamento mais fácil de replicação.
  - Cópia dinâmica de servidores + ambiente.

Virtualização - motivação

- Consolidação de servidores.
- Consolidação de aplicações.
- *Sandboxing.*
- Múltiplos ambientes de execução.
- Hardware virtual.
- Executar múltiplos SOs simultaneamente.

Virtualização - motivação

- Depuração
  - Depuração de aplicações de usuário sem preocupação com problemas de interrupção de outras aplicações/serviços.
- Migração
  - Facilita migração de software.
- Administração
  - Empacotar aplicações junto com ambiente de execução.
- Teste
  - Produção de cenários de teste arbitrários difíceis de produzir na prática.

Abstração e virtualização

- Abstração com interfaces bem definidas ajuda no desenvolvimento e manutenção.
- Escondem detalhes de implementação

- Ex.: SO abstrai sistema de arquivos e endereçamento.
  - Disco aparece como um conjunto de arquivos de tamanhos variados, escondendo setores e trilhas.
  - Programadores manipulam arquivos pelos nomes.

Abstração e virtualização

- Arquitetura do conjunto de instruções (ISA)
  - exemplo das vantagens de interfaces bem definidas.
- IA-32 (x86)
  - Intel e AMD implementam.
  - Softwares são desenvolvidos para esse conjunto de instruções
  - Devem compilar e executar corretamente em qualquer computador com processador IA-32.

Abstração e virtualização

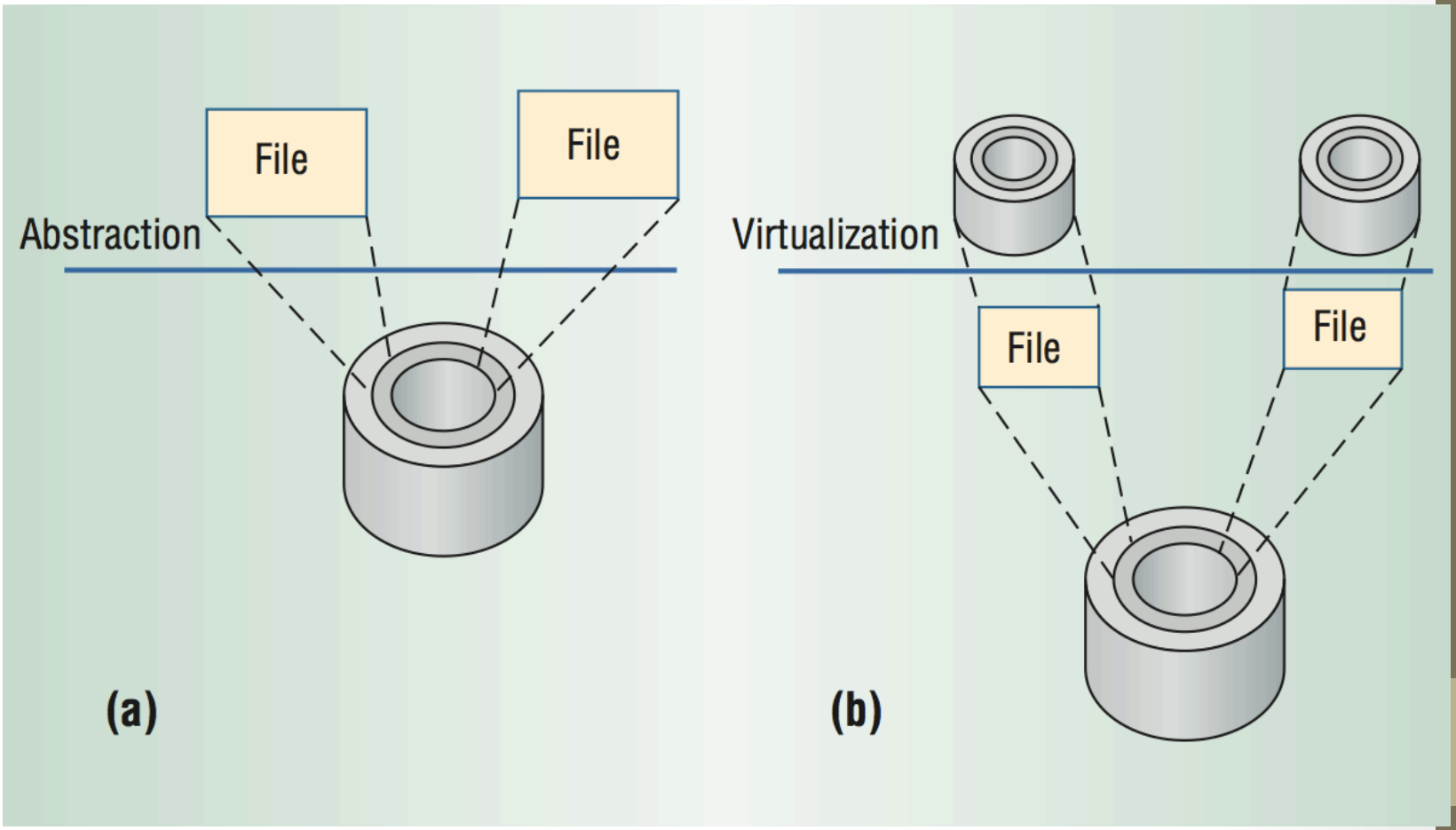
- Abstração possui limitações.
  - Binários compilados estão amarrados à arquitetura-alvo.
  - Impede interoperabilidade.
  - Importante em computadores heterogêneos conectados.

- Mapeamento para sistema real potencialmente diferente pode contornar esse problema.
  - Sistema virtual apresentado como outro sistema, ou múltiplos sistemas.

Abstração e virtualização

- Virtualização
  - Não necessariamente simplificar ou esconder detalhes.
- Ex.: virtualização de disco
  - 1 disco grande → dois virtuais menores, cada um com seu conjunto de trilhas e setores.
- Software de virtualização utiliza a abstração de arquivos como um passo intermediário para o mapeamento entre disco real e virtual.
- Escrita no disco virtual → escrita de arquivo no disco real.
- Nível de detalhes virtual X real
  - Não há abstração.

Abstração e virtualização



James E. Smith, Ravi Nair. The Architecture of Virtual Machines. IEEE Computer.

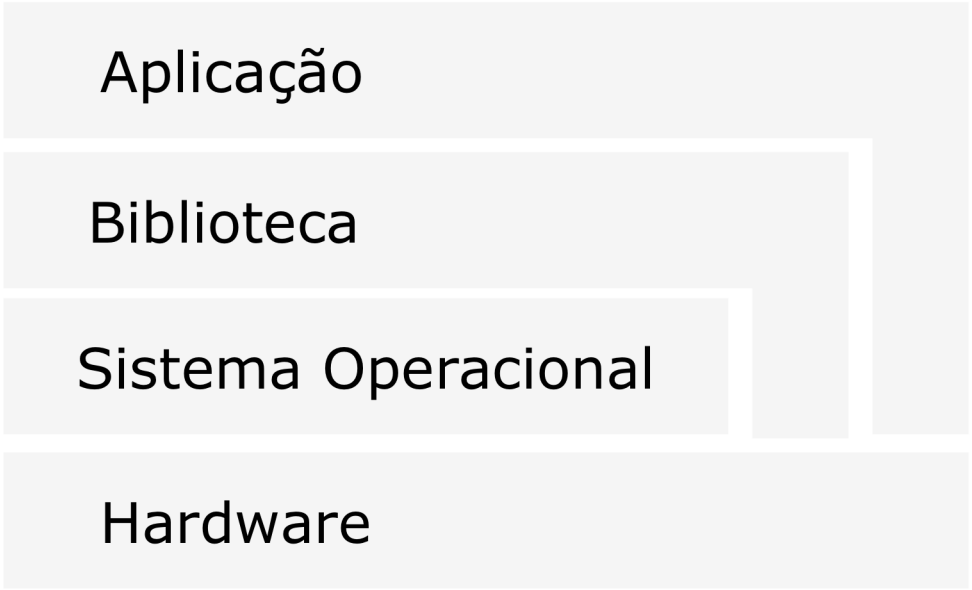
Arquiteturas de Máquinas Virtuais

- Aplicável a máquinas inteiras e outros componentes.
- Para discutir máquinas virtuais (virtual machines – VMs) é preciso entender arquitetura de sistemas de forma geral.
- Arquitetura: especificação formal de interfaces do sistema.
- Implementação: “personificação” de uma arquitetura.

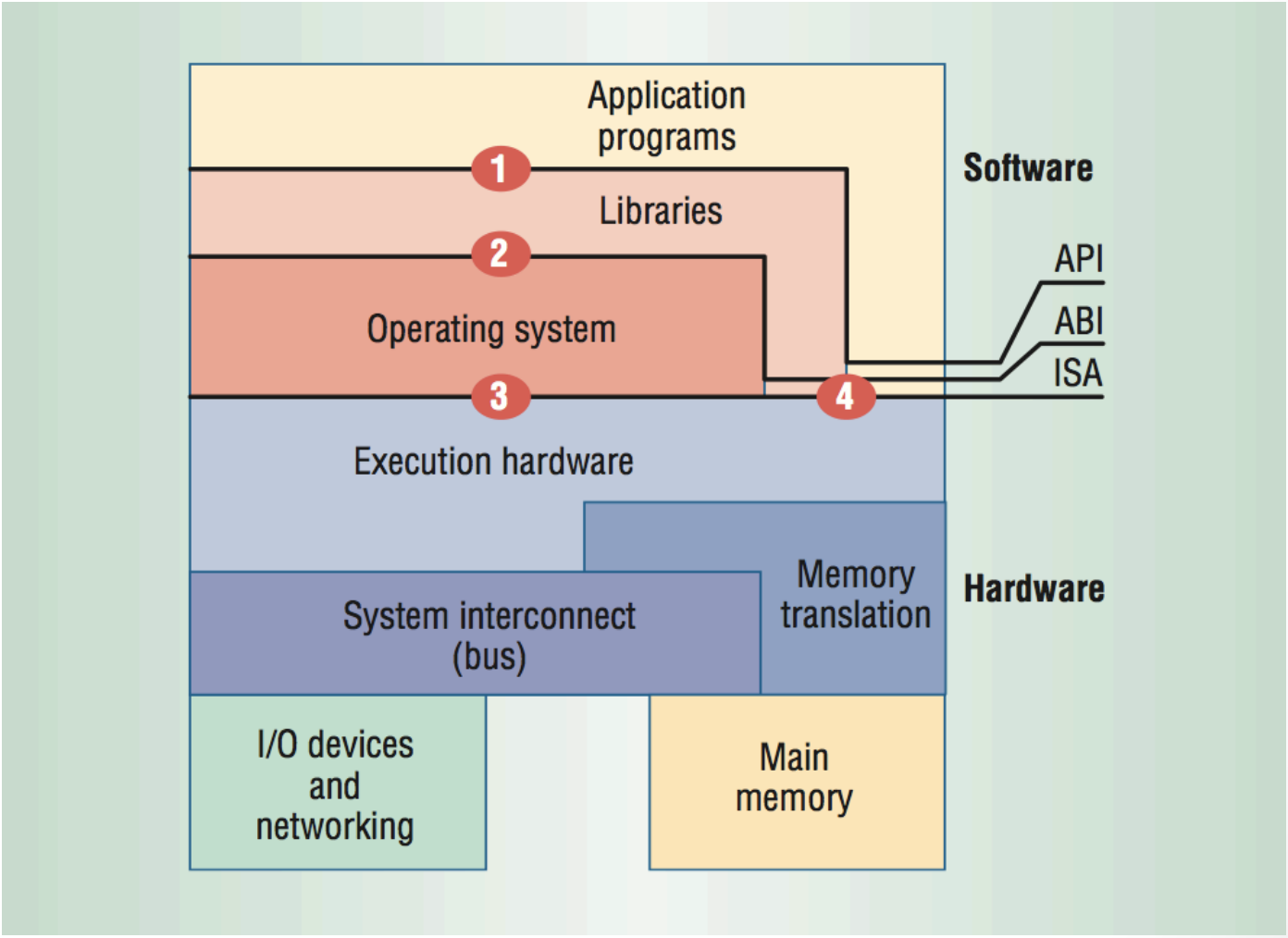
Arquiteturas de Máquinas Virtuais

- Em geral, sistemas de computadores oferecem 4 tipos diferentes de interfaces em 4 níveis diferentes:
  - 1: instruções de máquina que podem ser invocadas por qualquer pograma.
  - 2: instruções de máquina que podem ser invocadas somente por programas privilegiados, como o sistema operacional.
  - 3: Chamadas de sistema: oferecidas por um sistema operacional.
  - 4: Chamadas de biblioteca: conjunto conhecido como interface de programação de aplicativo (API).

Arquiteturas de Máquinas Virtuais

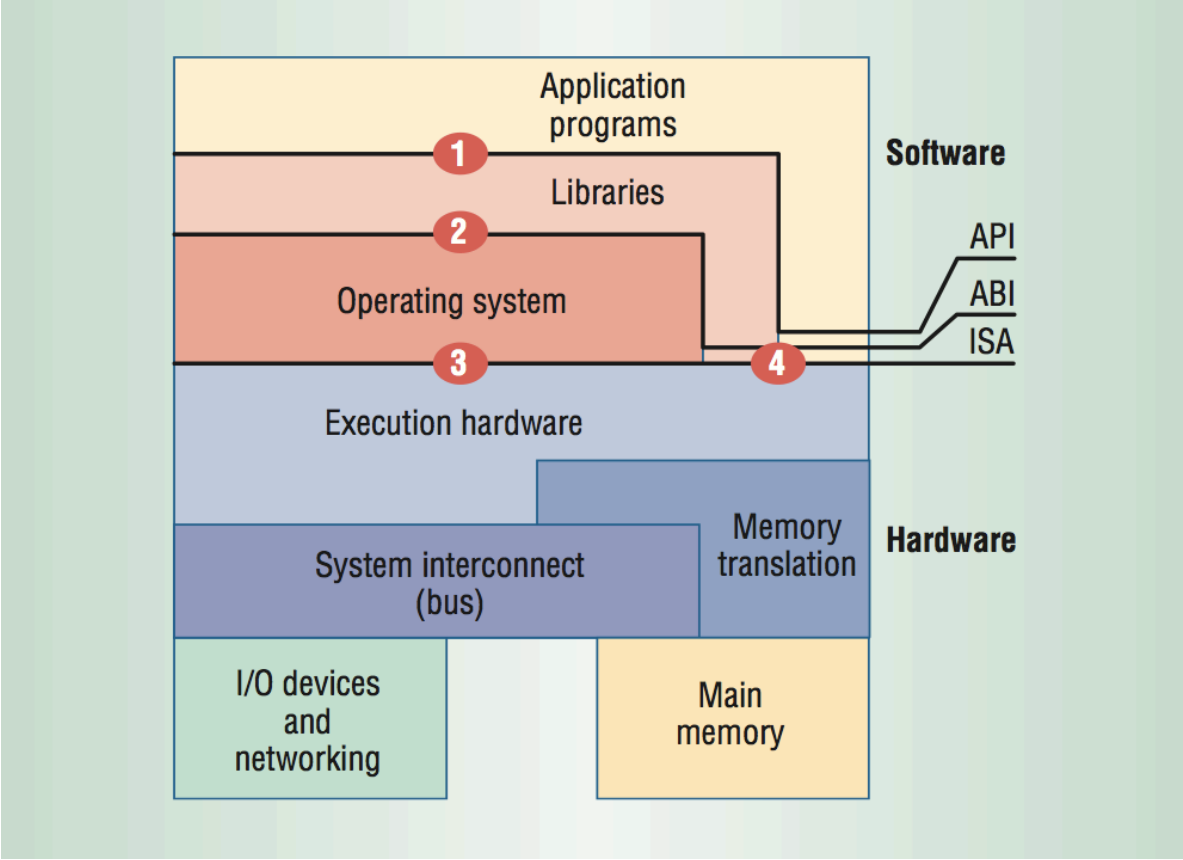


Arquiteturas de Máquinas Virtuais



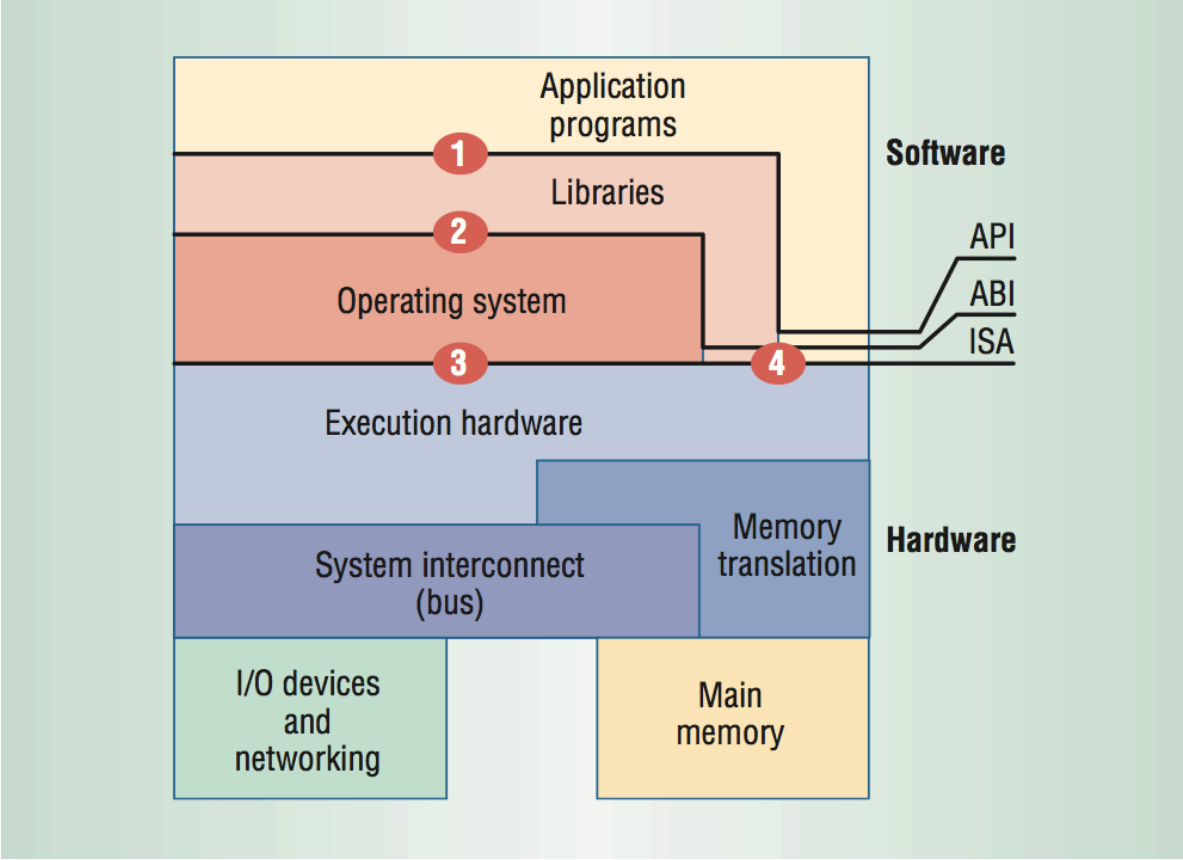
Arquiteturas de Máquinas Virtuais

- Instruction Set Architecture (ISA)



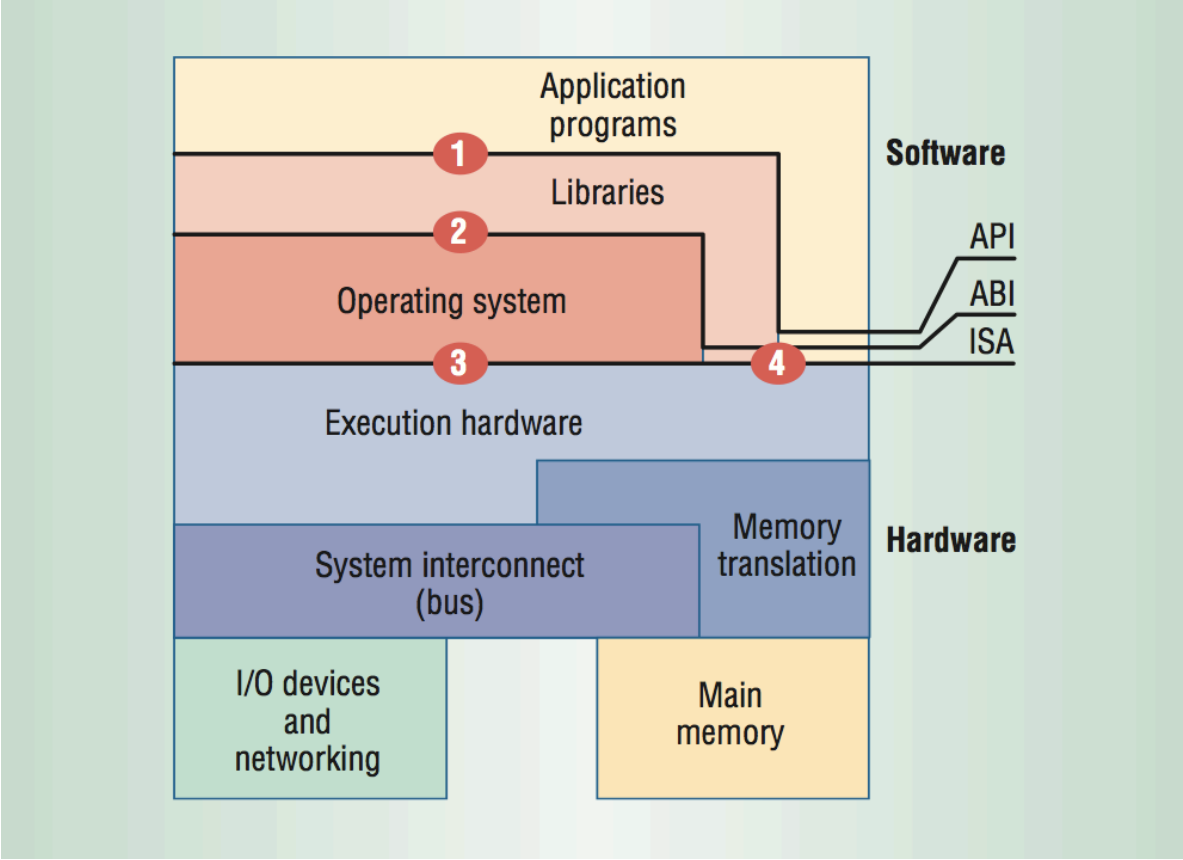
Arquiteturas de Máquinas Virtuais

- Application Binary Interface (IBA)



Arquiteturas de Máquinas Virtuais

- Application Programming Interface (API)



Arquiteturas de Máquinas Virtuais

- *Máquina:*
  - Espaço lógico de memória atribuído ao processo.
  - Instruções de nível de usuário.
  - Registradores que permitem a execução do código do processo.
- E/S é visível somente através de chamadas de sistema (SO).
- ABI define uma máquina como vista pelos processos.

Arquiteturas de Máquinas Virtuais

- Do ponto de vista do SO e das aplicações:
  - Sistema roda sobre uma ***máquina subjacente***
  - É capaz de suportar múltiplos processos simultaneamente.
- Processos compartilham dispositivos de E/S.
- Sistema "sobrevive" às idas e vindas dos processos.
- Aloca memória real e recursos de E/S aos processos
  - Controla acesso.
- Da perspectiva do sistema: hardware define a máquina
  - ISA fornece interface entre sistema e a máquina.

Arquiteturas de Máquinas Virtuais

- Também há ponto de vista diferente para máquinas virtuais.
- Máquina virtual de *processo*
  - Plataforma que executa apenas um processo.
  - Existe somente para suportar o processo.
  - Criado quando o processo é criado.
  - Termina quando processo termina.
- Máquina virtual de *sistema*
  - Fornece um ambiente de sistema completo e persistente.
  - Suporta um sistema operacional e seu conjunto de processos de usuário.
  - Fornece ao sistema operacional convidado acesso a recursos de hardware virtuais (rede, E/S etc.)

Arquiteturas de Máquinas Virtuais

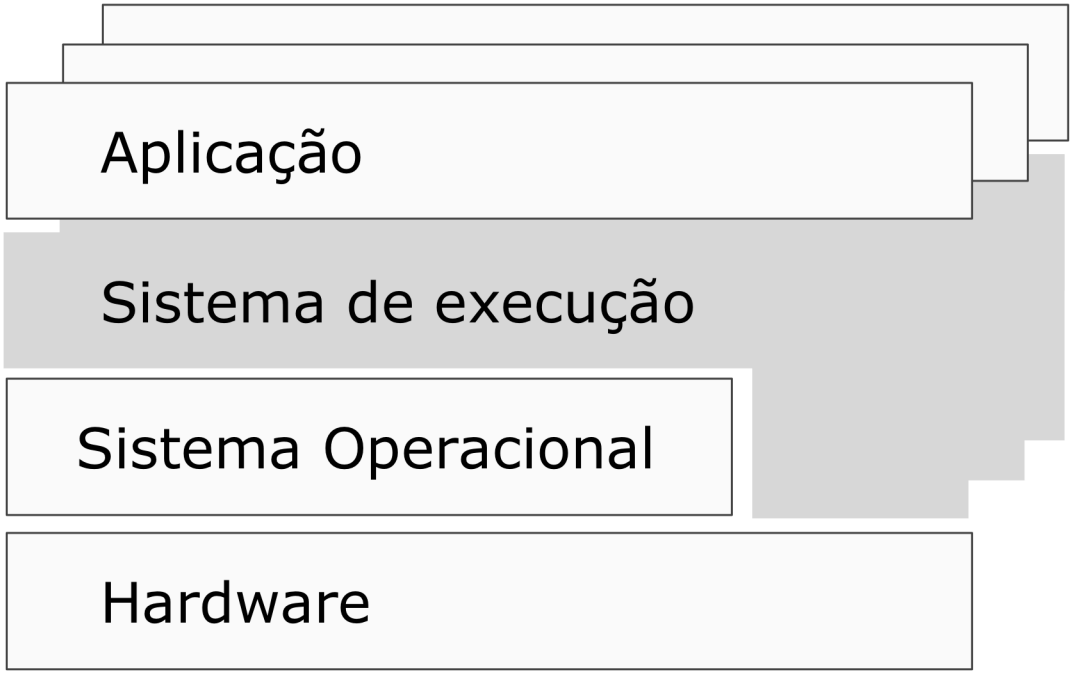
- *Convidado:* processo ou sistema em uma VM.
- *Hospedeiro:* plataforma que suporta a VM.
- Virtualização pode ocorrer de dois modos.
  - Virtualização de processo, através de *runtime software.*
  - Virtualização de máquina, através de *virtual machine monitor.*

Arquiteturas de Máquinas Virtuais

- Máquina virtual de processo
  - Sistema de execução com conjunto de instruções abstrato para executar aplicações.
  - Instruções interpretadas (p. ex. Java).
  - Emulação (Wine) – necessário imitar comportamento de chamadas de sistema (não trivial).
  - Chamada de máquina virtual de processo (Smith e Nair) / runtime software.

Arquiteturas de Máquinas Virtuais

Máquina virtual de processo

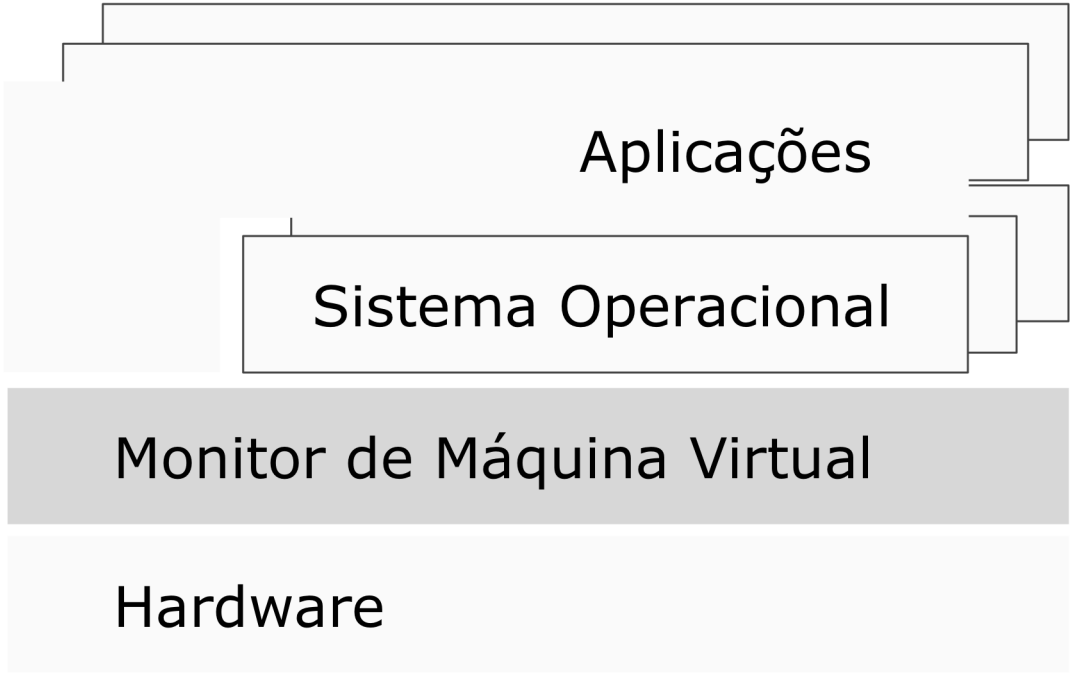


Arquiteturas de Máquinas Virtuais

- Máquina virtual de sistema
  - Protege completamente o hardware original.
  - Interface: conjunto de instruções completo do mesmo (ou de outro) hardware.
  - Pode ser oferecida simultaneamente a programas diferentes.
  - Vários sistemas operacionais executando independente e concorrentemente na mesma plataforma.
  - Camada chamada de Virtual Machine Monitor – VMM – ou hypervisor.
  - Exemplos de VMMs: VMWare, Xen, KVM.

Arquiteturas de Máquinas Virtuais

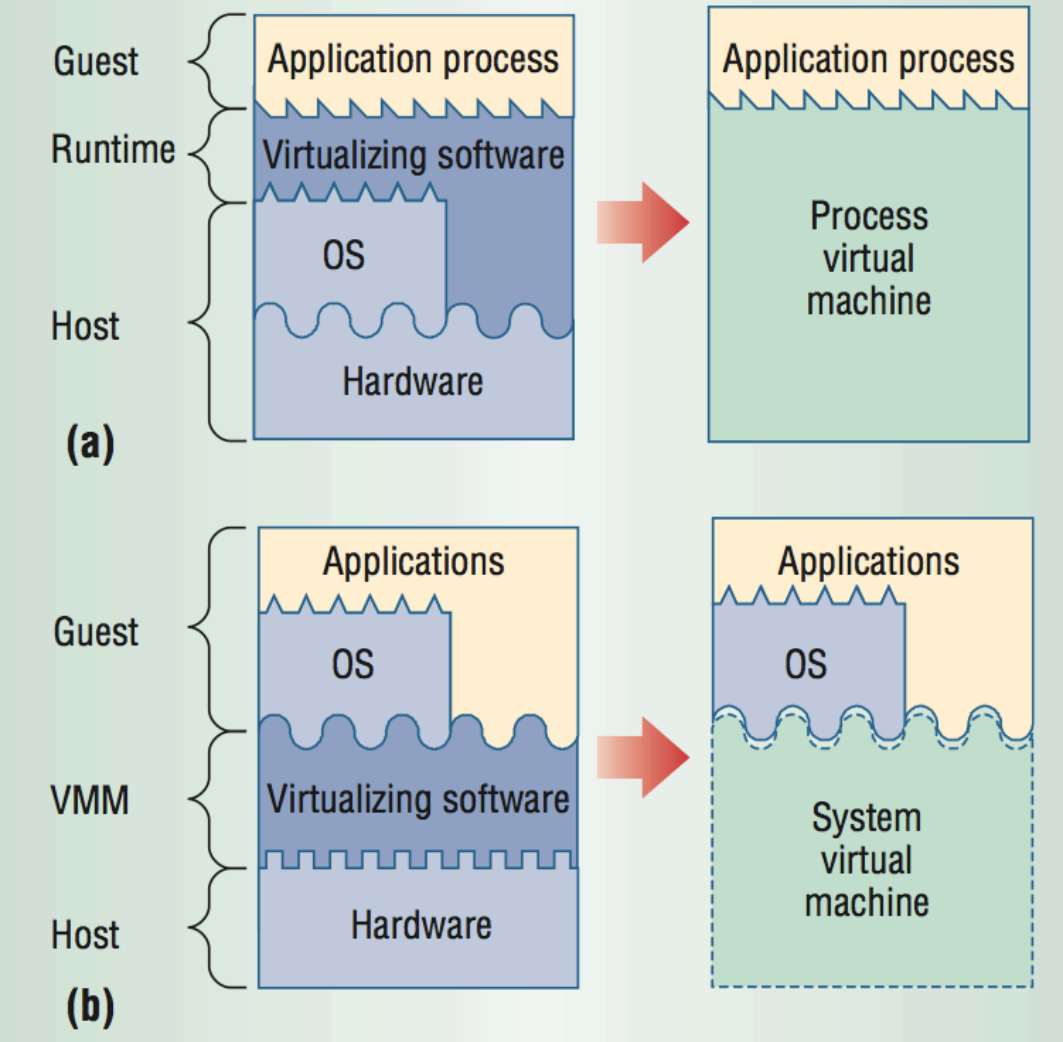
- Máquina virtual de sistema



Arquiteturas de Máquinas Virtuais

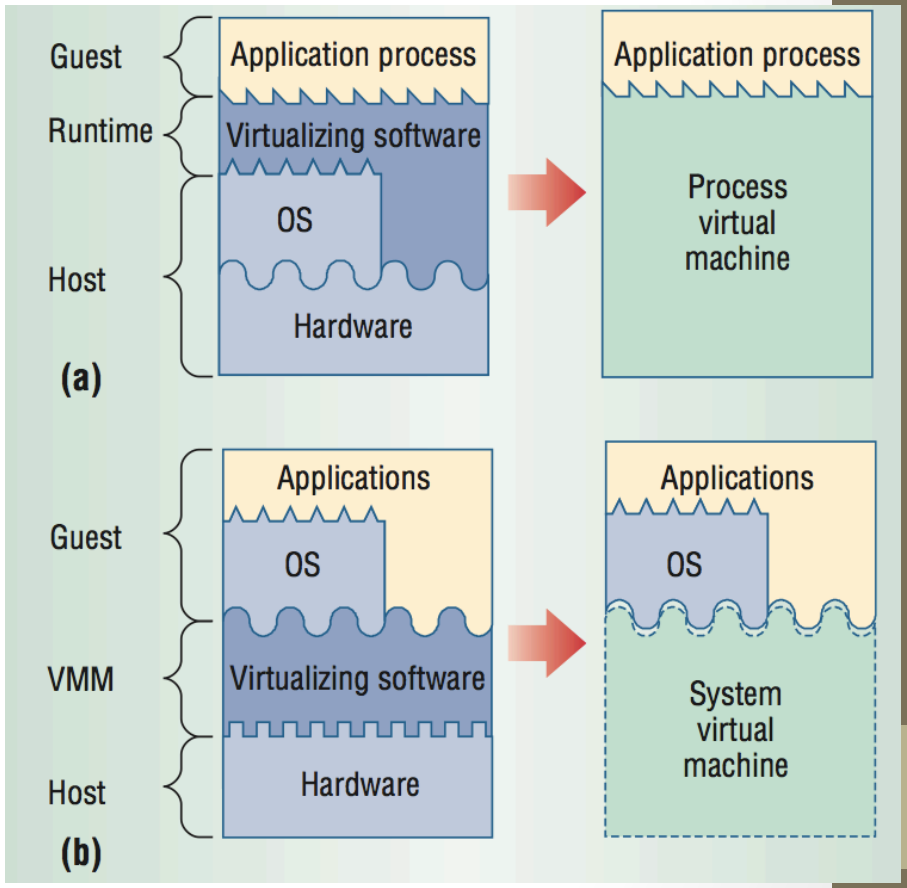
- Tipos de monitores de máquina virtual (VMMs)
  - Tipo 1: nativa (bare metal): executa sobre uma interface direta com hardware.
  - Tipo 2: hospedada (*hosted*): executa sobre um sistema operacional em uma máquina hospedeira.

Arquiteturas de Máquinas Virtuais



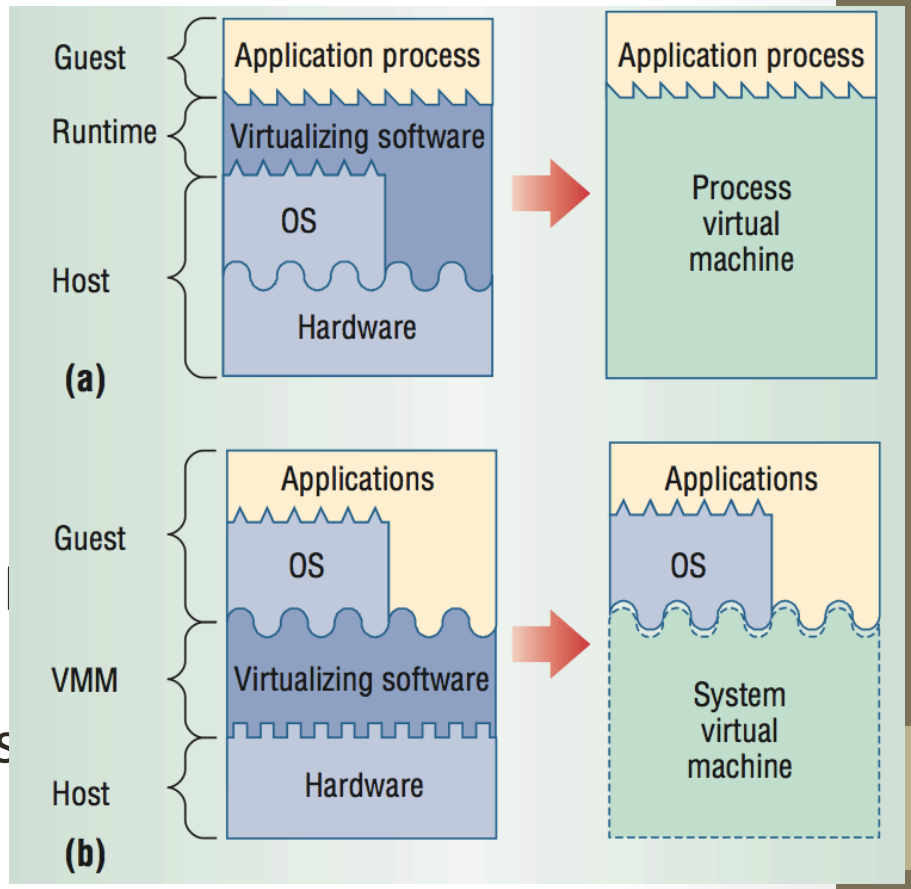
Arquiteturas de Máquinas Virtuais

- VM de processo: software de virtualização está no nível de ABI ou API.
- Emula instruções de nível de usuário e chamadas de sistema.



Arquiteturas de Máquinas Virtuais

- VM de sistema: software de virtualização está entre o hardware e o software convidado.
- Se mostra como ISA potencialmente diferente do hospedeiro.
- VMM muitas vezes tem o papel de fornecer recursos de hardware virtualizados ao invés de tradução de ISA

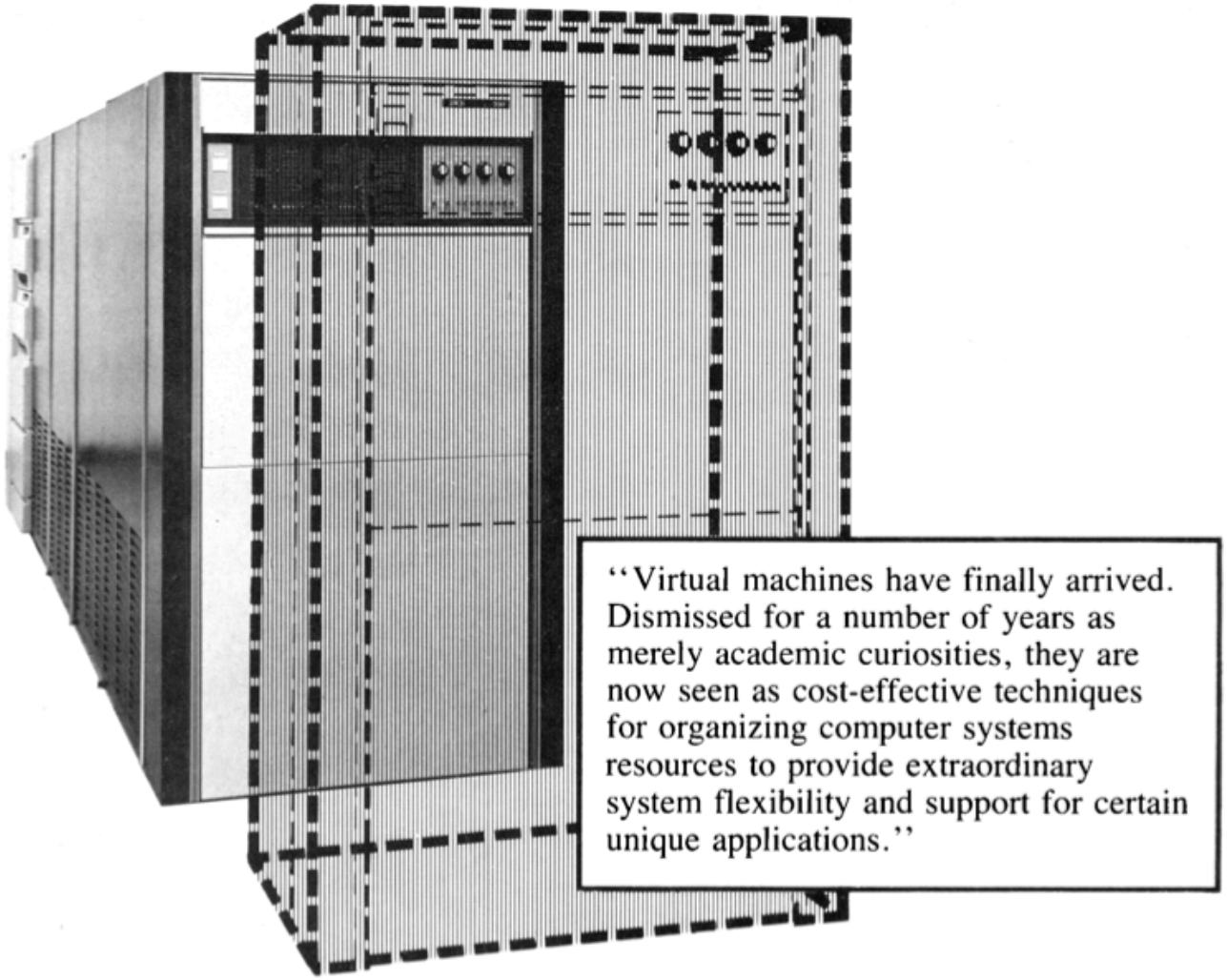


Arquiteturas de Máquinas Virtuais

- Importante para confiabilidade e segurança.
- Isolamento de uma aplicação completa e seu ambiente.
  - Falha não afeta outras VMs.
- Melhor portabilidade.
  - Desacopla hardware e software.
  - Permite mover ambiente completo.
- Máquinas paralelas
  - Permite consolidação de servidores.
  - Maximizar utilização.

Alguns conceitos em máquinas virtuais

Máquinas virtuais



Máquinas virtuais

- Simulação instrução por instrução: conhecida há décadas.
  - Ex.: aplicação para um computador X cujo hardware ainda está em desenvolvimento.
  - Simulador para X (processador, memória, periféricos) que rode em máquina de propósito geral G.
  - Programas que rodam em G poderão rodar em X com mesmos resultados.
  - Espaço de memória simulado, dispositivos simulados, executar instruções na máquina simulada.
  - Camada que filtra e protege os recursos da máquina G.

Máquinas virtuais

- Múltiplos programas:
  - Múltiplas cópias do simulador.
  - Simulador capaz de dividir o tempo entre aplicações.
  - Ilusão de múltiplas cópias da interface de hardware-software da máquina X em G.

Máquinas virtuais

- X e G arbitrários
  - Software de simulação pode ser muito complexo
    - Desempenho impraticável.
    - Mais utilizado para desenvolvimento de software.
- X e G idênticos
  - Muitas cópias da interface hardware-software de G em G.
  - Cada usuário com sua cópia privada da máquina G.
  - Escolha do SO para rodar em sua máquina privada.
  - Desenvolver/depurar seu próprio SO.
  - Simuladores não interferem um no outro.
  - *Slowdown* menor que para X diferente de G.

Máquinas virtuais

- Surgimento dos VMMs
  - Necessidade de simuladores mais eficientes de múltiplas cópias de uma máquina sobre seu próprio hardware.
- Parte do software para máquinas simuladas roda sobre o hardware, sem interpretação de software.

- Chamados de Virtual Machine Systems.
- Máquinas simuladas: máquinas virtuais (VMs).
- Software simulador: virtual machine monitor (VMM).

Máquinas virtuais

- VMM: interface única → ilusão de muitas máquinas.
- Cada interface (VM) é uma réplica eficiente do sistema de computação original.
  - Todas as instruções de processador (privilegiadas ou não).
  - Todos os recursos dos sistema (memória e E/S).
- VMs em paralelo → diversos SOs (núcleos privilegiados) concorrentemente.
- VMs fornecem réplicas isoladas de um ambiente em um sistema de computação.

Máquinas virtuais

- Recursos extras (CPU, memória) são usados pelo VMM.
  - Potencial queda na vazão do sistema.
- Manter estado do processador virtual.
  - Integridade de todos os registradores visíveis, bits de estado e locais de memória reservada (controle de interrupção) devem ser preservados.
  - Captura e simulação de instruções privilegiadas.
  - Suporte a paginação em máquinas virtuais.

Máquinas virtuais

- Podem ser utilizadas para manter sistemas antigos
  - Novos sistemas podem ser testados
  - Programas podem ser convertidos.
- Atualizar/adicionar novos dispositivos sem alterar SO da máquina virtual
  - VM já suporta o dispositivo virtualizado.
  - Usuário tira vantagem do dispositivo atualizado sem necessidade de alteração de software.

Máquinas virtuais

- Teste de softwares de rede.
- Confiabilidade de software através de isolamento.
  - VMM é provavelmente correta: pequena e verificável.
- Segurança de dados.

Máquinas virtuais

- VMM
  - Camada software/hardware físico programável,
  - Transparente ao software acima
  - Usa eficientemente o hardware abaixo dela.

- De forma similar:
  - Virtualização de rede
  - Virtualização de armazenamento
  - Fornecem capacidade de multiplexar, em um único recurso físico, vários sistemas virtuais isolados uns dos outros.

Máquinas virtuais de sistema

Máquinas virtuais de sistema

- VM
  - Fornece ambiente completo de sistema
- VMM
  - 1 plataforma de hardware.
  - Múltiplas VMs → múltiplos ambientes de sistema operacional independentes simultaneamente.
- Lembrete: isolamento entre sistemas concorrentes no mesmo hardware.
  - Característica importante de máquinas virtuais.
  - Sem interferência em caso de falha.

Máquinas virtuais de sistema

- VMM fornece, primordialmente, replicação de plataforma.
- Problema central: dividir recursos de hardware limitados entre múltiplos sistemas operacionais convidados.
- VMM tem acesso e gerencia todos os recursos de hardware.

Máquinas virtuais de sistema

- SO convidado e suas aplicações são gerenciadas sob controle (escondido) do VMM.
- SO realiza instrução privilegiada ou acesso a recurso: VMM intercepta a operação, realiza verificações, e a realiza em nome do SO convidado.
- SO convidado não é ciente dessa camada.

Máquinas virtuais de sistema

- Para usuário, sistemas de VM fornecem essencialmente a mesma funcionalidade
  - Diferem na forma de implementação.
- Sistema de VMs clássico
  - VMM sobre o hardware.
  - Modo de privilégio mais alto.
  - Sistemas convidados: privilégios reduzidos
    - Permite a interceptação pela VMM.
    - Ações de SO convidado: seriam interação direta com o hardware; são tratadas pela VMM.

Máquinas virtuais de sistema

- Hosted VMs:
  - Software de virtualização roda sobre um sistema operacional hospedeiro.
  - Vantagem: usuário instala VMM como um software típico.
  - Software de virtualização pode se apoiar no sistema operacional hospedeiro para utilizar drivers de dispositivos e outros serviços de nível mais baixo.
  - VMWare GSX Server

Máquinas virtuais de sistema

- Paravirtualização:
  - Modificações no SO hospedeiro.
  - Interface para um sistema similar mas não idêntico ao hardware nativo subjacente.
  - Interface de paravirtualização especialmente projetada para contornar características que tornam difícil a virtualização do ISA subjacente.
  - Ganho de desempenho; menor portabilidade e compatibilidade.
  - Ex.: Xen.

Máquinas virtuais de sistema

- Whole-system VMs:
  - Hospedeiro e sistema convidado podem não utilizar mesmo ISA (Ex. Windows / Power-PC).
  - Whole-system VMs virtualizam todo o software, incluindo SO e aplicações.
  - ISA diferentes: necessário emular códigos das aplicações e do SO.
  - Virtual PC.

Máquinas virtuais de sistema

- Multiprocessor virtualization
  - Hospedeiro é máquina grande e multiprocessada.
  - Particionar em sistemas menores multiprocessados
    - Distribui os recursos de hardware
  - Particionamento físico: recursos físicos separados para cada sistema virtualizado.
    - Alto grau de isolamento.
  - Particionamento lógico: hardware é multiplexado no tempo entre as diferentes partições.
    - Melhora utilização dos recursos.
    - Perde-se benefícios de isolamento de hardware.

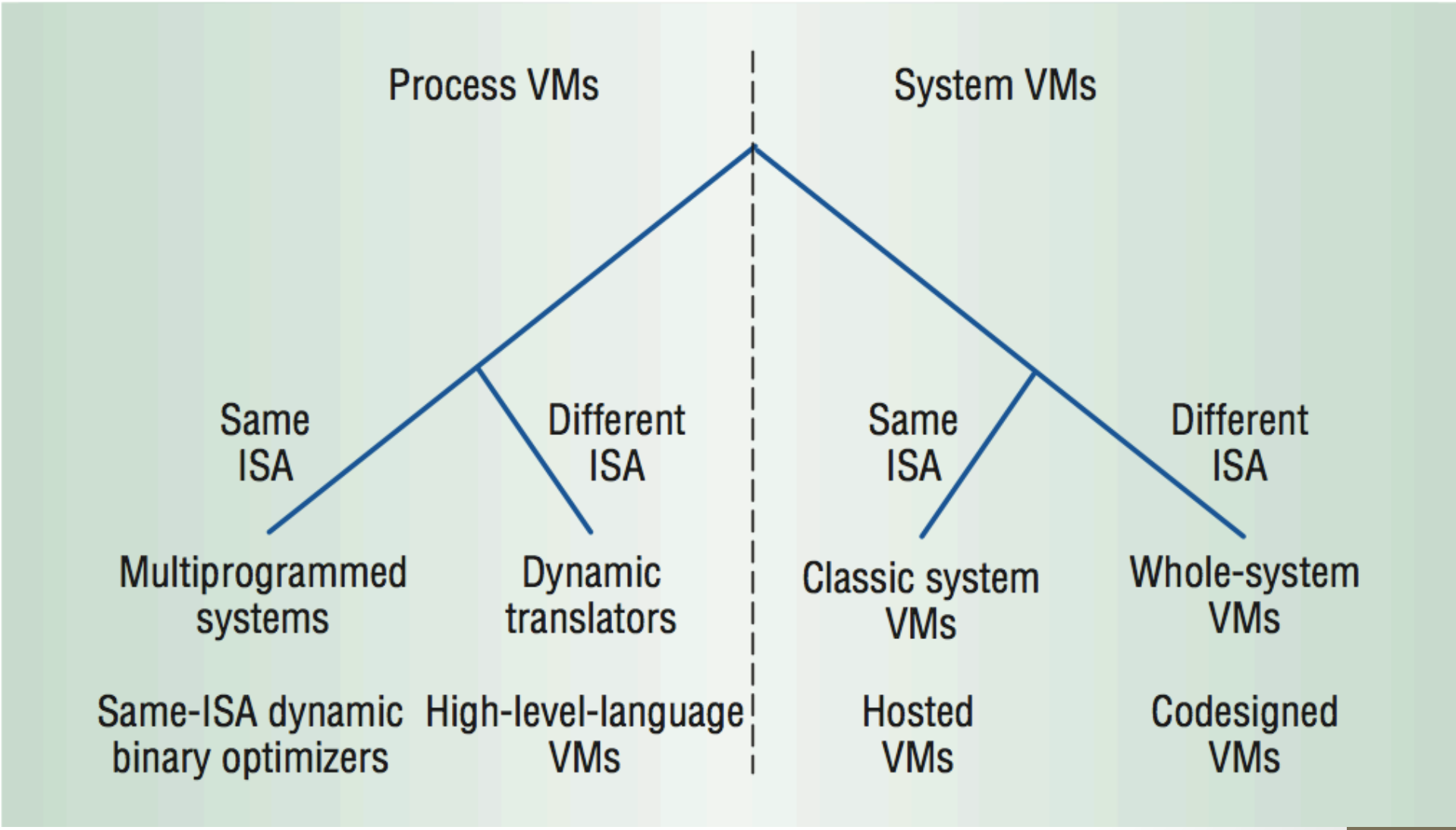
Máquinas virtuais de sistema

- Codesigned VMs
  - Implementam ISA novo e proprietário focado em melhoria de desempenho e eficiência energética.
  - ISA do hospedeiro: novo ou extensão de ISA existente.
  - Não possui aplicações nativas.
  - VMM tem propósito de emular ISA do software convidado.
  - VMM reside em região de memória oculta dos softwares convencionais.
  - Inclui tradutor binário que converte instruções do convidado em sequências otimizadas de instruções do ISA do hospedeiro.

Máquinas virtuais de sistema

- Codesigned VM: Transmeta Crusoe
  - Hardware: VLIW.
  - Convidado: Intel IA-32
  - Economia de energia.

Taxonomia


Process VMs
System VMs
Same ISA
Different ISA
Same ISA
Different ISA
Multiprogrammed systems
Dynamic translators
Classic system VMs
Whole-system VMs
Same-ISA dynamic binary optimizers
High-level-language VMs
Hosted VMs
Codesigned VMs