

Virtualização

Computação em Nuvem

Pontifícia Universidade Católica de Campinas

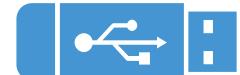
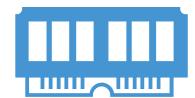
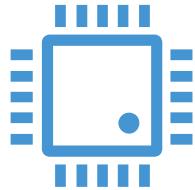
Prof. Dr. Denis M. L. Martins

Objetivos de Aprendizado

- Compreender os fundamentos de virtualização e sua evolução.
- Diferenciar tipos de hypervisors e ambientes de execução (VMs vs. Containers).
- Analisar o papel da virtualização em modelos de serviço Cloud (IaaS, PaaS, SaaS).
- Discutir desafios práticos: performance, segurança, migração e gerenciamento de recursos

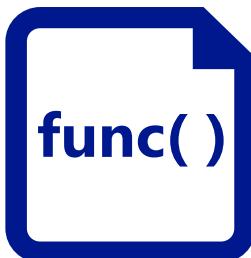
Motivação

what we have



hardware
(devices)

have-to-want



software

what we want



features

Fonte da Imagem: [OS OER](#).

Motivação

Imagine uma organização que necessita de servidores para atender três demandas distintas:

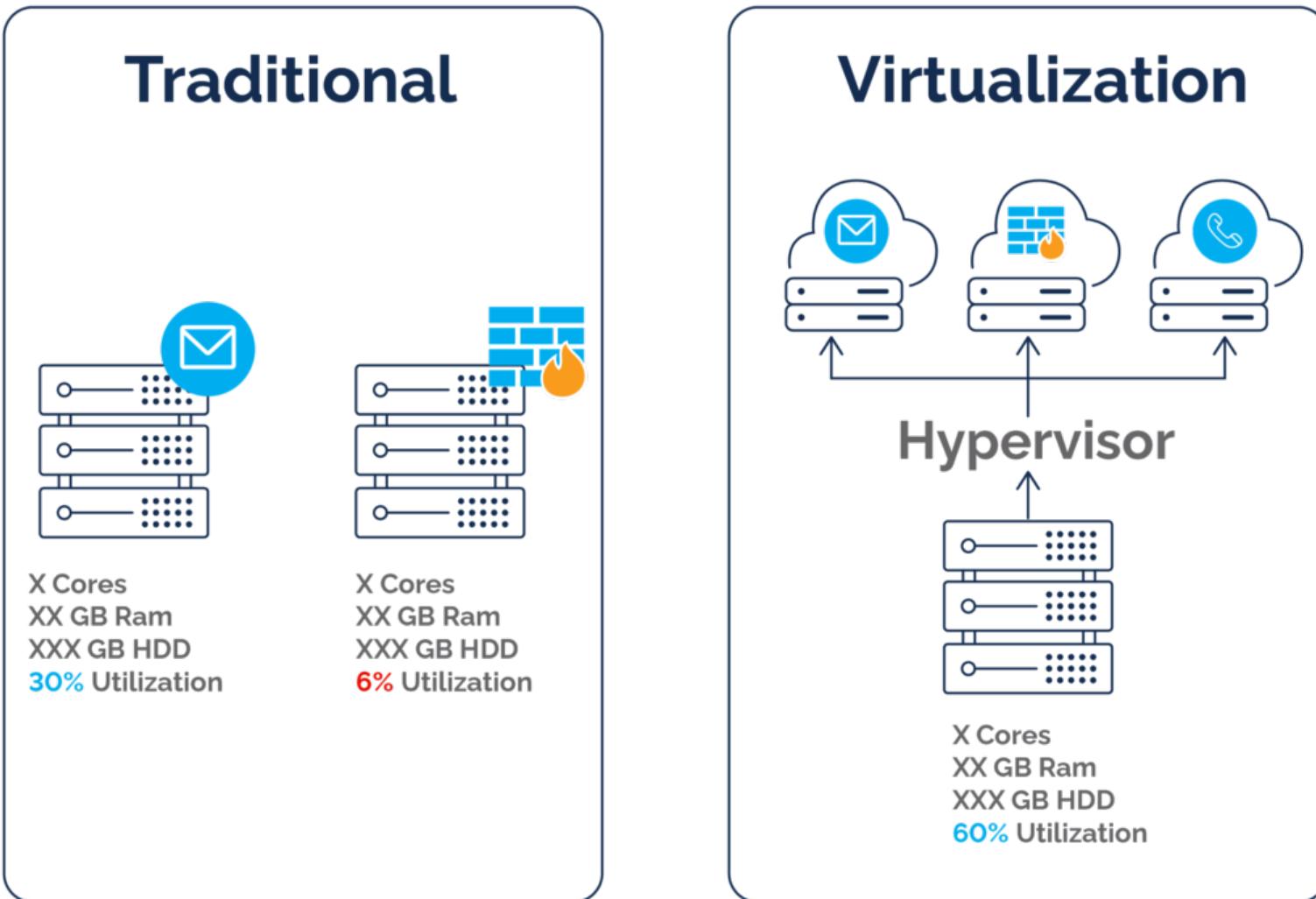
- **Armazenamento seguro de e-mail corporativo.**
- **Execução de um portal voltado ao cliente.**
- **Operação de aplicações internas de negócio.**

Para cumprir esses requisitos, a empresa decide alocar **três servidores físicos dedicados**, um por cada aplicação.

Função	Sistema Operacional	Requisito de Capacidade
E-mail corporativo	Windows	Maior capacidade de armazenamento
Portal para clientes	Linux	Alto poder de processamento para suportar grande fluxo de tráfego do site
Aplicações internas	iOS (ambiente de execução)	Memória RAM adicional

Virtualização

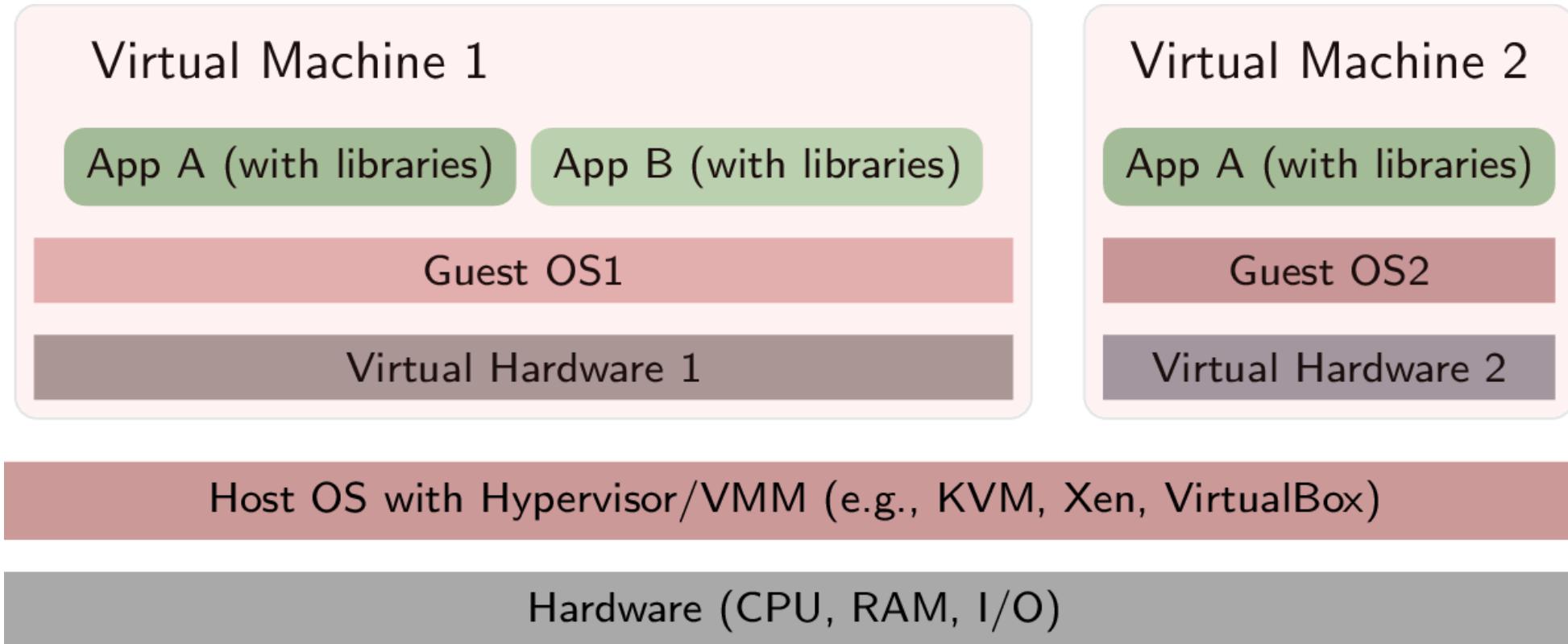
- Com virtualização, a empresa pode criar três servidores digitais, ou máquinas virtuais, em um único servidor físico.
 - **Vantagem:** Menos hardware e menos despesas relacionadas.
- Ela especifica os requisitos do sistema operacional para as máquinas virtuais e pode usá-las como os servidores físicos.



Fonte da Imagem: Parkplace Technologies.

Virtualização

Virtualização promove uma interface de hardware (virtual).



Fonte da Imagem: [IT Systems by Jens Lechtenboerger](#).

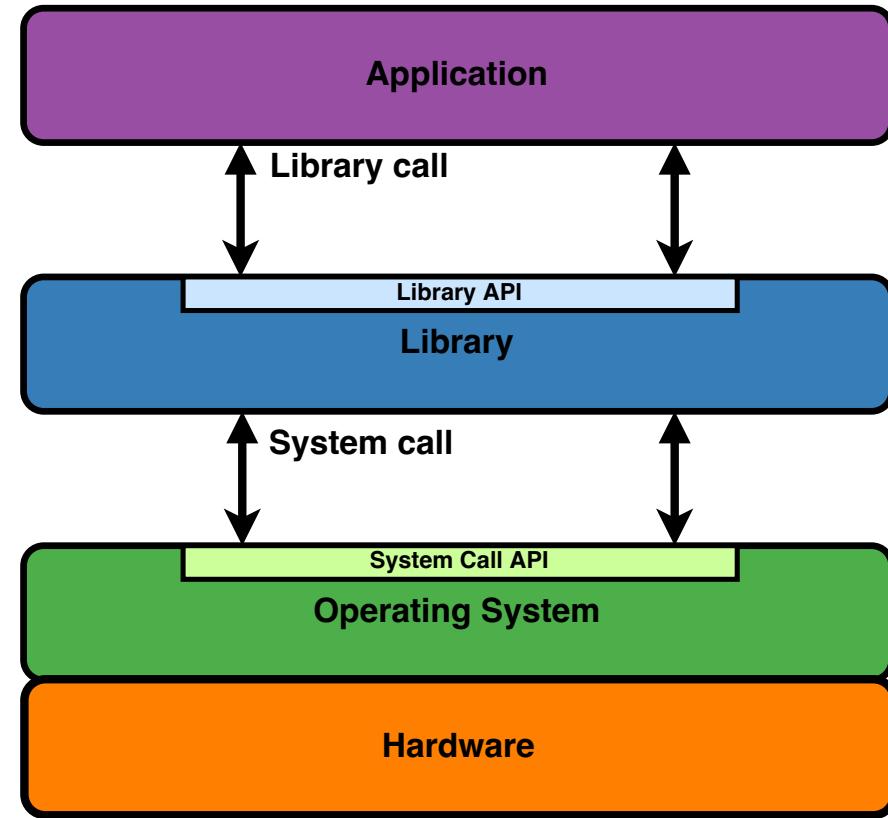
Virtualização e Computação em Nuvem

Virtualização é a **base da Cloud Computing**

- Permite:
 - Elasticidade → criação e remoção dinâmica de recursos.
 - Multilocação (multi-tenancy).
 - Infraestrutura como Serviço (IaaS).

Relembrando: Sistemas Operacionais

- SO é um **software** que:
 - Utiliza recursos de hardware de um sistema computacional, e
 - Provê suporte para execução de outros software.
- SO atua como intermediário entre o hardware e os programas do usuário.
- Garante a execução eficiente e segura de múltiplos processos.

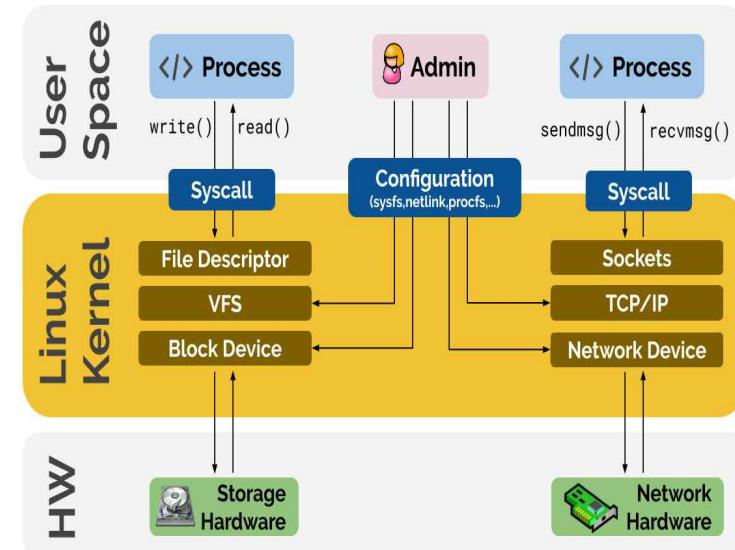


Software Stack. Fonte da Imagem: [OS OER](#).

Relembrando: Kernel de um SO

- **Função geral:** Gerenciar todos os recursos de hardware: CPU, memória e E/S (I/O).
- Fornecer um conjunto de APIs, independentes da arquitetura e do hardware, para permitir que aplicações e bibliotecas no espaço do usuário utilizem os recursos de hardware.
- Lidar com acessos e uso concorrentes dos recursos de hardware por diferentes aplicações.

Kernel Architecture

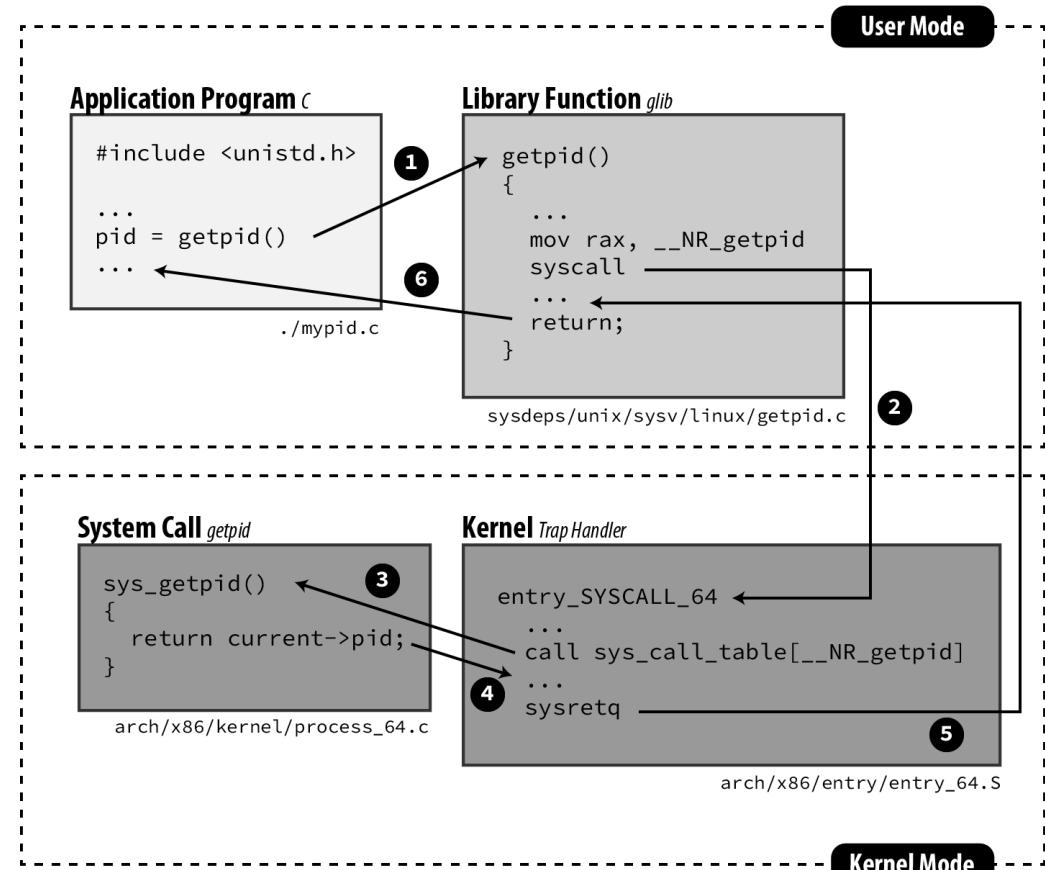


5

Arquitetura geral do Kernel Linux.
Fonte da Imagem: [eBPF - Rethinking the Linux Kernel](#)

Relembrando: Chamadas de Sistema

- **System calls**: A interface entre o sistema operacional e os programas de usuário é o conjunto (`open` , `close` , `read` , `fork` , `execve` , etc.).
- Encapsulada pela biblioteca C: as aplicações geralmente não fazem uma chamada de sistema diretamente, mas sim utilizam a função correspondente da biblioteca C.
- Exemplo: chamada de sistema `getpid` em sistemas Intel/Linux coloca o número 20 no registrador `eax` (20 é o número correspondente à chamada de sistema `getpid`) e então executa `INT 0x80`, que gera uma interrupção.



Fonte da imagem: [Julien Sobczak](#)

Objetivos da Virtualização

- **Otimização** do uso de recursos computacionais.
- **Isolamento** de ambientes e aplicações.
- **Escalabilidade** e elasticidade.
- **Redução de custos** em hardware e manutenção.

Tipos de Virtualização

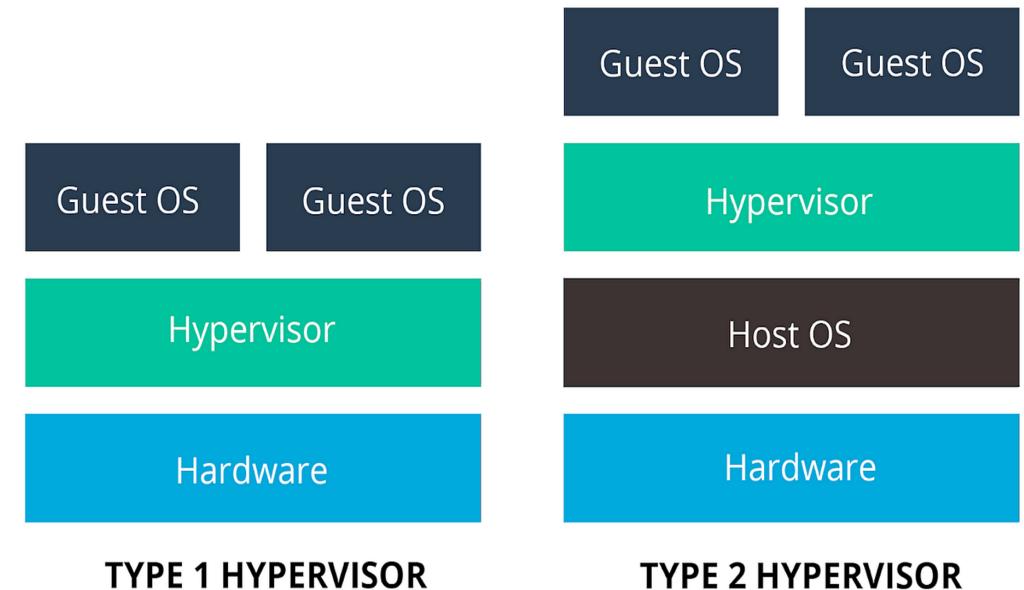
- **Virtualização de Servidores**: Consolidação de vários servidores em uma única máquina física.
- **Virtualização de Armazenamento**: Combinação de múltiplos dispositivos de armazenamento em um único recurso lógico.
- **Virtualização de Rede**: Segmentação e controle de tráfego em redes virtuais (ex.: Software-defined networking).
- **Virtualização de Aplicações**: Aplicações executadas em ambientes isolados, sem instalação direta.

Máquinas Virtuais (VMs)

- **Instâncias virtuais** de sistemas operacionais que operam em um ambiente isolado criado pelo hypervisor.
 - Cada VM age como um servidor independente, com seu próprio sistema operacional e aplicativos, proporcionando flexibilidade e isolamento entre diferentes ambientes de trabalho.
- **Imagens de VM**: Arquivos que contêm uma cópia completa de um sistema operacional, aplicativos, e configurações de uma máquina virtual.
 - Facilitam a implantação rápida de novas VMs, clonagem de ambientes e recuperação de desastres.

Hipervisor

- Software que permite a **criação e execução** de máquinas virtuais (VMs) ao fornecer uma **camada de abstração** entre o hardware físico e os sistemas operacionais virtuais.
- Gerencia a alocação de recursos físicos, permitindo a execução de múltiplos sistemas operacionais em um único hardware físico.
- **Tipo 1 (bare-metal)**: executa diretamente no hardware (Ex.: VMware ESXi, Xen, Hyper-V).
- **Tipo 2 (hosted)**: executa sobre um sistema operacional (Ex.: VirtualBox, VMware Workstation).



Fonte da Imagem: [Techspot](#).

Virtual Machine Monitor (VMM)

- VMM gerencia a alocação de CPU, memória, dispositivos de E/S e rede.
 - O ambiente virtualizado não deve reduzir excessivamente a velocidade de execução dos programas.
 - Entretanto, o acesso ao hardware virtualizado requer intervenção do VMM, acarretando trabalho adicional e sobrecarga.
- Programas virtualizados devem ser restringidos aos recursos atribuídos pelo VMM.
- VMM precisa ter a capacidade de recuperar o controle sobre os recursos já alocados.

Vantagens da Virtualização

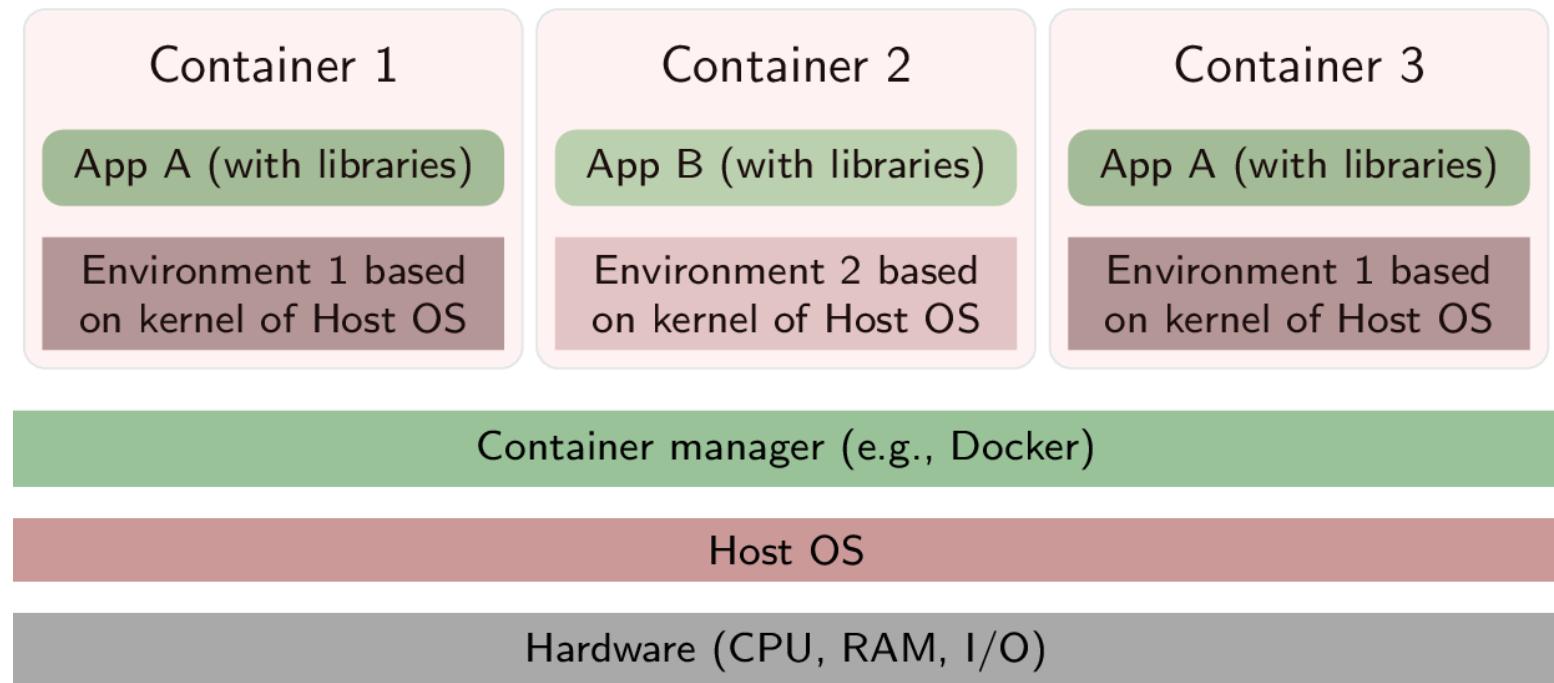
- Melhor aproveitamento de hardware.
- Redução de custos com energia e manutenção.
- Flexibilidade para testes e desenvolvimento.
- Migração e recuperação de desastres facilitadas.

Desafios e Limitações

- Sobrecarga de desempenho (overhead).
- Dependência do hipervisor.
- Complexidade de gerenciamento.
- Questões de segurança em ambientes multi-tenant.

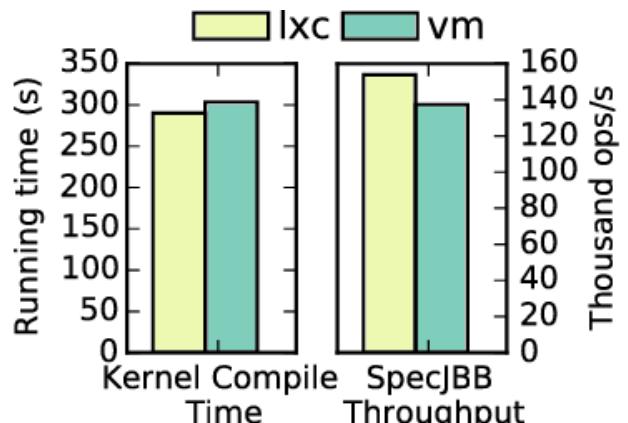
Containers

"Containerization (e.g., with Docker) as **lightweight** variant of virtualization"

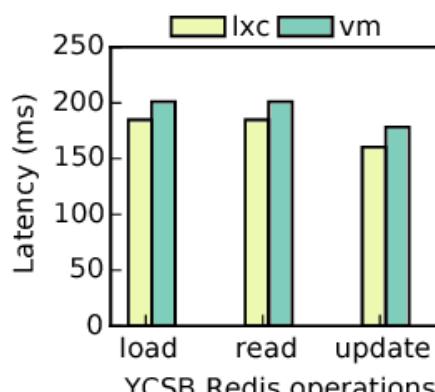


Fonte da Imagem: [IT Systems by Jens Lechtenboerger](#)

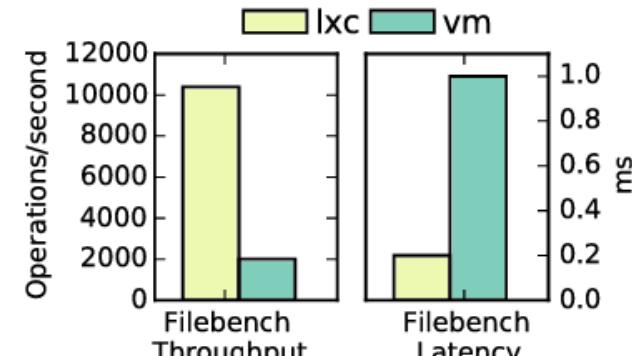
Desempenho



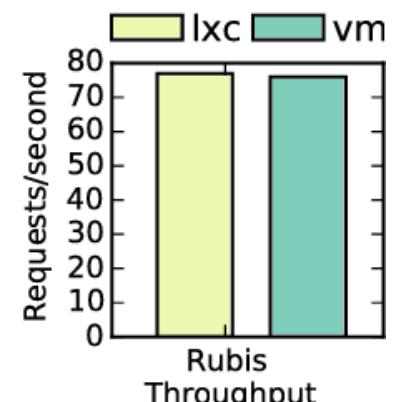
(a) CPU intensive



(b) Memory intensive



(c) Disk intensive



(d) Network intensive

Performance overhead of KVM is negligible for our CPU and memory intensive workloads, but high in case of I/O intensive applications.

Unlike CPU and memory operations, I/O operations go through the hypervisor—contributing to their high overhead.

Fonte: [Containers and Virtual Machines at Scale: A Comparative Study](#).

Resumo

- **Virtualização**: Abstração de recursos físicos (CPU, memória, I/O) em máquinas virtuais independentes.
 - Permite execução simultânea de múltiplos sistemas operacionais no mesmo hardware.
- **Hipervisor (VMM)**: Gerencia recursos, isolamento e migração das VMs.
 - Tipos: **bare-metal** (Tipo 1) vs **hosted** (Tipo 2).
- **Benefícios**: redução de custos de hardware & energia, provisionamento sob demanda, recuperação via snapshots, live migration, failover automático.
- **Desafios**: latência de E/S, gerenciamento de múltiplas VMs & políticas, segurança e isolamento.
- **Relevância na Computação em Nuvem**: Base para infraestruturas **IaaS** (Amazon EC2, Google Compute Engine).
- Leitura recomendada: [The Ideal Versus the Real: Revisiting the History of Virtual Machines and Containers](#)

Perguntas e Discussão

- A virtualização elimina totalmente a necessidade de hardware dedicado?
- Quais problemas de segurança podem surgir em ambientes altamente virtualizados?
- Qual a diferença entre “VM snapshot” e “VM backup”?
- Como escolher entre hypervisor de Tipo-1 e Tipo-2?
- De que maneira o hipervisor (VMM) influencia a latência de E/S em workloads intensivos em disco ou rede, e quais técnicas podem mitigar essa sobrecarga?
- Como os mecanismos de isolamento do hipervisor garantem segurança entre VMs distintas, e quais riscos persistem mesmo com tal isolamento?