ACH 2147 — Desenvolvimento de Sistemas de Informação Distribuídos

Aula 07: Processos (parte 2)

Prof. Renan Alves

Escola de Artes, Ciências e Humanidades — EACH — USP

18/03/2024

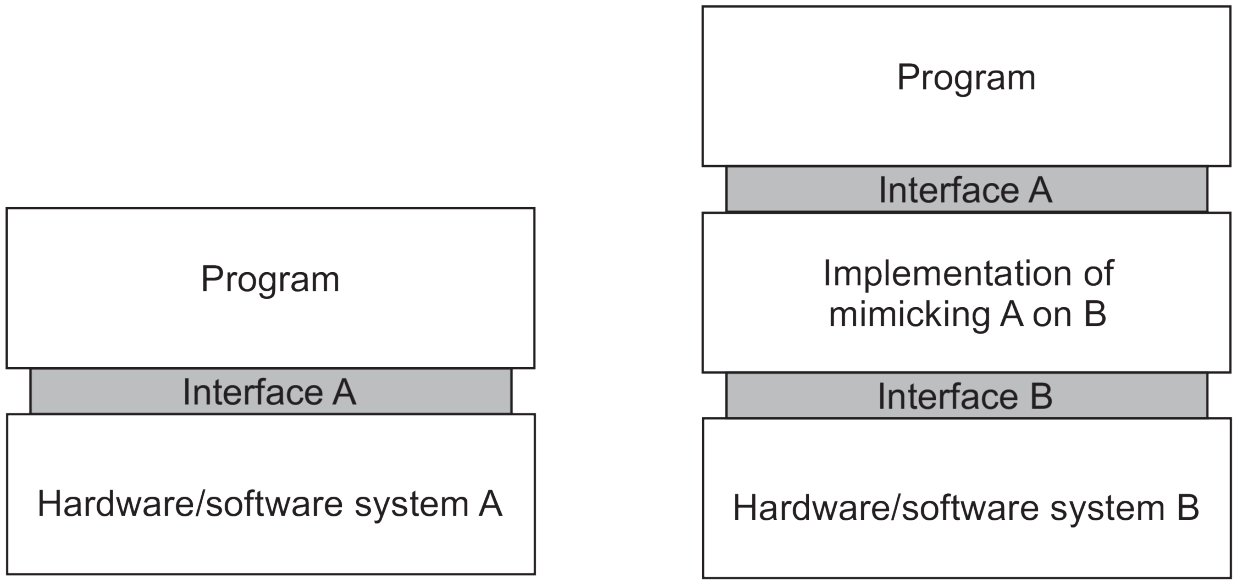
Virtualização

Observação

Virtualização é importante:

- Hardware muda mais rápido que software
- Facilidade de portabilidade e migração de código
- Isolamento de componentes com falhas ou sob ataque

Princípio: imitando interfaces

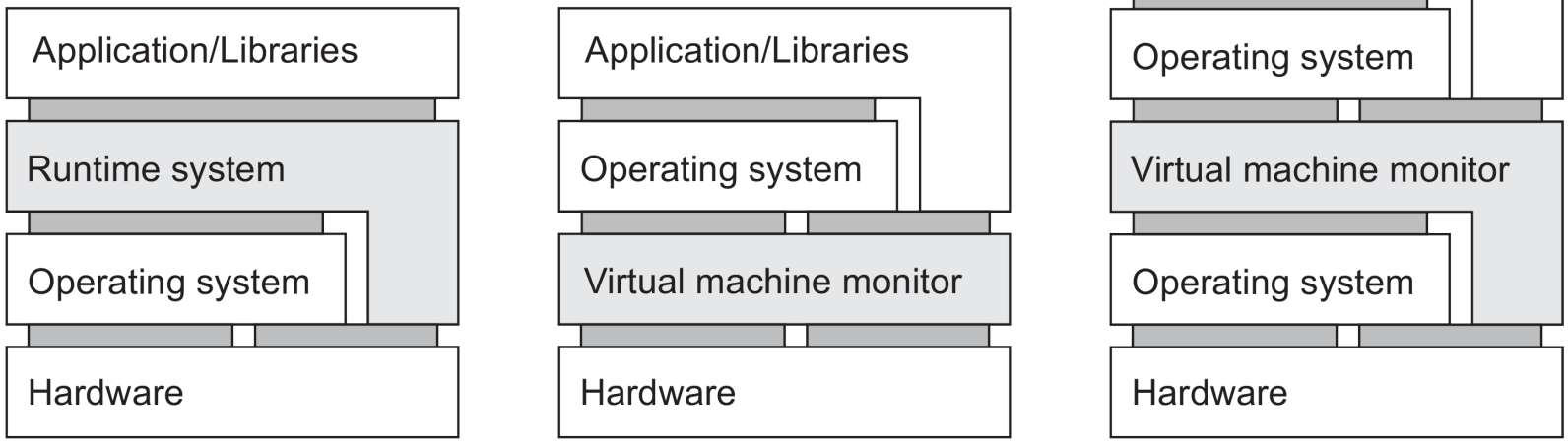


Imitando interfaces

Quatro tipos de interfaces em três níveis diferentes

1. Arquitetura de conjunto de instruções: o conjunto de instruções de máquina, com dois subconjuntos:
   - Instruções privilegiadas: permitidas a serem executadas apenas pelo sistema operacional.
   - Instruções gerais: podem ser executadas por qualquer programa.
2. Chamadas de sistema (syscall) oferecidas por um sistema operacional.
3. Chamadas de biblioteca, conhecidas como Application Programming Interface (API)

Formas de virtualização



(a) VM de processo (b) VMM Nativo (c) VMM Hospedado

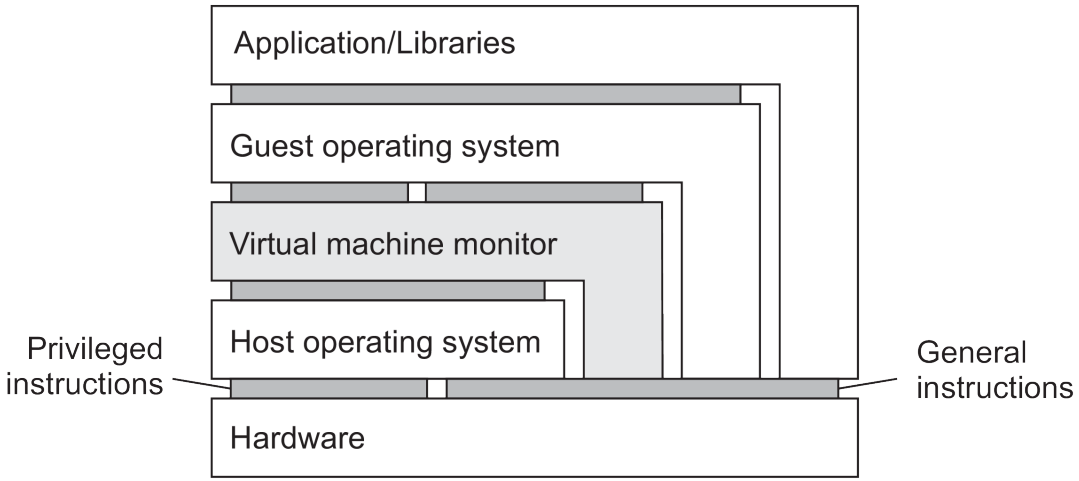
*VMM = Virtual Machine Monitor

Diferenças

(a) Conjunto separado de instruções, um interpretador/emulador, rodando em cima de um SO.
(b) Instruções de baixo nível, junto com um sistema operacional mínimo.
(c) Instruções de baixo nível, mas delegando a maioria do trabalho para um sistema operacional completo.

Explorando as VMs: desempenho

Refinando a organização



- Instrução privilegiada: se e somente se executada no modo usuário, causa uma trap (armadilha) para o sistema operacional
- Instrução não privilegiada: o restante

Instruções especiais

- Instrução sensível ao controle: pode afetar a configuração de uma máquina (por exemplo, uma configuração que afeta o registro de realocação ou tabela de interrupção).
- Instrução sensível ao comportamento: efeito é parcialmente determinado pelo contexto (por exemplo, `POPF` sobe uma flag de habilitação de interrupção, mas apenas no modo sistema).

Condição para virtualização

Condição necessária

Para qualquer computador convencional, um monitor de máquina virtual pode ser construído se o conjunto de instruções sensíveis para este computador for um subconjunto do conjunto de instruções privilegiadas.

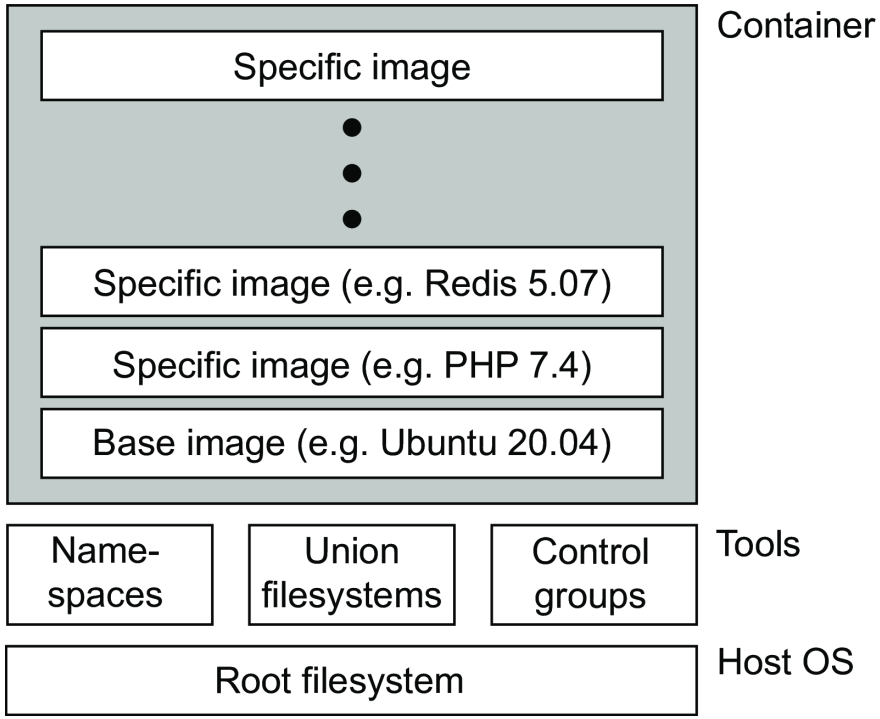
Problema: a condição nem sempre é satisfeita

Pode haver instruções sensíveis que são executadas no modo usuário sem causar uma trap para o sistema operacional.

Soluções

- Emular todas as instruções
- Encapsular instruções sensíveis não privilegiadas para desviar o controle para o VMM
- Paravirtualização: modificar o sistema operacional convidado (guest), seja impedindo instruções sensíveis não privilegiadas, ou tornando-as não sensíveis (ou seja, alterando o contexto).

Containers



- Namespaces: uma coleção de processos em um container recebe sua própria visão de identificadores
- Sistema de arquivos em união (union file system): combina vários sistemas de arquivos em camadas, permitindo a operação de `write` apenas na camada mais alta.
- Grupos de controle: restrições de recursos podem ser impostas a uma coleção de processos.

Exemplo: PlanetLab

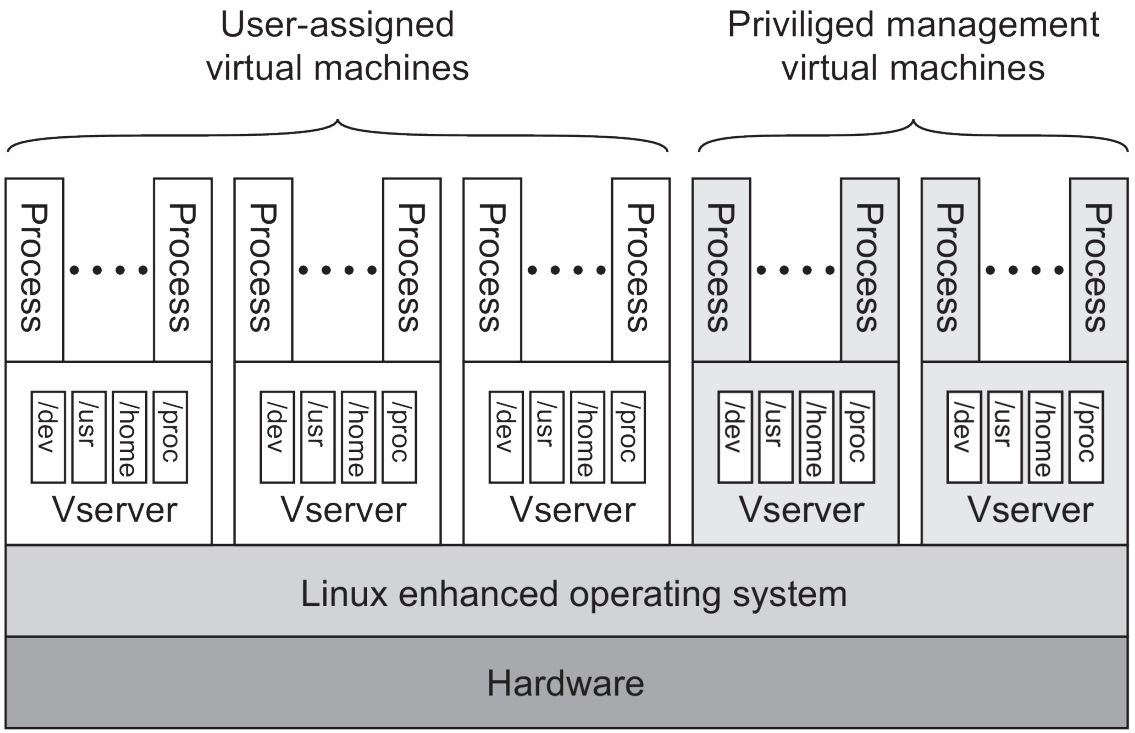
Essência

Diferentes organizações contribuem com máquinas, que posteriormente são compartilhados para vários experimentos.

Problema

Precisamos garantir que diferentes aplicações distribuídas não atrapalhem umas às outras $\Rightarrow$ virtualização

PlanetLab: Organização básica



Vserver

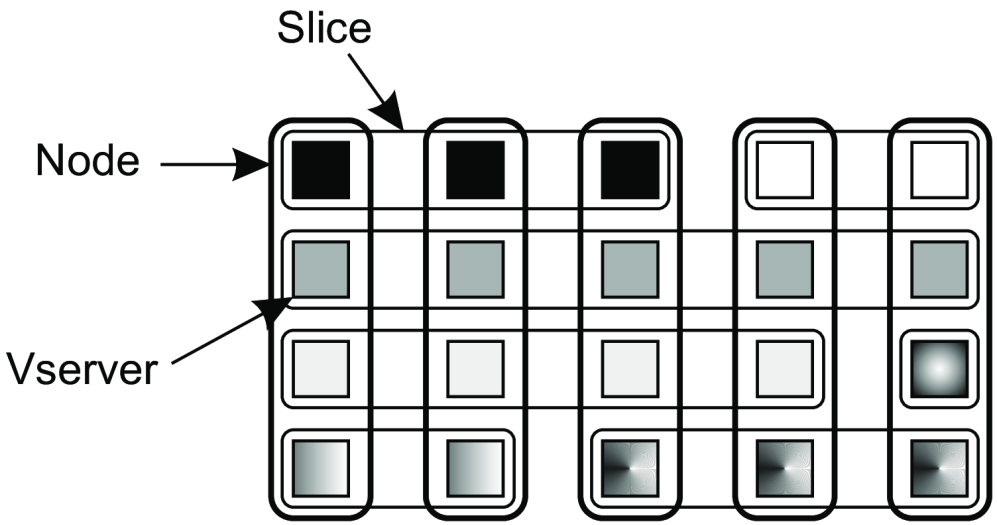
Ambiente independente e protegido com suas próprias bibliotecas, versões de servidor, etc. Aplicações distribuídas são atribuídas a uma coleção de vservers distribuídos em várias máquinas

PlanetLab Vservers e slices

Essência

- Cada Vserver opera em seu próprio ambiente (cf. `chroot`).
- As melhorias do Linux incluem o ajuste adequado dos IDs de processo (por exemplo, `init` ter ID 0).
- Dois processos em diferentes Vservers podem ter o mesmo ID de usuário, mas não implica o mesmo usuário.

Separação leva a slices



VMs e computação em nuvem

Três tipos de serviços em nuvem

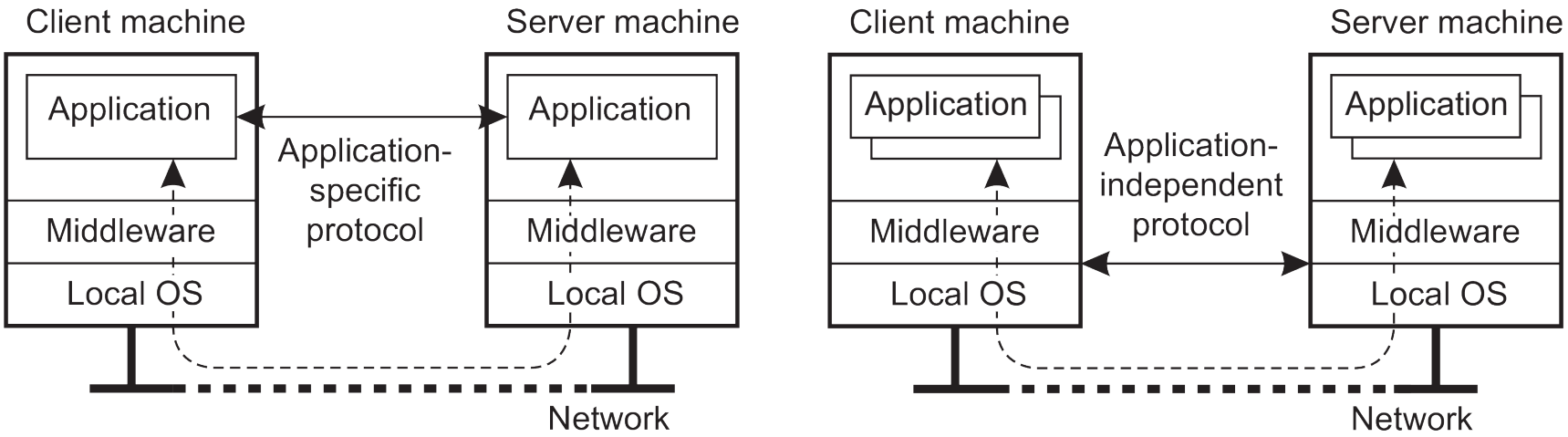
- Infrastructure-as-a-Service cobrindo a infraestrutura básica
- Platform-as-a-Service cobrindo serviços em nível de sistema
- Software-as-a-Service contendo aplicativos de fato

IaaS

Em vez de alugar uma máquina física, um provedor de nuvem alugará uma VM (ou VMM) que pode estar compartilhando uma máquina física com outros clientes $\Rightarrow$ isolamento quase completo entre os clientes (embora o isolamento de desempenho possa não ser alcançado).

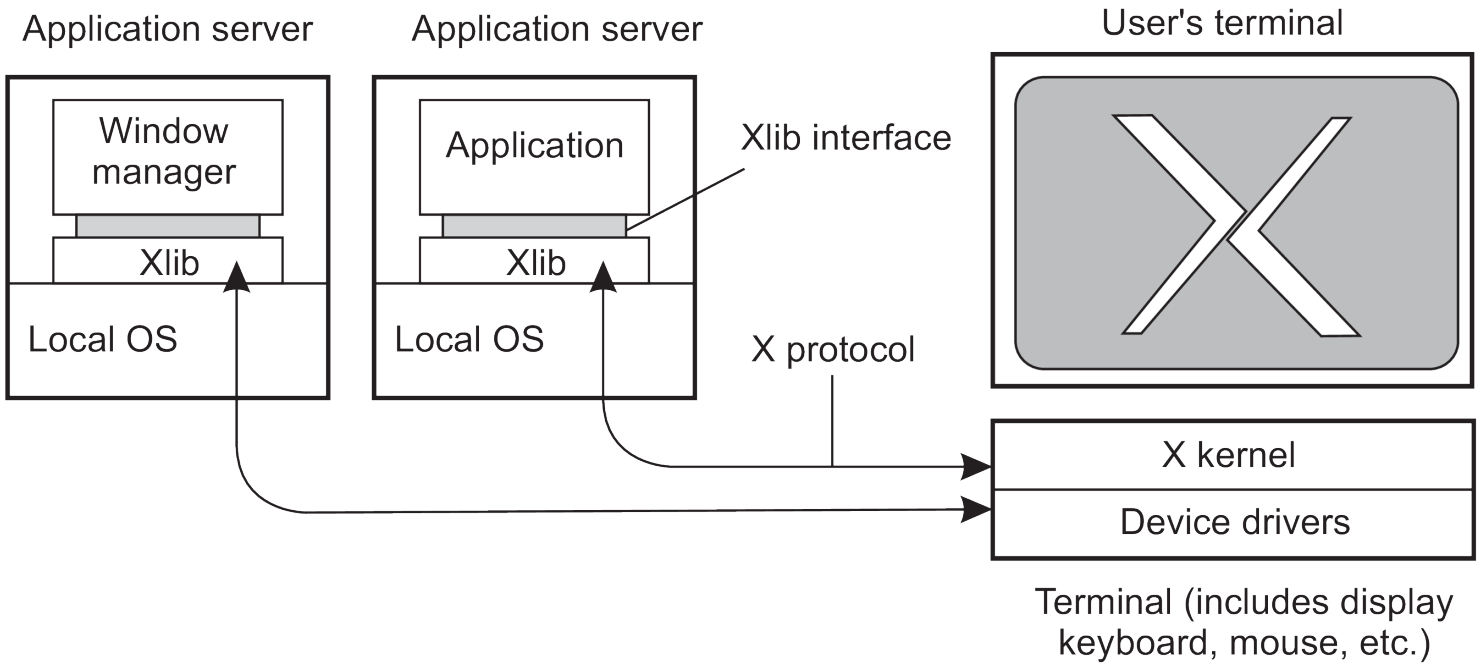
Interação cliente-servidor

Distinguir soluções em nível de aplicação e middleware



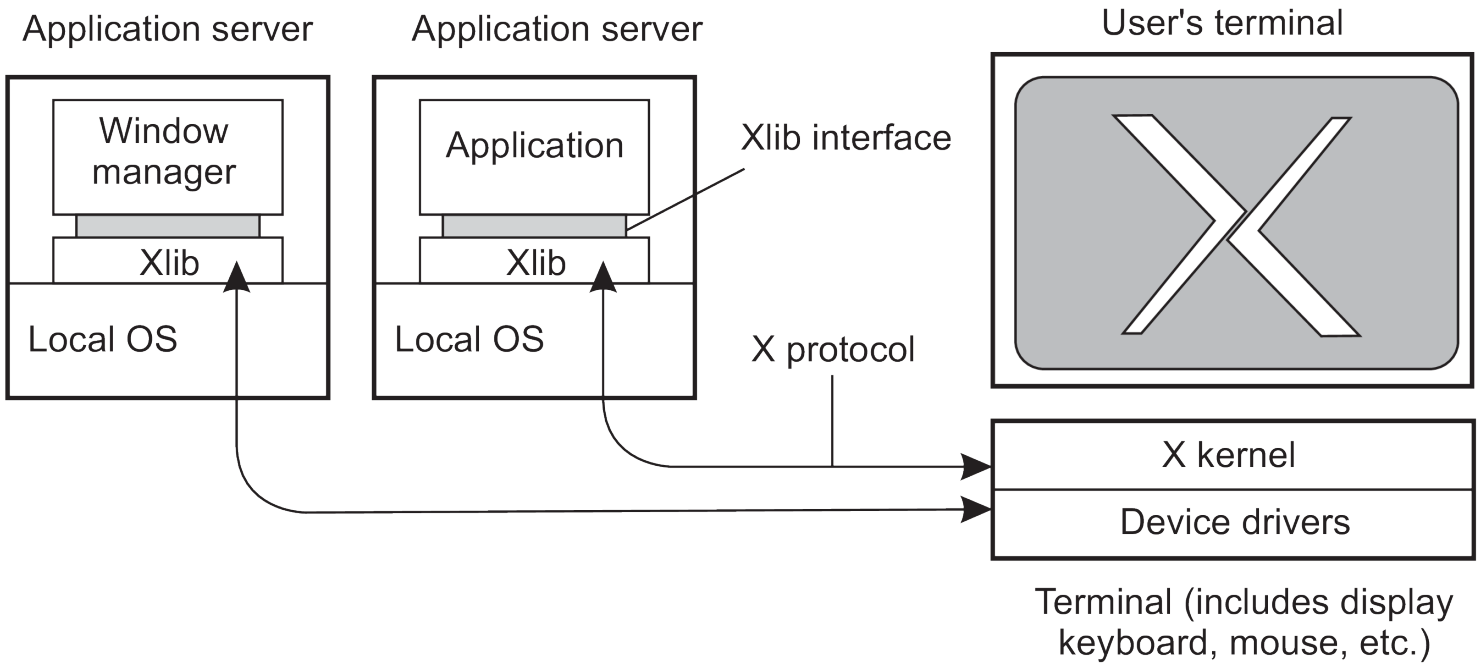
Exemplo: O sistema X Window

Organização básica



Exemplo: O sistema X Window

Organização básica



Cliente e servidor X

O aplicativo age como um cliente para o X-kernel, este último sendo executado como um servidor na máquina do cliente.

Melhorando o X

Observações

- Muitas vezes não há uma separação clara entre a lógica da aplicação e os comandos da interface do usuário
- Aplicações tendem a operar de maneira altamente síncrona com um kernel X

Abordagens alternativas

- Permitir que os aplicativos controlem o display completamente, até mesmo a nível de pixel (por exemplo, VNC)
- Fornecer apenas algumas operações de alto nível no display (dependentes dos drivers de vídeo locais), permitindo operações de exibição mais eficientes.

Ambiente de desktop virtual

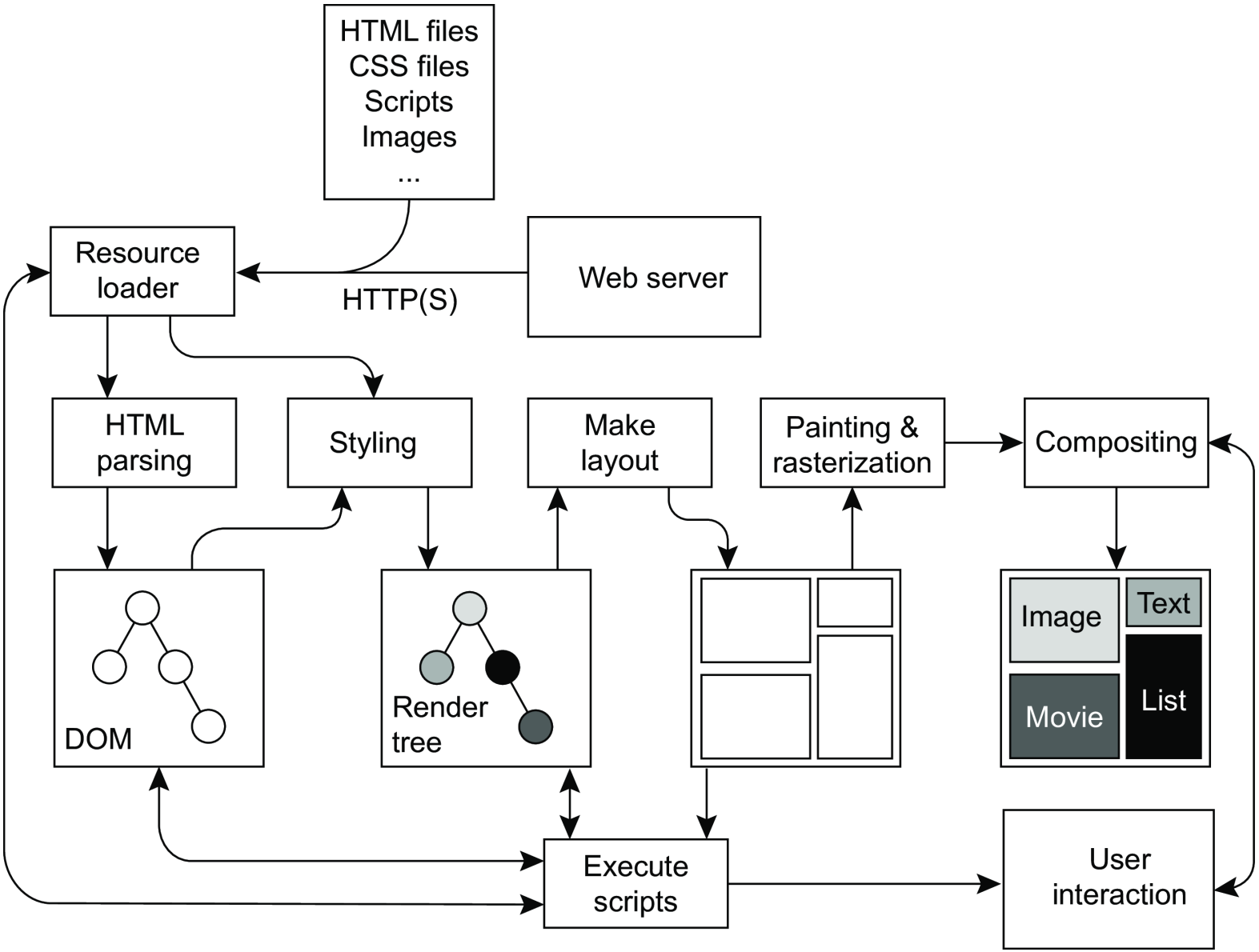
Desenvolvimento lógico

Com um número crescente de aplicações baseadas em nuvem, a questão é como usar essas aplicações a partir do ambiente do usuário?

- Problema: desenvolver uma interface de usuário em rede definitiva
- Resposta: usar um navegador da Web para estabelecer uma experiência transparente

O Google Chromebook

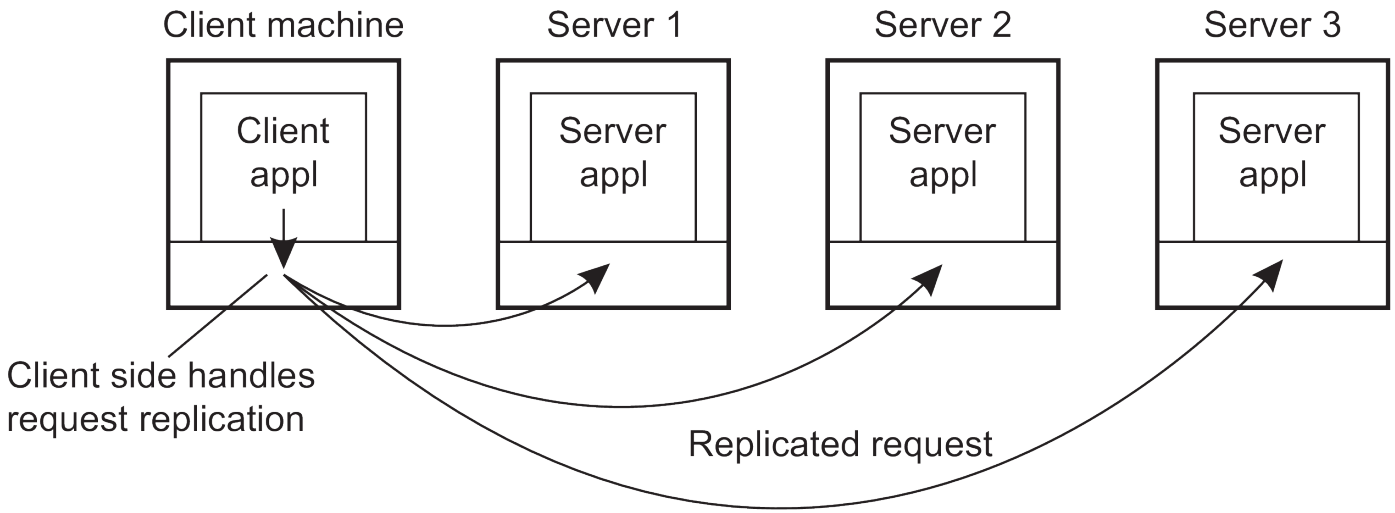
A anatomia de um navegador da Web



Software do lado do cliente

Geralmente feito para alcançar transparência de distribuição

- Transparência de acesso: stubs do lado do cliente para RPCs
- Transparência de localização/migração: permite que o software do lado do cliente acompanhe a localização do servidor
- Transparência de replicação: múltiplas invocações tratadas pelo stub do cliente:



- Transparência de falha: frequentemente pode ser colocada apenas no cliente (estamos tentando mascarar falhas de servidor e de comunicação).