

ACH 2147 — Desenvolvimento de Sistemas de Informação Distribuídos

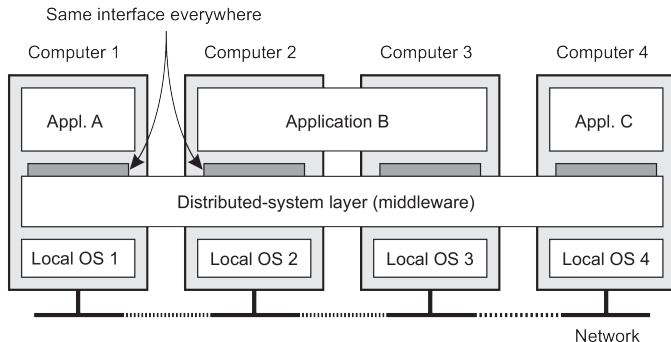
Aula 04: Arquiteturas (02)

Prof. Renan Alves

Escola de Artes, Ciências e Humanidades — EACH — USP

08/03/2024

Middleware: o SO dos sistemas distribuídos



Paralelo com SO:

- Gerenciador de recursos: comunicação entre aplicações, serviços de segurança, mascaramento e recuperação de falhas
- Contém componentes e funções comumente utilizados que não precisam ser implementados separadamente pelas aplicações.

Usando componentes legados para construir um middleware

Premissa

Construção de um sistema a partir da junção de componentes pré-existentes.

Problema

As interfaces oferecidas por um componente legado provavelmente não são adequadas para todas as aplicações.

Solução

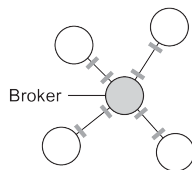
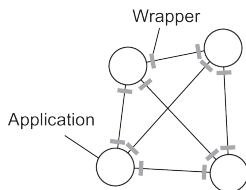
Um **wrapper** ou **adaptador** oferece uma interface compreensível pela aplicação cliente. Suas funções são transformadas nas disponíveis no componente.

Exemplo

Usar uma “casca” RESTful

Organizando wrappers

Duas soluções: 1-para-1 ou através de um broker



Complexidade com N aplicações/componentes

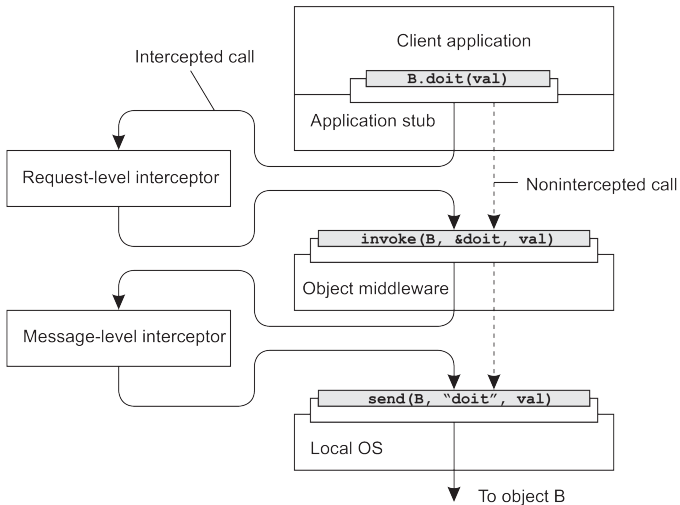
- **1-para-1**: requer $N \times (N - 1) = \mathcal{O}(N^2)$ wrappers
- **broker**: requer $2N = \mathcal{O}(N)$ wrappers

Desenvolvendo middleware com interceptadores

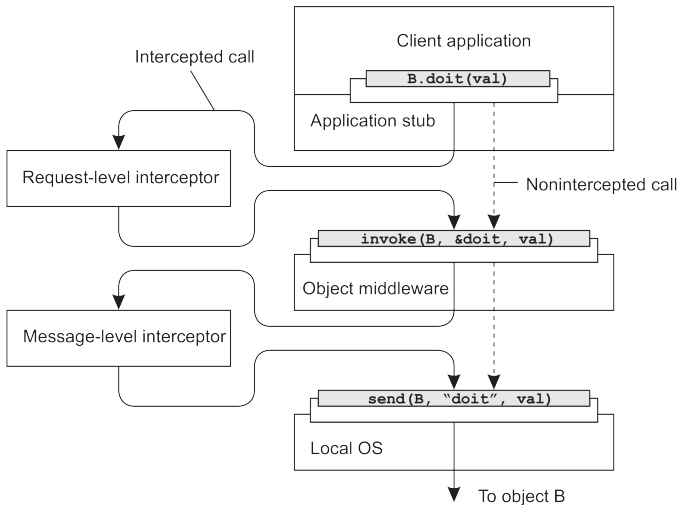
Interceptador

- Elemento de software introduzido no fluxo de execução de um programa
- Permite alteração do comportamento
- Ferramenta que pode permitir agregar funcionalidades sem modificar diretamente o middleware

Interceptar o fluxo de controle usual

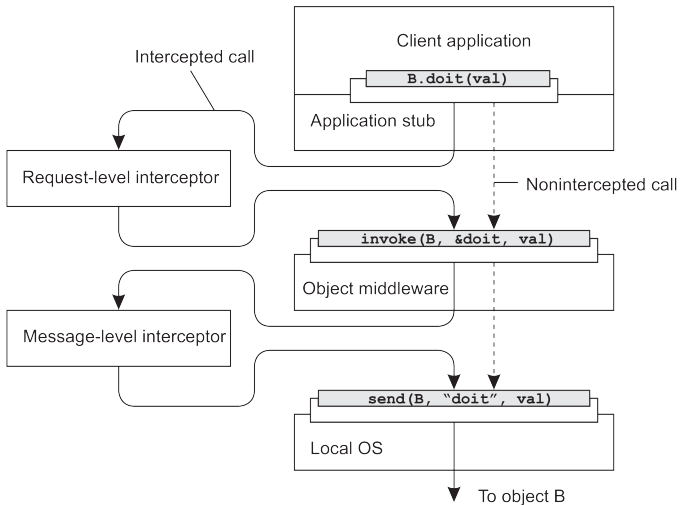


Interceptar o fluxo de controle usual



E se B estiver replicado?

Interceptar o fluxo de controle usual



E se B estiver replicado? E se `val` for grande?

Middleware adaptável

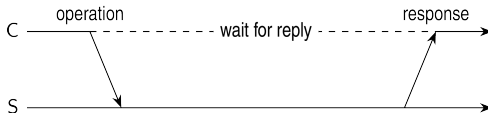
- **Necessidade de adaptação:**
 - Mudanças no ambiente: mobilidade, falhas de rede e hardware.
- **Mudanças em tempo de execução:**
 - Alguns sistemas não podem ser desligados: requer adaptação dinâmica.
- **Como resolver?**
 - Uso de configuração dinâmica, *late binding* e *dynamic loading*.
- **Desafios em sistemas distribuídos:**
 - Aumenta responsabilidade do middleware.
 - Dificuldade em manter estado coerente.
 - Dependência complexa entre componentes.

Arquiteturas centralizadas

Modelo Cliente-Servidor Básico

Características:

- Há processos que oferecem serviços (**servidores**)
- Há processos que usam serviços (**clientes**)
- Clientes e servidores podem estar em máquinas diferentes
 - UDP vs TCP
- Clientes seguem o modelo de requisição/resposta ao usar os serviços

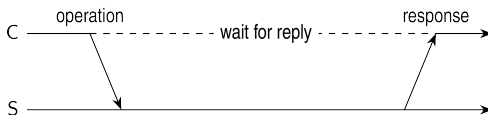


Arquiteturas centralizadas

Modelo Cliente-Servidor Básico

Características:

- Há processos que oferecem serviços (**servidores**)
- Há processos que usam serviços (**clientes**)
- Clientes e servidores podem estar em máquinas diferentes
 - UDP vs TCP
- Clientes seguem o modelo de requisição/resposta ao usar os serviços



Mas...

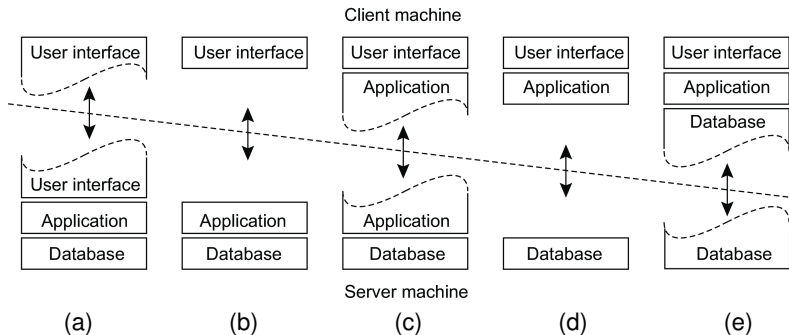
A distinção entre cliente e servidor nem sempre é óbvia

Arquiteturas centralizadas em várias camadas

Algumas organizações tradicionais

- **De uma camada:** configuração de terminal burro/mainframe
- **De duas camadas:** configuração cliente/servidor único
- **De três camadas:** cada camada em uma máquina separada

Configurações tradicionais de duas camadas

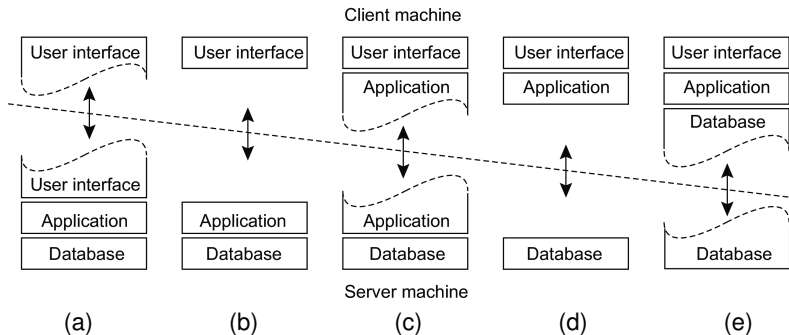


Arquiteturas centralizadas em várias camadas

Algumas organizações tradicionais

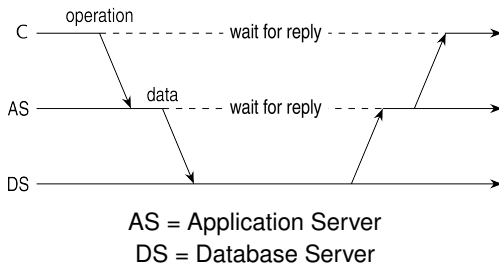
- **De uma camada:** configuração de terminal burro/mainframe
- **De duas camadas:** configuração cliente/servidor único
- **De três camadas:** cada camada em uma máquina separada

Configurações tradicionais de duas camadas (qual é melhor?)



Sendo cliente e servidor ao mesmo tempo

Arquitetura de três camadas



Exemplo

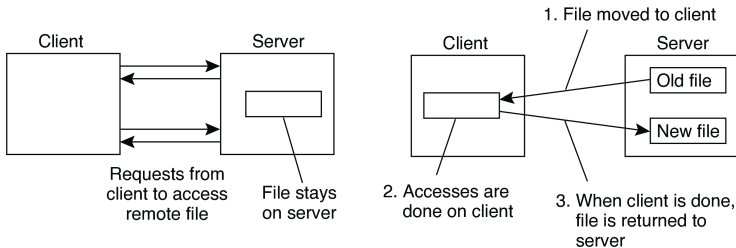
Transaction Processing Monitor (TPM)

Exemplo: Sistemas de Arquivos em Rede

Fundamentos

Cada servidor NFS (Network File System) fornece uma visualização padronizada de seu sistema de arquivos local: cada servidor suporta o mesmo modelo, independentemente da implementação do sistema de arquivos.

O modelo de acesso remoto do NFS



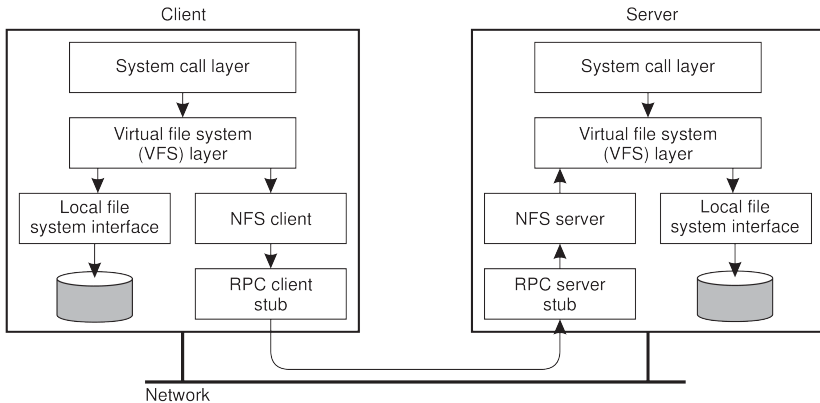
Nota

Acesso remoto

Upload/download

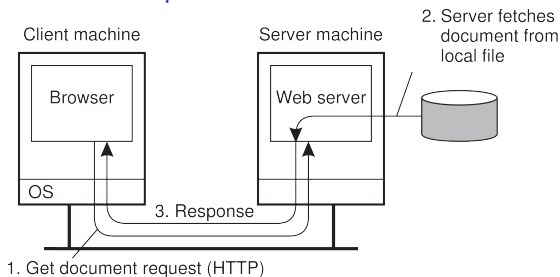
O FTP é um modelo típico de upload/download. O mesmo pode ser dito para sistemas como o Dropbox.

Arquitetura do NFS



Exemplo: Servidores Web simples

Voltando aos velhos tempos...

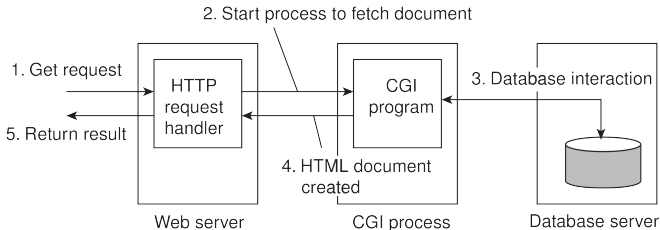


...a vida era simples:

- Um site consistia em uma coleção de arquivos HTML
- Os arquivos HTML podiam se referir uns aos outros por um [hiperlink](#)
- Um servidor Web basicamente precisava apenas de um hiperlink para encontrar um arquivo
- Um navegador se encarregava de renderizar corretamente o conteúdo de um arquivo

Exemplo (continuação): Servidores Web menos simples

Ainda nos velhos tempos...



...a vida ficou um pouco mais complicada:

- Um site foi construído em torno de um banco de dados com conteúdo
- Uma página da Web ainda poderia ser referenciada por um [hiperlink](#)
- Um servidor Web basicamente precisava apenas de um hyperlink para buscar um arquivo
- Um programa separado (**Common Gateway Interface (CGI)**) [compunha](#) uma página
- Um navegador se encarregava de renderizar corretamente o conteúdo de um arquivo