

ARQUITETURAS DISTRIBUÍDAS

PARTE 02

DCE540 - Computação Paralela e Distribuída

Atualizado em: 2 de agosto de 2021

Iago Carvalho

Departamento de Ciência da Computação



ORGANIZAÇÃO DE MIDDLEWARE

Como vimos na última aula, existem diversas arquiteturas de software

- Baseadas em camadas
- Baseadas em objetos
- Baseadas em recursos
- Baseadas em eventos

Todas elas são baseadas em *middlewares*

Independente da arquitetura de software desenvolvida, um *middleware* deve implementar dois diferentes *design patterns*

- *Wrapper*
- *Interceptor*

Um *wrapper* (ou adaptador) é um *design pattern* que oferece

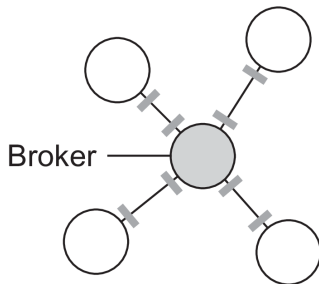
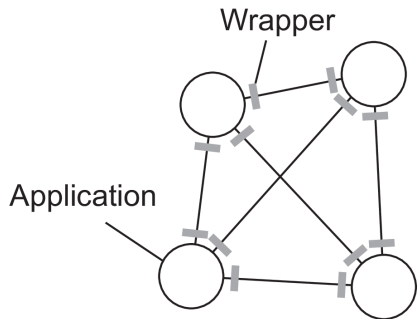
- Interface de comunicação entre componentes
- Facilidade de acesso a componentes legados

Novamente, existe um paralelo muito grande com Engenharia/Desenvolvimento de softwares

Existem duas formas de se implementar *wrappers*

- Na interface de cada componente
- Um componente *wrapper*

WRAPPERS



Similar a um *Interrupt handler* de um sistema operacional

- Recebe uma requisição
- Interrompe o fluxo normal da aplicação
- Trata a requisição
- Retoma o fluxo normal da aplicação

Responsável por fazer a invocação de objetos remotos

- Um componente utiliza um método de outro
- Componentes localizados em nós diferentes
- Localização de cada componente é transparente
 - O *interceptor* é responsável por esta transparência

Na aula passada, nós já estudamos arquitetura de software

- Como aplicações estão organizadas
- Conectores
- Interfaces
- Componentes

Além da arquitetura de software, também é importante falar sobre a arquitetura de sistema

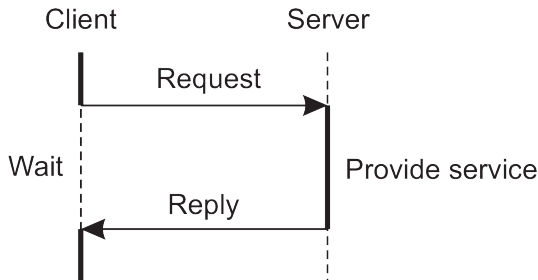
- Onde cada software (ou aplicação) está localizado
- Como será a interação entre cada componente

Centralizada, distribuída ou híbrida

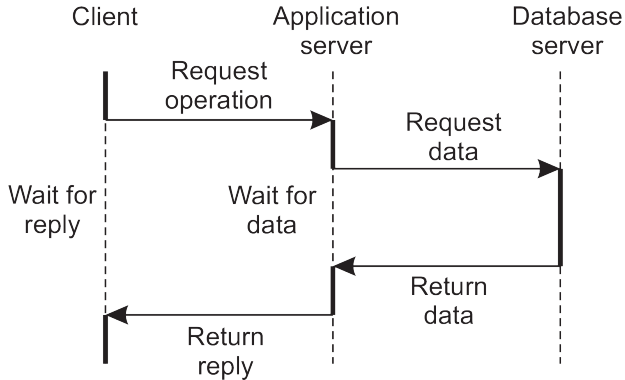
ARQUITETURA DE SISTEMA CENTRALIZADA

É a maneira mais simples de pensarmos

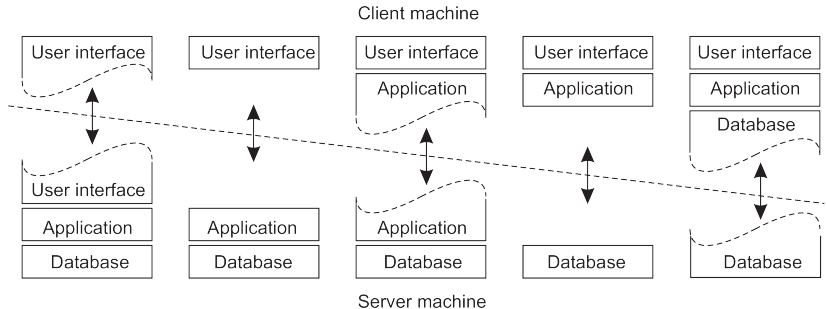
- Cliente-servidor
- Uma ou mais camadas
- Protocolo orientado a conexão ou não
 - Exemplo: TCP e UDP



ARQUITETURA DE SISTEMA CENTRALIZADA MULTI-CAMADAS



ORGANIZAÇÃO DE ARQUITETURA DE SISTEMA CENTRALIZADA



Basicamente, são as arquiteturas de sistemas Peer-to-Peer (P2P)

Neste tipo de arquitetura, dizemos que temos uma distribuição horizontal

- Clientes e servidores divididos em partes
- Cada cliente ou servidor opera uma parte do sistema
 - Uma parte dos dados, por exemplo
- Balanceamento de carga

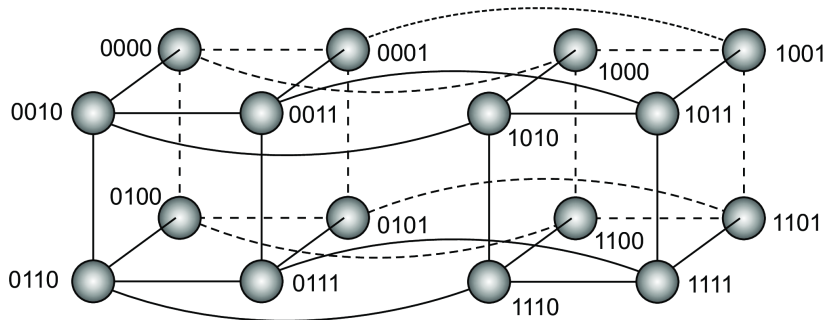
Nós são organizados em uma topologia específica

- Anel
- Grid
- Árvore
- ...

Cada nó tem um identificador único

- Um nó pode mandar mensagem para outros utilizando este identificador
- Tabela hash distribuída

Estrutura de hipercubo



Não existe uma tabela hash distribuída com o endereço dos nós do sistema

Ao invés disso, cada nó possui uma lista com o endereço de seus vizinhos

- Tabela construída de forma *ad-hoc*
- Tabela construída de forma dinâmica
 - A tabela permite inclusões, remoções e modificações

Vantagens:

- Fácil de incluir ou remover nós do sistema
 - Escalabilidade do sistema

Desvantagens:

- Dados não podem ser localizados instantaneamente
 - É necessário realizar uma busca
 - *Flooding*
 - *Random walks*

Busca em largura

1. Um nó interessado em um dado faz a requisição para seus vizinhos
2. Caso o vizinho possua a informação, ele responde
3. Caso contrário, ele retransmite a requisição para seus vizinhos
 - Caso o vizinho já tenha recebido a requisição antes, ele não retransmite mais

Necessário informar o tempo de vida (TTL) da mensagem

- Muito curto: dado provavelmente não será encontrado
- Muito longo: muito caro, grande número de troca de mensagens

Busca em profundidade

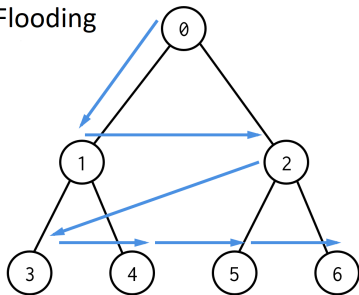
1. Um nó interessado em um dado faz a requisição para um de seus vizinhos, escolhido de forma aleatória
2. Caso o vizinho possua a informação, ele responde
3. Caso contrário, ele retransmite a requisição para outro de seus vizinhos, de forma aleatória

Necessário informar o máximo número de retransmissões

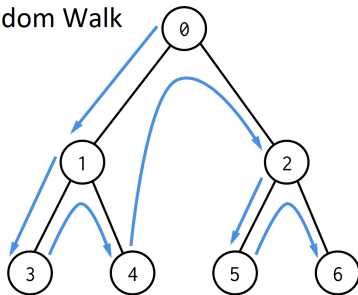
Além disso, pode-se iniciar múltiplos *random walks* em paralelo

- Cada um iniciando em um vizinho diferente

Flooding



Random Walk



Construídas utilizando hierarquia de nós

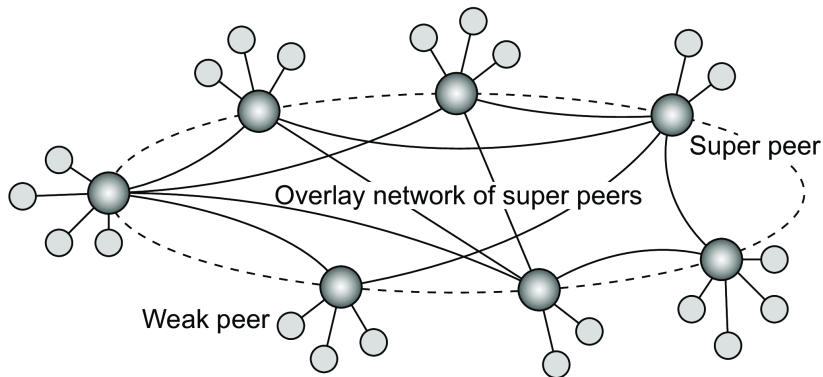
- Super nós
 - Escolhidos através de uma eleição
 - Devem ter alta disponibilidade
- Nós fracos
 - Se ligam a um super nó

Existe uma arquitetura P2P estruturada entre os super nós

Cada super nó funciona como *gateway* para uma outra rede P2P

- Pode ser estruturada ou não

ARQUITETURA P2P SEMI-ESTRUTURADA



Combinam características das arquiteturas centralizadas e descentralizadas

- BitTorrent
- Domain Names Service (DNS)
- Internet Service Providers (ISP)

Estruturas são muito mais complexas e utilizadas para construir sistemas distribuídos de larga escala