

# ARQUITETURAS DISTRIBUÍDAS

DCE540 - Computação Paralela e Distribuída

Atualizado em: 28 de julho de 2021

Iago Carvalho

Departamento de Ciência da Computação



Um sistema distribuído costuma ser complexo

- Diversos softwares
- Múltiplas máquinas (nós) espalhados

É necessário que tudo seja ordenado e organizado da melhor maneira possível

- Organização lógica
  - Como os softwares e serviços estão organizados
  - Como um software ou serviço interage com outros
- Organização física
  - Literalmente, onde cada nó está localizado no mundo real
  - Sistemas de comunicação entre nós

O objetivo da arquitetura de sistemas distribuídos é realizar uma separação entre software e hardware

- Um software e seus dados necessariamente tem que estar em um (ou mais) nós da rede
- A principal ideia de um sistema distribuído é tornar os dados, recursos e serviços transparentes ao usuário
  - Este não precisa saber onde estão localizados os dados, recursos e serviços que está utilizando
- Esta separação é realizada por um *middleware*

Um sistema distribuído pode ser implementado utilizando três tipos de arquitetura

1. **Centralizada** ← foco desta disciplina

- Um único nó da rede (servidor) implementa a maioria dos serviços
- Sistemas cliente-servidor
- Exemplo: caixa eletrônico

2. Descentralizada

- Camadas distribuídas entre múltiplos nós da rede
- Múltiplos nós, cada um processando uma parcela das requisições

3. Híbrida

- Combinação das duas acima

Refere-se a organização lógica dos softwares e serviços em um sistema distribuído

- A escolha de uma arquitetura de software é uma das principais decisões a serem tomadas durante o projeto de um sistema distribuído

Baseada em **componentes** e **conectores**

## Componente

- Um *pedaço* do sistema
- Possui função e interfaces bem definidas
- Similar a um método em programação orientada a objetos
- Pode ser substituído, desde que as interfaces sejam preservadas

## Conector

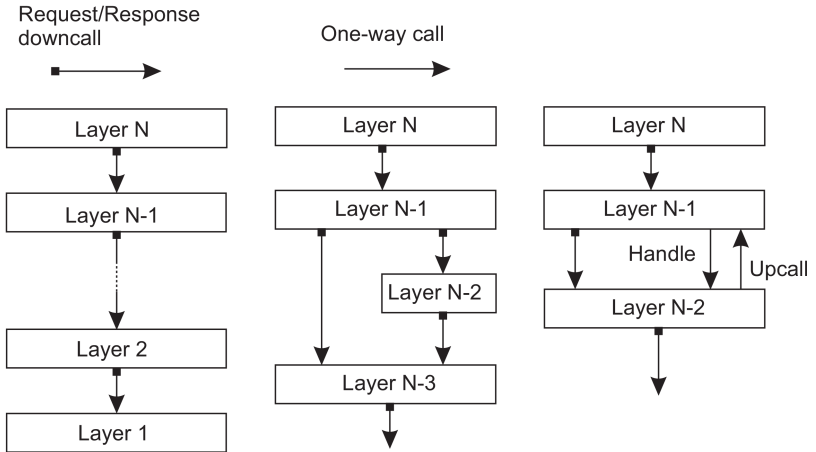
- Controla e descreve a maneira como componentes comunicam-se uns com os outros
- Controla o fluxo de dados e informação entre os componentes

Dependendo da organização dos componentes e seus conectores, diversas arquiteturas de software podem ser implementadas

Entre as diversas arquiteturas, podemos destacar quatro

1. Baseadas em camadas
2. Baseadas em objetos
3. Baseadas em recursos
4. Baseadas em eventos

# ARQUITETURA DE SOFTWARE BASEADAS EM CAMADAS





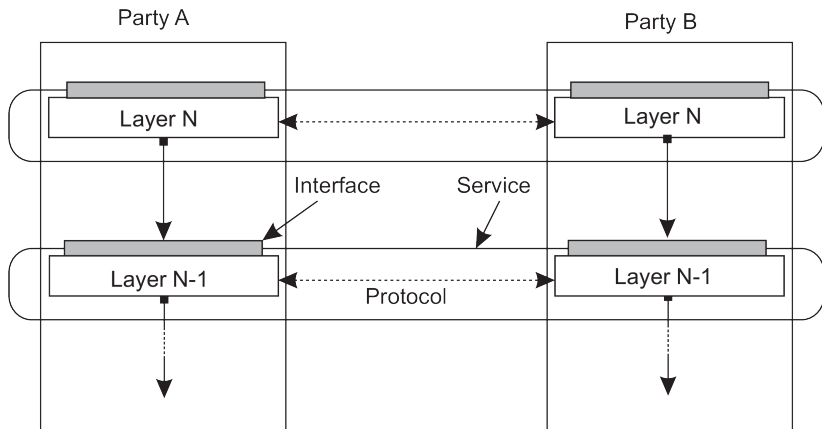
Uma camada necessariamente tem que se comunicar com outras

- A camada deve prover uma interface de comunicação (conector)
- Comunicação entre camadas é realizada através deste conector

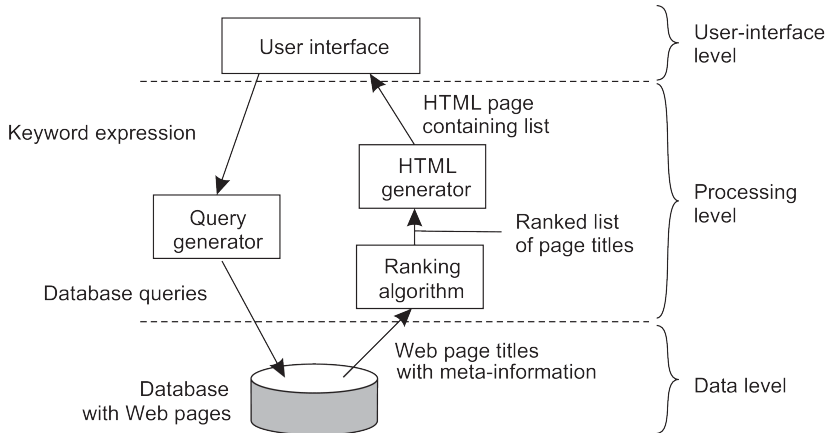
Entretanto, uma camada pode enviar requisições para camadas de outros nós da rede

- Tarefa também da interface (conector)
- O conector implementa um protocolo de comunicação
  - Exemplos de protocolos: TCP e UDP

# ARQUITETURAS BASEADAS EM CAMADAS - PROTOCOLOS



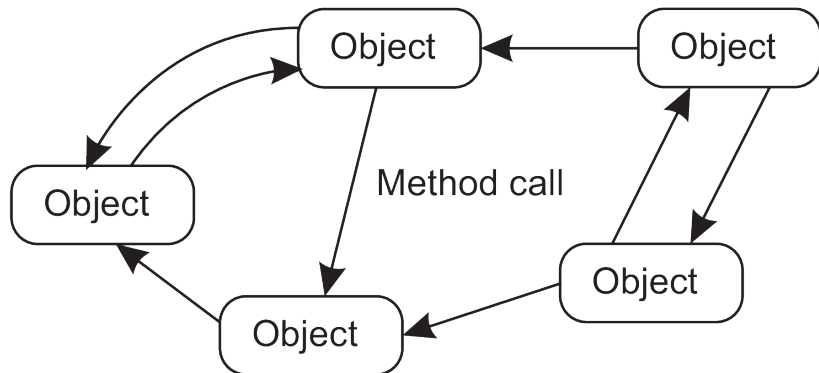
# EXEMPLO: MÁQUINA DE BUSCA NA WEB



# ARQUITETURAS BASEADAS EM OBJETOS

Arquitetura de software de um sistema distribuído onde cada componente é um objeto

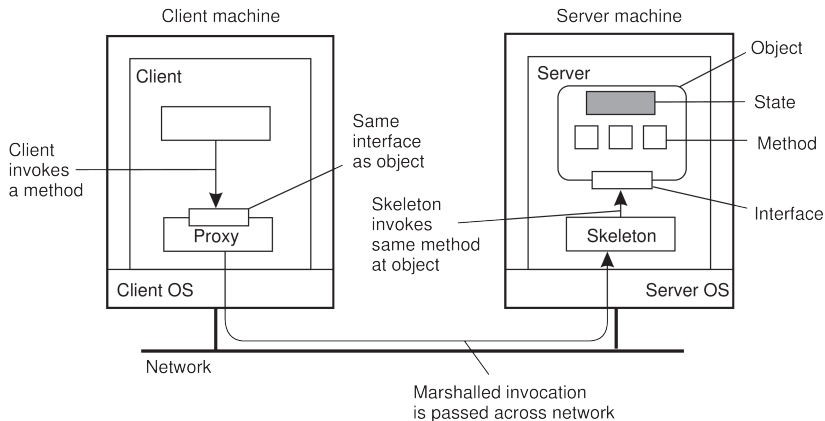
- Extremamente similar ao conceito de orientação de objetos em desenvolvimento de software



# OBJETOS REMOTOS

É comum que um nó da rede concentre todos os objetos

- Somente as interfaces de acesso a eles são disponíveis
- Arquitetura orientada a serviços



É uma arquitetura baseada em objetos distribuídos

- Serviços REST (Representational State Transfer)
- Arquiteturas RESTful

É uma maneira ordenada de se implementar diversos objetos distribuídos

Interface oferece quatro operações básicas

- PUT
- POST
- GET
- DELETE

Também conhecidas como arquiteturas *publish-subscribe*

Neste tipo de arquitetura, existe uma clara separação entre processos e comunicação

- Vamos nos focar na coordenação entre processos

Comunicação pode ser

- Síncrona
- Assíncrona

Da mesma maneira, podem existir dois tipos de referências na comunicação

- Referência explícita
- Referência indireta

# ARQUITETURAS BASEADAS EM EVENTOS

Assim, podemos definir quatro tipos de coordenação de processos, baseados nas diferentes maneiras de referência e comunicação

		Comunicação	
		Síncrona	Assíncrona
Referência	Explícita	Direta	Caixa de e-mails
	Indireta	Baseado em eventos	Espaço de dados compartilhados



## Coordenação direta

- Comunicação síncrona
- Referência explícita

Processos comunicam-se entre si somente se ambos estiverem ativos no mesmo momento

Paralelo com uma ligação telefônica

Coordenação no estilo de caixa de correio (ou e-mail)

- Comunicação assíncrona
- Referência explícita

Processos trocam mensagens entre si

- Não precisam estar ativos ao mesmo tempo
- Mensagem é enviada e fica aguardando resposta

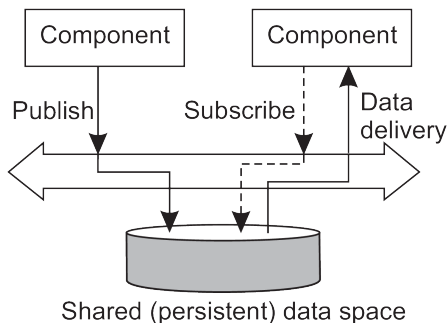
# ESPAÇO DE DADOS COMPARTILHADO

Coordenação no estilo espaço de dados compartilhado

- Comunicação assíncrona
- Referência indireta

Um processo envia uma requisição (ou arquivo, ou dado) que fica armazenado em um servidor

Em qualquer momento posterior, outro processo pode acessar o que foi enviado



# BASEADO EM EVENTOS

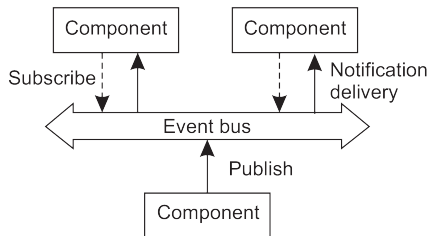
## Coordenação baseada em eventos

- Comunicação síncrona
- Referência indireta

Um processo envia uma requisição (ou arquivo, ou dado)

Cria uma notificação avisando o que ele fez

Outros processos podem acessar



# BASEADO EM EVENTOS

