

# Sistemas Distribuídos

## Aula II – Modelos e Arquiteturas de Sistemas

---

PROF. DR. RAFAEL TEIXEIRA SOUSA

# Modelos

---

1. Modelo Físico
2. Modelo de Arquitetura
3. Arquiteturas de Sistemas

# Modelo Físico

---

Organização de componentes de hardware e conectividade

Modelo básico:

- Componentes que realizam troca de mensagens

Três gerações de modelos físicos

1. Sistemas primitivos
2. Sistemas adaptados para a internet
3. Sistemas contemporâneos

# Modelo Físico

---

## 1. Sistemas primitivos

- 1970 – 1980
- Redes locais de compartilhamento de arquivos e recursos (ex: impressoras)
- Homogêneos

## 2. Sistemas adaptados para a internet

- 1990
- Sistemas distribuídos geograficamente e heterogêneos
- Google – 1996
- Primeiros middlewares

## 3. Sistemas contemporâneos

- Nós móveis (notebooks e smartphones)
- Nós autônomos (cloud computing)

# Modelo Físico

---

## 4. Sistemas distribuídos de sistemas

- Ultra Large Scale
- Internet como uma rede de redes

| <i>Sistemas distribuídos</i> | <i>Primitivos</i>  | <i>Adaptados para Internet</i>   | <i>Contemporâneos</i>   |
|------------------------------|--|--|---|
| <i>Escala</i>                | Pequenos   | Grandes  | Ultragrandes  |
| <i>Heterogeneidade</i>       | Limitada (normalmente, configurações relativamente homogêneas) | Significativa em termos de plataformas, linguagens e <i>middleware</i> | Maiores dimensões introduzidas, incluindo estilos de arquitetura radicalmente diferentes                  |
| <i>Sistemas abertos</i>      | Não é prioridade   | Prioridade significativa, com introdução de diversos padrões           | Grande desafio para a pesquisa, com os padrões existentes ainda incapazes de abranger sistemas complexos  |
| <i>Qualidade de serviço</i>  | Em seu início  | Prioridade significativa, com introdução de vários serviços            | Grande desafio para a pesquisa, com os serviços existentes ainda incapazes de abranger sistemas complexos |

# Modelos de Arquitetura

---

Projeto arquitetônico do sistema

1. Entidades em comunicação
2. Paradigmas de comunicação
3. Funções e responsabilidades
4. Posicionamento

# Entidades em comunicação

---

1. Objetos
2. Componentes
3. Serviços Web

# Paradigmas de comunicação

---

1. Comunicação entre processos

2. Invocação remota

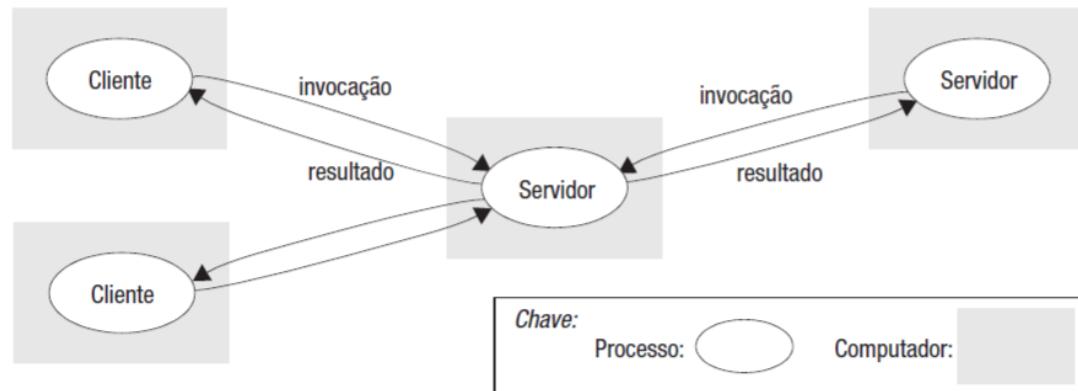
- RPC
- RMI

3. Comunicação Indireta

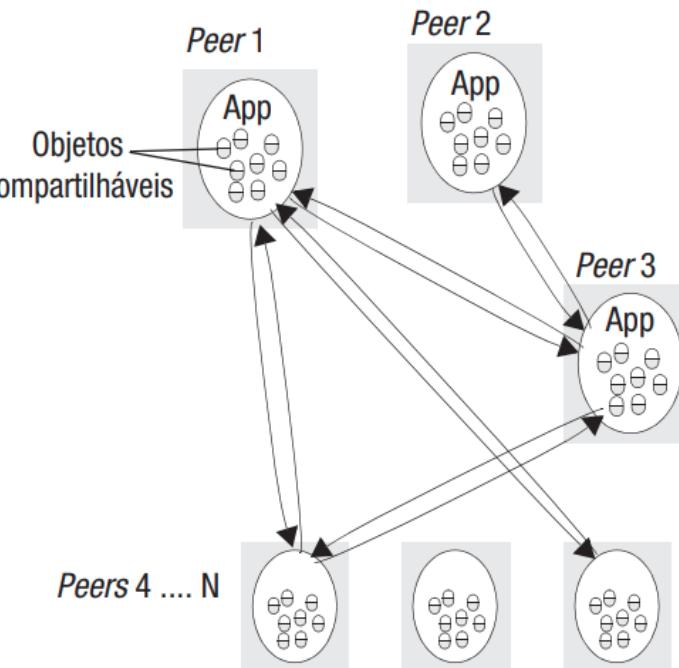
- Comunicação em grupo
- Sistemas publicar-assinar
- Filas de mensagem

# Funções e responsabilidades

Cliente-servidor

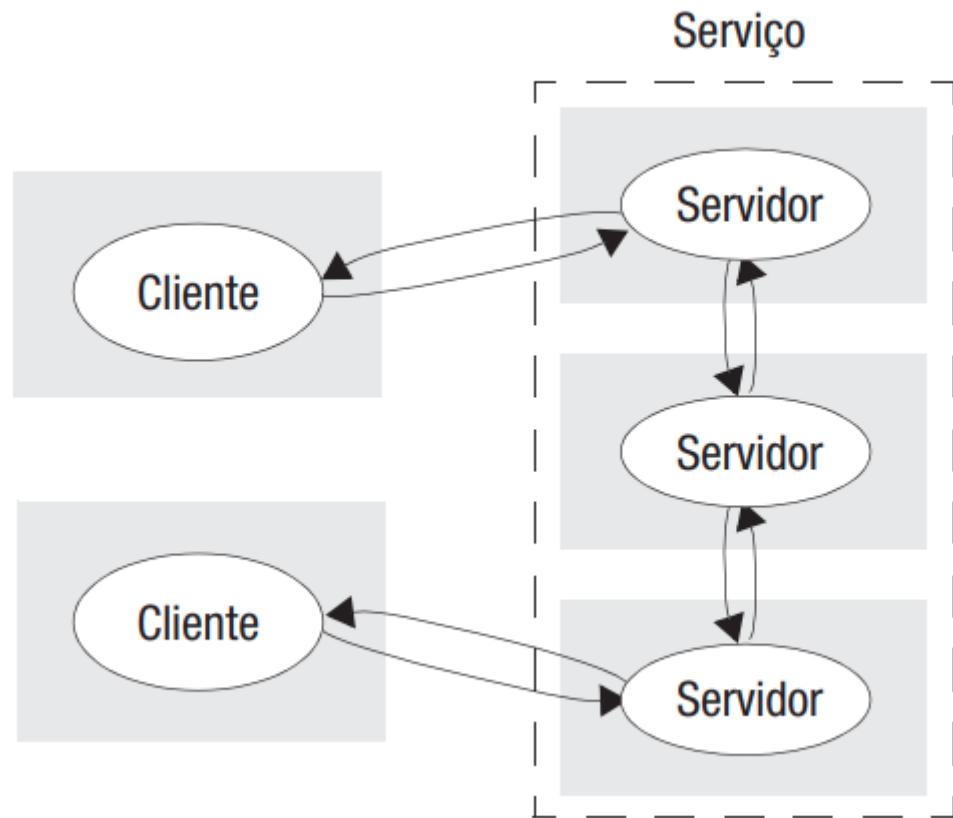


Peer-to-Peer



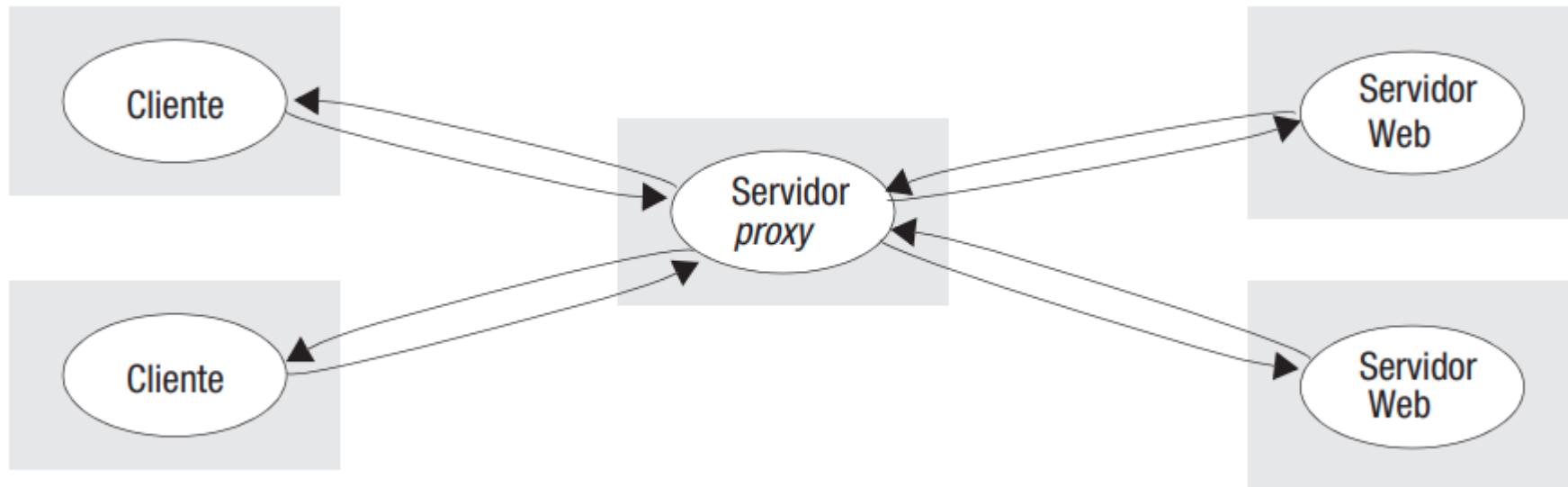
# Mapeamento de serviços

---



# Uso de cache

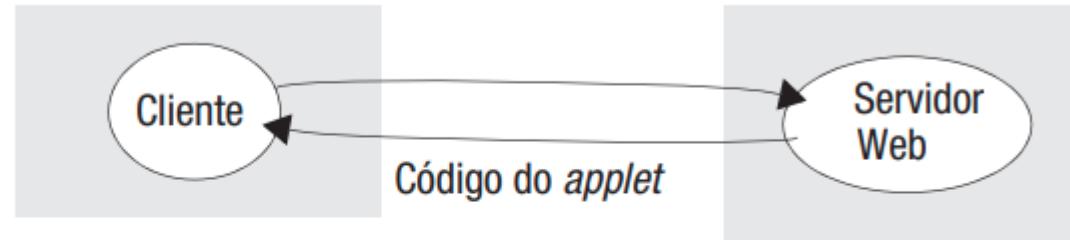
---



# Código móvel

---

a) Requisição do cliente resulta no download do código de um *applet*



b) O cliente interage com o *applet*



# Modelos de Arquitetura

---

Projeto arquitetônico do sistema

**1.** Entidades em comunicação

- Objetos
- Componentes
- Serviços Web

**2.** Paradigmas de comunicação

- Comunicação entre processos
- Chamada de procedimento/Invocação remota
- Sistemas publicar-assinar

**3.** Funções e responsabilidades

- Cliente-servidor
- Peer-to-peer
- Cache
- Código móvel

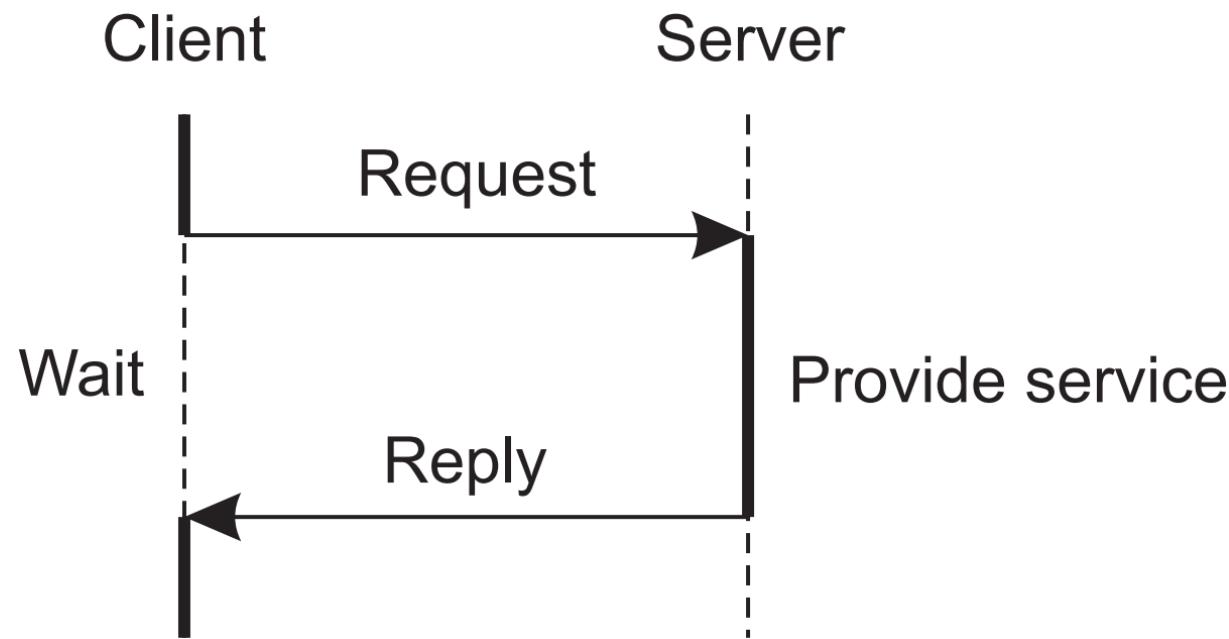
# Arquiteturas de Sistemas

---

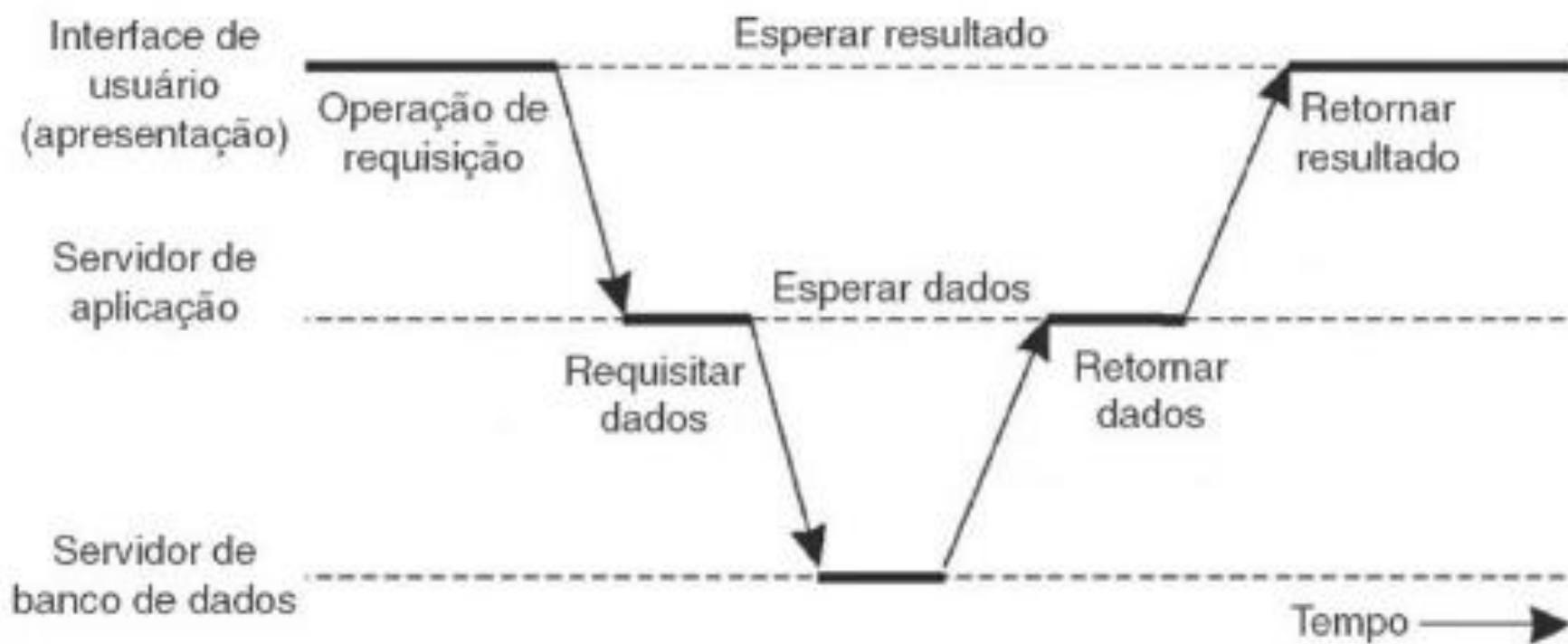
1. Arquiteturas centralizadas
  - Cliente/Servidor: Netflix, site de notícias, internet banking
2. Arquiteturas descentralizadas
  - Sistemas Peer-to-Peer
3. Arquiteturas híbridas
  - Skype, WhatsApp e o BitTorrent

# Cliente/Servidor

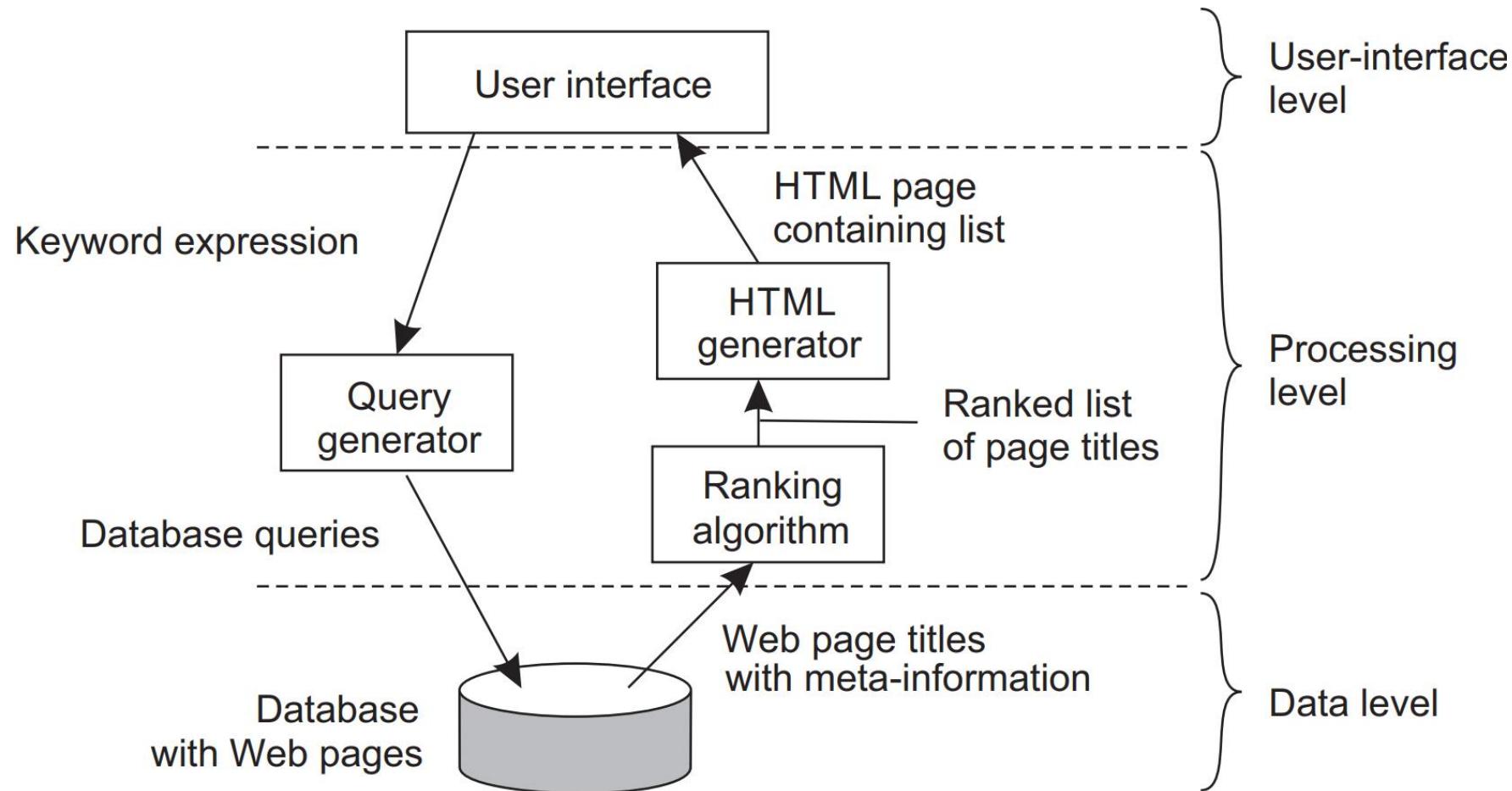
---



# Cliente/Servidor em camadas



# Cliente/Servidor em camadas



# Netflix

---



# Arquiteturas Decentralizadas

---

## Distribuição Vertical

- Componentes logicamente diferentes em máquinas diferentes
- Cada máquina é projetada para um grupo específico de funções

## Distribuição Horizontal

- Cliente ou servidor pode ser fisicamente subdividido em partes logicamente equivalentes
- Possui porção própria de dados
- Balanceamento de carga
- Papéis não muito bem definidos

# Peer-to-Peer

---

Define-se em arquitetura estruturada

Estruturada

- Mecanismo determinístico e estruturado
- Tabela de Hash Distribuída (DHT)
- Dados e nós recebem uma chave aleatória
- Desafio
  - Dada uma chave de um dado, mapear para o identificador de um nó

# Chord

---

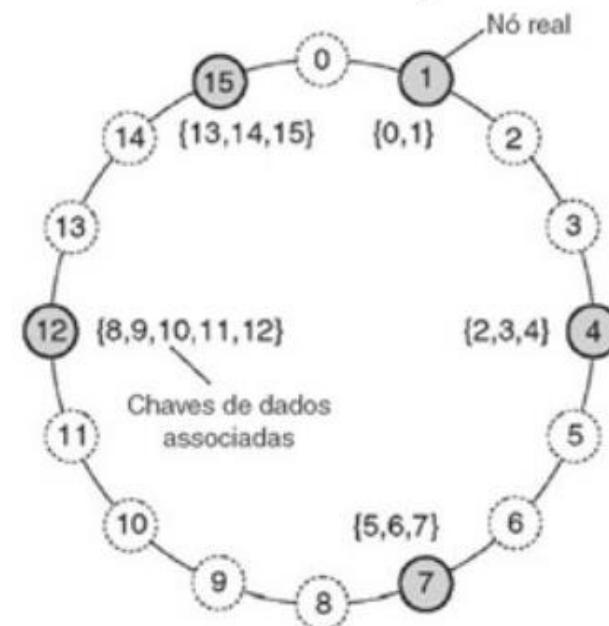
Implementação de um DHT para redes Peer-to-Peer

Nós organizados logicamente em um anel

Cada nó recebe um identificador aleatório

Em um sistema com N nós

- Qualquer requisição e resposta de busca  $O(\log N)$
- Entrada e saída de nós da rede, a informação sobre o roteamento é atualizada  $O(\log^2 N)$



# Arquiteturas Híbridas

---

Centralizada + Descentralizada

Centralizada

- Servir de diretório

Descentralizada

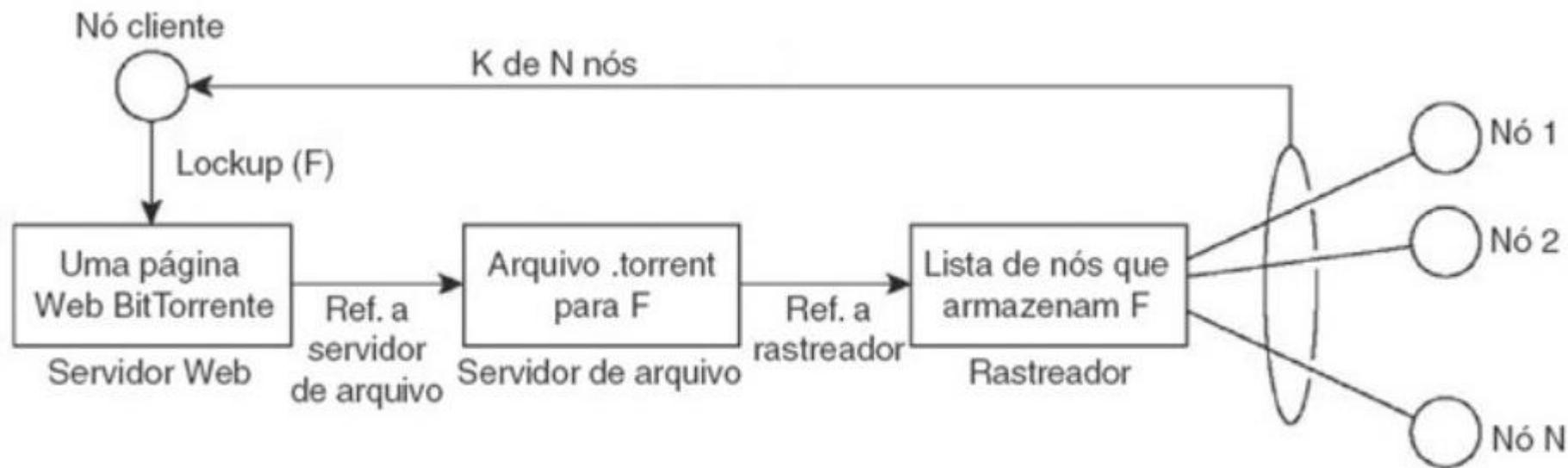
- Distribuição de conteúdo

# BitTorrent

Sistemas Distribuídos Colaborativos

Inicialmente um sistema de procura Cliente/Servidor

Subsequentemente, os nós juntam-se para dar um mecanismo descentralizado de colaboração



# WhatsApp

## Understanding WhatsApp architecture

