

# GERENCIAMENTO DE REPLICAÇÕES

DCE540 - Computação Paralela e Distribuída

Atualizado em: 15 de outubro de 2021

Iago Carvalho

Departamento de Ciência da Computação



# COMO, QUANDO E ONDE REPLICAR

Dois problemas devem ser resolvidos na tomada de decisão da replicação

1. Onde os servidores de réplicas estão localizados
2. Onde os dados replicados estão localizados

O primeiro tem como objetivo estabelecer o local físico (ou geográfico) dos servidores

O segundo diz respeito a qual dado será replicado em cada um dos servidores

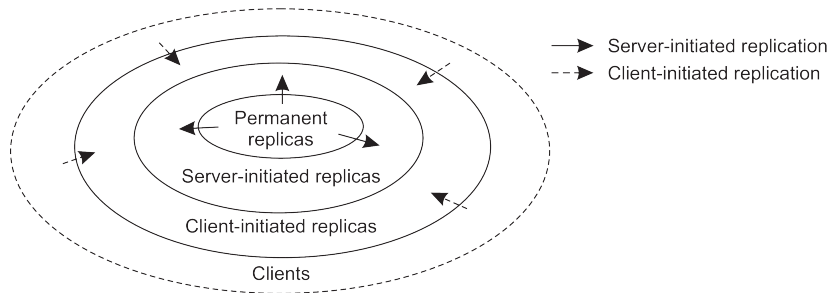
Este era um problema grave a alguns anos atrás

Entretanto, com o avanço da velocidade (e estabilidade) das conexões, isto deixou de ser um problema

Atualmente, existem heurísticas capazes de computar boas localizações para servidores em questão de segundos

- O problema pode ser resolvido praticamente em tempo real

# REPLICAÇÃO DE CONTEÚDO E LOCALIZAÇÃO



# RÉPLICAS PERMANENTES

Estas são as réplicas iniciais do sistema distribuído

- Normalmente é um pequeno conjunto
- Estas réplicas são permanentes e nunca são removidas

Representa a cópia original dos dados

- Seus espelhos através da internet

# RÉPLICAS INICIADAS PELOS SERVIDORES

Muito comum hoje em dia com ambiente em nuvem e DevOps

Um servidor pode criar um número de réplicas adicionais para suportar um grande número de requisições

- Estas réplicas normalmente são temporárias
- Réplicas são *read-only*

Esta estratégia muitas vezes é utilizada quando ocorrem eventos específicos

- Por exemplo, venda de ingressos para festivais

## RÉPLICAS INICIADAS PELOS CLIENTES

Este tipo de replicação de dados não pode ser controlado pelo sistema distribuído

Uma réplica iniciada pelo cliente é um dado (ou conjunto de dados) *cache*

- Localizado na própria máquina do cliente ou em algum dispositivo de rede muito próximo a ele
- Muito útil quando o cliente faz diversas requisições a um mesmo conjunto de dados

Este tipo de réplica tem um tempo de vida limitado e é de responsabilidade do cliente atualizar seus dados

A primeira coisa a se preocupar é com quais dados propagar

No geral, existem três opções

1. Propagar uma notificação da atualização dos dados
  - Protocolo de invalidação
2. Propagar o dado alterado (por completo)
3. Propagar somente a alteração que o dado sofreu
  - Replicação ativa



Os updates serão enviados pelo servidor ou serão requisitados pelos clientes?

Esquema baseado em *push*

- Os dados são replicados a partir do servidor
- Utilizado quando deseja-se manter uma consistência apertada
- Útil quando a maioria das operações dos clientes são de leitura
- Maior custo de rede

Esquema baseado em *pull*

- Clientes tem que requisitar a atualização dos dados
- Útil junto com protocolos de invalidação
- Tempo de resposta relativamente alto
  - Pode ser necessário atualizar os dados ao fazer uma requisição

# UNICAST OU MULTICAST

Como transferir os dados?

## *Unicast*

- Atualização é realizada em uma comunicação direta entre dois dispositivos
- Necessárias  $N$  mensagens para atualizar todo o sistema distribuído
- Útil no caso de esquema de atualização baseado em *pull*

## *Multicast*

- Atualização é realizada de um dispositivo para diversos outros
- Um único processo de comunicação é necessário para atualizar quantos servidores forem necessários
- Útil no caso de esquema de atualização baseado em *push*