



Particionamento

Amanda Gabrielle Polanski
Maio/2018



Agenda

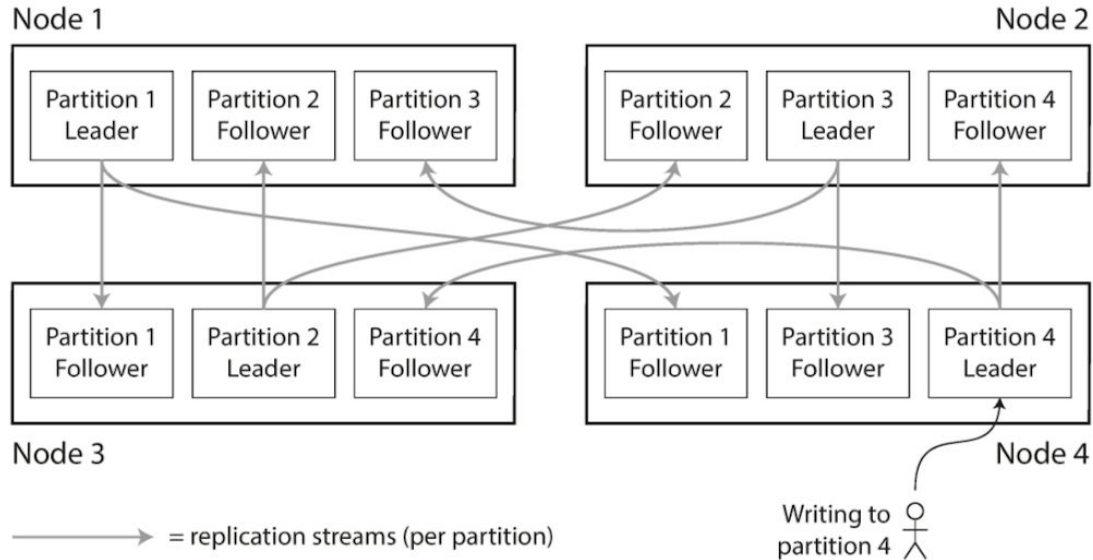
- Tipos de particionamento
- Como evitar hot spots
- Estratégias de balanceamento
- Roteamento



Introdução

- Particionamento para banco de dados grandes
- Tem como objetivo escalabilidade
- Arquitetura shared-nothing
 - Arquitetura onde há várias máquinas rodando a base de dados
 - Podemos chamá-las de nodos

Particionamento e Replicação



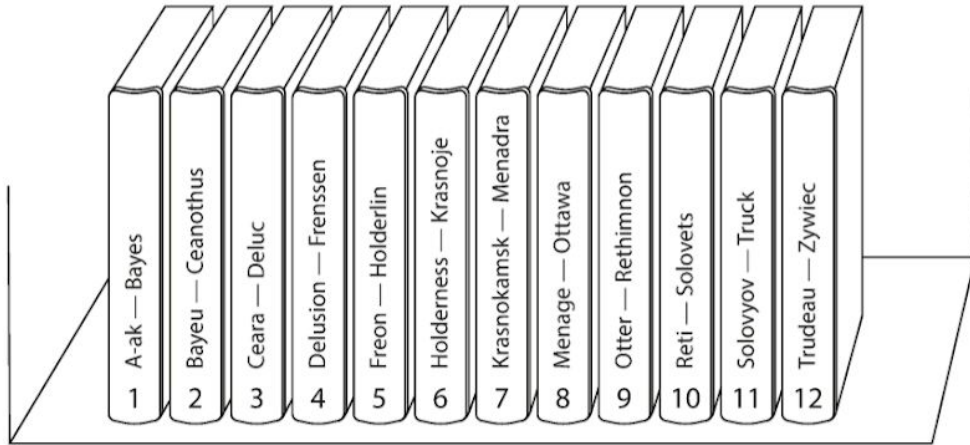


Particionamento por key-value

- **Hot spot:** uma partição sobrecarregada com registros
- Acesso de registros pela chave primária
 - Ex: dicionário



Particionamento por key-range



- Partições limitadas
- Problema: hot spot



Particionamento com Hash

- Determina uma partição para cada chave
- Distribuição justa das chaves
- Ordem acaba sendo perdida



Evitando hot spots

- Hash ajuda, mas ainda há falhas
 - Ex: twiter de famosos
- Colocar um valor random na chave
 - Será distribuído entre várias partições
 - Problema: só pode ser feito com poucas chaves pois gera overhead



Particionamento e índices secundários

- Índices secundários são usados para contar aparições
- Usado para modelar dados
- Indexação por documento e indexação por termo



Particionamento e índices secundários

Documento

- Conhecida por *local index*
- Cada partição é separada
- Informações não estão organizadas pelo índice

Termo

- Conhecida por *global index*
- Há uma tabela de índices global
- O termo procurado determina a partição do índice
- Leitura é mais rápida
- Escrita é mais complicada
- Updates são assíncronos



Balanceando partições

Mudanças que ocorrem no banco de dados:

- Quantidade de queries aumenta
- Quantidade de dados aumenta
- Caso uma máquina desligue, precisa ter uma para substituir



Balanceando partições

- Todas essas mudanças precisam que dados mudem de um nodo para outro
- Objetivos do balanceamento
 - Carga de dados igualmente distribuída
 - Permitir escrita e leitura
 - Mexer apenas o necessário



Estratégias de balanceamento

- Número fixo de partições
- Particionamento dinâmico
- Particionamento proporcional aos nodos
- Balanceamento manual ou automático?



Número fixo de partições

- São criadas muito mais partições do que existem nodos
- Divide as partições existentes
 - Muitas partições grandes: custo alto para balancear
 - Partições pequenas: overhead de informação



Particionamento dinâmico

- Proporcional ao tamanho do banco de dados
- Key-range e Hash
- Partição grande: divide os registros com outra partição
- Nodo grande: cria-se um novo nodo
- **Desvantagem:** quando ainda há apenas uma partição



Particionamento proporcional aos nodos

- Proporcional à quantidade de dados
- Hash
- Número fixo de partições por nodo
- O número de nodos fica estável



Balanceamento manual ou automático?

- Manual
 - Sujeito a falhas humanas
- Automático
 - Imprevisível
 - Processo caro
 - Baixa performance



Métodos de solicitação

Como um cliente pode saber em qual nodo fazer uma solicitação?

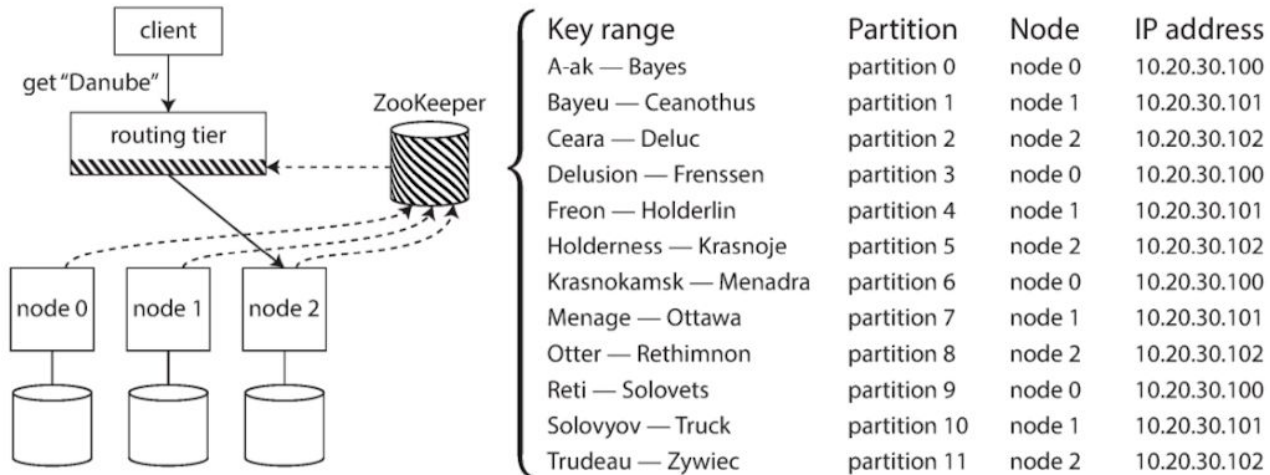
- Permitir que acessem qualquer nodo
- Camada de roteadores
- Prévio conhecimento do mapa de particionamento



Solicitação de roteamento

- Mapa com todas as partições
- Os nodos mantêm a camada de roteamento atualizada
- Usuário ainda precisa do IP para se conectar

Solicitação de roteamento



////// = the knowledge of which partition is assigned to which node



Execução paralela de queries

- PPM: processamento paralelo em massa
- Queries grandes são otimizadas



Exercícios

1. Explique as diferenças entre os processos de particionamento, mostrando suas vantagens e desvantagens
2. Para cada tipo de balanceamento, dê um exemplo de situação em que poderia ser utilizado