Famílias de Bancos de Dados NoSQL

Prof. Klayton R. Castro

IDP

March 14, 2024

Teorema CAP

- Explora o trade-off entre consistência, disponibilidade e tolerância à partição em sistemas distribuídos.
- Afirma que é impossível para um sistema distribuído garantir simultaneamente as três propriedades.

Consistência

- Todos os nós veem os mesmos dados ao mesmo tempo.
- Equivalente à ter uma única cópia atualizada dos dados.

Disponibilidade

- Cada solicitação recebe uma resposta, sem garantia de que contém a versão mais recente dos dados.
- O sistema está sempre online.

Tolerância à Partição

- O sistema continua a operar, apesar de haver "partições" (falhas de comunicação).
- As partições não impedem o trabalho do sistema como um todo.

Escolhendo Duas

- Sistemas devem decidir qual das três propriedades é menos crítica para sacrificar.
- Decisões dependem do tipo de aplicação e dos requisitos de negócio.

Propriedades ACID

- Atomicidade, Consistência, Isolamento, Durabilidade.
- Focado em confiabilidade em sistemas de banco de dados.

Propriedades BASE

- Basicamente disponível, Estado suave, Eventual consistência.
- Mais flexível que ACID, adequado para sistemas distribuídos de grande escala.

ACID vs. BASE

- ACID garante segurança em transações, enquanto BASE oferece maior escalabilidade.
- Escolha depende do equilíbrio desejado entre consistência e disponibilidade.

Introdução às Famílias de NoSQL

- NoSQL oferece flexibilidade, escalabilidade e desempenho.
- Adequado para grandes volumes de dados e estruturas de dados não relacionais.

Banco de Dados NoSQL: Documento

- Armazena dados em documentos (JSON, XML, etc.).
- Útil para dados hierárquicos.
- Exemplos: MongoDB, Couchbase.

Banco de Dados NoSQL: Chave-Valor

- Estrutura simples, armazenando dados como um par chave-valor.
- Rápido acesso a dados através da chave.
- Exemplos: Redis, DynamoDB.

Banco de Dados NoSQL: Colunar

- Armazena dados por colunas ao invés de linhas.
- Ideal para análise de grandes volumes de dados.
- Exemplos: Cassandra, HBase.

Banco de Dados NoSQL: Grafo

- Armazena dados em nós e arestas, representando entidades e seus relacionamentos.
- Ideal para dados interconectados.
- Exemplos: Neo4j, Amazon Neptune.

Conclusão

- A escolha da família NoSQL depende da natureza dos dados e dos requisitos do sistema.
- O entendimento do Teorema CAP é crucial para o design de sistemas distribuídos.
- As propriedades ACID e BASE refletem diferentes abordagens para a consistência de dados e disponibilidade em bancos de dados.

Desafio do Grupo 1

Desafio: Criar um sistema eficiente que minimize os tempos de resposta para usuários acessando um serviço online sob demanda variável.

- Foco na otimização do armazenamento e recuperação de dados frequentemente acessados.
- Garantir que o conteúdo popular esteja rapidamente disponível durante picos de tráfego.

Desafio do Grupo 2

Desafio: Projetar uma solução para o gerenciamento e distribuição em tempo real de conteúdo diversificado para uma plataforma digital de notícias.

- Permitir atualizações ágeis e flexíveis de conteúdo.
- Prover meios eficazes para categorizar e pesquisar informações complexas.

Desafio do Grupo 3

Desafio: Desenvolver uma abordagem para coletar, armazenar e analisar grandes volumes de dados de séries temporais de dispositivos IoT.

- Facilitar a análise detalhada de tendências ao longo do tempo.
- Ajudar na previsão e detecção de eventos significativos.

Desafio do Grupo 4

Desafio: Construir um sistema para modelar e resolver questões de logística e otimização de rotas.

- Considerar variáveis como distâncias, custos e capacidades.
- Produzir soluções eficientes para o planejamento e execução de rotas de entrega.