NOSQL

ROTEIRO

Introdução

Consistência

Escalabilidade

Teorema CAP

Taxonomia

NoSQL Databases

Conclusão

INTRODUÇÃO

INTRODUÇÃO

A quantidade de dados gerados, armazenados e processados atingiu escalas inéditas com a Web 2.0

Exige soluções além dos bancos de dados relacionais

Necessidade de novos requisitos de escabilidade e disponibilidade

INTRODUÇÃO

Organizações menores e a comunidade de software livre

Soluções de Banco de dados não relacionais

NoSQL ("Not only SQL")

Termo genérico para uma classe definida de banco de dados não-relacionais que rompe uma longa história de bancos relacionais com propriedades ACID.

NOSQL

NoSQL surgiu como uma solução para questão de escabilidade no armazenamento e processamento de grandes volumes de dados na Web 2.0

Apresentam as seguintes características:

- Não relacional;
- Distribuído;
- De código aberto;
- Escalável horizontalmente;
- Ausência de esquema ou esquema flexível;
- Suporte à replicação nativo;
- Acesso via APIs simples

NOSQL

Fatores que favoreceram o seu surgimento:

- **Natureza dos dados:** Constatação de que os dados web não são estruturados;
- **Importância do paralelismo para o processamento de grande volume de dados:** Uso de hardware comum e barato para operação dos sistemas
- **Distribuição do sistema em escala global:** O software deve ser robusto o suficiente para tolerar constantes e imprevisíveis falhas de hardware e de infra-estrutura

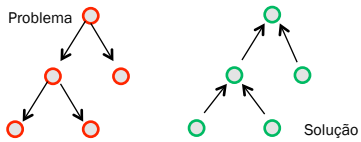
NOSQL

Conceitos que permitem a implementação das funcionalidades do NoSQL:

- Map/Reduce:
  - Suporta o manuseio de grande volume de dados distribuídos ao longo dos nós na rede;
  - Map step: Problemas são divididos em subproblemas que são distribuídos entre outros nós na rede;
  - Reduce step: Subproblemas são resolvidos e são passados para o nó pai.

NOSQL

- Map/Reduce:



- Eventual Consistency:
  - **Teorema CAP (Consistency, availability, partition tolerance):**
    - Fundamental para que algumas das soluções consigam atingir níveis maiores de escalabilidade;
    - As propriedades ACID criam dificuldades ao se distribuir o banco de dados.

NOSQL

- Eventual Consistency:
  - **Teorema CAP (Consistency, availability, partition tolerance):**
    - Em um dado instante, só é possível garantir duas dessas três propriedades;
    - Sistemas optam por disponibilidade e tolerância a partição.
  - **BASE em vez de ACID**
    - Basicamente disponível, estado leve e consistente em momento indeterminado;
    - Indica que deve-se planejar um sistema de forma a tolerar inconsistências temporárias quando se quer priorizar disponibilidade.

NOSQL

- Consistent hashing:
  - Suporta mecanismos de armazenamento e recuperação em banco de dados distribuídos onde o número de nós na rede está sujeito a mudanças;
  - Evitam uma extensa migração de dados resultantes dessas mudanças.
- Multiversion concurrency control (MVCC):
  - Suporta transações **simultâneas** no banco de dados;
  - Oposto ao clássico gerenciamento de transações que faz uso de locks;
  - Permite **operações em paralelo** de escrita e leitura.

NOSQL

- Vector clocks:
  - São utilizados para gerar uma ordenação dos eventos em um sistema;
  - Como os nós da rede podem atualizar um conjunto de dados de forma desordenada, qualquer nó deve ser capaz de determinar a versão atual de um certo conjunto de dados.

TAXONOMIA

Tipos mais comuns de banco de dados NoSQL:

- Banco de Dados Orientado a Documentos
  - Documentos são coleções de atributos (que podem ser multivalorados) e valores;
  - Em geral não possuem esquema;
  - Boa opção para armazenamento de dados semi-estruturados;
  - Um documento pode estar embutido em outro, podendo criar problemas de inconsistência;
    - Pode ser evitado através da normalização.

TAXONOMIA

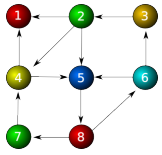
- Armazém chave-valor
  - Tabelas de hash distribuídas;
  - Armazenam objetos indexados por chaves, e possibilitam a sua busca a partir dessas chaves;
  - Ligado ao problema da localização de objetos em redes peer-to-peer;
  - Vantagem na forma de lidar com grande volume de dados em um ambiente distribuido.

TAXONOMIA

- Banco de Dados de famílias de colunas de muitos registros
  - Utilizado quando se quer ler algumas poucas colunas;
  - Guardam os dados contiguamente por coluna;
  - A escrita de um novo registro é bem mais custosa do que em um banco de dados tradicional;
  - Mais interessantes para *processamento analítico online (OLAP)*.

TAXONOMIA

- Banco de Dados de Grafos
  - Está diretamente relacionado a um modelo de dados estabelecido (modelo de grafos);
  - Representa os dados e/ou o esquema dos dados como grafos dirigidos;
  - Fazem uso de conceitos de grafos (caminhos, vizinhos e sub-grafos);



TAXONOMIA

- Banco de Dados de Grafos
  - Dá suporte ao uso de *restrições sobre os dados*;
  - Deve ser utilizado quando informações sobre a interconectividade ou a topologia dos dados é **mais importante** (ou tão importante quanto) os dados propriamente ditos;
  - Técnicas de consulta permitem a recuperação de entidadess e de suas relações baseados em algoritmos tais como Breadth First Search, Depth First Search, determination of Hamilton cycles.

NOSQL DATABASES

Serão comparados através dos seguintes aspectos:

- Modelo de Dados
- Modelo de Consultas
- Sharding
- Replicação
- Consistência
- Arquitetura
- Manipulação de falhas

NOSQL DATABASES

1. Dynamo
   - É um banco de dados distribuídos do tipo armazém chave-valor;
   - Utilizado internamente pela *Amazon*;
   - Provê uma simples API de consultas;
   - Cada nó dentro da rede tem as mesmas responsabilidades;
   - Faz uso da **replicação otimista com MVCC**.

NOSQL DATABASES

- Dynamo
  - Model de consulta:
    - Get(key);
    - Put(key, context, value)
  - Sharding
    - Os dados são particionados como uma variante do consistent hashing.

NOSQL DATABASES

- Dynamo
  - Replicação
    - Dynamo usa o MVCC para permitir a sincronização entre as suas réplicas;
    - Cada atualização gera uma nova versão atualizada;
    - Utiliza o vector clocks para controle de versão atualizada.

NOSQL DATABASES

- Amazon S3 Simple Storage Service
  - É um banco de dados distribuídos do tipo armazém chave-valor;
  - Não é exclusivo da Amazon;
  - Pode armazenar até 5GB de dados;
  - Organiza os dados em segmentos;
  - O cliente pode especificar um local geográfico para guardar o segmento, e também pode especificar as restrições de acesso do mesmo;
  - Todos os dados são replicados.

NOSQL

- Amazon S3 Simple Storage Service
  - Modelo de consulta

Create Bucket:	Creates a Bucket with a unique name.
Delete Bucket:	Deletes a Bucket.
List Keys:	Lists all keys stored inside a Bucket.
Read:	Reads the data belonging to a unique key from a Bucket.
Write:	Writes the data associated with a unique key into a Bucket.
Delete:	Deletes an key and its data from the storage.

NOSQL

- Amazon S3 Simple Storage Service
  - Consistência
    - Uma operação de escrita é atômica;
    - Não provê um mecanismo de lock.

NOSQL

- SimpleBD
  - Armazenamento chave-valor;
  - Cloud Service da Amazon;
  - Fácil de usar: livre de esquema
  - Apresenta limitações;

NOSQL

- SimpleBD
  - Modelo de Dados
    - Dados organizados em domínios;
    - Domínios podem ter muitos itens;
    - Itens podem ter muitos atributos;
    - String;
    - Chave do item é seu nome;

NOSQL

- SimpleBD
  - Modelo de Consultas
    - Comando select (como o do SQL):

```
select lista_resultado or count (*) from dominio [where] [ordenação] [limite];
```

    - Não é permitido Join;
    - Comparações: apenas entre item e valor constante;

NOSQL

- SimpleBD
  - Sharding:
    - Não suporta sharding automático (particionamento de dados tratado na camada de aplicação);
  - Replicação:
    - Todos os dados são replicados (segurança e performance);
  - Consistência:
    - Consistência eventual;
    - Não usa MVCC.

NOSQL

- BigTable
  - Uma das soluções da Google pra lidar com a quantidade de dados;
  - Implementação não disponível para o público;
  - Utilizado no Google App Engine;

NOSQL

- BigTable
  - Modelo de Dados:
    - Chave-valor: chave consiste de uma tupla (chave da linha, chave da coluna e timestamp);
    - Linha: chave da linha e várias colunas;
    - Colunas: organizadas por famílias de colunas e conteúdo ordenado pelo timestamp;
    - String e int64

NOSQL

- BigTable
  - Modelo de Consultas:
    - Indexação pela tupla (linha, coluna, timestamp);
    - Não suporta Join;
    - Biblioteca cliente em C++;
    - Scanner: aplicação define filtros para as chaves das linhas, as famílias das colunas e os timestamps, além de permitir interação com os resultados da consulta;
    - Resultados ordenados pelas chaves das linhas.

NOSQL

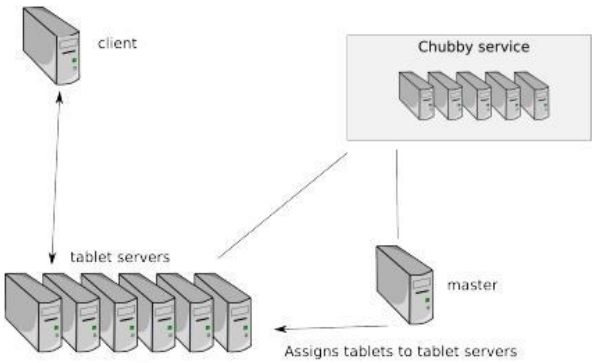
- BigTable
  - Sharding:
    - Particionamento em vários tablets;
    - Cada tablet está associado a um servidor;
    - Um servidor mestre guarda toda meta-informação dos tablets e sua localização, além de fazer associações de novos a tablets a servidores desocupados;

NOSQL

- BigTable
  - Replication:
    - Não pode haver replicação;
    - Utiliza GFS (Google File Service) para lidar com a replicação no nível de arquivos;
  - Consistência:
    - Provê forte consistência;
    - Operações podem ser feitas apenas dentro de uma mesma linha;

NOSQL

- BigTable
  - Arquitetura:



NOSQL

- BigTable
  - Manipulação de Falhas:
    - Falha um servidor com tablet:
      - Realocação de tablets para outro servidor (caso o seu tenho falhado e perdido o lock);
      - Log de todas as operações de escrita;
    - Falha do servidor mestre:
      - Eleição de novo servidor mestre;
    - Falha do Chubby:
      - Cluster com 5 servidores e necessidade de no mínimo 3 para associar um lock a um novo tablet.

NOSQL

- BigTable
  - Manipulação de Falhas:
    - Falha um servidor com tablet:
      - Realocação de tablets para outro servidor (caso o seu tenho falhado e perdido o lock);
      - Log de todas as operações de escrita;
    - Falha do servidor mestre:
      - Eleição de novo servidor mestre;
    - Falha do Chubby:
      - Cluster com 5 servidores e necessidade de no mínimo 3 para associar um lock a um novo tablet.

NOSQL

- MongoDB
  - Sem esquemas;
  - Orientado à documentos (BSON);
  - Open source;
  - Pode ser utilizado para armazenar grandes dados binários (fotos/vídeos);
  - C++.

NOSQL

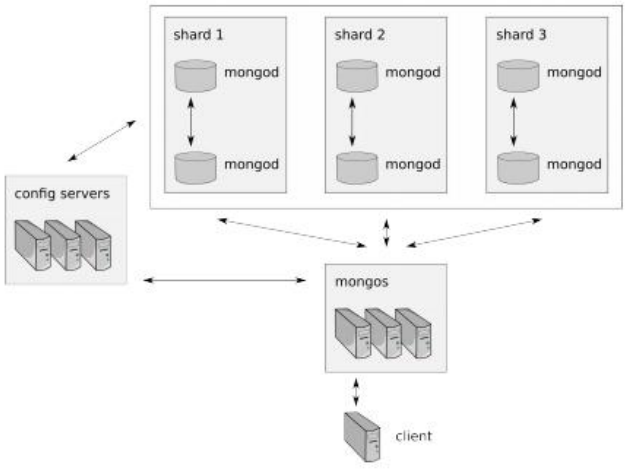
- MongoDB
  - Modelo de Dados:
    - Armazena documentos BSON;
    - Suporta estruturas aninhadas;
    - Suporta modificações simples de atributos (in-place);
    - Chave primária: ID do documento
    - Cada campo do documento pode ser indexado pelo desenvolvedor;
    - Coleções;

NOSQL

- MongoDB
  - Modelo de Consulta:
    - Linguagem de consulta X Invocação de métodos;
    - Consultas podem ser alteradas dinamicamente;
    - Consulta pode abranger todos os documentos de uma coleção, inclusive os aninhamentos;
    - Operações comparativas, condicionais, lógicas, de ordenação por vários atributos e de “group by”;
    - Permitida uma agregação por consulta;
    - Agregação mais complexa, apenas com uso de Map Reduce;

NOSQL

- MongoDB
  - Arquitetura:



NOSQL

- MongoDB
  - Sharding:
    - Shard keys (index) particionam as coleções de documentos em *shards*;
    - Organizado de forma ordenada em *chunks*;
    - *Chunks* são usados pelo MongoDB para rebalancear os *shards;*

NOSQL

- MongoDB
  - Consistência:
    - Não tem controle de concorrência de versão;
    - Não tem gerenciamento de transações;
    - Consistência eventual;

NOSQL

- MongoDB
  - Manipulação de Falhas
    - Se um nó falha, pode haver perda de informação e geração de arquivos corrompidos;
    - Se o nó faz parte de um par de replicação, seu par assume as tarefas de ambos;
    - Se um dos servidores de configuração falha, o MongoDB não poderá executar tarefas de realocação dos *chunks* ou de migração, até que volte a funcionar ou seja substituído.

NOSQL

- Comparação dos BDs NoSQL
  - Modelo de relacionamentos tratado apenas na camada de aplicação;
  - Sorting
    - Atributo único e múltiplos atributos
  - Consultas com range
    - Atributo único e múltiplos atributos

NOSQL

- Comparação dos BDs NoSQL
  - Agregação
    - Simple DB – suporta somente contagem
    - Mongo DB – suporta “group by”
    - Big Table e Mongo DB – Map Reduce
  - Durabilidade
    - Replicação em pelo menos um HD depois do commit
    - Replicação em diferentes máquinas
    - Integridade

NOSQL

- Conclusão
  - Tecnologia recente
  - Diferentes BDs para diferentes aplicações
  - Importância do Teorema CAP
  - Utilização no Facebook, Twitter e Foursquare
  - É esperar pra ver!

NOSQL

- Dúvidas?

Amora Taveira (acat) Suzana Fragoso (smpf)