

Armazenamento de Dados Abertos com NoSQL: Um estudo de caso com Dados do Bolsa Família e NoSQL Cassandra

Jorge Luiz Andrade

Universidade de Brasília

jorgeluizandrade@outlook.com

5 de dezembro de 2017

Introdução

Introdução

Introdução

Problema

Banco de Dados Relacionais podem não apresentar um desempenho satisfatório ao operar grandes volumes de dados

Introdução

Problema

Banco de Dados Relacionais podem não apresentar um desempenho satisfatório ao operar grandes volumes de dados

Hipótese

O uso de múltiplas máquinas em um ambiente Cassandra distribuído pode oferecer um melhora do desempenho que justifique sua utilização na análise de dados abertos.

Introdução

Objetivos

Comparar o desempenho de um banco Cassandra para inserções e consultas em diferentes tamanhos de *cluster* e de volumes de dados;

- ▶ Desenvolver uma aplicação para inserção e busca dos dados do Bolsa Família;
- ▶ Realizar testes de inserção e busca com diferentes configurações;
- ▶ Comparar o desempenho do Cassandra nas diferentes situações;

Bancos Relacionais

- ▶ Proposto em 1970 por Edgar Codd;
- ▶ Conjunto de relações entre tuplas;

Propriedades ACID

- ▶ Atomicidade;
- ▶ Consistência;
- ▶ Isolamento;
- ▶ Durabilidade;

Garantem a validade do esquema, mas sacrificam desempenho e disponibilidade.

Normalização

- ▶ **1FN**: Cada coluna deve guardar apenas uma informação (valores atômicos);
- ▶ **2FN**: Atributos não-chave devem depender integralmente da chave primária da tabela;
- ▶ **3FN**: Atributos não-chave não podem ser determinados por outros atributos não-chave;

Modelos relacionais possuem restrições, como as propriedades ACID e Normalização, gerando problemas de escalabilidade e rigidez de esquema.

NoSQL

- ▶ Termo utilizado pela primeira vez em 1998(Strozzi NoSQL)
- ▶ Google Bigtable(2006) e Amazon's Dynamo(2007)

Teorema CAP

- ▶ Proposto em 2000 por Eric Brewer, define limitações em sistemas distribuídos;
- ▶ Revisado em 2012;
- ▶ Consistência;
- ▶ Disponibilidade;
- ▶ Tolerância a partições

Chave-Valor

Consiste em uma tabela *hash*, com consultas a um valor a partir de uma chave.

- ▶ Berkeley DB;
- ▶ Amazon DynamoDB;

Documentos

Acesso à um documento de esquema flexível a partir de uma chave.

- ▶ CouchDB;
- ▶ MongoDB;

Grafos

Dados altamente conectados, com consultas baseadas em relacionamentos.

- ▶ Neo4j
- ▶ OrientDB

Colunas

Dados armazenados em famílias de colunas. Possui esquema flexível, permitindo a modificação de colunas a qualquer momento.

- ▶ HBase
- ▶ **Cassandra**

Programa Bolsa Família

Programa de transferência de renda criado em 2003.

Em 2016, atendia 13,9 milhões de famílias, que recebiam uma média de R\$ cada, totalizando R\$27,4 bilhões.

- ▶ Dados disponibilizados no Portal da Transparência;
- ▶ Arquivos mensais em formato .csv;

Dados Utilizados

Foram utilizados um total de trinta arquivos, referentes aos meses de Julho de 2014 a Dezembro de 2016. Os arquivos totalizam 16Gib de tamanho e cerca de 14 mil registros.

Campo	Tipo	Utilizado
UF	Text	Sim
Código SIAFI Município	Int	Sim
Nome Município	Text	Sim
Código Função	-	Não
Código Subfunção	-	Não
Código Programa	-	Não
Código Ação	-	Não
NIS Favorecido	Bigint	Sim
Nome Favorecido	Text	Sim
Fonte-Finalidade	Text	Sim
Valor Parcela	Double	Sim
Mês Competência	Timestamp	Sim

Modelo de Dados

- ▶ Fator de replicação de 1 (sem tolerância a falhas);
- ▶ *SimpleStrategy* (*datacenter* único);
- ▶ Criação do ambiente com uso de CQL;

Código 1: Código CQL para criação do keyspace

```
CREATE KEYSPACE bolsa_familia WITH replication
  ↪ = {'class': 'SimpleStrategy', '
  ↪ replication_factor': 1};
```

Código 2: Código CQL para criação da tabela

```
CREATE TABLE bolsa_familia.dados (uf TEXT,
  ↪ periodo TIMESTAMP, valor DOUBLE,
  ↪ nis_favorecido BIGINT, cod_municipio INT,
  ↪ fonte TEXT, nome_favorecido TEXT,
  ↪ nome_municipio TEXT, PRIMARY KEY(
  ↪ nis_favorecido, periodo, valor));
```

Arquitetura do Ambiente

- ▶ *Cluster* composto por seis máquinas Intel i5-4570 3.20GHz, 16GB de RAM, disco rígido de 500GB, com sistema operacional Ubuntu;
- ▶ Cliente Cassandra versão 3.0.4;
- ▶ Configuração do arquivo *cassandra.yaml*;

Código 3: Configuração cassandra.yaml

```
cluster_name: 'BolsaFamilia_Cluster_C2M_FR1'

num_tokens: 256

partitioner: org.apache.cassandra.dht.
    ↪ Murmur3Partitioner

seed_provider:
- class_name: org.apache.cassandra.locator.
    ↪ SimpleSeedProvider

parameters:
```


Carga dos Dados

A aplicação desenvolvida realiza a filtragem dos campos e tratamento dos valores:

- ▶ Remoção do separador de milhares(,) em Valor Parcela;
- ▶ Alteração do padrão de data de MM/AAAA para DD/MM/AAAA;

Foi realizada a carga com dois volumes de dados, correspondentes a dezoito e trinta meses do programa Bolsa Família.

Tabela: Volume de dados

Carga	Tamanho
18 meses	8,79 GB
30 meses	14,69 GB

Carga dos Dados

Inserção realizada com uso do *driver* da *Datastax*, por meio de query CQL, tendo seus parâmetros substituídos.

Código 4: Código CQL para inserção

```
INSERT INTO bolsa_familia.dados (uf,  
    ↪ cod_municipio, nome_municipio,  
    ↪ nis_favorecido, nome_favorecido, fonte,  
    ↪ valor, periodo) VALUES (?, ?, ?, ?, ?, ?,  
    ↪ ?, ?)
```

Tempos de Inserção

Tabela: Tempos de Inserção

Volume	2 nós	4 nós	6 nós
18 meses	1h	55m	52m
30 meses	2h31m	2h19m	2h06m

Tabela: Comparativo

Volume	2 para 4 máquinas	4 para 6 máquinas	Média
18 meses	8,70%	4,22%	6,46%
30 meses	8,13%	8,94%	8,54%

Tempos de Inserção

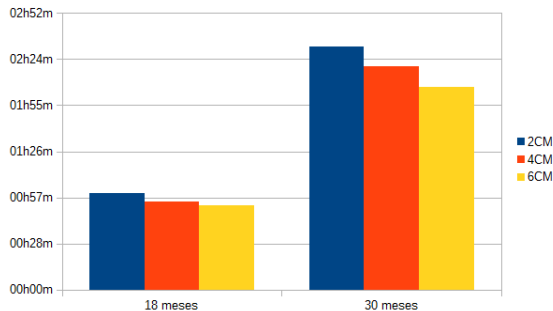


Figura: Tempos de Inserção

Consultas

As consultas também foram realizadas por meio do *driver* da *Datastax*. Foram realizadas 30 consultas, buscando um registro específico por chave primária de forma aleatória.

Código 5: Código CQL para consulta

```
SELECT * FROM dados WHERE nis_favorecido =  
    ↪ 00020915229557 AND periodo = '2014-07-01'  
    ↪ AND valor = 147.00
```

Tempos de Consulta

Tabela: Tempos de Consulta

Volume	2 nós	4 nós	6 nós
18 meses	10,26 s	1,95 s	1,73 s
30 meses	12,98 s	4,38 s	1,43 s

Tabela: Comparativo

Volume	2 para 4 máquinas	4 para 6 máquinas	Média
18 meses	81,00%	11,41%	46,20%
30 meses	66,21%	67,48%	66,85%

Tempos de Consulta

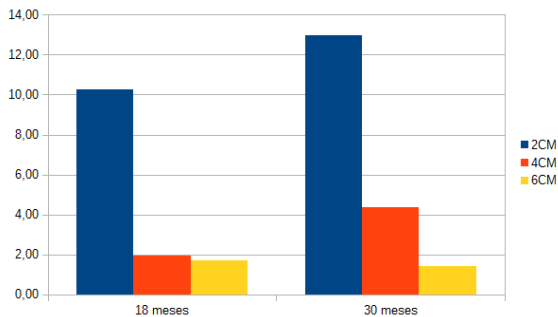


Figura: Tempos de Consulta

Resultados

Comparação do aumento do número de máquinas:

- ▶ Melhora média de 7,5% na inserção dos dados;
- ▶ Melhora média de 56,53% na busca dos dados;

Trabalhos futuros

- ▶ Isolamento da rede no ambiente utilizado;
- ▶ Comparação com outros bancos;
- ▶ Implementar diferentes modelagens no banco Cassandra;

Bibliografia