

UNIVERSIDADE DE BRASÍLIA

Faculdade do Gama

Sistemas de Banco de Dados 2

Tecnologias de Banco de Dados (TI-BD)

Banco de Dados de Colunas

Ana Carolina Carvalho da Silva

190063441

Brasília, DF

2019

Banco de Dados de Colunas

1. Definição

Banco de Dados de Colunas (ou Verticais) é o banco de dados em que o método de armazenamento é coluna por coluna ao invés de linha por linha, ou seja, os dados são armazenados em colunas individuais. Assim, o sistema busca apenas os atributos que são solicitados em vez de ler uma linha com informações a mais do que as necessárias.

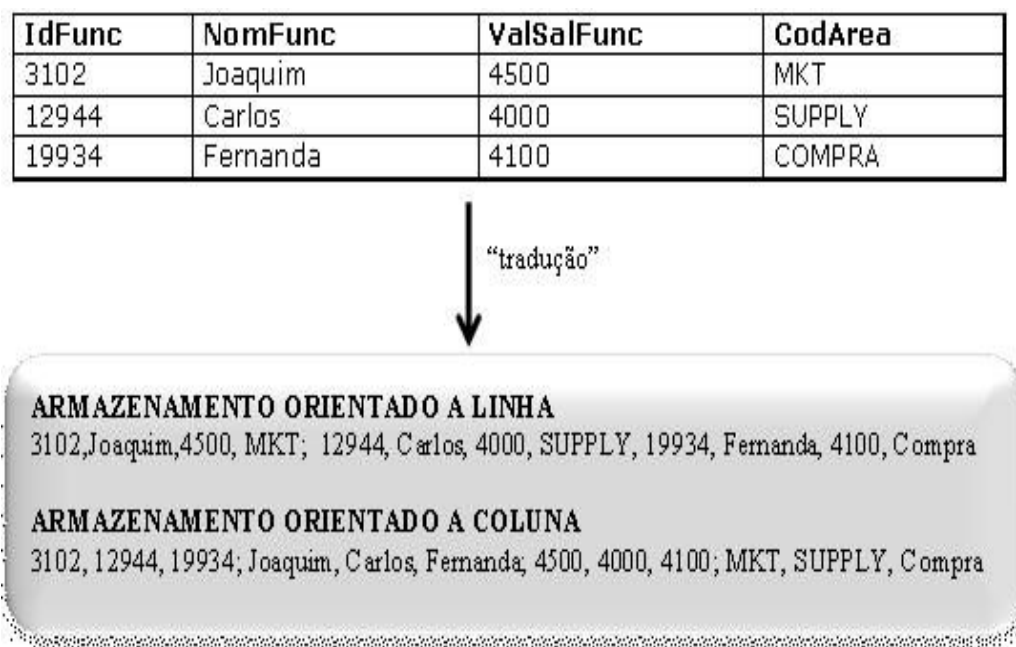


Figura 1: Exemplo de armazenamento orientado em Colunas e Linhas.

Fonte: DevMedia (2008)

1.1 Descrição do armazenamento

Segue uma descrição de como o armazenamento em coluna é eficiente e como a recuperação de dados na memória é otimizada com esse tipo de construção.

1.1.1 Armazenamento em banco de dados orientados a linhas

SSN	Name	Age	Addr	City	St
101259797	SMITH	88	899 FIRST ST	JUNO	AL
892375862	CHIN	37	16137 MAIN ST	POMONA	CA
318370701	HANDU	12	42 JUNE ST	CHICAGO	IL

101259797 892375862 318370701	468248180 378568310 231346875 317346551 770336528 277332171 455124598 735885647 387586301
-----------------------------------	---

Block 1

Figura 3: Exemplo de armazenamento orientado em Colunas.

Fonte: AWS (2019)

Ao usar o armazenamento de colunas, cada bloco de dados irá armazenar valores de uma única coluna de várias linhas.

Na figura 2, cada bloco de dados possui valores de campo da coluna até três vezes mais do que muitos outros registros de armazenamento ao qual é orientado a linhas. E essa informação nos leva a considerar que ler o mesmo número de valores do campo de coluna para um mesmo número de registros exige um terço das operações de entrada e saída, se comparado com o armazenamento compatível feito em linhas.

2. Objetivos

Os sistemas de banco de dados de colunas têm como principal objetivo a melhoria de desempenho e de armazenamento. O armazenamento orientado a colunas para tabelas do banco de dados é um fator importante para a performance de consulta analítica, pois ele reduz expressivamente os requisitos gerais de entrada e saída de disco e diminui a quantidade de dados que você precisa carregar do disco.

O motivo é que uma consulta leva menos tempo de tráfego de dados entre o disco e a memória e pode buscar um grande números de informações relevantes através de uma operação de leitura de dados. Já o banco de dados de linhas recupera as informações desejadas mas também todos os atributos

que não são relevantes para a consulta em questão. O que ocasionou a grande demanda que os bancos orientados a linha não estavam mais conseguindo suprir.

Assim como os demais bancos de dados NoSQL, os bancos de dados em colunas foram desenvolvidos para aumentar a escala horizontal usando clusters distribuídos de hardware de baixo custo para aumentar o throughput, o que os torna ideais para data warehousing e processamento de Big Data.

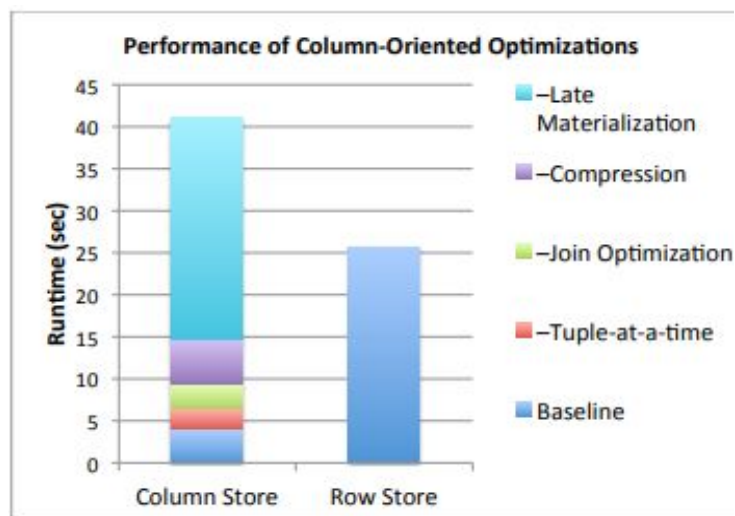


Figura 4: Exemplo do desempenho de otimizações orientadas por Colunas (Column Store) comparadas com orientadas por Linhas (Row Store).

Fonte: *The Design and Implementation of Modern Column-Oriented Database Systems* (2012)

3. Vantagens

- Como citado em objetivos, o desempenho é a principal vantagem do banco de dados, sendo superior a outros bancos quando o assunto é análise e consulta de dados.
- A capacidade de armazenamento ocorre pelo fato de ser comprimido uma coluna por vez. Isso justifica o porque a compressão chega a ser de 60% ou mais, o tornando bem mais eficiente que outros bancos, como o por linhas.
- Adequado para distribuição em diferentes armazenamentos ou seja, o banco é altamente escalável.
- Vários SGBDs colunares que aplicam métodos de compressão apresentam bons resultados em redução de espaço em disco, o que ocasionou em ser um dos quesitos principais na escolha de um SGBD. principalmente para grandes empresas. O motivo é que são necessários menos ciclos da CPU para a realização da operação.
- Os SGBDs de modelo colunar podem comprimir dados com uma proporção maior do que muitos SGBDs de modelos relacionais, o tornando uma importante vantagem.

4. Desvantagens

- As operações de alterações e exclusões diminuem a eficiência do banco. Isso ocorre porque quando se há necessidade de alteração de informações e acabam tendo que modificar diversas colunas, e por estarem em colunas individuais serão realizados descompressão de diversos arquivos diferentes.
- SGBDs Verticais têm dificuldades para realização de transações. Caso tenha que trabalha com transações, o recomendado é utilizar outro tipo de tecnologia de banco de dados.
- As gravações de informações acabam se tornando caras. Pois cada linha é dividida em várias colunas e cada gravação acaba sendo reescrita muitas vezes.
- Consultas usando *JOINS* podem reduzir o desempenho do banco.

5. Exemplos de Uso

5.1. BigTable

BigTable é um SGBD orientado a colunas de armazenamento de dados e de alto desempenho criado pela Google em 2004.

Ele funciona como um mapa multidimensional distribuído com três dimensões (chave de linha, chave de coluna e timestamp, que mostra quando um evento ocorreu).



Figura 5: Logo da BigTable

Fonte: Google Cloud

Frequentemente o BigTable é usado para gerar e modificar dados armazenados por várias aplicações como Google Book Search, Google Code, Google Maps, Gmail e YouTube. As razões do Google para desenvolver o BigTable incluem a preocupação da escalabilidade e melhor controle das características de desempenho.

Em 2015, foi disponibilizado uma versão gratuita e pública foi disponibilizada.

5.2. HBase

O Apache HBase é um banco de dados distribuído orientado a colunas que foi modelado a partir do BigTable.

Sua linguagem de pesquisa é em Java e é conhecido por armazenar grande quantidade de dados, além de ser *open source*.

O objetivo do Hbase é de processar facilmente vários terabytes de dados.

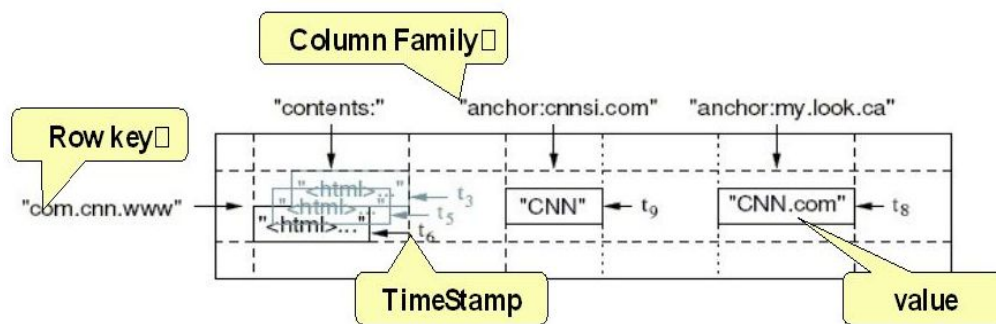


Figura 6: Modelo de Dados do BigTable, onde o Modelo do HBase é similar com o da imagem acima.

Fonte: SlidePlayer - *Data Model and Storage in NoSQL Systems (Bigtable, HBase)*

5.3. Cassandra

Apache Cassandra é um banco de dados não relacional e colunar criada pelo *Facebook* em 2008, seu modelo de dados é baseado do *BigTable* do *Google* e sua arquitetura baseada em *DynamoDB* da *Amazon*.

Assim como Hbase, Cassandra é *open source* e seu principal objetivo é manipular grandes quantidades de dados em servidores em comuns. É também usado para escrita e leitura, mas apenas se o banco foi bem modelado, pois para cada consulta haverá uma tabela.

The Evolution of Cassandra

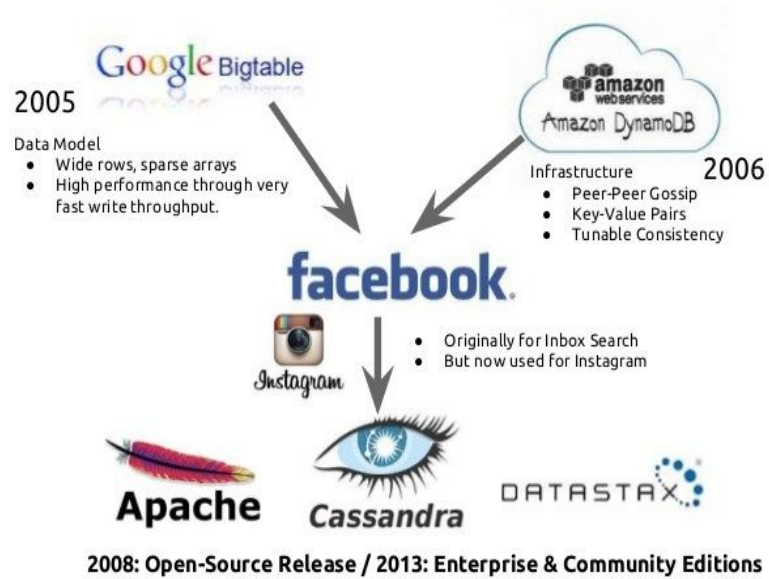


Figura 7: Evolução da Cassandra

Fonte: QuindaNandas - *Apache Cassandra - Cassandra Tutorial*

Referências

1. ABADI, D. et al. *The Design and Implementation of Modern Column-Oriented Database Systems*. Editora Now. 2012.
2. AMAZON WEB SERVICES. **Armazenamento colunar**. 2019. Disponível em:
<https://docs.aws.amazon.com/pt_br/redshift/latest/dg/c_columnar_storage_disk_mem_mgmt.html>. Acesso em: 08 set. 2019.
3. BARROSO, Isaiás. *Banco de Dados Orientado a Colunas*. WordPress. 2012. Disponível em:
<<https://isaiasbarroso.wordpress.com/2012/06/20/banco-de-dados-orientado-a-colunas/>>. Acesso em: 08 set. 2019.
4. CHANG, F., et. al. *Bigtable: A Distributed Storage System for Structured Data*. Google Inc.
5. 10. *CLOUD BIGTABLE*. *GoogleCloud*. Disponível em:
<<https://cloud.google.com/bigtable/>>. Acesso em: 08 set. 2019
6. *COLUMN-ORIENTED DATABASE TECHNOLOGIES*. *DbBest*. 2012
Disponível em:
<<https://www.dbbest.com/blog/column-oriented-database-technologies/>>. Acesso em 08 set. 2019.
7. ELTABAKH, Mohamed. *Data Model and Storage in NoSQL Systems (Bigtable, HBase)*. 2015. 27 slides. Disponível em:
<<https://slideplayer.com/slide/10014970/>>. Acesso em: 08 set. 2019
8. GOMES, André. Banco de dados noSQL Família de Colunas. 2017. (17m57s). Disponível em:
<https://www.youtube.com/watch?v=5WYw_DJpm3s>. Acesso em 08 set. 2019.
9. JOSKO, João. *SGBD RELACIONAIS ORIENTADOS A COLUNA: UMA NOVA ROUPAGEM AO DATA WAREHOUSING*. *DevMedia*. 2008.
Disponível

em:<<https://www.devmedia.com.br/sqbd-relacionais-orientados-a-coluna-uma-nova-roupagem-ao-data-warehousing-parte-01/11349>> Acesso em 08 set. 2019.

10. MERIAT, Vítor. *Column Store x Row Store e o In-Memory*. VitorMeriat. 2015. Disponível em:
<<https://www.vitormeriat.com.br/2015/07/17/column-store-x-row-store-e-o-in-memory/>>. Acesso em 08 set. 2019.
11. NANDA, Quindas. *Apache Cassandra - Cassandra Tutorial*. Blogspot. 2017. Disponível em:
<<http://quindasnanda.blogspot.com/2017/01/apache-cassandra-cassandra-tutorial.html>>. Acesso em 08 set. 2019.
12. O QUE É UM BANCO DE DADOS VERTICAL. *PtComputer*. Disponível em:
<<http://ptcomputador.com/Software/database-software/113045.html>>. Acesso em: 08 set. 2019.
13. SABINO, Alexandre. *Apache Cassandra: confira absolutamente tudo a respeito*. Medium. 2018. Disponível em:
<<https://medium.com/nstech/apache-cassandra-8250e9f30942>>. Acesso em 08 set. 2019
14. SOARES, B. BOSCARIOLI C. *Modelo de Banco de Dados Colunar: Características, Aplicações e Exemplos de Sistemas*. 2012. 10f. Artigo Científico - Universidade Estadual do Oeste do Paraná (UNIOESTE), Cascavel, 2012.