

UNIVERSIDADE DE BRASÍLIA

Faculdade do Gama

Sistemas de Banco de Dados 2

Tecnologias de Banco de Dados (TI-BD)

Banco de Dados NoSQL

Welison Lucas Almeida Regis, 17/0024121

Brasília, DF

2019

1. Banco de Dados NoSQL

O termo NoSQL, majoritariamente interpretado como “não apenas SQL”, representa os bancos de dados não relacionais (VAISH, 2013, p. 19). A nova tecnologia adveio do desejo de projetistas de grandes organizações em ter bancos de dados que não precisassem de estruturas e regras presentes no modelo tradicional e também armazenassem uma grande quantidade de dados, isto é, um sistema mais simples, porém robusto (TOMIO, 2015, p. 21). Desta forma, surgiram os primeiros Sistemas Gerenciadores de Banco de Dados (SGBDs) que cumpriam tais necessidades, como o BigTable da Google (LEITE; BONOMO, 2016, p. 7).

1.1. ACID x BASE

Diante da necessidade de escalabilidade, os projetistas depararam-se com empecilhos em relação as quatro propriedades dos bancos relacionais, cujo acrônimo é ACID: atomicidade, consistência, isolamento e durabilidade. Assim, para ganhar escalabilidade, surgiu o BASE (Basically Available, Soft State, Eventually Consistent). Essas novas propriedades priorizam a disponibilidade do serviço e, decorrente disso, garante a consistência do banco somente quando necessário, eventualmente. (LEITE; BONOMO, 2016, p. 7-8).

1.2. Teorema CAP

Acompanhado de mudanças na maneira como os dados até então eram guardados, estudos realizados pelo professor Eric Brewer da Universidade Da Califórnia Berkeley, resultaram no chamado “Teorema CAP”. Conforme o teorema, apenas dois dos seguintes três aspectos podem ser totalmente encontrados em um banco de dados ao mesmo tempo (LEITE; BONOMO, 2016, p. 8):

- **Consistência** (Consistency): Todos os nós veem a mesma versão da informação em dado tempo.
- **Disponibilidade** (Availability): todos os clientes podem ter acesso aos

dados quando quiserem.

- **Tolerante a Partição** (Partition Tolerant): o sistema continua funcionando mesmo se algumas partições perderem a comunicação.

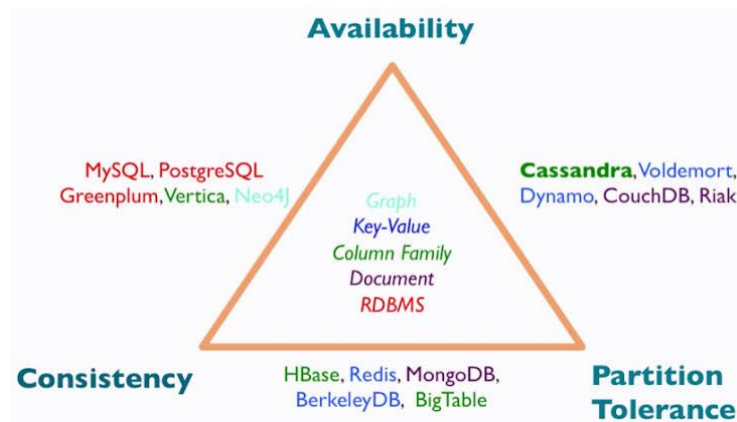


Figura 1 - Teorema CAP. Fonte: Google Images (2019)

1.3. Modelos de Dados NoSQL

Há muitos bancos de dados NoSQL no mercado, porém, podem ser agrupados por semelhanças nas estruturas internas ou na forma como tratam os dados. São 4 as classificações do NoSQL: Chave-Valor, baseado em documento, baseado em coluna e baseado em grafo (ANICETO; XAVIER, 2014, p. 17-18).

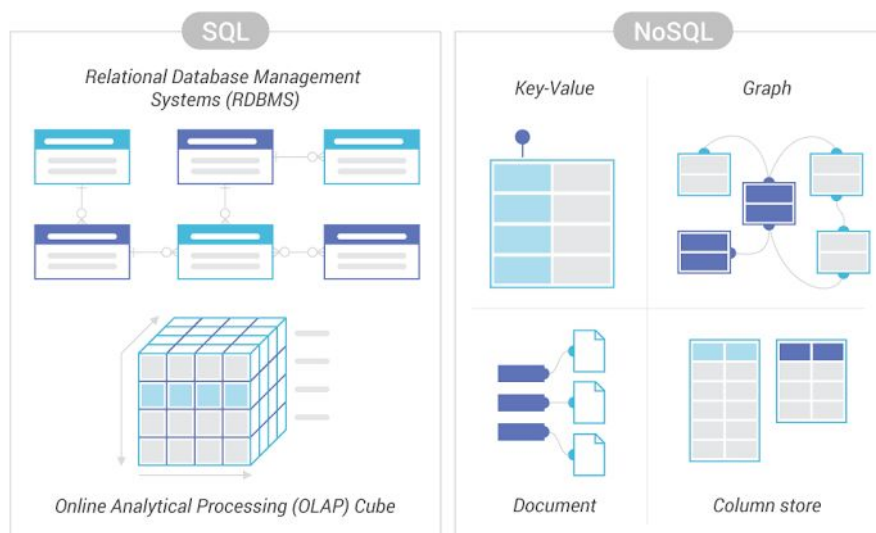


Figura 2 - Modelos NoSQL e SQL. Fonte: Google Images (2019)

1.3.1. Banco de Dados Chave-Valor

O modelo chave-valor foi o precursor do movimento NoSQL. O mapeamento de dados é realizado conforme um dicionário, isto é, com a chave há como endereçar os valores desejados. O banco de dados chave-valor é livre de esquema, sendo os valores isolados e independentes, e o relacionamento é tratado pela lógica da aplicação. O modelo chave-valor é uma solução apropriada para aplicações que se baseiam em um único tipo de objeto. Exemplos de bancos chave-valor: Berkeley DB, Dynamo, Redis. (LEITE; BONOMO, 2016, p. 8)

1.3.2. Banco de Dados Baseado em Grafo

Nesse modelo, utiliza-se a Teoria dos Grafos para conceber um modelo de banco de dados através de grafos dirigidos. Tem-se como componentes básicos os nodos, relacionamentos e propriedades, representados pelos vértices do grafo, arestas e atributos, respectivamente. As principais aplicações que se beneficiam dessa tecnologia são as redes sociais, redes de informação, redes tecnológicas e redes biológicas. Exemplos de bancos baseados em grafos: Neo4j, GraphDB e InfoGrid. (LEITE; BONOMO, 2016, p. 11)

1.3.3. Banco de Dados Baseado em Documentos

Nesse tipo de banco, utiliza-se a ideia de coleções de documentos, geralmente salvos em JSON (Javascript Object Notation) ou XML (Extensible Markup Language), que armazenam atributos e valores sem um esquema definido. Bancos baseados em documento geralmente suportam índices secundários e diversos tipos de documentos (objetos), além de documentos aninhados ou listas. Possui fácil manutenção e consultas dinâmicas. Exemplos de bancos baseados em documento: MongoDB, CouchDB e Riak. (LEITE; BONOMO, 2016, p. 10)

1.3.4. Banco de Dados Baseado em Colunas

Os banco de dados baseados em coluna estabelecem a estrutura dos valores como um conjunto pré-definido de colunas. Tem-se como principais definições: as colunas, super colunas e família de colunas (LEITE; BONOMO, 2016, p. 9-10):

- Coluna: valor atômico. Representado por nome e valor.
- Super colunas: associam colunas. Associação semântica.
- Família de colunas: abrange colunas e super colunas.

Esse modelo é adequado para grandes volumes de dados, de modo que o modelo de dados pode ser eficientemente particionado. Pode-se citar como exemplos de bancos baseados em coluna: BigTable, PNUTS e Cassandra. (LEITE et. al., 2016, p. 9-10)

2. Principais Objetivos do NoSQL

Conforme pode-se notar, há diversos modelos de banco de dados NoSQL, cada um com suas peculiaridades para resolver diferentes problemas. Porém, em sua maioria, possuem características em comum, que acabam por se tornar características dos bancos de dados NoSQL, são elas: (LEITE; BONOMO, 2016, p. 8-9)

- Alta escalabilidade, performance e disponibilidade;
- Processamento distribuído;
- Flexibilidade de esquema;
- Suporte a replicação de dados;
- Armazenamento de dados estruturados e não-estruturados;
- API simples para acesso aos dados;
- Maior flexibilidade às propriedades ACID;
- Modelo de dados mais recente.

De acordo com a tabela a seguir (OLIVEIRA, 2019), pode-se notar os objetivos para cada tipo de modelo disponível em NoSQL:

| | Performance | Escalabilidade | Flexibilidade | Complexidade |
|--------------------|-------------|-----------------|---------------|--------------|
| Chave-Valor | Alta | Alta | Alta | Nenhuma |
| Colunas | Alta | Alta | Moderada | Baixa |
| Documentos | Alta | Variável (alta) | Alta | Baixa |
| Grafos | Variável | Variável | Alta | Alta |

Tabela 1 - Comparação entre modelos de dados. Fonte: do autor (2019).

3. Vantagens da Tecnologia NoSQL

A tecnologia NoSQL, desenvolvida decorrente de necessidades no mercado, foi um marco em relação aos modelos de dados anteriores. Comparando-se essa tecnologia com o modelo relacional de dados, pode-se analisar o seguinte:

- **Escalabilidade elástica:** diferente dos bancos relacionais, que são escaláveis verticalmente (investimento em upgrades de hardware, software), os bancos NoSQL são escaláveis horizontalmente, isto é, permite maior elasticidade de escala, caso necessário. Há a possibilidade de escalar horizontalmente um banco relacional, porém o custo é muito alto. (VAISH, 2013, p. 31-35)
- **Esquemas dinâmicos:** os bancos de dados NoSQL não precisam de esquemas para começar a trabalhar com dados, já nos bancos de dados relacionais, é necessário definir um esquema primeiro. A versatilidade nos esquemas permitem aumentar performance de manipulação de informações e trabalhar com dados aglomerados. (SADALAGE; FOWLER, 2013, p. 23-25)
- **Open Source:** a maior parte dos bancos de dados NoSQL são de código aberto, isto é, não é um software proprietário, como é corriqueiro nos bancos de dados relacionais. (OLIVEIRA, 2019)
- **Aplicações Big Data:** os bancos NoSQL possuem muita performance ao lidar com aglomerados de dados em escalas altíssimas, o que sobressai a performance dos bancos relacionais. Tem-se buscas mais

dinâmicas devido, por exemplo, a ausência de queries complicadas que englobam joins e outros recursos dos bancos relacionais. (VAISH, 2013, p. 31-35)

4. Desvantagens da Tecnologia NoSQL

Embora seja um modelo de dados com muitas vantagens, o NoSQL possui um público-alvo e uma finalidade às vezes distintas da tecnologia de banco de dados relacional, como é possível notar pelos seguintes pontos:

- **Confiabilidade:** as aplicações NoSQL, em geral, desagregam-se das propriedades ACID dos bancos relacionais a fim de ganhar mais performance, escalabilidade etc. Nesse sentido, acarreta-se em desvantagens em relação a confiabilidade do serviço, fora que os desenvolvedores deverão prover seus próprios meios para garantir o devido funcionamento coeso do sistema, o que aumenta a complexidade do código. (VAISH, 2013, p. 21-22)
- **NoSQL não compatível com SQL:** significa que há uma curva de aprendizagem grande no aprendizado de tecnologias NoSQL, pois há muitas disparidades de um modelo para outro. (OLIVEIRA, 2019)
- **Modelo de dados recente:** devido a ser uma tecnologia recente, depara-se com um cenário de maior instabilidade, atualizações constantes, suporte ainda prematuro, falta de documentação ou uma comunidade muito nova. (OLIVEIRA, 2019)

| | NoSQL Databases | Bancos Relacionais |
|------------------------|--------------------------|--------------------|
| Performance | Alto | Baixo |
| Confiabilidade | Muito baixo | Bom |
| Disponibilidade | Bom | Bom |
| Consistência | Muito baixo | Bom |
| Armazenamento de dados | Otimizado para big datas | Razoável |
| Escalabilidade | High | Alto (custoso) |

Tabela 2 - Diferença Bancos Relacionais e Bancos Não Relacionais. Fonte: OLIVEIRA (2019).

5. NoSQL no Mercado

5.1. Neo4j

O Neo4j é um software livre, escrito em Java, tendo como diferencial os algoritmos de grafos, como Dijkstra, Shortest Path, A* etc., para manipulação da base de dados, além de conseguir garantir as propriedades ACID. Estrutura-se através de nós do grafo, os quais contêm dados, e realiza o relacionamento através das arestas do grafo. (TOMIO, 2015, p. 23-24)

Pelo “Teorema CAP”, o Neo4j encaixa-se no segmento consistência e disponibilidade, isto é, há uma grande dificuldade em ser tolerável a partição devido ao formato de grafo. Utiliza-se de técnicas de replicação e failover (substituição de servidor em caso de falha) para alcançar alta disponibilidade.

5.2. MongoDB

O MongoDB é um software livre, escrito na linguagem C++, tendo como diferencial a sua linguagem de consulta, por ser baseada em documentos. Estrutura-se hierarquicamente, inicialmente com as coleções que agrupam documentos, esses no formato JSON, com um agregado de campos com um par chave-valor. (TOMIO, 2015, p. 32-34)

Pelo “Teorema CAP”, o MongoDB busca a consistência e a tolerância à partição. Logo, percebe-se também que a consistência é fator de atenção no MongoDB, que utiliza também de replicação e failover para prover escalabilidade.

5.3. MongoDB x Neo4j

Comparação entre características do MongoDB e do Neo4j:

| | MongoDB | Neo4j |
|----------------------------------|--|-------------------------------------|
| Descrição | Baseado em documentos | Baseado em grafos |
| Licença | Open Source | Open Source |
| Esquema | Livre de esquema | Livre de esquema e esquema opcional |
| Tipificado | Sim | Sim |
| Método de particionamento | Sharding | Não possui |
| ACID | Sim, porém com consistência eventual/imediata. | Sim |

Tabela 3 - Comparação entre tecnologias MongoDB e Neo4j. Fonte: do autor (2019).

5.4. Caso de Sucesso - MongoDB e Adobe

Na sua página oficial do MongoDB, há uma área reservada aos relatos de sucesso dos clientes. Nesse espaço, há diversas depoimentos de empresas renomadas no mercado de tecnologia, como pode-se citar: adobe, sega, facebook, google.

Como um caso de sucesso, pode-se falar no uso do MongoDB na aplicação Experience Manager 6.0 da Adobe. Devido a uma necessidade de escalabilidade, a Adobe adotou o MongoDB de forma a armazenar petabytes de dados para gerenciamento e entrega de experiências digitais em suas aplicações. Com o suporte ao MongoDB, a Adobe adicionou outra opção que oferece uma escalabilidade horizontal para gravações e leituras no serviço. O MongoDB auxilia também na hospedagem das próprias instâncias dos clientes do Experience Manager. (MONGODB, 2014)

6. REFERÊNCIAS

- [1] VAISH, G. **Getting Started with NoSQL: 1ª Edição**. Birmingham, Reino Unido, Packt Publishing. 2013.
- [2] SADALAGE, P. J.; FOWLER, M. J. **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence**. Addison-Wesley, 1ª edição. 2013.
- [3] ANICETO, Rodrigo Cardoso; XAVIER, Renê Freitas. **Um Estudo Sobre a Utilização de Banco de Dados NoSQL Cassandra em Dados Biológicos**. 2014. 50 f. TCC (Graduação) - Curso de Ciência da Computação, Ciência da Computação, Universidade de Brasília, Brasília, 2014. Disponível em: <http://bdm.unb.br/bitstream/10483/7927/1/2014_RodrigoCardosoAniceto_Ren_eFreireXavier.pdf>. Acesso em: 07 set. 2019.
- [4] TOMIO, Gustavo Veloso. **Utilizando a tecnologia de banco de dados NoSQL: um Caso Prático**. 2015. 119 f. TCC (Graduação) - Curso de Ciência da Computação, Informática, UTFPR, Paraná, 2015. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/7402/1/PG_COCIC_2015_1_03.pdf>. Acesso em: 08 set. 2019.
- [5] LEITE, Hermano Portella; BONOMO, Igor da Silva. **Análise Comparativa de Projeto e Administração de Banco de Dados Entre os SGBDs Cassandra e MySQL**. 2016. 61 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade de Brasília, Brasília, 2016. Disponível em: <http://bdm.unb.br/bitstream/10483/15681/1/2016_HermanoPortellaLeite_IgorDaSilvaBonomo_tcc.pdf>. Acesso em: 08 set. 2019.
- [6] COMPANY, MongoDB. **MongoDB Delivers Multi-Petabyte Data Store Option for Adobe Experience Manager**. 5 jun. 2014. Disponível em: <<https://www.mongodb.com/press/mongodb-delivers-multi-petabyte-data-store-option-adobe-experience-manager?c=69bb399984>>. Acesso em: 8 set. 2019.
- [7] OLIVEIRA, Lucas. **Everything you need to know about NoSQL databases**. 5 jun. 2019. Disponível em: <<https://dev.to/lmolivera/everything-you-need-to-know-about-nosql-databases-3o3h#comparison>>. Acesso em: 6 set. 2019.