UniCeub

Ciência da Computação

Camila Lauriano - 22510864

Mateus Omaki - 22454201

Banco de Dados NoSQL

Brasília – DF

2025

1. INTRODUÇÃO

Ao decorrer da história da ciência da computação, houve múltiplas mudanças em diversos componentes desse campo - desde linguagens diferentes para se programar até novos processos e plataformas. Nesse contexto, algo que se mantém constante é a evolução dos modelos de bancos de dados visando sua optimização. Assim, durante cerca de duas décadas, o modelo vigente e mais utilizado foi o modelo de banco de dados relacional.

Contudo, com a popularização da internet e a maior acessibilidade da tecnologia, o mercado da Ciência da Computação se ampliou e, assim novos problemas em relação ao armazenamento de dados veio a superfície. Com isso iniciou-se o movimento “NoSQL”, que, no começo, interpretava-se como “Não SQL”, ou seja, um movimento contrário ao uso de SGBDs relacionais. Porém, ao passar do tempo a sigla passou a ser interpretada como “Não Apenas SQL”, ressaltando a necessidade de sempre pesquisar e escolher o melhor modelo para cada caso, mesmo que esse seja um modelo relacional.

No livro “NoSQL: como armazenar os dados de uma aplicação moderna”, David Paniz, exemplifica a necessidade do uso de um modelo NoSQL para o registro de discos de uma banda, em que cada disco teria diferentes atributos associados e que, caso armazenados em um modelo SQL, haveria tabelas com dezenas de colunas e diversos campos nulos. Dito isso, é importante ressaltar que, no geral, a aplicação do NoSQL não se trata de resolver problemas de resolução impossível no modelo relacional, mas sim sobre tornar tais soluções mais práticas e eficientes.

Em síntese, em um contexto em que são necessárias soluções mais maleáveis para abranger uma gama maior de aplicações de um banco de dados, o modelo NoSQL se tornou um aliado aos cientistas de dado quando se trata da optimização do armazenamento de dados por oferecer a adaptabilidade que lhes é necessária.

Nesse sentido, a seguir, este ensaio discorrerá brevemente sobre os fundamentos desse modelo, sua relação com OLTP e OLAP, e uma comparação técnica entre dois SGBDs.

2. FUNDAMENTOS DE NoSQL

2.1. CONCEITO

Bancos de dados NoSQL, não sistemas não relacionais com modelos de armazenamento de dados mais flexíveis, diferentemente do modelo tradicional de

tabelas e tuplas. Nesse modelo, é possível acomodar dados não estruturados ou semiestruturados, o que permite maior adaptabilidade a mudanças nos dados, além de bancos de dados NoSQL serem optimizados para trabalhar em alta velocidade.

2.2. HISTÓRICO E MOTIVAÇÃO

Antes da grande popularização da internet nos anos 2000, os bancos de dados empresariais costumavam lidar com quantias de dados consideravelmente menores e menos detalhadas, porém, após tal popularização, as grandes empresas começaram a lidar com uma gama de usuários muito maior e os sites começaram a monitorar as atividades desses usuários de forma muito mais detalhada e, a cada momento, com maior velocidade – esse armazenamento de dados altamente variados, em grande volume e velocidade é o que conhecemos hoje como Big Data.

Nesse novo cenário, notou-se que o manejo de dados pelo modelo relacional SQL tornou-se inviável, visto que a maioria das databases SQL não eram pensadas para rodar em clusters – conjunto de maquinas mais baratas que funcionam em sincronia com um único sistema –, que começou a ser adotado por ser mais barato do que investir em um único servidor de alta performance e por ser mais seguro, visto que os danos pela interrupção do funcionamento de uma das máquinas em um cluster são consideravelmente menores do que os causados pela interrupção do funcionamento em um servidor inteiro, além do fato de a maioria dos SGBDs relacionais cobrarem a licença de uso por unidade de servidor.

Assim, tornou se evidente que outra opção era necessária para melhor se adaptar a nova realidade. E então, iniciou-se a idealizar o modelo NoSQL, um modelo de banco de dados orientado a rodar em clusters e mais eficiente no manejo de dados com relações mais complexas. Desse modo, sendo um modelo com melhor escalabilidade e maior eficiência no tratamento de dados em ampla quantidade e variação.

2.3. CATEGORIAS DE NoSQL

Existem 4 categorias de bancos de dados não relacionais, entre elas estão:

- **Chave-valor:** os dados são armazenados em pares, uma chave e um valor, cada um sendo um registro único, a chave identifica o registro e o valor pode ser uma estrutura de dados simples ou complexa.

- **Colunar:** todos os registros são parte de uma mesma tabela mas cada um pode ter diferentes colunas, os dados são organizados verticalmente e de forma contígua em cada linha.
- **Documento:** cada registro tem uma coleção específica, porém a coleção armazena os dados em formato não estruturado ou semiestruturado, como JSON ou XML.
- **Grafo:** cada registro é um nó em um grafo, e são interligados por relacionamentos.

2.4. VANTAGENS E DESVANTAGENS

Existem múltiplas vantagens no uso de bancos de dados não relacionais, dentre elas a escalabilidade horizontal – é um modelo criado e pensado para funcionar em clusters, ou seja, sempre que necessário há a possibilidade de adicionar uma nova máquina ao sistema –, flexibilidade – permite o armazenamento de dados com estruturas variadas –, alta disponibilidade – ao utilizar replicação de dados, garante que o sistema continue disponível mesmo após a falha de um servidor –, alto desempenho – são otimizados para processar grandes volumes de dados em alta velocidade – e custos reduzidos – em sua maioria são modelos open-source, o que elimina o custo com licenciamento.

Contudo, essa tecnologia, assim como todas as outras, tem suas desvantagens, dentre elas estão a eventual inconsistência – a maioria sacrifica o ACID (atomicidade, consistência, isolamento e durabilidade) pelo BASE (basically available, soft state e eventually consistente, ou seja, disponibilidade, maleabilidade e se torna consistente ao longo do tempo) –, consultas mais complexas – podem ser mais difíceis de executar e um pouco mais lentas – e falta de padrão universal – não há um padrão definido para esse formato de banco de dados, nem para sua linguagem de query.

3. OLTP e OLAP

3.1. DEFINIÇÃO

OLTP (Online Transaction Processing) designa os sistemas operacionais com dados de transação. Fornecem essa assistência não somente em uma visão operacional como também processa dados de vários sistemas e suas variadas informações.

OLAP (Online Analytical Processing) é executado dentro de uma Data Warehouse para análise de volumes massivos de dados, como também auxilia na visualização das informações gerenciais.

3.2. DIFERENÇAS

Os sistemas OLAP usam modelos de dados multidimensionais, armazenando em “cubos”, sua arquitetura foca na leitura de dados e realiza consultas complexas eficientemente. Exigem alta performance e alto armazenamento (de TB até PB), sua performance processa dados em grandes lotes periodicamente.

Os sistemas OLTP são unidimensionais, organizados em tabela com entidades e seus atributos. Sua arquitetura foca nas operações de gravação de dados, os requisitos de um sistema OLTP exigem menos armazenamento (GB) e mais performance de processamento de fluxo, o qual o tempo de processamento é em milissegundos ou menos, tendo as atualizações do banco de dados em tempo real.

3.3. NoSQL E O ENCAIXE AOS MODELOS

Os bancos NoSQL tem sua abordagem mais voltada para os grandes volumes de dados, com escalabilidade horizontal e flexibilidade no modelo de dados.

O OLTP lida com quantidades enormes de transições em tempo real, os bancos de dados que são adaptados a essa carga de trabalho seguem com as abordagens de ACID e Schema definido. O ACID (Atomicidade, Consistência, Isolamento e Durabilidade) é uma propriedade que garante a integridade das transações e o Schema definido é a tabela em que os dados serão armazenados, facilitando leitura, escrita e consistência.

A escalabilidade horizonta dos bancos NoSQL são uma vantagem nos cenários de altas cargas de transições, por conta da abordagem de consistência eventual os dados não são atualizados em fluxo, prejudicando a consistência, e, sua modelagem flexível e sem esquema fixo não garantem a integridade de transações complexas.

NoSQL não é ideal para os cenários comumente enfrentados em OLTP por sua falta de consistência forte e integridade de dados.

O OLAP lida com consultas complexas de enormes volumes de dados, vistos em data warehouses, onde há enormes quantidades de dados históricos para análise, com o sistema visando consultas complexas e agregações. As consultas e agregações são usadas em larga escala para análises complexas contendo somas, médias e percentis.

Os bancos NoSQL conseguem armazenar altos volumes de dados, mas não lidam com as consultas analíticas complexas, não tem a mesma capacidade de agregação que os bancos OLAP.

Com isso, os bancos NoSQL não são ideais para os cenários comuns de bancos de dados OLAP, eles não cumprem as análises complexas com agregações.

3.4. APLICAÇÃO OLTP/OLAP COM SQL E NoSQL

3.4.1. OLTP:

SQL: Para bancos de varejo há um sistema de Internet Banking para registrar depósitos, saques e transferências em tempo real. Já em hospitais costuma-se ter bancos para prontuários eletrônicos, consultas agendadas e prescrições.

NoSQL: Os aplicativos de delivery armazenam pedidos com atualização rápida de status. Em jogos online é aplicado guardando progresso dos jogadores, inventário e pontos.

3.4.2. OLAP:

SQL: Data warehouse corporativo encontra-se a integração de vendas, marketing e finanças para relatórios. Na saúde pública os bancos de dados servem como base para dashboards de epidemiologia (casos por região, tendência temporal).

NoSQL: Em análise de logs os bancos processam bilhões de eventos para detectar falhas ou padrões de uso em tempo real. Em Analytics em e-commerce é utilizado na análise de comportamento de compra e recomendação de produtos.

4. COMPARAÇÃO TÉCNICA: MySQL e MongoDB

4.1. MySQL

Um SGDB open-code feito pela Oracle, o MySQL providencia o armazenamento de dados relacional, o formato sendo de linhas em colunas, operando no modelo cliente-servidor, onde o usuário usa comandos interagindo com o servidor e modificando os dados. É capaz de replicar e agrupar via sincronização para eficiência de execução, SSL, mascaramento de dados, possibilidade de back-up de múltiplas maneiras etc.

4.2. MongoDB

Um banco de dados NoSQL open-code que providencia grande armazenamento de dados não estruturados, ele troca as linhas e colunas por documentos e coleções. Uma coleção pode conter vários documentos que possuem pares de chave-valor (a unidade básica de dados. Capaz da replicação do mesmo dado, distribuindo-o em

vários servidores, suporte a consulta ad hoc, fragmentação dos dados, balanceamento de carga e escala horizontal.

4.3. MySQL vs MongoDB

O MySQL é utilizado em cenários de exigência de alta segurança e integridade referencial ao dado, definição do esquema com dados organizados, que não necessariamente precisa ser atualizado constantemente, ele lida muito bem em tráfego de baixo volume.

O MongoDB é visto cenários em que há várias fontes de dados e diferentes formatos para integrar, o back-end sendo melhor para alto desempenho, alta escalabilidade e alto tráfego por causa de sua escalabilidade horizontal, lida muito bem com estruturas complexas que se encontram em evolução constante.

5. CONCLUSÃO

Em suma, o modelo NoSQL surgiu após a necessidade de lidar com volumes de dados maiores e com tipos de dados mais variáveis, visto que era necessário um modelo de base de dados mais maleável, escalável e com menor custo monetário. O modelo NoSQL cumpre os requisitos da nova realidade e pode ser utilizado desde maneiras mais simples até maneiras mais complexas. Assim, notou-se que bancos de dados relacionais nem sempre seriam a opção mais viável para todos os problemas.

Na atualidade, há um consenso de que a comparação entre SQL e NoSQL não se trata de superioridade de um sobre o outro, mas sim da adaptabilidade e compatibilidade de cada um em relação a cada problema em específico, sendo os problemas que exigem mais maleabilidade e escalabilidade, normalmente, resolvidos com mais eficiência com modelos NoSQL.

Desse modo, é notório que, nas empresas modernas, esse novo modelo tornou-se um aliado, ao gerar maior eficiência no manejo de vastos volumes de informações, sendo assim uma opção mais viável e econômica em casos como os de empresas que trabalham com big data, redes sociais, análises em tempo real, aplicações móveis, e-comerce, entre outras.

Em síntese, sua maleabilidade, escalabilidade e eficiência a torna a melhor opção para lidar com amplos volumes de dados não estruturados ou semiestruturados com boa performance e volcidade.

FONTES

NoSQL Distilled – Martin Fowler e Pramod J. Sadalage

NoSQL: como armazenar os dados de uma aplicação moderna – David Paniz

https://nelsonfrugeri-tech.medium.com/entenda-de-uma-vez-por-todas-os-conceitos-b%C3%A1sicos-dos-banco-de-dados-chave-valor-5ba44c1a9ab8

https://www.purestorage.com/br/knowledge/what-is-a-columnar-database.html#:~:text=Voc%C3%AA%20pode%20estar%20familiarizado%20com,coluna%20e%20n%C3%A3o%20por%20linhas.

https://aws.amazon.com/pt/nosql/#:~:text=de%20dados%20NoSQL-,O%20que%20s%C3%A3o%20bancos%20de%20dados%20NoSQL?,a%20come%C3%A7ar%20a%20us%C3%A1%2Dlo.

https://www.homehost.com.br/blog/banco-de-dados/nosql/#:~:text=Desafios%20e%20Limita%C3%A7%C3%B5es%20do%20NoSQL,de%20dados%2C%20como%20sistemas%20financeiros.

https://aws.amazon.com/pt/compare/the-difference-between-olap-and-oltp/

https://www.dataex.com.br/oltp-e-olap/

https://www.devmedia.com.br/banco-de-dados-nosql-um-novo-paradigma-revista-sql-magazine-102/25918#3

https://medium.com/@sounder.rahul/oltp-olap-systems-for-beginners-data-engineers-b3d6af741472

https://www.singlestore.com/blog/pre-modern-databases-oltp-olap-and-nosql/

https://aws.amazon.com/pt/compare/the-difference-between-mongodb-vs-mysql/

https://www.astera.com/pt/type/blog/mongodb-vs-mysql/