

## NewSQL: Teste de Performance

LUCAS ALVIENTE PEREIRA; Emerson de Vasconcelos Vieira; Matheus da Rosa Moeller Chaves; Ana Marilza Pernas Fleischmann

*Universidade Federal de Pelotas – {lapereira, edvvieira, mdrmchaves, marilza}@inf.ufpel.edu.br*

### 1. INTRODUÇÃO

Bancos de dados são usados para armazenar quantidades variáveis de dados, sendo aplicados para diferentes fins (sistemas para Internet, sistemas de informação gerenciais, pequenas aplicações, entre outras). Os Sistemas de Banco de Dados (SBDs) mais utilizados na atualidade são os denominados Relacionais (SBDRs), os quais utilizam uma linguagem de consulta padronizada, denominada SQL (Structured Query Language) (MANEGOLD et al, 2002).

De acordo com (MELTON, SIMON, 1993), SQL é uma linguagem para criação e manipulação de banco de dados relacionais gerenciados por SGBDR. Os Sistemas de Gerenciamento de Banco de Dados Relacionais (SGBDR) são sistemas que gerenciam esses bancos, compostos por tabelas e colunas, podendo haver relações entre elas.

A criação, consulta, inserção e outras modificações dentro dessas tabelas são feitas por meio da linguagem SQL, tornando-se um recurso poderoso juntamente com o suporte a transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade), que garantem que as operações sejam executadas de forma consistente e confiável.

Porém, a modernização das aplicações e o crescimento da internet e do fluxo de dados trafegados em aplicações móveis exige a proposta de novos paradigmas e tecnologias para os bancos de dados. Estes paradigmas, comumente chamados de NoSQL, não suportam as propriedades ACID e não utilizam o SQL, podem ser divididos em: chave-valor, orientados a documentos, orientados a grafos e distribuídos.

Tantas aplicações modernas, como em Internet das Coisas (do inglês, Internet of Things - IOT) e redes sociais, quanto as tradicionais transações bancárias utilizam uma grande quantidade de dados e enfrentam o desafio de lidar com volumes significativos de dados. Nestes casos, um dos principais desafios é como lidar com a necessidade de armazenamento de grandes volumes de dados.

Nesse contexto, uma solução para o armazenamento eficiente desses dados é a distribuição entre servidores. Esta abordagem, conhecida como Escalabilidade Horizontal, envolve a alocação de recursos em diferentes servidores, geralmente máquinas simples, visando reduzir custos (DOS SANTOS et al., 2020). Dessa forma, o crescimento ocorre de forma horizontal, distribuindo-se pelo sistema onde não é expandido o hardware e sim o número de servidores, os servidores distribuídos podem estar num mesmo local ou em locais diferentes, por esse motivo, comumente não é utilizado as propriedades ACID, pois como os dados estão distribuídos é difícil manter as propriedades.

Existem várias soluções NoSQL voltadas ao problema de escalabilidade de dados e distribuição, dentre elas este artigo destaca o NewSQL. De acordo com (STONEBRAKER, 2010) um dos principais objetivos do NewSQL é fornecer escalabilidade horizontal para processamento de transações online em tempo real

(OLTP), mantendo a integridade e a consistência dos dados, inclusive suportando transações ACID, mesmo com uma arquitetura distribuída.

Portanto, o objetivo deste trabalho é comparar o desempenho de um banco de dados NewSQL com um banco de dados relacional por meio da realização de testes de benchmark.

## 2. METODOLOGIA

Foi realizada uma pesquisa bibliográfica sobre os temas propostos neste trabalho, como NewSQL e suas principais ferramentas. Neste contexto, foram selecionados dois bancos de dados para serem utilizados na avaliação de desempenho.

Foi selecionado o uso do VoltDB como representante de um banco de dados NewSQL em memória, notando sua capacidade de operar na memória e de aceitar as propriedades ACID tradicionalmente associadas a bancos de dados relacionais. Como alternativa, foi selecionado o PostgreSQL para representar a categoria dos bancos de dados relacionais.

Para a comparação foi utilizado o VoltDB na versão 9.2.1 em um ambiente Docker, enquanto o PostgreSQL na versão 15.2-1.pgdg110+1 diretamente na máquina. A configuração da máquina utilizada para as avaliações é um sistema Windows 11 com 16GB de RAM e um processador Intel Core i5-10300H operando a uma velocidade de 4.10GHz.

Não foi possível encontrar uma ferramenta de benchmark compatível com o PostgreSQL e o VoltDB, portanto, foi usado um exemplo disponibilizado pelo VoltDB em seu repositório GitHub. Este exemplo consiste em 3 testes desenvolvidos em Java, sendo que dois deles fazem uso de conexões diretas, enquanto o terceiro utiliza JDBC (Java Database Connectivity). Todos esses testes foram executados em um ambiente Docker em um único nó de execução.

Foi desenvolvido um programa em Java específico para o PostgreSQL, seguindo o mesmo exemplo utilizado no VoltDB. Contudo, no caso do PostgreSQL, a conexão com o banco de dados foi estabelecida exclusivamente por meio do JDBC, e a execução ocorreu diretamente na máquina, dispensando o uso do Docker.

O teste de benchmark que foi utilizado é um programa que roda uma simulação de votação por telefone, o teste foi executado durante 2 minutos e durante sua execução foi efetuado milhares de requisições para inserir um voto para um candidato específico.

Para validar um voto, é necessário:

- Verificar se o candidato existe, se não existe é contabilizado como "Negado (Candidato)".
- Verificar se o número de telefone já não atingiu o limite de votos máximo (2), se sim é contabilizado como "Negado (Máximo Votos)".
- Caso contrário, o voto é registrado no banco de dados e contabilizado como "Aceito".

## 3. RESULTADOS E DISCUSSÃO

Foram realizados 4 testes, sendo 3 testes diferentes no VoltDB e 1 no PostgreSQL.

- Teste realizado usando a biblioteca oficial do VoltDB e executado de forma assíncrona.
- Teste realizado usando a biblioteca oficial do VoltDB e executado de forma síncrona.
- Teste realizado usando a biblioteca JDBC com drive do VoltDB.
- Teste realizado usando a biblioteca JDBC com drive do PostgreSQL.

Como resultado do benchmark, retornava o número de votos válidos e com erros.

Benchmark				
Resultados	VoltDB (Async)	VoltDB (Sync)	VoltDB (JDBC)	Postgresql (JDBC)
Aceito	15.193.157	5.495.800	5.508.887	53.959
Negado (Candidato)	153.725	55.783	0	511
Negado (Máximo Votos)	67	5	0	0
Negado (Error)	0	0	0	0
Total	15.346.949	5.551.588	5.508.887	54.470
Inserções/sec	127.891	46.263	45.907	454

Figura 1 - Resultado obtidos.

Como mostrado na Figura 1, a versão utilizando o VoltDB de forma assíncrona foi a que teve melhor desempenho na computação dos votos no período de 2 minutos, conseguindo inserir mais de 120 mil votos por segundos.

Isso é possível, pois esse teste utiliza a biblioteca oficial do VoltDB, garantindo melhor conexão e uso dos procedimentos armazenados, também vale notar, que o uso assíncrono garante que o VoltDB não aguarde a conclusão de um procedimento antes de continuar.

O segundo e terceiro testes, foram 3 vezes mais lento que o primeiro, porém, obtiveram ótimos resultados comparado ao quarto teste que utilizou o banco de dados PostgreSQL.

Abaixo, na Figura 2 temos o gráfico do *Throughput* a cada 5 segundos durante o período de 2 minutos, dos 4 testes citados, como podemos notar, o primeiro, segundo e terceiro testes, mantiveram-se estáveis durante todo teste, já o quarto teste, houve uma queda significativa conforme o tempo passava.

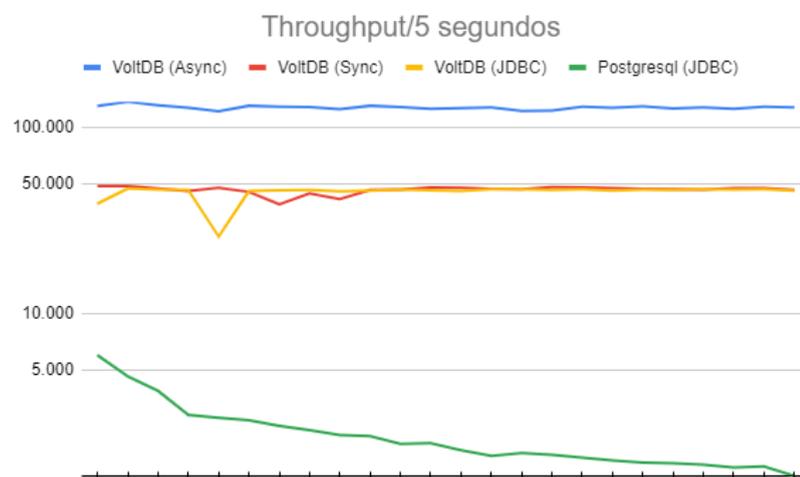


Figura 2 - *Throughput* dos testes.

## 4. CONCLUSÕES

Banco de dados relacionais são muitos úteis quando queremos a consistência dos dados, porém, não é possível escalar de maneira eficiente, para isso, utilizamos banco de dados não relacional.

Neste trabalho foi apresentado o paradigma NewSQL, que une as vantagens de um banco de dados relacional usando as propriedades ACID, juntamente com a escalabilidade horizontal, com uma arquitetura distribuída e processamento de transações online em tempo real.

Foi concluído que a tecnologia escolhida para o benchmark, VoltDB é ideal para aplicações que dependam de alto desempenho, enquanto mantém as propriedades ACID, garantindo confiabilidade, alta disponibilidade e escalabilidade, já que o VoltDB usa o máximo de desempenho para tabelas e procedimentos armazenados particionados entre os nós do cluster.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

MANEGOLD, Stefan et al. **Understanding, modeling, and improving main-memory database performance**. Universiteit van Amsterdam [Host], 2002.

MELTON, Jim; SIMON, Alan R. **Understanding the New SQL: A Complete Guide Morgan Kaufmann Publishers**. San Francisco, CA, 1993.

STONEBRAKER, Michael. **New opportunities for new sql**. *Communications of the ACM*, v. 55, n. 11, p. 10-11, 2012.

DOS SANTOS, Cedryk Augusto et al. **Análise de Escalabilidade Horizontal em um cluster HBase**. *Revista Ibérica de Sistemas e Tecnologias de Informação*, n. E27, p. 371-383, 2020.