Avaliando a Performance de SGBDs na Inserção e Consulta de Dados de Séries Temporais

Marcelo Costa de Lima[1], Daniel Lichtnow[1]

[1]Colégio Politécnico – Universidade Federal de Santa Maria (UFSM)
Santa Maria – RS – Brasil

marcelocosta084@gmail.com, daniel.lichtnow@ufsm.br

Abstract. *The aim of this work is to evaluate the performance of Database Management Systems (DBMS) in the insertion and retrieval of time series data, aiming to identify when Time Series Databases should be used instead of Relational Databases. Experiments were conducted using PostgreSQL, InfluxDB, and TimeScaleDB, varying the volume of data and assessing the execution time. The initial results and the comparison with other studies indicated the need to evaluate the requirements of each use case to determine the type of DBMS to be used for this data.*

Resumo. *O objetivo deste trabalho é avaliar a performance de SGBDs na inserção e consulta de dados de séries temporais procurando identificar quando Bancos de Dados de Séries Temporais devem ser usados no lugar de Bancos de Dados Relacionais. Foram feitos experimentos com PostgreSQL, InfluxDB e TimeScaleDB variando o volume de dados e avaliando o tempo de execução. Os resultados iniciais e a comparação com outros trabalhos indicaram a necessidade de avaliar os requisitos de cada cenário de uso para definir o tipo de SGBD a ser utilizado para estes dados.*

1. Introdução

Dados de séries temporais são resultado de medidas feitas em intervalos de tempo sequenciais. Exemplos destes dados são dados de umidade, preço de ações, dados climáticos, etc. O advento da Internet das Coisas - *Internet of Things – IoT,* onde objetos físicos estão conectados à Internet podendo a partir de sensores obter dados sobre seus estados e sobre o ambiente que os cerca, aumentou o volume deste tipo de dado. Neste contexto, foram criados SGBDs denominados Bancos de Dados de Séries Temporais (*Time Series Databases*) para lidar com dados de séries temporais [Jensen et al. 2017].

A partir disto, o objetivo deste trabalho é realizar experimentos para comparar bancos de dados de séries temporais com os bancos de dados relacionais, mediante operações de inserção/carga de dados e de consultas. Os testes iniciais indicaram que com crescimento do volume de dados, a performance de um banco de dados de séries temporais supera o do banco relacional testados. A motivação para o presente trabalho, surgiu do uso de sensores para controle de estufas de produção de verduras, hortaliças e frutas em pequenas propriedades, no contexto de projetos desenvolvidos na UFSM[1].

[1] https://portal.ufsm.br/projetos/publico/projetos/view.html?idProjeto=61876

Buscou-se então verificar até que ponto é necessário e útil adotar um banco de dados de série temporal em substituição ao modelo relacional. Os resultados obtidos foram comparados com trabalhos similares e isto indicou que é necessário definir os requisitos de uso dos dados em cenário específicos, uma vez que a performance é dependente do volume de dados e tipo de consultas.

2. Dados Temporais e Banco de Dados

Dados de uma série temporal possuem a medição repetida de parâmetros ao longo do tempo e são normalmente as medições, uma vez feitas não são atualizadas, sendo continuamente acumulados para posterior consulta e análise. A consulta a estes dados envolve normalmente conjuntos de dados. Embora a existência de abordagens para armazenar, acessar e analisar esse tipo de dados sejam relativamente novas, a preocupação com dados de séries temporais não é algo recente [Dunning et al. 2014].

Dados de série temporal podem ser armazenados em bancos relacionais e não-relacionais (*NoSQL).* Existem ainda bancos relacionais e *NoSQL* adaptados/estendidos para atender os requisitos gerados pelo armazenamento e consultas de dados de séries temporais. E existem bancos de dados criados especificamente para gerenciar dados de séries temporais, denominados Bancos de Dados de Séries Temporais. Um *survey* sobre estes últimos é apresentado em [Jensen et al. 2017] e hoje persiste a preocupação em dispor de SGBDs adequados para estes dados [Wang et al 2023].

3. Testes Realizados

Para avaliação foi feita a opção de considerar o uso de SGDBs relacionais, adaptação dos relacionais e nativos. Foram escolhidos 3 SGBDs: *PosgreSQL*, *TimeScaleDB* e *InfluxDB*. A escolha destes SGBDs foi feita com base no uso e na popularidade destes SGBDs, seguindo o levantamento presente em [DB-Engines 2024], onde é possível constatar que o *InfluxDB* é o mais popular dentre os SGBDs de séries temporais e que o *PosgreSQL* é o SGBD que ganhou mais popularidade no ano de 2023. A escolha pelo *TimeScaleDB* deve-se também ao fato de que este é uma extensão do *PostgreSQL*, podendo ser considerada uma solução intermediária entre um SGBD de séries temporais e um relacional, o que facilita seu uso dado a popularidade dos SGBDs relacionais.

O objetivo principal dos testes foi avaliar o desempenho (tempo de resposta) de inserção e leitura de dados nos bancos de dados. Para os testes de inserção de dados, foram utilizados cenários simulando diferentes cargas de trabalho, variando o tamanho dos conjuntos de dados. Foram utilizados dados fictícios. Os testes de leitura de dados foram realizados com consultas que faziam agregação dos dados.

Os testes foram realizados usando a ferramenta *Apache JMeter* [Apache, 2023] que é uma ferramenta de código aberto que realiza testes de carga e de estresse em recursos oferecidos por sistemas computacionais. O *Apache JMeter* foi originalmente projetado para testar aplicações web, mas seu uso se expandiu para outras funções de teste. Os SGBDs a serem testados e as ferramentas necessárias para o trabalho foram instalados e configurados através de um container *Docker* rodando diretamente a imagem dos bancos de dados em uma máquina (AMD Ryzen 5 4500, 3.6GHz, Cache 11MB, 6 núcleos; 16 MB de RAM, SSD 512).

Para a realização dos testes, foi criada uma base de dados com os mesmos dados, em cada um dos SGBDs. Na base de dados foi criada uma estrutura onde eram

armazenadas a hora da medição (*timestamp*), o local (cidade), a temperatura e a umidade. Foram gerados dados fictícios. No teste de carga, foi analisado o comportamento de cada SGBD ao aumentar a quantidade de registros inseridos no banco de dados. O teste de consultas envolveu submeter o banco de dados a consultas que buscavam quantidades distintas de dados nos bancos, sendo avaliado o tempo necessário para realização.

A Tabela 1 mostra os resultados do tempo gasto para realizar a inserção dos dados na tabela, sendo que o número de linhas inseridas variava a cada teste. Constata-se que quando o volume de dados aumenta o *InfluxDB* apresenta resultados melhores.

Tabela 1. Volume de dados e linhas inseridas e tempo gasto na operação

Inserções número de linhas	*PostgreSQL* tempo gasto	*TimeScaleDB* tempo gasto	*InfluxDB* tempo gasto
1.000	2,034 s	2,618 s	1,758 s
10.000	20,894 s	23,527 s	14,333 s
100.000	3 min 32s	3 min 44s	2 min 5s
1 milhão	34 min 4s	35 min 28s	16 min 37s
5 milhões	2h 50 min 57s	2h 51min 14s	1h 38min 53s

Já para os testes de consulta foram realizados dois experimentos. No primeiro experimento, foram inseridas 500.000 linhas distribuídas ao longo de 5 dias (100.00 linhas por dia) em cada banco, e depois realizada uma consulta para retornar esses dados agrupados por dia. No segundo experimento, foram inseridos 5 milhões de linhas distribuídas ao longo de 5 dias (1 milhão por dia), e após também foi feita uma consulta para retornar esses dados agrupados por dia. Os resultados dos testes são apresentados na Tabela 2 e como no caso das inserções, o SGBD *InfluxDB* apresenta resultados melhores.

Tabela 2. Tempo gasto para realização das consultas em cada SGBD

Número de linhas recuperadas	*PostgreSQL* tempo gasto	*TimeScaleDB* tempo gasto	*InfluxDB* tempo gasto
500 mil	137 ms	142 ms	91 ms
5 milhões	1291 ms	1952 ms	142 ms

Nos experimentos realizados, o *InfluxDB* apresentou os melhores resultados, especialmente na medida que o volume de dados foi sendo aumentado. Resultado semelhante pode ser constatado em [Shah, Jat e Sashidhar 2022] e [Hao 2021], onde o *InfluxDB* é comparado com o *TimeScaleDB* dentre outros bancos. Consultas com agregação de dados performam melhor no *InfluxDB* do que no *PostgreSQL* e isto é algo constatado também em [Musa 2019]. O *TimeScaleDB,* nos testes realizados mostrou performance inferior ao *InfluxDB,* sendo que chamou mais a atenção os resultados do *TimeScaleDB* em comparação com o *PostgreSQL*, onde esperava-se que o *TimeScaleDB* tivesse resultados, sendo que algo similar aconteceu em [Rasch 2018].

Buscou-se verificar como foi esta avaliação em outros trabalhos.m [Grzesik e Mrozek 2020] em uma consulta similar a utilizada no presente trabalho, os resultados do *PostgreSQL* foram melhores do que os do *TimeScaleDB*. Já os testes de inserção de dados com o *TimeScaleDB* precisam ser mais bem avaliados, pois os experimentos mostrados na Tabela 2 indicam que a performance do *TimeScaleDB* melhora em relação

ao PostgreSQL na medida em que o volume de dados aumenta, algo que é destacado em [Freedman 2017].

4. Considerações Finais

Embora iniciais, pode-se a partir dos experimentos constatar que quando o volume de dados é pequeno, não existe a necessidade de uso de bancos de dados de série temporais. Também o tipo de aplicação onde estão os dados de série temporais tem grande influência sobre a escolha do SGBD, conforme pode ser constatado em trabalhos como [Mostafa 2022] [Bamford 2023] [Wang 2023]. Assim, trabalhos futuros poderiam envolver a definição de cenários de uso específicos para os experimentos.

Referências

Apache (2023) Apache Jmeter Acessado em 09 jun 2023. Disponível em: https://jmeter.apache.org/. Acesso em : 16/02/2023

Bamford, T., et al (2023) Multi-Modal Financial Time-Series Retrieval Through Latent Space Projections. In Proceedings of the Fourth ACM International Conference on AI in Finance (pp. 498-506).

DB-Engines. DB-Engines Ranking (2023) Acessado em 09 fev 2024. Disponível em: https://db-engines.com/en/ranking. Acesso: 16/02/2023

Dunning, T. et al. (2014) Time Series Databases: New Ways to Store and Access Data. O'ReillyMedia, Incorporated.

Freedman, M. (2017) Time-series data: Why(and how) to use a relational database instead of NoSQL. Disponível em: http://tinyurl.com/2vy69sy2 Acesso: 16/02/2023

Grzesik, P., & Mrozek, D. (2020) Comparative analysis of time series databases in the context of edge computing for low power sensor networks. In Computational Science–ICCS 2020: 20th International Conference, Amsterdam, Springer

Hao, Y. et al. (2021) Ts-benchmark: A benchmark for time series databases. In: IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021. p. 588-599.

Jensen, et al C. (2017) Time series management systems: A survey. IEEE Transactions on Knowledge and Data Engineering, 29(11), 2581-2600.

Mostafa, J., et al (2022) SciTS: A Benchmark for Time-Series Databases in Scientific Experiments and Industrial Internet of Things. In Proc. of the 34th International Conference on Scientific and Statistical Database Management (pp. 1-11).

Musa, E. et al. (2019) Comparison of relational and time-series databases for real-time massive datasets. MIPRO Computers in Technical Systems, p. 1065–1070.

Rasch, E. L. (2018) Uma aplicação para carga de dados de monitoramento da geometria de linhas férreas. Trabalho de Conclusão de Curso, UFSM, Santa Maria

Shah, B.; Jat, P.; Sashidhar, K. (2022) Performance study of time series databases. arXiv preprint arXiv:2208.13982.

Wang, et al. (2023) Apache IoTDB: A Time Series Database for IoT Applications. Proceedings of the ACM on Management of Data 1.2: 1-27.