

## AVALIAÇÃO DE DESEMPENHO DE BANCOS DE DADOS DE SÉRIES TEMPORAIS

**Crístofer Rutzen, Adriano Vogel**

Laboratório de Pesquisas Avançadas para Computação em Nuvem (LARCC),  
Faculdade Três de Maio (SETREM), Três de Maio, RS, Brasil.

### RESUMO

Bancos de Dados de Séries Temporais (TSDBs) são otimizados para armazenar e consultar dados que variam ao longo do tempo. O objetivo deste trabalho é propor um novo método comparativo para avaliação de TSDBs com uma abordagem mais estruturada na condução de experimentos. Os resultados demonstram que o método proposto captura características relevantes e novas tendências de desempenho dos TSDBs. O QuestDB apresenta o melhor desempenho em *queries* (consultas) repetitivas, o InfluxDB oferece maior estabilidade para *queries ad hoc*. Por outro lado, o MongoDB tem desempenho limitado, mostrando-se inadequado para *queries* em dados de séries temporais.

**Palavras-chave:** Bancos de Dados. *Benchmark* de Desempenho. IoT.

## 1 INTRODUÇÃO

O aumento massivo de dados sendo produzidos por dispositivos e softwares demanda abordagens eficientes para armazenamento e processamento. Um exemplo de dados produzidos são os *time series datapoints* que contêm registros de séries temporais com *timestamps* e os valores coletados, como uma medição de um sensor de temperatura de *Internet of Things* (IoT) em um momento específico do tempo.

O desempenho dos bancos de dados se tornou um fator desafiador demandando abordagens mais eficientes. Dessa forma, surgiram os bancos de dados de séries temporais (TSDBs), que são bancos de dados otimizados para armazenar, processar e consultar dados que variam ao longo do tempo. Porém, os métodos de avaliação utilizados nos *benchmarks* da literatura são limitados em termos de métodos, métricas, e cenários dos experimentos (KHELIFATI et al., 2023; HAO et al., 2021).

Dessa forma, os principais objetivos deste estudo são: I) Identificar o banco de dados aberto mais otimizado para dados de séries temporais; II) Avaliar se a tendência de desempenho da literatura é semelhante com dados reais de sensores de IoT (KHELIFATI et al., 2023); III) Medir a variação entre as execuções das *queries* nos diferentes bancos de dados, que é algo usado em diferentes casos de uso; IV) Medir e

comparar o desempenho de diferentes bancos de dados de forma mais isolada.

As principais contribuições deste artigo são: I) Uma nova metodologia de avaliação para *Time Series Databases*; II) Um novo *benchmark* para *TSDBs* com novas *queries* e dados reais de entrada; III) Avaliação experimental de desempenho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Bancos de dados de séries temporais (TSDBs) são soluções projetadas especificamente para lidar com dados organizados cronologicamente, possibilitando seu armazenamento, gerenciamento e consulta. Esse tipo de dado é essencial em cenários que exigem monitoramento constante, como sensores IoT, análise de métricas de desempenho e registros de sistemas. Esses sistemas suportam técnicas de compressão de dados, indexação temporal e otimização para inserções em larga escala. Contudo, a variedade de abordagens e arquiteturas adotadas por diferentes TSDBs dificulta comparações diretas entre eles. Nesse contexto, o uso de *benchmarks* se apresenta como uma estratégia fundamental para avaliar essas soluções de maneira uniforme, considerando tanto a capacidade de ingestão contínua quanto a eficiência nas consultas (KHELIFATI et al., 2023).

Existem *benchmarks* comparando o desempenho de banco de dados de séries temporais e geradores de dados artificiais que podem aumentar sua base de dados para o experimento (KHELIFATI et al., 2023; HAO et al., 2021). Entretanto, existe também a falta de informações comparando a diferença entre a primeira *query* de um tipo e repetições da mesma, algo que deve-se levar em consideração quando o banco de dados possui *queries* frequentes. Junto a isso, não há comparação de resultados entre dados reais e dados artificiais utilizados nos experimentos ou uma explicação da correlação entre os resultados e a arquitetura dos TSDBs.

## 3 METODOLOGIA

Foram definidas as seguintes *queries* considerando a relevância para o domínio de IoT. A **Query 1** consulta e retorna todos os dados da base de dados, escolhida com o objetivo de verificar a velocidade de consulta (*scans*) em volumes significativos de dados. A **Query 2** consulta com filtro para retornar os dados com temperatura maior que 20°C, foi selecionada pois é uma das consultas comuns para atualizar *dashboards*. Por fim, a **Query 3** calcula e retorna a média de todos os dados de temperatura e

umidade da base de dados, sendo uma *query* mais robusta de monitoramento, como para escolher um período de tempo e analisar a temperatura média.

Foram utilizados dados de entrada reais e idênticos coletados periodicamente de sensores de temperatura e umidade para aumentar a representatividade dos dados. Cada *datapoint* possui 4 campos: “*timestamp*”, “*value*” que salva um *Float* com o valor de temperatura ou umidade, “*field*” que define o que foi coletado (temperatura ou umidade) e o campo “*sensor*” que define qual sensor coletou o dado. Os *datapoints* são salvos em formato CSV ou *Comma-Separated Values*. Os *datapoints* foram coletados durante treze dias, totalizando 28.247.749 *datapoints* (1,6 GB de espaço em disco).

Foram comparados os seguintes bancos de dados. **InfluxDB** (INFLUXDB, 2025), que é um Banco de dados com suporte a métricas temporais na versão *Open Source*. O InfluxDB tem o armazenamento orientado a colunas e utiliza TSM (*Time Structured Merge Tree*) para compressão dos dados armazenados. O banco indexa automaticamente métricas. **QuestDB** (QUESTDB, 2025) também foca em séries temporais na versão *open source* (8.2.1) utilizada. A estrutura de dados do QuestDB é orientada a colunas e emprega índices baseados em tempo. **MongoDB** (MONGODB, 2025) é classificado como banco de dados NoSQL orientado a documentos, mas possui suporte a séries temporais em suas versões recentes. Sua compatibilidade com Linux e o uso em larga escala justificam sua inclusão neste estudo. A versão utilizada para os experimentos foi o *MongoDB Community* 8.0.9, com a licença *Server Side Public License*. O processamento dos dados é orientado a linhas (*row-based*) no MongoDB.

Os experimentos foram executados em um servidor alocado na nuvem privada no LARCC (*Laboratory of Advanced Research on Cloud Computing*) <sup>1</sup>, equipado com processador Intel Xeon Silver 4108 com 8 cores e 16 *Hyperthreads* e 64 GB de memória RAM DDR4. Os bancos foram instalados em VMs utilizando o sistema operacional Ubuntu *Server* 24.04. Cada VM foi criada com 4 CPUs, 16 GB de RAM e 50 GB de espaço em disco, e todos os bancos foram instalados em VMs instanciadas no mesmo servidor. Durante os experimentos, apenas a VM do banco sendo testado estava em execução para diminuir interferências externas.

#### 4 APRESENTAÇÃO, ANÁLISE E DISCUSSÃO DOS RESULTADOS

A Figura 4 apresenta o desempenho considerando o tempo de execução das *queries*, onde se nota que as *queries* tiveram a tendência da primeira execução ter

<sup>1</sup> <<https://larcc.setrem.com.br/>>

desempenho inferior às médias das demais execuções. A primeira execução da *query* é relevante pois mostra o desempenho sem uso de *caches*. No caso do QuestDB, a primeira repetição da *Query 1* é 13,4x mais lenta do que as demais e 3,4x na *Query 2*. Na *Query 3* o QuestDB demonstra uma variação menor. A variação significativa percebida entre as repetições das mesmas *queries* no QuestDB ocorre pois na primeira execução os dados precisam ser carregados dos discos para a memória (colunar por design) e criar estruturas intermediárias de execução (como índices temporários). Nas repetições, esses dados já estão totalmente carregados em memória, reduzindo o tempo de acesso. O QuestDB pode realizar otimizações de plano de execução, em que a primeira execução pode envolver a análise e geração de um plano de execução completo, já nas repetições, esse plano pode ser reutilizado ou otimizado, tornando as execuções seguintes mais rápidas. No MongoDB essa variação é também bastante perceptível, sendo a primeira repetição 4.49x mais lenta na *Query 1* e 3.5x na *Query 2*.

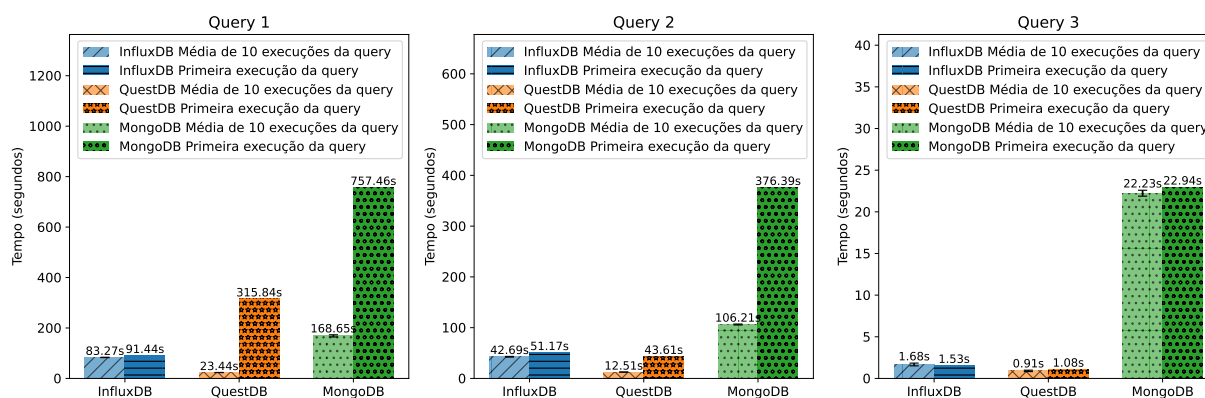


Figura 1 – Tempo de Execução das *Queries* (em segundos).

O MongoDB apresentou desempenho inferior em todos os experimentos. Na *Query 3*, foi 13,2 vezes e 24,4 vezes mais lento que o InfluxDB e o QuestDB, respectivamente. Como armazenamento de dados no MongoDB é *row-based* (cada *datapoint* é armazenado em uma linha), a busca por dados específicos de campos como “temperatura” ou “umidade” se torna menos eficiente, pois precisa percorrer todos os campos para chegar ao campo buscado, justificando seu desempenho inferior nas *queries*.

O InfluxDB apresenta resultados estáveis e as médias são mais rápidas que a primeira execução, pois ele é projetado com otimizações específicas para *queries* temporais, como o armazenamento em blocos organizados por tempo, indexação automática por série e campo e operações de agregação otimizadas internamente.

O InfluxDB normalmente carrega apenas os blocos de dados necessários para a resolução da *query* reduzindo o impacto de *caches*, contribuindo para a estabilidade das execuções percebida nos experimentos.

## 5 CONCLUSÃO

Neste artigo, foi proposto um novo método para avaliar TSDBs, uma forma mais estruturada de executar experimentos de forma mais isolada e justa possível, e avaliação experimental de tecnologias com código disponível e gratuito. Os resultados demonstram que o método utilizado na literatura que executa as *queries* com os dados em cache (KHELIFATI et al., 2023) não é representativo para *queries ad-hoc*.

O QuestDB apresenta o melhor desempenho para *queries* repetitivas, sendo adequado para casos de uso como *dashboards* e para *queries* robustas que envolvam filtros e agregações, como visto nas *Queries 2* e *3*. Por outro lado, a estabilidade do InfluxDB o torna relevante para *ad-hoc queries*, e.g., para analistas de dados que executam *queries* no nível analítico como coletar dados para treinar modelos. Mesmo suportando séries temporais, o desempenho do MongoDB não é comparável ao de bancos de dados especializados em séries temporais.

Existem diversos trabalhos futuros, destacando-se: I) Complementar a avaliação incluindo execuções em *clusters* com tolerância a falhas; III) Propor novas métricas e métodos de avaliação para a eficiência na utilização dos recursos computacionais para a ingestão de dados, execução de *queries*, compressão de dados.

## REFERÊNCIAS

HAO, Y. et al. TS-Benchmark: A Benchmark for Time Series Databases. In: **IEEE 37th International Conference on Data Engineering**. IEEE, 2021. p. 588–599. Disponível em: <<https://doi.org/10.1109/ICDE51399.2021.00057>>.

INFLUXDB. **InfluxData (Home Page)**. 2025. Available on: <https://www.influxdata.com>. Access date: 09 June.

KHELIFATI, A. et al. TSM-Bench: Benchmarking Time Series Database Systems for Monitoring Applications. **Proc. VLDB Endow.**, v. 16, n. 11, p. 3363–3376, 2023.

MONGODB. **MongoDB: The World's Leading Modern Database (Home Page)**. 2025. Available on: <https://www.mongodb.com>. Access date: 09 June.

QUESTDB. **QuestDB**. 2025. Available on: <https://github.com/questdb/questdb>. Access date: 09 June.