

Comparação e Benchmark de Sistemas de Gerenciamento de Bancos de Dados Não Convencionais*

Arthur Lennon Firmino da Silva¹, Vitor Oliveira Ropke¹

¹Departamento de Computação – Universidade Federal Rural do Semi-Árido (UFERSA)

Caixa Postal 137 – 59.625-900 – Mossoró – RN – Brasil

{arthur.silva74206,vitor.ropke}@alunos.ufersa.edu.br

Abstract. *This study emphasizes the importance of careful evaluation and comparison of non-conventional database management systems. Through virtualization it was possible to test and explore the evaluation criteria with different commands and database management systems. Most systems achieved approximate execution times within a few milliseconds. It is noteworthy that the appropriate choice of these technologies can significantly impact system performance and customer satisfaction.*

Resumo. *Este estudo enfatiza a importância da avaliação criteriosa e da comparação de sistemas de gerenciamento de bancos de dados não convencionais. Através da virtualização foi possível testar e explorar os critérios de avaliação com diferentes comandos e sistemas de gerenciamento de bancos de dados. A maioria dos sistemas obteve tempos de execução aproximados com algumas dezenas de milissegundos. Destaca-se que a escolha adequada dessas tecnologias pode impactar significativamente o desempenho do sistema e a satisfação do cliente.*

1. Introdução

No universo da tecnologia da informação, a gestão de dados desempenha um papel de destaque, tornando-se cada vez mais crucial à medida que a complexidade e o volume de informações crescem exponencialmente. Nesse contexto, a busca por sistemas de gerenciamento de bancos de dados não convencionais tem se tornado uma necessidade premente. Esses sistemas representam alternativas inovadoras aos tradicionais bancos de dados relacionais, permitindo o armazenamento e processamento eficaz de dados em diversos formatos e diferentes tipos de estruturação [Grolinger *et al.* 2013].

Comparar e realizar benchmarks em sistemas de gerenciamento de bancos de dados não convencionais é uma prática essencial para a seleção e avaliação dessas soluções [Domaschka 2021]. Este artigo explora a importância dessa abordagem e os principais critérios usados para avaliar o desempenho e a eficiência desses sistemas. Além disso, ele analisa as características distintas e pontuais, apresentando casos de uso relevantes com exemplos práticos.

Este artigo está organizado da seguinte forma: a seção 2 apresenta trabalhos relacionados a este. A seção 3 descreve as especificações de hardware da plataforma usada, bem como o processo de Live Boot, a caracterização do processo de instalação dos SGBDs usados e dos testes realizados com base nessas tecnologias. A seção 4 mostra os resultados apresentados utilizando gráficos para formalizar a ocorrência dos processos. Por fim, a seção 5 evidencia a relevância de uma análise minuciosa e da comparação entre sistemas de gestão de bancos de dados não tradicionais.

* Adaptação para OpenOffice.org 1.1 feita por Roland Teodorowitsch (roland@ulbra.tche.br) em 29 mar. 2005.

2. Trabalhos relacionados

Segundo Curnow (1976), um programa utiliza um benchmark para testar sua performance. Sendo assim, benchmarks (que também podem ser programas) são importantes meios de averiguação de diferentes sistemas. É possível comparar o desempenho em relação a outros programas que possuem propósitos iguais ou semelhantes. Caso seja um aplicativo inovador, o benchmark permite avaliar se possui um nível mínimo de performance. Desta forma, os desenvolvedores procuram, ao máximo, diminuir os valores de tempo de execução geral de seu sistema, até mesmo avaliando individualmente, cada módulo.

De parte da armazenagem de dados, há uma diferença fundamental entre banco de dados (BD), que armazena dados, e sistema de gerenciamento de bancos de dados (SGBD), que manipula BDs [Ogli 2022]. Chopra (2010) explica que o SGBD é um sistema de software que permite ao usuário definir, criar, manter e controlar o acesso ao banco de dados. O autor complementa que um sistema de banco de dados é a união de um BD com um SGBD. Nele, o usuário consegue executar comandos básicos de inserção, leitura, edição e remoção (conhecido como CRUD [Create, Read, Update e Delete]).

Em relação aos testes de performance dos SGBDs, Ayub e Ali (2018) usaram o Microsoft SQL Server para comparar o desempenho dos bancos de dados armazenados na memória principal do computador (popularmente conhecido como memória RAM) e no disco de armazenamento (disco rígido ou HD). Observaram que a execução na memória obteve menores tempos de execução. Contudo, os autores deixam claro que a migração de um sistema de armazenamento para outro, causa gargalos e possíveis conflitos envolvendo a indexação, devendo ser feita com cautela.

Kolonko (2018) utilizou um programa de benchmark chamado YCSB para comparar o desempenho dos SGBDs OracleDB e MongoDB. Este último alcançou os melhores resultados quanto à latência e taxa de transferência, sendo o OracleDB, mais estável. Ou seja, os melhores e piores resultados desse SGBD são próximos e constantes. Pandey (2020) fez a mesma abordagem, usando o YCSB, mas ao invés do OracleDB, utilizou o MySQL. Novamente, o MongoDB alcançou melhores resultados de performance, destacando-se em altos números de operações (vários comandos CRUD).

Samonte e Samonte (2022) fizeram testes com foco na tratativa de erros em algumas operações de transação no banco de dados (leitura, escrita, atualização ou remoção), envolvendo controle de concorrência, entre outros. Os SGBDs utilizados pelas autoras foram MySQL e MongoDB. Assim como em Kolonko (2018) e Pandey (2020), os resultados de desempenho foram melhores para o MongoDB.

Liang e Lu (2010) escolheram o melhor SGBD para o aplicativo MBS, um conjunto de soluções para análise de dados de espectrometria de massa de alta dimensão. Eles selecionaram mais de 20 SGBDs para testes, diminuindo para 4 o número de sistemas que foram “classificados” após análise qualitativa. Os escolhidos foram PostgreSQL, SQLite, Firebird e MySQL. Desses, o Firebird alcançou os melhores resultados, cumprindo os requisitos que os autores propuseram para a aplicação MBS.

Pires, do Nascimento e Salgado (2008) usaram o aplicativo de benchmark AS³AP, dedicado a fazer testes em bancos de dados. Foi escolhido o MySQL e PostgreSQL para comparação. O primeiro alcançou resultados melhores na maioria dos testes, onde o último obteve resultado superior no módulo de carga e estrutura (a tabela é criada e dados são inseridos a partir de arquivos de texto).

Por fim, Vieira (2020) testou o MySQL, Oracle Database, SQL Server, IBM DB2 e PostgreSQL usando o HammerDB como aplicativo de benchmark, também

dedicado a testes de bancos de dados, assim como o AS³AP. O SQL Server obteve o melhor resultado, na média NOPM (Novos Pedidos por Minuto, métrica usada pelo benchmark para comparar os BDs), seguido por DB2, Oracle, PostgreSQL e, por fim, MySQL.

Diante desses trabalhos, este estudo destaca, além dos comparativos de desempenho, a usabilidade e facilidade no uso dos SGBDs; marca a presença ou ausência de interface gráfica que impacta significativamente a experiência do usuário, principalmente o iniciante; e mostra que, em alguns momentos, o processo de instalação do SGBD é mais complexo pelos meios oficiais, levando a outros métodos mais simples de instalação, como fontes do próprio sistema operacional, por exemplo.

3. Metodologia

3.1. Especificações de hardware da plataforma de testes

Os testes foram realizados numa máquina virtual, cujo hospedeiro possui as seguintes especificações de hardware: placa-mãe ASUS H110M-CS/BR; processador Intel Core i3-7100; memória Corsair Vengeance LPX 32 GB (2×16 GB) 2.400 MHZ DDR4; placa de vídeo ASUS CERBERUS-GTX1050TI-O4G; SSD SATA Crucial MX500 500 GB. A máquina virtual usada é o VirtualBox da Oracle e foi configurada com as seguintes especificações: 16 GB de memória principal; 2 núcleos de processador; 100 GB de armazenamento; 128 MB de memória de vídeo e aceleração 3D habilitada. Todas as outras configurações foram mantidas como padrão. O sistema operacional do hospedeiro e da máquina virtual é o Linux Mint 21.2 Cinnamon 64-bit.

3.2. Live Boot do Linux

O Linux Mint foi executado por uma tecnologia do Linux que permite o uso do sistema operacional sem a necessidade de instalá-lo numa unidade de armazenamento permanente, como SSD ou HDD. Essa tecnologia é chamada de Live Boot Linux. Ela foi utilizada porque, por padrão, todas as instalações no sistema são desfeitas após a reinicialização do sistema. Sendo assim, após a instalação e teste dos SGBDs, o sistema é reiniciado para iniciar os testes no próximo SGBD. Dessa forma, os testes não serão impactados negativamente por não terem outros SGBDs consumindo recursos da máquina. O fluxograma mostrado na [Figura 1](#), mostra com mais clareza esse procedimento.

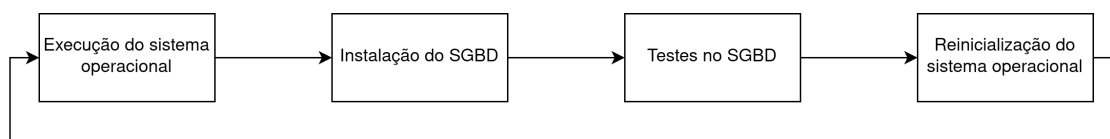


Figura 1. Fluxograma do procedimento para testar cada SGBD

3.3. Instalação dos SGBDs

Os SGBDs foram preferencialmente instalados a partir do site oficial de cada sistema. Quando o processo de instalação não é claro ou é complexo, opta-se por utilizar os pacotes do Linux disponíveis nos repositórios do Linux Mint, acessados através do Terminal. Apesar de alguns pacotes não estarem na última versão em comparação com a observada no site oficial, a intenção é fazer o SGBD estar pronto para uso o mais rápido possível, com o mínimo de passos necessários.

Foram 7 SGBDs instalados. São eles: MongoDB, MySQL, PostgreSQL, SQLite, Cassandra, Neo4j e OrientDB. Estes foram escolhidos por serem gratuitos e por mostrarem, de alguma forma, o tempo de execução dos comandos no próprio SGBD. Alguns possuem um programa separado para interface gráfica, tornando a interação

com o usuário mais simples. Outros são executados apenas em terminal. Sendo assim, os SGBDs foram instalados usando os recursos mínimos para funcionarem. A [Tabela 1](#) mostra o tipo de instalação de cada SGBD (se foi pelo site oficial ou pelos pacotes do Linux) e o armazenamento ocupado pelo aplicativo e suas dependências, caso tenham.

Tabela 1. Tipo de instalação e armazenamento ocupado por cada SGBD

SGBD	Tipo de instalação	Armazenamento ocupado
MongoDB 7.0.2	Site oficial	385 MB
MySQL 8.0.34	Pacotes do Linux	241 MB
PostgreSQL 16.0	Site oficial	181 MB
SQLite 3.41.2	Pacotes do Linux	3 MB
Cassandra 4.1.3	Site oficial	59 MB
Neo4j 1.5.8	Site oficial	790 MB
OrientDB 3.2.23	Site oficial	66 MB

Apesar do MySQL e PostgreSQL serem normalmente utilizados com bancos de dados convencionais, eles possuem recursos que os fazem trabalhar com objetos, sendo essas estruturas não convencionais em bancos de dados.

3.4. Testes

Os SGBDs foram submetidos à execução de um CRUD, onde cada comando teve seu tempo de execução avaliado e, se necessário, convertido. A ordem foi a seguinte: inserção de um dado (ROW); leitura de todos os dados; leitura de um dado; atualização do dado; remoção de um dado.

O dado inserido possui os atributos “id”, “nome”, “sobrenome” e “idade” com seus respectivos valores iniciais “1”, “Fulano”, “Sicrano” e “99”. Após a atualização do dado, os valores ficaram da seguinte forma: “1”, “Sicrano”, “Fulano” e “1”. A [Tabela 2](#) mostra os valores desse dado antes (no início do teste) e depois (no final do teste).

Tabela 2. Valores do dado antes e depois da atualização

Atributos	Antes da atualização	Depois da atualização
Id	1	1
Nome	Fulano	Sicrano
Sobrenome	Sicrano	Fulano
Idade	99	1

4. Resultados

Os SGBDs exibiram seus resultados usando unidades de medida de tempo, diferentes. Foram nanossegundos, microssegundos, milissegundos e segundos. A medida escolhida foi o milissegundo, pois, os resultados se concentraram nessa unidade e a maioria dos SGBDs exibiram essa medida. A [Tabela 3](#) mostra os resultados dos tempos de execução de cada comando CRUD por cada SGBD.

Tabela 3. Valores do tempo de execução de cada comando de cada SGBD

SGBD	Tempo de execução em milissegundos					
	Create Inserir	Read all Ler todos	Read One Ler um	Update Atualizar	Delete Remover	Soma Total
MongoDB	0,214	0,216	0,335	0,177	0,156	1,098
MySQL	5,218	0,478	0,676	3,040	1,750	11,162
PostgreSQL	1,925	0,954	1,857	1,261	0,848	6,845
SQLite	0,236	0,289	0,182	0,157	0,064	0,928
Cassandra	26,52	5,358	7,733	31,29	3,583	74,484
Neo4j	3	2	5	6	9	25
OrientDB	159	85	39	38	55	376

MongoDB e SQLite conseguiram os menores valores de execução dos comandos, com todos ficando abaixo de 1 milissegundo. SQLite conseguiu o melhor resultado no total, seguido por MongoDB, PostgreSQL, MySQL, Neo4j, Cassandra e, por fim, OrientDB. A classificação final, em ordem crescente da soma dos tempos de execução dos comandos, é mostrada na [Tabela 4](#).

Tabela 4. Classificação dos SGBDs em ordem crescente das somas dos tempos de execução dos comandos

SGBD	Soma dos tempos de execução
SQLite	0,928
MongoDB	1,098
PostgreSQL	6,845
MySQL	11,162
Neo4j	25
Cassandra	74,484
OrientDB	376

A maioria dos SGBDs conseguiram baixos tempos de execução total de seus comandos, com OrientDB obtendo um valor muito alto em comparação aos outros. A distribuição desses valores é melhor observada na [Figura 2](#).

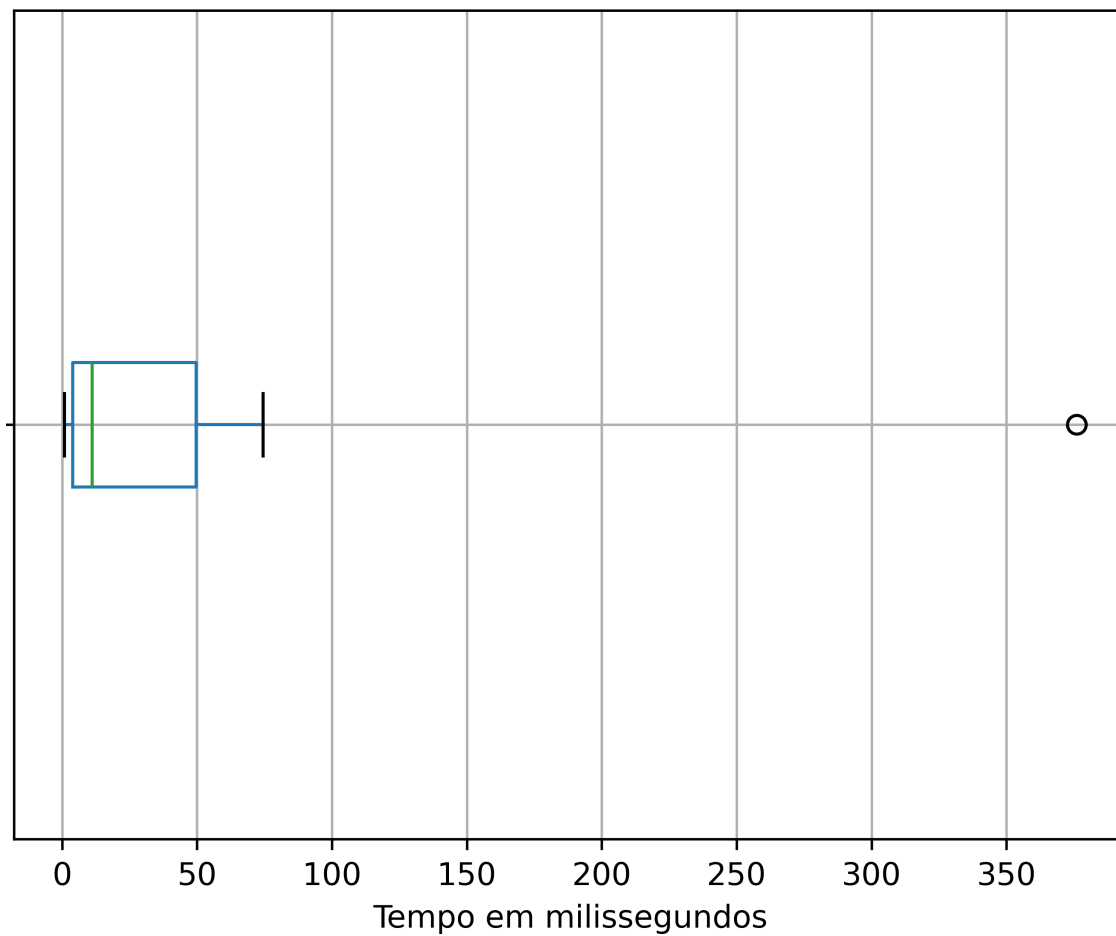


Figura 2. Distribuição da soma dos tempos de execução de cada SGBD

Há outra análise a ser feita, relacionada a quanto cada SGBD gasta seu tempo em cada comando. A [Tabela 3](#) mostra em quantos milissegundos cada comando foi realizado. A [Tabela 5](#) mostra as porcentagens de cada comando em relação à soma deles.

Tabela 5. Porcentagem de cada comando em relação ao total do tempo de execução para cada SGBD

SGBD	Create	Read All	Read One	Update	Delete
MongoDB	19,49%	19,67%	30,51%	16,12%	14,21%
MySQL	46,75%	4,28%	6,06%	27,24%	15,68%
PostgreSQL	28,12%	13,94%	27,13%	18,42%	12,39%
SQLite	25,43%	31,14%	19,61%	16,92%	6,90%
Cassandra	35,60%	7,19%	10,38%	42,01%	4,81%
Neo4j	12,00%	8,00%	20,00%	24,00%	36,00%
OrientDB	42,29%	22,61%	10,37%	10,11%	14,63%

É possível observar que o MongoDB gastou a maioria do tempo em leitura e a menor parte em remoção, assim como SQLite e ao contrário de Neo4j. MySQL teve maior tempo em inserção e menor em leitura. PostgreSQL teve maior tempo em inserção e menor em remoção. Cassandra teve maior tempo em atualização e menor em remoção. OrientDB passou a maioria do tempo na inserção e, na menor parte, atualizando. Em resumo, todos os SGBDs tem sua particularidade em relação a qual comando é executado em maior ou menor tempo.

5. Conclusões

Apesar dos SGBDs possuírem propósitos diferentes e trabalharem com diferentes volumes de dados, a comparação é uma forma importante de medir seu desempenho em relação aos concorrentes. Isso passa desde o processo de instalação até a execução dos comandos. Vários passos para se instalar um SGBD, documentação confusa, ausência de interface gráfica ou até interface gráfica confusa traz impactos negativos para os clientes do banco de dados em questão.

A simplificação do processo de instalação, facilidade no uso da interface gráfica, e baixo tempo de resposta e execução de comandos, são de suma importância para a boa avaliação de um SGBD. As empresas que gerenciam os SGBDs devem buscar constantemente comparar outros serviços para aperfeiçoar seus sistemas aos seus clientes.

Além disso, após a execução de milhões de comandos, alguns milissegundos a mais de tempo podem fazer a diferença. Sendo assim, a otimização dos SGBDs é importante para alcançar tempos decrescentes de operação para salvar o tempo de carregamento de aplicativos.

Referências

- Ayub, M. B. e Ali, N. (2018) “Performance comparison of in-memory and disk-based databases using transaction processing performance council (TPC) benchmarking”, *Journal of Internet and Information Systems*, 8(1), p. 1–8. doi: 10.5897/jiis2018.0106.
- Chopra, R. (2010) *Database Management System (DBMS) A Practical Approach*. Nova

- Deli, India: S Chand.
- Curnow, H. J. (1976) “A synthetic benchmark”, *The computer journal*, 19(1), p. 43–49. doi: 10.1093/comjnl/19.1.43.
- Domaschka, J. (2021) *What is Database Benchmarking?*, benchANT. Disponível em: <https://benchant.com/blog/database-benchmarking>.
- Grolinger, K. *et al.* (2013) “Data management in cloud environments: NoSQL and NewSQL data stores”, *Journal of Cloud Computing Advances Systems and Applications*, 2(1). doi: 10.1186/2192-113x-2-22.
- Kolonko, K. (2018) *Performance comparison of the most popular relational and non-relational database management systems*. Blekinge Institute of Technology. Disponível em: <https://www.diva-portal.org/smash/get/diva2:1199667/FULLTEXT02.pdf>.
- Liang, X. e Lu, Y. (2010) *EVALUATION OF DATABASE MANAGEMENT SYSTEMS*. Halmstad University. Disponível em: <https://www.diva-portal.org/smash/get/diva2:367006/fulltext01>.
- Ogli, R. A. R. (2022) “THE DIFFERENCE BETWEEN THE CONCEPTS OF DATABASE AND DATABASE MANAGEMENT SYSTEM”, *Archive of Conferences*, p. 33–34. Disponível em: <https://www.conferencepublication.com/index.php/aoc/article/view/1845>.
- Pandey, R. (2020) *Performance benchmarking and comparison of cloud-based databases MongoDB (NoSQL) vs MySQL (relational) using YCSB*. National College of Ireland. doi: 10.13140/RG.2.2.10789.32484.
- Pires, C. E. S., do Nascimento, R. O. e Salgado, A. C. (2008) “Comparativo de Desempenho entre Bancos de Dados de Código Aberto”. Disponível em: https://www.researchgate.net/publication/242213657_Comparativo_de_Desempenho_entre_Bancos_de_Dados_de_Codigo_Aberto.
- Samonte, M. J. C. e Samonte, S. A. (2022) “Evaluation of Database Management Systems and Techniques Applied in Distributed Systems”, *Proceedings of the International Conference on Industrial Engineering and Operations Management*, p. 196–204. Disponível em: <https://ieomsociety.org/proceedings/2022nigeria/29.pdf>.
- Vieira, L. F. (2020) *COMPARAÇÃO DE PERFORMANCE DE SISTEMAS GERENCIADORES DE BANCO DE DADOS*. Universidade Tecnológica Federal do Paraná. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/25977>.