

Estudo Comparativo entre MongoDB e PostgreSQL usando Embedding Documents

André Merlo¹, Jorge N. S. Pavão¹, Luiz C. de Freitas¹,
Jorge Soares¹, Diego Brandão¹, Kele Belloze¹

¹Programa de Pós-Graduação em Ciência da Computação
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (Cefet/RJ)
Rio de Janeiro – RJ – Brasil

{jorge.soares, kele.belloze}@cefet-rj.br

Abstract. *Database Management Systems (DBMSs) play a crucial role in e-Science due to the need to manage and analyze large volumes of data efficiently. MongoDB and PostgreSQL are two of the most used DBMSs today, each with unique characteristics that meet different data storage and processing needs. This work presents a comparative analysis between MongoDB and PostgreSQL, evaluating the performance and efficiency of both systems when dealing with complex queries and data analysis workloads. Preliminary results indicate that although MongoDB shows efficiency in specific queries, PostgreSQL outperformed in most tests.*

Resumo. *Os Sistemas de Gerenciamento de Banco de Dados (SGBDs) desempenham um papel primordial em e-Science para gerenciar e analisar grandes volumes de dados de forma eficiente. MongoDB e PostgreSQL são dois dos SGBDs amplamente utilizados, cada um com características próprias que atendem a diferentes necessidades de armazenamento e processamento de dados. Este trabalho apresenta uma análise comparativa entre o MongoDB e o PostgreSQL, avaliando o desempenho e a eficiência de ambos os sistemas ao lidar com consultas complexas e cargas de trabalho de análise de dados. Resultados preliminares mostram que, embora o MongoDB mostre eficiência em consultas específicas, o PostgreSQL teve melhor desempenho na maioria dos testes.*

1. Introdução

Nos últimos anos, a tecnologia de bancos de dados evoluiu significativamente para atender às crescentes demandas de armazenamento e processamento de dados em larga escala no contexto de *e-science* (Khan et al., 2023). Em meio a essa evolução, os bancos de dados NoSQL (*Not Only SQL*) emergiram como uma solução poderosa para casos de uso que exigem flexibilidade, escalabilidade e desempenho. Diferente dos sistemas de banco de dados relacional tradicionais, que utilizam tabelas rígidas e um esquema fixo, os bancos de dados NoSQL oferecem estruturas de armazenamento diversificadas, como chave-valor, documentos, colunas largas e grafos.

Dentre os Sistemas de Gerenciamento de Banco de Dados (SGBD) NoSQL, o MongoDB¹ se destaca por sua abordagem flexível e escalável para o armazenamento e

¹<https://www.mongodb.com/docs/manual>

recuperação de dados. O sistema adota um modelo orientado a documentos, no qual os dados são armazenados em documentos BSON (Binary JSON), que são estruturas de dados flexíveis e autocontidas. Esses documentos podem conter campos de diferentes tipos de dados, como strings, números, arrays e até mesmo outros documentos aninhados, permitindo uma representação rica e hierárquica dos dados. Uma das características mais distintivas do MongoDB é a sua capacidade de *embedding documents*, que permite incorporar documentos relacionados dentro de um único documento. Essa técnica simplifica a modelagem de dados ao eliminar a necessidade de junções complexas, tornando as consultas mais eficientes e reduzindo a complexidade do esquema. Além disso, ele oferece suporte a índices secundários e busca em texto completo, permitindo consultas rápidas e flexíveis em grandes conjuntos de dados.

Outro aspecto importante do MongoDB é sua capacidade de escalar horizontalmente, distribuindo os dados em vários servidores e garantindo alta disponibilidade e tolerância a falhas. O MongoDB utiliza o conceito de *sharding* para distribuir os dados em vários nós, permitindo que o sistema mantenha o desempenho e a capacidade de armazenamento mesmo com o crescimento exponencial dos dados. Ainda, o sistema oferece uma variedade de recursos avançados, como suporte a transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade) em nível de documento, agrupamentos complexos usando o framework de agregação e ferramentas de análise embutidas, como o MongoDB Compass. Esses recursos tornam o MongoDB uma escolha popular para uma ampla gama de aplicativos, desde pequenas *startups* até grandes empresas, que buscam uma solução flexível e escalável para suas necessidades de armazenamento de dados.

Dadas as características do MongoDB é possível notar que ele apresenta recursos bastante compatíveis com o PostgreSQL², um SGBD objeto-relacional, que é reconhecido por sua robustez, confiabilidade e extensibilidade, além de sua capacidade avançada de extensão e personalização. Outro aspecto fundamental do PostgreSQL é sua robustez em termos de transações e concorrência. O PostgreSQL implementa o modelo ACID, garantindo a integridade dos dados e a confiabilidade das transações mesmo em ambientes de alta concorrência. Em termos de desempenho, o PostgreSQL é altamente aprimorado, com um otimizador de consultas sofisticado que busca encontrar o plano de execução mais eficiente para consultas complexas. Com a introdução de recursos como particionamento de tabelas e paralelismo de consultas, o PostgreSQL tem sido capaz de lidar com cargas de trabalho cada vez mais exigentes em ambientes de produção.

Diante deste contexto, este trabalho apresenta uma análise comparativa entre os sistemas MongoDB e PostgreSQL, avaliando o desempenho e a eficiência de ambos ao lidar com consultas complexas e cargas de trabalho de análise de dados. A metodologia adota o *Transaction Processing Performance Council Benchmark H* (TPC-H)³, um *benchmark* amplamente reconhecido e utilizado para avaliar o desempenho de SGBDs em cargas de trabalho de análise de decisões. Os resultados foram comparados em termos de tempo de execução em segundos. Os resultados mostram que o MongoDB destaca-se pela sua eficiência na recuperação de dados complexos e aninhados, em consequência à sua capacidade de *embedding documents*, enquanto que PostgreSQL apresenta forte capacidade de processamento para consultas relacionais complexas, especialmente quando

²<https://www.postgresql.org/files/documentation/pdf/16/postgresql-16-A4.pdf>

³<https://www.tpc.org/tpch/>

otimizado corretamente com índices e estratégias de consulta eficientes.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 oferece uma breve revisão dos trabalhos relacionados, seguida pela metodologia na Seção 3. A avaliação experimental e os resultados são discutidos nas Seções 4 e 5, respectivamente. Finalmente, a Seção 6 apresenta as considerações finais e as propostas de continuação deste trabalho.

2. Trabalhos Relacionados

A literatura apresenta diversos trabalhos que abordam aspectos diferentes do MongoDB em comparação com bancos de dados relacionais como o MySQL, e características de modelos relacionais como a normalização de dados. Esta seção apresenta aqueles que tem maior semelhança com o trabalho aqui descrito.

Zhao et al. (2013) exploraram o desafio de modelagem de dados do MongoDB sob a perspectiva do modelo relacional. Este estudo fornece uma visão sobre como adaptar os princípios e técnicas do modelo relacional para atender às necessidades de modelagem de dados em um ambiente NoSQL como o MongoDB.

O estudo conduzido por Kanade et al. (2014) examinou como o MongoDB aborda a organização e consulta de dados por meio de técnicas de normalização e *embedding*. Os autores analisaram as diferenças entre o modelo orientado a documentos do MongoDB e os modelos relacionais tradicionais, explorando o impacto dessas abordagens no desempenho das consultas e na estruturação dos dados.

Győrödi et al. (2015) conduziram um estudo comparativo entre MongoDB e MySQL, explorando as diferenças fundamentais entre os dois SGBDs. A pesquisa analisou aspectos como desempenho, capacidade de escalabilidade e flexibilidade na modelagem de dados, fornecendo *insights* valiosos para profissionais de TI e pesquisadores interessados em escolher a melhor solução para suas necessidades específicas. Outro trabalho que compara o MongoDB e MySQL é apresentado por Filip and Čegan (2020). Os autores utilizaram uma base de dados climatológicos, avaliando o uso de índices para melhorar o desempenho de ambos os sistemas.

Llano-Ríos et al. (2020) realizaram uma comparação entre dois sistemas SQL (MariaDB e PostgreSQL) com dois sistemas NoSQL (MongoDB e CouchBase) utilizando como *benchmark* o TPC-H. Eles focaram em tarefas voltadas à tomada de decisão, não avaliando todas as consultas do TPC-H. Apesar disso, os autores mostraram que o *design* do armazenamento de documentos apresenta desafios e que nem sempre é claro qual *design* funcionaria para consultas complexas e *ad hoc*.

Makris et al. (2021) compararam o PostgreSQL com o MongoDB utilizando uma base de dados de séries-temporais com dados de rastreamento de embarcações. Eles verificaram uma superioridade do PostgreSQL em quase todas as consultas realizadas, com exceção das consultas de interseção de polígonos. Os autores observaram também que o tempo médio de resposta é reduzido com o uso de índices no caso do MongoDB.

Figueiredo et al. (2022) apresentaram uma comparação entre o PostgreSQL, o MongoDB e o HarperSB utilizando algumas consultas disponibilizadas pelo TPC-H. Os autores focaram nas operações de inserção de dados, demonstrando que o HarperDB seria

interessante se necessário trabalhar com NoSQL e SQL em uma em uma única solução de banco de dados. No entanto, nenhuma estrutura de índices ou *embedding* foi utilizada.

Costa et al. (2022) também compararam o PostgreSQL e o MongoDB utilizando algumas consultas disponibilizadas pelo TPC-H. Os autores também focaram em aspectos de inserção de dados, só que mostrando como a encriptação de dados impacta o desempenho dos sistemas. Eles buscavam entender se era mais vantajoso usar a criptografia no lado do cliente (PostgreSQL) ou no Rest (MongoDB). Segundo os autores, o uso da criptografia no Rest melhorou o desempenho das inserções no banco de dados.

Khan et al. (2023) apresentaram uma revisão sobre bancos de dados SQL e NoSQL no contexto de nuvens computacionais. Os autores avaliaram questões de portabilidade e interoperabilidade de tais sistemas, e envidaram esforços em comparações entre o MongoDB e o SGBD relacional Oracle. Segundo os autores, os bancos de dados NoSQL, com suas estruturas especificamente adaptadas, podem ser a melhor opção para a análise de Big Data, enquanto os bancos de dados SQL são mais adequados para fins de processamento de transações *on-line*.

Os artigos descritos nos apoiaram na condução do presente trabalho, de modo a um entendimento mais abrangente do estado da arte. Nosso trabalho se diferencia por oferecer experimentos com versões atualizadas dos SGBDs e principalmente pelo uso do *benchmark* TPC-H com todas as suas 22 consultas, além de uma comparação com uma versão do PostgreSQL com índices adicionados ao modelo de dados.

3. Metodologia

A metodologia adotada neste estudo envolveu a aplicação do *Transaction Processing Performance Council Benchmark H* (TPC-H), um *benchmark* amplamente reconhecido e utilizado para avaliar o desempenho de SGBDs em cargas de trabalho de análise de decisões (TPC, 2022). Ele simula um ambiente de *data warehouse* com uma série de consultas complexas que representam cenários reais de negócio, sendo composto por um conjunto de 22 consultas SQL que abrangem uma ampla gama de operações analíticas. Estas consultas são projetadas para testar a eficiência dos SGBDs em executar tarefas como seleção, junção, agregação e ordenação.

O esquema de banco de dados utilizado pelo TPC-H é padronizado. Ele consiste em oito tabelas principais, que representam entidades comuns em ambientes de negócios, como clientes, pedidos, itens e fornecedores. Tais tabelas são inter-relacionadas por meio de chaves estrangeiras, o que permite a execução de consultas complexas que refletem operações comuns em sistemas de suporte à decisão. Para a aplicação do TPC-H, desenvolveu-se uma série de etapas para a configuração, migração e testes dos SGBDs MongoDB e PostgreSQL. A Figura 1 resume essas etapas.

Nas etapas de trabalho do PostgreSQL, representada pela cor azul, houve a necessidade de ajuste das consultas. Verificou-se que algumas não estavam com todos os parâmetros preenchidos. Para resolver esse problema, utilizou-se a especificação do TPC-H. A documentação do *benchmark* disponibiliza para cada uma das 22 consultas a forma de validação dos resultados das consultas e os parâmetros a serem utilizados para ajustar tais consultas (TPC, 2022).

Nas etapas de trabalho do MongoDB, representada pela cor verde, utilizou-se a

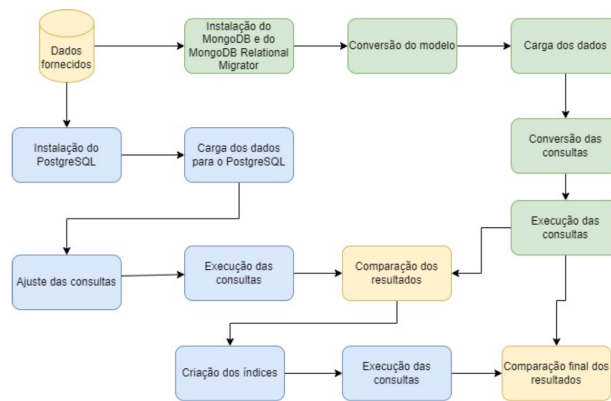


Figura 1. Etapas para comparação de desempenho de consultas entre MongoDB e PostgreSQL utilizando TPC-H

ferramenta *MongoDB Relational Migrator*⁴ para realizar o processo de migração dos dados do PostgreSQL para o MongoDB. Esta ferramenta permitiu a criação do modelo de dados *embedding documents*, no qual os dados relacionados foram incorporados em documentos únicos, conforme recomendado pela abordagem *document-oriented* do MongoDB. Após a migração dos dados, realizou-se uma série de testes em diferentes modelagens da *collection* no MongoDB. Isso incluiu a avaliação de diferentes abordagens de modelagem, como a utilização do documento principal “*orders*” e “*lineitems*” (Figura 2) em testes separados, bem como a adoção do modo misto recomendado pela ferramenta, em que algumas tabelas foram absorvidas por outras, resultando na criação de várias *collections*. Optou-se pelo modelo, na qual a *collection* “*orders*” (tabela *Orders* no modelo de dados do TPC-H) foi definida como documento principal.

A ferramenta *MongoDB Relational Migrator* também oferece a conversão de consultas entre os bancos de dados relacional e NoSQL. No entanto, observou-se que a migração completa das consultas não foi totalmente realizada. Novamente, necessitou-se de um processo manual de verificação e mudança dos comandos, a fim de que as consultas produzissem o resultado esperado.

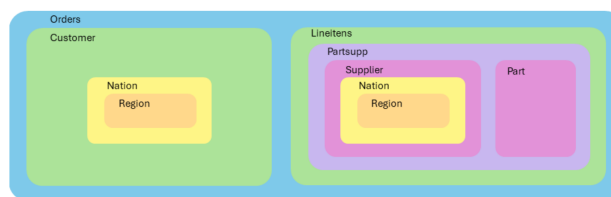


Figura 2. Modelo Adotado do MongoDB

Após as etapas para organização do ambiente, as consultas foram realizadas e os resultados comparados (representados na Figura 1 pela cor laranja).

4. Avaliação Experimental

Os SGBDs foram instalados em uma máquina com processador Intel Core i7-7500U, 16Gb de memória RAM, Windows 10. Para o desenvolvimento de programas para

⁴<https://www.mongodb.com/products/tools/relational-migrator>

execução das consultas nos SGBDs, foi utilizada a linguagem de programação Python 3.11. Os testes foram executados na mesma máquina de instalação dos SGBDs, evitando atrasos de transmissão de dados em redes. Todos os programas desenvolvidos, consultas convertidas e resultados completos estão disponibilizados no repositório público⁵. Em ambos os SGBDs, foram conduzidas dez sessões de execução das consultas. Cada sessão consistiu de uma execução de cada uma das 22 consultas do TPC-H. Os tempos resultantes foram registrados em arquivo em meio permanente para análise subsequente.

As consultas do *benchmark* TPC-H possuem as seguintes características: Q1: agregação; Q2: subconsultas e junções complexas; Q3: filtro e agregação; Q4: filtragem; Q5: junções entre múltiplas tabelas; Q6: consultas e agregações condicionais; Q7: junções e filtros condicionais; Q8: junções complexas e agregações condicionais; Q9: operações aritméticas e agregações; Q10: junções entre tabelas; Q11: operações de filtragem; Q12: filtragem; Q13: contagem de registros; Q14: agregações condicionais; Q15: agregação e ordenação; Q16: junções complexas e subconsultas; Q17: filtro e agregação; Q18: junções e filtragem de registros de alto volume; Q19: filtragem e agregação com múltiplos critérios; Q20: subconsultas e junções condicionais; Q21: junções complexas e filtragem; e Q22: filtragem e contagem.

No primeiro teste realizado, verificou-se que, devido à consulta 20 do TPC-H, o tempo médio de execuções de consultas do PostgreSQL foi muito superior ao MongoDB (10,5 seg. vs. menos de 2 seg., respectivamente). De 22 consultas, o PostgreSQL foi mais rápido em 18 delas; porém, a diferença na consulta 20 foi tão grande que no tempo total o MongoDB apresentou-se melhor. Por este motivo, decidiu-se criar índices para tentar melhorar o desempenho da consulta 20. Para não favorecer o PostgreSQL, optou-se por criar índices apenas nas chaves estrangeiras. Ao término dos testes, foram calculadas a média e o desvio-padrão do tempo de execução de cada consulta.

5. Resultados

A Figura 3 apresenta uma visão geral do desempenho de todas as consultas avaliadas em relação aos diferentes SGBDs testados. Cada grupo de barras representa uma consulta específica, enquanto as diferentes cores indicam os testes com o MongoDB, PostgreSQL sem índice e PostgreSQL com índice.

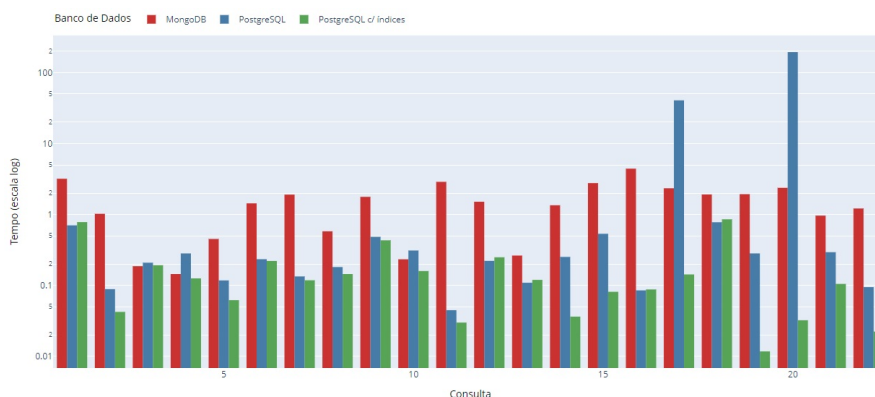


Figura 3. Média dos resultados obtidos em cada consulta por banco de dados.

⁵https://github.com/jnpavao/tpch_mongodb

A Tabela 1 apresenta o tempo gasto (em segundos) de cada uma das 22 consultas (*queries*) para MongoDB, PostgreSQL e PostgreSQL com criação de índices. A Tabela informa também a média e o desvio padrão do tempo de execução de cada consulta.

Tabela 1. Tempos de execução (em segundos) das consultas do TPC-H nos SGBDs MongoDB e PostgreSQL (com e sem índices)

Querie	Banco	1	2	3	4	5	6	7	8	9	10	Média	Desvio
Query1	MongoDB	3.080	3.060	3.440	3.135	3.031	3.230	3.358	3.460	3.056	3.059	3.191	0.160
Query1	PostgreSQL	0.849	0.699	0.690	0.693	0.676	0.675	0.701	0.672	0.696	0.669	0.702	0.050
Query1	PostgreSQL c/ ind.	0.818	0.710	0.811	0.743	0.728	0.757	0.796	0.841	0.791	0.809	0.780	0.041
Query2	MongoDB	1.004	1.012	1.061	0.987	1.076	0.963	1.065	1.067	0.969	1.026	1.023	0.040
Query2	PostgreSQL	0.125	0.087	0.087	0.094	0.076	0.080	0.078	0.089	0.087	0.083	0.089	0.013
Query2	PostgreSQL c/ ind.	0.047	0.039	0.043	0.038	0.038	0.039	0.042	0.043	0.051	0.044	0.042	0.004
Query3	MongoDB	0.188	0.193	0.180	0.185	0.183	0.222	0.180	0.177	0.175	0.184	0.187	0.013
Query3	PostgreSQL	0.315	0.209	0.210	0.189	0.198	0.199	0.203	0.194	0.180	0.191	0.209	0.036
Query3	PostgreSQL c/ ind.	0.244	0.176	0.168	0.195	0.180	0.191	0.194	0.182	0.191	0.207	0.193	0.020
Query4	MongoDB	0.138	0.138	0.143	0.150	0.152	0.158	0.140	0.141	0.143	0.147	0.145	0.006
Query4	PostgreSQL	0.264	0.290	0.289	0.284	0.291	0.277	0.278	0.282	0.288	0.282	0.282	0.008
Query4	PostgreSQL c/ ind.	0.100	0.107	0.110	0.180	0.111	0.142	0.137	0.116	0.136	0.118	0.126	0.022
Query5	MongoDB	0.456	0.482	0.464	0.444	0.452	0.458	0.454	0.443	0.436	0.433	0.452	0.014
Query5	PostgreSQL	0.101	0.128	0.126	0.115	0.129	0.112	0.128	0.112	0.119	0.109	0.118	0.009
Query5	PostgreSQL c/ ind.	0.072	0.057	0.059	0.058	0.059	0.061	0.060	0.065	0.064	0.066	0.062	0.004
Query6	MongoDB	1.425	1.411	1.432	1.467	1.416	1.503	1.426	1.417	1.417	1.449	1.436	0.027
Query6	PostgreSQL	0.224	0.222	0.235	0.246	0.222	0.270	0.227	0.227	0.240	0.225	0.234	0.014
Query6	PostgreSQL c/ ind.	0.219	0.196	0.234	0.242	0.199	0.215	0.245	0.229	0.219	0.219	0.221	0.016
Query7	MongoDB	1.920	1.856	2.005	1.989	1.876	1.865	1.919	1.871	1.900	1.875	1.908	0.049
Query7	PostgreSQL	0.118	0.138	0.151	0.145	0.139	0.125	0.136	0.140	0.123	0.128	0.134	0.010
Query7	PostgreSQL c/ ind.	0.134	0.112	0.115	0.115	0.111	0.115	0.115	0.127	0.122	0.118	0.118	0.007
Query8	MongoDB	0.581	0.562	0.571	0.600	0.577	0.559	0.623	0.562	0.581	0.577	0.580	0.019
Query8	PostgreSQL	0.199	0.175	0.205	0.171	0.176	0.191	0.176	0.185	0.173	0.166	0.182	0.013
Query8	PostgreSQL c/ ind.	0.155	0.141	0.139	0.144	0.140	0.144	0.145	0.146	0.146	0.148	0.145	0.004
Query9	MongoDB	1.875	1.731	1.701	1.773	1.832	1.821	1.703	1.744	1.826	1.722	1.773	0.058
Query9	PostgreSQL	0.451	0.504	0.479	0.497	0.489	0.489	0.476	0.481	0.495	0.482	0.484	0.014
Query9	PostgreSQL c/ ind.	0.452	0.397	0.414	0.465	0.419	0.401	0.455	0.434	0.433	0.454	0.432	0.023
Query10	MongoDB	0.240	0.234	0.228	0.234	0.250	0.224	0.223	0.225	0.228	0.245	0.233	0.009
Query10	PostgreSQL	0.330	0.311	0.315	0.309	0.311	0.294	0.303	0.310	0.293	0.330	0.311	0.012
Query10	PostgreSQL c/ ind.	0.167	0.149	0.145	0.151	0.151	0.164	0.161	0.160	0.180	0.167	0.160	0.010
Query11	MongoDB	2.805	2.840	2.943	2.903	2.954	2.894	2.800	2.940	2.838	2.943	2.886	0.057
Query11	PostgreSQL	0.044	0.045	0.051	0.047	0.045	0.044	0.043	0.042	0.042	0.046	0.045	0.003
Query11	PostgreSQL c/ ind.	0.028	0.030	0.026	0.030	0.031	0.032	0.030	0.030	0.030	0.032	0.030	0.002
Query12	MongoDB	1.485	1.494	1.489	1.502	1.548	1.441	1.539	1.634	1.455	1.532	1.512	0.052
Query12	PostgreSQL	0.211	0.199	0.227	0.226	0.234	0.228	0.214	0.218	0.226	0.232	0.222	0.010
Query12	PostgreSQL c/ ind.	0.250	0.288	0.240	0.251	0.235	0.208	0.243	0.251	0.262	0.264	0.249	0.020
Query13	MongoDB	0.266	0.258	0.254	0.257	0.256	0.265	0.279	0.266	0.279	0.255	0.264	0.009
Query13	PostgreSQL	0.156	0.102	0.108	0.103	0.108	0.101	0.101	0.102	0.111	0.101	0.109	0.016
Query13	PostgreSQL c/ ind.	0.114	0.148	0.112	0.106	0.112	0.133	0.129	0.111	0.112	0.121	0.120	0.012
Query14	MongoDB	1.321	1.320	1.325	1.322	1.341	1.323	1.498	1.336	1.389	1.317	1.349	0.054
Query14	PostgreSQL	0.257	0.254	0.247	0.244	0.251	0.259	0.252	0.260	0.250	0.245	0.252	0.005
Query14	PostgreSQL c/ ind.	0.038	0.036	0.036	0.036	0.034	0.035	0.039	0.035	0.038	0.037	0.036	0.002
Query15	MongoDB	2.859	2.697	2.848	2.731	2.792	2.771	2.764	2.720	2.756	2.766	2.770	0.049
Query15	PostgreSQL	0.517	0.558	0.531	0.569	0.575	0.506	0.556	0.508	0.507	0.521	0.535	0.026
Query15	PostgreSQL c/ ind.	0.100	0.076	0.079	0.074	0.083	0.078	0.080	0.080	0.083	0.080	0.081	0.007
Query16	MongoDB	4.653	4.277	4.323	4.287	4.267	4.326	5.062	4.254	4.258	4.571	4.428	0.250
Query16	PostgreSQL	0.097	0.083	0.087	0.083	0.082	0.084	0.086	0.084	0.082	0.084	0.085	0.004
Query16	PostgreSQL c/ ind.	0.090	0.087	0.083	0.094	0.085	0.087	0.089	0.088	0.088	0.088	0.088	0.003
Query17	MongoDB	2.412	2.382	2.275	2.303	2.299	2.337	2.329	2.365	2.341	2.383	2.342	0.041
Query17	PostgreSQL	40.460	40.245	40.608	40.730	40.281	41.479	40.625	40.557	40.714	40.930	40.663	0.335
Query17	PostgreSQL c/ ind.	0.137	0.133	0.134	0.137	0.138	0.134	0.134	0.200	0.144	0.138	0.143	0.019
Query18	MongoDB	1.968	1.849	1.950	1.852	1.922	1.864	1.962	1.841	1.966	1.991	1.916	0.056
Query18	PostgreSQL	0.748	0.823	0.774	0.769	0.766	0.781	0.784	0.757	0.744	0.826	0.777	0.027
Query18	PostgreSQL c/ ind.	0.803	0.947	0.846	0.790	0.807	0.789	0.875	0.892	0.919	0.884	0.855	0.054
Query19	MongoDB	1.922	2.022	1.932	1.990	1.909	1.890	1.985	1.863	1.935	1.897	1.934	0.048
Query19	PostgreSQL	0.293	0.287	0.274	0.278	0.276	0.278	0.278	0.293	0.289	0.283	0.283	0.007
Query19	PostgreSQL c/ ind.	0.015	0.009	0.011	0.013	0.010	0.013	0.013	0.011	0.011	0.011	0.012	0.002
Query20	MongoDB	2.358	2.457	2.349	2.335	2.441	2.331	2.367	2.325	2.324	2.496	2.378	0.060
Query20	PostgreSQL	197.585	201.893	195.814	195.147	198.517	188.337	191.336	192.878	194.742	191.352	194.760	3.773
Query20	PostgreSQL c/ ind.	0.034	0.032	0.027	0.028	0.030	0.035	0.032	0.031	0.034	0.041	0.032	0.004
Query21	MongoDB	0.928	0.997	1.002	0.954	0.943	0.932	0.991	0.977	0.945	0.972	0.964	0.026
Query21	PostgreSQL	0.308	0.303	0.287	0.301	0.283	0.294	0.295	0.309	0.280	0.281	0.294	0.010
Query21	PostgreSQL c/ ind.	0.095	0.163	0.095	0.098	0.097	0.095	0.101	0.101	0.103	0.104	0.105	0.019
Query22	MongoDB	1.260	1.247	1.217	1.190	1.185	1.208	1.238	1.197	1.204	1.222	1.217	0.024
Query22	PostgreSQL	0.103	0.091	0.090	0.091	0.086	0.085	0.131	0.090	0.091	0.088	0.095	0.013
Query22	PostgreSQL c/ ind.	0.026	0.023	0.022	0.022	0.020	0.020	0.023	0.025	0.021	0.022	0.022	0.002

Observa-se que nas consultas 3, 4 e 10, o MongoDB apresenta desempenho melhor se comparado com a versão do PostgreSQL sem índices. Embora isso também aconteça nas consultas 17 e 20, nestas é notável a discrepância do MongoDB em relação ao PostgreSQL sem a aplicação do índice. Em contrapartida, nas consultas Q1, Q12, Q13, Q16 e Q18, a aplicação do índice no PostgreSQL não resultou em melhores desempenhos.

6. Considerações Finais

Este estudo apresentou uma análise comparativa entre os sistemas de gerenciamento de banco de dados MongoDB e PostgreSQL, com foco em suas capacidades ao lidar com

consultas complexas e cargas de trabalho de análise de dados, utilizando o *benchmark* TPC-H. A metodologia aplicada permitiu uma avaliação detalhada do desempenho de ambos os sistemas, considerando a eficiência na recuperação de dados complexos e a robustez no processamento de consultas relacionais.

As contribuições deste trabalho para a área de *e-Science* são relevantes, pois fornecem uma base sólida para a escolha do sistema de banco de dados mais adequado para diferentes tipos de aplicações científicas. A análise mostrou que, embora o MongoDB tenha vantagens em termos de flexibilidade e desempenho em consultas específicas, o PostgreSQL provou ser a solução mais eficiente em cenários que exigem consultas relacionais complexas e otimizações com índices.

Como trabalhos futuros, propomos a exploração de outros *benchmarks*, como o TPC-DS, para avaliar o desempenho dos sistemas em cargas de trabalho de análise de dados mais diversificadas. Além disso, pretendemos investigar o consumo de memória dos SGBDs e implementar estudos de caso em ambientes de produção, a fim de observar o comportamento dos sistemas em situações reais. Finalmente, sugerimos a extensão das comparações para incluir outros modelos de SGBDs NoSQL, como chave-valor, orientado a grafos, e colunas largas, ampliando assim o escopo das aplicações em *e-Science*.

Referências

- Costa, M. et al. (2022). Database encryption performance impact on postgresql and mongodb. In *Marketing and Smart Technologies: Proceedings of ICMarTech 2021, Volume 1*, pages 121–127. Springer.
- Figueiredo, D. et al. (2022). Performance evaluation between harperdb, mongo db and postgresql. In *International Conference on Marketing and Technologies*, pages 85–94. Springer.
- Filip, P. and Čegan, L. (2020). Comparison of mysql and mongodb with focus on performance. In *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, pages 184–187. IEEE.
- Győrödi, C. et al. (2015). A comparative study: Mongodb vs. mysql. In *13th International Conference on Engineering of Modern Electric Systems (EMES)*, pages 1–6.
- Kanade, A. et al. (2014). A study of normalization and embedding in mongodb. In *2014 IEEE International Advance Computing Conference (IACC)*, pages 416–421.
- Khan, W. et al. (2023). Sql and nosql database software architecture performance analysis and assessments-a systematic literature review. *Big Data and Cognitive Computing*.
- Llano-Ríos, T. F. et al. (2020). Evaluating nosql systems for decision support: An experimental approach. In *IEEE International Conference on Big Data (Big Data)*, pages 2802–2811.
- Makris, A. et al. (2021). Mongodb vs postgresql: A comparative study on performance aspects. *GeoInformatica*, 25:243–268.
- TPC (2022). Tpc benchmarkm h (decision support) standard specification revision 3.0.1. Acesso em: 27 de julho de 2024.
- Zhao, G. et al. (2013). Modeling mongodb with relational model. In *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*, pages 115–121.