Desempenho de SGBDs NoSQL no armazenamento de mídias digitais

Performance Evaluation of NoSQL data store for digital media

Jonathan de Oliveira Assis, Vanessa C O Souza, Melise M V Paula, João Bosco S Cunha Instituto de Matemática e Computação Universidade Federal de Itajubá (UNIFEI) Itajubá, Brasil

jonathan-o.a@hotmail.com, {vanessasouza, melise, bosco}@unifei.edu.br

Resumo — Com a utilização cada vez maior das redes sociais, o armazenamento e recuperação de vídeos e imagens tem sido objeto de atenção quando o assunto é desempenho. Este trabalho tem como objetivo avaliar o desempenho dos bancos NoSQL em operações envolvendo mídias digitais e, posteriormente, na simulação de uma rede social, comparando os resultados com um banco relacional. Os resultados encontrados sugerem que os bancos NoSQL podem ser mais eficientes que os bancos relacionais em aplicações que envolvem mídias digitais.

Palavras Chave - NoSQL; mídias digitais; desempenho.

Abstract — With the increasing use of social networks, the storage and retrieval of videos and images has been the object of attention when it comes to performance. This study aims to evaluate the performance of NoSQL databases in transactions involving digital media and in the simulation of a social network, comparing results with a relational database. The results showing that the NoSQL databases are more efficient than relational databases in applications involving digital media.

Keywords - NoSQL; digital media; performance.

I. INTRODUÇÃO

Atualmente, os bancos de dados relacionais ainda têm sido a opção mais provável quando o assunto é a persistência de dados. A utilização de tabelas normalizadas pode proporcionar um bom desempenho para as aplicações com baixo volume de dados. No entanto, com o aumento na quantidade, variedade e velocidade das informações armazenadas nas aplicações, foi possível identificar algumas limitações do modelo relacional, especialmente com relação à escalabilidade. Neste cenário, os modelos de banco de dados NoSQL (*Not Only SQL*) surgiram como alternativa.

Outro aspecto que caracteriza o cenário atual dos dados é a forma como os ambientes virtuais vêm sendo utilizados. Nas redes sociais, por exemplo, a disseminação de vídeos, imagens e músicas tem se tornado mais frequente. Desta forma, faz-se necessário investigar alternativas que resultem em soluções otimizadas para a persistência deste tipo de dado considerando os mecanismos disponíveis.

O tema justifica-se, pois os modelos relacionais podem apresentar dificuldades para persistir mídias digitais. Nesses SGBDs, o BLOB (*Binary Large Object*) é o tipo de dado mais comumente utilizado para o armazenamento de mídias. Além disso, o modelo de dados relacional precisa ser estendido para gerenciar com eficiência dados multimídia [11].

O objetivo desse trabalho é explorar o armazenamento desses dados complexos e não estruturados pelos modelos de bancos de dados não relacionais. O trabalho foi executado em duas etapas. Na primeira etapa, o objetivo foi investigar o comportamento de três SGBDs não relacionais na manipulação de dados complexos. Para esta análise, foi considerado o framework YCSB (Yahoo! CloudServing Benchmark), que é uma ferramenta capaz de realizar benchmarks em diferentes sistemas gerenciadores de bancos de dados [5].

A partir desses resultados, na segunda etapa, foi selecionado um SGBD não relacional que foi comparado a um SGBD relacional na simulação de uma rede social. A escolha de uma rede social para esta análise se justifica por ser este um dos tipos de aplicações que manipulam um volume considerável de mídias digitais atualmente. Os resultados sugerem que os bancos de dados NoSQL podem ser mais eficientes que os bancos relacionais quando o assunto é manipulação de mídias digitais.

O artigo está organizado da seguinte maneira: Na seção II, são apresentados alguns trabalhos relacionados considerados como base teórica para esta análise. A seção III resume a metodologia adotada para a execução dos testes. Na seção IV, são apresentados os resultados e as análises realizadas. Já a seção V apresenta as conclusões e considerações finais.

II. TRABALHOS RELACIONADOS

Ao pesquisar sobre trabalhos relacionados, foram identificados diversos artigos que realizam a comparação de desempenho entre os bancos NoSQL, como em [7, 9, 10]. No entanto, não foram identificados trabalhos que tratassem de operações envolvendo imagens ou vídeos e comparasse as diferentes formas de armazenamento destes elementos, com exceção de [14], que comparou o uso do MongoDB e do MySQL no armazenamento de imagens médicas.

Em [10], os autores apresentaram um estudo com dois objetivos principais: verificar o amadurecimento e consolidação dos SGBDs NoSQL e testar a performance e a escalabilidade dos mesmos. O desempenho foi medido utilizando a configuração padrão do YCSB e comparando os resultados dos bancos Redis e Cassandra (Não relacionais) com o MySQL (Relacional), variando as cargas de operações em um espaço de tempo. Como resultado, o Redis foi melhor nos testes de desempenho. Além disso, os autores afirmaram que o Cassandra apresentou uma maior consolidação, tanto no mercado, quanto na área acadêmica, quando comparado ao Redis.

No trabalho de [9], são descritas as vantagens na utilização de um banco de dados NoSQL, além de uma análise de desempenho do MongoDB, comparando o tempo de inserção e de pesquisa para 500 (quinhentos) mil registros com o banco relacional MySQL. O estudo conclui que o SGBD não relacional é bem mais rápido em ambas as operações.

Em [7], é apresentado um estudo comparativo entre um banco de dados relacional (MySQL) e um banco de dados não relacional (MongoDB) utilizando funções em PHP para calcular os tempos de inserção/leitura/exclusão/atualização de cada banco, simulando uma aplicação de um fórum de discussão online. Como resultado, o MongoDB se mostrou mais eficiente na maioria das comparações.

Em [14], o desempenho de armazenamento e recuperação de imagens médicas com diferentes tamanhos foram avaliados no MySQL e no MongoDB. Esse estudo mostrou que o MongoDB obteve melhor performance que o MySQL, mesmo com o aumento do tamanho das imagens.

III. METODOLOGIA

Na primeira etapa do estudo, foram realizados testes *standalone* com os SGBDs: Redis (Chave-Valor), Cassandra (*bigtable*) e MongoDB (Orientado a Documentos). Para a definição do *benchmarking*, foi utilizada a ferramenta YCSB, que precisou ser customizada para possibilitar os testes com dados complexos, já que por padrão este framework trabalha apenas com números e textos gerados automaticamente. Detalhes das modificações feitas podem ser vistos em [1].

Para compor o conjunto de amostras, foram obtidas cem imagens e três vídeos, ambos com tamanhos variados que foram selecionados aleatoriamente no site LoremPixel [12]. Além disso, foi utilizada uma máquina dedicada com as seguintes configurações: Sistema Operacional Debian 8.2.0, Processador Intel Core 2 Quad, 4 GB de RAM e 250 GB de HD.

As métricas avaliadas foram latência e vazão. As cem imagens e os três vídeos recuperados foram transformados em hexadecimal e armazenados em arquivos de texto, para que fosse possível a manipulação dos mesmos pela ferramenta YCSB.

Após a finalização dos testes da primeira etapa, foi selecionado um SGBD NoSQL para a segunda etapa que teve seu desempenho comparado ao de um banco relacional em uma aplicação web, desenvolvida em PHP, que simulou o armazenamento e leitura de mídias digitais em uma rede social, utilizando o mesmo conjunto de amostras da primeira etapa.

IV. RESULTADOS

A. Etapa 1: Testes de Performance

Para a realização do *benchmarking* com o YCSB, foram consideradas duas cargas de trabalho pré-configuradas na ferramenta: Carga A (50% leitura e 50% escrita) e Carga B (95% leitura e 5% escrita), também chamadas de Workload A e B. Os testes mediram a latência e a vazão para cinco, sete, doze, quinze e vinte mil operações.

As Fig. 1 e 2 apresentam o fluxo de dados (vazão) para ambas as cargas. Observa-se que o Redis possui um fluxo maior, ou seja, consegue realizar um número maior de operações por segundo se comparado aos outros dois bancos.

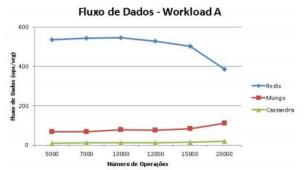


Figura 1. Fluxo de Dados (Vazão) - Carga A

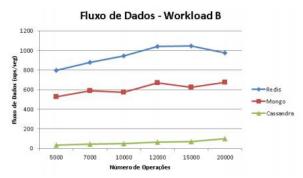


Figura 2. Fluxo de Dados (Vazão) – Carga B

As Fig. 3 e 4 sumarizam os resultados de latência para a carga de trabalho A, enquanto as Fig. 5 e 6 sumarizam os resultados para a carga B.

Com relação à latência, os resultados foram similares ao de vazão. A performance do Redis foi melhor, seguida pelo MongoDB e o Cassandra. Os resultados corroboram com [10], no qual o SGBD Cassandra apresentou um fluxo de operações por segundo bem menor que o Redis, utilizando dados numéricos. Segundo [10], o bom desempenho do Redis devese a seu modelo de dados e estrutura, que utiliza tabela hash e funciona em memória.

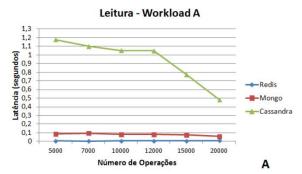


Figura 3. Latência de Leitura - Carga A



Figura 4. Latência de Escrita - Carga A



Figura 5. Latência de Leitura - Carga B

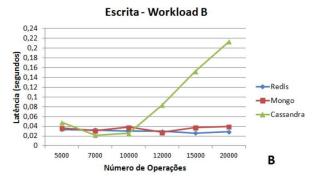


Figura 6. Latência de Escrita - Carga B

Outro ponto a se destacar é a diferença entre a leitura e escrita no Cassandra. A escrita foi cerca de 58% (cinquenta e oito por cento) mais rápida do que a leitura na carga A e cerca de 50% (cinquenta por cento) na carga B. De acordo com [3] e [5], a justificativa pode ser dada pela escrita otimizada realizada pelo SGBD. Esta otimização se dá tanto pelo uso de algoritmos de compressão, quanto pelo uso de cache/log para operações de escrita [13, 15]. [16] faz uma revisão detalhada sobre o processo de escrita e leitura do Cassandra. [15] classifica a leitura do Cassandra como 'medíocre' e a escrita como 'ótima'.

Nos testes realizados, observou-se também que na carga de leitura, o Cassandra tende a melhorar a latência com o aumento do número de operações. Já na escrita, esse comportamento se inverte.

Nos estudos feitos por [2] e [3], tanto para a carga A, quanto para a carga B, o Cassandra possui vantagem sobre o MongoDB. Nesses estudos, a latência é medida em relação ao número de nós da rede, ao invés do número de operações. Desta forma, pode-se concluir que em um ambiente distribuído, a constância na escalabilidade do Cassandra supera a do Mongo, obtendo um desempenho melhor.

Por outro lado, nos testes *stand-alone* realizados nesse estudo, o Mongo apresentou um desempenho um pouco melhor que o Cassandra, especialmente na escrita da Carga A. Esse resultado pode ser devido às características do Mongo, que é otimizado para leitura [3]. [15] explicam que os bancos orientados a documentos, como o Mongo, herdaram dos SGBDs relacionais a estratégia de serem mais lentos na escrita para prover melhor performance de recuperação, e que, quando comparados com os orientados a coluna, apresentam melhor performance de leitura.

Outro destaque importante dos resultados é a escrita da Carga B, onde os SGBDs analisados obtiveram resultados parecidos até dez mil operações. A partir de doze mil operações, a latência do Cassandra aumenta, enquanto a do Redis e do MongoDB mantêm-se na mesma faixa de valores. A carga B tem apenas 5% de escrita, o que permite concluir que, nessas condições, a performance dos SGBDs analisados é bastante semelhante.

B. Etapa 2: Testes na Aplicação Web

Nesta seção são apresentados os resultados dos testes de desempenho com arquivos multimídia realizados na aplicação web que simula uma rede social. Foram implementadas duas funcionalidades: a inserção de um arquivo (mídia) associado a um usuário, e a recuperação de todas as mídias pertencentes a um determinado usuário (escolhido aleatoriamente). Para garantir que a cada teste de inserção todos os usuários tivessem o mesmo número de imagens, uma função de controle foi implementada. Apenas a latência foi analisada nesta etapa.

Apesar de ter se sobressaído nos testes, o Redis não foi o banco de dados NoSQL escolhido para ter seu desempenho analisado em uma aplicação web. Segundo [6], bancos de chave-valor não são indicados para consulta de dados, já que a busca só pode ser realizada pela chave do par chave-valor, além de não serem eficientes no relacionamento entre dados. No caso de uma rede social, tanto a busca pelo conteúdo dos registros quanto o relacionamento entre dados devem existir. Por isso, o banco não relacional escolhido foi o MongoDB que mostrou melhor performance do que o Cassandra e que modela bem relacionamentos.

O MongoDB foi comparado ao MySQL em duas abordagens diferentes que variaram em relação à forma de armazenamento: na primeira, o conteúdo da imagem foi gravado no banco. Na segunda, apenas o *link* da imagem foi gravado. As latências de leitura e escrita foram medidas para 20 (vinte), 40 (quarenta), 60 (sessenta) e 80 (oitenta) *threads*, onde cada uma executava a aplicação de forma independente, simulando conexões simultâneas de 20 (vinte) usuários diferentes. Os valores apresentados se referem às médias das latências obtidas na realização de 10 (dez) testes, sendo que cada teste retornou a latência média de todas as *threads* que foram executadas.

Em um primeiro momento, para a primeira abordagem, foram consideradas duas modelagens diferentes no MongoDB. Na primeira modelagem, cada imagem foi inserida em um documento diferente, contendo o identificador do usuário e as informações da imagem. Na segunda modelagem, ao invés de um documento por imagem, foi criado um único documento que incorporava todas as imagens de um usuário em um vetor de imagens. Assim, a cada nova inserção, o vetor contendo as imagens recebia um novo elemento. Detalhes dos modelos implementados podem ser vistos em [1].

Considerando a gravação da imagem diretamente no banco, a Modelagem 2 (*array* de mídias) obteve uma pequena vantagem na latência de leitura, conforme mostra a Fig. 7. Já nas operações de escrita, a Modelagem 1 (um documento por mídia) foi cerca de 15 vezes mais rápida do que a modelagem 2, conforme a Fig. 8. Como os resultados de leitura foram bem semelhantes e os resultados de escrita com a Modelagem 1 tiveram uma diferença significativa, essa foi a escolhida para ser utilizada nos testes de comparação com o MySQL.

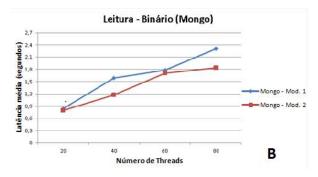


Figura 7. Latência de leitura - Tipos de modelagem

As Fig. 9 e 10 apresentam as latências médias de escrita e leitura em relação à quantidade de *threads* executadas. O MongoDB obteve latência média cerca de 9 vezes menor do que o MySQL na leitura e 19 vezes menor na escrita, mantendo-se muito mais estável com a variação de *threads*. Os resultados corroboram com os de [14], que mostram a grande vantagem do SGBD não relacional tanto na recuperação quanto no armazenamento de imagens binárias.

Considerando a segunda abordagem, o armazenamento apenas do *link* da mídia digital no banco de dados, o MySQL obteve melhor desempenho de leitura do que o MongoDB, com latência cerca de 70% (setenta por cento) menor que seu concorrente, como mostra a Fig. 11. No entanto, a latência para ambos os SGBDs é muito baixa, não permitindo concluir sobre a melhor alternativa para se utilizar nesse caso.



Figura 8. Latência de escrita - Tipos de modelagem

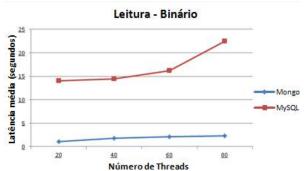


Figura 9. Latência de leitura – Armazenamento binário

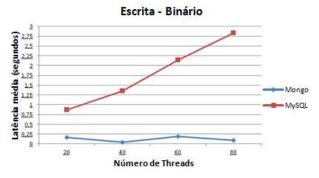


Figura 10. Latência de escrita - Armazenamento binário

Já para a escrita, o MongoDB obteve uma latência cerca de duas vezes menor que o MySQL, conforme apresentado na Fig. 12.

A diferença no tempo médio de latência tanto de escrita quanto de leitura entre as duas abordagens (mídia e *link*) é extremamente significativa. Nos resultados, fica evidente que ler e escrever apenas o *link* da imagem é muito mais rápido se comparado a seu arquivo binário, o que já era esperado, já que é um tipo de dado simples e relativamente pequeno. Por outro lado, o tempo final da aplicação precisa levar em conta a recuperação da mídia no disco.

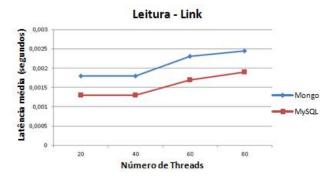


Figura 11. Latência de leitura – Armazenamento link

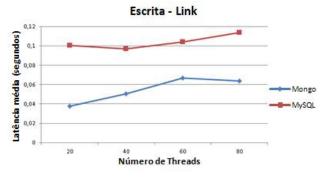


Figura 12. Latência de escrita - Armazenamento link

Na maioria dos testes, o MongoDB obteve grande vantagem em relação ao MySQL. Na leitura do binário, os resultados podem ser considerados um indicativo de que o uso do SGBD relacional pode ser impraticável em aplicações reais. Estes resultados corroboram com os de [7] e [9], que apesar de terem sido realizados com dados estruturados, também mostram a grande vantagem do MongoDB sobre o MySQL. Isso pode ser um indicativo de que os bancos não relacionais lidam melhor com arquivos de estruturas diferentes – imagens, vídeos – do que o banco relacional.

V. Conclusões

Com base nos resultados e metodologia utilizados, pode-se concluir que os bancos não relacionais podem ser mais eficientes quando é necessário armazenar os arquivos de mídias digitais diretamente no banco. Quando se trata de armazenar apenas o *link* do arquivo - que redirecionam para o arquivo armazenado em um servidor - , os bancos relacionais apresentam boa performance, sendo até mais rápidos na leitura do que o banco NoSQL. Além disso, as performances dos bancos NoSQL para mídias digitais apresentam as mesmas características que aquelas encontradas na literatura especializada da área para outros tipos de dados, na vazão e latência das cargas A e B do YCSB e para testes *stand-alone*.

Os resultados desse trabalho também vão de encontro com a indagação de [4], que em seu livro levanta a questão de '*Por que criar aplicações novas com conceitos antigos?*'. Atualmente os desenvolvedores estão habituados a armazenar no banco de dados apenas o *link* da imagem e não a imagem propriamente dita. Em bancos relacionais isso faz sentido, visto que esses SGBDs têm dificuldade em lidar com dado não estruturado e em escalar. No entanto, os testes realizados nesse trabalho indicam que os SGBDs NoSQL podem ser eficientes para tratar mídias.

Dessa forma, é necessário repensar a forma de desenvolver aplicações, tanto as especializadas, cujo foco é o armazenamento de mídias, quanto outras que eventualmente necessitem armazenar esse tipo de dado. Essa mudança de paradigma passa também pelo conceito de persistência poliglota [6], onde uma mesma aplicação usa diferentes métodos de persistência, para diferentes necessidades da aplicação.

O armazenamento de mídias digitais por meio de mecanismos de persistência pode trazer diversas vantagens, especialmente na manutenção desse ambiente, e, nesse contexto, os bancos NoSQL são uma alternativa robusta e eficaz.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Assis, J. O. (2015). "Performance evaluation of non-relational database models (NoSQL) in the storage of digital media". Monograph (bachelor in Information Systems). "Avaliação de desempenho dos modelos de bancos não relacionais (NoSQL) no armazenamento de mídias digitais". Monografía (Bacharel em Sistemas de Informação). Instituto de Matemática e Computação Universidade Federal de Itajubá (IMC/UNIFEI).
- [2] Abramova, V., Bernardino, J. NoSQL databases: MongoDB vs cassandra. In Proceedings of the International Conference on Computer Science and Software Engineering, pages 14-22. 2013. ACM. DOI=http://dx.doi.org/10.1145/2494444.2494447
- [3] Bazar, C., Iosif, C. S. "The Transition from RDBMS to NoSQL. A Comparative Analysis of Three Popular Non-Relational Solutions: Cassandra, MongoDB and Couchbase". Database Systems Journal, 5(2),

- 2014. p. 49-59.
- [4] Boaglio, F. MongoDB: Build New Applications with New Technologies. MongoDB: Construa Novas Aplicações com Novas Tecnologias. Casa do Código. 2015.
- [5] Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., Sears, R. Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM symposium on Cloud computing*, 2010. p. 143-154. ACM. DOI=http://dx.doi.org/10.1145/1807128.1807152
- [6] Fowler, M.;Sadalage, P. J. NoSQL Essential. NoSQL Essencial. São Paulo: Novatec, 2013. 220 p.
- [7] Gyorodi, C., Gyorodi, R., Pecherle, G., Olah, A. A comparative study: MongoDB vs. MySQL. 2015 13th International Conference on Engineering of Modern Electric Systems, EMES 2015, 2015. p. 0– 5.DOI= https://doi.org/10.1109/EMES.2015.7158433
- [8] Lóscio, B. F., Oliveira, H. R. D., Pontes, J. C. D. S. "NoSQL in the development of collaborative Web applications". "NoSQL no desenvolvimento de aplicações Web colaborativas". In VIII Simpósio Brasileiro de Sistemas Colaborativos. 2011. Paraty, RJ.
- [9] Nyati, S. S., Pawar, S., & Ingle, R.. "Performance evaluation of unstructured NoSQL data over distributed framework". In International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, p. 1623-1627. IEEE. DOI= http://dx.doi.org/10.1109/ICACCI.2013.6637424
- [10] Souza, V. C. O., Santos, M. V. C. (2015). "Maturing, Consolidation and Performance of NoSQL Databases-Comparative Study". "Amadurecimento, Consolidação e Performance de SGBD NoSQL -Estudo Comparativo". In Anais do XI Simpósio de Sistemas de Informação, SBSI 2015, p. 235- 238. Goiânia, GO.
- [11] Andrade, N. S. Multimedia Information Systems. Dissertation (Master in Public Administration). Sistemas de Informação Multimídia. 1998. 73 f. Dissertação (Mestrado em Administração Pública) - Fundação João Pinheiro.
- [12] LoremPixel. Disponível em: http://lorempixel.com/
- [13] Abadi, D. J., Boncz, P. A., Harizopoulos, S. Column-Oriented Database Systems. In: Proceedings of the VLB Endowment. 2009, v. 2, n. 2, p.1664-1665. DOI: https://doi.org/10.14778/1687553.1687625.
- p.1664-1665. DOI: https://doi.org/10.14778/1687553.1687625.
 [14] Rebecca, D. R., & Shanthi, I. E. "A NoSQL Solution to efficient storage and retrieval of Medical Images". *International Journal of Scientific & Engineering Research*, 7(2), February-2016.
- [15] Lourenço, J. R., Cabral, B., Carreiro, P., Vieira, M., & Bernardino, J. "Choosing the right NoSQL database for the job: a quality attribute evaluation". *Journal of Big Data*, 2(1), 18. 2015. DOI: http://doi.org/10.1186/s40537-015-0025-0
- [16] Lopes, S. F. (2016). "Study and analysis of NoSQL Cassandra's architecture". Monograph (bachelor in Information Systems). "Estudo e análise da arquitetura do SGBD NoSQL Cassandra". Monografia (Bacharel em Sistemas de Informação). Instituto de Matemática e Computação – Universidade Federal de Itajubá (IMC/UNIFEI).