Estudo Comparativo de Desempenho entre Bancos de Dados NoSQL Distribuídos

Kaique J. Costa, Kalvin V. V. Santos, Roger T. Rojas, Wendel S. Duarte Junior, Calebe P. Bianchini

FCI – Universidade Presbiteriana Mackenzie - São Paulo, Brasil

contato: calebe.bianchini@mackenzie.br

Abstract. *With the growth of NoSQL databases on the market and the great amount of data in today's digital world, the amount of data often exceeds the computing power of one machine, which makes it necessary to utilize a distributed database system. In this scenario, many IT professionals don't know which database is the best option when starting a new project. This paper has the goal of making a performance evaluation and comparison between three popular NoSQL databases (Redis, MongoDB, and Cassandra), distributing data across three distinct nodes that are subject to specific scenarios of data processing (read and write) so that the IT community can have a better understanding of those technologies.*

Resumo. *Com o crescimento dos bancos de dados NoSQL no mercado e a grande quantidade de dados no mundo digital atual, é necessário utilizar um sistema distribuídos de banco de dados que consiga ter poder computacional suficiente para processá-lo. Muitos profissionais de tecnologia, ao iniciar um novo projeto de software, não conhecem qual a melhor opção de banco escolher. Este estudo tem como objetivo comparar o desempenho de três bancos de dados NoSQL (Redis, MongoDB e Cassandra), distribuindo seus dados em três máquinas e em cenários específicos de processamento de dados (leitura e escrita), visando auxiliar visualizar melhor o comportamento dessas tecnologias.*

1. Introdução

Na era da informação, aplicações e sistemas de *software* estão cada vez mais lidando com grandes volumes de dados, tornando necessário a construção de arquiteturas de *software* robustas e escaláveis. Entre as principais necessidades que as empresas de tecnologia enfrentam, destacam-se duas: suportar um alto tráfego de informações de usuários simultaneamente utilizando um serviço e a capacidade de responder às interações dos usuários rapidamente, através de bases de dados geograficamente próximas ao cliente final [Abadi 2012].

Mediante a essas necessidades, o paradigma do NoSQL ganhou força como uma alternativa escalável ao modelo relacional de dados. O termo NoSQL foi inicialmente introduzido em 1998, por Carlo Strozzi para se referir a um banco de dados de código aberto desenvolvido por ele. Nos últimos anos, o termo se refere a uma categoria de bancos de dados que não garante as propriedades ACID (acrônimo de Atomicidade, Consistência, Isolamento e Durabilidade) e que são facilmente escaláveis de forma horizontal e distribuída. Esses bancos estruturam dados de diferentes formas, como

orientados a documentos, chave-valor e grafos [Berg, Seymour e Goel 2013].

Atualmente, há uma grande variedade de bancos NoSQL distribuídos. Cada tecnologia segue um tipo de implementação, o que favorece diferentes casos de uso e cenários de desempenhos. Por exemplo, algumas tecnologias favorecem grandes volumes de operações de escrita simultânea, enquanto outras favorecem grandes volumes de leituras. A escolha adequada de uma determinada tecnologia NoSQL pode determinar o sucesso de um sistema ou produto.

Com base nessas informações o objetivo do estudo é investigar e comparar o desempenho de três bancos de dados não relacionais distribuídos em diferentes cenários: inserção, atualização e leitura de dados com diferentes cargas de dados.

2. Experimentos

Para a realização deste estudo foram escolhidos três bancos de dados não relacionais: Redis, MongoDB e Cassandra. Cada banco de dados foi distribuído em três máquinas. O Cassandra e o Redis utilizaram um sistema de replicação, onde os dados foram persistidos nas três máquinas. Já o MongoDB utilizou um sistema de distribuição de dados criando uma máquina principal (servidor de configuração), onde reside tanto as configurações do servidor distribuído e quanto um *router*, que recebe todas as requisições e redireciona para os demais nós executores.

Outra escolha importante para este estudo foi a ferramenta de *benchmark*. Essa escolha foi baseada em outros trabalhos, como [Cooper et al 2010], que considera a ferramenta amplamente utilizada e testada, garantindo que os resultados são confiáveis. A ferramenta escolhida foi a Yahoo! Cloud Serving Benchmark.

Para os experimentos, foram também escolhidos dois tipos diferentes de *workloads*: o primeiro é chamado de *read-only*, sendo que 100% das operações são de leitura; já o segundo é o *write-only*, sendo que ele contém 100% de operações de escrita. Cada *workload* foi executado três vezes, com quantidade de operações diferentes: cem mil, um milhão e cinco milhões.

Por fim, para a realização dos experimentos foram utilizadas máquinas do MackCloud[1], sendo que cada máquina possui dois processadores Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz, 128 GB de memória RAM e 82 TB de *storage* interconectadas por rede Infiniband. As versões finais do ambiente do experimento adotadas foram:

- CentOS Linux 7 (Core)
- Redis: 6.2.6
- Cassandra: 3.11.9
- MongoDB: 5.0.2
- YCSB: 0.15.0

[1] Agradecemos ao MackCloud (https://mackcloud.mackenzie.br), Centro Multidisciplinar de Computação Científica e Nuvem da Universidade Presbiteriana Mackenzie, pelo apoio na realização desta pesquisa.

3. Resultados

É possível observar o evidente desempenho superior do Redis em todos os testes. Além disso, nota-se que o Cassandra teve um desempenho melhor do que o MongoDB nas três diferentes quantidades de operações, como mostra a Figura 1.

Como o Redis não precisa fazer uma operação de escrita em disco para cada linha inserida, mas armazena os dados em estruturas de dados em memória principal, isso pode justificar a grande diferença em relação aos demais bancos que armazenam os dados em disco. Além disso, apesar do Cassandra e do MongoDB armazenarem os dados em disco, o desempenho significativamente superior observado pelo Cassandra nas operações de escrita pode estar relacionado a dois principais fatores: a operação de escrita envolve armazenar os dados em arquivos sequenciais e *append-only* chamados *commit log* [Lakshman e Malik 2010]; o sistema de arquivos do Cassandra é baseado na estrutura de dados em árvore LSM (*Log-structured merge-tree*), que tem uma complexidade de tempo de inserção mais baixa do que uma busca [O'Neil et al 1996].

A Figura 2 apresenta os dados das operações de leitura. A diferença entre o desempenho do Cassandra e MongoDB foi menor, uma vez que a quantidade de operações por segundo do Cassandra diminuiu e a do MongoDB aumentou. Uma possível explicação para essa diferença pode estar no fato do modelo de dados usado em cada banco, uma vez que o modelo orientado a documento, utilizado pelo MongoDB, possui um desempenho melhor para operações de leitura em relação ao orientado à coluna, utilizado pelo Cassandra. Isso ocorre pois os dados no modelo de documentos estão mais próximos um do outro, já que foram salvos em formato JSON. Assim, todos os dados estão dentro de um objeto maior e, ao buscar por esse objeto, todos os dados contidos nele já são previamente armazenados [Newbedev 2021].

4. Conclusão

Os objetivos propostos no início da pesquisa, de capturar *benchmarks* de cada banco de dados distribuído NoSQL e realizar uma comparação entre os desempenhos em cada cenário foram alcançados. Comparando a média da quantidade de operações por segundo por cada banco, nota-se que diferenças de desempenho significativas foram exemplificadas de acordo com qual tecnologia foi utilizada em certo tipo de *workload* que um sistema ou produto possa esperar receber.

Para uma persistência de dados de grandes *workloads* de escrita, o Cassandra pode ser considerado uma boa solução, maximizando a quantidade de operações executadas por segundo. Enquanto para uma persistência com *workloads* de leitura, o MongoDB se mostrou uma boa escolha. Além disso, o Redis é certamente uma tecnologia que oferece um alto desempenho, especialmente na operação de leitura, em troca da utilização de um tipo de memória mais cara e volátil, como a memória RAM ao invés de memória em disco.

Referências

Abadi D. (2012) “Consistency Tradeoffs in Modern Distributed Database System Design”, Artigo Acadêmico - Yale University.

Berg K., Seymour T., Goel R. (2013) “*History of Databases*”, International Journal of Management & Information Systems.

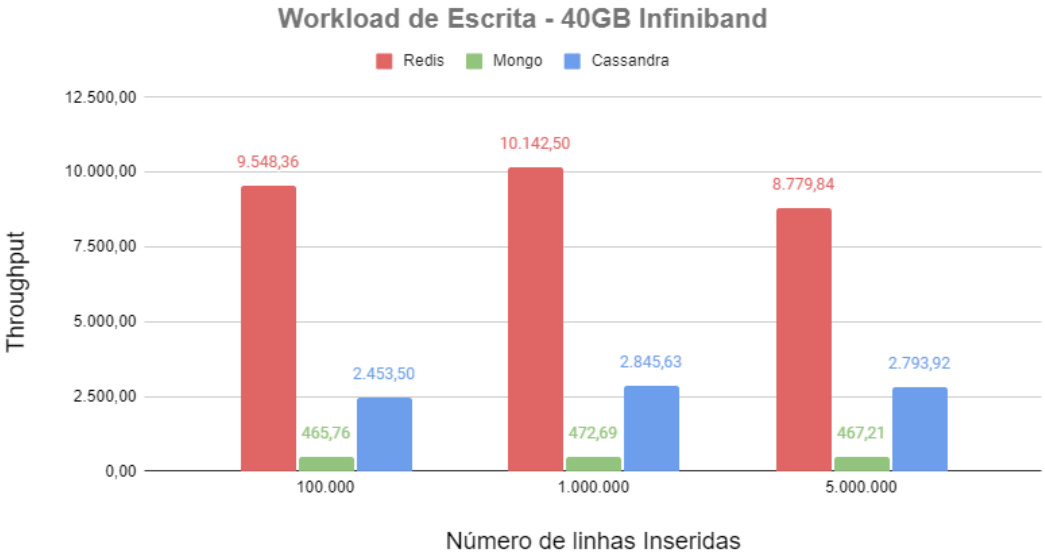

Figura 1. Comparação do *throughput* para o *workload* que contém apenas operações de escrita.

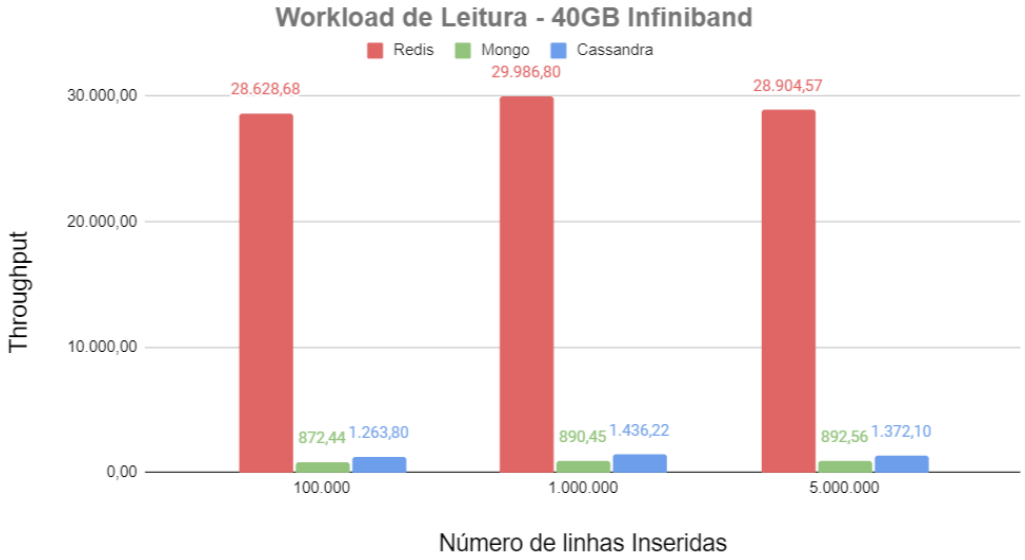

Figura 2. Comparação do *throughput* para o *workload* que contém apenas operações de leitura.

Cooper B, et al. (2010) "*Benchmarking Cloud Serving Systems with YCSB, Yahoo! Research*", Santa Clara, CA, USA.

Lakshman A. Malik P. (2009) "*Cassandra - A Decentralized Structured Storage System*". Artigo acadêmico - Cornell Bower University

NEWBEDEV (2021). "*How does column-oriented NoSQL differ from document-oriented?*". Disponível em: <https://newbedev.com/how-does-column-oriented-nosql-differ-from-document-oriented>. Acesso em: 23 nov 2021.

O'Neil, P., et al. (1996). "*The Log-Structured Merge-Tree (LSM-Tree)*". Acta Informatica 33, 351–385.