Avaliação de Desempenho para Banco de Dados com Genoma em Nuvem Privada

Luan Dopke[1], **Dinei A. Rockenbach**[1,2], **Dalvan Griebler**[1,2]

[1] Laboratório de Pesquisas Avançadas para Computação em Nuvem (LARCC), Faculdade Três de Maio (SETREM), Três de Maio, Brasil

[2]Escola Politécnica, Grupo de Modelagem de Aplicações Paralelas (GMAP), Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, Brasil

{luandopke,dineiar}@gmail.com,dalvan.griebler@pucrs.br

Resumo. *Os bancos de dados são ferramentas particularmente interessantes para a manipulação de dados gerados através do sequenciamento de DNA. Este artigo tem como objetivo avaliar o desempenho de três bancos de dados com cargas relacionadas ao sequenciamento de DNA: PostgreSQL e MySQL como bancos de dados relacionais e MongoDB como banco de dados NoSQL. Os resultados demonstram que o PostgreSQL se sobressai aos demais.*

1. Introdução

O processo de sequenciamento do DNA consiste em descobrir a sequência em que ocorrem as bases nitrogenadas em uma molécula de DNA. Esse processo auxilia em áreas da biologia e da medicina, como estudos evolutivos e identificação de doenças. O resultado do sequenciamento pode ser salvo em arquivos de diversos formatos, como o FASTA, que é facilmente processado por softwares mas ocupa grande espaço de armazenamento.

Os dados genômicos podem ser manipulados por bancos de dados relacionais (SQL) ou não relacionais (NoSQL). Os bancos relacionais são caracterizados por dados estruturados em tabelas e colunas e oferecerem transações ACID. Por outro lado, os bancos NoSQL costumam oferecer maior versatilidade na estrutura (*schema*), foco em disponibilidade ou tolerância a falhas e menos garantias em aspectos como as transações ACID. Os diferentes bancos de dados NoSQL já foram avaliados por [Rockenbach et al. 2018].

A avaliação de bancos de dados para manipular dados genômicos já foi abordada por [Celesti et al. 2019], que compara o desempenho dos bancos de dados MySQL (relacional) e MongoDB (NoSQL) para buscar sequências do DNA humano. A avaliação baseou-se em duas consultas: (I) realizando a busca através do nome do projeto, numero da região e sequência, correspondentes a um nucleotídeo específico; e (II) realizando a busca através da sequência de bases nitrogenadas. O MongoDB apresentou melhor desempenho do que o MySQL. O trabalho de [Karimi et al. 2014] utilizou índices bitmap para melhorar a velocidade de busca de genomas de bactérias em um ambiente com Hadoop MapReduce, Hive e Hbase.

O objetivo deste trabalho é expandir as avaliações de desempenho realizadas por [Celesti et al. 2019], porém compreendendo a avaliação de três bancos de dados: PostgreSQL, MySQL e MongoDB. Será avaliado o comportamento dos bancos de dados em um ambiente de Nuvem Privada, com 2, 4, 6, 8 e 10 núcleos de processamento no servidor de banco de dados, portanto avaliando como esses bancos de dados utilizam os recursos de ambientes multi-núcleo executando uma única consulta, representando um ambiente em que um único pesquisador deseja explorar todos os recursos de hardware.

2. Experimentos

O conjunto de dados utilizado nos testes foi a sequência do DNA humano obtido do National Center for Biotechnology Information (NCBI)[1], no formato FASTA, que foi processado por um algoritmo responsável por carregar os dados nos bancos de dados a serem testados. Para os bancos relacionais foram utilizadas quatro tabelas: `project`, com a descrição do projeto; `geneticcode`, para armazenar os arquivos do projeto; `region`, com os dados dos genomas desses arquivos; e `sequence`, com as sequências de bases nitrogenadas de cada região do genoma. No MongoDB, os dados foram estruturados na forma de um único documento, contendo informações sobre o projeto, a região e a sequência de bases nitrogenadas. Todos os bancos de dados foram testados em suas configurações padrão, nenhuma alteração para melhorar o desempenho foi aplicada.

O ambiente de avaliação foi a Nuvem Privada do LARCC[2], gerenciada pela ferramenta OpenNebula versão 5.12.0.3, já analisada e comparada a outras ferramentas por [Vogel et al. 2016]. Foram criadas três máquinas virtuais (uma para cada banco de dados) com 20 GB de RAM cada, e outra para atuar como cliente, executando as consultas e processando os resultados, com 10 GB de RAM, todas com o sistema operacional Ubuntu Server 20.04.1 LTS. O ambiente físico é composto por duas máquinas HP Proliant DL385 G6, com um processador AMD Opteron 2425 2100 MHz 6-Core e 32 GB de memória RAM DDR3, cada uma possuindo 5 interfaces de rede Gigabit.

Para avaliar os bancos de dados, foi desenvolvido um script em linguagem Python que é responsável por executar a consulta aos bancos, captar o tempo de execução da mesma, reiniciar o banco de dados e limpar o cache de página das máquinas virtuais que hospedam os bancos de dados, utilizando o comando *drop_caches* do Kernel a cada nova consulta. As consultas foram testadas uma após a outra, sequencialmente, para não haver concorrência no banco de dados e repetidas 30 vezes, portanto os resultados apresentados representam a média dessas execuções. A conexão aos bancos de dados foi implementada com as bibliotecas `psycopg2` versão 2.8.6, `mysql-connector` versão 8.0.22 e `pymongo` versão 3.11.2. Foram avaliadas as mesmas consultas do trabalho de [Celesti et al. 2019], a primeira consulta pode ser visualizada na Figura 1 e busca o código identificador de uma sequência de um nucleotídeo através do nome do projeto, número da região e número da sequência para ambas as linguagens.

```sql
SELECT sequenceId FROM project p
JOIN geneticcode c on p.projectid = c.projectId
JOIN region e on e.geneticcodeid = c.geneticcodeid
JOIN sequence s on s.regionId = e.regionId
WHERE p.name = 'human' AND e.regionNumber = 5 AND s.sequenceNumber = 2
```

```javascript
sequence.find({'name':'human', 'regionnumber':5, 'sequencenumber':2})
```

Figura 1. Primeira consulta executada em SQL e MongoDB.

A segunda consulta demonstrada na Figura 2 analisa a capacidade de encontrar uma sequência de bases nitrogenadas específica, retornando a sequência completa, a região e o código genético onde a sequência foi encontrada. A consulta SQL para os bancos relacionais usa o operador `LIKE` para realizar a busca, enquanto que para o MongoDB, é utilizado o operador `$regex`, análogo ao `LIKE` dos bancos relacionais.

[1] https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.25

[2] https://larcc.setrem.com.br/

```sql
SELECT * FROM project p
JOIN geneticcode c on p.projectId = c.projectId
JOIN region r on r.geneticCodeId = c.geneticcodeid
JOIN sequence s on s.regionId = r.regionid
WHERE s.necleotides LIKE '%CCTTGCCTCCTTCAGACTTGTACTTAAAAATCTCAGTAAG%'
```

```javascript
sequence.find({'nucleotides':
    {'$regex': "CCTTGCCTCCTTCAGACTTGTACTTAAAAATCTCAGTAAG"}
})
```

Figura 2. Segunda consulta executada em SQL e MongoDB.

O tempo de execução (em segundos) das duas consultas pode ser visualizado no eixo Y dos gráficos na Figura 3. O eixo X representa a quantidade de núcleos de processamento (2, 4, 6, 8 e 10), cada banco é representado por uma coluna de cor diferente. Na primeira consulta o desempenho do PostgreSQL foi melhor do que os demais bancos de dados, enquanto que o MongoDB obteve o pior desempenho (2750% pior do que o PostgreSQL). Identifica-se que não houve mudança significativa de desempenho ao aumentar o número de núcleos de processamento, portanto nenhum dos bancos foi capaz de aproveitar os recursos disponíveis para atender a uma única consulta de um único cliente.

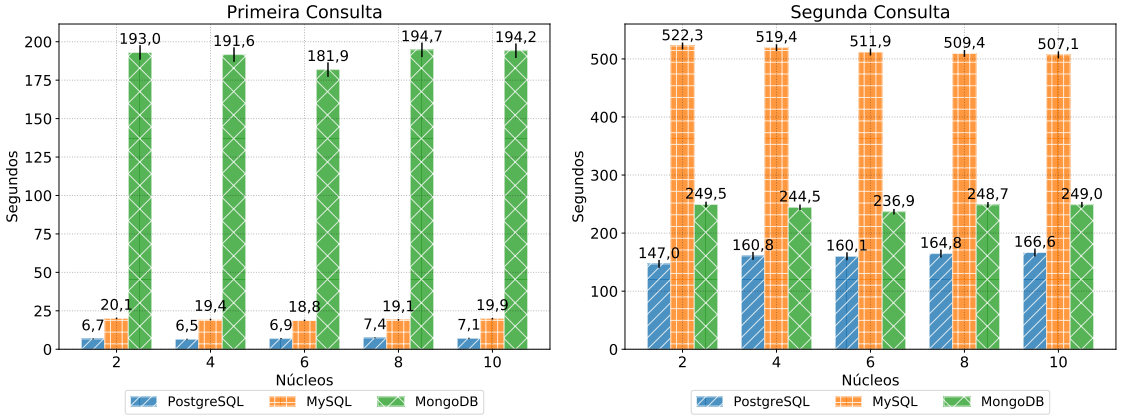

Figura 3. Avaliação de desempenho das consultas.

Na segunda consulta constata-se que novamente o PostgreSQL obteve o melhor cenário, conseguindo destacar-se dos demais bancos de dados, porém desta vez, o MySQL obteve resultados inferiores ao banco de dados NoSQL MongoDB, que apresentou um tempo 210% melhor. Quando comparado ao PostgreSQL, o MySQL foi 320% mais lento. Mais uma vez, nota-se que ao avaliar as consultas em um ambiente em que não há concorrência, a quantidade de núcleos disponíveis não impacta no resultado.

A Figura 4 demonstra o consumo de recursos durante a execução da segunda consulta, em ambiente com 6 núcleos. Representando o consumo de recursos, o eixo Y demonstra o uso de CPU para o primeiro gráfico, enquanto no segundo gráfico é apresentado o consumo de memória RAM, ao passo que o eixo X exibe o tempo decorrido na execução da consulta. É possível perceber que o PostgreSQL consome cerca de 40% da capacidade de processamento disponível, portanto ele faz processamento paralelo para acelerar a consulta. Os resultados indicam que MySQL e MongoDB utilizam apenas um único núcleo para fazer o processamento da consulta. O gráfico de monitoramento de

memória demonstra que o PostgreSQL utilizou mais memória RAM, chegando próximo dos 100%, seguido do MongoDB com aproximadamente 80% e por último o MySQL utilizando em torno de 55% da memória disponível.

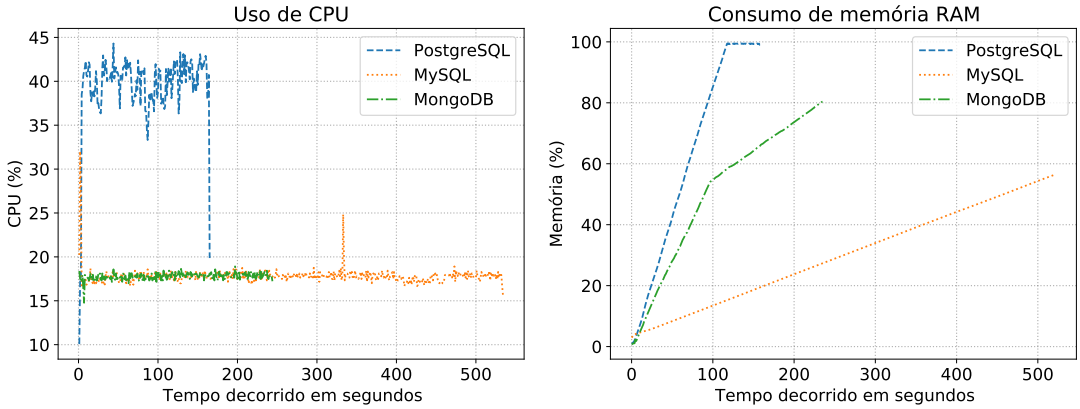

Figura 4. Monitoramento do consumo de recursos nas instâncias com 6 núcleos.

3. Conclusões

O artigo apresentou uma avaliação de desempenho de três bancos de dados, PostgreSQL e MySQL como bancos de dados relacionais e MongoDB como alternativa NoSQL. Conclui-se que em ambas as consultas realizadas, o banco de dados relacional PostgreSQL obteve maior desempenho, comprovando-se uma ferramenta mais rápida na manipulação de dados genômicos. Ao avaliar o desempenho das consultas configurando diferentes quantidades de núcleos, não houve diferença significativa de desempenho, comprovando que na execução de apenas uma consulta em um cliente, o número de núcleos não impacta no resultado final. Como trabalho futuro, deseja-se avaliar o desempenho dos bancos de dados compreendendo o conjunto de dados provido através do sequenciamento de DNA em ambiente que dispõe de consultas concorrentes com diversos núcleos, além de mensurar o impacto da criação de índices no desempenho das consultas.

Agradecimentos Os autores agradecem ao Laboratório de Pesquisa Avançada em Computação em Nuvem (LARCC/SETREM, Brasil) por fornecer recursos de computação.

Referências

Celesti, F., Celesti, A., Galletta, A., Fazio, M., and Villari, M. (2019). Optimizing the research of dna sequences in a nosql document database: A preliminary study. In *2019 IEEE Symposium on Computers and Communications (ISCC)*, pages 1153 – 1158, Barcelona, Spain. IEEE.

Karimi, R., Bellatreche, L., Girard, P., Boukorca, A., and Hajdu, A. (2014). Binos4dna: Bitmap indexes and nosql for identifying species with dna signatures through metagenomics samples. In *ITBAM 2014: Information Technology in Bio- and Medical Informatics*, pages 1 – 14, Munich, Germany. Springer, Cham.

Rockenbach, D. A., Anderle, N., Griebler, D., and Souza, S. (2018). Estudo Comparativo de Bancos de Dados NoSQL. *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação (REABTIC)*, 1(8).

Vogel, A., Griebler, D., Maron, C. A. F., Schepke, C., and Fernandes, L. G. (2016). Private iaas clouds: A comparative analysis of opennebula, cloudstack and openstack. In *24th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 672–679, Heraklion Crete, Greece. IEEE.