ESTUDO COMPARATIVO ENTRE POSTGRESQL (SGBD) E APACHE CASSANDRA (NOSQL)

Antonia Tarsilla C. Lima [1], Vinícius F. Diógenes [1], Jeferson Q. Pereira [1].

[1] Campus Pau dos Ferros - Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN) - Pau dos Ferros - RN - Brasil.
tarsillacosta.lima@gmail.com, vinicius_sfo@hotmail.com, jefersonqueiroga@gmail.com

Abstract. *Today's information is vital to thousands of organizations, and businesses, which have huge data sources stored on multiple systems. The choice of the database should be judicious, since they store information of extreme relevance, so the databases must present high power of reliability. This article aims to explain the main characteristics of the NoSQL banks and the relational databases SGBDs, in addition to comparing a migration of two text files one with 3 and another with 6 thousand data, in a program made in Java using PostgreSQL (SGBD) and Apache Cassandra (NoSQL).*

Keywords: *NoSQL, Database, SGBDs, migration, JavaFX.*

Resumo. *Hoje em dia as informações são vitais para milhares de organizações, e empresas, que possuem gigantescas fontes de dados armazenadas em diversos sistemas. A escolha do Banco de Dados deve ser criteriosa, pois armazenam informações de extrema relevância, assim os bancos devem apresentar alto poder de confiabilidade. Com isso esse artigo tem como objetivo explicar as principais características dos bancos NoSQL e os bancos relacionais SGBDs, além de fazer uma comparação com uma migração de dois arquivo texto um com 3 e outro com 6 mil dados, em um programa feito em Java utilizando a ferramenta PostgreSQL (SGBD) e Apache Cassandra (NoSQL).*

Palavras-chave: *NoSQL, Banco de Dados, SGBDs, migração, JavaFX.*

1. Introdução

O universo digital se expande a cada dia. Um Estudo da EMC (NYSE: EMC), empresa multinacional norte-americana, líder do mercado internacional de armazenamento de dados, divulgou em 09 de março de 2014 que já existem disponíveis hoje no mundo quase 1 septilhão de bits, total semelhante ao de estrelas conhecidas no céu, segundo a Agência Espacial Europeia. A estimativa é que, até 2020, o número de dados armazenados em computadores, servidores, celulares, smartphones e tablets sejam, no mínimo, multiplicados por seis; um volume tão gigantesco, que os especialistas passaram a medi-lo em termos de distância da Terra à Lua [Machado, 2014].

George et al. (2000) fala que desta forma a internet torna-se um dos principais meios de criação de transporte, difusão e armazenamento de informação para todos os objetivos, sendo um meio diverso, simples e de baixo custo, servindo como um eficaz componente para o ambiente de negócios, que predomina nos sistemas econômicos atuais.

De modo breve, os bancos de dados são definidos como um conjunto de dados relacionados entre si, armazenados segundo uma determinada estrutura de forma que possam ser recuperados quando necessário. No estudo realizado neste artigo, serão utilizadas duas bases de dados, uma utilizando o modelo relacional, é um modelo de dados representativo (ou de implementação), adequado a ser o modelo subjacente de um Sistema Gerenciador de Banco de Dados (SGBD), que se baseia no princípio de que todos os dados estão armazenados em tabelas (ou, matematicamente falando, relações). E o outro utilizando o modelo NoSQL (termo usado para descrever bancos de dados não relacionais de alto desempenho. Os bancos de dados NoSQL usam diversos modelos de dados, incluindo documentos, gráficos, chave-valor e coluna).

O objetivo deste artigo é de apresentar um estudo comparativo de desempenho entre estes dois tipos de banco de dados: um SGBD e outro NoSQL. Os bancos escolhidos foram PostgreSQL (SGBD) e o Apache Cassandra (NoSQL), este estudo terá como fundamento demonstrar qual banco se sairá melhor com uma grande quantidade de dados a ser migrada, ajudando o desenvolvedor na hora de escolher uma ferramenta no qual ele possa ter mais confiança, tendo o conhecimento de qual possa auxiliá-lo naquele momento. A presente aplicação, criada para a migração destes dois bancos foi criada em Java utilizando a ferramenta JavaFX, nela foi criada um simples CRUD, com as funções de novo, editar, listar e remover, além da classe de migração e conexão para ambos os bancos.

Este artigo foi estruturado da seguinte maneira: a seção 2 apresenta de maneira simples e rápida a definição sobre alguns pontos principais para entendimento do projeto, na seção 3 é explicado o ambiente utilizado para os testes, seção 4 demonstra os resultados dos testes obtidos e na seção 5, uma breve conclusão sobre seus resultados.

2. Metodologia

Para a realização deste trabalho, primeiramente foi realizado um levantamento teórico sobre a área de bancos de dados e a introdução da tecnologia de NoSQL e dos SBDSs relacionais. Em seguida, um experimento foi realizado numa aplicação desenvolvida em Java. Para isto, dois modelos de classes foram construídos, uma tendo por base o modelo relacional (SGBD) e outra o modelo NoSQL, a fim de posteriormente estes modelos serem comparados.

O PostgreSQL foi escolhido para o desenvolvimento deste trabalho por ser um Sistema Gerenciador de Banco de Dados Objeto-Relacional (SGBD) livre, ou seja, dá suporte aos modelos de banco de dados relacional e objeto-relacional, e o Apache Cassandra por ser um bancos de dados não relacional de alto desempenho além de ser um sistema de código aberto projetado para gerenciar grandes volumes de dados em tempo real, permitindo resposta imediata e suporte a pontos de falha de categoria NoSQL.

Ao fim das comparações uma avaliação foi feita para decidir sobre a viabilidade de se utilizar banco de dados NoSQL e SGBDs relacionais, em migrações com dados muito altos, demonstrado no gráfico 1 da seção 5, estudo comparativo.

3. Referencial Teórico

Nesta seção são detalhados os conceitos e definições necessárias para o entendimento do trabalho realizado.

3.1 Modelos Relacionais

O modelo relacional foi criado por Edgar F. Codd, na década de 70 e começou a ser usado com o advento dos bancos de dados relacionais, nos anos 80. A ideia de modelo relacional se baseia no princípio de que as informações em uma base de dados podem ser consideradas como relações matemáticas além de poderem ser representadas, de maneira uniforme, através do uso de tabelas onde as linhas representam as ocorrências de uma entidade e as colunas representam os atributos de uma entidade do modelo conceitual [Siqueira, 2014].

3.1.1 PostgreSQL

O PostgreSQL é um banco de dados objeto-relacional(SGBD) baseado no POSTGRES versão 4.2 que foi desenvolvido pelo Departamento de Ciência e Computação da Universidade da Califórnia em Berkeley. Ele foi o pioneiro em muitos conceitos que só foram disponíveis muito depois em alguns sistemas de banco de dados comerciais [Guilherme, 2006]. Entre as principais vantagens do modelo relacional podemos citar:

- Independência total dos dados;
- Visão múltipla dos dados;
- Redução acentuada na atividade de desenvolvimento. Particularmente para extração de dados para relatórios e consultas específicas do usuário;
- Maior segurança no acesso aos dados;
- Maior agilidade para consulta/atualização;
- Qualidade dos dados garantida por restrições de integridade (identidade, referencial e de domínio).

3.2 Modelo NoSQL (Not only SQL)

O NoSQL é uma abordagem ao projeto de banco de dados que pode acomodar uma grande variedade de modelos de dados, incluindo formatos de valor-chave, documento, coluna e gráfico. O NoSQL, que significa "não apenas SQL ", é uma alternativa aos bancos de dados relacionais tradicionais em que os dados são colocados em tabelas e o esquema de dados é cuidadosamente projetado antes da construção do banco de dados. Os bancos de dados NoSQL são especialmente úteis para trabalhar com grandes conjuntos de dados distribuídos[Bonfioli, 2018].

3.2.1 Apache Cassandra

Cassandra é um projeto de sistema de banco de dados distribuído, escalável de segunda geração, reúne a arquitetura do Dynamo, Amazon e do modelo de dados baseado em BigTable do Google[George, 2018].

Cassandra está em uso em Netflix, eBay, Twitter, Urban Airship, Constant Contact, Reddit, Cisco, OpenX, Digg, Cloudkick, Ooyala, e mais empresas que possuem grandes conjuntos de dados ativos. O maior cluster Cassandra conhecido tem mais de 300 TB de dados em mais de 400 máquinas [George, 2018].

4. Ambiente

Os testes foram realizados em um computador com processador Pentium® Dual-Core CPU E5500 @ 2.80GHz, com o sistema operacional Windows 7. A versão do PostgreSQL usada foi a 9.5 e a do Apache Cassandra foi a 3.11.1.

Para criação do programa de teste, a linguagem Java foi utilizada junto com o (JavaFX Scene Builder 2.0), utilizada com seus respectivos drivers para conexão com o banco de dados. Para o postgreSQL foi utilizado o drive (Driver JDBC do PostgreSQL) e para o Apache Cassandra (Cassandra JDBC Driver).

As Figuras 1 e 2 apresentam os códigos de inserção dos dados nos dois modelos distintos, lembrando que o código de inserção do postgreSQL pode ser utilizado tanto para ele quanto para inserções em bancos de dados Mysql ou distintos. Na figura 3, demonstra uma pequena parte do arquivo texto utilizado nos testes das migrações.

```
public boolean inserir(Aluno aluno) {
      String query = "INSERT INTO public.\" Aluno\ "(\" nomeAluno\","
            + "\"matriculaAluno\", "
            + "\"Curso_idCurso\", \"Campus_idCampus\","
            + "\"SituacaoAluno_idSituacaoAluno\")"
            + " VALUES (?, ?, ?, ?, ?, ?);";

      try {
         PreparedStatement stmt = connection.prepareStatement(query);
         stmt.setString(1, aluno.getNomeAluno());
         stmt.setString(2, aluno.getMatriculaAluno());
         stmt.setInt(3, aluno.getCurso().getId_Curso());
         stmt.setInt(4, aluno.getCampus().getId_Campus());
         stmt.setInt(5, aluno.getSituacaoAluno().getIdSituacaoAluno());
         stmt.execute();
         return true;
      } catch (SQLException ex) {
         Logger.getLogger(AlunoDAO.class.getName()).log(Level.SEVERE, null, ex);
         return false;
      } }
```

Figura 1. Demonstração do código de inserção postgreSql.

```
public boolean inserir(Aluno aluno) {
      String query = "INSERT INTO Aluno(nomeAluno,matriculaAluno,Curso_idCurso,"
            + "Campus_idCampus,SituacaoAluno_idSituacaoAluno)"
            + " VALUES (?, ?, ?, ?, ?, ?)";
      try {
         PreparedStatement prepared = session.prepare(query);
         ResultSet rs = session.execute(prepared.bind(
               UUID.randomUUID(),
               aluno.getNomeAluno(),
               aluno.getMatriculaAluno(),
               aluno.getCurso().getId_Curso(),
               aluno.getCampus().getId_Campus(),
               aluno.getSituacaoAluno().getIdSituacaoAluno()));
         return true;
      } catch (Exception ex) {
         Logger.getLogger(AlunoDAO.class.getName()).log(Level.SEVERE, null, ex);
         return false;
```

```
    }
}
```

Figura 2. Demonstração do código de inserção Apache Cassandra.

20161011010038;Abdênego Benjamim de Araújo Moreno;Técnico de Nível Médio em Edificações;CNAT;Matriculado 19951100017;Abdenego da Silva Santos;Técnico de Nível Médio em Construção Civil;CNAT;Matriculado 20171012090034;Abdias da Silva Tavares;Engenharia de Energia;CNAT;Matriculado 20172021020005;ABDIEL NONATO SOARES DA CUNHA SILVA;Técnico de Nível Médio em Edificações, na Forma Integrada, Modalidade EJA;MO;Matriculado 20141128070709;Abdon Francisco Santana Neto;Técnico de Nível Médio em Multimídia;CAL;Matriculado 20161054110002;Abdon Soares de Souza Junior;Licenciatura em Informática;IP;Matriculado 20172153040103;Abel Gomes de Oliveira;Tecnologia em Gestão Ambiental;EAD;Matriculado 20162023040017;Abel Soares de Souza Neto

Figura 3. Demonstração arquivo texto.

5. Estudo comparativo

Com base nos desempenhos dos bancos, foi realizado o teste comparativo entre o Apache Cassandra e PostgreSQL na migração de dados.

O Gráfico 1 apresenta o desempenho em segundos de cada banco, demonstrando qual banco se saiu melhor em tempo de migração de dados, ao observar o gráfico, pode se ver que o postgreSQL teve um tempo maior, chegando a mais de três minutos com o arquivo texto de 6 mil dados, e com o arquivo texto de 3 mil, chegando a mais de dois minutos, pode se observar então que o Apache Cassandra executou ambas as ações de migração com os arquivos de maneira bem mais rápida, com uma vantagem de quase um minuto a menos.

No PostgreSQL o tempo foi calculado pelo próprio ambiente de dados, e Apache Cassandra seu tempo foi calculado no próprio terminal, sendo processadas de uma só vez todas as linhas do arquivo texto em cada ambiente.

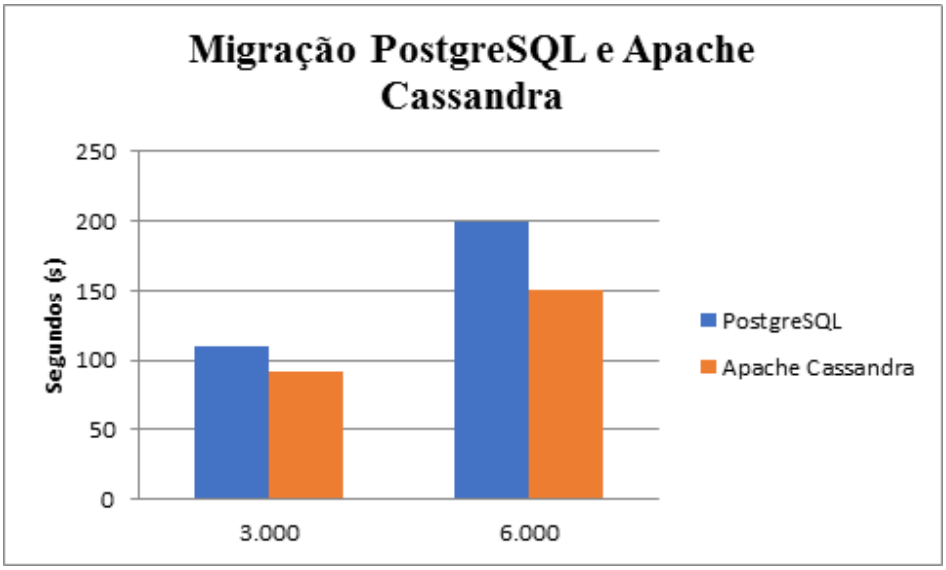

Gráfico 1 . Migração dos dados

6. Considerações Finais

O presente trabalho apresentou a comparação da migração de distintas quantidades de dados para os bancos de dados PostgreSQL e Apache Cassandra. Ambos os bancos foram utilizados conforme as configurações padrão de cada um, no ambiente de teste proposto.

Com os resultados pode-se concluir diante os testes que o Apache Cassandra teve um tempo de migração de dados muito melhor que o PostgreSQL, com um bom rendimento de leitura, ele foi mais flexível e minimizou o tempo, diferentemente do PostgreSQL, cumprindo o combinado de que eles oferecem mais flexibilidade para gravação, não precisam obedecer a uma concepção prévia de seu formato. Novas propriedades podem ser adicionadas a uma entidade do sistema no instante de gravação. Com isso, o time de desenvolvimento amplia a agilidade para criar e testar novas funções sobre informações novas.

O PostgreSQL (SGBD) possui uma vantagem maior aos bancos NoSQL pois possui interface com diversos ambientes e linguagens de programação, como C, C++, MS Visual Basic, Perl e Java. Todos esses recursos proporcionam ao programador e ao administrador de banco de dados realizarem suas tarefas e atender suas perspectivas mais específicas.

O trabalho realizado contribuiu com a pesquisa sobre a recente introdução da tecnologia NoSQL e SGBDs relacionais, demonstrando em prática qual se sairia melhor com uma grande gama de dados a ser migrada, auxiliando o desenvolvedor na escolha da melhor ferramenta para um projeto futuro no qual necessita migrar muitos dados de forma direta e rápida.

7. Referências Bibliográficas

MACHADO, Andre Machado. Estudo da EMC prevê que Volume de Dados Virtuais Armazenados Será Seis Vezes Maior em 2020. Disponível em: <https://oglobo.globo.com/sociedade/tecnologia/estudo-da-emc-preve-que-volume-de-dados-virtuais-armazenados-sera-seis-vezes-maior-em-2020-12147682>. Acesso em: 01 jan. 2018.

SIQUEIRA, Fernando de Siqueira. Banco de Dados I, Modelo Relacional – 2014. Disponível em: < https://sites.google.com/site/uniplibancodedados1/aulas/modelo-relacional >. Acesso em: 15 jan. 2018.

GUTIERRY, Gutierry Antonio. Conheça a geração de banco de dados Nosql e NewSQL. Disponível em: < https://www.devmedia.com.br/conheca-a-geracao-de-banco-de-dados-nosql-e-newsql/33202 >. Acesso em: 13 fev. 2018.

História Apache Cassandra NoSQL, ELDER STROPARO. Disponível em: < http://elderstroparo.blogspot.com.br/2013/10/historia-apache-cassandra nosql.html>. Acesso em: 20 jan. 2018.

GEORGE, L. et al. A era da informação: considerações sobre o desenvolvimento das tecnologias da Informação. Disponível em: <http://www.egov.ufsc.br/portal/sites/default/files/a_era_da_informacao.pdf>. Acesso em: 26 fev.2018.

TECHTARGET**.** NoSQL (Not Only SQL database). Disponível em: < http://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL>. Acesso em: 26 fev. 2018.

BONFIOLI, Guilherme Ferreira Bonfili. Banco de dados relacional e objeto-relacional: uma comparação usando PostgreSQL. Disponível em: < http://repositorio.ufla.br/bitstream/1/8354/1/MONOGRAFIA_Banco_de_dados_relacional_e_objeto_relacional_uma_compara%C3%A7%C3%A3o_usando_postgresql.pdf>. Acesso em: 01 mar. 2018.