DINo: uma ferramenta para importação de dados em bancos de dados NoSQL

Angelo Augusto Frozza[1,2], Geomar Schreiner[1], Rian Brüggemann[1], Ronaldo dos Santos Mello[1]

[1]Universidade Federal de Santa Catarina (UFSC)
Campus Universitário Trindade - CP 476 - CEP88040-900 - Florianópolis (SC), Brasil

[2]IFC - Instituto Federal Catarinense - Campus Camboriú
Rua Joaquim Garcia, S/N - CP 2016 - CEP 88340-055 - Camboriú (SC), Brasil

angelo.frozza@ifc.edu.br,{geomarschreiner,riancvb}@gmail.com, r.mello@ufsc.br

Abstract. *This paper presents DINo, a tool to help import relational data to NoSQL databases. Its main characteristics are: be multiplatform; support various types of DBMS (both relational and NoSQL); be flexible, allowing the user to make changes to the data mapping, publicly available. Tests were performed with data from an OpenStreetMap database.*

Resumo. *Este artigo apresenta o DINo, uma ferramenta para importação de dados relacionais para bancos de dados NoSQL. Suas principais características são: multiplataforma; suportar diversos tipos de SGBDs (relacionais e NoSQL); permitir alteração no mapeamento dos dados; e, estar disponível publicamente. São apresentados testes realizados com dados do OpenStreetMap.*

1. Introdução

A explosão no uso de *Big Data* fez com que grandes empresas demandassem por bancos de dados (BDs) capazes de gerenciar grandes volumes de dados de forma eficaz e com um alto desempenho. Nesse contexto, os tradicionais BDs relacionais apresentam algumas limitações quanto a alta concorrência em operações de leitura e escrita, armazenamento de *Big Data* de forma eficiente, suporte a escalabilidade horizontal, além da garantia de um serviço rápido e ininterrupto (alta disponibilidade) [Han et al. 2011].

Levando em consideração essas necessidades, uma variedade de novos sistemas de BDs surgiu, focados principalmente no baixo custo de operação e manutenção. Esses BDs são popularmente conhecidos por NoSQL, que é um acrônimo para "*Not Only* SQL". Os BDs NoSQL geralmente são classificados de acordo com o modelo de dados adotado: *chave-valor*, *documentos*, *colunar* e *orientado a grafos* [Sadalage and Fowler 2012] [Cattell 2011, Sadalage and Fowler 2012, Ruiz et al. 2015], sendo os três primeiros também chamados de modelos baseados em chave.

Como toda tecnologia nova, o surgimento dos BDs NoSQL também trouxe a necessidade de desenvolvimento de novas ferramentas que permitam melhorar a produtividade de quem usa esse tipo de BD [Ruiz et al. 2015], com destaque neste artigo para ferramentas que auxiliam na importação de dados relacionais para BDs NoSQL.

O objetivo deste artigo é apresentar o *DINo* - uma ferramenta para importação de dados relacionais para BDs NoSQL, que visa auxiliar desenvolvedores de aplicações, em

especial aqueles que tem pouco domínio sobre NoSQL, facilitando a migração de dados. O artigo está organizado da seguinte forma: na seção 2 são discutidos alguns trabalhos relacionados; a seção 3 apresenta a ferramenta e suas estratégias de mapeamento de BD relacional para NoSQL; e, a seção 4 apresenta as considerações finais e trabalhos futuros.

2. Trabalhos relacionados

Em geral, a tarefa de migração pode ser realizada de diversas maneiras: *(i)* através de *scripts* para migração de um BD antigo para um novo BD [Mojeprojekty 2017]; *(ii)* pelo uso de ferramentas de migração para casos específicos [Murari et al. 2016]; ou *(iii)* pelo uso de ferramentas com suporte a diversos modelos de BDs [Vale and Rocha 2011].

A ferramenta proposta por [Murari et al. 2016] permite a migração de um BD *Firebird* para o BD NoSQL *MongoDB* e foi desenvolvida para realizar a migração de dados de uma aplicação específica. [Vale and Rocha 2011] propõem uma ferramenta que suporta diversos SGBDs (Sistemas Gerenciadores de Banco de Dados) relacionais como entrada, bem como, diversas abordagens NoSQL como saída. No entanto, até o momento, só foi implementado o suporte para os BDs *MySQL* e *MongoDB*, respectivamente.

[Santos Neto et al. 2013] propõem um conjunto de requisitos para uma ferramenta de migração, destacando: envolver estruturas diferentes (na entrada e na saída); executar de forma paralela ou distribuída; permitir migração incremental; permitir reengenharia de dados; suportar paradigmas de BDs diferentes; migrar BDs com modelagens diferentes; testabilidade; e, boa usabilidade. O DINo busca atender a maior parte dos requisitos.

No processo de migrar uma base relacional para um BD NoSQL é importante analisar como pode ser feito o mapeamento entre os modelos. Algumas propostas são apresentadas por [Zhao et al. 2014, Schreiner et al. 2015, Claudino et al. 2015, Poffo 2016].

3. A ferramenta DINo

O *DINo - **D**ata **I**nsertion in **No**SQL* tem por objetivo auxiliar na migração de dados de um BD relacional para um BD NoSQL. Suas principais características são: suporte a diferentes SGBDs relacionais; suporte a diferentes modelos de dados NoSQL; processamento paralelo, através do uso de *threads*; interface simples e interativa.

A interface é dividida em três partes: *(1) Source*; *(2) Target*; *(3) Execute* (Figura 1). Em *Source*, são inseridas as informações para conexão com o SGBD de origem. Atualmente, o DINo suporta o *PostgreSQL*, mas pode ser atualizado para fornecer suporte a outros SGBDs relacionais através da alteração nos componentes das conexões JDBC. Não são necessários outros mapeamentos, uma vez que estes são realizados no *script* SQL de leitura de dados. Após conectar com o SGBD, o usuário deve selecionar um *database* e, em seguida, uma tabela desse *database*. Uma lista de colunas da tabela é apresenta ao usuário (Figura 2). Na sequência, é escolhido o banco NoSQL de destino e estabelecida a conexão com esse banco (Figura 2). Por fim, o usuário define como deve ser feito o mapeamento dos dados do BD *source* para o BD *target*, ou seja, define a composição da *chave* e do *valor* (Figura 3). Após definidos os parâmetros de conexão e as informações a serem importadas, os próximos passos são gerar o *script* de importação através do botão "*Generate sql*" e selecionar o botão "*Import*" para que o processo tenha início. O mapeamento automatizado de tabelas relacionais para modelos NoSQL é um tópico complexo, que é

altamente sensível ao domínio. Desta forma, este trabalho realiza a exportação de tabelas individualmente para que o usuário tenha total controle dos mapeamentos realizados. Além disso, o campo de edição da SQL flexibiliza o mapeamento na ferramenta, permitindo, por exemplo, a criação de junções com mais de uma tabela no banco relacional ou modificar o formato de algum dado. Diferente dos bancos de dados relacionais, os BDs NoSQL não implementam o conceito de relacionamento. Assim, o DINo não realiza o mapeamento das chaves estrangeiras (ou relacionamentos), os valores são tratados como colunas normais. Quando é feito o mapeamento de um BDR para um NoSQL, os relacionamentos devem ser resolvidos no momento do mapeamento. Caso deseje, o usuário pode realizar o mapeamento utilizando junções alterando o *script* SQL gerado pelo DINo antes de iniciar a importação dos dados.

O processo de migração é muito oneroso, implicando que cada registro do BD relacional seja consultado e inserido no BD NoSQL. Desta forma, a fim de melhorar o desempenho na migração dos dados, utiliza-se paralelismo para ler e inserir os dados. Durante a importação dos dados, o DINo busca quantos núcleos do processador estão ociosos e para cada núcleo ocioso é criada uma *thread* de importação. Os total de dados a ser importado é dividido pelo número de *threads* (cada *thread* é representada por uma página dos dados relacionais). As threads não tomam conhecimento uma da outra e não possuem mecanismos para importar dados de chaves estrangeiras ou outras restrições, cada uma apenas recebe um conjunto de dados e os insere sequencialmente no BD NoSQL.

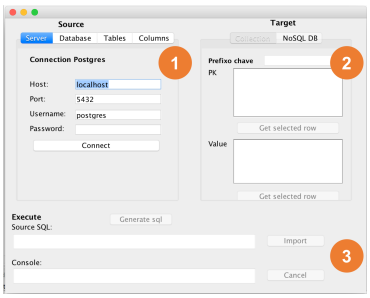

Figura 1. Interface do DINo

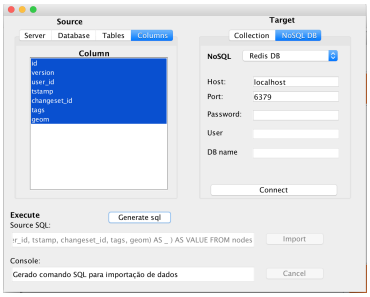

Figura 2. Conexão com o NoSQL

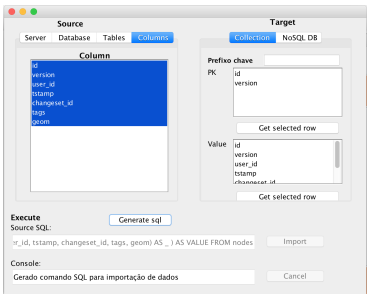

Figura 3. Mapeamento chave-valor

O DINo mapeia dados para os paradigmas NoSQL baseados em chave da seguinte forma:

NoSQL chave-valor : Consiste em definir (Figura 3): *(i)* a chave - que pode ser composta por um prefixo e um ou mais campos da tabela relacional; *(ii)* o valor - que é um documento JSON formado pelos campos selecionados na tabela relacional;

NoSQL documento : Os dados de cada tupla são agrupados em um documento JSON, que é armazenado em uma *collection* que possui o mesmo nome da tabela.

NoSQL colunar : Segue a regra: *(i)* o nome da tabela relacional é usado para criar uma *família de colunas*; *(ii)* cada coluna da tabela relacional corresponde a uma coluna no banco NoSQL; *(iii)* define-se qual é a chave da família de colunas.

Alguns testes preliminares foram realizados utilizando uma base de pontos de interesse do Uruguai disponibilizada pelo *OpenStreetMap*[1]. A tabela *nodes* (1842994

[1]http://download.geofabrik.de/south-america/uruguay.html

tuplas) foi importada para uma instância do banco *Redis* e para o *MongoDB*, ambos os BDs rodando localmente em uma máquina com processador Intel i5-7200U, 8 GB de RAM e Disco rígido de 1 TB. A importação para o *Redis* levou em média 2 minutos e 15 segundos. Já, a importação para o *MongoDB* levou em média 7 min e 53 segundos.

4. Considerações finais

A ferramenta DINo está em sua versão inicial e foi testada com os BDs *PostgreSQL* (*source*) e *MongoDB*, *Redis* e *Cassandra* (*target*), demonstrando bom desempenho por causa do uso de *threads*, além das demais características já citadas. O código fonte está disponível através do *link* `https://github.com/gbd-ufsc/DINo` e aceita contribuições de terceiros. Como trabalhos futuros, entre outros recursos, pretende-se: melhorar o desempenho da importação; permitir gerar esquemas dos BDs NoSQL criados; adicionar suporte a novos BDs; e, suportar outros tipos de estruturas de entrada, como arquivos CSV (*Comma Separated Values*) e arquivos textos.

Este trabalho foi apoiado por bolsa de iniciação científica (IC) do CNPq.

Referências

Cattell, R. (2011). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record*, 39(4):12.

Claudino, M., Souza, D., and Salgado, A. C. (2015). Mapeamentos conceituais entre os modelos Relacional e NoSQL : uma abordagem comparativa. *Revista Principia*, (28):37–50.

Han, J., Haihong, E., Le, G., and Du, J. (2011). Survey on NoSQL database. *International Conference on Pervasive Computing and Applications, ICPCA 2011*, pages 363–366.

Mojeprojekty (2017). Código fonte do projeto *sql-to-redis*. Disponível em: `https://github.com/mojeprojekty/sql-to-redis`.

Murari, M. A., Cunha, G. B. d., and Silveira, S. R. (2016). Desenvolvimento de um software para migração de um banco de dados relacional Firebird, para o não relacional MongoDB. *Revista SETREM*, (28):115–123.

Poffo, J. P. (2016). *Projeto lógico de bancos de dados NOSQL colunares a partir de esquemas conceituais entidade-relacionamento estendido (EER)*. PhD thesis.

Ruiz, D. S., Morales, S. F., and Molina, J. G. (2015). Inferring Versioned Schemas from NoSQL Databases and its Applications. *LNCS*, 9381(October):467–480.

Sadalage, P. J. and Fowler, M. (2012). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*.

Santos Neto, P. d. A. et al. (2013). Requisitos para ferramentas de migração de dados. In *SBSI - Simpósio Brasileiro de Sistemas de Informação*, pages 887–898. SBC.

Schreiner, G. A., Duarte, D., and dos Santos Mello, R. (2015). SQLtoKeyNoSQL. In *17th iiWAS*, pages 1–9, New York, New York, USA. ACM Press.

Vale, F. and Rocha, L. (2011). Nosqlayer: a framework for migrating relational datasets to nosql models. In *REIC*, volume 14. SBC.

Zhao, G., Lin, Q., Li, L., and Li, Z. (2014). Schema Conversion Model of SQL Database to NoSQL. In *IEEE 3PGCIC*, pages 355–362. IEEE.