Desafios no Mapeamento de Esquemas Conceituais Geográficos para Esquemas Físicos Híbridos SQL/NoSQL

Danilo B. Seufitelli, Mirella M. Moro, Clodoveu A. Davis Jr.

Universidade Federal de Minas Gerais, Belo Horizonte – MG – Brazil

{daniloboecha, mirella, clodoveu}@dcc.ufmg.br

Abstract. *To the best of our knowledge, there is no generic mapping from conceptual schemas to NoSQL physical schemas. This paper tackles such problem in the context of geographic databases. We discuss the solution of mapping conceptual schemas to hybrid relational/NoSQL physical schemas.*

Resumo. *Até onde pudemos determinar, não existem ainda propostas genéricas para produzir esquemas físicos para estruturas complexas NoSQL (documentos, grafos, etc). Este artigo apresenta questionamentos quanto ao mapeamento da modelagem conceitual para esquemas físicos híbridos, de modo a conciliar modelos relacionais e não relacionais.*

1. Introdução

A modelagem conceitual de dados geográficos envolve abstrações que vão além da expressividade dos modelos de dados convencionais. Modelos de dados geográficos, como o OMT-G [1], incluem primitivas para definir alternativas de representação e relacionamentos espaciais. O mapeamento de esquemas conceituais geográficos para esquemas lógicos e físicos precisa levar em conta a semântica dessas primitivas e definir a implementação final em um sistema de gerenciamento de bancos de dados (SGBD) geográfico, como o PostGIS ou o Oracle Spatial. Embora esse mapeamento tenha sido estudado para o caso de bancos de dados objeto-relacionais espacialmente estendidos [6], a crescente disponibilidade de gerenciadores NoSQL indica que podem existir situações em que seu uso em aplicações pode ser mais vantajoso. Por exemplo, um SGDB NoSQL orientado a grafos oferece melhor desempenho em tarefas de roteamento, como demonstrado em [11].

Por outro lado, a implementação de restrições de integridade espaciais [3] é uma tarefa típica dos SGBDs tradicionais, porém não disponível nos gerenciadores NoSQL. Assim, acreditamos que existam situações em que um enfoque híbrido, ou seja, esquemas físicos que combinem SGBD relacionais e NoSQL, seja o mais indicado. Portanto, este trabalho propõe avaliar o potencial para mapeamento de esquemas conceituais geográficos para esquemas híbridos, com componentes relacionais e NoSQL.

A seguir, a Seção 2 discute brevemente trabalhos relacionados. A Seção 3 apresenta o processo de mapeamento de dados geográficos para diferentes representações NoSQL, apontando diversos desafios na hora de definir tais mapeamentos. É também apresentada uma discussão sobre os desafios encontrados. A Seção 4 conclui este artigo.

2. Trabalhos Relacionados

O OMT-G é um modelo de dados orientado a objetos que oferece primitivas para a modelagem da geometria e da topologia dos dados espaciais através de três conceitos principais: classes, relacionamentos e restrições de integridade espaciais [1]. O modelo permite

a especificação de diferentes alternativas de representação geográfica e classes de objetos com múltiplas representações.

O mapeamento de esquemas OMT-G para esquemas lógicos e de implementação foi estudado por Hora et al. [6], tendo como alvo SGBD objeto-relacionais e esquemas GML. Foram definidos algoritmos que estabelecem a equivalência entre representações conceituais mais complexas em OMT-G e estruturas de representação geográfica mais simples (e.g., pontos, linhas, polígonos) para o esquema de implementação, adicionando elementos para a implementação concomitante de restrições de integridade, de modo a respeitar a semântica das primitivas conceituais. Esses elementos incluem asserções (`CHECK`), restrições convencionais (`CONSTRAINT`s) e funções de verificação topológica implementadas como gatilhos (*triggers*).

O mapeamento para SGBDs NoSQL, por outro lado, não é ainda abordado na literatura. Isso provavelmente decorre do fato de existirem sistemas NoSQL em quatro arquiteturas distintas de armazenamento de dados: chave-valor, orientado a colunas, orientado a documentos e orientado a grafos [7, 10]. Dentre os argumentos para adoção de SGBD NoSQL estão o crescimento horizontal escalável, que visa prover uma grande quantidade de operações de leitura e escrita por segundo. Esses sistemas também notabilizam-se por serem replicáveis, potencialmente distribuídos entre vários servidores, e terem interface ou protocolo de acesso simples. Além disso, possuem um sistema de paralelismo e controle de concorrência menos estrito que o gerenciamento de transações em bancos relacionais, com distribuição eficiente de índices e uso intensivo de memória. Outra característica importante é ter a possibilidade de realizar alterações estruturais dinâmicas, em contraste com a relativa rigidez das estruturas tabulares dos SGBD relacionais.

Bugiotti et al. [2] propuseram uma metodologia de projeto de banco de dados NoSQL com o objetivo de projetar uma "boa"representação de dados NoSQL e visando obter escalabilidade, desempenho e consistência em aplicações Web da nova geração. Experimentos mostraram que o projeto de implementação deve ser conduzido com cuidado, pois o desempenho e a coerência das operações de acesso aos dados podem ser consideravelmente afetados.

Com relação a esquemas híbridos, Moro et al. [9] descrevem o ReXSA, uma ferramenta para projetar esquemas de banco de dados que combinam o armazenamento de dados relacionais e dados XML. Em outras palavras, o ReXSA avalia e recomenda um esquema de banco de dados que harmoniza modelos de dados relacionais e XML.

Assim como [9], também consideramos mapear esquemas conceituais para esquemas que combinem dados de naturezas diversas. Porém, este trabalho difere dos trabalhos citados em dois pontos: ao contário dos demais que consideram o projeto de bancos de dados NoSQL para dados convencionais, nós focamos nas especificidades dos dados geográficos; e estudamos questões relativas ao processo de mapeamento de esquemas conceituais geográficos para esquemas de implementação híbridos NoSQL/SQL.

3. Processo de Mapeamento

O mapeamento de um esquema conceitual geográfico para um esquema de implementação envolve decisões sobre diversos fatores. Com a flexibilidade dos distintos formatos utilizados por gerenciadores NoSQL, um fator a ser analisado é o tipo de armazenamento

Figura 1. Esquema OMT-G: Representação de ruas e bairros

físico a ser utilizado (e.g. chave-valor, documentos, grafos e família de colunas). Em cada um desses formatos, existem diversas possibilidades de organização dos dados e recursos de indexação. Há ainda a dificuldade na representação dos relacionamentos espaciais e restrições de integridade espaciais, visto que gerenciadores NoSQL são livres de esquema e não possuem o mesmo conceito de chave estrangeira dos SGBD relacionais.

Para dados geográficos, muitos dos bancos de dados NoSQL adotam o formato GeoJSON, que é um formato para a codificação de uma variedade de estruturas de dados geográficos. O GeoJSON adere aos padrões estabelecidos pelo Open Geospatial Consortium (OGC) e suporta os seguintes tipos de geometria: *Point, LineString, Polygon, MultiPoint, MultiLineString,* e *MultiPolygon*. Listas de geometrias são representadas por um *GeometryCollection*. Geometrias com propriedades adicionais são objetos *Feature*, e as listas de características são representados pela *FeatureCollection*.

Entretanto, dadas a complexidade e as peculiaridades das aplicações geográficas, criar uma estrutura de banco de dados por meio de esquemas GeoJSON não é uma atividade simples. Além disso, é mais fácil especificar e entender os conceitos e os relacionamentos de um sistema usando um esquema conceitual geográfico, para aproveitar a natureza visual dos diagramas de classes e outras primitivas, antes de tentar codificar diretamente as estruturas de banco de dados em esquemas GeoJSON.

Como exemplo da modelagem conceitual, a Figura 1 contém um fragmento de diagrama de classes utilizando o OMT-G. A cidade é formada por um polígono que é subdividido em bairros (subdivisão planar), de modo a estabelecer uma relação de composição espacial: toda cidade é formada por bairros, e não há bairros que se sobrepõem e nem que excedam os limites das cidades. Cada bairro possui segmentos de rua, relacionados em rede com seus cruzamentos. Nesse tipo de relacionamento, deve ser assegurado que, para cada nó exista pelo menos um arco, e a cada arco correspondam sempre dois nós.

Para exemplificar as dificuldades do mapeamento entre esquemas conceituais geográficos e esquemas de implementação NoSQL, apresentamos a seguir o mapeamento do esquema da Figura 1 para de três tipos distintos de gerenciadores NoSQL: orientados a documentos, a grafos e a família de colunas.

Mapeamento para SGBD Orientado a Documentos. Bancos de dados orientados a documentos utilizam um conjunto de coleções de atributos e valores, onde um atributo pode ser multivalorado, formando assim os documentos. Estes documentos são autodescritivos, com uma estrutura hierárquica em árvore, que pode conter mapas, coleções e valores escalares [5]. Neste artigo, é considerado o MongoDB como exemplo de gerenciador NoSQL orientado a documentos, pois suporta dados geográficos (GeoJSON) com o uso de índices espaciais. O mapeamento do diagrama da Figura 1 para MongoDB utilizando o GeoJSON como esquema de implementação é apresentado na Figura 2.

Existem diversas dificuldades quanto à representação das características geográficas no MongoDB. A primeira é a representação dos relacionamentos, pois a solução

Figura 2. Ruas e bairros na modelagem do GeoJSON

apresentada não especifica os relacionamentos espaciais entre as classes. As alternativas de solução incluem a utilização de pares chave-valor para esta representação e a utilização de vetores de subdocumentos. Com subdocumentos são três possibilidades: *(i)* segmentos de logradouro como subdocumento de cruzamento de rua (Cidade (Bairro (Cruzamento (Segmento)))); *(ii)* cruzamentos de vias como subdocumentos de segmentos de logradouro (Cidade (Bairro (Segmento (Cruzamento)))); *(iii)* todos os documentos em um mesmo nível, formando um documento único (Cidade, Bairro, Segmento, Cruzamento).

Tal diversidade provoca uma série de questionamentos, como por exemplo, determinar a melhor alternativa para atualização dos dados. A abordagem com todos os documentos aglomerados em um único documento (mesmo nível) provoca redundância dos dados. Desse modo, é necessário identificar se o tempo gasto para uma atualização de tais dados justificaria o uso de tal abordagem. Nota-se que este problema de padrões de projeto é conhecido no contexto de dados XML, no qual existem diversos formatos para os esquemas, que podem variar de acordo com o número dos seus elementos globais ou tipos (bonecas russas, fatia de salame, persianas, e o jardim do Eden) [9].

Outra dificuldade é a verificação das restrições de integridade espaciais, que em soluções SQL é realizada em *triggers*. No caso do MongoDB, verificar tais restrições requer implementar funções que utilizem as operações de log (*oplog*) para simular ações das tradicionais *triggers* de bancos de dados SQL.

Mapeamento para SGBD Orientado a Grafos. SGBDs orientados a grafos não possuem esquema, e dados são coleções de nós e arestas interligados em grafo. Cada nó representa uma entidade (ex., cidade ou bairro) e cada aresta uma ligação ou relação entre dois nós [4, 8]. Aqui, consideramos o gerenciador Neo4J para representar esta categoria, pois possui módulo para dados geográficos (Neo4J Spatial). A Figura 3 ilustra duas possibilidades para o armazenamento físico dos dados modelados em grafos: mapear diretamente cada classe para um nó (Figura 3a), e mapear as classes que representam a rede urbana com cruzamentos de rua como nós e segmentos de rua como arestas (Figura 3b).

Para verificar as restrições de integridade espaciais com o Neo4J, pode-se utilizar o *TransactionEventHandler*, similar às *triggers* dos SGBDs relacionais. Os relacionamentos espaciais podem ser representados através dos arcos que ligam os nós, como por exemplo a rede urbana formada por segmentos de rua e seus cruzamentos, conforme a Figura 1. Embora a representação em grafo apresente dificuldades no processamento de certos tipos de consultas (ex. vizinhos mais próximos), oferece melhor desempenho em tarefas de roteamento e conectividade em rede [11].

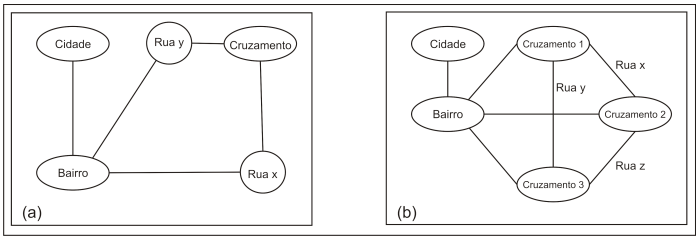

Figura 3. Ruas e bairros na modelagem em grafos

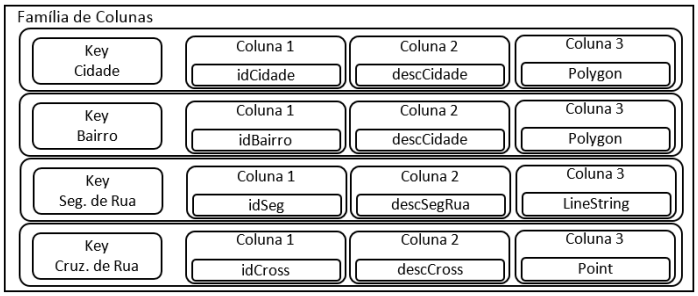

Figura 4. Ruas e bairros na modelagem em família de colunas

Mapeamento para SGBD Orientado a Colunas. Enquanto o modelo relacional define uma tabela como uma coleção de linhas, o modelo orientado a colunas (como o nome indica) organiza segundo uma coleção de colunas [12]. Aqui, consideramos o MonetDB como exemplo de gerenciador NoSQL orientado a colunas, pois este SGBD possui o *geom*, que é sua extensão para dados espaciais. A Figura 4 mapeia a Figura 1 onde cada classe corresponde a uma família de colunas, e cada uma dessas famílias possui uma chave única (*key*). Ainda é necessário usar chaves estrangeiras para representar o relacionamento entre as entidades. Outra possibilidade requer redundância dos dados, em que cada família de colunas seria uma subfamília de colunas da anterior, formando uma superfamília de colunas. O MonetDB suporta o uso de *triggers* para a verificação das restrições de integridade espaciais.

Discussão. Vários fatores da modelagem física de dados podem influenciar diretamente o desempenho dos sistemas gerenciadores NoSQL para aplicações geográficas [11]. Algumas dessas questões foram apresentadas e discutidas neste artigo, e resumidos pela Tabela 1. Considere $\surd$ para sim, $\pm$ para parcialmente e χ para não.

A variedade de soluções, estruturas e esquemas de implementação indicam que, em princípio, esses tipos de SGBD podem atuar de forma complementar. Enquanto, por exemplo, a consulta a estruturas em rede é mais eficiente se realizada em um SGBD orientado a grafos, e hierarquias territoriais são mais bem representadas em um SGBD orientado a documentos, a atualização de dados respeitando restrições de integridade espaciais

Tabela 1. Comparação entre gerenciadores NoSQL

	MongoDB	Neo4J	MonetDB
Trigger p/ RIE	χ	$\pm$	$\surd$
Relacionamentos	$\pm$	$\surd$	$\pm$
Redundância	$\surd$	χ	$\pm$
ACID	χ	$\surd$	$\surd$
SQL	χ	χ	$\surd$

(RIE) é provavelmente melhor executada em SGBD relacionais. Um estudo comparativo de desempenho foi apresentado por Santos et al. [11], confirmando o potencial para criação de esquemas de implementação híbridos, usando o melhor de cada alternativa.

4. Conclusões

Neste artigo, apresentamos as dificuldades de mapear dados geográficos para SGBDs NoSQL. É importante que as questões levantadas sejam avaliadas levando em consideração todos os tipos de modelagem física que os sistemas gerenciadores NoSQL utilizam (documentos, grafos, colunas, etc.). Novos estudos permitirão melhorar o mapeamento da modelagem conceitual para a modelagem física, unindo as características peculiares aos tipos de relacionamentos espaciais com os formatos que ofereçam melhor desempenho em consultas e atualizações de dados. Tais aspectos são os grandes gargalos de uma aplicação geográfica, bem como uma análise de parâmetros intrínsecos à aplicação a ser construída. Por exemplo, quais os tipos de dados mais frequentes, qual a carga de trabalho que será submetida ao SGBD e qual será a utilização mais frequente, entre consultas e atualizações. Desta forma, o objetivo será mapear um esquema conceitual para um modelo físico híbrido SQL/NoSQL de dados geográficos, que seja capaz de reunir as melhores características de cada paradigma para obter o máximo de desempenho em aplicações geográficas.

Agradecimentos. Trabalho parcialmente financiado por CAPES, CNPq e FAPEMIG.

Referências

[1] K. A. V. Borges, C. A. Davis Jr., and A. H. F. Laender. OMT-G: an object-oriented data model for geographic applications. *GeoInformatica*, 5(3):221–260, 2001.

[2] F. Bugiotti, L. Cabibbo, P. Atzeni, and R. Torlone. Database Design for NoSQL Systems. In *ER*, pages 223–231, 2014.

[3] S. Cockcroft. A taxonomy of spatial data integrity constraints. *GeoInformatica*, 1(4):327–343, 1997.

[4] H. Hashem and D. Ranc. An Integrative Modeling of Bigdata Processing. *International Journal of Computer Science and Applications*, 12(1):1–15, 2015.

[5] C. He. Survey on NoSQL Database Technology. *JASEI*, pages 50–54, 2015.

[6] A. C. Hora, C. A. Davis Jr, and M. M. Moro. Mapping Network Relationships from Spatial Database Schemas to GML Documents. *JIDM*, 2(1):67–74, 2011.

[7] K. Kaur and R. Rani. Modeling and querying data in NoSQL databases. In *Int'l Conference on Big Data*, pages 1–7, 2013.

[8] L. B. Marinho et al. Extracting geospatial preferences using relational neighbors. *JIDM*, pages 364–478, 2012.

[9] M. M. Moro, L. Lim, and Y.-C. Chang. Schema advisor for hybrid relational-XML DBMS. In *SIGMOD*, pages 959–970, 2007.

[10] P. J. Sadalage and M. Fowler. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education, 2012.

[11] P. O. Santos, M. M. Moro, and C. A. Davis Jr. Comparative Performance Evaluation of Relational and NoSQL Databases for Spatial and Mobile Applications. In *DEXA*, 2015.

[12] M. Saxena, Z. Ali, and V. K. Singh. NoSQL Databases-Analysis, Techniques, and Classification. *JoADMS*, 1(2):13–24, 2014.