Um Estudo sobre Arquiteturas e Metadados em Data Lakes

Jéssica Xafranski Rodrigues, Ronaldo dos Santos Mello

[1]Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC)
88.040-900 – Florianópolis – SC

jesicaxafranski@gmail.com, r.mello@ufsc.br

Abstract. *The large amount of data generated today through the Internet and by organizations requires new approaches to manage them. Data lakes are repositories of large data and have been an alternative to manage heterogeneous data and metadata. This work presents a study about architectures and metadata management approaches for data lake management systems.*

Resumo. *A grande quantidade de dados hoje gerada pela Internet e por organizações requer novas abordagens de gerenciamento desses dados. Data lakes são repositórios de dados volumosos e têm sido uma alternativa para gerenciar dados e metadados heterogêneos. Este trabalho apresenta um estudo sobre arquiteturas e abordagens para gerenciamento de metadados no contexto de sistemas de gerência de data lake.*

1. Introdução

Atualmente, grandes quantidades de dados heterogêneos são gerados todos os dias de diferentes formas, com diversas origens e destinos, como dispositivos eletrônicos conectados à Internet, aplicativos de computador e telefones celulares [Hashem et al. 2016]. Muitos desses dados auxiliam organizações em tomadas de decisão para a melhoria de seus processos de gestão. Entretanto, a maior parte desses dados, em seu formato original, requer processamentos custosos para extrair informações úteis, inclusive os seus metadados, que são necessários para caracterizar os dados [Ravat and Zhao 2019b].

Nesse contexto começam a ganhar espaço atualmente os chamados *Data Lakes (DL)*: repositórios de grandes volumes de dados variados que são mantidos em seus formatos brutos (estruturados ou não) para ser manipulados e transformados livremente conforme as necessidades de recuperação e analíticas da organização [Nargesian et al. 2019]. Entretanto, para o gerenciamento eficiente de um DL se faz necessária uma arquitetura bem planejada para um sistema de gerência de DL (SGDL), bem como um controle de metadados que facilite o acesso aos seus dados.

Assim sendo, este trabalho apresenta um panorama resumido das pesquisas atuais sobre arquitetura de um SGDL e gerenciamento de metadados em DLs com o objetivo de auxiliar interessados no assunto. Não foi encontrado na literatura nenhum estudo que analisa esses dois tópicos. Um processo de revisão sistemática considerando a *string* de busca *("metadata"OR "management" OR "architecture"OR "governance") AND "data lake"* foi realizado, resultando em 18 trabalhos selecionados entre os anos de 2017 e 2021. Estes trabalhos não são detalhados aqui por restrições de espaço.

Este artigo está organizado em mais 3 seções. A próxima seção sumariza as pesquisas em arquiteturas para SGDL, a seção 3 tem o mesmo objetivo para metadados, e a seção 4 é dedicada às considerações finais.

2. Arquitetura de um SGDL

Com relação a abordagens para arquitetura de um SGDL, 22,2% das propostas utilizam o modelo de *arquitetura de zonas* [Ravat and Zhao 2019a]. A primeira zona é a *Raw data zone*, cujo objetivo é realizar a ingestão de dados em seu formato original e armazená-los em um repositório sem que haja a necessidade prévia de um processamento. Na *Process zone* ocorre a transformação dos dados nativos para dados valorados, isto é, os usuários fazem uma análise prévia de dados que devem ser utilizados e definem requisitos de integridade e metadados para facilitar o acesso. Já a *Access zone* armazena todos os dados/metadados processados e os usuários podem acessá-los. Por fim, a *Governance zone* garante segurança, integridade e qualidade aos dados, dando suporte às zonas.

Em 27,8% dos trabalhos utiliza-se a arquitetura de *processos* [Hai et al. 2021]. O primeiro processo desta arquitetura é chamado de *Ingestion*, onde ocorre a ingestão, extração e modelagem dos dados. O segundo processo (*Maintenance*), organiza os metadados para facilitar posterior acesso aos dados. Neste processo também ocorre a busca por possíveis relações ocultas entre os dados através de tarefas de integração de dados. O último processo (*Exploration*) permite o acesso aos dados pré-processados. As consultas utilizam índices por palavras-chave ou valores de atributos.

Em 22,2% dos trabalhos não foi especificado o tipo de arquitetura utilizada. Eles apenas detalham alguns dos processos comentados nas arquiteturas propostas anteriormente. Por fim, 27,8% dos trabalhos abordam apenas processos de *classificação* dos dados para fins de geração de metadados que facilitem um melhor gerenciamento do DL.

A Tabela 1 apresenta as principais diferenças entre essas duas abordagens arquiteturais para SGDL. Na arquitetura de processos, todos os componentes possuem comunicação direta com a camada de armazenamento. Considera-se essa uma arquitetura de mais *baixo nível*, pois os processos estão mais distantes de políticas de gerenciamento e mais próximos da camada de armazenamento.

Tabela 1. Comparação entre arquiteturas para SGDL

Arquitetura de Zonas	Arquitetura de Processos
alto nível	baixo nível
processos complexos	processos mais simples
esquema maleável	esquema mais rígido

Já na arquitetura de zonas, a camada de armazenamento possui comunicação somente com a *Raw data zone*. Considera-se essa arquitetura de mais *alto nível*, pois ela possui uma relação mais forte com as políticas e a estrutura organizacional de governança. Nessa arquitetura os dados estão mais polidos para acesso, diferente da arquitetura de processos, onde os dados podem ser acessados em todos os níveis de tratamento, ou seja, desde o dado bruto até o dado totalmente pré-processado.

Os processos executados em uma arquitetura de zonas são processos muitas vezes complexos e com mais etapas. Os dados tendem a permanecer mais tempo dentro de

uma zona e sofrer todos os processos necessários para migrarem para a próxima zona. Já na arquitetura de processos temos processos menores e os dados tendem a passar menos tempo dentro de um processo para passar à próxima etapa.

A arquitetura de processos possui uma estrutura de dados mais rígida, uma vez que gera um esquema único e pronto para acesso pelos usuários. Ela tende a não sofrer muitas alterações em sua estrutura, sendo indicado para gerenciamento de metadados mais estáveis. Já a arquitetura de zonas possui relação mais direta com a camada de governança, podendo sofrer alterações em sua estrutura de acordo com as políticas a serem adotadas no gerenciamento de metadados. Desta forma, ela é mais maleável, sendo indicada para o gerenciamento de metadados que possuem mudanças constantes.

3. Metadados em DL

O gerenciamento de metadados em DL não apresenta um consenso na literatura. Mesmo assim, uma proposta atual em termos de classificação de metadados em DL, embasada em estudos anteriores, é a seguinte [Diamantini et al. 2021]:

- *Business*: regras de negócio, como limites mínimo e máximo de um atributo;
- *Operational*: informações geradas pelos processos de tratamento dos dados, como qualidade e proveniência;
- *Technical*: informações sobre o formato e esquema do dado.

Outro estudo bastante citado na literatura apresenta um modelo de objetos para metadados em DLs [Sawadogo et al. 2019]. Uma de suas principais contribuições são um conjunto de processos que apoiam a instanciação desse modelo: *(i) semantic enrichment*: gerar descrições de contexto dos dados, como *tags*; *(ii) indexing*: definir estruturas de indexação adequadas a cada natureza de dados (estruturado, semiestruturado e não-estruturado); *(iii) link generation and conservation*: detectar similaridades entre dados e definir relacionamentos entre eles; *(iv) polymorphism*: manter múltiplas representações de um mesmo dado; *(v) versioning*: suportar alterações nos metadados e conservar estados anteriores; *(vi) usage tracking*: registrar interações entre usuários e o DL. Outras propostas de modelos de metadados que seguem essa mesma linha são o *Google GOODS* [Halevy et al. 2016] e o *CoreKG* [Beheshti et al. 2018].

Outro ponto analisado são as tecnologias para armazenamento e/ou gerenciamento de metadados. A Tabela 2 sumariza esse ponto. A tecnologia mais citada para persistência de metadados, durante os processos de tratamento e manipulação, é o formato JSON. Diversos SG de bancos de dados (SGBDs) hoje já apresentam suporte a esse formato. BDs relacionais e a linguagem SQL também se sobressaem, por serem uma tecnologia madura, seguido do Spark, que é utilizado por muitos dos trabalhos como tecnologia que engloba toda a cadeia de processos de gerenciamento de dados e metadados. Além dessas, temos outras tecnologias usadas em menor escala, como é o caso de alguns SGBDs NoSQL, ou mesmo menções gerais ao uso de SGBDs, porém sem detalhar qual produto.

Por fim, menciona-se o tipo de armazenamento de metadados utilizado pelos trabalhos: *(i) nuvem*; *(ii) local*; ou *(iii) ambos*. Trabalhos que utilizam ambos (local e nuvem) são as dominantes (66,67%) seguido de soluções na nuvem (22,22%). Um percentual de 11,11% não menciona o tipo de armazenamento e nenhum trabalho utiliza armazenamento totalmente local, provavelmente devido a questões de desempenho.

Tabela 2. Tecnologias para armazenamento/gerenciamento de metadados em DL

Tecnologia	% Trabalhos	Tecnologia	% Trabalhos
JSON	12,5%	NoSQL	6,25%
SQL	9,38%	SGBD	6,25%
Spark	9,38%	Hadoop	4,69%
XML	7,81%	AWS	4,69%
HDFS	6,25%	MongoDB	4,69%
Neo4j	6,25%		

4. Considerações Finais

Este artigo apresenta tendências de pesquisa e desenvolvimento no que se refere a arquiteturas e metadados em DLs. Um ponto negativo da literatura atual é a carência de trabalhos que apresentem uma arquitetura detalhada em termos de algoritmos utilizados pelos processos ou zonas de um SGDL. O mesmo vale para processos de descoberta, extração, integração e catalogação de metadados. Mesmo assim, espera-se que este artigo seja um referencial inicial para os interessados no assunto.

Como trabalhos futuros, sugere-se uma avaliação das arquiteturas propostas segundo diferentes critérios, como desempenho e funcionalidade, bem como dos modelos de metadados em termos de abrangência de conceitos e adequação à Lei Geral de Proteção de Dados (LGPD).

Referências

Beheshti, A., Benatallah, B., Nouri, R., and Tabebordbar, A. (2018). CoreKG: A Knowledge Lake Service. *Proc. VLDB Endow.*, 11(12):1942–1945.

Diamantini, C. et al. (2021). An Approach to Extracting Topic-guided Views from the Sources of a Data Lake. *Inf. Syst. Frontiers*, 23(1):243–262.

Hai, R., Quix, C., and Jarke, M. (2021). Data Lake Concept and Systems: A Survey. *CoRR*, abs/2106.09592.

Halevy, A. Y. et al. (2016). Goods: Organizing Google's Datasets. In *International Conference on Management of Data*, pages 795–806. ACM.

Hashem, I. A. T. et al. (2016). The Role of Big Data in Smart City. *Int. J. Inf. Manag.*, 36(5):748–758.

Nargesian, F. et al. (2019). Data Lake Management: Challenges and Opportunities. *Proc. VLDB Endow.*, 12(12):1986–1989.

Ravat, F. and Zhao, Y. (2019a). Data Lakes: Trends and Perspectives. In *Database and Expert Systems Applications,*, volume 11706 of *LNCS*, pages 304–313. Springer.

Ravat, F. and Zhao, Y. (2019b). Metadata Management for Data Lakes. In *ADBIS Short Papers and Workshops*, volume 1064 of *Communications in Computer and Information Science*, pages 37–44. Springer.

Sawadogo, P. N., Kibata, T., and Darmont, J. (2019). Metadata Management for Textual Documents in Data Lakes. In *International Conference on Enterprise Information Systems*, pages 72–83. SciTePress.