Data Lake Management: Challenges and Opportunities

Fatemeh Nargesian
University of Toronto
fnargesian@cs.toronto.edu

Erkang Zhu
University of Toronto
ekzhu@cs.toronto.edu

Renée J. Miller
Northeastern University
miller@northeastern.edu

Ken Q. Pu
UOIT
ken.pu@uoit.ca

Patricia C. Arocena
TD Bank Group
prg@cs.toronto.edu

ABSTRACT

The ubiquity of data lakes has created fascinating new challenges for data management research. In this tutorial, we review the state-of-the-art in data management for data lakes. We consider how data lakes are introducing new problems including dataset discovery and how they are changing the requirements for classic problems including data extraction, data cleaning, data integration, data versioning, and metadata management.

PVLDB Reference Format:
Fatemeh Naregsian, Erkang Zhu, Renée J. Miller, Ken Q. Pu, Patricia C. Arocena. Data Lake Management: Challenges and Opportunities. *PVLDB*, 12(12): 1986-1989, 2019.
DOI: https://doi.org/10.14778/3352063.3352116

1. INTRODUCTION

A data lake is a massive collection of datasets that: (1) may be hosted in different storage systems; (2) may vary in their formats; (3) may not be accompanied by any useful metadata or may use different formats to describe their metadata; and (4) may change autonomously over time. Enterprises have embraced data lakes for a variety of reasons. First, data lakes decouple data producers (for example, operational systems) from data consumers (such as, reporting and predictive analytics systems). This is important, especially when the operational systems are legacy mainframes which may not even be owned by the enterprise (as is common in many enterprises such as banking and finance). For data science, data lakes provide a convenient storage layer for experimental data, both the input and output of data analysis and learning tasks. The creation and use of data can be done autonomously without coordination with other programs or analysts. But the shared storage of a data lake coupled with a (typically distributed) computational framework, provides the rudimentary infrastructure required for sharing and re-use of massive datasets.

Proceedings of the VLDB Endowment, Vol. 12, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3352063.3352116

While some of the data in a lake is extracted, transformed, and loaded into existing database management systems (DBMS) or data warehouses, some of it may be exclusively consumed on-demand by programming environments to perform specific data analysis tasks. Moreover, the value of some of this data is transient, meaning additional analysis is required to create information with sufficient value to load into a data warehouse. Even though some of this data is not destined for traditional DBMS, there are still many open and fascinating data management research problems.

Current data lakes provide reliable storage for datasets together with computational frameworks (such as Hadoop or Apache Spark), along with a suite of tools for doing data governance (including identity management, authentication and access control), data discovery, extraction, cleaning, and integration. These tools help individual teams, data owners and consumers alike, to create and use data in a data lake using a self-serve model. But many challenges remain. First, we are only at the beginning of being able to exploit the work of others (their search, extraction, cleaning, and integration effort) to help in new uses of a data lake. Systems like IBM's LabBook propose using the collective effort of data scientists to recommend new data visualization or analysis actions over new datasets or for new users [25]. Still challenges and opportunities remain in being able to *collectively* exploit how data lakes are used. Second, data lakes are currently mostly intermediate repositories for data. Currently, this data does not become actionable until it is cleaned and integrated into a traditional DBMS or warehouse. A grand challenge for data lake management systems is to support on-demand query answering meaning data discovery, extraction, cleaning, and integration done at query time over massive collections of datasets that may have unknown structure, content, and data quality. Only then would the data in data lakes become actionable.

2. DATA LAKE ARCHITECTURE

Figure 1 shows a high-level view of a common data lake. The data sources may include legacy operational systems (operating in Cobol or other formats), information scrapped from the Web and social media, or information from for-profit data brokers (such as Thompson Reuters or Lexis-Nexis). Operational systems often export all data as strings to avoid having to deal with type mismatches. The actual type information and metadata may be represented in numerous different formats. Other data may be pure documents, semi-structured logs, or social media information.

Data lakes vary in their ability to support a unified view over all or portions of the lake.

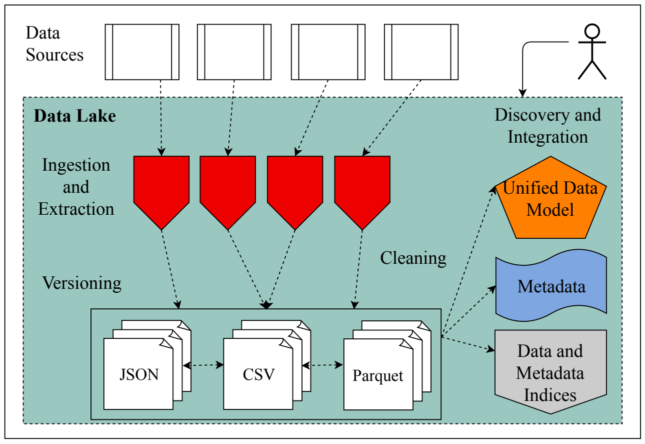

Figure 1: Example Data Lake Management System.

3. CHALLENGES AND OPPORTUNITIES

Our tutorial will be focused around the following main topics. For each, we will discuss the state-of-the-art and also present a vision for open problems that need to be studied.

3.1 Data Ingestion

Modern data lakes support data ingestion for a large variety of systems. Examples of ingestion are Web crawlers that create data files containing web pages and Open Data crawlers that archive Open Data repositories using Open Data endpoints. Enterprises usually develop proprietary software to handle end-to-end data ingestion.

The main task at this stage is bookkeeping of files for versioning and indexing purposes. Since ingestion often needs to interface with external data sources with limited bandwidth, it needs to be done with high degree of parallelism and low latency. This means that ingestion does not perform any deep analysis of the downloaded data. However, it is still possible to apply shallow data sketches on the downloaded contents and its metadata (if available) to maintain a basic organization of the ingested datasets. Simple data sketches (such as checksums) can also be used for duplicate detection and multi-versioning of evolving datasets. Open challenges in data ingestion are to support real-time ingestion of high velocity data with more sophisticated indexing to make this data more immediately available for analysis.

3.2 Data Extraction

Data ingestion creates raw datasets in a specific data format (e.g. a textual or binary encodings). Data extraction is the task of transforming this raw data to a pre-determined data model. This abstraction from raw data to a data model may be intertwined with preparation for tasks such as discovery, integration, and cleaning. For example, CLAMS unifies heterogeneous lake data into RDF for cleaning purposes [14]. Table extraction allows the abstraction of data into attributes (sets of values) that can be indexed for efficient data discovery [43].

An example of a current extractor is DeepDive, which extracts relational data from the lakes of text, tables, and images by relying on a user-defined schema and a small number of rules [41]. The automatic extraction of tables from web pages has been very well studied for over 10 years [11, 19, 27]. An example of large scale table extraction is the Google Web Table project which combines hand-written heuristics and statistically-trained classifiers to detect relational tables among HTML tables and assigns synthetic headers when it is required [5]. Another project leverages the principles of table construction to extract data tables with more complex structures such as tables that contain group headers [1]. The recent DATAMARAN project extracts relational data from semi-structured log files [18]. Table extraction from adhoc spreadsheets remains a challenge [7]. Contributions to extraction from the programming language community include PADS, a declarative data description language, together with a compiler and tools for parsing and extracting data from files [17, 45].

Data extraction is very well studied, yet opportunities remain for advancement. Today, extraction is done typically one file at a time. We are not yet taking full advantage of the "wisdom of the lake" to fully apply knowledge learned from previous extractions (and from humans in the extraction loop) to future extractions.

3.3 Data Cleaning

While cleaning enterprise data has received significant attention over the years, little work has been done on cleaning within the context of a data lake. Logical and relational data cleaning typically requires correct schema information including integrity constraints [39]. However, in data lakes the data may be stored in schema-less heterogeneous formats or using schemas that are only specified at application level. Although enriching data with schema information is one of the main goals of metadata management systems, postponing cleaning to the later stages of data processing may result in the propagation of errors through operations such as discovery and integration. CLAMS is an early approach that explicitly addresses the problem of cleaning raw heterogeneous lake data [14]. It enforces quality constraints on the data right after ingestion and extraction. CLAMS loads heterogeneous raw data sources in a unified data model and enforces quality constraints on this model. An interesting opportunity in lake data cleaning is leveraging the lake's wisdom and performing collective data cleaning. Furthermore, since data lake operations such as extraction can themselves introduce systematic errors to the lake, it is important to investigate the underlying conditions and operations that cause these errors [38].

3.4 Dataset Discovery

Due to the sheer size of data in data lakes and the absence or incompleteness of a comprehensive schema or data catalog, data discovery has become an important problem in data lakes. To address the data discovery problem, some solutions focus on generating and enriching data catalogs as well as facilitating search on them. We consider these below with other data lake metadata management techniques. Other solutions operate on raw data (and existing metadata) to perform discovery [9, 29, 43]. In query-driven discovery, a user starts a search with a query (dataset or keywords) and the goal is to find similar datasets to the query [4, 6] or datasets that can be integrated (joined or unioned) with the query [31, 43]. This is achieved by defining measures and constructing efficient index structures that are specialized for the unique characteristics of data lakes. Navigation (or exploration) is an alternative to search. Data discovery can be done by allowing a user to navigate over a linkage graph [44, 42, 16, 15] or a hierarchical structure created to facilitate exploration of the lake [30]). An interesting direction in discovery is analysis-driven discovery which is the problem of augmenting a dataset with relevant data (new training samples and features) with the purpose of performing learning tasks.

3.5 Metadata Management

Unlike data warehouses or DBMS, data lakes may not be accompanied with descriptive and complete data catalogs. Without explicit metadata information, a data lake can easily become a data swamp. Data catalogs are essential to on-demand discovery and integration in data lakes as well as raw data cleaning. In addition to extracting metadata from sources and enriching data with meaningful metadata (such as detailed data description and integrity constraints), metadata management systems need to support efficient storage of metadata (specially when it becomes large) and query answering over metadata.

An example of a metadata management system is Google Dataset Search (GOODS) that extracts and collects metadata for datasets generated and used internally by Google [21]. The collected metadata ranges from dataset-specific information such as owners, timestamps, and schema to relationships among multiple datasets such as their similarity and provenance. GOODS makes datasets accessible and searchable by exposing their collected metadata in dataset profiles. Constance is another example that in addition to extracting metadata from sources enriches data sources by annotating data and metadata with semantic information [20]. Constance makes the generated metadata accessible in a template-based query answering environment. In contrast, the Ground project collects the context of data which includes applications, behaviors, and changes of data and stores the metadata in queryable graph structures [23]. Skluma extracts deeply embedded metadata, latent topics, and contextual metadata from files in various formats in a data lake [36] and allows topic-based discovery [36].

Metadata discovery provides the data abstraction that is crucial to data understanding and discovery, yet opportunities remain in better extracting and connecting knowledge from lakes and incorporating this knowledge into existing (general or domain-specific) knowledge bases.

3.6 Data Integration

Traditional paradigms for integration, including data federation [22] and data exchange [13], have at best limited value in data lakes. We will survey some Big Data Integration techniques that tackle dynamic data which may be of very poor data quality [10] to consider how they apply to data lakes. These techniques differ from pay-as-you-go data integration that automatically construct a mediated schema from various sources [8]. We will discuss the requirements of on-demand integration, that is the task of integrating raw data from a data lake at query time. The challenge of on-demand integration lies in first finding datasets that contain relevant data, and then integrating them in a meaningful way. Relevant data may be modeled as data that *augments* known entities with new attributes or properties, as done in Infogather [40]. Alternatively, relevant data may be a schema that is described by keyword queries expressed over attribute names or other metadata [32, 34]. Schema mapping permits the exchange of information between data sets using different schemas [12, 37] and recent work permits mapping discovery over incomplete (or inconsistent) schemas and examples [26]. In sample-driven schema mapping, users describe the schema using a set of tuples [33, 35]. To give users flexibility in describing a schema, in multi-resolution schema mapping, the user can describe schemas using a set of constraints of various resolutions, such as incomplete tuples, value ranges, and data types [24]. Nonetheless, on-demand schema mapping remains a grand challenge. Importantely, discovery and integration are intertwined operations in data lakes [34, 43, 31]. A new paradigm, called query-driven discovery, finds tables that join or union with a query table [43, 31]. Most of these solutions perform integration on relational data. However, to achieve on-demand data integration on data lakes, we must be able to manage the heterogeneity of lakes and potentially perform on-demand extraction and cleaning as part of integration.

3.7 Dataset Versioning

Data lakes are dynamic. New datasets and new versions of existing files enter the lake at the ingestion stage. Additionally, extractors can evolve over time and generate new versions of raw data. As a result, data lake versioning is a cross-cutting concern over all stages of a data lake. Of course vanilla distributed file systems are not adequate for versioning-related operations. For example, simply storing all versions may be too costly for large datasets, and without a good version manager, just using filenames to track versions can be error-prone. In a data lake where there are usually many users, it is even more important to clearly maintain correct versions and versioning information. Furthermore, as the number of versions increases, efficiently and cost-effectively providing storage and retrieval of versions is going to be an important feature of a successful data lake system. One early approach DataHub provides a git-like interface by supporting operations such as version creation, branching, merging, and viewing difference between datasets [2, 3]. Open challenges include managing schema evolution and the peculiarity of data formats between versions and detection of versions.

4. OUTLINE AND SCOPE

We will focus on the challenges and open problems in data lake management and the state-of-the-art techniques in the areas we described in Section 3. The tutorial is designed for data management and data science audience.

5. BIO SKETCHES

For five years, Nargesian, Pu, Zhu, and Miller have been studying data lakes and developing new data discovery paradigms. Their results were featured in a VLDB 2018 keynote [28]. Arocena is a Big Data practitioner with a PhD in Data Integration and several years of industry experience with data lakes. She currently works at TD Bank.

6. REFERENCES

[1] M. D. Adelfio and H. Samet. Schema extraction for tabular data on the web. *PVLDB*, 6(6):421–432, 2013.

[2] A. P. Bhardwaj, S. Bhattacherjee, A. Chavan, A. Deshpande, A. J. Elmore, S. Madden, and A. G.

Parameswaran. DataHub: Collaborative data science & dataset version management at scale. In *CIDR*, 2015.

[3] S. Bhattacherjee, A. Chavan, S. Huang, A. Deshpande, and A. Parameswaran. Principles of dataset versioning: Exploring the recreation/storage tradeoff. *PVLDB*, 8(12):1346–1357, 2015.

[4] W. Brackenbury, R. Liu, M. Mondal, A. J. Elmore, B. Ur, K. Chard, and M. J. Franklin. Draining the data swamp: A similarity-based approach. HILDA, pages 13:1–13:7, 2018.

[5] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. Webtables: Exploring the power of tables on the web. *PVLDB*, 1(1):538–549, 2008.

[6] M. J. Cafarella, A. Y. Halevy, and N. Khoussainova. Data integration for the relational web. *PVLDB*, 2(1):1090–1101, 2009.

[7] Z. Chen, S. Dadiomov, R. Wesley, G. Xiao, D. Cory, M. J. Cafarella, and J. Mackinlay. Spreadsheet property detection with rule-assisted active learning. In *CIKM*, pages 999–1008, 2017.

[8] A. Das Sarma, X. Dong, and A. Halevy. Bootstrapping pay-as-you-go data integration systems. In *SIGMOD*, pages 861–874, 2008.

[9] D. Deng, R. C. Fernandez, Z. Abedjan, S. Wang, M. Stonebraker, A. K. Elmagarmid, I. F. Ilyas, S. Madden, M. Ouzzani, and N. Tang. The data civilizer system. In *CIDR*, 2017.

[10] X. L. Dong and D. Srivastava. *Big Data Integration*. Synthesis Lectures on Data Management. 2015.

[11] J. Eberius, K. Braunschweig, M. Hentsch, M. Thiele, A. Ahmadov, and W. Lehner. Building the dresden web table corpus: A classification approach. In *Symposium on Big Data Computing*, pages 41–50, 2015.

[12] R. Fagin, L. M. Haas, M. A. Hernández, R. J. Miller, L. Popa, and Y. Velegrakis. Clio: Schema mapping creation and data exchange. In *Conceptual Modeling: Foundations and Applications - Essays in Honor of John Mylopoulos*, pages 198–236, 2009.

[13] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. *Theory of Computer Science*, 336(1):89–124, 2005.

[14] M. H. Farid, A. Roatis, I. F. Ilyas, H. Hoffmann, and X. Chu. CLAMS: bringing quality to data lakes. In *SIGMOD*, pages 2089–2092, 2016.

[15] R. C. Fernandez, Z. Abedjan, F. Koko, G. Yuan, S. Madden, and M. Stonebraker. Aurum: A data discovery system. In *ICDE*, pages 1001–1012, 2018.

[16] R. C. Fernandez, E. Mansour, A. A. Qahtan, A. K. Elmagarmid, I. F. Ilyas, S. Madden, M. Ouzzani, M. Stonebraker, and N. Tang. Seeping semantics: Linking datasets using word embeddings for data discovery. In *ICDE*, pages 989–1000, 2018.

[17] K. Fisher and D. Walker. The PADS project: an overview. In *ICDT*, pages 11–17, 2011.

[18] Y. Gao, S. Huang, and A. Parameswaran. Navigating the data lake with datamaran: Automatically extracting structure from log datasets. In *SIGMOD*, pages 943–958, 2018.

[19] W. Gatterbauer and P. Bohunsky. Table extraction using spatial reasoning on the CSS2 visual box model. In *AAAI*, pages 1313–1318, 2006.

[20] R. Hai, S. Geisler, and C. Quix. Constance: An intelligent data lake system. In *SIGMOD*, pages 2097–2100, 2016.

[21] A. Halevy, F. Korn, N. F. Noy, C. Olston, N. Polyzotis, S. Roy, and S. E. Whang. Goods: Organizing google's datasets. In *SIGMOD*, pages 795–806, 2016.

[22] D. Heimbigner and D. McLeod. A federated architecture for information management. *ACM Trans. Inf. Syst.*, 3(3):253–278, 1985.

[23] J. M. Hellerstein, V. Sreekanti, J. E. Gonzalez, J. Dalton, A. Dey, S. Nag, K. Ramachandran, S. Arora, A. Bhattacharyya, S. Das, M. Donsky, G. Fierro, C. She, C. Steinbach, V. Subramanian, and E. Sun. Ground: A data context service. In *CIDR*, 2017.

[24] Z. Jin, C. Baik, M. Cafarella, and H. V. Jagadish. Beaver: Towards a declarative schema mapping. In *HILDA*, pages 10:1–10:4, 2018.

[25] E. Kandogan, M. Roth, P. M. Schwarz, J. Hui, I. G. Terrizzano, C. Christodoulakis, and R. J. Miller. LabBook: Metadata-driven social collaborative data analysis. In *IEEE Big Data*, pages 431–440, 2015.

[26] A. Kimmig, A. Memory, R. J. Miller, and L. Getoor. A collective, probabilistic approach to schema mapping. In *ICDE*, pages 921–932, 2017.

[27] O. Lehmberg, D. Ritze, R. Meusel, and C. Bizer. A large public corpus of web tables containing time and context metadata. In *WWW*, pages 75–76, 2016.

[28] R. J. Miller. Open data integration. *PVLDB*, 11(12):2130–2139, 2018.

[29] R. J. Miller, F. Nargesian, E. Zhu, C. Christodoulakis, K. Q. Pu, and P. Andritsos. Making open data transparent: Data discovery on open data. *IEEE Data Eng. Bull.*, 41(2):59–70, 2018.

[30] F. Nargesian, K. Q. Pu, E. Zhu, B. G. Bashardoost, and R. J. Miller. Optimizing organizations for navigating data lakes, 2018. arXiv:1812.07024.

[31] F. Nargesian, E. Zhu, K. Q. Pu, and R. J. Miller. Table union search on open data. *PVLDB*, 11(7):813–825, 2018.

[32] R. Pimplikar and S. Sarawagi. Answering table queries on the web using column keywords. *PVLDB*, 5(10):908–919, 2012.

[33] L. Qian, M. J. Cafarella, and H. V. Jagadish. Sample-driven schema mapping. In *SIGMOD*, pages 73–84, 2012.

[34] A. D. Sarma, L. Fang, N. Gupta, A. Y. Halevy, H. Lee, F. Wu, R. Xin, and C. Yu. Finding related tables. In *SIGMOD*, pages 817–828, 2012.

[35] Y. Shen, K. Chakrabarti, S. Chaudhuri, B. Ding, and L. Novik. Discovering queries based on example tuples. In *SIGMOD*, pages 493–504, 2014.

[36] T. J. Skluzacek, R. Kumar, R. Chard, G. Harrison, P. Beckman, K. Chard, and I. T. Foster. Skluma: An extensible metadata extraction pipeline for disorganized data. In *IEEE International Conference on e-Science*, pages 256–266, 2018.

[37] B. ten Cate, P. G. Kolaitis, and W. C. Tan. Schema mappings and data examples. In *EDBT*, pages 777–780, 2013.

[38] X. Wang, M. Feng, Y. Wang, X. L. Dong, and A. Meliou. Error diagnosis and data profiling with data x-ray. *PVLDB*, 8(12):1984–1987, 2015.

[39] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. *PVLDB*, 4(5):279–289, 2011.

[40] M. Yakout, K. Ganjam, K. Chakrabarti, and S. Chaudhuri. Infogather: Entity augmentation and attribute discovery by holistic matching with web tables. In *SIGMOD*, pages 97–108, 2012.

[41] C. Zhang, J. Shin, C. Ré, M. J. Cafarella, and F. Niu. Extracting databases from dark data with deepdive. In *SIGMOD*, pages 847–859, 2016.

[42] E. Zhu, D. Deng, F. Nargesian, and M. R. J. Josie: Overlap set similarity search for finding joinable tables in data lakes. In *SIGMOD*, 2019. To appear.

[43] E. Zhu, F. Nargesian, K. Q. Pu, and R. J. Miller. LSH ensemble: Internet-scale domain search. *PVLDB*, 9(12):1185–1196, 2016.

[44] E. Zhu, K. Q. Pu, F. Nargesian, and R. J. Miller. Interactive navigation of open data linkages. *PVLDB*, 10(12):1837–1840, 2017.

[45] K. Q. Zhu, K. Fisher, and D. Walker. Learnpads++ : Incremental inference of ad hoc data formats. In *PADL*, pages 168–182, 2012.