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

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, and data versioning. The tutorial will cover enterprise data lakes and data lakes that are being used to support data science. Our focus will be on the exciting new open research challenges that data lakes are inspiring.

PVLDB Reference Format:
Fatemeh Naregsian, Erkang Zhu, Renée J. Miller. Data Lake Management: Challenges and Opportunities. *PVLDB*, 12(xxx): xxxx-yyyy, 2019.
DOI: https://doi.org/10.14778/xxxxxxx.xxxxxxx

1. INTRODUCTION

A data lake is a massive collection of raw files that: (1) may be hosted in different, typically distributed, 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.

While some of the data in a lake is important for data analysis and will be extracted, transformed, and loaded into existing DBMS or data warehouses, some of it may be exclusively consumed 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 data management systems, there are still many open and fascinating data management research problems.

Current data lakes provide reliable storage for datasets together with computational framework (such as Hadoop or Spark), along with a suite of tools for doing data governance (including identity management, authentication and access controls), 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 (the search, extraction, cleaning and integration effort of others) to help in new uses of a data lake. Systems like IBM's LabBook propose using the collective effort of others to recommend new data visualization or analysis actions over new data sets or for new users [23]. 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 collection of files that may have unknown structure, content, and data quality. Only then would the data in data lakes become actionable.

Next, we outline some of the challenges and the state-of-the-art solutions.

2. DATA LAKE ARCHITECTURE

Figure 1 shows a very high-level view of a common data lake. The data sources may include legacy operational systems (operating in Cobol or other older formats), informa-

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

tion scrapped from the Web and social media, for profit data brokers (such as Thompson Reuters or LexisNexis). 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 (unstructured data) or semi-structured logs, or social media information.

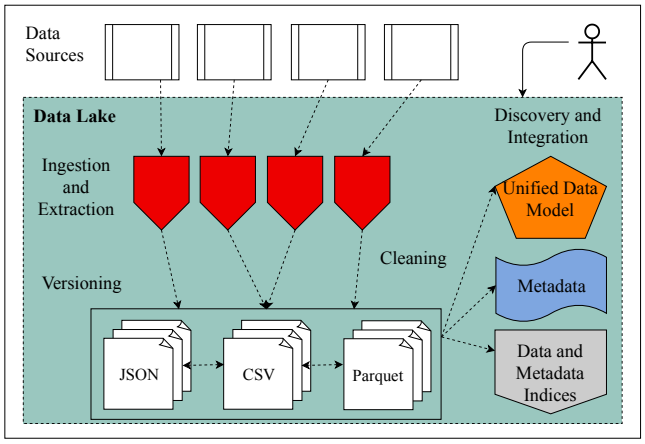

Figure 1: The Architecture of a Data Lake Management System.

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 webpages and Open Data crawlers that archive Open Data repositories using application specific API endpoints (e.g., the CKAN API[1]). Enterprises have their own ingestion pipelines from their operational systems.

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 data files. 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 [13]. Table extraction allows the abstraction of data into attributes (sets of values) that can be indexed for efficient data discovery [38].

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 [36]. Another example is the automatic extraction of tables from web pages (a topic that has been very well studied for over 15 years [18]). This entails parsing web pages, determining whether an HTML table contains relational data, and detecting attribute values and tuple boundaries. Extraction may also try to detect metadata such as table and attribute names. We discuss metadata recovery in more detail in Section 3.5. 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]. Other projects include Web Data Commons [24] and Dresden Web Table Corpus [11]. Another project leverages the principles of table construction to extract data tables with more complex structures such as tables that contains group headers [1]. The recent DATAMARAN project extracts relational data from semi-structured log files [17]. Table extraction from adhoc spreadsheets remains a challenge [7].

Contributions to extraction from the programming language community include PADS, a declarative data description language, a compiler and tools for parsing data files and transforming tuples and attributes to data types and data structures in a programming language [16, 40].

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 extraction.

3.3 Data Cleaning

Data cleaning is very well studied for enterprise data, but 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 [34]. However, in data lakes the data is stored in schema-less heterogeneous files and schemas 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 results in the propagation of error through operations such as discovery and integration. CLAMS is an early approach that explicitly addresses the problem of cleaning raw heterogeneous lake data [13]. 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 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 [33].

[1]https://ckan.org

3.4 Dataset Discovery

Due to the sheer size of data in data lakes and the absence of a meaningful 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 cover these in detail with the lake metadata management techniques. Others operate on raw data (and existing metadata) to perform discovery [25, 9]. In query-driven discovery, a user starts search with a query (dataset or keywords) and the goal is to find similar datasets to the query [6, 4] or datasets that can be integrated (joined or unioned) with the query [27, 38]. This is achieved by defining measures and constructing efficient index structures that are specialized for the unique characteristics of data lakes. On the other hand, in data-driven discovery, a user navigates a data lake (represented as a linkage graph [39, 37, 15, 14] or a hierarchical structure [26]) to find data of her interest. 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 are not accompanied with descriptive and complete data catalogs. Without explicit metadata associated with datasets, 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 data 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 large scale metadata management system is Google Dataset Search (GOODS) that extracts and collects metadata for datasets generated and used internally by Google as they are created, used, and updated [20]. The collected metadata ranges from dataset-specific information such as owners, timestamps, schema to relationships among multiple datasets such as their similarity and provenance. GOODS makes datasets accessible and searchable to users by exposing their collected metadata in dataset profiles. Constance is another example that in addition to extracting metadata from data sources enriches data sources by annotating data and metadata with semantic information [19]. Constance makes the generated metadata accessible to users in a template-based query answering environment. Ground thinks differently about metadata. It collects the context of data which includes applications, behaviors, and changes of data and stores the metadata in queryable graph structures [22]. Skluma extracts deeply embedded metadata, latent topics, and contextual metadata for files of various formats in a data lake [32] and allows topic-based discovery [32].

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

3.6 Data Integration

Traditional paradigms for integration, including data federation [21] and data exchange [12], 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 which automatically constructs 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 finding datasets that contain relevant data 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 [35]. Alternatively, relevant data may be described by keyword queries expressed over attribute names or other metadata [29, 31]. A new paradigm, called query-driven discovery, finds tables that join or union with a query table [38, 27]. Importantly, discovery and integration are intertwined operations in data lakes [31, 38, 27]. 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 as part of integration.

3.7 Dataset Versioning

Data lakes are dynamic. New files 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 across 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, for which there are usually many users, it is even more important to clearly maintain correct versions being used and evolving across different users. 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. DataHub is a large-scale dataset version control. It provides a git-like interface by supporting operations such as version creation, branching, merging, and viewing difference between datasets [2, 3]. Still data lake versioning systems need to deal with schema evolution.

4. OUTLINE AND SCOPE

We plan a three hour tutorial. 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, along with data governance. We will ground this discussion with an overview of the capabilities of current commercial data lake products and research systems [28, 30]. 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 Open Data Lakes and developing new data discovery paradigms. Their results were featured in Miller's VLDB 2018 keynote on Open Data Integration.

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

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

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

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

[20] 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.

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

[22] 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.

[23] 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 *BigData*, pages 431–440. IEEE ComputerSociety, 2015.

[24] 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.

[25] 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.

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

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

[28] S. R. Patil, W. G. Gong, and L. Coyne. Hortonworks data platform with ibm spectrum scale. Technical report, IBM, 2017.

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

[30] R. Ramakrishnan, B. Sridharan, J. R. Douceur, P. Kasturi, B. Krishnamachari-Sampath, K. Krishnamoorthy, P. Li, M. Manu, S. Michaylov, R. Ramos, N. Sharman, Z. Xu, Y. Barakat, C. Douglas, R. Draves, S. S. Naidu, S. Shastry, A. Sikaria, S. Sun, and R. Venkatesan. Azure data lake store: A hyperscale distributed file service for big data analytics. In *SIGMOD*, pages 51–63, 2017.

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

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

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

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