Data Lake

Based on Data lake concept and systems: a survey by Rihan et al

Rokan Uddin Faruqui

Associate Professor
Dept of Computer Science and Engineering
University of Chittaong, Bangladesh
Email: *rokan@cu.ac.bd*

Outline

1. Data Lake

2. Use cases of data lakes

3. DL Architecture

Outline

1. Data Lake
2. Use cases of data lakes
3. DL Architecture

Data Lake

A data lake is

- a flexible, scalable data storage and management system
- ingests and stores raw data from heterogeneous sources in their original format
- provides query processing and data analytics in an on-the-fly manner.

Data lakes store raw data.

A data lake ingests raw data in its original format from heterogeneous data sources, fulfills its role as a storage repository, and allows users to query and explore the data.

The ingestion of raw data may lead to the absence of schema information, constraints, and mappings, which are not defined explicitly or required initially for a data lake.

Without any metadata, the data lake is hardly usable as the structure and semantics of the data are not known, which turns a data lake quickly into a 'data swamp'.

A data lake is not only a storage system.

The primary role of a data lake is a data repository, which can be a centralized repository or a set of distributed repositories.

It provides a set of functions to manage and govern the data such that it can be used later

In the existing literature the term ***data lake*** *is used in both cases: solely the storage layer (i.e., Hadoop or certain databases), or a system providing storage and further services (e.g., metadata management).*

Data lakes support on-demand data processing and querying.

*One important feature of data lakes is **on-the-fly**, or **on-demand**, which indicates that schema definition, data integration, or indexing should be done only if necessary at the time of data access.*

In data lakes, the ingestion of data sources could be light weight, as it is not necessary to force schema definitions and mappings beforehand.

The metadata in data lakes matures incrementally

Data Warehouse vs. Data Lakes

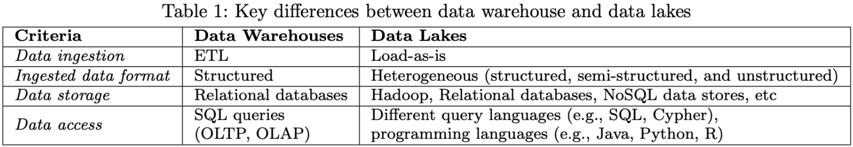

Table 1: Key differences between data warehouse and data lakes

Criteria	**Data Warehouses**	**Data Lakes**
Data ingestion	ETL	Load-as-is
Ingested data format	Structured	Heterogeneous (structured, semi-structured, and unstructured)
Data storage	Relational databases	Hadoop, Relational databases, NoSQL data stores, etc
Data access	SQL queries (OLTP, OLAP)	Different query languages (e.g., SQL, Cypher), programming languages (e.g., Java, Python, R)

Data Warehouse vs. Data Lakes

A data warehouse (DW) manages, integrates, and aggregates data from multiple sources, and provides data analytics for decision making via multi-dimensional data cubes in data marts.

In data warehouses, heterogeneous source datasets need to first go through the Extract/transform/load (ETL) process

DW data is stored and handled as structured data in relational databases or cube structures.

Data Warehouse vs. Data Lakes

In data lakes the transformation step is delayed and data is loaded in its original structure (i.e., load-as-is) to reduce upfront cleaning and integration effort and to make full source data available for later data analysis (i.e., pay-as-you-go).

Data lakes aim to process more heterogeneous sources including semi-structured and unstructured sources, and to manage these different data models efficiently using multiple dedicated kinds of storage

DW usually applies (multi-dimensional extensions of) SQL queries, while a data lake may need to support different query languages (e.g., SQL, Cypher11, JSONiq12), and programming languages (e.g., Python, Java, R)

Data lakes and dataspaces

The early dataspace prototypes tackled the problems of how to organize, integrate, discover, and query the data from several loosely interrelated data sources.

The dataspace concept can be considered a complement to the data lake approach supporting sovereign inter-organizational cooperation.

Outline

1. Data Lake
2. Use cases of data lakes
3. DL Architecture

Multiple heterogeneous raw data inputs

Alice is a scientist who studies the milling process, and she has two problems with the data.

First, vast and heterogeneous data is generated, which she needs to store and organize. Such production data includes binary image files from cameras, CSV and JSON files from different sensors, and ontologies enriched with her own annotation for the milling process. She hopes to store the heterogeneous data in raw formats, as transformation would be time-consuming, and might lose certain information and jeopardize her future research.

Second, instead of browsing through massive, diverse data files, she hopes to have a data management system, which provides her easy access to the raw data. A data lake would be a good solution to solve Alice's problems.

Use Case:

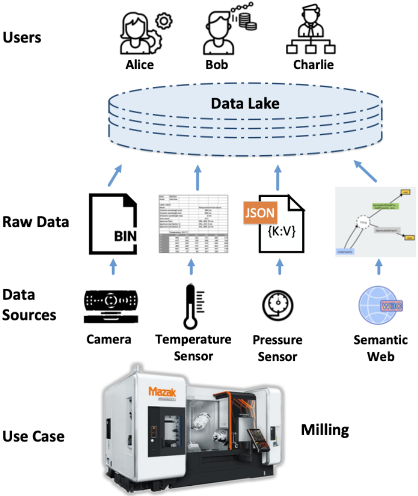

Fig. 1: Use case example: data lakes for manufacturing

Data integration and transformation

Bob is an industrial consultant, who has similar production data but in different formats, structures, and terminologies from Alice. Charlie is also a milling scientist but with a different research goal. He has another set of milling machines, sensors, and engineering models. For data analysis, he uses machine learning (ML) and runs Python scripts instead of merely queries.

Both Bob and Charlie want to integrate and compare their data and results with Alice. The three users hope to have a data lake, which can combine these independent data sources, help them easily find the datasets relevant for their own use cases, transform the data flexibly, and provide query answering and data analytics.

On-demand data processing and querying

As a researcher, Alice has her research questions and solutions evolving with time. She may introduce a new sensor, which has different machine-generated data formats compared to the existing datasets in the data lake. She may have new queries or want to use the raw data in a different manner.

In addition, in the beginning the usage of some raw datasets was unclear, which she just stored without using. Now she has created an engineering model for the unused data. Similar dynamic analytical requirements also apply to the use cases of Bob and Charlie. Therefore, they would like to have a data lake, which supports data processing, integration, and querying in an on-the-fly manner rather than being fixed from the beginning.

Outline

1. Data Lake
2. Use cases of data lakes
3. DL Architecture

DL Architecture

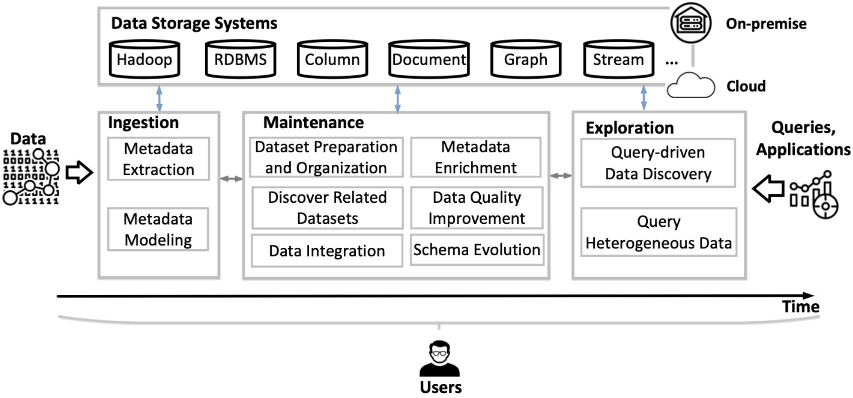

Fig. 2: Proposed architecture for existing DL solution categorization

Review this article : *P. Sawadogo and J. Darmont. On data lake architectures and metadata management. Journal of Intelligent Information Systems, pages 1–24, 2020.*