Lakehouse by Databricks

Wenqing Wei (wwei73)

Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics

Michael Armbrust[1], Ali Ghodsi[1,2], Reynold Xin[1], Matei Zaharia[1,3]
[1]Databricks, [2]UC Berkeley, [3]Stanford University

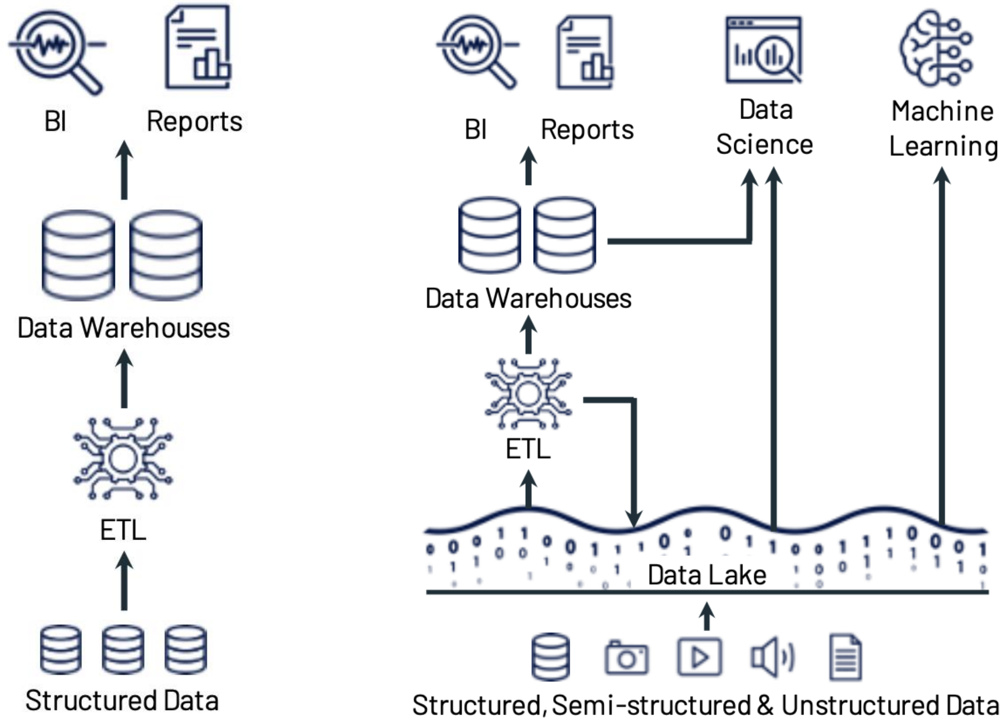
Abstract

This paper argues that the data warehouse architecture as we know it today will wither in the coming years and be replaced by a new architectural pattern, the Lakehouse, which will (i) be based on open direct-access data formats, such as Apache Parquet, (ii) have first-class support for machine learning and data science, and (iii) offer state-of-the-art performance. Lakehouses can help address several major challenges with data warehouses, including data staleness, reliability, total cost of ownership, data lock-in, and limited use-case support. We discuss how the industry is already moving toward Lakehouses and how this shift may affect work in data management. We also report results from a Lakehouse system using Parquet that is competitive with popular cloud data warehouses on TPC-DS.

data at low cost, but on the other hand, punted the problem of data quality and governance downstream. In this architecture, a small subset of data in the lake would later be ETLed to a downstream data warehouse (such as Teradata) for the most important decision support and BI applications. The use of open formats also made data lake data directly accessible to a wide range of other analytics engines, such as machine learning systems [30, 37, 42].

From 2015 onwards, cloud data lakes, such as S3, ADLS and GCS, started replacing HDFS. They have superior durability (often >10 nines), geo-replication, and most importantly, extremely low cost with the possibility of automatic, even cheaper, archival storage, e.g., AWS Glacier. The rest of the architecture is largely the same in the cloud as in the second generation systems, with a downstream data warehouse such as Redshift or Snowflake. This two-tier data lake + warehouse architecture is now dominant in the industry in

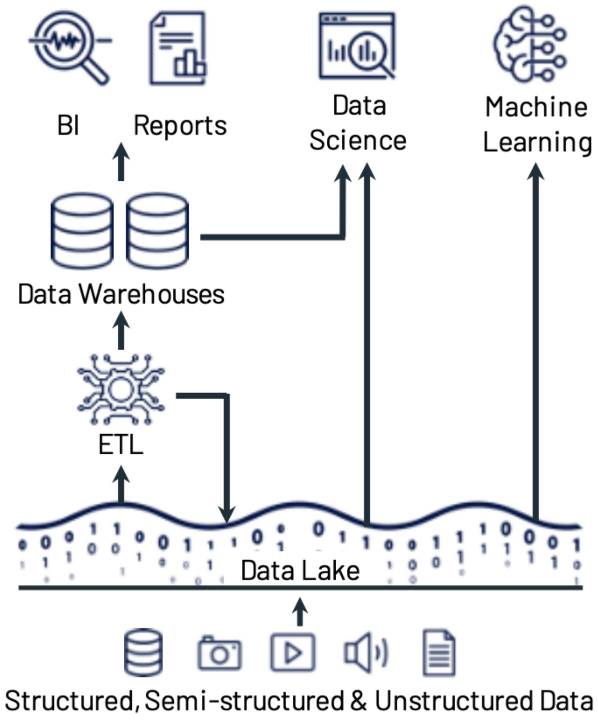
Context: previous OLAP solutions



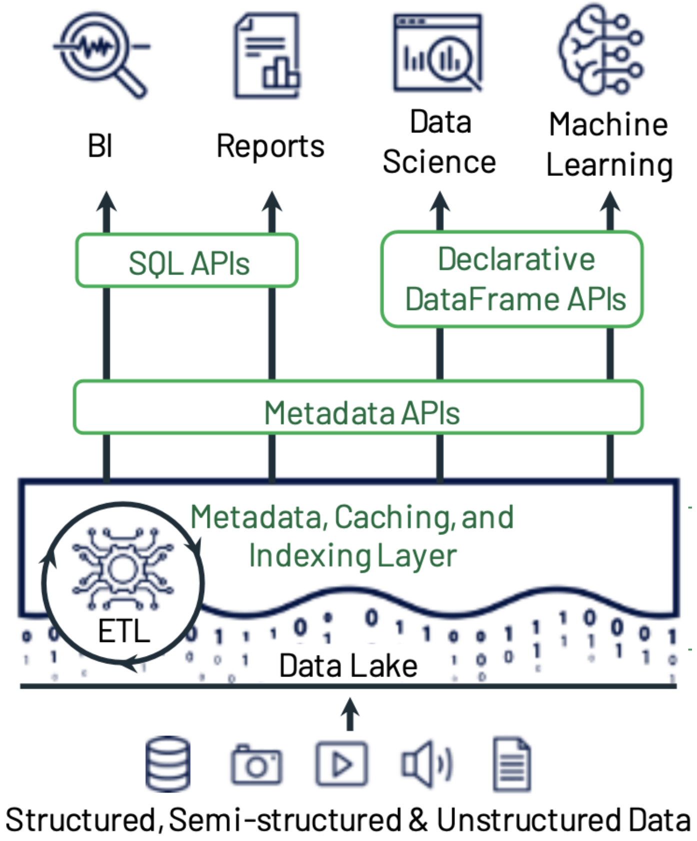
(a) First-generation platforms. (b) Current two-tier architectures.

	First-generation	2-tier architecture
Architecture	Data Warehousing	Data Lake & Data Warehousing
Data Managment	Reliable	Inconsistency
Data Format	Structured data; schema-on-write	Semi/Unstructured data; schema-on read
ML support	n/a	n/a
Cost	Expensive	Low

Solution: Lakehouse

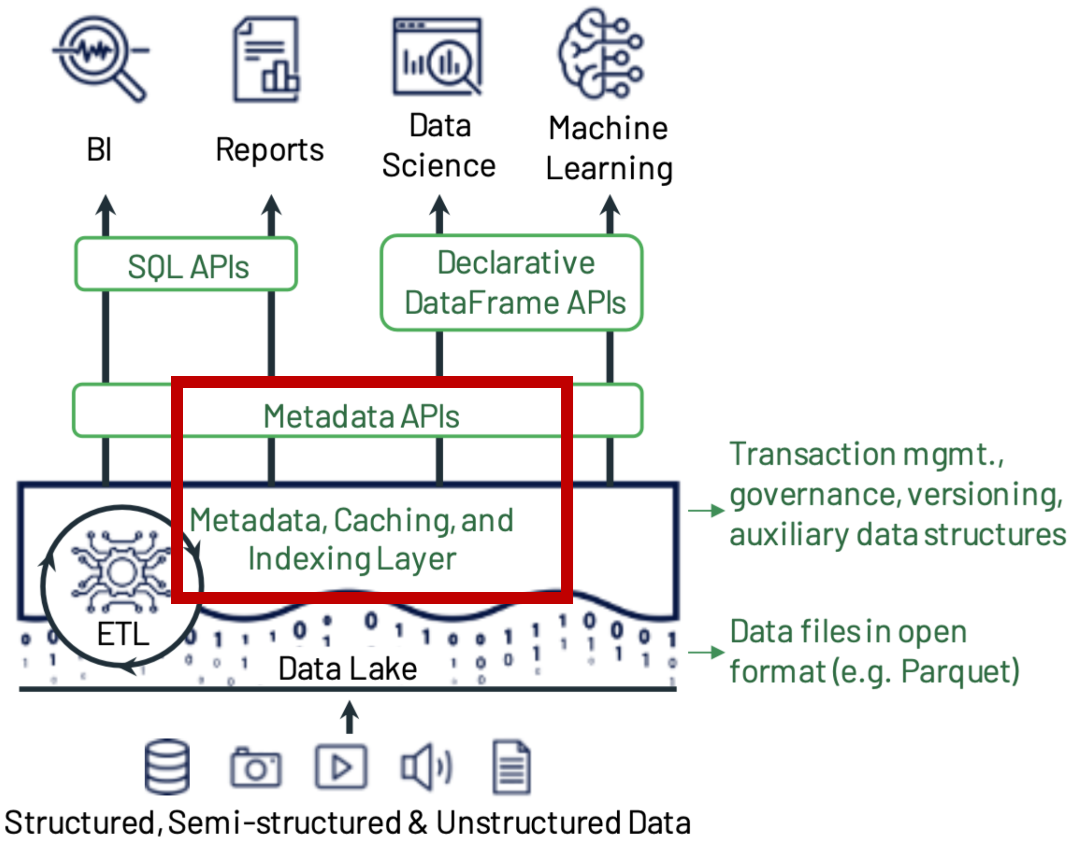


(b) Current two-tier architectures.



- Management features of Data Warehouse
  - ACID transactions, version, index...
- Directly-accessible storage of Data Lake
- Low cost (No double storage)

Architecture & Techniques



Metadata Layer

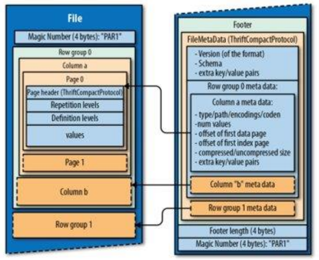
- Implement ACID transaction logic
- Data quality enforcement
- Time travel, Zero-copy cloning, ….

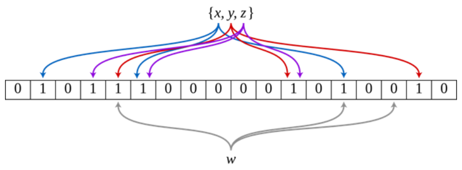
Technique:

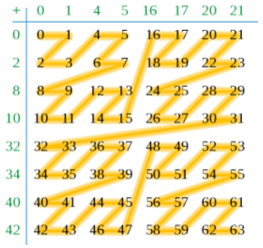
- Transaction log of table objects (which objects are part of a table version);
- Auxiliary data: Parquet, Bloom filter;
- Data layout: ordering records with Z-order / Hilbert Curve
- *easy to transfer from delta lake system

Delta Lake

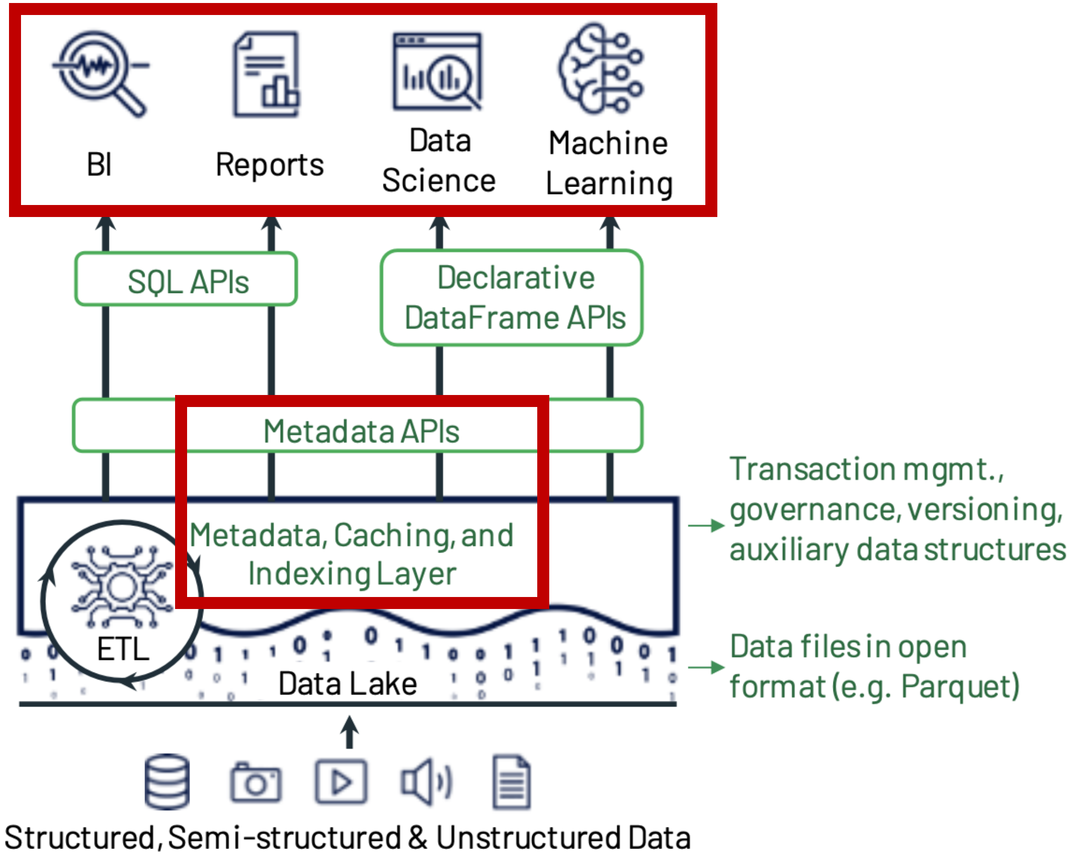
Delta Engine







Architecture & Techniques



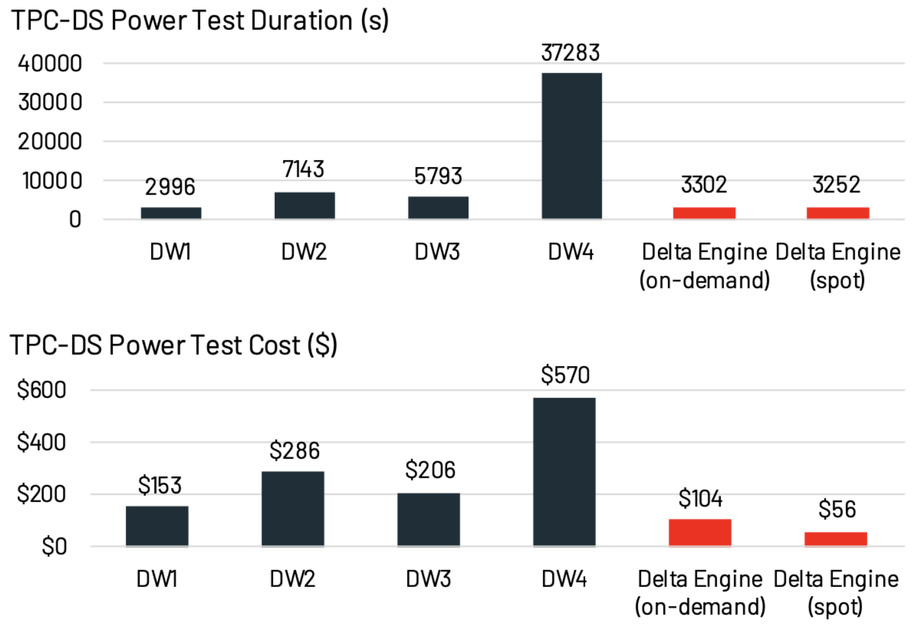
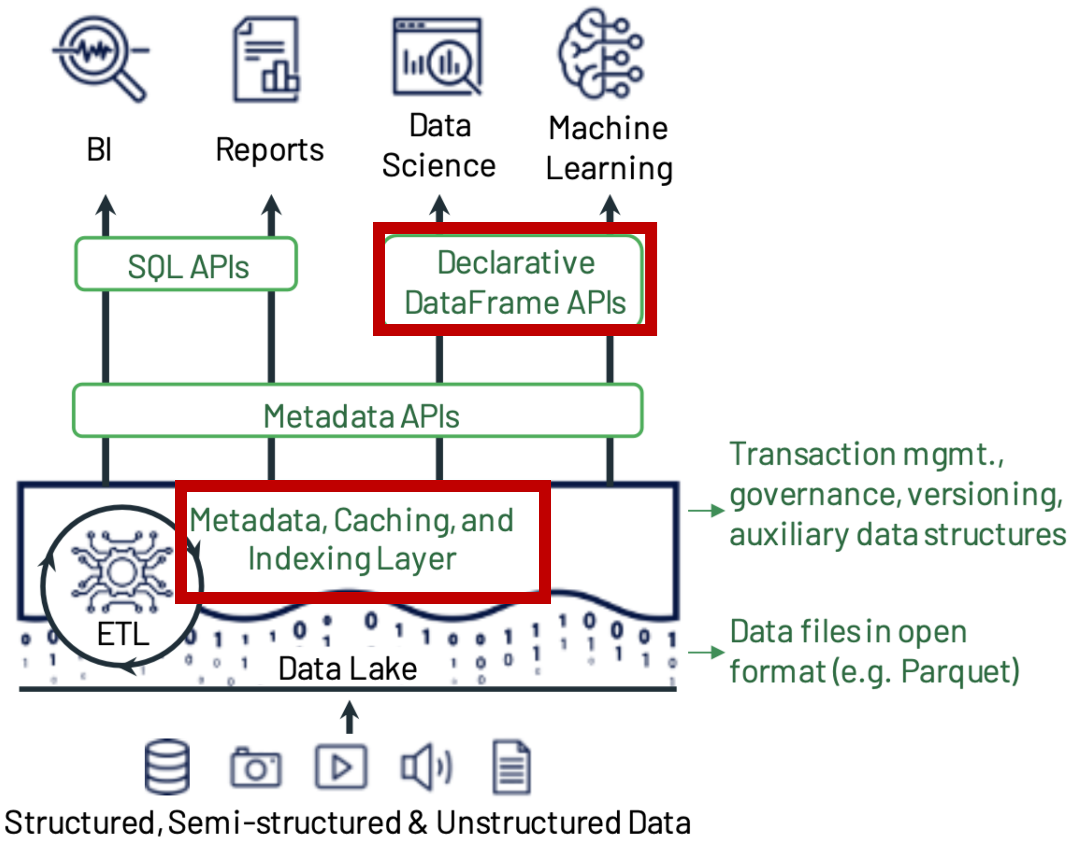


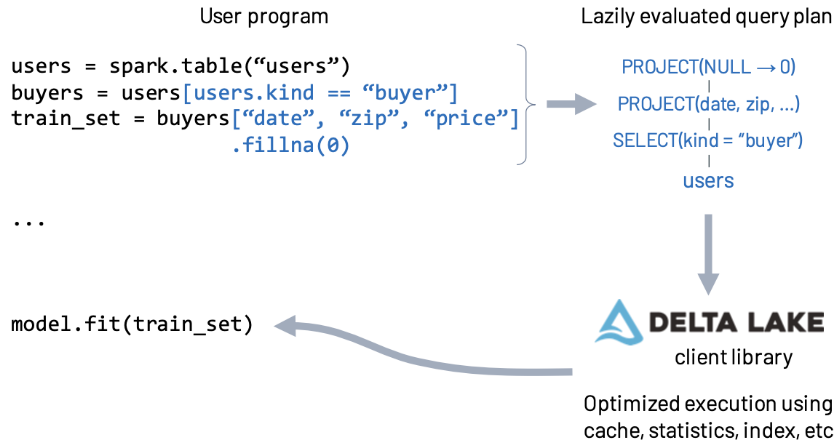
Figure 3: TPC-DS power score (time to run all queries) and cost at scale factor 30K using Delta Engine vs. popular cloud data warehouses on AWS, Azure and Google Cloud.

Architecture & Techniques



Dataframe API

- Map data preparation logic to query plan



Future ML API

- Push ML logic to query plan (factorized ML)

Technique: Delta Engine, Delta Lake

Source

- https://pages.cs.wisc.edu/~yxy/cs839-f23/papers/lakehouse.pdf
- https://www.oreilly.com/library/view/operationalizing-the-data/9781492049517/ch04.html
- https://en.wikipedia.org/wiki/Z-order_curve