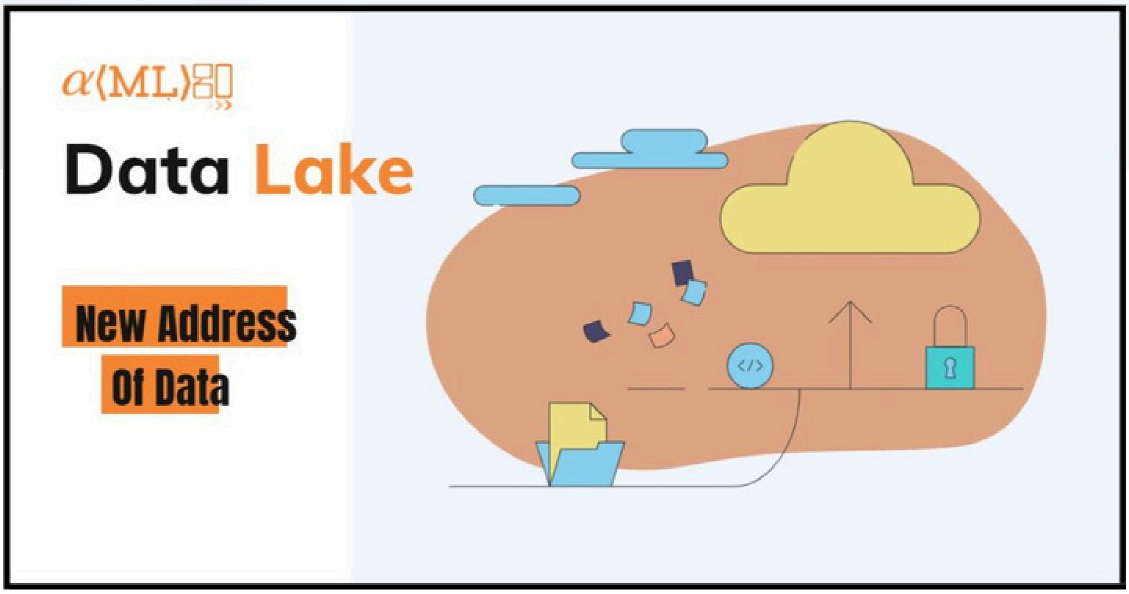
Lake Is The New Address Of Data



□ 15 December 2022 ⊠ amlgo
□ Data Engineering, Featured

Data has always been an integral part of human life. With the evaluation of the former, the latter has also evolved.

Data Lake is like a large container similar to real lakes and rivers. Just like in a lake you have multiple tributaries coming in, a data lake has structured data, unstructured data, and machine-to-machine data logs flowing through in real time.

Unlike a hierarchal Data warehouse where data is stored in Files and Folders, Data Lake has a flat architecture. Every data element in a Data Lake is given a unique identifier and tagged with metadata information.

Why Data Lake?

There are many reasons why we need a data lake. Some of the practical applications are:

- With the increase in data volume, data quality, and metadata, the quality of Decision analyses also increases. As it covers all the abnormalities or outliers.
- Data Lake offers business Insights
- It gives 360 degrees view of data and makes analysis more robust.
- Build Industry interface
- Increased Governance of Operations
- Revenue Enhancement (lead, prospect, and with real-time sales tracking one can have a good impact on revenue
- Real-time data analysis
- Forecasting & Planning
- Helps in finding out the areas of improvement

How to identify the correct tool for the Job?

Organizations are attracted to data lakes to reduce friction and complexity in their IT infrastructure and store large volumes of data without the need for lengthy data transformation on ingest.

However, Companies find that simply gathering all of the data into object storage such as Amazon S3 does not mean one has an operational data lake, to actually put that data to use in analytics or machine learning, developers need to build **ETL flows** that transform raw data into structured datasets they can query with SQL.

Due to the multiple options available in the marketplace for data lake ETL, (from open-source frameworks such as Apache Spark to managed solutions offered by companies like Databricks and Stream Sets,
and purpose-built data lake pipeline tools such as Upsolver).

One should be very alert and mindful while evaluating such software, it's important to understand the challenges of data lakes compared to traditional database ETL, and to choose a platform that will be able to address these specific hurdles.

Factors companies should consider when evaluating a data transformation platform for their data lake

1. Ability to perform complex stateful transformations

Data Lakes provide numerous cost and agility advantages for data Companies, but when considering important data analysis actions (joins, aggregations, and other stateful operations) options can be limited. These abilities play a crucial role in the interpretation of data from flowing through multiple sources and are readily available in traditional ETL frameworks.

In data warehousing, a common approach is to rely on an ELT (extract-load-transform) process in which data is sent to an intermediary. Stateful transformations are then performed using the database's processing power, historical data already accumulated, before being loaded into the data warehouse table.

In a data lake, relying on a database for every operation defeats the purpose of reducing costs and complexity by decoupling the architecture. When evaluating data lake engineering tools, make sure to choose a transformation engine that can perform stateful operations in memory and support joins and aggregations without an additional database.

2. Support for evolving schema-on-read

Databases and SQL are formulated & build around structured tables with a pre-defined schema. whereas, In reality, organizations and new data sources often generate large volumes of semi-structured data from streaming sources such as mobile applications, server logs, online advertising, and connected devices.

Semi-structured data is often much greater in size/volume and lacks a consistent schema or structure, making it extremely challenging to query within a traditional data warehouse. This leads to much less data being actively stored and utilized. Hence, data lakes have become the go-to solution for working with large volumes of semi-structured or unstructured data, as they enable organizations to store virtually-unlimited amounts of raw data in its original format.

However, it’s impossible to query data without some kind of schema – and hence data transformation tools need to be able to extract the schema from raw data and to update it as new data is generated and the data structure changes, to continuously make the data available for querying in analytics tools. One specific challenge to keep in mind in this regard is the ability to query arrays with nested data, which many ETL tools struggle with, but can often be seen in sources such as user app activity.

3. Optimized object storage for improved query performance

A database optimizes its file system to return query results quickly optimised query performance is important to ensure data is readily available to answer business-critical questions, as well as to control infrastructure costs. (This feature is missing in ‘vanilla’ data lake storage, where data is stored as files in folders on cloud object storage such as Amazon S3).

Trying to read raw data directly from object storage will often result in poor performance (up to 100-1000x higher latencies). Data needs to be stored in columnar formats such as Apache Parquet and small files need to be merged to the 200 MB-1 GB range in order to ensure high performance, and these processes should be performed on an ongoing basis by the ETL framework in place.

Traditional ETL tools are built around batch processes in which the data is written once to the target database. with continuous event streams stored in a data lake, this approach is typically inadequate. Hence, organizations should opt for data transformation tools that write the data to the lake multiple times to continuously optimize the storage layer for query performance.

4. Integration with metadata

One of the main reasons to implement a data lake is to be able to store large amounts of data now and decide how to analyze it later. Data lakes are meant to be flexible and open to support a wide variety of analytics.

A core element of this open architecture is to store metadata separately from the engine that queries the data. This provides a level of data engine agnosticism which makes it easy to replace query engines or to use multiple engines at the same time for the same copy of the data. For example, Hive, Meta-store, or AWS Glue. Data Catalogue can be used to store metadata, which can then be queried using Apache Presto, Amazon Athena, and Redshift Spectrum – with all queries running against the same underlying data.

So, any selected data integration tool must support this open architecture functionality with the metadata catalog both storing and continuously synchronizing metadata with every change (location of objects, schema, partition) so that data remains easily accessible by various services.

5. Enabling ‘time-travel’ on object storage

One of the advantages of storing raw data in a data lake is the ability to ‘replay’ a historical state of affairs. This is hard to achieve in databases as they store data in a mutable state, which makes testing a hypothesis on historical data impossible in many cases – e.g., if we choose to reduce costs by pre-processing or pruning the data before loading it into the database. Even when it is possible, the performance stress and costs of running such a query could make it prohibitive, creating tension between operations and exploratory analysis.

Data lakes are based on an event-sourcing architecture where raw data is stored untouched. Historical data is streamed from object storage for quick validation when a hypothesis presents itself.

Data lake tools should reduce the friction of orchestrating such ad-hoc workloads and make it easy to extract a historical dataset, without creating large operational overhead. This is achieved by ensuring multiple copies of the data are stored according to a predetermined retention policy, while data lineage, schema changes, and metadata are kept to retrieve a previous state of the data.

6. Ability to update tables over time

While databases typically support updates and deletes to tables, data lakes are composed of partitioned files which predicate on an append-only model. This can create difficulty in storing transactional data, implementing CDC in a data lake, or addressing regulatory concerns. To comply with GDPR or CCPA requests, organizations must be able to accurately retrieve and delete any copy of PII stored on their infrastructure.

Modern data lake engineering tools should provide the means to bypass this limitation by enabling upserts in the storage layer and in the output tables being used for analytic purposes. To do so, these tools can leverage in-memory indices or key-value stores that enable developers to access and update records in object storage.

Amlgo Labs *offers organizations an extremely scalable, cloud-based solution that can fundamentally simplify their data Analytics experience, eliminating complexity &*

cost, and instead focusing on gaining the insights that the data provides. ***Amlgo Labs*** *offers a complete end-to-end data analytics solution with their highly trained consultants & Services.*