The Rise of Open Table Formats: Diving into the Next Decade of Data Lakes (Panel Discussion)

Walaa Eldin Moustafa

Senior Staff Software Engineer, LinkedIn

Abstract

The evolution of data lakes and their constituent technologies has been marked by remarkable milestones that have reshaped the landscape of data management and processing. Fifteen years ago, Hadoop won the terabyte sort benchmark [1], an achievement that signified a historic shift in the data processing domain. Following this victory, numerous initiatives were undertaken to formulate and standardize data exchange protocols. File formats like Avro [2], ORC [3], and Parquet [4] emerged as the de facto standards, embodying efficient data serialization and deserialization protocols for large-scale processing. Building on the concept metadata as a foundation, the Hive engine [5] was introduced, coupled with the Hive metastore and the Hive table format, which abstracted these intricate file formats. In doing so, Hive granted big data users the powerful capability of leveraging a SQL abstraction atop vast data lakes, simplifying complex operations and making them more accessible to a broader audience. This marked a transformative period, setting the stage for the subsequent rise of further modern big data compute engines, such as Spark [6], Presto [7], Trino [8], Flink [9], BigQuery [10], Azure Synapse [11], and Redshift [12], along with an emerging set of open table formats like Iceberg [13], Delta Lake [14], and Hudi [15].

Modern open table formats represent pivotal advancements in the continuous evolution of data lakes. While early data lake technologies laid the foundation, these open table formats brought about a renaissance in how data practitioners interacted with massive datasets. One of the most transformative features introduced by these systems is their native support for ACID transactions, ensuring atomicity, consistency, isolation, and durability even at massive scales. This capability revamped the trustworthiness and reliability of big data operations, enabling consistent reads and writes. Furthermore, they integrated versioning at the core of their design, allowing users to maintain, access, and revert to historical data states, a crucial feature for auditability and reproducibility. Change Data Capture (CDC), another seminal feature, provided the means to capture and process granular data modifications, eliminating the cumbersome and inefficient batch-reload paradigms of the past. Additionally, the introduction of incremental compute facilitated more frequent and efficient updates by processing only the altered data segments, rather than entire datasets. Finally, modern table formats enhanced performance by utilizing fine-grained statistics and advanced indexing. These innovations allow for more efficient data access and querying, drastically reducing processing times and resource consumption. Collectively, these advancements not only bridged many of the traditional gaps in data lake architectures but also drastically elevated the user experience, making data processing more robust, performant, agile, and user-friendly.

As we sail into the horizon of the next decade of data lakes, a myriad of compelling topics emerge, reflecting both the challenges and opportunities in this rapidly evolving domain.

(i) The debate between the future of open and proprietary table formats arises, with proponents of each vouching for community-driven innovation, or enterprise-driven robustness.

(ii) A central theme of data lake evolution is the direction of influence in API design between compute engines and table formats. The question at hand revolves around interdependence and interplay: Do the evolving needs and architectures of compute engines primarily shape the API design of table format specifications? Or is it the innovations and constraints in table format specifications that predominantly drive the API designs of engine connectors? Unpacking this interaction becomes pivotal in understanding the trajectory of future advancements and in harnessing the combined potential of compute engines and table formats.

(iii) Interoperability between table formats, along with SQL interoperability between compute engines and table format specifications, is paramount. Such interoperability ensures that data practitioners aren't locked into a particular ecosystem but can seamlessly transition and integrate diverse systems, fostering a richer, more collaborative data environment.

(iv) Security, audit, and lineage features are more critical than ever, as the world becomes more data-centric and regulations grow stringent. Ensuring the traceability of data transformations, access controls, and secure data handling mechanisms are at the forefront of discussions.

(v) The intricacies of bookkeeping data lakes and the meticulous task of maintaining data lake tables present challenges that require novel solutions, blending both technological innovation and best practices.

(vi) Lastly, the recent pervasiveness of AI introduces an array of new requirements for data lakes. As machine learning and deep learning models become integral to business processes, data lakes must adapt to cater to the specific needs of AI workflows. In essence, the proliferation of AI will surely define a new chapter for data lakes, demanding architectural and functional enhancements to meet the unique challenges and potentials of AI-driven analytics.

Looking back at the last decade, it's evident that the data community has made significant progress, turning challenges into meaningful innovations. As we look forward to the next ten years, many questions remain. However, one clear fact stands out: data will continue to be of an ever-growing importance in our society, serving as a key catalyst in advancing education, healthcare, and economic equity for wider communities and future generations.

Panel Members:
Ryan Blue, Co-creator of Apache Iceberg.
Rahul Potharaju, Senior Engineering Manager, Databricks.
Nishith Agarwal, Co-creator of Apache Hudi.
Dain Sundstrom, Co-creator of Trino.
Justin Levandoski, Engineering Director, Google BigQuery.
Jesus Camacho Rodriguez, Principal Research Engineering Manager, Microsoft

References

[1] Hadoop Wins Terabyte Sort Benchmark, https://hadoop.apache.org/news/2008-07-xx-terabyte-sort.html, 2008.
[2] Apache Avro, https://avro.apache.org/, 2022.
[3] Apache ORC, https://orc.apache.org/, 2023.
[4] Apache Parquet, https://parquet.apache.org/, 2023.
[5] Apache Hive, https://hive.apache.org/, 2023.
[6] Apache Spark, https://spark.apache.org/, 2018.
[7] Presto, https://prestodb.io/, 2022.
[8] Trino, https://trino.io/, 2023.
[9] Apache Flink, https://flink.apache.org/, 2023.
[10] Google BigQuery, https://cloud.google.com/bigquery, 2023.
[11] Azure Synapse, https://azure.microsoft.com/en-us/products/synapse-analytics, 2023.

Joint Workshops at 49th International Conference on Very Large Data Bases (VLDBW'23) — Second International Workshop on Composable Data Management Systems (CDMS'23), August 28 - September 1, 2023, Vancouver, Canada

[12] Amazon Redshift, https://aws.amazon.com/redshift/, 2023.
[13] Apache Iceberg, https://iceberg.apache.org/, 2023.
[14] Delta Lake, https://delta.io/, 2023.
[15] Apache Hudi, https://hudi.apache.org/, 2021.