CLOUDERA

WHITEPAPER

INTRODUCING APACHE ICEBERG:

The Case for an Open Data Lakehouse Powered by Cloudera

The key to unlocking the full potential value of data is implementing an open, flexible, and unified lakehouse architecture.

Table of Contents

Introduction

Data teams are under pressure to deliver a wide range of analytics use cases, from real-time Business Intelligence (BI) to AI-powered applications, to delivering insights to a growing number of technical and non-technical data consumers. At the same time, they must manage rapidly expanding volumes of structured, semistructured, and unstructured data, which is often distributed across several data stores, in multiple clouds and on-premises.

One solution that solves many of the challenges of managing and delivering access to data is an open data lakehouse, an architecture that combines the flexibility and scalability of data lake storage with the data management, data governance, and analytics performance of the data warehouse. A key component of the open data lakehouse architecture is an open table format, which provides warehouse functionality and ease of data management, delivers high-performance analytics, and gives teams full control over their data and the freedom to leverage any execution engine.

This paper explores Cloudera's implementation of Apache Iceberg, an open table format that is quickly gaining popularity for its ability to deliver high-performance analytics at petabyte scale while dramatically simplifying data operations. With Cloudera and Apache Iceberg, organizations can build and implement an open data lakehouse architecture across all of their cloud and on-premises environments to satisfy virtually any analytic workload.

The Need for the Data Lakehouse

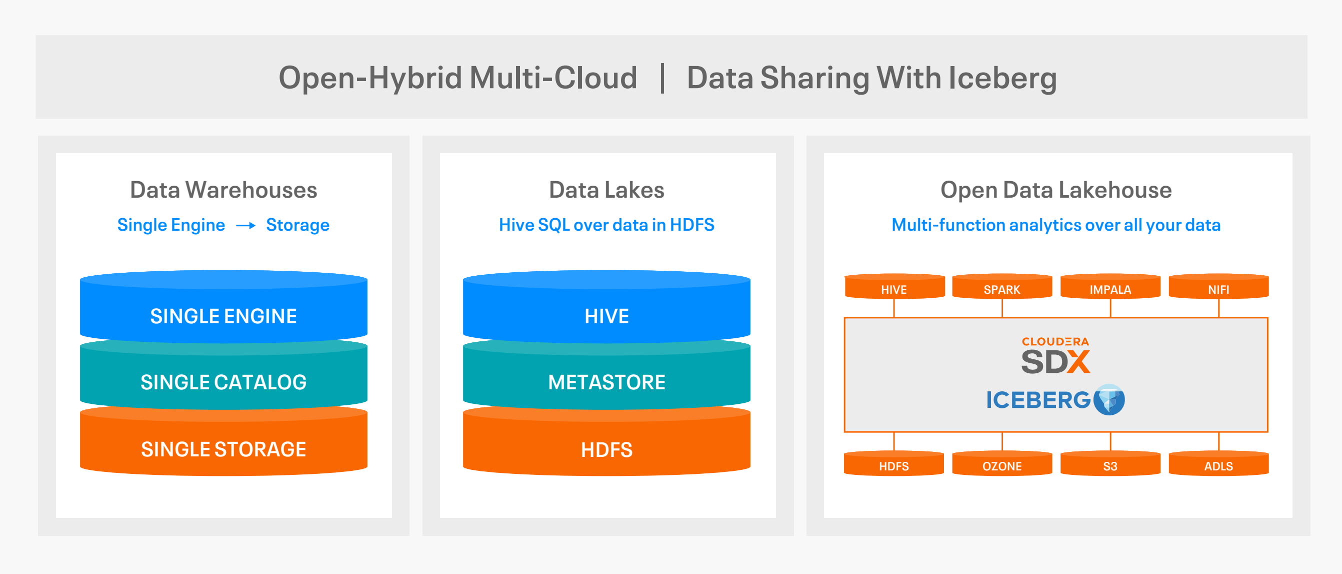
Decades ago, businesses adopted legacy data warehouses, which were primarily designed to ingest, store, and analyze structured data from on-premises business systems. As data grew in volume, variety, and velocity, these proprietary, appliance-based systems struggled to scale efficiently, and as the demand for insights grew, data teams could not meet Service-Level Agreements (SLAs) for query performance.

To address those challenges, many organizations turned to the data lake, which provided cheap, efficient, and infinitely scalable storage for large volumes and a variety of data types. However, the data lake was not built for analytics. As a result, most companies found themselves managing a complex, distributed architecture of data lakes and data warehouses, and moving and copying data from the data lake into the warehouse to make that data available for BI and reporting.

Query engines directly on data lake storage, as well as providing access to data in the data lake from relational databases via external tables, were the first attempts to circumvent the inefficiencies of distributed architectures and unify the data. Organizations querying data directly in the data lake discovered that standardizing on open file formats like Apache Parquet and Apache ORC provided substantial compression and performance benefits over file formats like CSV and JSON, but the data lake still lacked the write functionality and the ease of management of the data warehouse.

Open table formats build on the compression and performance benefits of open file formats and effectively bridge the gap between the data lake and data lakehouse, providing full data management functionality with Atomicity, Consistency, Isolation, Durability (ACID) guarantees to support any data engineering task. Additionally, they deliver interoperability, so data consumers can leverage the best tool for each workload, and quickly adopt new innovations to work with the data.

The next section explores the components of an open data lakehouse in more detail.



The Components of an Open Data Lakehouse

Gartner defines a data lakehouse as an architecture that "integrate[s] and unif[ies] the capabilities of data warehouses and data lakes, aiming to support AI, BI, ML and data engineering on a single platform."[1] An open data lakehouse stack leverages four essential components to satisfy these criteria:

1. **The storage layer.** Object storage, either in the cloud or on-premises, represents the primary landing zone for significant volumes of customer and operational data, and it is the foundation of the data lakehouse.
2. **The table layer.** Open table formats like Apache Iceberg bring data warehouse functionality to data lake storage, including transactional consistency, performance benefits, optimal storage utilization, and more.
3. **The engine layer.** Execution engines enable data consumers to interact with the data, and the open data lakehouse provides engine freedom so users can choose the best tool for each workload, including BI & reporting, AI & ML, and data engineering.
4. **The optimization layer.** Query accelerators like indexing, in-memory caching, query planning, and more accelerate query performance.

As a result, the open data lakehouse provides several benefits above and beyond what data teams can achieve with a data lake or a data warehouse. It provides:

- **A unified view.** By leveraging the flexibility and scalability of object storage, organizations can store and analyze structured, semistructured, and unstructured data and make it accessible without performing complex transformations to move it into proprietary formats.
- **A single, consistent, high-performance version of the data.** Open table formats provide transactional guarantees, optimization capabilities like compaction and vacuum, ease-of-management functionality like schema evolution and hidden partitioning, and time travel and rollback that make it easy to recover from mistakes. The net result is that data teams can easily deliver a single, consistent, high-performance view of the data for every analytic use case.
- **Engine freedom.** The interoperability of open table formats means that organizations can choose the best tool for each analytic workload. Data teams maintain full control over their data, and they are not reliant on proprietary formats or data copies to meet performance SLAs. Additionally, in a data industry that is rapidly advancing, data teams maintain the ability to quickly adopt new tools and services.
- **Lower Total Cost of Ownership (TCO).** The simplified architecture of the open data lakehouse means customers ultimately spend less money to achieve better results. Businesses often achieve cost savings in the form of eliminating complex Extract, Transform & Load (ETL) or ELT processes and maintaining multiple data copies associated with legacy and cloud data warehouses, optimizing query performance and storage utilization, and making data teams more efficient and productive.

Approaches to Implementing a Data Lakehouse

Based on the components in the previous section, there are several approaches to implementing a data lakehouse that broadly fall into three buckets, primarily based on the organization's choice of table format.

The Old Way: Hive Tables

Early data lakes used Hadoop Distributed File System (HDFS) for cheap, efficient storage on-premises, and Hive tables were optimized for Hive SQL. Its primary use case was batch ETL, but it struggled with latency for BI and reporting workloads, especially for queries that required near-real-time or interactive speed. These challenges were primarily related to the architecture of the table format, and its dependence on Hive Metastore.

As data continued to grow, so did the challenges of managing Hive tables. They were primarily designed for read-only workloads and not for managing schema changes or reads and writes from multiple engines. As an example, if partitions changed, or if columns needed to change, the entire table would often need to be rebuilt, and downstream applications might also be impacted.

A Modern Alternative: Delta Lake

Modern open table formats evolved to address the architectural challenges of Hive tables. Delta Lake is one such table. It is an open-source solution that leverages data lake storage solutions including HDFS, Amazon Simple Storage Service (Amazon S3), and Microsoft Azure Data Lake Storage (Microsoft ADLS). It was initially developed by Databricks and it extends the capabilities of Apache Spark to provide robust data management functionality, including ACID transactions, data versioning, schema enforcement, and time travel. Similar to how Hive was optimized for HDFS, Delta Lake is optimized for Spark.

However, Delta Lake may not be the best fit for all enterprises. Here are a couple of scenarios that may indicate the need for a better option.

[1] Gartner. "Exploring Lakehouse Architecture and Use Cases." January 11, 2022. **https://www.gartner.com/en/documents/4010269**

1. Delta is optimized for Spark. However, integration with other engines is not always available. Data teams who require special connectors, transformations, and self-built optimizations to use other engines might experience performance issues.

2. Delta Lake only supports Parquet files today. If data is in ORC or Avro formats, data teams first need to write the files in Parquet, which creates additional management overhead and cost.

3. Every time in-place partitioning of a table evolves, Delta Lake users must recreate each table, so tables with in-place partitions that evolve regularly require additional management overhead and cost.

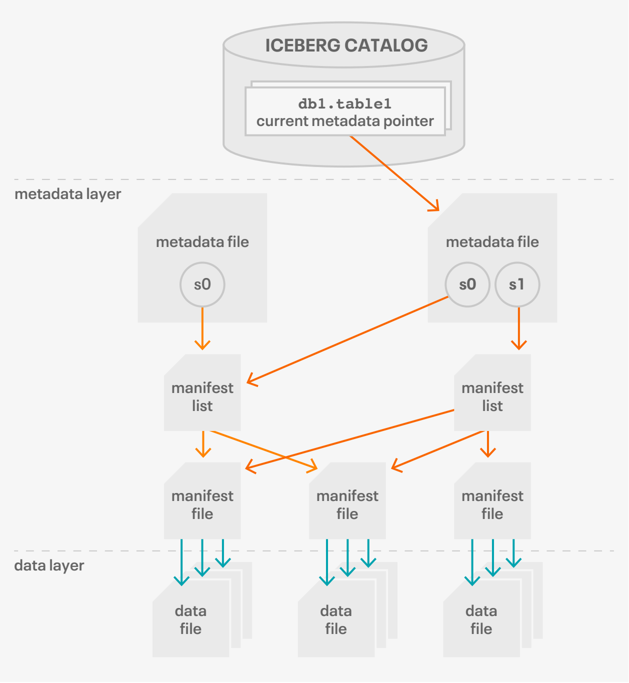
4. If data teams have to manage data at scale, Delta Lake may not be the best choice. While Delta Lake is scalable, some users note that scaling data in it can be both challenging and expensive.

5. Although Delta Lake code is open source, one vendor makes up well over 75% of the project's contributions. As a result, many of the innovations and development priorities of the project are dictated by how Databricks chooses to grow their business. Additionally, Delta Lake is limited in terms of the commercial and Open Source Software (OSS) ecosystem around it, and the commercial version of Delta Lake contains features that are not in the open-source version, so companies who value openness and interoperability should factor in the limitations of Delta Lake.

Apache Iceberg: High-Performance Analytics on Massive Datasets

Apache Iceberg is an open table format designed specifically for high-performance analytics on large volumes of data. It was initially developed by Netflix to address the limitations and challenges associated with managing and processing massive datasets in cloud storage systems like HDFS, Amazon S3, Azure Data Lake Storage, Ozone, and more. By abstracting the storage from the compute engines, Iceberg provides the flexibility to connect any data store with any engine. By implementing a new architecture that maintains table metadata in its own layer, Iceberg eliminates the dependency on metastores, which was one of the primary limitations for Hive tables.

Today, Iceberg has the fastest adoption rate among open table formats in the market, and it is becoming the de facto standard for many companies, primarily due to its openness, its engine-agnostic architecture, and its vendor-neutral development. Iceberg currently has the broadest commercial and Open-Source-Software (OSS) ecosystem, the most diverse set of committers, and the largest community among table formats. Whereas Hive and Spark were tightly coupled with their respective table formats, Iceberg enables customers to choose any engine.



Unlike Hive tables, Apache Iceberg tables maintain their own metadata layer.

Graphic illustration based on image source: The Apache Software Foundation. **https://iceberg.apache.org/assets/external/iceberg.apache.org/assets/images/iceberg-metadata.png**

Apache Iceberg Features

Apache Iceberg includes many features that make it easier than ever for data teams to efficiently manage and deliver access to a high-performance view of their data to all of their data consumers.

- **Compaction:** Iceberg rewrites small files into larger files to optimize read query performance. This feature solves a common challenge for customers who are ingesting small files, such as in a streaming or a micro-batching ingestion workload.

- **Vacuum:** The Vacuum feature removes unused files to optimize storage utilization.

- **Hidden Partitioning:** Iceberg does not require user-maintained partition columns—it produces partition values for rows in a table, and avoids reading unnecessary partitions automatically. Moreover, partitions can change over time. The net result is that data teams can deliver a high-performance view of the data in a table with lower manual effort, even as the data in the table changes over time.

- **Schema Evolution:** Iceberg supports in-place table evolution. Data teams can add, drop, rename, update, or reorder columns in a table without rewriting the physical data files.

- **Time Travel and Rollback:** Iceberg generates a snapshot whenever users create or modify a table. Snapshots use metadata pointers to recreate a view of the table at a point in time. The snapshot feature enables data teams to track and audit how a table has changed over time, and even roll back unwanted changes and return the table to a previous state.

- **Multi-Engine Support:** No single execution engine is optimized for every workload. Iceberg enables data teams and data consumers to choose the right engine for each workload, whether it is streaming or batch ingestion, machine learning, exploratory BI, or transactional processing.

Cloudera Delivers an Open Data Lakehouse Built on Iceberg

Iceberg provides a strong foundation for an open data lakehouse, enabling easy data management and high performance with the flexibility to manage BI and reporting, AI & ML, and data engineering workloads with a variety of execution engines.

Cloudera builds on top of that foundation, providing a single platform for all analytic workloads that can be deployed in any public or private cloud, with optimizations for high-performance analytics, and with unified security and governance across environments.

The Iceberg open data lakehouse with Cloudera provides many advantages over traditional and cloud data warehouses and data lakehouse alternatives.

Eliminate Data Silos

Traditional approaches to data warehousing workloads in particular came with some severe drawbacks as the volumes and variety of data grew: new data required movement and transformation via ETL or ELT processes, and engines often required BI cubes and extracts to meet performance SLAs. These pipelines and data copies became additional assets that data teams needed to manage and maintain over time, and data consumers experienced bottlenecks for data access and performance. Managing multiple data copies often meant that data consumers were not working with the same version of the data. Additionally, the entire stack became much more expensive to maintain.

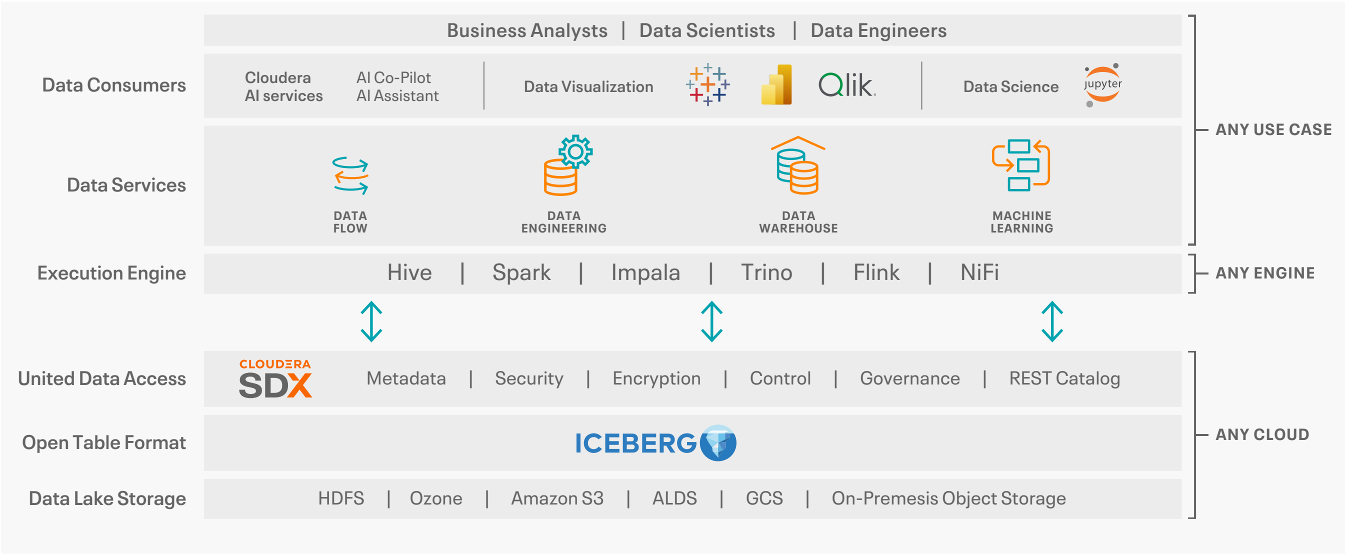
The modern cloud data warehousing approach is built on this exact architectural pattern, except it's in the cloud, and consumption-based pricing models often incentivize those complexities for vendors.

Cloudera breaks down silos by providing direct access to Iceberg tables in the data lake, eliminating the need for additional data movement, ETL processes, and data copies. Data lands one time in object storage, data is presented in Iceberg, and multiple users can access and leverage that data concurrently using their engine of choice, so every data consumer is working with a consistent and accurate version of the data.

Data Engineering Productivity

Data teams in many organizations spend the majority of their time responding to support tickets including requests for access to new data sources, dashboard creation, performance issues, and more. In fact, a study of data teams conducted by Monte Carlo Data found that nearly half of a data engineer's time is spent on data quality issues, and nearly 74% of data teams report "that business stakeholders often identify issues first, suggesting a reactive rather than proactive approach to data management."[2]

By reducing architectural complexity with an open data lakehouse on Iceberg, data engineers can spend their time on more high-value projects. Cloudera customers have reported up to a 20% improvement in data productivity.



Support for Hybrid & Multi-Cloud Deployments

Most businesses manage a distributed infrastructure with a mix of public clouds and on-premises storage solutions, with critical customer and operational data stored in multiple locations. Cloudera is the only open data lakehouse that gives customers the ability to deploy anywhere the data resides, all connected by a common control plane and unified security and governance solution, so data consumers have easy access to all of their data, and data teams have a consistent data management experience across all of their environments.

REST Catalog Integration

Cloudera supports the REST catalog specification, an open catalog implementation for Iceberg. This integration enhances interoperability by enabling access to tables in a catalog by any Cloudera or third-party execution engine that can read or write to Iceberg. It eliminates data silos, prevents data duplication, simplifies data pipelines, and improves productivity across analytic tools and services while reducing the Total Cost of Ownership. Cloudera reinforces its commitment to openness and builds on the benefits of the REST catalog with end-to-end, unified data security, governance, and metadata management.

Multi-Engine Support for Multi-Function Analytics

Cloudera provides multi-function capabilities that cover the entire data lifecycle, including streaming and batch ingestion, data engineering, interactive and exploratory analytics, machine learning and AI, and more. Customers can leverage best-of-breed engines and tools for each workload, enabling more data consumers to leverage data in their roles.

Storage Duplication

Storage duplication as a result of additional data movement, data copies, and distributed data stores can lead to many challenges, including stale data, operational inefficiencies, and additional costs. By delivering access to a single source of data and eliminating unnecessary processes, customers typically see up to a 50% reduction in storage costs.

Unified Security and Governance

Unified governance is critical for companies managing distributed data architectures. Data teams need to democratize access to data for a wide range of analytic use cases while also centralizing security and governance so the data is safe, reliable, and highly available.

Cloudera provides a centralized control plane and unified governance solution called Shared Data Experience (SDX) that integrates with Iceberg to extend the value of the metadata within the tables, providing full lineage of all of the data from the sources to the presentation layer.

By leveraging the snapshot feature of Iceberg, Cloudera customers have access to a full history of the changes made to a table, and they can even use time travel to recreate a certain view or a dashboard. Data scientists who require AI and ML explainability can recreate the results of a model at a specific point in time. Finally, data teams can fully roll back a table to recover from mistakes.

With Cloudera, data teams maintain full, centralized control and governance of their environment, and data consumers can easily access and leverage a consistent and accurate view of their data.

The Safest Path From Trusted Data to Trusted AI

As business leaders race to capitalize on the perceived value of AI to the business, data teams are focusing on the foundation of AI models: the data itself. Trusting the data used for AI initiatives is paramount for organizations seeking accurate and reliable insights. The value of adopting an open data lakehouse architecture is in its flexibility for handling diverse data types in a secure and controlled environment, enabling data teams to power trusted AI models with trusted data at any scale. With open machine learning models, organizations can move from historic and real-time to predictive insights that power the most transformative business use cases in production today.

In summary, the open data lakehouse, built on Iceberg and powered by Cloudera, simplifies data architectures, provides unmatched engine and deployment flexibility, centralizes security and governance, and empowers every data consumer to leverage their data for real-time and predictive analytics that create business value.

Iceberg and Cloudera Help a Social Media Company Scale their Data Platform

A Japanese social media company specializing in financial technology and business commerce encountered several critical challenges in their data management and scalability as data volumes grew. They had trouble managing data silos across the organization, keeping the data fresh in all of those silos, resolving data synchronization issues, and maintaining complex data pipelines. And as their data scaled, traditional methods of changing schemas and partitions involved multiple table rewrites, and it became very expensive. Moreover, when data was locked into proprietary databases, visibility and access to that data for newer AI use cases was difficult to deliver.

The company chose Iceberg on Cloudera as the solution to these challenges. By adopting Iceberg, they established comprehensive data governance practices, ensured scalability across their data infrastructure, and achieved superior performance over other data storage formats. By using open standards and formats, they could provide access to data for wider teams with more tools. This transformation empowered

[2] Monte Carlo Data. "The Annual State of Data Quality Survey." March 2023. **https://www.montecarlodata.com/blog-data-quality-survey**

them to meet their business goals effectively and efficiently. The company's Chief Data Officer shared some insights into their Iceberg implementation at a recent Cloudera online event.

How does Apache Iceberg support the company's unique data management needs?

Changing table definitions or adding and deleting columns from a table over its lifecycle is inevitable. The larger the scale of the data, the more costly this action can become. Iceberg facilitates schema changes to tables as well as partial data changes. Existing data formats may require data migration when columns are added or deleted, or when a table definition changes. Since Iceberg can make changes while maintaining the existing data, it has the potential for significant cost savings on large-scale data platforms. This was a big benefit for the company.

How has Iceberg on Cloudera enhanced data processing and analytics?

Iceberg contributes to the scalability of concurrent query update and delete operations. Traditional table formats may require data migration, which results in temporary suspension of existing data pipelines while adding or deleting data. This suspension caused a surge in system workloads and, as a result, increased the amount of time needed to deliver data to users. Alternatively, Iceberg can continue to update data while maintaining data consistency and accuracy, mitigating data pipeline-related delays due to data changes. Therefore, reliable data can be delivered to analysts without a surge in workloads. And the other services integrated in Cloudera's lakehouse, such as streaming, machine learning, and BI, provide one platform for everything, eliminating data movement and data copies.

How does Iceberg assist in overcoming data management challenges?

For global companies, it is imperative to comply with the ever-changing policies in different jurisdictions and with global privacy regulations. The data team needs to be agile in its data management processes as a result. One of the benefits of Iceberg is its change resilience, enabling robust yet flexible data management, so users can expect significant improvements to addressing business risks and improving time to service. Iceberg has a feature called time travel and rollback, allowing data teams to view data at a specific point in time. This is useful for tracking the history of changes and, if needed, restoring data to a previous state. For the company, Iceberg will continue to be one of the core technologies that will support quality data management now and in the future.

Best Practices for Implementing Apache Iceberg

The following best practices are compiled from conversations, feedback, and projects with customers who have implemented Iceberg Tables with Cloudera. For customers pursuing an open data lakehouse architecture, these tips can accelerate the delivery of high-performance views of the data, reduce storage costs, and simplify data management processes.

1. **Leverage In-Place Partition Evolution** to break data down into more granular partitions, improving query performance. Data consumers do not need to know how a table's partitions have changed to write performant queries thanks to hidden partitioning.

2. **Enable metadata** caching to reduce repeated reads of small Iceberg manifest files from remote storage by Impala Coordinators and the catalog. This caching improves query performance planning by up to 12 times. Cloudera has contributed this feature to the Iceberg open source community.

3. **Use row-level mutation** to optimize tables based on the analytic use case. Iceberg supports two types of row-level mutations: Copy on Write, suitable for batch processing and machine learning workloads, and Merge on Read, which is better for streaming analytics or real-time BI. Understand specific use cases and how data will be ingested, processed, and consumed to implement the proper row-level mutation strategy.

4. **Accelerate query performance with materialized views.** Cloudera provides this popular data warehouse capability on Iceberg tables to keep queries performant. Materialized views are also in Iceberg tables, and are accessible by other execution engines, so data teams can create, maintain, and evolve their business views in an interoperable format.

5. **Solve the small files problem with data compaction.** When customers ingest multiple small files, file-level operations, such as opening, scanning, and closing files, can significantly impact query performance. Compaction rewrites small files into larger files, accelerating performance. As with row-level mutations, understand specific use cases and query performance needs to understand how often compaction jobs should be run.

6. **Expire snapshots periodically** to reduce storage costs. Snapshots should be expired based on the organization's data retention policy, business requirements, and use cases.

7. **Utilize compression** to balance storage and performance requirements. Uncompressed data results in the fastest queries, but also consumes the most storage space. The Snappy Codec compresses the data, resulting in a slower query but improved storage utilization. Gzip results in the lowest storage requirement and the slowest queries. Choose the compression method that best fits within end-user SLAs for performance.

About Cloudera

Cloudera is the only true hybrid platform for data, analytics, and AI. With 100x more data under management than other cloud-only vendors, Cloudera empowers global enterprises to transform data of all types, on any public or private cloud, into valuable, trusted insights. Our open data lakehouse delivers scalable and secure data management with portable cloud-native analytics, enabling customers to bring GenAI models to their data while maintaining privacy and ensuring responsible, reliable AI deployments. The world's largest brands in financial services, insurance, media, manufacturing, and government rely on Cloudera to be able to use their data to solve the impossible—today and in the future.

To learn more, visit **Cloudera.com** and follow us on **LinkedIn** and **X**. Cloudera and associated marks are trademarks or registered trademarks of Cloudera, Inc. All other company and product names may be trademarks of their respective owners.

Cloudera, Inc. | 5470 Great America Pkwy, Santa Clara, CA 95054 USA

Cloudera.com

6310-001 July 25, 2024