Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2021, 8(4):48-53

Research Article ISSN: 2394 - 658X

Performance Optimization with Spark and Delta Lake

Ravi Shankar Koppula

ravikoppula100@gmail.com

ABSTRACT

Apache Spark and Delta Lake have become essential tools for professionals in the field of data engineering and data science who are seeking to process and manage vast amounts of data. By optimizing the performance of these tools, it is possible to greatly improve the speed and efficiency of data processing tasks. In this paper, we will explore various techniques for optimizing performance in these technologies. Key strategies include minimizing the creation and impact of small files, utilizing data skipping and Z-Ordering for efficient data querying, employing clustering for ingest time data, and maintaining accurate table statistics for improved querying performance. Additionally, we will discuss storage and processing optimizations, improvements in query performance through partitioning and Z-Ordering, simplified handling of incremental data changes through Change Data Capture (CDC), and the role of the Delta Optimizer in automating crucial optimizations. By implementing these optimization techniques, data professionals can significantly enhance the performance of Apache Spark and Delta Lake, resulting in faster insights and increased overall productivity.

Key words: Spark Performance Optimization, Delta Lake Optimization, Dataframe optimization, Table optimization

---

INTRODUCTION

Apache Spark is a unified analytics engine for big data processing, with built-in modules for streaming, SQL, machine learning, and graph processing. It's built on the Hadoop Distributed File System (HDFS) and designed to be highly performant. Delta Lake is an open-source storage layer that brings reliability to data lakes. Delta Lake provides ACID transactions, scalable metadata handling, and unifies streaming and batch data processing. Existing Delta Lake REST APIs make it easy to convert existing applications using HDFS APIs to use Delta Lake that's offered to the community under the Apache 2.0 License. This document will cover different performance improvement topics in the context of Spark and Delta Lake, covering SQL tuning, best practices, and performance optimization capabilities and user stories. Spark and Delta Lake are widely used by Data Engineers and Data Scientists who want to analyze data, and it is very important to achieve faster results, especially in Ad-Hoc Analysis. This document will also help users understand best practices and optimize their data processing. ACID (Atomicity, Consistency, Isolation, Durability) consistency is one of the most important factors provided by Delta Lake. Performance tuning is the improvement of system performance. Typically, in computer systems, the primary goal of performance tuning is to increase the throughput of a system or to decrease the response time for the end-user. Additional goals include judicious use of resources or avoiding unnecessary over-engineering. By measuring equipment and system performance, these tuning changes are often made to the system but must always be backed up with evidence that the tuning changes result in better performance. In order to achieve the result, many elements could be involved, work like code improvement, removing bottlenecks, simulation and modeling, or configuration changes. Performance optimization is an important part of the processing work. Any time you can cut the time or the number of resources that a given set of computations requires, you've freed up that time and resources for other computations. [1]

Overview of Spark and Delta Lake

By using Delta Lake with Spark, this combination becomes a very powerful tool for data engineering and data science. With the ability to read and write numerous data formats and excellent developer APIs, Spark offers

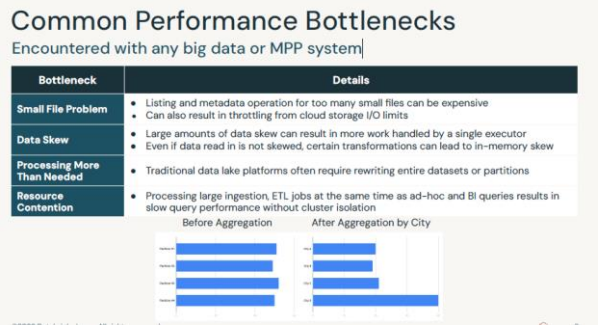
high amounts of data processing speed and flexibility to do countless tasks. This makes it a great tool for building machine learning models and data analysis. Delta Lake offers powerful auditing capabilities along with easy-to-create snapshots, which are ideal for machine learning workflows. Delta Lake's table versioning and schema evolution capabilities provide extra flexibility when working on data science projects. The reliability Delta Lake provides also brings peace of mind to the developer when running data engineering jobs that can take a long time to complete. The ability to rollback on any failed job is a luxury that is not often possible with other big data solutions.

Delta Lake is a storage layer that brings reliability to Data Lakes. It has SQL compatibility, so anything done in the past using Spark SQL's functionality can also be done. Its key features include ACID transactions, time travel, and schema enforcement. The way Delta Lake brings reliability to data lakes is by enabling the ability to have many batch and streaming in a single table, thereby reducing the need to create many different tables.

In simple words, Spark can be thought of as a distributed computational execution engine that was built to be faster and simple to use, which works with big data and Hadoop. It was developed in response to limitations in the Hadoop MapReduce cluster and was intended to increase the workload in response to hardware capabilities. The increase in web-scale data and data retention made it evident that a faster and more efficient way of working with big data was needed. Due to faster streaming of data and quicker computing, its use has expanded web general data processing to machine learning and many others. This fast speed and data processing is the main reason that when we talk about big data solutions, the first thought that comes to mind is Spark.[2]

Importance of Performance Optimization

There are several approaches that can be taken when addressing the need for performance optimization. In order to understand the need for performance optimization, a cost and benefit analysis should be conducted to determine if the level of optimization is worthwhile. If a query on a small data subset takes 5 minutes to compute, but the same query on the full dataset takes 10 hours and the user can get the results needed from the smaller query in a reasonable amount of time, there may be no need to optimize the larger query. However, if the query from the full dataset is something that will need to be computed several times and is critical to the business, the extra time spent optimizing the query may be well worth it. It is also important to understand what the expectations for scale are from the start of your data project. Listed below are the common performance bottlenecks.



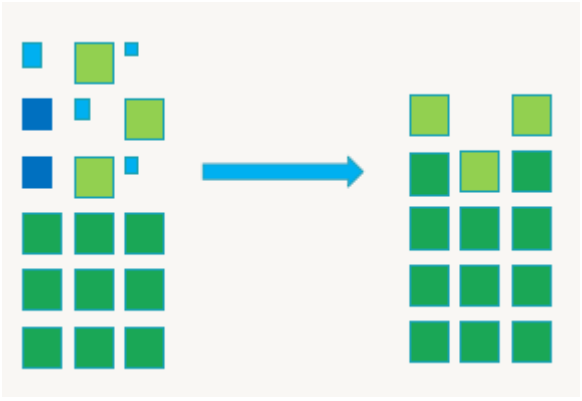
Delta Lake is intended to be used on big data ranging from terabytes to petabytes of data with millions of files. Under that level of scale, it is likely that performance tuning will become a necessity at some point in the project's lifetime. Finally, it is important to understand that there can be many ways to optimize a Spark job and that optimization is not a one-size-fits-all proposition. Data engineers will need to weigh the trade-off of system complexity vs performance gain and will likely need to iterate on these changes as the project evolves.

Stop Small Files from Slowing Your Data Lake:

Databricks Delta Lake addresses the issue of "small file syndrome" commonly experienced in traditional data lakes. This syndrome occurs when a large quantity of minuscule files leads to bottlenecks in performance.

Enhanced Reading Efficiency: Databricks dynamically optimizes the size of Delta Lake tables, eradicating the need to choose between reading speed and parallelism. There is no longer a concern of read performance being hindered by an excessive number of small files or limited parallel processing due to a scarcity of large files.

Automated Condensing of Small Files: Databricks' auto-optimize function automatically consolidates small files during the writing process, thereby decreasing storage overhead and enhancing the efficiency of subsequent read operations. This guarantees that your data lake remains streamlined and operates at its peak potential. [3]



Faster Queries with Data Skipping:

Data skipping is a highly effective technique that significantly reduces processing time by minimizing the amount of data scanned during queries. Let me explain how it works in a more formal tone: Firstly, Delta Lake offers Smart File Selection which automatically keeps track of statistics for each data file. These statistics include the minimum and maximum values for key columns. By analyzing these statistics, queries can identify files that are unlikely to contain relevant data based on the filter criteria, allowing them to be skipped entirely. Additionally, Delta Lake automatically gathers file-level statistics for the first 32 columns in your table. It is important to note that frequently used join columns should be among these first 32 columns in order to ensure efficient skipping. Alternatively, you have the option to adjust the number of columns for which statistics are collected. To further enhance data skipping, Delta Lake utilizes a technique called Z-Ordering. This technique is similar to indexing in relational databases but takes it a step further. Z-Ordering utilizes a multi-dimensional clustering approach, unlike traditional sorting methods. This enables even more effective data skipping, especially for queries with complex filters. [4]

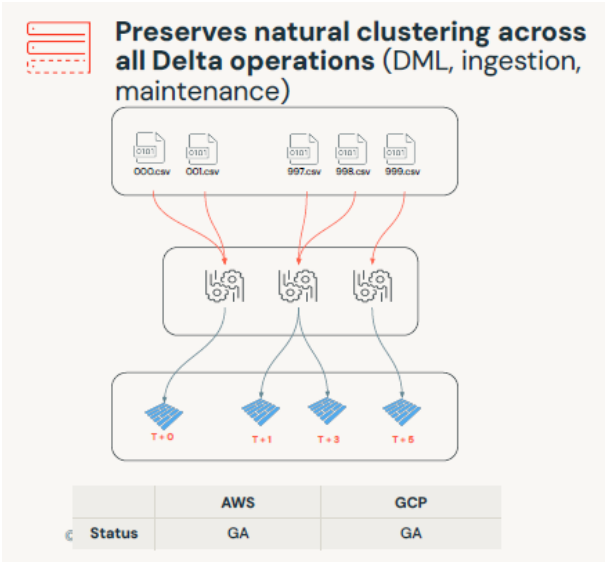
```
SELECT * FROM table WHERE col < 5
                ⬇
SELECT file_name FROM index
WHERE col_min < 5
```

file_name	col_min	col_max
file1.csv	6	8
file2.csv	3	10
file3.csv	1	4

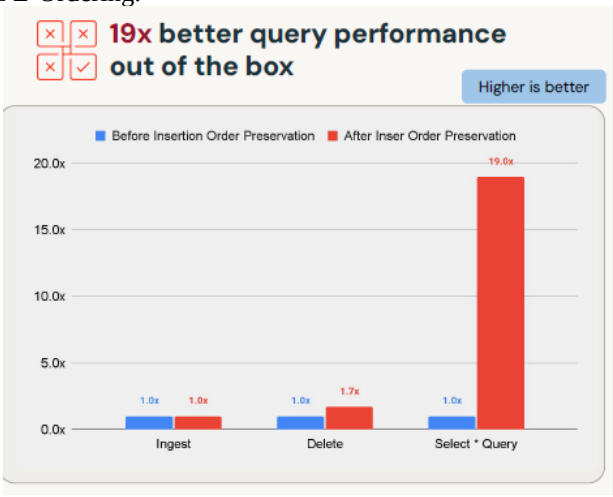
EFFORTLESS DATA SKIPPING WITH INGESTION TIME CLUSTERING:

Delta Lake offers a powerful built-in feature called Ingestion Time Clustering. This eliminates the need for manual partitioning or Z-Ordering to achieve data skipping benefits. Here's what makes it so advantageous:

- **Automatic Optimization:** Ingestion Time Clustering automatically groups data based on the time it's ingested into the lake. This natural clustering allows queries to efficiently skip irrelevant files during read operations.
- **No Additional Configuration:** Unlike traditional methods, Ingestion Time Clustering requires no complex setup or configuration. It works seamlessly "out of the box" with your Delta Lake tables.
- **Maintained Across Operations:** The natural clustering established by Ingestion Time Clustering persists across all Delta Lake operations, including data manipulation (DML), data ingestion, and data maintenance tasks. This ensures consistent performance benefits over time.



Databricks Delta Lake boasts **up to 19x faster query performance** out of the box, thanks to features like automatic data skipping and Z-Ordering.



Store Sales table with data naturally ordered by date

Understanding Table Statistics with Databricks

This table explains how keeping table statistics up-to-date improves query performance in Databricks using the Cost Based Optimizer (CBO).

Feature	Benefit	Description
Table Statistics	Improves query performance	Analyzes and stores data about your table's columns, including: * Number of rows * Minimum and maximum values * Data distribution
Cost Based Optimizer (CBO)	Chooses the most efficient execution plan	Leverages table statistics to estimate the cost of different query plans and selects the most efficient one.
Benefits	* Faster queries * More efficient resource utilization * Improved decision-making through faster data insights	

How Table Statistics Help

- Adaptive Query Execution (AQE): CBO uses statistics to adapt the query plan during execution, potentially leading to further performance gains.
- Join Optimization: CBO uses statistics to choose the optimal join type (e.g., inner join, left join) and select the most efficient build side for hash joins.
- Multi-way Join Ordering: CBO leverages statistics to determine the best order for processing multiple joins in a complex query.

Keeping Statistics Up-to-Date:

To ensure optimal performance, run the following command to collect statistics on all columns in your table:

SQL

```
ANALYZE TABLE mytable COMPUTE STATISTICS FOR ALL COLUMNS;
```

STREAMLINE YOUR DATA LAKE WITH DATABRICKS DELTA LAKE

This guide outlines key practices to optimize your data lake using Databricks and Delta Lake. By leveraging these features, you can achieve efficient data processing, faster queries, and minimized maintenance overhead.

Optimize Storage and Processing:

- Let Delta Lake Handle File Management: Delta Lake automatically tunes file size and compacts small files on write, eliminating the need for manual intervention and reducing storage overhead.
- Adaptive Query Execution (AQE) to the Rescue: Delta Lake's AQE automatically detects and handles skewed data, ensuring efficient query execution even with uneven data distribution.
- Natural Sort Order Benefits: For tables under 1 TB, Delta Lake preserves the natural sort order of data, eliminating the need for manual partitioning.

Boost Query Performance:

Z-Ordering for Efficient Data Skipping: Leverage Z-Ordering to significantly reduce data scanned during queries. Regularly create and maintain Z-Order indexes on high-cardinality columns frequently used in filters (consider a weekly maintenance job).

Table Statistics for Smarter Queries: Collect and update table statistics, especially for columns used in joins (consider a weekly maintenance job). This empowers the Cost Based Optimizer to choose optimal query execution plans.

Partitioning for Large Tables (Over 1 TB): For very large tables, consider partitioning data based on low-cardinality columns often used in filters (e.g., year, month, day) to further enhance data skipping.[5]

Embrace Change Data Capture (CDC):

Leverage Delta Lake's SQL DML: Utilize Delta Lake's SQL DML capabilities to implement a Change Data Capture (CDC) architecture. This allows you to process only the incremental data changes, streamlining processing and reducing resource consumption.[6]

Delta Optimizer: Streamlining Table Optimization

Tired of manually tuning Delta tables for optimal performance? Introducing Delta Optimizer, a field-managed tool that automates essential optimizations, saving you valuable time and effort.

How it Works:

- **Leveraging Query History and Data:** Delta Optimizer analyzes your query history and Delta transaction logs to build a comprehensive data profile for each table. This profile identifies the most critical columns for Z-Ordering, a technique that significantly accelerates queries.
- **Reducing Manual Work:** Delta Optimizer automates the process of discovering and applying the best Z-Ordering configurations. This is particularly beneficial when working with Delta tables primarily accessed through DBSQL Warehouse (analysts using SQL or BI tools with auto-generated SQL). By reducing the need for manual configuration, Delta Optimizer empowers analysts to focus on extracting insights from their data.

Key Benefits:

- Drastically reduced manual tuning: Free yourself from the time-consuming task of manually discovering optimal Z-Ordering configurations.
- Automated optimization: Delta Optimizer takes care of the heavy lifting, automatically applying optimizations based on your specific usage patterns.
- Improved query performance: Get the most out of your Delta tables with significantly faster queries, allowing you to access insights quicker.

CONCLUSION

In conclusion, leveraging the capabilities of Apache Spark and Delta Lake can significantly enhance the performance and reliability of big data processing and analysis. The key to achieving high performance in complex big data environments lies in optimization strategies that focus on effective data storage management and the implementation of advanced techniques such as data skipping and statistics. Databricks Delta Lake, with its powerful optimizations, plays a vital role in automating file management, reducing storage overhead, and improving query performance through data skipping, Z-Ordering, and comprehensive table statistics. The Delta Optimizer further simplifies the optimization process, making it highly efficient by utilizing query history and sophisticated data profiles to automate Z-Ordering, thus significantly reducing the need for manual tuning. By adhering to best practices, including harnessing the automatic file management capabilities of Delta Lake, maintaining accurate table statistics, and carefully considering partitioning strategies for large tables, as well as embracing the revolutionary Change Data Capture (CDC) technique for incremental data processing, organizations can achieve unparalleled optimized performance in their big data environments. This holistic approach to optimization not only ensures quicker and more reliable data processing and analysis but also facilitates a more efficient and productive utilization of resources in the ever-evolving era of big data.

REFERENCES:

[1]. Introduction to Apache Spark: A Unified Analytics Engine - Learning Spark, 2nd Edition [Book]," www.oreilly.com. https://www.oreilly.com/library/view/learning-spark-2nd/9781492050032/ch01.html

[2]. M. Kukreja, "Databricks Delta Lake — Database on top of a Data Lake," Medium, Sep. 02, 2020. https://towardsdatascience.com/databricks-delta-lake-database-on-top-of-a-data-lake-fbc45eab3841

[3]. "What are the challenges with data lakes?," Databricks, Mar. 21, 2020. https://www.databricks.com/discover/data-lakes/challenges

[4]. "How to optimize and increase SQL query speed on Delta Lake," Databricks, 2020. https://www.databricks.com/blog/2020/04/30/faster-sql-queries-on-delta-lake-with-dynamic-file-pruning.html

[5]. "How to Speed up SQL Queries with Adaptive Query Execution," Databricks, May 29, 2020. https://www.databricks.com/blog/2020/05/29/adaptive-query-execution-speeding-up-spark-sql-at-runtime.html

[6]. A. R. R. is the T. Director et al., "Databricks Performance: Fixing the Small File Problem with Delta Lake • Data-Driven," data-driven.com, Aug. 24, 2020. https://data-driven.com/2020/08/databricks-performance-fixing-the-small-file-problem-with-delta-lake/