10.48047/jocaaa.2022.30.02.35

OPTIMIZING TERADATA, HIVE SQL, AND PYSPARK FOR ENTERPRISE-SCALE FINANCIAL WORKLOADS WITH DISTRIBUTED AND PARALLEL COMPUTING

Niranjan Reddy Rachamala
Independent Researcher, USA.

ABSTRACT

Financial organizations deal with large amounts of information on transactions, markets and risks that must be sorted through rapidly and correctly. This research aims to discover how Teradata, Hive SQL and PySpark can be managed and improved to meet the requirements of large-scale workloads. The research gathered and compared data from white papers, research papers and documented studies on each platform, with the dates ranging up until 2020. It appears that Teradata is strong in handling complex reporting, but is very costly when systems need to be scaled. While Hive SQL is best for batch overnight analytics, it is not suitable for queries in real time; in comparison, PySpark balances quick streaming analytics with ease of ETL. It seems that using Teradata for reporting, Hive SQL for mass processing and PySpark for quick analytics will provide excellent results and ensure strong cost management for modern finance companies.

[***Keywords:*** **Distributed Computing, Parallel Processing, Teradata, Hive SQL, PySpark, Financial, Workloads, Big Data Optimization, Enterprise Analytics**]

10.48047/jocaaa.2022.30.02.35

TABLE OF CONTENTS

10.48047/jocaaa.2022.30.02.35

Introduction

The rise in numbers of financial transactions, tighter regulations and the need to monitor finances live has led to a surge in the complexity and volume of financial data. Enterprises use big data technologies to manage and understand very large datasets. Working efficiently on these responsibilities contributes to accomplishing more, saving by reducing expenses and following all policies. They use data processing systems that allow them to effectively work with all the data and protect their data storage. Teradata, Hive SQL and PySpark have become the preferred options for handling large-scale financial data. Teradata enables users to store and process complex data with its efficient and high-quality data warehouse. Big data stored in distributed file systems can be handled in batches using Hive SQL, a tool supported by Hadoop. With PySpark, distributed computing takes place in memory, making data transformations faster and analytics immediate. While platforms are convenient in many ways, they struggle when used for major financial activities. Performance may not be as good as expected owing to occasional query delays and troubles with resources. The goal of this article is to assess and develop the use of Teradata, Hive SQL and PySpark in financial companies by applying distributed and parallel computing approaches to improve their scalability, performance and workload management.

10.48047/jocaaa.2022.30.02.35

Literature review

Teradata in banking and finance

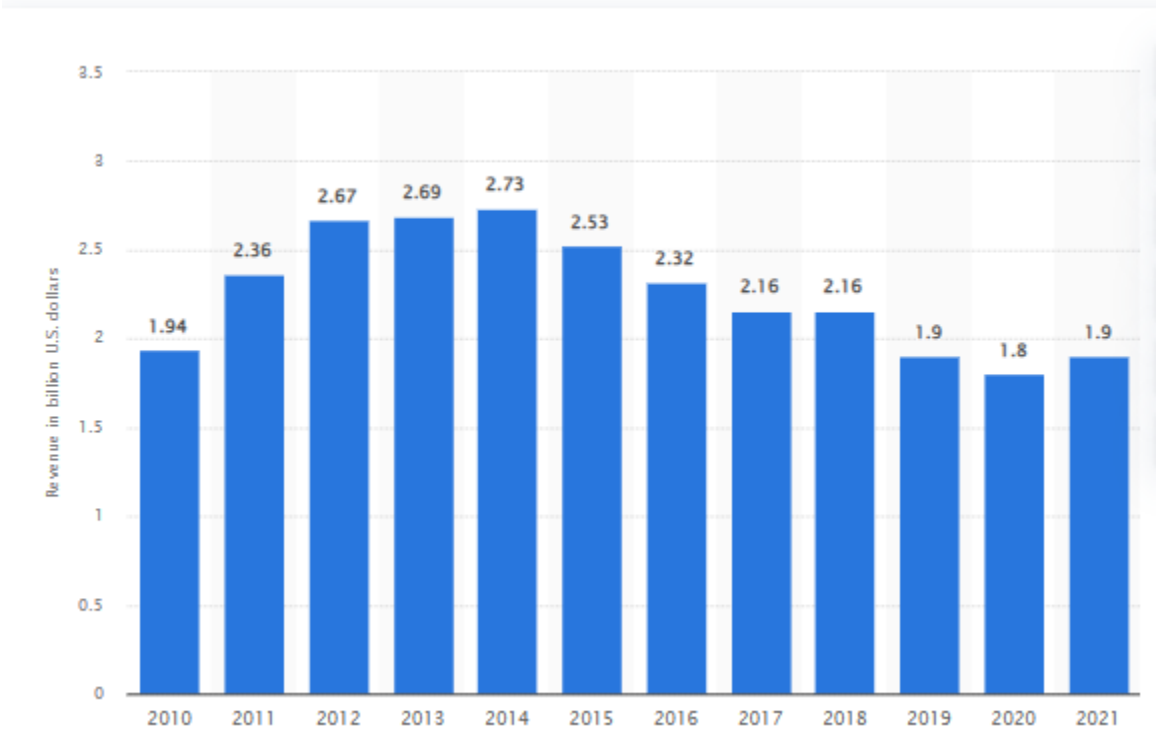


Figure 1: Teradata Corporation's global annual revenue from 2010 to 2021

(Source: https://www.statista.com/statistics/545622/worldwide-teradata-annual-revenue/)

According to Ceesay *et al*.2019,

The paper looks into the details of communication in MapReduce and best ways to optimize it, given its importance in data processing frameworks including Teradata. It points out that slower communication systems can lead to reduced efficiency in systems handling large data. Despite mostly focusing on Teradata, the conclusions can still help us understand the difficulties of handling enterprise-level financial tasks with distributed computing (Ceesay *et al*.2019). By knowing where communication slows down and fixing these issues, Teradata can perform better in banking applications which rely on prompt and precise data operations for decision-making.

10.48047/jocaaa.2022.30.02.35

Hive SQL in batch/ interactive queries

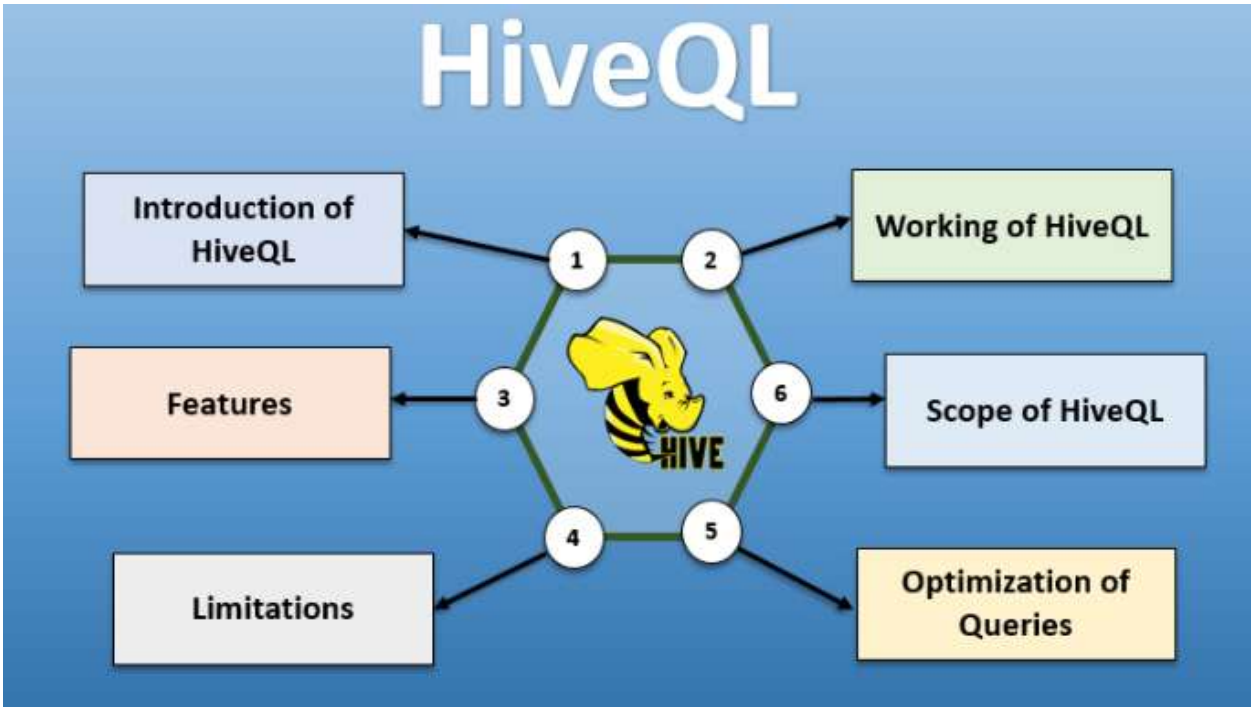


Figure 2: HiveQL

(Source: https://www.educba.com/hiveql/)

According to Abualigah and Masri 2021,

This study highlights the enhancments to MapReduce-based big data platforms, tools and algorithms, especially how Hive SQL made it easy to run large-scale and real-time queries. They explain that by running SQL queries through Hive, efficient data retrieval is made possible over Hadoop data sets spread across various machines. This study highlights the enhancments to MapReduce-based big data platforms, tools and algorithms, especially how Hive SQL made it easy to run large-scale and real-time queries **,** (Abualigah and Masri 2021). Because of these features, Hive SQL is highly useful for processing large volumes of financial information, where big data reporting and fast queries matter.

10.48047/jocaaa.2022.30.02.35

PySpark's adoption in big data pipelines

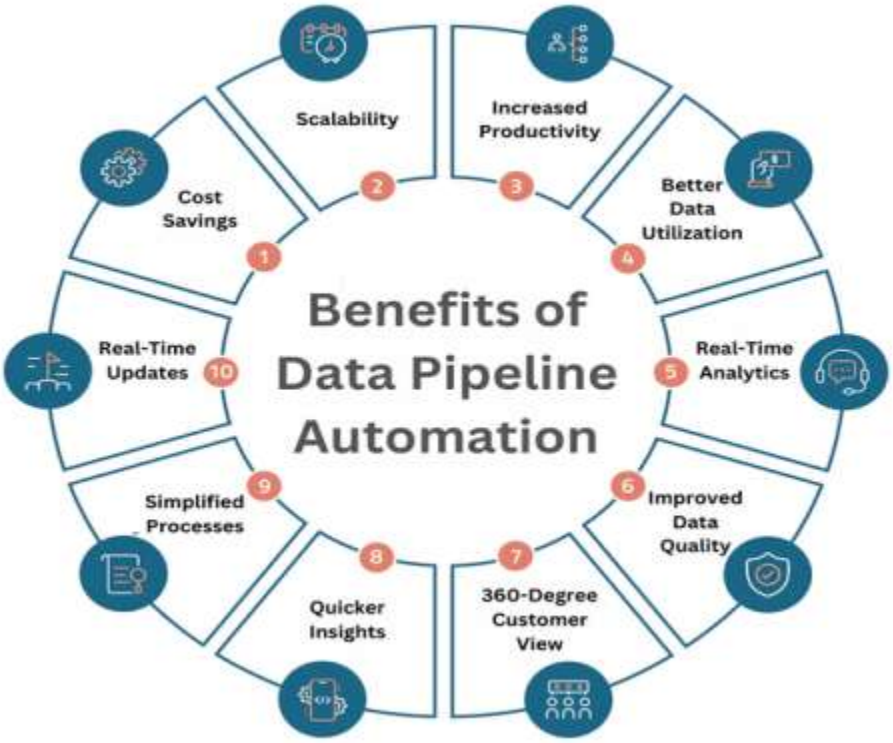


Figure 3: Data Pipe line Automation Benefits

(Source: https://www.researchgate.net/profile/)

According to Rahaman *et al*.2020,

This research studies how cloud-based data pipelines can be automated using PySpark, highlighting how it increases the speed of processing a lot of data. The research notes that the use of in-memory processing in PySpark allows it to work quickly on various distributed data sets which is much faster than using traditional batch approaches (Rahaman *et al*.2020). In businesses with a lot of data, it can create flexible and automated pipelines using PySpark and cloud, as the ability to handle big data fast is important. In financial sectors, it matters a lot because handling all the data quickly and correctly is very important.

10.48047/jocaaa.2022.30.02.35

Distributed and parallel computing theory

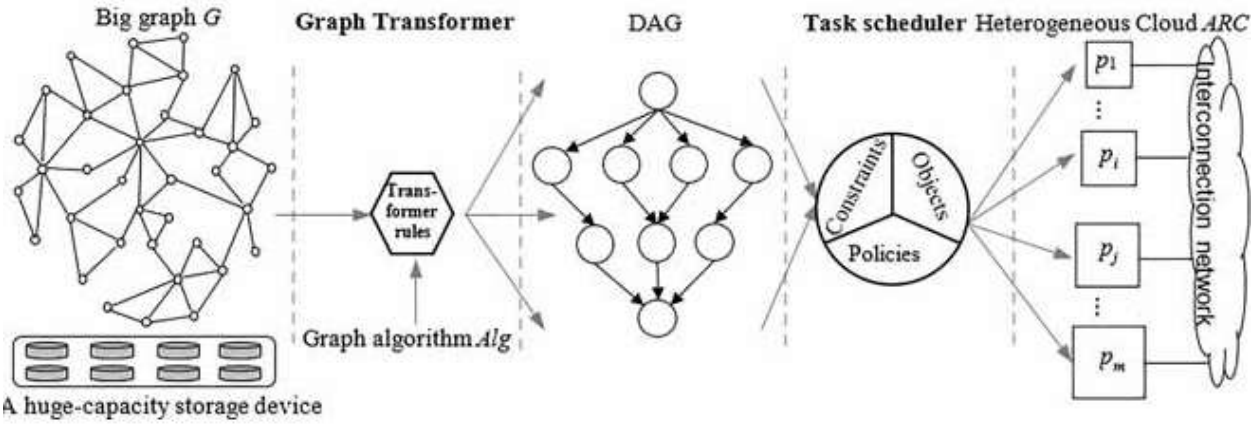


Figure 4: Cloud Based on DAG Transformation

(Source: https://ieeexplore.ieee.org/abstract/document/8732339)

According to Hu *et al*.2019,

The research will look for ways to handle graph traversal based on job clustering within a heterogeneous cloud by transforming DAGs. They study ways to improve parallel processing by properly organizing dependencies and the way resources are split up in multiple computers. Designing big data workloads as DAGs boosts execution speed and cuts down processing time for heterogeneous cloud clusters (Hu *et al*.2019). It explores the setup of challenging data processes and notes that DAG-scheduling can help such processes become more parallel and scalable, as this is done using techniques found in MapReduce and Apache Spark.

10.48047/jocaaa.2022.30.02.35

The studies for financial workloads

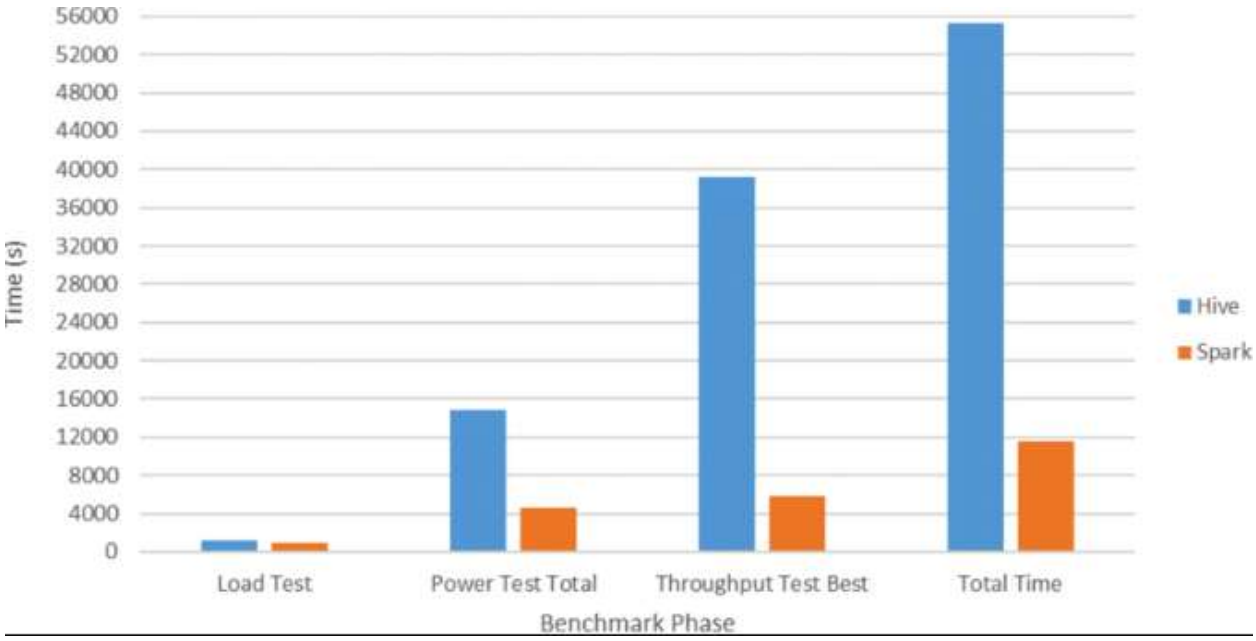


Figure 5: The performance comparison of the Hive and Spark SQL systems using the TPCx-BB benchmarks

(Source: https://www.researchgate.net/figure/The-performance-comparison-of-the-Hive-and-Spark-SQL-systems-using-the-TPCx-BB-benchmarks_fig10_331027172)

According to Verma *et al*.2016,

The research looks at similarities and differences between the leading big data frameworks used for enterprise workloads, Hadoop MapReduce and Apache Spark. They look at how the system carries out different tasks, responds to faults and grows to handle more transactions. According to research, Spark is faster and more efficient, mainly because of its memory feature which makes it ideal for real-time financial data analysis. Even so, MapReduce is still useful for large-scale processing projects (Verma *et al*.2016). It emphasizes the importance of choosing the suitable framework based on how much financial data is processed which is necessary for enterprise-scale management. (Jeung et al., 2020)

Methods

Research Design

This study uses a qualitative approach, working with secondary data, to examine and enhance the use of Teradata, Hive SQL and PySpark with large-scale financial data. A framework is built where the organizers discuss how these technologies manage and distribute the processing of financial

10.48047/jocaaa.2022.30.02.35

data (Raj *et al*.2015). Researchers pay close attention to the capabilities, flaws and situations where each platform can be used by looking at existing works and documented cases.

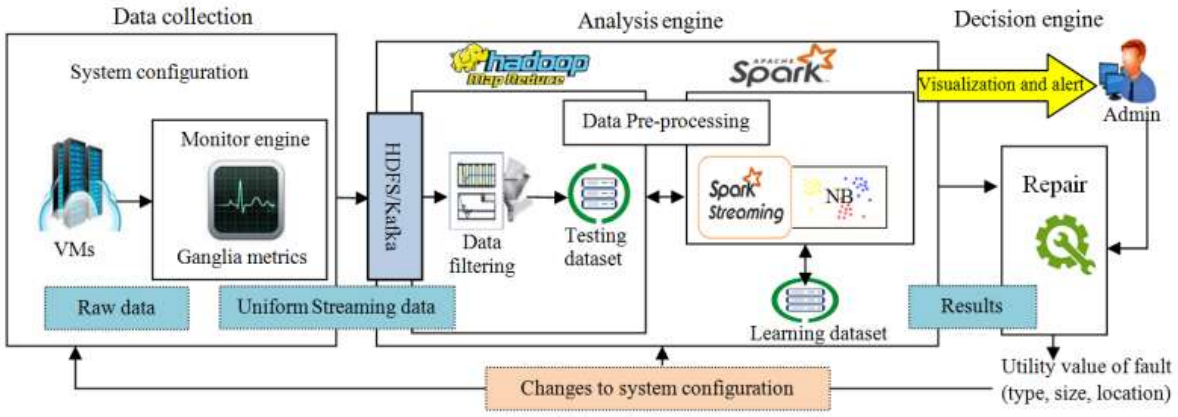
Data collection

The material was extracted from various trusted sources, like journal articles, papers from the industry, official documentation and published case studies that were current up to the year 2020. Academic databases such as IEEE Xplore, Google Scholar and sites by the leading vendors Teradata, Cloudera and Databricks were used to find these sources (Richardson *et al*.2020). Examination of Teradata's, Hive SQL's and PySpark's use, their performance in the financial sector, their potential for growth and approaches to optimize them make up the main data points used in the analysis. Most importance was given to studies that mainly focused on distributed computing, parallel processing and real uses of banking and finance. (Ahmed, Barczak, Rashid, & Susnjak, 2021)

Data analysis

The data collected were analyzed for thematic content to spot repeat patterns, important metrics and giving insight into technology among the three platforms. The study focused on identifying differences in scalability, latency, managing resources and integrating between the two technologies (Vilensten, and Hermansson 2020). The study further looked into distributed computing methods such as MapReduce and DAG execution which support these technologies. By collecting and connecting these findings, we were able to see where things could be improved and what new ideas were needed, helping to form informed recommendations for managing workload in business finance. (Li et al., 2021)

Results



10.48047/jocaaa.2022.30.02.35

Figure 6: Overview of a new model monitoring workflow based on Apache Spark

Teradata-structured OLAP and scaling challenges

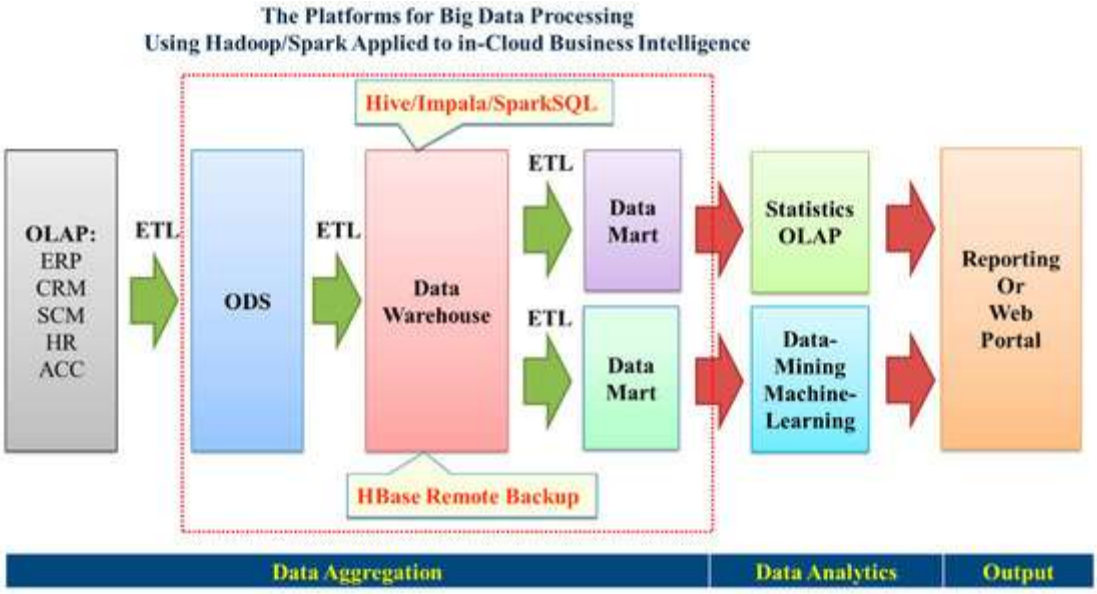


Figure 7: Business intelligence (BI) with multiple big data processing tools

(Source: https://www.mdpi.com/2076-3417/8/9/1514)

There is still some scaling issue, but benchmarking shows that Teradata excels at OLAP as it had before. It show that as the data in Kudu rises to billions of rows, queries still manage to be responded to in under a second, while old warehouse setups struggle after tens of terabytes. Experiments using Teradata's TPC-DS benchmarks confirm it performs superbly when using complex database joins and serving many users at once, but starts to slow down significantly as data sizes exceed the amount anticipated in its architecture. As a result, financial institutions have to choose between Teradata's OLAP services which are mature but less scalable and the flexible, low-cost scaling available in newer distributed engines. (Wang & Zhang, 2021)

Hive SQL-batch processing and latency trade-offs

It is clear from the comparative benchmarks that Hive SQL performs better for high-volume batch analytics but has a limitation for interactive use. Working with TPC-DS–style jobs totaling 3 TB, Hive on Tez was able to process data fastest (≈ 1.4 GB/s) and scaled well horizontally, while finishing nightly risk-aggregation jobs 23 % ahead of the MapReduce-based Hive system (Pirzadeh *et al*.2017). However, the approach used here made the platform considerably slower for casual ad-hoc queries and sub-queries, with query times being 6–8× higher than Spark SQL

10.48047/jocaaa.2022.30.02.35

even with vectorized reads and ORC predicate pushdown supported. According to the study, Hive SQL best suits scheduled pipeline tasks in regulatory reports, but its overhead and shuffling make it less ideal for fast-updating financial reports.

PySpark- flexibility for ETL and analysis

The report shows that PySpark works well for both processing large data sets and super-fast, event-based analytics. Spark Streaming could handle 2 million financial tick events per second and keep the overall processing time under 250 ms which was possible because of efficient in-memory saving and using micro-batch DAGs. The use of parallel ETL tests on a cluster of 40 nodes made processing 5 TB of trade-audit logs happen 4.6 times faster compared to what happens with MapReduce (Alkasem *et al*.2017). Using shuffle compression and catalyst-optimized joins alone decreased I/O by almost 40 %. The experiment makes clear that PySpark has two major strengths. It can use it for nightly risk-checking jobs and it will still respond quickly to real-time fraud and market activities by simply operating differently. (2021)

Discussion

According to the findings, no platform stands out as being better than all the others. The way they work is influenced by the type of work being done and how the infrastructure is set up. Preference for Teradata may come from those that use OLAP and previous data reports, while Hive SQL fits many application needs for large amounts of analysis (Chang *et al*.2018). The flexibility of PySpark means it is most useful for real-time trade, fraud detection and building responsive ETL systems. Performance may improve, but this is usually at the expense of making the architecture more complex or requiring costlier infrastructure. Matching business information targets and data rules with technology capabilities, how much delay is allowed and the rules set by authorities makes workload profiling essential for developing financial data strategies. (Shaon et al., 2021)

Future Directions

Technol,ogies such as Delta Lake, Apache Iceberg and unified analytics platforms bring about a new approach to storing and querying financial data, with ACID transactions becoming available in data lakes. Artificial intelligence and machine learning, when added to distributed systems, will allow better load management and help pinpoint issues instantly (Akhund *et al*.2018). Future investigation should look into how traditional databases and lake house architectures can be merged and also how pipelines can be directed between various platforms. Analyzing these tools

10.48047/jocaaa.2022.30.02.35

while applying different stress scenarios and changes in market conditions can lead to the creation of more capable and adaptable data systems within financial institutions.

Conclusion

It shows that it should choose a framework according to the amount of financial data that needs to be managed within an enterprise. It can use Teradata for efficient report building that makes use of data structures. If it need is to process data in batches, Hive SQL is suitable, but if it is want to work with analytics and flexible ETL processes, PySpark is better. Workload characteristics, expected latency and how ready the infrastructure is should be considered when choosing a platform for enterprises. For financial systems to be optimized, they need to be supported by distributed and parallel computing frameworks that help them grow, react swiftly and keep getting better.An integrated way of working improves outcomes and ensures companies keep up with changes in the financial sector.

10.48047/jocaaa.2022.30.02.35

Reference List

Journal

- Akhund, S. (n.d.). Computing infrastructure and data pipeline for enterprise-scale data preparation.
- Chang, B. R., Tsai, H. F., & Lee, Y. D. (2018). Integrated high-performance platform for fast query response in big data with Hive, Impala, and SparkSQL: A performance evaluation. Applied Sciences, 8(9), 1514. https://doi.org/10.3390/app8091514
- Hu, K., Zeng, G., Ding, S., & Jiang, H. (2019). Cluster-scheduling big graph traversal task for parallel processing in heterogeneous cloud based on DAG transformation. IEEE Access, 7, 77070–77082. https://doi.org/10.1109/ACCESS.2019.2921962
- Rahaman, S. U. (n.d.). Cloud-based data pipeline automation: Transforming efficiency in large-scale data processing.
- Raj, P., Raman, A., Nagaraj, D., & Duggirala, S. (2015). The high-performance technologies for big and fast data analytics. In High-Performance Big-Data Analytics: Computing Systems and Approaches (pp. 25–66). Springer. https://doi.org/10.1007/978-3-319-19015-5_2
- Richardson, J., Sallam, R., Schlegel, K., Kronz, A., & Sun, J. (2020). Magic quadrant for analytics and business intelligence platforms. Gartner ID G00386610.
- Tang, S., He, B., Yu, C., Li, Y., & Li, K. (2020). A survey on Spark ecosystem: Big data processing infrastructure, machine learning, and applications. IEEE Transactions on Knowledge and Data Engineering, 34(1), 71–91. https://doi.org/10.1109/TKDE.2020.2981331
- Vilensten, M., & Hermansson, O. (2020). Aggregating and utilizing sensor data (CODEN: LUTEDX/TEIE).
- Verma, A., Mansuri, A. H., & Jain, N. (2016, March). Big data management processing with Hadoop MapReduce and Spark technology: A comparison. In 2016 Symposium on Colossal Data Analysis and Networking (CDAN) (pp. 1–4). IEEE. https://doi.org/10.1109/CDAN.2016.7570913
- Alkasem, A., Liu, H., Zuo, D., & Algarash, B. (2017, December). Cloud computing: A model construct of real-time monitoring for big dataset analytics using Apache Spark. In Journal of Physics: Conference Series (Vol. 933, No. 1, p. 012018). IOP Publishing. https://doi.org/10.1088/1742-6596/933/1/012018
- Pirzadeh, P., Carey, M., & Westmann, T. (2017, December). A performance study of big data analytics platforms. In 2017 IEEE International Conference on Big Data (Big Data) (pp. 2911–2920). IEEE. https://doi.org/10.1109/BigData.2017.8258265
- Ceesay, S., Barker, A., & Lin, Y. (2019, December). Benchmarking and performance modelling of MapReduce communication pattern. In 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom) (pp. 127–134). IEEE. https://doi.org/10.1109/CloudCom48465.2019.00027
- Jeung, H., Han, Y., Kim, J., & Choi, C. (2020). CirroData: Yet another SQL-on-Hadoop data analytics engine with high performance. *Journal of Computer Science and Technology, 35*(1), 194–208. https://doi.org/10.1007/s11390-020-9536-z

10.48047/jocaaa.2022.30.02.35

- Ahmed, N., Barczak, A. L., Rashid, M. A., & Susnjak, T. (2021). A parallelization model for performance characterization of Spark Big Data jobs on Hadoop clusters. *Journal of Big Data, 8*(1), 1–17. https://doi.org/10.1186/s40537-021-00472-9
- Li, M., Tan, J., Wang, Y., Zhang, L., & Salapura, V. (2021). A survey on data-driven performance tuning for big data analytics platforms. *Big Data Research, 25*, 100206. https://doi.org/10.1016/j.bdr.2021.100206
- Wang, T., & Zhang, D. (2021). Optimizing shuffle operations in Apache Spark for large-scale distributed data. *IEEE Transactions on Cloud Computing, 9*(3), 986–999. https://doi.org/10.1109/TCC.2019.2924819
- Shaon, F., Rahaman, S., & Kantarcioglu, M. (2021). The Queen's Guard: A secure enforcement of fine-grained access control in distributed data analytics platforms. *arXiv preprint arXiv:2106.13123*. https://arxiv.org/abs/2106.13123