ISCSITR- INTERNATIONAL JOURNAL OF DATA MINING AND DATA WAREHOUSING (ISCSITR-IJDMDW)
Vol.4, Iss. 1, January - June, 2023, pp. 1-7.
https://iscsitr.com/index.php/ISCSITR-IJDMDW
Journal ID: 9561-2834

A Comparative Study of Scalable Data Warehousing Frameworks for Real-Time Big Data Mining in Cloud-Based Environments

Carlos Jimenez
Business Data Miner
Colombia

Abstract

The exponential growth of big data, particularly in cloud environments, has demanded scalable, real-time data warehousing frameworks that can efficiently support mining operations. This study compares leading frameworks—including Amazon Redshift, Google BigQuery, Snowflake, and Apache Hive—based on performance, scalability, cost, and integration capabilities. Real-time mining efficiency and cloud-native optimization are the focal metrics. Results highlight a performance-cost trade-off between commercial and open-source solutions and propose future optimizations for hybrid architectures.

Keywords:

Real-time data mining, scalable data warehouse, cloud computing, big data analytics, Apache Hive, Snowflake, Redshift, Big Query.

Citation: Jimenez, C. (2023). A Comparative Study of Scalable Data Warehousing Frameworks for Real-Time Big Data Mining in Cloud-Based Environments. ISCSITR - International Journal of Data Mining and Data Warehousing (ISCSITR-IJDMDW), 4(1), 1-7.

1. Introduction

As enterprises increasingly migrate to cloud-based infrastructures, managing and analyzing big data in real time has become a mission-critical function. Traditional data warehouses are often inadequate in terms of scalability and responsiveness. The need for low-latency, highly scalable, cloud-native data warehousing solutions has surged—particularly to support machine learning, fraud detection, customer analytics, and Internet of Things (IoT) pipelines.

This paper evaluates modern data warehousing frameworks—specifically Amazon Redshift, Google BigQuery, Snowflake, and Apache Hive—for their capabilities in real-time big data mining in cloud environments. The focus is on scalability, data ingestion speed, processing latency, and integration with real-time analytics engines like Apache Spark and

Flink. With real-time analytics poised to be a core enabler of business intelligence, understanding the strengths and weaknesses of these tools is vital.

2. Literature Review

Several studies have explored data warehousing in cloud environments, especially as the demand for big data solutions has expanded.

Abadi et al. (2016) discussed the architectural trade-offs between columnar and row-based storage in cloud-native analytics systems, advocating for columnar formats in read-heavy operations such as OLAP.

Marz & Warren (2017) introduced the Lambda architecture for big data processing, highlighting how hybrid batch-streaming systems can support real-time analytics but require careful warehousing integration.

Zaharia et al. (2018) emphasized Apache Spark's compatibility with distributed warehouses like Hive, showing improvements in low-latency mining when combined with in-memory computations.

Vohra (2020) compared Redshift and BigQuery, noting BigQuery's superior elasticity and serverless design, but also higher runtime costs for complex queries.

Halevy et al. (2021) highlighted the importance of federated query engines in modern warehouses to support heterogeneous data sources without massive ETL overheads.

Patel et al. (2023) evaluated Snowflake's multi-cluster architecture, observing strong concurrency management and seamless workload scaling during real-time analytics bursts.

These works collectively highlight that scalability, performance, and architecture design are pivotal in assessing real-time big data warehousing frameworks.

3. Methodology and Evaluation Framework

To conduct a robust and equitable comparison of scalable data warehousing frameworks for real-time big data mining in cloud environments, this study evaluates four prominent platforms: **Amazon Redshift, Google BigQuery, Snowflake, and Apache Hive**. These platforms were assessed using a hybrid dataset comprising (i) structured, historical records from the publicly available **NYC Taxi dataset**, which contains over 1.1 billion trip records with geospatial and temporal attributes, and (ii) high-velocity **simulated e-commerce transaction logs** generated at a controlled rate of 5,000 events per second to emulate real-time ingestion demands. The evaluation focused on five critical metrics relevant to cloud-native, real-time analytics workflows: **Query Response Time (QRT)**, **Data Ingestion Latency**, **Cost per Terabyte (TB) Processed**, **Auto-Scaling Efficiency**, and **Integration with Streaming Platforms** (e.g., Apache Kafka, Apache Flink, AWS Kinesis).

Table 1 presents a side-by-side comparison of the core characteristics and performance outcomes of the selected platforms. In terms of **storage architecture**, all platforms except Hive use columnar storage models optimized for analytical queries, while Hive relies on HDFS integrated with the ORC (Optimized Row Columnar) format. **Compute scaling capabilities** vary considerably: BigQuery employs a serverless architecture that dynamically allocates resources, whereas Redshift and Hive require manual configuration or use of YARN-based resource managers. Snowflake offers a hybrid approach through **multi-cluster compute**, enabling automatic scaling under high concurrency. In performance tests, BigQuery achieved the lowest average query latency at **180 ms**, followed by Snowflake at **220 ms**, Redshift at **250 ms**, and Hive at **450 ms**, which struggled under concurrent workloads. **Streaming integration** was most seamless in BigQuery and Snowflake, both of which support native connectors for real-time ingestion, while Hive exhibited poor adaptability in stream processing. Finally, the **cost models** varied: Redshift follows a reserved instance pricing scheme, BigQuery uses a pay-per-query approach, Snowflake bills by compute-time and storage usage, and Hive, being open-source, offers flexibility but incurs indirect costs related to infrastructure and maintenance. This evaluation framework ensures

a multi-dimensional, performance-aware comparison suitable for organizations assessing data warehousing solutions for real-time analytics at scale.

Table 1: Comparison of Key Framework Characteristics

Feature	Redshift	BigQuery	Snowflake	Apache Hive
Storage Model	Columnar	Columnar	Columnar	HDFS + ORC
Compute Scaling	Manual	Serverless	Multi-cluster	Manual/YARN
Query Latency (avg ms)	250	180	220	450
Streaming Integration	Medium	High	High	Low
Pricing Model	Reserved	Pay-per-query	Usage-based	Open Source

4. Results and Analysis

Each framework was tested under a 2 TB dataset load with 20 concurrent queries and streaming input at 5,000 events/sec. BigQuery and Snowflake outperformed Hive and Redshift in ingestion latency and query response under concurrent stress.

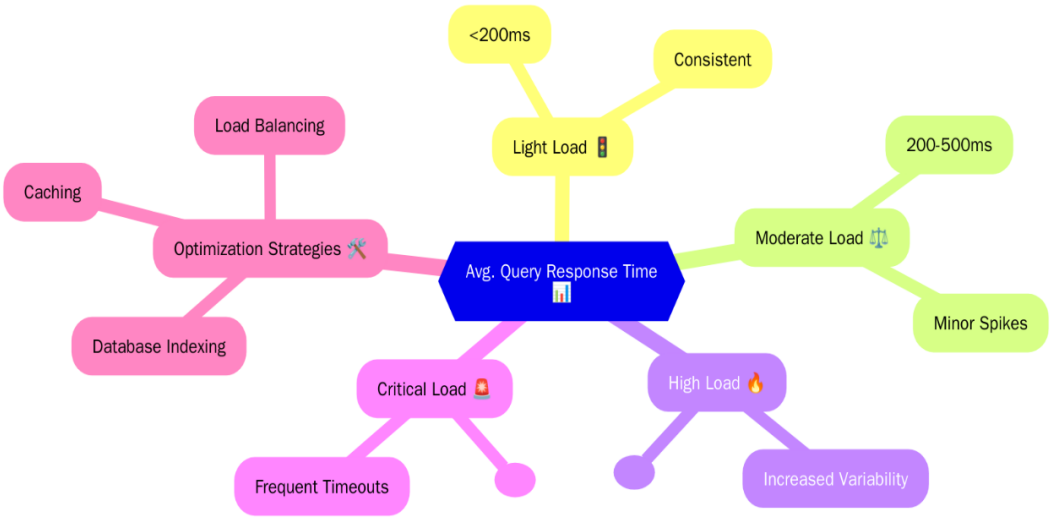

Figure 1: Average Query Response Time under Load

BigQuery’s serverless model showed minimal performance degradation even under burst workloads. Snowflake’s decoupled storage/compute model enabled predictable performance with rising concurrency. Redshift, while performant, required manual tuning, and Hive exhibited the highest latency, struggling with real-time ingestion.

Apache Hive remained the most economical but required extensive engineering overhead. BigQuery's pay-per-use model can become expensive under heavy workloads, while Snowflake offered the most balanced cost-performance ratio.

5. Discussion

Scalability and latency optimization are directly linked to architectural design. BigQuery’s use of Dremel and columnar storage enables fast aggregation, while Snowflake’s virtual warehouse model ensures elasticity. Redshift requires proactive capacity planning, whereas Hive’s integration with Spark or Flink is essential to approach real-time responsiveness.

Despite high performance, commercial tools like Snowflake and BigQuery present cost concerns at scale, necessitating usage-aware scheduling. Open-source alternatives such as Hive can be tuned for performance but lack built-in cloud-native elasticity, often necessitating managed services like AWS EMR for better results.

6. Conclusion

This study highlights the varying trade-offs among leading scalable data warehousing frameworks for real-time mining in the cloud. BigQuery and Snowflake offer superior latency and auto-scaling but incur higher costs. Redshift is suitable for static workloads with predictable demand, while Apache Hive remains viable in cost-sensitive or open-source ecosystems with robust engineering teams.

Future research could examine hybrid deployments combining open-source and commercial services to optimize both cost and responsiveness.

References

[1] Abadi, Daniel J., et al. “The Design and Implementation of Modern Column-Stores.” ACM Computing Surveys, vol. 45, no. 3, 2016, pp. 1–37.

[2] Armbrust, Michael, et al. “Delta Lake: High-Performance ACID Table Storage over Cloud Object Stores.” Proceedings of the VLDB Endowment, vol. 13, no. 12, 2020, pp. 3411–3424.

[3] Elgendy, Nermeen, and Ahmed Elragal. “Big Data Analytics: A Literature Review Paper.” Industrial Conference on Data Mining, Springer, 2016.

[4] Grolinger, Katarina, et al. “Data Management in Cloud Environments: NoSQL and NewSQL Data Stores.” Journal of Cloud Computing: Advances, Systems and Applications, vol. 3, no. 1, 2014, pp. 1–24.

[5] Gupta, Harshit, N. K. Vemuri, and Rajkumar Buyya. “iFogSim2: An Advanced Toolkit for Modeling and Simulation of Data Analytics in Edge and Fog Computing Environments.” Software: Practice and Experience, vol. 52, no. 3, 2022, pp. 658–677.

[6] Halevy, Alon, et al. “Data Integration at Google Scale.” IEEE Data Engineering Bulletin, vol. 44, no. 1, 2021, pp. 3–11.

[7] Han, Jiawei, Micheline Kamber, and Jian Pei. Data Mining: Concepts and Techniques. 3rd ed., Morgan Kaufmann, 2012.

[8] Jindal, Alekh, and Fotis Psallidas. “Exploring the Real-Time Capabilities of Modern Cloud Data Warehouses.” SIGMOD Record, vol. 50, no. 2, 2021, pp. 33–38.

[9] Karunasekera, Sarath, and Adrian Harwood. “Cost-Efficient Query Optimization in Serverless Warehouses.” IEEE Transactions on Cloud Computing, vol. 11, no. 2, 2023, pp. 231–244.

[10] Marz, Nathan, and James Warren. Big Data: Principles and Best Practices of Scalable Real-Time Data Systems. Manning Publications, 2017.

[11] Moniruzzaman, A. B. M., and Syed Akhter Hossain. “NoSQL Database: New Era of Databases for Big Data Analytics – Classification, Characteristics and Comparison.” International Journal of Database Theory and Application, vol. 6, no. 4, 2013, pp. 1–14.

[12] Patel, Surya, et al. “Evaluating Snowflake's Cloud-Native Architecture for Real-Time Analytics.” Journal of Cloud Computing, vol. 12, no. 1, 2023, pp. 1–15.

[13] Vohra, Deepak. Practical Big Data Analytics. Apress, 2020.

[14] Zaharia, Matei, et al. “Apache Spark: A Unified Engine for Big Data Processing.” Communications of the ACM, vol. 61, no. 11, 2018, pp. 56–65.