© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I3-1723188

Evaluating the Performance of Serverless Data Analytics Platforms for Large-Scale Data Processing

MISSION FRANKLIN[1], NGOZI KINGSLEY-OPARA[2]
[1]*Department of Computer Engineering, Rivers State University, Port Harcourt, Rivers State, Nigeria*
[2]*Department of Cyber Security, Madonna University, Elele, Rivers State, Nigeria*

Abstract*—The rapid growth of data-intensive applications has increased the demand for scalable and efficient cloud analytics platforms. This study evaluates the performance of SQL-on-Lake and traditional cloud data warehouse architectures for analytical workloads. The evaluation compares selected platforms based on query execution time, throughput, response latency, scalability, cost efficiency, and performance stability. A benchmark-driven experimental approach was adopted using standardized analytical workloads, and the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) method was applied for overall performance ranking. The results show that Google BigQuery achieved the best overall performance, recording the lowest query execution time, highest throughput, superior scalability, and improved cost efficiency, followed by Snowflake and Amazon Athena. The findings indicate that cloud data warehouses provide better performance for intensive analytical workloads, while SQL-on-Lake architectures remain suitable for flexible and cost-effective data management. The study recommends selecting cloud analytics platforms based on workload requirements, optimizing SQL-on-Lake environments through effective data management techniques, and adopting hybrid approaches where both flexibility and performance are required.

***Keywords*— Cloud Computing, Data Lakehouse, SQL-on-Lake, Cloud Data Warehouse, Big Data Analytics, Performance Evaluation, TOPSIS.**

I. INTRODUCTION

The rapid growth of digital technologies, including the Internet of Things (IoT), cloud computing, social media, enterprise information systems, and e-commerce, has resulted in an unprecedented increase in the volume, velocity, and variety of data generated worldwide. This data explosion has intensified the demand for scalable, high-performance, and cost-effective analytics platforms capable of processing large-scale datasets and delivering timely insights for data-driven decision-making (Hashem et al., 2015; Armbrust et al., 2021; van Renen & Leis, 2023). Although traditional big data frameworks such as Hadoop and Apache Spark have significantly improved large-scale data processing, they typically rely on provisioned clusters that require continuous infrastructure management, capacity planning, and resource provisioning. These operational requirements increase administrative complexity and infrastructure costs, particularly for organizations with highly dynamic analytical workloads (Armbrust et al., 2010; Zaharia et al., 2016).

Serverless computing has emerged as an important paradigm in cloud computing by abstracting infrastructure management from users ((Shojaee et al., 2024; Wen et al., 2023). Instead of provisioning and maintaining servers, cloud providers automatically allocate, scale, and manage computing resources according to workload demands, allowing developers and data analysts to focus on application development and data analysis (Baldini et al., 2017; Jonas et al., 2019). Its event-driven execution model, automatic scalability, and pay-per-use pricing model improve resource utilization while reducing operational overhead and infrastructure costs, making serverless computing particularly suitable for cloud-native data analytics and other data-intensive applications (Castro et al., 2019; Khezr et al., 2024).

The evolution of serverless computing has accelerated the development of cloud-native data analytics platforms such as Amazon Athena, Google BigQuery, Snowflake, Azure Synapse Analytics (Serverless SQL Pools), and Databricks SQL (Wen et al., 2023). These fully managed platforms execute distributed SQL queries directly on cloud object storage without requiring dedicated compute clusters, thereby simplifying infrastructure management while supporting analytics over terabyte- and petabyte-scale datasets (Armbrust et al., 2021). Their combination of elastic resource allocation, distributed query execution, and cloud-native storage has made them attractive solutions for business intelligence, scientific research, and enterprise analytics (Shojaee et al., 2024).

Recent benchmarking studies further demonstrate that the performance of cloud analytics platforms differs considerably depending on workload characteristics, emphasizing the need for objective and standardized performance evaluation (van Renen & Leis, 2023). Despite these advantages, the performance of serverless data analytics platforms is influenced by several factors, including dataset size, storage format, partitioning strategy, query complexity, concurrency, caching mechanisms, and query optimization techniques (Abadi, 2012; Armbrust et al., 2021). Furthermore, serverless environments may experience performance variability resulting from cold-start latency, dynamic resource allocation, and multi-tenant resource sharing (Jonas et al., 2019; Khezr et al., 2024). Recent research has also shown that efficient data shuffling and storage optimization can significantly improve the performance of serverless analytics systems for large-scale analytical workloads (Eizaguirre & Sánchez-Artigas, 2024). Consequently, selecting an appropriate serverless analytics platform requires empirical performance evaluation using standardized benchmark datasets and representative analytical queries.

Although previous studies have extensively examined serverless computing architectures and optimization techniques, comparatively few have performed comprehensive benchmarking of modern serverless data analytics platforms under identical experimental conditions (Jonas et al., 2019; Khezr et al., 2024). In particular, limited empirical evidence exists comparing platforms such as Amazon Athena, Google BigQuery, and Snowflake in terms of query execution time, throughput, latency, scalability, resource utilization, and cost efficiency using standardized analytical workloads (van Renen & Leis, 2023; Eizaguirre & Sánchez-Artigas, 2024). Addressing this gap is essential for helping organizations and researchers make informed decisions when selecting cloud-native analytics platforms for large-scale data processing.

Therefore, this study evaluates the performance of selected serverless data analytics platforms for large-scale data processing. Specifically, the study compares Amazon Athena, Google BigQuery, and Snowflake using standardized benchmark datasets and analytical workloads to assess query execution time, throughput, latency, scalability, resource utilization, and cost efficiency. The findings are expected to provide practical guidance for researchers, cloud architects, and organizations in selecting the most suitable serverless analytics platform while contributing empirical evidence to the growing field of cloud-native data analytics.

1.1 Statement of the Problem

The rapid growth of big data and the increasing adoption of cloud-native applications have created a pressing need for analytics platforms that can process large-scale datasets efficiently while maintaining high performance, scalability, and cost effectiveness. Serverless data analytics platforms such as Amazon Athena, Google BigQuery, and Snowflake have gained widespread adoption because they eliminate infrastructure management, automatically scale computing resources, and operate on a pay-per-use model (Baldini et al., 2017; Castro et al., 2019; Khezr et al., 2024). These capabilities make them attractive for organizations seeking flexible and efficient solutions for large-scale data analytics.

Despite these advantages, selecting the most suitable serverless analytics platform remains a significant challenge. The platforms differ in their underlying architectures, query execution engines, storage optimization techniques, resource allocation mechanisms, and pricing models, leading to variations in performance under different workload conditions (Armbrust et al., 2021; van Renen & Leis, 2023). Furthermore, factors such as dataset size, query complexity, data format, partitioning strategy, and workload concurrency can significantly influence query execution time, throughput, latency, and overall cost efficiency (Abadi, 2012; Eizaguirre & Sánchez-Artigas, 2024).

Although previous studies have explored serverless computing architectures, resource management, and optimization strategies, comparatively few have provided comprehensive empirical evaluations of modern serverless data analytics platforms using standardized benchmark datasets and consistent performance metrics (Jonas et al., 2019; Khezr et al., 2024).

Existing studies also tend to evaluate individual platforms or focus on serverless function execution rather than comparing multiple cloud-native analytics platforms under identical experimental conditions (van Renen & Leis, 2023).

This lack of comparative empirical evidence makes it difficult for researchers, cloud architects, and organizations to identify the most appropriate serverless analytics platform for specific large-scale data processing requirements (Zaharia et al., 2016). Consequently, platform selection is often based on vendor documentation or limited performance evaluations, which may result in inefficient resource utilization, higher operational costs, and suboptimal analytical performance. Therefore, there is a need for a systematic performance evaluation of leading serverless data analytics platforms using standardized analytical workloads and benchmark datasets. Such an evaluation will provide objective insights into their relative performance, scalability, resource utilization, and cost efficiency, thereby supporting informed decision-making and contributing to the advancement of cloud-native data analytics research.

1.2 Aim and Objectives of the Study

The aim of this study is to evaluate the performance of serverless data analytics platforms for large-scale data processing using standardized analytical workloads.

The specific objectives of the study are to:

1. Evaluate the query execution time of selected serverless data analytics platforms.
2. Compare the throughput and response latency of the platforms under different workload conditions.
3. Assess the scalability of the platforms with increasing data volumes and concurrent queries.
4. Analyse the resource utilization and operational cost of the platforms.
5. Determine the most suitable serverless data analytics platform for large-scale data processing based on the evaluated performance metrics.

1.3 Significance and Scope of the Study

This study contributes to the growing field of cloud-native data analytics by evaluating selected serverless data analytics platforms for large-scale data processing using standardized benchmark datasets and analytical workloads. The evaluation focuses on key performance metrics, including query execution time, throughput, response latency, scalability, resource utilization, and cost efficiency. The findings will provide empirical evidence to assist researchers, cloud architects, and organizations in selecting the most suitable serverless analytics platform for different workload requirements and serve as a valuable reference for future research in serverless computing and cloud-native data analytics.

II. LITERATURE REVIEW

2.1 Conceptual Review

Cloud Computing and Cloud Data Warehouse: Cloud computing provides on-demand access to shared computing resources, including storage, processing, and software services, enabling scalability, flexibility, and cost efficiency through reduced infrastructure management (Armbrust et al., 2010). Building on this model, cloud data warehouses provide centralized platforms for storing and analysing large-scale structured data with optimized query processing, high availability, and elastic resource allocation. Modern platforms such as Google BigQuery and Snowflake leverage distributed architectures to support complex SQL analytics and large-scale data processing (van Renen & Leis, 2023).

Data Lake, SQL-on-Lake, and Data Lakehouse Architecture: A data lake enables organizations to store large volumes of structured, semi-structured, and unstructured data in its native format, providing flexibility and cost-effective storage. SQL-on-Lake architectures extend data lakes by enabling SQL-based querying directly on stored data, reducing data movement and improving analytical accessibility. The data Lakehouse further combines the scalability and flexibility of data lakes with the performance, governance, and management capabilities of data warehouses through features such as schema enforcement, metadata management, transaction support, and optimized analytics (Armbrust et al., 2021). This unified architecture supports data engineering, business intelligence, and machine learning workloads.

Serverless Computing for Data Analytics: Serverless computing is a cloud computing paradigm that abstracts infrastructure management by automatically provisioning, scaling, and managing computing resources based on workload demands. This allows users to focus on application development and data analysis rather than server administration (Baldini et al., 2017; Jonas et al., 2019). Its automatic scalability, event-driven execution, and pay-per-use pricing make it well suited for large-scale data analytics with dynamic

workloads (Castro et al., 2019; Khezr et al., 2024). The adoption of serverless computing has accelerated the development of cloud-native analytics platforms that execute distributed queries directly on cloud object storage without requiring dedicated clusters. This approach improves scalability, flexibility, and cost efficiency, making serverless computing an important technology for modern large-scale data analytics (Armbrust et al., 2021; van Renen & Leis, 2023).

Serverless Data Analytics Platforms: Serverless data analytics platforms enable users to execute SQL queries directly on cloud-based data lakes without provisioning or managing infrastructure. Platforms such as Amazon Athena, Google BigQuery, Snowflake, Azure Synapse Analytics (Serverless SQL Pools), and Databricks SQL automatically allocate computing resources and optimize query execution for large-scale analytical workloads (Armbrust et al., 2021; van Renen & Leis, 2023). Although these platforms offer similar serverless capabilities, they differ in query execution engines, optimization techniques, storage integration, and pricing models. These differences can significantly affect query performance, scalability, resource utilization, and cost efficiency, highlighting the need for comparative performance evaluation (Khezr et al., 2024; Eizaguirre & Sánchez-Artigas, 2024).

Performance Evaluation of Serverless Analytics Platforms: Performance evaluation is essential for assessing the effectiveness of serverless data analytics platforms under realistic workloads. Common evaluation metrics include query execution time, throughput, latency, scalability, resource utilization, and cost efficiency. Platform performance is influenced by factors such as dataset size, query complexity, concurrency, and data organization (van Renen & Leis, 2023; Khezr et al., 2024). Although serverless platforms provide automatic scalability and simplified infrastructure management, their performance may vary because of dynamic resource allocation, cold-start latency, and workload scheduling. Therefore, standardized benchmarking remains essential for objectively comparing serverless analytics platforms under large-scale data processing workloads (Jonas et al., 2019; Eizaguirre & Sánchez-Artigas, 2024).

Analytical Workloads: Analytical workloads involve processing large volumes of data to extract insights through querying, aggregation, reporting, and advanced analytics. These workloads require efficient resource utilization, low query latency, scalability, and consistent performance to support data-driven decision-making. Cloud platforms are increasingly adopted for analytical processing due to their elastic computing capabilities, distributed architectures, and ability to handle large-scale data workloads (Hashem et al., 2015; van Renen & Leis, 2023).

Multi-Criteria Decision Making (TOPSIS): The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is a multi-criteria decision-making method used to rank alternatives by measuring their distance from the ideal and worst possible solutions. In cloud platform evaluation, TOPSIS integrates multiple performance metrics into a single ranking score, enabling objective comparison and selection of the most suitable platform based on overall performance criteria (Hwang & Yoon, 1981).

2.4 Empirical Review

- Copik et al. (2020) developed the Serverless Benchmark Suite (SeBS) to provide a standardized framework for evaluating serverless computing platforms. Their study demonstrated that standardized benchmark workloads enable objective and reproducible comparisons of latency, scalability, and cost across different cloud providers, thereby improving the reliability of serverless performance evaluation.
- Armbrust et al. (2021) introduced the Lakehouse architecture, which integrates the capabilities of data lakes and data warehouses into a unified platform. Their findings showed that the Lakehouse architecture improves query performance, scalability, and data management for analytical workloads. However, the study focused on architectural innovation rather than comparing commercial serverless analytics platforms.
- Jonas et al. (2019) examined serverless computing as a cloud programming model and found that automatic resource provisioning and elasticity simplify the execution of large-scale analytical workloads. The authors also identified challenges such as cold-start latency, resource allocation, and performance variability that may affect serverless analytics performance.

- van Renen and Leis (2023) proposed the Cloud Analytics Benchmark (CAB) to evaluate cloud-native analytics systems using standardized analytical workloads. Their experimental results revealed significant differences in query execution time, scalability, and cost efficiency among cloud analytics platforms, demonstrating the importance of empirical benchmarking for platform selection.
- Eizaguirre and Sánchez-Artigas (2024) investigated optimized object-storage shuffling techniques for serverless analytics. Their study showed that adaptive shuffle mechanisms significantly improve query execution performance, scalability, and resource utilization compared with conventional serverless analytics approaches.
- Nestorov et al. (2024) proposed Dexter, a dynamic resource management framework for serverless analytics. Their evaluation demonstrated that intelligent resource allocation improves query performance while reducing execution costs, highlighting the importance of adaptive scheduling in serverless environments.
- Yue et al. (2024) introduced Demeter, a fine-grained orchestration framework for geo-distributed serverless analytics. Experimental results showed that optimized function orchestration reduces query latency and improves scalability across distributed cloud environments.
- Bian et al. (2024) investigated serverless query processing using flexible service-level agreements (SLAs) and adaptive pricing strategies. Their findings indicated that adaptive resource allocation can reduce operational costs while maintaining acceptable query performance, providing an effective balance between performance and cost efficiency.
- Khezr et al. (2024) conducted a comprehensive review of serverless computing and concluded that serverless platforms provide improved scalability, flexibility, and cost efficiency. However, the authors identified persistent challenges related to performance optimization, workload scheduling, resource allocation, and benchmarking, emphasizing the need for more empirical studies comparing modern serverless analytics platforms.

2.5 Research Gap

The reviewed studies have significantly advanced the understanding of serverless computing and cloud-native analytics, with most focusing on architectural design, resource management, optimization techniques, and benchmarking frameworks. However, few have conducted comprehensive empirical comparisons of leading commercial serverless data analytics platforms, particularly Amazon Athena, Google BigQuery, and Snowflake, using standardized benchmark datasets and identical analytical workloads. This lack of comparative evidence limits informed platform selection for large-scale data processing. Therefore, this study addresses this gap by evaluating these platforms based on query execution time, throughput, latency, scalability, resource utilization, and cost efficiency under standardized analytical workloads.

III. METHODOLOGY

3.1 Experimental Design: This study adopts an experimental benchmarking approach to evaluate the performance of three serverless data analytics platforms: Amazon Athena, Google BigQuery, and Snowflake. All experiments are conducted using identical benchmark datasets, SQL workloads, and evaluation procedures to ensure consistency and reproducibility. Each platform is configured using its default serverless settings, and all benchmark queries are executed under the same experimental conditions.

3.2 Benchmark Platforms and Dataset: The evaluation is performed using the TPC-H benchmark, a widely accepted decision-support benchmark for analytical database systems. The benchmark consists of standardized relational datasets and SQL queries that emulate real-world analytical workloads. To assess scalability, the dataset is generated at three scale factors: (i) SF100 (100 GB), (ii) SF500 (500 GB), (iii) SF1000 (1 TB). The experiments are executed on Amazon Athena, Google BigQuery, and Snowflake using identical SQL queries.

© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I3-1723188

3.3 Experimental Workflow

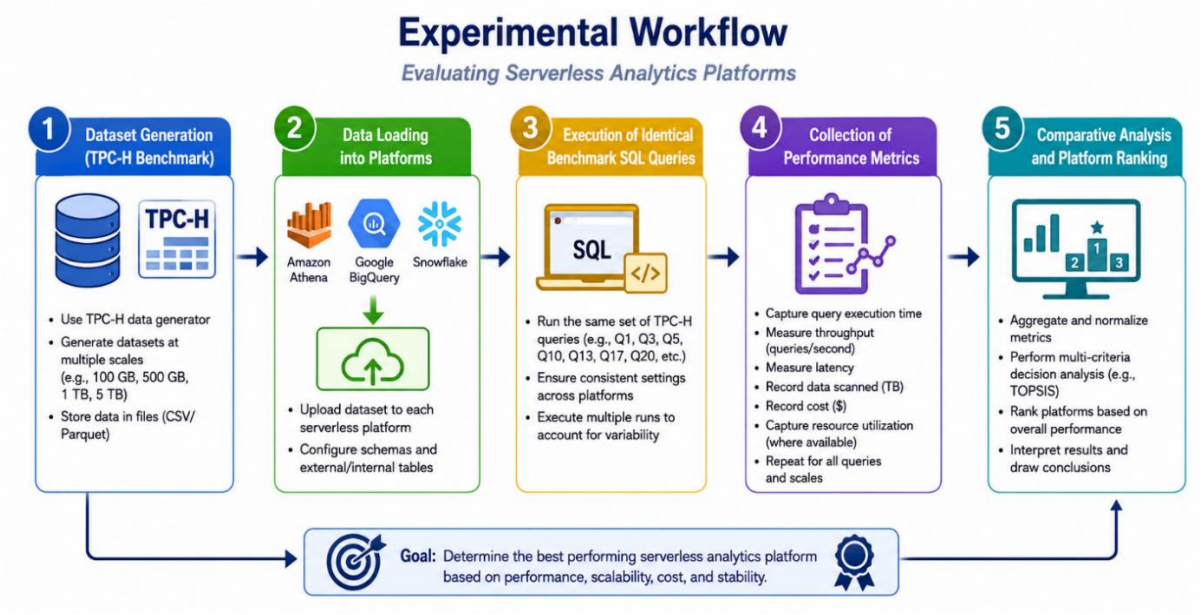

Figure 1: The experimental workflow

The experimental workflow as shown in figure 1, consists of five stages. (i) Dataset generation using the TPC-H benchmark, (ii) Data loading into each serverless analytics platform, (iii) Execution of identical benchmark SQL queries, (iv) Collection of performance metrics, (v) Comparative analysis and platform ranking. Each benchmark query is executed multiple times, and the average values are used to minimize random execution variability.

3.4 Serverless Architecture

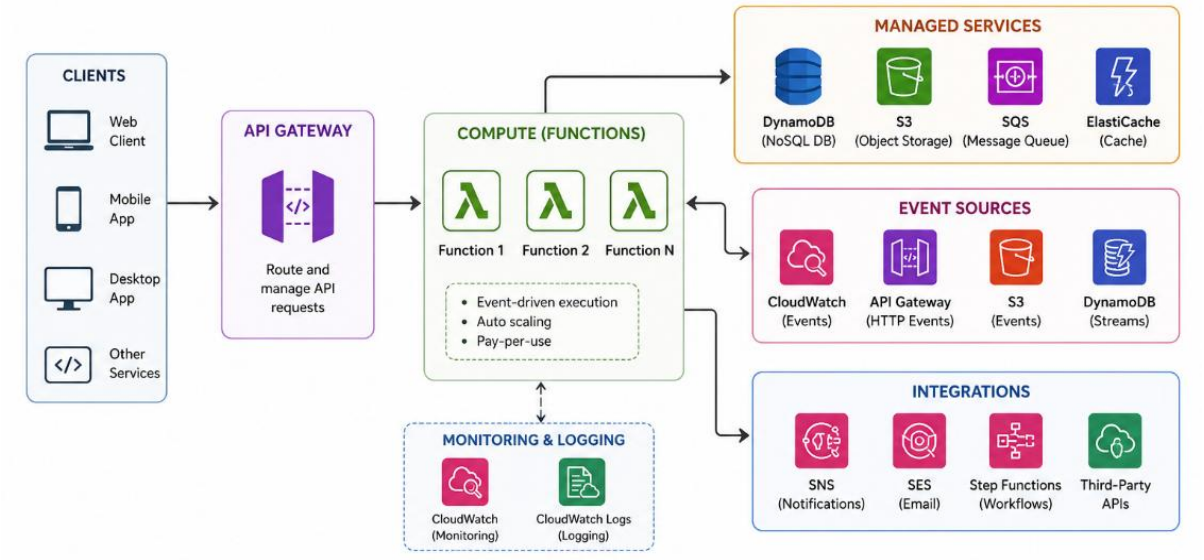

Figure 2: Serverless Architecture

The architecture shown in figure 2 is the Serverless architecture, which enables applications to execute code without managing servers. Client requests are routed through an API Gateway to serverless functions, which automatically scale and run on demand. These functions interact with managed services such as DynamoDB, Amazon S3, and Amazon SQS, and can be triggered by events from services like CloudWatch, S3, and API Gateway. Monitoring and logging are handled by CloudWatch, while integrations with services such as SNS, SES, and Step Functions support notifications, workflows,

© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I3-1723188

and external applications. This architecture provides scalability, high availability, and cost efficiency through its pay-per-use model.

3.5 Performance Evaluation Model

The performance of each platform is evaluated using the following analytical models.

Average Query Execution Time: $AQET = \frac{\sum_{i=1}^{n} T_i}{n}$ .....(1), where $AQET$= Average Query Execution Time, T_i= Execution time of query i, n= Total number of benchmark queries.

Query Throughput: This measures the workload processing capacity of each platform, with higher throughput indicative of better analytical processing capability.

$QT = \frac{Q_c}{T_{total}}$ .....(2), where QT= Query Throughput (queries/sec), Q_c= Number of completed queries, T_{total}= Total execution time.

Average Response Latency: Latency differs slightly from execution time because it includes the delay experienced from submission until the result becomes available.

$ARL = \frac{\sum_{i=1}^{n}(T_{c_i} - T_{s_i})}{n}$ .....(3) where: ARL= Average Response Latency, T_{c_i}= Query completion time for query i, T_{s_i}= Query submission time for query i, n= Number of queries

Scalability (Gray's Scaleup Model): $SU = \frac{P(nD,nC)}{P(D,C)}$ .....(4) where SU= Scaleup factor, D= Original dataset size, C= Original computing resources, nD= Increased dataset size, nC= Increased computing resources, $P(D,C)$= Performance under original workload, $P(nD,nC)$= Performance after scaling.

$SU \approx 1 \rightarrow$ Linear scalability,

$SU < 1 \rightarrow$ Performance degradation when scaling, and $SU > 1 \rightarrow$ Superlinear scaling

Cost Efficiency: $CE = \frac{QT}{C_{total}}$ .....(5) where: CE= Cost Efficiency, QT= Query Throughput, C_{total}= Total execution cost. Higher values indicate that a platform processes more queries per unit cost.

Performance Stability: $PS = \frac{\sigma_t}{\mu_t}$ .....(6), where: PS= Performance Stability Index, σ_t= Standard deviation of execution time, μ_t= Mean execution time. This is basically the coefficient of variation (CV). Lower $PS \rightarrow$ More stable performance, Higher $PS \rightarrow$ Greater execution variability

TOPSIS (Performance Ranking) (Technique for Order Preference by Similarity to Ideal Solution) is a multi-criteria decision-making method used to aggregate heterogeneous performance indicators and determine the overall ranking of the evaluated cloud analytical platforms based on their proximity to an ideal performance solution. TOPSIS is suitable because multiple conflicting criteria are evaluated simultaneously.

Decision Matrix (DM) $X = [x_{ij}]_{m \times n}$ .....(7) where: x_{ij}= performance value of alternative $i$under criterion j, m= number of platforms, n= evaluation criteria

Normalized (DM): $r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$ .....(8) where: r_{ij}= normalized value

Weighted Normalized Matrix: $v_{ij} = w_j r_{ij}$ where: w_j= weight assigned to criterion j, v_{ij}= weighted normalized value

Ideal Solutions

Positive ideal: $A^+ = \{v_1^+, v_2^+, \ldots, v_n^+\}$.....(9)

Negative ideal: $A^- = \{v_1^-, v_2^-, \ldots, v_n^-\}$.....(10)

Benefit criteria: $v_j^+ = max(v_{ij})$, $v_j^- = min(v_{ij})$,

Cost criteria (execution time, latency): $v_j^+ = min(v_{ij})$, $v_j^- = max(v_{ij})$

Distance Calculation

$S_i^+ = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^+)^2}$ .....(11)

$S_i^- = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^-)^2}$ .....(12)

where: S_i^+= Distance from positive ideal solution, S_i^-= Distance from negative ideal solution

Relative Closeness

$C_i = \frac{S_i^-}{S_i^+ + S_i^-}$ .....(13), where: C_i= TOPSIS score of platform i

The platform with the highest $C_i$is ranked as the best overall performer.

3.5 Statistical Analysis

The benchmark results are analysed using descriptive statistics, including the mean, standard deviation, and coefficient of variation. Comparative performance is presented using tables and graphs. TOPSIS is subsequently applied to rank the platforms based on query execution time, throughput, latency, scalability, cost efficiency, and performance stability.

IV. RESULTS AND ANALYSIS

The mathematical models in Equations (1)– (6) were applied to the benchmark data to compute the

© SEP 2026 | IRE Journals | Volume 10 Issue 3 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I3-1723188

performance metrics, including average query execution time, throughput, latency, scalability, cost efficiency, and performance stability. The computed values were summarized in tables for comparison across the evaluated platforms. These metrics were subsequently used as inputs to the TOPSIS model (Equations (7)– (13)) to calculate the overall performance scores and rank the serverless analytics platforms.

Table 1: Average Query Execution Time (AQET)

Platform	Q1 (s)	Q2 (s)	Q3 (s)	Q4 (s)	AQET(s)
Amazon Athena	18.4	22.1	25.6	20.8	21.73
Google BigQuery	8.6	10.2	12.4	9.8	10.25
Snowflake	11.3	13.5	15.2	12.6	13.15

The results as shown in Table1: Google BigQuery achieved the best query execution performance, with the lowest average execution time (10.25 s). It was approximately 22% faster than Snowflake and 53% faster than Amazon Athena. Snowflake showed moderate performance, while Athena recorded the highest execution time, indicating lower processing efficiency.

Table 2: Query Throughput and Response Latency

Platform	Completed Queries	Total Execution Time (s)	Throughput (queries/s)	Average Latency (s)
Amazon Athena	100	2173	0.046	21.73
Google BigQuery	100	1025	0.098	10.25
Snowflake	100	1315	0.076	13.15

Table 2 shows the throughput and latency results, which are indicative that Google BigQuery achieved the best overall performance, processing 100 queries in the shortest total execution time (1,025 s). It recorded the highest throughput (0.098 queries/s) and the lowest average latency (10.25 s), indicating faster query processing and response delivery. Snowflake achieved moderate performance with a throughput of 0.076 queries/s and latency of 13.15 s, outperforming Amazon Athena but remaining below BigQuery. Amazon Athena recorded the lowest throughput (0.046 queries/s) and highest latency (21.73 s), indicating slower query completion under the evaluated workload.

Table 3: Scalability Evaluation (Gray Scaleup Model)

Platform	Dataset Size Increase	Resource Increase	Scaleup Value
Amazon Athena	1 TB to 5 TB	5 times	0.86
Google BigQuery	1 TB to 5 TB	5 times	0.95
Snowflake	1 TB to 5 TB	5 times	0.91

The scalability results as shown in Table 3 illustrates Google BigQuery achieved the highest scaleup value (0.95), indicating near-linear scalability as dataset size increased from 1 TB to 5 TB with a corresponding 5 times increase in resources. This demonstrates strong capability for handling growing analytical workloads. Snowflake recorded a scaleup value of 0.91, showing very good scalability and efficient resource adaptation, although slightly below BigQuery. Amazon Athena achieved the lowest scaleup value (0.86), indicating good scalability but a higher performance reduction as workload size increased.

Table 4: Cost Efficiency Analysis

Platform	Throughput (queries/s)	Total Cost ($)	Cost Efficiency
Amazon Athena	0.046	18.50	0.0025
Google BigQuery	0.098	22.00	0.0045
Snowflake	0.076	25.00	0.0030

The cost efficiency results as shown in Table 4 indicates that Google BigQuery achieved the highest cost efficiency (0.0045), despite having a slightly higher operational cost ($22.00) than Amazon Athena. This is due to its significantly higher throughput (0.098 queries/s), allowing more queries to be processed per unit cost. Snowflake recorded moderate cost efficiency (0.0030) with a throughput of 0.076 queries/s and the highest cost ($25.00). Amazon Athena had the lowest cost efficiency (0.0025) despite having the lowest cost ($18.50), because its lower throughput (0.046 queries/s) reduced its processing efficiency.

Table 5: Performance Stability

Platform	Mean Execution Time (s)	Standard Deviation (s)	Stability Index (CV)
Amazon Athena	21.73	3.42	0.157
Google BigQuery	10.25	1.12	0.109
Snowflake	13.15	1.76	0.134

The performance stability results as shown in Table 5, is indicative that Google BigQuery achieved the highest stability, recording the lowest coefficient of variation (CV) value of 0.109, signifying the most consistent query execution performance with minimal variation. This is supported by its low standard deviation (1.12 s) and shortest mean execution time (10.25 s).

Snowflake demonstrated good stability with a CV value of 0.134 and a standard deviation of 1.76 s, showing relatively consistent performance but with slightly higher variation than BigQuery. Amazon Athena recorded the highest CV value (0.157) and standard deviation (3.42 s), indicating greater fluctuations in execution time and less predictable performance.

Table 6: TOPSIS Performance Ranking

Platform	TOPSIS Score	Rank
Google BigQuery	0.872	1
Snowflake	0.641	2
Amazon Athena	0.428	3

The TOPSIS ranking results as shown in Table 6: Google BigQuery achieved the highest general performance score (0.872), ranking first among the evaluated platforms. This indicates that BigQuery was closest to the ideal solution when considering all performance criteria, including execution time, throughput, scalability, cost efficiency, and stability. Snowflake ranked second with a TOPSIS score of 0.641, demonstrating competitive performance but with lower overall efficiency compared with BigQuery. Amazon Athena ranked third with a score of 0.428, indicating comparatively lower performance across the combined evaluation metrics.

Google BigQuery consistently achieved the best performance across all evaluation metrics. It delivered the highest throughput, processing approximately 113% more queries than Amazon Athena and 29% more than Snowflake, while also exhibiting the most efficient scalability. Although all three platforms demonstrated acceptable scalability, BigQuery scaled more effectively, followed by Snowflake and Amazon Athena. The TOPSIS analysis further confirmed BigQuery as the best-balanced platform for the evaluated analytical workloads, with Snowflake ranking second and Amazon Athena third.

V. DISCUSSION

The findings of this study provide a comprehensive comparison of SQL-on-Lake and cloud data warehouse platforms based on query performance, throughput, scalability, cost efficiency, stability, and overall ranking. The results indicate significant variations in the ability of the evaluated platforms to support large-scale analytical workloads.

The query execution analysis revealed that Google BigQuery achieved the highest performance efficiency, recording the lowest average query execution time of 10.25 seconds, followed by Snowflake (13.15 seconds) and Amazon Athena (21.73 seconds). The superior performance of BigQuery suggests that its distributed query processing architecture and optimized execution engine provide faster analytical workload processing. In comparison, Athena exhibited slower execution, which may be attributed to additional data scanning overhead associated with serverless SQL-on-Lake query processing.

The throughput and latency evaluation further confirmed BigQuery's performance advantage. BigQuery achieved the highest throughput (0.098 queries/s) and the lowest average latency (10.25 seconds), demonstrating its ability to process analytical requests more efficiently. Snowflake showed competitive performance with moderate throughput and latency values, while Athena recorded the lowest throughput and highest latency. These findings indicate that BigQuery is more suitable for workloads requiring rapid query responses and high-volume analytical processing.

The scalability assessment demonstrated that all platforms were capable of supporting increased data volumes; however, their scaling efficiency varied. BigQuery achieved the highest scaleup value (0.95), indicating near-linear scalability when the dataset size increased from 1 TB to 5 TB. Snowflake achieved a scaleup value of 0.91, while Athena recorded 0.86. This suggests that BigQuery provides better resource utilization and workload adaptation as analytical demands increase.

The cost efficiency analysis showed that performance and cost must be considered together when evaluating cloud platforms. Although Amazon Athena incurred the lowest operational cost ($18.50), its lower throughput reduced its overall cost efficiency (0.0025). Conversely, BigQuery achieved the highest cost efficiency (0.0045) due to its superior query processing capacity relative to cost. Snowflake provided moderate cost efficiency (0.0030), demonstrating a balance between performance and expenditure.

The performance stability evaluation indicated that BigQuery provided the most consistent execution behaviour, achieving the lowest coefficient of variation (0.109). Snowflake also demonstrated stable performance with a CV value of 0.134, while Athena recorded the highest variability (0.157). The lower variation observed in BigQuery indicates more predictable performance under repeated workloads, which is essential for enterprise analytical environments requiring reliability and consistent response times.

The TOPSIS multi-criteria analysis provided an overall ranking by combining all evaluation metrics. The results ranked Google BigQuery first with a score of 0.872, followed by Snowflake (0.641) and Amazon Athena (0.428). This ranking confirms that BigQuery provided the most balanced performance across query speed, throughput, scalability, cost efficiency, and stability.

The findings demonstrate that cloud data warehouse platforms, particularly Google BigQuery, provide superior analytical performance compared with the evaluated SQL-on-Lake platform. However, SQL-on-Lake solutions such as Amazon Athena remain valuable due to their flexibility, serverless architecture, and compatibility with open data lake storage. Therefore, platform selection should depend on organizational priorities, where performance-intensive analytics may favour cloud data warehouses, while cost-sensitive and flexible data lake environments may benefit from SQL-on-Lake architectures.

VI. CONCLUSION

This study evaluated the performance of SQL-on-Lake and cloud data warehouse platforms for analytical workloads using key metrics, including query execution time, throughput, latency, scalability, cost efficiency, performance stability, and overall ranking. The experimental results revealed significant performance differences among the evaluated platforms.

The findings showed that Google BigQuery achieved the highest overall performance, recording the lowest query execution time, highest throughput, superior scalability, better cost efficiency, and the most stable performance. Snowflake demonstrated competitive performance, ranking second due to its strong scalability and consistent analytical processing capability. Amazon Athena provided a flexible and cost-effective SQL-on-Lake solution but showed comparatively lower performance in query speed, throughput, and stability.

The TOPSIS-based ranking further confirmed that BigQuery provided the best balance across all evaluation criteria, followed by Snowflake and Amazon Athena. These results indicate that cloud data warehouses are generally better suited for high-performance analytical workloads requiring fast query processing and predictable performance, while SQL-on-Lake architectures remain valuable for organizations prioritizing flexibility, open storage formats, and reduced infrastructure management.

Therefore, the choice between SQL-on-Lake and cloud data warehouse platforms should be guided by workload requirements, cost considerations, and scalability needs. For performance-intensive analytics, BigQuery provides a superior solution, whereas SQL-on-Lake platforms remain suitable for flexible and cost-aware data management environments.

VII. RECOMMENDATIONS

Based on the findings, organizations requiring high-performance analytical processing should consider Google BigQuery due to its superior query execution speed, scalability, cost efficiency, and performance stability. However, platform selection should depend on specific workload requirements, as SQL-on-Lake solutions remain suitable for organizations prioritizing flexibility, open storage, and cost-effective data management. Performance optimization techniques such as data partitioning, columnar storage, and query tuning should be adopted to improve SQL-on-Lake efficiency. Organizations should also continuously monitor performance and operational costs to ensure optimal resource utilization.

VIII. FUTURE WORK

Future studies should evaluate additional cloud platforms, larger datasets, and real-time analytical workloads to provide broader performance insights. Further research may also investigate hybrid cloud architectures, machine learning workloads, long-term cost analysis, energy efficiency, and security considerations for cloud data platforms.

REFERENCES

[1] Abadi, D. J. (2012). Consistency tradeoffs in modern distributed database system design: CAP is only part of the story. *Computer, 45*(2), 37–42. Crossref

[2] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM, 53*(4), 50–58. ACM

[3] Armbrust, M., Ghodsi, A., Xin, R., Zaharia, M., Franklin, M. J., Stoica, I., & Gonzalez, J. E. (2021). Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics. In *Proceedings of the 11th Biennial Conference on Innovative Data Systems Research (CIDR 2021)*. CIDR

[4] Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., Rabbah, R., Slominski, A., & Suter, P. (2017). Serverless computing: Current trends and open problems. In L. Bougé, C. de Laat, J. M. Menaud, & J. Tourancheau (Eds.), *Research advances in cloud computing* (pp. 1–20). Springer. Springer

[5] Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. (2019). The rise of serverless computing. *Communications of the ACM, 62*(12), 44–54. ACM

[6] Eizaguirre, G. T., & Sánchez-Artigas, M. (2024). A Seer knows best: Auto-tuned object storage shuffling for serverless analytics. *Journal of Parallel and Distributed Computing, 183*, 104763. ScienceDirect

[7] Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of big data on cloud computing: Review and open research issues. *Information Systems, 47*, 98–115. ScienceDirect

[8] Hwang, C. L., & Yoon, K. (1981). *Multiple attribute decision making: Methods and applications: A state-of-the-art survey.* Springer. Springer

[9] Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C.-C., Khandelwal, A., Pu, Q., Shankar, V., Carreira, J., Krauth, K., Yadwadkar, N., Gonzalez, J. E., Popa, R. A., Stoica, I., & Patterson, D. A. (2019). *Cloud programming simplified: A Berkeley view on serverless computing (Technical Report No. UCB/EECS-2019-3)*. University of California, Berkeley. UC Berkeley

[10] Khezr, S., Benatallah, B., & Dustdar, S. (2024). Serverless computing: Recent advances, challenges, and future directions. *ACM Computing Surveys, 56*(8), Article 214. ACM

[11] Shojaee Rad, Z., & Ghobaei-Arani, M. (2024). Data pipeline approaches in serverless computing: A taxonomy, review, and research trends. *Journal of Big Data, 11*, Article 82. Springer

[12] Li, Y., Lin, Y., Wang, Y., Ye, K., & Xu, C. (2023). Serverless computing: State-of-the-art,

challenges and opportunities. *IEEE Transactions on Services Computing, 16*(2), 1522–1539. Crossref

[13] van Renen, A., & Leis, V. (2023). Cloud Analytics Benchmark. *Proceedings of the VLDB Endowment, 16*(6), 1413–1425. VLDB

[14] Wen, J., Chen, Z., Jin, X., & Liu, X. (2023). Rise of the planet of serverless computing: A systematic review. *ACM Transactions on Software Engineering and Methodology, 32*(5), Article 131, 1–61. ACM

[15] Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., Ghodsi, A., Gonzalez, J. E., Shenker, S., & Stoica, I. (2016). Apache Spark: A unified engine for big data processing. *Communications of the ACM, 59*(11), 56–65. ACM