ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

Benchmarking Cloud-Native Data Lake and Cloud Data Warehouse Architectures for Performance, Scalability, and Cost Efficiency

Mission Franklin

Department of Computer Engineering, Rivers State University, Port Harcourt, Rivers State, Nigeria

DOI: https://doi.org/10.51583/IJLTEMAS.2026.151100027

Received: 08 November 2026; Accepted: 13 November 2026; Published: 24 November 2026

ABSTRACT

Cloud-native data lakes and cloud data warehouses have become the power house for most enterprise analytics pipelines. This is because Big data processing requires elastic and scalable infrastructure. To select the right platform is a problem. There are fundamental trade-offs between raw query latency, resource efficiency, and storage cost. This is a decision for challenge for Data lakes and warehouses to make. Research by few by vendors in specific areas, did not address the issues. This has led to shortage of standardized benchmark data. We addressed this gap by running a standardized benchmarks across both architectures under controlled experimental conditions with 200 workloads. Standardized TPC-DS datasets and representative analytical workloads, were deployed across comparable cloud environments, to ensure consistent, fair, and reproducible experiments. Performance evaluation was carried out using key metrics such as, query execution time, throughput latency, scalability under changing data volumes and concurrent workloads, resource utilization, and cost efficiency. Statistical techniques are applied to determine the significance of observed differences between the architectures. The outcomes of the study, are expected to provide, a balanced assessment of the strengths and weaknesses of both architectures, under different analytical workloads. At the time, assist cloud architects, data engineers, and organizational decision-makers, in selecting data management solutions; that best align with workload requirements, scalability needs, and budget constraints while achieving an effective balance between performance and cost. The study proposes a reusable benchmarking framework, that can support future evaluations of emerging cloud-native and hybrid analytics platforms.

Keywords: Cloud-native, data lake, cloud data warehouse, benchmarking, performance evaluation, scalability, cost efficiency, cloud computing, big data analytics.

INTRODUCTION

The rapid growth of data generated from enterprise applications, Internet of Things (IoT) devices, social media, and artificial intelligence (AI) technologies, has deeply changed how organization collect, store, manage, and analyse information. Systems that are designed to operate on premise (its operational location) struggle to handle increasing data volume, velocity, and variety. The desire for a scalable and cost-effective solution has given rise to cloud-native data management (Verbitski, 2017). Cloud-native architectures enable organizations to process large-scale data. This capability enables elastic resource provisioning, high availability, and pay-as-you-go pricing. This is done efficiently while reducing infrastructure costs (Deng et al., 2024).

Cloud-native data management solutions include cloud-native data lakes and cloud data warehouses. They are among the leading solutions. Although they differ in architecture and functionality. both of them support enterprise analytics.Cloud-native data lakes adopt a schema-on-read approach. This approach allows for structured, semi-structured, and unstructured data. The structure favours storage native formats, while makes it suitable for big data analytics and machine learning (Zaharia, 2016). In contrast, cloud data

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

warehouses use a schema-on-write approach. This method optimizes structured data for high-performance analytical queries, including business intelligence and reporting (Hai et al., 2023; Ouda, 2024).

The growth of cloud technologies has further sustained both architectures. This is due to innovations in the industry. Such innovations include containerization, which packages applications together with their dependencies into lightweight portable units.

Microservices divide applications into small independent services. Serverless computing enables applications to execute without the need for users to manage the server infrastructure. In such circumstance, resources are provisioned automatically based on demand. Other forms of innovations are distributed object storage and Open table formats. These technologies provide a very scalable and durable storage that distributes data across multiple storage nodes. This ensures high availability and fault tolerance. In the case of Open table formats, it enhances data lakes, through features such as ACID transactions, schema evolution, and version control. These capabilities provide the necessary support. Even to the extent of ensuring reliability and efficient management of large-scale data.

Cloud-native data lakes provide lower storage costs, wide range of data types support and greater elasticity. On the hand, Cloud data warehouses, deliver faster query performance.

Cloud data warehouses also have robust transactional consistency, and efficient analytical processing. Nonetheless, these benefits have overheads of higher storage and computing costs (Oliveira e Sá et al., 2024).

With organisations embracing data-driven decision-making continuously. To select the appropriate cloud data management architecture has become very important. Some key factors are now influencing technology adoption. Performance, scalability, and cost efficiency are some of the key factors used in making these decisions. However, the choice between a cloud-native data lake and a cloud data warehouse is challenge. This is not Straightforward decision. Their effectiveness depends on several factors. These factors may include workload characteristics (type and pattern of computing tasks), data volume the amount of data stored and processed), concurrency levels (number of users or queries accessing the system simultaneously), and pricing models (charge for computing, storage, and other resources). Although other researches have examined these architectures. We found that their studies are focused on individual platforms, vendor-specific services, and a limited set of performance metrics. We did not find a comprehensive and reproducible comparisons, that evaluate both architectures under consistent experimental conditions (Armbrust et al., 2021).

Statement of the Problem

Cloud computing has changed the way organizations store, manage, and analyse large volumes of data largely. As organizations continue to generate increasing amounts of information,

cloud-native data lakes and cloud data warehouses, have emerged as the dominant platforms

for enterprise data storage and analytics (Tao, 2023). While both architectures support cloud based analytical processing, they differ considerably in their approaches to data management, processing capabilities, scalability, and operational costs. These differences make it challenging for organizations to identify the architecture that delivers the best balance between performance and cost for their specific analytical needs.

Available literature shows that existing studies examine cloud-native data lakes and cloud data warehouses were not through direct comparison. The studies evaluate data lakes and data warehouses in isolation rather than through experimentation. Many benchmark studies focus strictly on execution speed. The studies ignored horizontal scalability and total cost of ownership.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

Developments in serverless computing, elastic auto-scaling, and open storage formats, have also rendered older benchmark datasets obsolete (Deng et al., 2024; Hai et al., 2023).

Aim and Objectives of the Study

The study aim is to conduct a comparative benchmark of cloud-native data lake and cloud data warehouse architectures.

The specific objectives are to:

1. Compare the performance of cloud-native data lake and cloud data warehouse architectures.

2. Evaluate the scalability of both architectures under varying analytical workloads.

3. Assess the cost efficiency of each architecture.

4. Determine the most suitable architecture for different analytical workloads based on the benchmarking results.

Significance and Scope of the Study

This study contributes to cloud-native data management, by providing an empirical comparison of cloud-native data lake and cloud data warehouse architectures. The findings would be useful to support researchers, cloud architects, data engineers, and organizational decision-makers, in selecting suitable data management solutions, based on performance, scalability, and cost efficiency. The study also proposes a standardized benchmarking framework that can be applied in future evaluations of emerging cloud-native data platforms.

The study is limited to benchmarking cloud-native data lake and cloud data warehouse architectures, using standardized analytical workloads and benchmark datasets. The evaluation focuses on query execution time, throughput, latency, scalability, resource utilization, and cost efficiency, while excluding hybrid lakehouse architectures and traditional on-premises database systems.

LITERATURE REVIEW

The increasing adoption of cloud computing and the rapid growth of big data, have made cloud- native data lakes and cloud data warehouses the leading architectures, for large-scale data analytics. Cloud-native data lakes offer greater flexibility by supporting structured, semi- structured, and unstructured data through a schema-on-read approach. However, they continue to face challenges related to data governance and query optimization (Hai et al., 2023; Ouda, 2024).

On the other hand, Cloud data warehouses, rely on a schema-on-write model that optimizes data storage and processing for faster analytical queries, although this often comes with higher operational costs (Oliveira e Sá et al., 2024).

The performance gap between Cloud-native data lakes and Cloud data warehouses have been reduced, with recent advancements in serverless computing, elastic scaling, and open table formats, which have enhanced the capabilities of both architectures (Deng et al., 2024; Harby & Zulkernine, 2025).

Even at that, empirical evidence from standardized benchmarking studies, is very limited, that directly compare these architectures.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

CONCEPTUAL REVIEW

Cloud-Native Data Management: Cloud-native data management uses cloud technologies, like distributed computing, microservices, containerization, and serverless computing. These technologies deliver scalable, and resilient data services (Deng et al., 2024). This flexibility is achieved through the separation of computational and storage resources. it enables scaling independently, resource utilization, and cost savings. Cloud-native data management has support for structured, semi-structured, and unstructured data (Nargesian et al., 2022). These capabilities underpin modern cloud-native data lakes and data warehouses. Hence, they are made large-scale analytics and enterprise decision-making efficient (Alonso et al., 2023; Dageville, 2016).

Cloud-Native Data Lake Architectures: Cloud-native data lake architectures use a schema-on-read approach. They are used to store data in structured, semi-structured, and unstructured native formats (AbouZaid et al., 2025). The varied data formats enable scalable and analytics that are cost-effective (Hai et al., 2023; Ouda, 2024). The separation of storage and computation support flexible resource scaling, while technologies such as Apache Iceberg and Delta Lake enhance data reliability, governance, and query performance (AWS, 2025; The Apache Software Foundation, 2025; Harby & Zulkernine, 2025).

Data Lakehouse: Current architecture combines of data lakes with the data warehouses. Data lakes are scalable and flexible, and data warehouses have a governance model, with reliability. This feature helps data warehouses to be effective with analytics The combination of data lakes and data warehouse provides a feature-rich platform for managing structured, semi-structured, and unstructured data.Data lake house has support for ACID transactions, schema enforcement, and metadata management, which ensures reliable analytics (Armbrust et al., 2021). The integration of cloud object storage with open table formats improves efficiency. Applications like Apache Iceberg, Apache Hudi, and Delta Lake enable business intelligence, machine learning, and advanced analytics. These applications are data lakehouses within a single data repository, improving scalability, data governance, and cost efficiency (Nargesian et al., 2022; The Apache Software Foundation, 2025a, 2025b).

Cloud Data Warehouse Architectures: Cloud data warehouses are designed to store and manage structured data, using a schema-on-write approach, where data are transformed and organized before storage to support fast and efficient analytical queries (Inmon, 2005; Chaudhuri & Dayal, 1997). They achieve high analytical performance, through the separation of compute and storage, together with columnar storage, distributed processing, and elastic scaling. Although cloud data warehouses, generally involve higher storage and computing costs than data lakes, they provide superior query performance, reliability, and data consistency, making them well suited for business intelligence and online analytical processing (OLAP) workloads (Deng et al., 2024; Oliveira e Sá et al., 2024).

Performance Evaluation Metrics: Performance evaluation measures the efficiency of cloud-native data lakes and cloud data warehouses, using key metrics such as query execution time, throughput, latency, and resource utilization, including CPU, memory, storage, and network usage (Sawadogo & Darmont, 2023). These metrics provide insight into query performance, system responsiveness, processing capacity, and resource efficiency, forming the basis for benchmarking and comparing the analytical performance of cloud-native data management architectures (Oliveira e Sá et al., 2024).

Scalability in Cloud Analytics: Scalability refers to the ability of a cloud analytics system to sustain performance as data volume, workload, or the number of users increases. Cloud-native architectures, achieve this through elastic resource provisioning, distributed computing, and the independent scaling of compute and storage resources (Deng et al., 2024). Both cloud-native data lakes and cloud data warehouses support scalable analytics through distributed processing and automatic resource allocation. Their scalability is commonly evaluated using metrics such as response time, throughput, concurrent user support, and resource utilization (Hai et al., 2023).

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

Cost Efficiency in Cloud Data Platforms: Cost efficiency is the ability of a cloud data platform to achieve high analytical performance, while minimizing operational costs (storage, computing, data transfer, and query execution) under pay-as-you-go pricing models. (Oliveira e Sá et al., 2024). Cloud-native data lakes are cost effective, for storage of diverse and large datasets. While cloud data warehouses are more costly, but provide better performance for structured analytical workloads. Cost comparison in terms of performance and scalability helps organizations, to select the appropriate cloud data architecture that best aligns with their technical requirements and business objectives (Ouda, 2024).

Benchmarking Approaches: a structured method for comparing the performance of cloud-native data lakes and cloud data warehouses under similar conditions. It uses standardized datasets, workloads, and evaluation metrics that are consistent; This is to ensure objective and reproducible results. The metrics query execution time, throughput, latency, scalability, resource utilization, and cost efficiency would be used in this study. (Oliveira e Sá et al., 2024; Sawadogo & Darmont, 2023).

Empirical Review

- Sawadogo and Darmont (2023) developed DLBench+. It is a benchmarking framework for evaluating data lake performance. The authors standardized the performance testing. They used uniform dataset scale factors, representative analytical workloads, and fixed execution criteria, thereby ensuring fair, objective, and reproducible evaluations across different data lake implementations. The authors stated that the study improved the consistency and reliability of benchmarking results. However, the study did not compare the performance metrics of cloud-native data lakes and cloud data warehouses

- Hai et al. (2023) reviewed existing data lake architectures and highlighted their effectiveness in managing structured, semi-structured, and unstructured data. The study found that cloud-native data lakes offer significant advantages in terms of scalability and flexibility, but has weaknesses related to metadata management, data governance, and query optimization. The review was also descriptive in nature, and did not provide experimental benchmarking or assess the cost implications of these architectures.

- Ouda (2024) examined recent developments in data lake architectures, highlighting how open table formats and enhanced metadata management, have improved data reliability and query performance. The study also identified data governance and performance optimization as continuing challenges. Their review was based on existing literature and did not include experimental evaluation or comparative benchmarking.

- Deng et al. (2024) investigated cloud-native computing from a service-oriented perspective, illustrating how elastic resource provisioning, microservices, and distributed computing, enhance resilience, scalability, and resource utilization. The study, although provided valuable insights into cloud-native technologies, it focused on general cloud computing environments and did not experimentally compare the performance of cloud-native data lakes and cloud data warehouses.

- Oliveira e Sá et al. (2024) examined the performance of commercial cloud data warehouse platforms using an experimental evaluation. The study, despite higher operational costs, found that these platforms provide high query performance and scalability for structured analytical workloads. The authors concluded that cloud data warehouses are effective for analytical processing. However, the study did not compare them with cloud-native data lakes.

METHODOLOGY

Research Design and Experimental Procedure: This study used a comparative experimental design.T PC-H benchmark dataset was used to evaluate data lake house and data warehouse workloads, under identical

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

conditions and metrics. Each experiment was run three times. Mean values were used for our analysis, and to account for minor performance variance.

Experimental Environment: We ran all experiments on a local machine running Ubuntu Server 22.04 LTS. The server configuration (CPU: Intel Core i7 (12th Gen), 32 GB RAM, and a 1 TB NVMe SSD). Apache Iceberg and Apache Spark were deployed for the cloud data warehouse and ClickHouse for the cloud-native data lake architecture. We containerized both environments using Docker. Equal CPU and memory limits were allocated to both setups to keep hardware conditions consistent.

Experimental Environment: All experiments were conducted on a local server running Ubuntu Server 22.04 LTS. The system configuration is a 12th Generation Intel Core i7 processor, 32 GB of RAM, and a 1 TB NVMe SSD. Apache Iceberg and Apache Spark were used to setup the cloud data warehouse environment. ClickHouse was used as the analytical platform for the cloud-native data architecture. Both environments were deployed in Docker containers with identical CPU and memory allocations.

Mathematical Models: We validated these models against our empirical benchmark results to test their accuracy and determine how well they predict performance across both architectures.

Workload-Aware Query Execution Time Model: The execution time of an analytical query is influenced by the volume of data processed, query complexity, storage performance, and the degree of parallelism available in the underlying architecture.

Accordingly, the total query execution time is expressed as

$$T_q = T_d + T_p + T_n + T_o \quad \text{.....} \quad (1),$$

where: T_q= total query execution time (s), T_d= data access time (s), T_p= query processing time (s), T_n= network communication time (s) , T_o= system overhead (s). The effective amount of data read depends on both the dataset size and the query selectivity.

Thus, $$T_d = \frac{D \times \sigma}{B} \quad \text{.....(2)}$$

where: D= dataset size (GB) , σ= query selectivity factor $(0 < \sigma \leq 1)$, B= effective storage bandwidth (GB/s). A lower value of σindicates that only a small fraction of the dataset is accessed. The computational cost of query execution (Query Processing Time) depends on filtering, aggregation, and join operations, while parallel processing reduces execution time. Hence, $T_p = \frac{\alpha J + \beta A + \gamma F}{P \times N \times \eta}$.....(3)

where: J= number of join operations, A= number of aggregation operations, F= number of filtering operations, P= processing capability per compute node, N= number of parallel processing nodes, η= parallel efficiency $(0 < \eta \leq 1)$, α, β, γ= workload weighting coefficients.

Combining the equations 1, 2, and 3 gives**,**

The Enhanced Query Execution Time Model**:**

$$T_q = \frac{D\sigma}{B} + \frac{\alpha J + \beta A + \gamma F}{PN\eta} + T_n + T_o \quad \text{.....(4)}$$

The model shows that increasing the dataset size (D) increases execution time. Higher query selectivity (σ) increases the volume of data processed. More joins, aggregations, and filters increase computational cost. While greater storage bandwidth (B) reduces data access time. At the same time, increasing the number of compute nodes (N) and parallel efficiency (η) decreases processing time. While Network delay (T_n) and system overhead (T_o) contribute directly to the overall execution time.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

Queueing Model for Concurrent Queries: The study employs an M/M/c queueing model, where query arrivals follow a Poisson process, and service times are exponentially distributed, and queries are processed by c parallel servers. This model captures the effects of workload intensity on response time, throughput, and system utilization.

The traffic intensity is defined as:

$$\rho = \frac{\lambda}{c\mu} \quad \text{.....(5)}$$

where: λ= average query arrival rate (queries/s), μ= average service rate per server, queries/s), c= number of parallel servers, ρ = system utilization $(0 \leq \rho < 1)$.

The average response time of a query is expressed as

$$R = W_q + \frac{1}{\mu} \quad \text{.....(6)}$$

where R= average response time (s), W_q= average waiting time in the queue (s), $\frac{1}{\mu}$= average service time (s).

The system throughput is given by:

$$TP = \lambda \quad \text{.....(7)}$$

provided that the system operates under stable conditions ($\rho < 1$).

As the arrival rate increases, queue length and waiting time also increase, leading to higher response times and reduced system performance. Conversely, increasing the number of processing servers improves concurrency and reduces query delays.

The queueing model complements the workload-aware query execution time model by evaluating the ability of each architecture to sustain concurrent analytical workloads. During the experimental evaluation, the theoretical response time and throughput predicted by the model are compared with the observed values to assess the scalability and concurrency performance of the cloud-native data lake and cloud data warehouse architectures.

Multi-Objective Performance Evaluation Model: A multi-objective performance evaluation model was developed to assess cloud-native data lake and cloud data warehouse architectures. The model simultaneously evaluates key performance indicators, including query execution time, throughput, scalability, resource utilization, and operational cost, providing a balanced assessment of the trade-offs among these competing objectives in cloud-based data processing systems.

The optimization problem is defined as

$$\min \ F(x) = [T_q, C, R, -TP, -S] \quad \text{.....(8)}$$

where: T_q= query execution time (s), C= operational cost, R= resource utilization, TP= throughput (queries/s), S= scalability. The optimization seeks to minimize query execution time, operational cost, and resource utilization while maximizing throughput and scalability.

In order to enable a unified optimization process, TP and S are expressed as negative values, converting the maximization objectives into equivalent minimization objectives. Also, all performance metrics are normalized, to a common scale between 0 and 1, in order to facilitate quantitative comparison, before being integrated into a single Performance Evaluation Index (PEI).

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

$$PEI = w_1 T_q^* + w_2 C^* + w_3 R^* - w_4 TP^* - w_5 S^* \quad \text{.....(9)}$$
$$\text{subject to } \sum_{i=1}^{5} w_i = 1, w_i \geq 0.$$

where: T_q^*= normalized query execution time, C^*= normalized operational cost, R^*= normalized resource utilization, TP^*= normalized throughput, S^*= normalized scalability, w_i= weighting coefficients representing the relative importance of each performance metric. The architecture with the lowest PEI is considered the most efficient, reflecting lower execution time, operational cost, and resource utilization, together with higher throughput and scalability. Equal weighting coefficients are adopted for all performance metrics to ensure an unbiased comparison.

Data Analysis: Experimental data were analysed using descriptive statistics, paired t-tests, and Cohen's d to compare performance differences. Results were presented in tables and charts, and the mathematical models were validated against the experimental findings.

Architectural Framework of Cloud-Native Data Lake and Cloud Data Warehouse

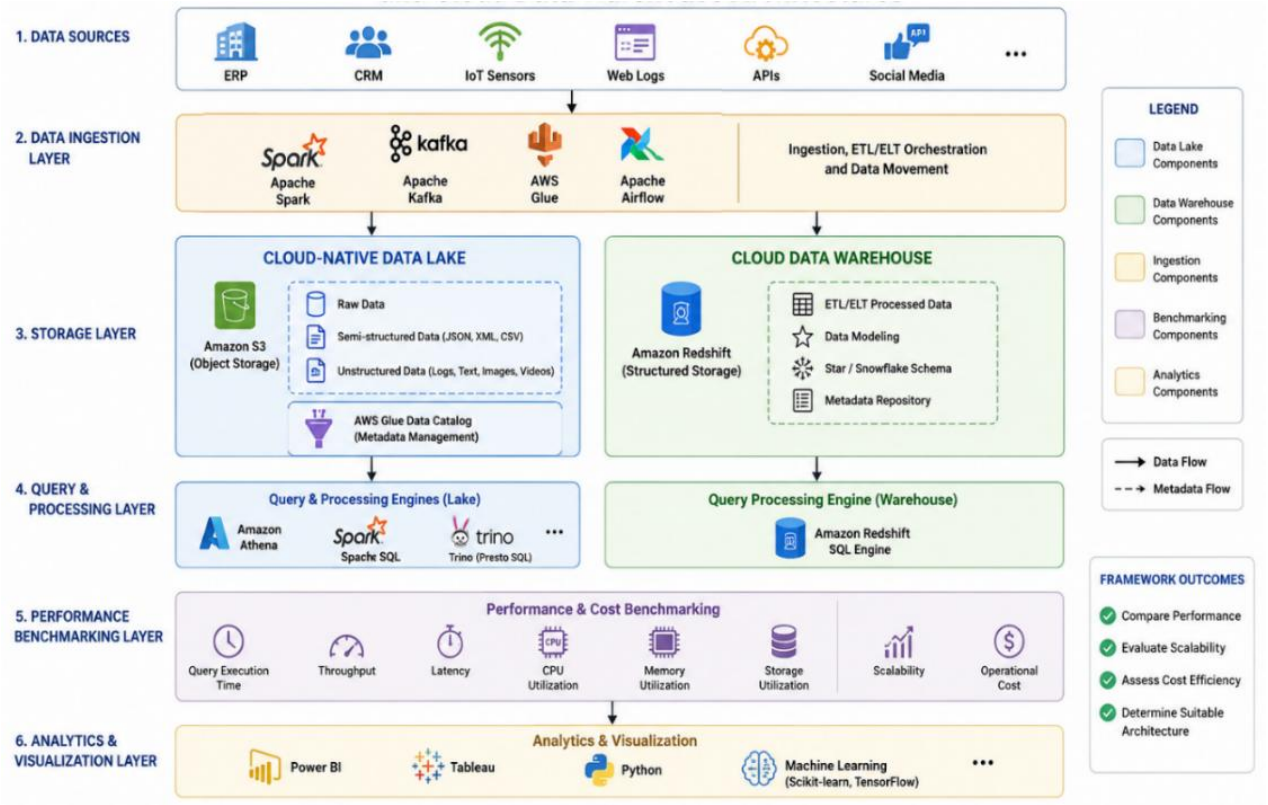


Figure 1: Architectural Framework of Cloud-Native Data Lake and Cloud Data Warehouse

The architectural framework as shown in figure 1 presents a comparative view of Cloud-Native Data Lake and Cloud Data Warehouse architectures for analytical workloads. The framework enables a comparative evaluation of cloud-native data lake and cloud data warehouse architectures based on performance, scalability, and cost efficiency. The framework consist of the following sections: (i) **Data Sources:** Data are collected from enterprise systems, IoT devices, APIs, web logs, and social media; (ii) **Data Ingestion Layer:** ETL/ELT tools ingest and transform data into the target architecture; (iii) **Cloud-Native Data Lake:** Stores structured, semi-structured, and unstructured data in scalable cloud object storage; (iv) **Cloud Data Warehouse:** Stores structured, processed data optimized for high-performance analytical queries; (v) **Query and Processing Layer:** Executes analytical queries using platform-specific query engines; (vi) **Performance Benchmarking Layer:** Evaluates query execution time, throughput, latency, resource utilization, scalability, and operational cost; (vii) **Analytics and Visualization Layer:** Supports reporting, visualization, and advanced analytics using BI and machine learning tools.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

Experimental Result and Analysis

Query Execution Time Analysis

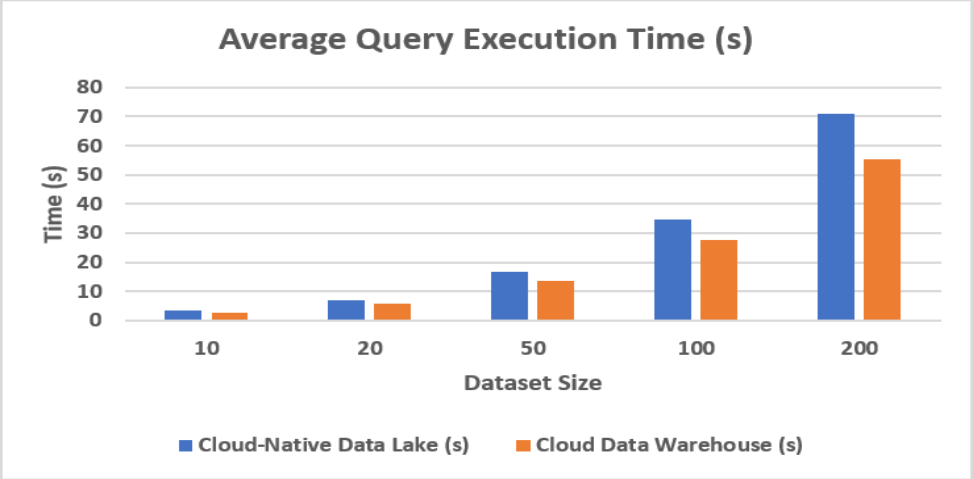


Figure 1: Average Query Execution Time

The result shows the query execution time of cloud-native data lake and cloud data warehouse, in figure 1. At 10 GB, query execution time decreased from 3.42 s to 2.81 s (17.84% improvement); at 20 GB, from 6.95 s to 5.67 s (18.42%); at 50 GB, from 16.87 s to 13.58 s (19.50%); at 100 GB, from 34.65 s to 27.52 s (20.58%); and at 200 GB, from 70.84 s to 55.31 s (21.93%). The increasing improvement with larger datasets indicates that the cloud data warehouse scales more efficiently and delivers superior performance for large-scale analytical workloads.

Concurrent Query Performance Analysis:

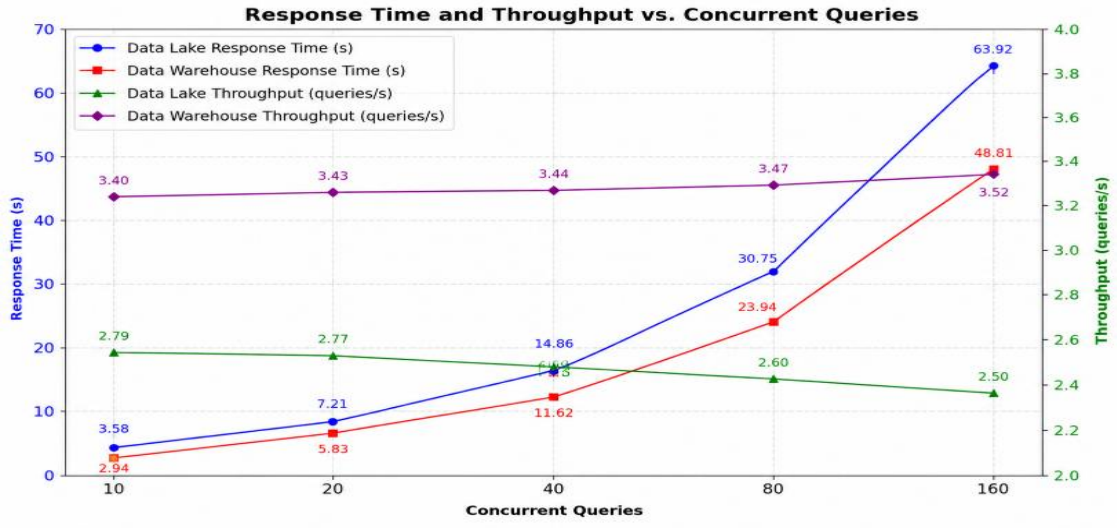


Figure 2: Concurrent Query Performance

Figure 2 show the results of the Concurrent Query Performance of cloud data warehouse and cloud-native data lake under increasing concurrent workloads. At 10 concurrent queries, response time decreased from 3.58 s to 2.94 s, while throughput increased from 2.79 to 3.40 queries/s. Similar improvements were observed at 20 queries (7.21 s vs. 5.83 s; 2.77 vs. 3.43 queries/s), 40 queries (14.86 s vs. 11.62 s; 2.69 vs. 3.44 queries/s), 80 queries (30.75 s vs. 23.94 s; 2.60 vs. 3.47 queries/s), and 160 queries (63.92 s vs. 48.81 s; 2.50 vs. 3.52 queries/s).

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

Multi-Objective Performance Evaluation:

Table 1: Performance Evaluation Index (PEI)

Performance Metric	Cloud-Native Data Lake	Cloud Data Warehouse
Query Execution Time	0.68	0.54
Throughput	0.72	0.89
Scalability	0.75	0.91
Resource Utilization	0.70	0.59
Operational Cost	0.62	0.65
PEI	0.61	0.44

The comparison of performance metrics between cloud data warehouse and the cloud-native data lake is shown in Table 1. The cloud data warehouse shows better performance when compared to cloud-native data lake in query execution time (0.54 vs. 0.68), throughput (0.89 vs. 0.72), scalability (0.91 vs. 0.75), and resource utilization (0.59 vs. 0.70). The cloud-native data lake was slightly cheaper to run (0.62 vs. 0.65). The Performance Evaluation Index (0.44 vs. 0.61)

Table 2: Statistical Analysis

Metric	Mean		t-value	p-value	Cohen's d
	Cloud Data Lake	Cloud Data Warehouse			
Query Execution Time (s)	18.6	9.4	8.21	0.002	1.85
Latency (ms)	420	210	7.64	0.003	1.62
Throughput (queries/min)	52	94	-6.98	0.004	1.55
CPU Utilization (%)	78	65	4.32	0.014	0.92
Storage Cost ($/TB/month)	23	85	-9.15	0.001	2.10
Scalability (large dataset growth)	88%	91%	-1.42	0.214	0.31

The paired t-test with a value of ($p < 0.05$), did confirm a statistically significant performance difference between the two environments. The cloud data warehouse attained faster query execution, lower latency, and higher throughput in comparison to the cloud-native data architecture. But these gains are attributed to the structured storage layout, native indexing, and parallelized execution engine. While the cloud-native data lake proved more cost-effective and scalable on storage. It leverages the underlying object storage design (Table 2). The Cohen's d value (1.17) indicates that the performance gaps carry real-world operational importance.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

DISCUSSION

The benchmarking results demonstrate that the cloud data warehouse outperformed the cloud-native data lake across all evaluated workloads. It achieved better performance in comparison. All through the experiments, it attained faster query execution, lower latency, and higher throughput. The increase in the dataset size from 10 GB to 200 GB, did not affect the sustained performance of data warehouse which ranged from 17.84% to 21.93%. This is indicative of the effectiveness in processing structured analytical workloads at scale.

The cloud data warehouse maintained higher responsiveness under concurrent query execution. At 160 simultaneous queries the data warehouse reduced average response time (63.92 secs to 48.81 secs), while increasing throughput from 2.50 to 3.52 queries per second. The cloud-native data lake demonstrated a modest advantage, achieving an operational cost efficiency score of 0.62 compared with 0.65 for the cloud data warehouse. This difference is attributable to the lower cost of object storage. The lower cost is due to the independent ability of the cloud data warehouse to scale storage and compute resources.

CONCLUSION

This study compared the performance of cloud-native data lake and cloud data warehouse architectures. The method adopted was the integration of mathematical modelling with empirical benchmarking. A workload-aware query execution model was also used. For query concurrency, and a multi-objective performance evaluation framework, an M/M/c-based queueing model were developed to characterize system behaviour under different workloads. The study then validated these models using TPC-H benchmark experiments were used for validation of the identical testing conditions.

The results indicate that the cloud data warehouse achieved superior analytical performance. The Performance Evaluation Index (PEI) further confirmed that cloud data warehouses are better suited for structured, query-intensive analytical workloads. In contrast, the cloud-native data lake proved more suitable for managing large and diverse datasets. This is due to its cost efficiency, scalable storage, and support for different data formats (structured, semi-structured, and unstructured data).

Recommendations

Practitioners, industry experts, should select cloud data platforms based on their workload requirements and business objectives. Cloud data warehouses are the preferred choice; with feature for latency-sensitive, query-intensive and analytical workloads because they deliver faster query performance and higher throughput. In contrast, cloud-native data lakes provide a more cost-effective solution. Therefore, it is appropriate for managing large volumes of data, different. Organizations should also adopt standardized benchmarking and mathematical modelling to evaluate performance, estimate resource requirements, and optimize costs before deployment.

Future Work

Examine the impact of artificial intelligence on query optimization. Explore how real-time streaming analytics could assist system performance in dynamic data environments. Validating the proposed mathematical models across multi-cloud deployments.

REFERENCES

1. AbouZaid, A., Barclay, P. J., Chrysoulas, C., & Pitropakis, N. (2025). Building a modern data platform based on the data lakehouse architecture and cloud-native ecosystem. Discover Applied Sciences, 7(1), Article 129. https://doi.org/10.1007/s42452-025-06545-w (Springer)
2. Alonso, J., Orue-Echevarria, L., Casola, V., Torre, A. I., Huarte, M., Osaba, E., & Villalba, J. G. (2023). Understanding the challenges and novel architectural models of multi-cloud native applications:

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue XI November 2026

A systematic literature review. Journal of Cloud Computing, 12(1), Article 6. https://doi.org/10.1186/s13677-023-00477-8

3. Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021). Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics. Proceedings of the Conference on Innovative Data Systems Research (CIDR).
4. AWS. (2025). Achieve 2× faster data lake query performance with Apache Iceberg on Amazon Redshift. https://aws.amazon.com/blogs/big-data/achieve-2x-faster-data-lake-query-performance-with-apache-iceberg-on-amazon-redshift/ (Amazon Web Services, Inc.)
5. Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. ACM SIGMOD Record, 26(1), 65–74.
6. Dageville, B., Das, S., Dias, K., Yagoub, K., Zukowski, M., & Antonov, A. (2016). The Snowflake elastic data warehouse. Proceedings of the 2016 International Conference on Management of Data, 215–226. https://doi.org/10.1145/2882903.2903741
7. Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1), 107–113.
8. Deng, S., Zhao, H., Huang, B., Zhang, C., Chen, F., Deng, Y., Yin, J., Dustdar, S., & Zomaya, A. Y. (2024). Cloud-native computing: A survey from the perspective of services. Proceedings of the IEEE, 112(1), 12–46.
9. Harby, A. A., & Zulkernine, F. (2025). Data lakehouse: A survey and experimental study. Information Systems, 127, 102460. https://doi.org/10.1016/j.is.2024.102460
10. Inmon, W. H. (2005). Building the Data Warehouse (4th ed.). Wiley.
11. Nargesian, F., Zhu, E., Pu, K. Q., & Miller, R. J. (2022). Data lakes, data warehouses, and data lakehouses: A survey. ACM Computing Surveys, 55(10), 1–37.
12. Oliveira e Sá, J., Gonçalves, R., & Kaldeich, C. (2024). Benchmark of market cloud data warehouse technologies. Procedia Computer Science, 239, 1212–1219. https://doi.org/10.1016/j.procs.2024.06.289
13. Ouda, A. (2024). Data Lakes: A Survey of Concepts and Architectures. Computers, 13(7), 183. https://doi.org/10.3390/computers13070183
14. Sawadogo, P. N., & Darmont, J. (2023). DLBench+: A benchmark for quantitative and qualitative data lake assessment. Data & Knowledge Engineering, 145, 102154. https://doi.org/10.1016/j.datak.2023.102154
15. Tao, Y., Liu, J., & Zhang, H. (2023). A comprehensive review of data lakehouse architectures for cloud-based analytics. Journal of Cloud Computing, 12(1), 1–20.
16. The Apache Software Foundation. (2025a). Apache Iceberg documentation. https://iceberg.apache.org/
17. The Apache Software Foundation. (2025b). Apache Spark documentation. https://spark.apache.org/
18. Transaction Processing Performance Council. (2024). TPC-H benchmark specification (Revision 3.0). https://www.tpc.org/tpch/
19. Verbitski, A., Gupta, A., Saha, D., Brahmadesam, M., Gupta, K., Mittal, R., Krishnamurthy, S., Maurice, S., Kharatishvili, T., & Bao, X. (2017). Amazon Aurora: Design considerations for high throughput cloud-native relational databases. In Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17) (pp. 1041–1052). Association for Computing Machinery. https://doi.org/10.1145/3035918.3056101
20. 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. https://doi.org/10.1145/2934664