Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2024, 11(11):10-14**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Performance Comparison of Redshift vs BigQuery for Large-Scale Data Analytics

Rameshbabu Lakshmanasamy

Senior Data Engineer, Jewelers Mutual Group

Abstract: Choosing the right solution for large-scale analytical processing in today's modern world can be challenging. Amazon Redshift and Google BigQuery, both present high-performance solutions that scale well, yet they have different structures concerning performance and cost. In this paper, Redshift and BigQuery are compared based on query performance and scalability with varying volumes of data and query complexity. Thus, by comparing the advantages and disadvantages of performance customization of Redshift versus the automatic scalability of BigQuery, we offer organizations specific recommendations as to which of these tools is the most suitable for analytics tasks, when to prioritize query complexity or over dataset size, and which option is preferable in the long term, when considering costs.

Keywords: Amazon Redshift, Google BigQuery, Query Optimization/tuning, Performance, MPP, Scalability, Elasticity

1. Introduction

Amazon Redshift it is a fast and scalable cluster-based data warehouse service that is fully managed by Amazon Web Services to use columnar storage and massively parallel processing (MPP) for query response (Hook & Porter, 2021). Distribution keys and node configurations serve as the features that afford fine-grained control over the system, especially in the case of structured data analytics. Database users can scale computation and storage independently in Redshift but need to adjust performance on their own as more data is added.

Google BigQuery, on the other hand is a serverless architecture, automated and non-provisioned data warehouse solution. The pricing method adopted is pay-per-query, which hides the complexity of the underlying infrastructure while processing data. Adopting BigQuery means an organization will not have to worry about servers, thus making it ideal for organizations that want a serverless solution where hardware is taken care of by the developers (Hook & Porter, 2021).

2. Benchmarking Query Performance Across Various Data Sizes and Complexities

Conducting a performance benchmark between Amazon Redshift and Google BigQuery requires a systematic approach to ensure comparability. Both platforms must be tested under equal conditions, allowing businesses to evaluate their strengths and weaknesses before choosing the most suitable option for their business intelligence needs.

To establish a reliable benchmarking environment for debugging and tuning, datasets ranging from 100 GB to several terabytes (1 TB, 10 TB, and beyond) were utilized. The selected workloads included read-dominant queries, such as SELECT statements, and write-dominant workloads for INSERT and UPDATE operations (Veeresh Biradar, 2024, April 12). This methodology enabled an assessment of how each platform managed varying levels of data complexity and volume, helping to identify the specific tasks and scenarios best suited for each solution.

Testing on Amazon Redshift was performed in a multi-node environment, utilizing both dense compute nodes (dc2) and dense storage nodes (ds2). Redshift's architecture focuses on optimizing query performance through careful management of distribution and sort keys (Demirbaga, et al., 2024). Additionally, compression

techniques were applied as needed, and routine maintenance tasks like VACUUM and ANALYZE were executed to prevent performance degradation. Concurrency scaling was activated to allow multiple queries to run simultaneously during high-traffic periods. While these manual optimizations are essential for maintaining optimal performance as data grows, they also introduce additional complexities in system management.

In contrast, Google BigQuery operates as a serverless solution that automatically allocates resources according to query volume. Users benefit from not needing to provision hardware or manage nodes, which shifts the focus toward analytics rather than infrastructure management. The testing environment leveraged BigQuery's partitioning and clustering capabilities, enhancing the performance of read operations, particularly when filtering and grouping by time-based columns. BigQuery's billing model, based on query execution rather than resource allocation, allows it to optimize resources dynamically according to query size and complexity, providing consistent performance across varying workloads.

To enhance comparability between platforms, the benchmarking design minimized variability by ensuring that query patterns, dataset sizes, and workload types closely mirrored each other. This careful approach reduced environmental discrepancies and yielded performance results free from inherent biases.

The benchmarking exercise included three main types of queries: basic SELECT operations, complex JOIN operations, and various analytical queries. Simple SELECT queries assessed basic calculations, such as SUM, AVG, COUNT, and MIN/MAX, providing an initial performance baseline for both platforms (Miryala et al., n.d.). Additionally, multi-JOIN operations were conducted to simulate real-world scenarios where data from multiple sources needed to be combined. Finally, complex analytical queries employing window functions, subqueries, and GROUP BY clauses were executed to evaluate each platform's capabilities in handling intricate data analytics standards in big data scenarios.

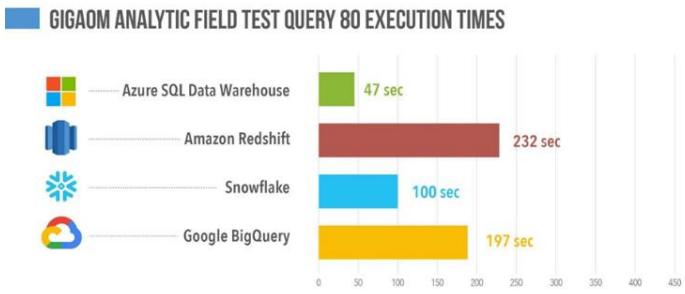
For datasets up to 100 GB, both Redshift and BigQuery exhibited excellent performance for simple queries, executing SELECT statements and aggregations quickly and efficiently. Redshift's optimized node configurations and distribution keys contributed to consistent performance, while BigQuery's serverless model enabled rapid execution of these basic operations. However, as query complexity increased, particularly for multi-JOIN operations and analytical queries, BigQuery generally outperformed Redshift due to its automatic resource management capabilities.

As the datasets grew to medium sizes (100 GB to 1 TB), Redshift remained reliable when adequately optimized. Nevertheless, its cluster-based architecture necessitated careful management of distribution and sort keys as performance began to degrade with more complex queries. In contrast, BigQuery maintained steady performance with medium datasets, leveraging partitioning and clustering to enhance its capabilities for analytical workloads.

With large datasets (1 TB to 10 TB and beyond), Redshift faced significant challenges, particularly with complex queries involving multiple JOINs and extensive analytical functions. Frequent maintenance using VACUUM and ANALYZE was often necessary to maintain efficiency. Although Redshift's concurrency scaling helped mitigate some performance issues, ongoing optimization became a bottleneck for large datasets. Conversely, BigQuery handled larger datasets more seamlessly, maintaining low query times even for compute-intensive tasks. However, the cost of processing queries in BigQuery raised concerns, especially with its pay-per-query pricing model. It could result in higher expenses for executing complex queries on extensive data volumes.

3. Analysis of Cost-Performance Trade-Offs Between the Two Platforms

Amazon Redshift and Google BigQuery are two prominent cloud-based platforms for large-scale data analysis. Each has distinct pricing structures influenced by workload size, query complexity, and usage patterns. Understanding these differences enables organizations to optimize performance while minimizing costs.



GIGAOM *Figure 2 -- Query 80. GigaOm Analytic Field Test Execution Time.*

4. Pricing Models

Amazon Redshift operates on a node-based billing system, where the instance types in the ecosystem determine charges. It offers an on-demand pricing model, ideal for short-term use but potentially costly over the long term. For more stable, predictable usage patterns, reserved instances are available at reduced rates for one to three years. Redshift's pricing encompasses both compute and storage resources, utilizing dense compute nodes (dc2) for high I/O workloads and dense storage nodes (ds2) for economical HDD storage (Nathan et al., 2024). Effective resource management requires careful planning regarding compute and storage parameters.

Google BigQuery, by contrast, employs a serverless model where charges are incurred per query based on the volume of data processed. This eliminates the need for users to manage nodes or clusters, as compute and storage costs are separate. Users pay storage fees based on the amount of data stored and computation costs depending on the data scanned by queries (Goss & Subramany, 2021, May). BigQuery's prefetch pricing model allows businesses with steady workloads to buy a certain number of query slots monthly, making it easier to predict costs for stable usage patterns.

5. Cost Considerations for Different Data Sizes

Small Datasets (up to 100 GB): For small datasets and infrequent queries, Redshift’s cluster-based pricing can be burdensome due to the fixed costs of maintaining idle clusters. Conversely, BigQuery’s pay-per-query model is more advantageous, as users pay only for the data processed. This makes BigQuery suitable for low-frequency workloads, allowing for occasional queries at minimal costs.

Medium to Large Datasets (100 GB to 10 TB): As datasets grow, Redshift’s pricing can become more manageable, especially with reserved instances. However, to achieve optimal performance, users must effectively manage distribution and sort keys. In BigQuery, while costs can escalate with frequent querying of large datasets, utilizing partitioning and clustering can help minimize expenses. Users must decide between the cost-effectiveness of the pay-per-query model versus the flat-rate pricing for regular queries.

6. Performance vs. Cost for analytical workloads

Both platforms have strengths when comparing cost performance. Redshift is advantageous for high-concurrency workloads, allowing multiple complex queries to run simultaneously without significantly increasing costs (Van Renen & Leis, 2023). For businesses with frequent, predictable queries, Redshift's reserved pricing is often more economical.

In contrast, BigQuery is well-suited for unpredictable workloads, automatically adjusting resources to meet demand. However, the pay-per-query model can lead to higher costs with increased usage. While Redshift allows for fine-tuning of performance through distribution and sort keys, BigQuery’s serverless architecture simplifies management, making it a strong choice for organizations handling data-intensive tasks (Van Renen & Leis, 2023).

7. Supporting Metrics and Numbers

Several key performance indicators (KPIs) and metrics are essential when evaluating Amazon Redshift and Google BigQuery for big data processing.

Average Query Response Time is a primary metric that indicates how well each platform processes queries based on incoming data volume and quality. For small datasets (up to 100 GB), both Redshift and BigQuery perform similarly, with simple queries executing in milliseconds to seconds. However, as dataset sizes increase, performance disparities become evident. For medium datasets (100 GB to 1 TB), Redshift typically outperforms BigQuery, incredibly when fine-tuned with distribution and sort keys. While BigQuery remains cost-effective, its performance can lag, particularly with complex queries involving large joins or subqueries. For larger datasets (1 TB to 10 TB), Redshift's optimized clusters maintain high-cost efficiency, although query execution times can vary based on partitioning and other optimizations. Conversely, BigQuery may experience longer query times as data volumes grow unless users leverage features like clustering or materialized views.

Compute Resources also play a crucial role. Redshift requires users to manage and optimize compute resources by selecting node types and quantities in clusters, resulting in higher costs for larger or more complex queries. In contrast, BigQuery operates on a serverless model, automatically provisioning compute resources based on query demands. This flexibility eliminates manual management but can lead to increased costs as data volumes scale.

Additionally, the cost per query is an important consideration. Redshift charges a fixed price per second for storage, making it expensive for light workloads or idle clusters. BigQuery's pay-per-query model is more efficient for small datasets, as users only pay for the data processed. For larger datasets, Redshift's costs increase with additional nodes, but reserved instances stabilize query costs. BigQuery's costs scale directly with the data processed, making it ideal for dynamic workloads.

Finally, storage costs differ between platforms. Redshift's pricing is contingent on the cluster type, with SSDs offering higher I/O performance at a premium. Users may need to scale storage nodes as data volumes increase, driving up costs. BigQuery, however, separates storage and computing, charging $0.02 per GB per month, with further reductions for inactive data after 90 days, making it suitable for storing large, infrequently accessed datasets.

8. Additional Insights and Factors to Consider

When choosing between Amazon Redshift and Google BigQuery for big data processing, workload type, and scalability are primary criteria to decide on when choosing the correct platform for an organization.

Workload Types

Redshift most effectively performs ETL operations, transforming data with the help of SQL statements and supporting bulk load operations. By comparison, BigQuery is geared towards ELT processes because the data warehousing platform has serverless operating characteristics and may immediately process data after loading. Reporting & business intelligence—Redshift, because of its ability to manage resources well, enables predictable, scheduled queries. At the same time, BigQuery is perfect for ad hoc analysis that is often on-demand while using BI tools like Tableau.

Scalability and Elasticity

Redshift's scalability is related to its cluster architecture, where new nodes must be added by hand and can be done offline. Concurrency scaling in Redshift: for the same, there is the possibility to increase the load for a certain period without a permanent increase in the number of clusters (Salqvist, 2024). BigQuery, on the other hand, can scale resources infinitely without the end user having to configure it as it is a fully serverless solution manually. On the one hand, it is beneficial in regard to fluctuating workloads; on the other hand, it may produce uncertain expenses when there are no definite rules for resource consumption.

9. Conclusion

This comparative study posits that while Amazon Redshift and Google BigQuery are both capable solutions for large-scale data analytics, they satisfy different organizational requirements. In the case of small data up to 100GB, BigQuery does not require expensive nodes and is more flexible as it has a serverless architecture, while in Redshift, we have additional costs for writing data. However, on the larger datasets, Redshift has dedicated resources for predictable workloads, which are consistently performant for these predictable workloads, which can be compared to BigQuer's on-demand scaling for unpredictable workloads (AirByte, 2024, March). Redshift is preferred for non-stopping and complex analytics inquiries, including inquiries involving ETL

processes. BigQuery is best suited to occasional querying and procedures that can be easily solved using big data with little to no infrastructure.

Further, both platforms are likely to grow in the future; Redshift is expected to concentrate on concurrency and real-time analytics, while BigQuery can build on AI/ML and other efficiency features and cost optimization. If both continue to improve and develop, the organization will choose which to use depending on the need at a given time.

References

[1]. AirByte. (2024, March). BigQuery vs. Redshift: Comparing Two Leading Data Warehouse Solutions. Airbyte.com; Airbyte. https://airbyte.com/data-engineering-resources/bigquery-vs-redshift

[2]. Demirbaga, Ü., Aujla, G. S., Jindal, A., & Kalyon, O. (2024). Cloud Computing for Big Data Analytics. In Big Data Analytics: Theory, Techniques, Platforms, and Applications (pp. 43-77). Cham: Springer Nature Switzerland. https://link.springer.com/chapter/10.1007/978-3-031-55639-5_4

[3]. Hook, D. W., & Porter, S. J. (2021). Scaling scientometrics: Dimensions on Google BigQuery as an infrastructure for large-scale analysis. Frontiers in Research Metrics and Analytics, 6, 656233. https://www.frontiersin.org/articles/10.3389/frma.2021.656233/full

[4]. Miryala, N. K., & Gupta, D. Big Data Analytics in Cloud–Comparative Study. DOI, 10, 22312803. https://www.researchgate.net/profile/Divit-Gupta/publication/376892490_Big_Data_Analytics_in_Cloud_-Comparative_Study/links/658e5eec3c472d2e8e9a411e/Big-Data-Analytics-in-Cloud-Comparative-Study.pdf

[5]. Nathan, V., Singh, V., Liu, Z., Rahman, M., Kipf, A., Horn, D., ... & Kraska, T. (2024, June). Intelligent Scaling in Amazon Redshift. In Companion of the 2024 International Conference on Management of Data (pp. 269-279). https://dl.acm.org/doi/abs/10.1145/3626246.3653394

[6]. Van Renen, A., & Leis, V. (2023). Cloud analytics benchmark. Proceedings of the VLDB Endowment, 16(6), 1413-1425. https://dl.acm.org/doi/abs/10.14778/3583140.3583156

[7]. Veeresh Biradar. (2024, April 12). Redshift vs BigQuery: 7 Critical Differences. Learn | Hevo; Hevo Data. https://hevodata.com/learn/redshift-vs-bigquery/