Lecture #21: Amazon Redshift Re-invented

15-721 Advanced Database Systems (Fall 2024)

https://www.cs.cmu.edu/~15721-f24//

Carnegie Mellon University

Author: Xianqi Wang (Andrew ID: xianqiw)

1 Introduction

Amazon Redshift, launched in 2013 by Amazon Web Services (AWS), revolutionized the data warehousing industry as the first fully-managed, petabyte-scale, cloud-based enterprise-grade solution. By eliminating the complexity and high costs associated with traditional on-premise data warehouses, Redshift empowered businesses to efficiently analyze massive datasets using familiar business intelligence tools. It rapidly gained traction, becoming AWS's fastest-growing service and now supports tens of thousands of customers processing exabytes of data daily across the globe.

Over the years, Amazon Redshift has continued to evolve to meet diverse use cases, incorporating innovations such as tiered storage, cross-cluster data sharing, and the AQUA query acceleration layer. These enhancements ensure unmatched performance and scalability. The introduction of autonomics has further simplified usability, culminating in Redshift Serverless, which enables users to run analytics without infrastructure management. Additionally, Redshift's integration with the broader AWS ecosystem—supporting data lakes, streaming ingestion, and machine learning—has extended its capabilities far beyond traditional data warehousing.

2 Architecture and Design Principles

2.1 Separation of Compute and Storage

Amazon Redshift employs a disaggregated architecture that separates compute and storage layers to enhance scalability and cost efficiency. The compute layer, comprising a leader node and multiple worker nodes, handles query parsing, optimization, and execution, while the storage layer is built on Amazon S3, ensuring durability with 11-nines of reliability. By decoupling these layers, Redshift allows independent scaling of compute resources and storage capacity.

For instance, an e-commerce company analyzing seasonal sales data can dynamically scale compute resources during peak shopping periods without increasing storage costs unnecessarily. Data frequently accessed remains cached on high-speed SSDs attached to compute nodes, while colder data resides in S3.

This architecture also facilitates features such as elastic resizing and cross-cluster data sharing. The elastic resizing capability ensures clusters remain balanced during scaling operations, with data rehydrated into local SSD caches based on workload patterns. Cross-cluster data sharing enables read-only access to datasets across independent clusters, simplifying collaborative analytics.

Overall, the separation of compute and storage not only optimizes performance and cost but also introduces flexibility in handling diverse workloads.

- **Compute Layer:** The compute layer consists of a leader node for query parsing, optimization, and coordination. Compute nodes perform the actual data processing.
- **Storage Layer:** The storage layer relies on Amazon S3, offering unmatched durability (11-nines) and scalability to handle vast amounts of data.

The separation enables businesses to scale compute and storage resources independently. For example, during seasonal spikes, an online retailer can increase compute resources to handle analytics on customer behavior while keeping storage requirements steady.

Key Benefit: This modular approach optimizes cost and performance, ensuring high efficiency even during fluctuating workloads.

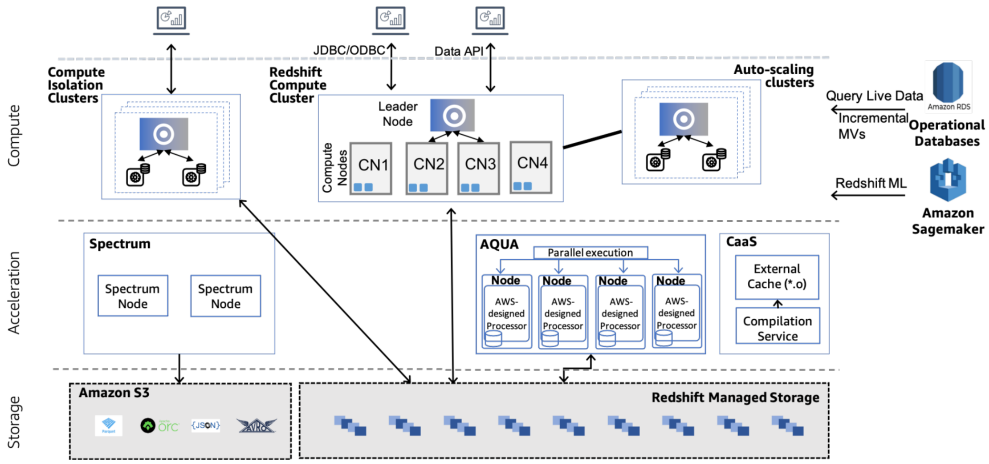


Figure 1: Architecture of Amazon Redshift

2.2 Dynamic Query Compilation

Dynamic query compilation is a cornerstone of Redshift's performance. Each query fragment is dynamically compiled into C++ code optimized for the underlying hardware, enabling efficient execution. By leveraging techniques such as SIMD vectorization and late materialization, Redshift minimizes data movement and maximizes parallelism.

Consider a query joining customer and order datasets by `customer_id`. Redshift's optimizer generates a physical plan that co-locates relevant data on the same compute node, avoiding network overhead. The code snippet below illustrates how this process works:

- Utilizes SIMD vectorized instructions to parallelize tasks efficiently.
- Implements co-located joins to minimize shuffle operations across nodes.
- Compresses data to reduce I/O overheads.

For instance, a co-located join eliminates the need for data transfer across the network when tables are partitioned using the same distribution key. This drastically reduces query latency.

Example: Joining datasets on a common key such as `customer_id` results in optimized query execution and faster performance.

```
SELECT c.name, SUM(o.amount)
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
WHERE o.date > '2024-01-01'
GROUP BY c.name;
```

3 Scalability and Performance Enhancements

3.1 Elastic Scaling

Redshift's elasticity addresses fluctuating workload demands, enabling dynamic scaling of compute nodes. During high-demand periods, such as end-of-quarter financial reporting, additional nodes can be provisioned to meet workload demands, ensuring timely query execution. Conversely, during periods of low activity, nodes can be scaled down to save costs. For example, a retailer preparing for a major sales event like Black Friday can temporarily scale out its Redshift cluster to handle the surge in transactions. Afterward, the cluster can be scaled back to its baseline configuration. Elastic scaling operates seamlessly, with data rehydrated into local caches as needed, ensuring minimal disruption to operations. This capability highlights Redshift's adaptability, providing cost-effective performance tailored to business needs. Redshift provides seamless elasticity by dynamically scaling compute resources based on workload demand:

- Additional nodes are auto-provisioned during peak usage, such as quarterly financial reporting.
- Nodes are scaled down during idle periods, saving costs.

For instance, during a retail sales event like Black Friday, Redshift scales horizontally to handle millions of transactions and scales back afterward.

3.2 Concurrency Scaling

Concurrency scaling enhances Redshift's ability to handle high-query workloads by automatically provisioning temporary clusters. These clusters absorb excess traffic, maintaining fast query response times. Each Redshift cluster can handle up to 200 concurrent queries, with additional clusters activated on-demand.

For instance, an analytics platform serving thousands of users can rely on concurrency scaling to ensure a smooth user experience, even during traffic spikes. Temporary clusters are seamlessly integrated into the system and deactivated when no longer needed, optimizing resource utilization. Concurrency scaling allows thousands of users to run queries simultaneously by spinning up temporary compute clusters:

- Supports up to 200 concurrent queries per cluster.
- Automatically distributes workload to secondary clusters, reducing queue times.

3.3 Performance Optimization

Performance optimizations in Redshift include the use of materialized views, query caching, and adaptive query execution. Materialized views store precomputed query results, significantly reducing execution times for recurring queries. Query caching reuses results from frequently executed queries, further enhancing performance.

Additionally, Redshift employs adaptive query execution, which dynamically adjusts execution strategies based on runtime statistics. This approach ensures efficient use of system resources, even for complex queries involving large datasets.

The combination of these features ensures that Redshift consistently delivers industry-leading performance across a wide range of use cases. Redshift excels in performance through:

- **Materialized Views:** Precomputes results for faster access.
- **Query Caching:** Reuses results of frequently run queries.
- **Adaptive Query Execution:** Dynamically adjusts execution strategies based on resource usage.

Example: Creating a materialized view for regional sales:

```
CREATE MATERIALIZED VIEW sales_summary AS
SELECT region, SUM(sales) AS total_sales
FROM transactions
GROUP BY region;
```

This improves performance for reports accessed multiple times daily.

4 Integration with the AWS Ecosystem

4.1 Zero-ETL Data Ingestion

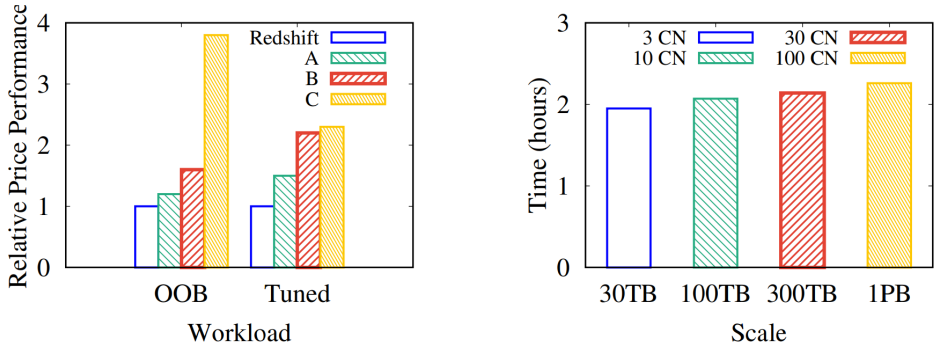
Redshift integrates seamlessly with AWS services like Aurora and DynamoDB via zero-ETL pipelines:

- Directly reads Aurora's storage layer for near-real-time updates.
- Ingests DynamoDB streams to handle high-velocity data.

Example: A logistics company using DynamoDB to track deliveries can sync real-time updates into Redshift, enabling live dashboards.

4.2 Support for Semi-Structured Data

The `SUPER` data type supports querying semi-structured data like JSON. With the PartiQL query language, users can access nested data effortlessly:



(a) Price-Performance comparison (b) Scaling TPC-DS from 30TB to 1PB

Figure 2: Price-Performance and Scalability

```
SELECT json_field->'customer', json_field->'order_value'
FROM orders_table
WHERE json_field->'status' = 'Shipped';
```

This simplifies processing of diverse data formats without requiring preprocessing.

5 Advanced Analytics and Machine Learning

5.1 In-Database Model Training

Redshift integrates machine learning workflows into SQL. Users can train models directly within the database:

```
CREATE MODEL sales_forecast
FROM SELECT * FROM sales_data
USING 'linear_regression';
```

Trained models can be deployed for inference, making it easier to incorporate predictions into analytics.

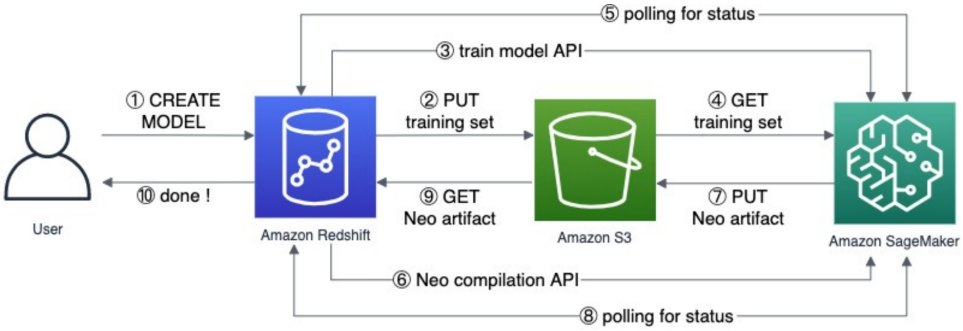


Figure 3: Redshift: Create model

5.2 Real-Time Predictions

Redshift allows real-time predictions by integrating with SageMaker:

```
SELECT customer_id, predict_churn(features)
FROM customer_profiles
WHERE last_purchase_date > '2024-01-01';
```

This helps businesses identify at-risk customers and deploy retention strategies promptly.

5.3 Streaming Analytics

Redshift supports streaming ingestion from AWS services like Kinesis. This enables near-instant analytics for use cases such as fraud detection and IoT monitoring.

Conclusion

Amazon Redshift remains a trailblazer in the cloud data warehouse space. Its architectural innovations, scalability, and seamless integration with analytics and machine learning make it a preferred choice for enterprises. Redshift's ability to adapt to evolving business needs ensures its continued relevance and success in the cloud era.