Snowflake vs Redshift: Which Cloud Data Warehouse is Right for You?

Naresh Dulam, Vice President Sr Lead Software Engineer, JP Morgan Chase, USA

Abhilash Katari, Engineering Lead, Persistent Systems Inc, USA,

Karthik Allam, Big Data Infrastructure Engineer, JP Morgan & Chase, USA

Abstract:

Cloud data warehouses have fundamentally changed how businesses manage and analyze large volumes of data, offering enhanced speed, scalability, and flexibility. Two of the most prominent platforms in this space, Snowflake and Amazon Redshift, stand out for their ability to support complex analytical workloads. Still, they differ significantly in their architecture and capabilities. Snowflake, known for its unique multi-cluster, shared-data architecture, offers high scalability & performance by decoupling storage and computing, enabling users to scale resources independently and optimize cost efficiency. Its ability to automatically scale & handle concurrent workloads without affecting performance makes it a popular choice for modern, data-intensive businesses. On the other hand, Amazon Redshift, a part of the AWS ecosystem, provides a more traditional, columnar data warehouse architecture designed to deliver fast query performance for large-scale datasets. With deep integration into the AWS cloud, Redshift is often the go-to choice for organizations already using AWS services, as it benefits from native integrations with tools like Amazon S3, AWS Lambda, & more. While Redshift offers robust performance and strong data compression capabilities, its scalability is more limited than Snowflake's ability to separate computing & storage. Cost structures also vary, with Snowflake charging based on actual usage, offering more predictable pricing. At the same time, Redshift follows an on-demand or reserved pricing model that can be advantageous for longer-term workloads. Additionally, Snowflake's ease of use, particularly its user-friendly interface and SQL compatibility, contrasts with Redshift's slightly steeper learning curve. Both platforms excel in different areas, and choosing the right one depends on various factors, including organizational goals, existing cloud infrastructure, and specific data processing needs. By weighing performance, cost, scalability, and

© 2018 All Rights Reserved

ecosystem fit, businesses can determine which platform is best suited to support their data warehouse requirements.

Keywords: Snowflake, Amazon Redshift, cloud data warehousing, performance, scalability, cost, analytics, data integration, data storage, concurrency, semi-structured data, real-time analytics, query optimization, cloud ecosystem integration, security features, deployment flexibility, on-demand scaling, pay-as-you-go pricing, reserved pricing, query performance, data sharing, data lakes, multi-cloud, ETL processes, data governance, business intelligence, SQL compatibility, data modeling, data architecture, workload management, auto-scaling, cloud-native architecture, elasticity, high availability, data replication, data partitioning, columnar storage, data compression.

1.Introduction

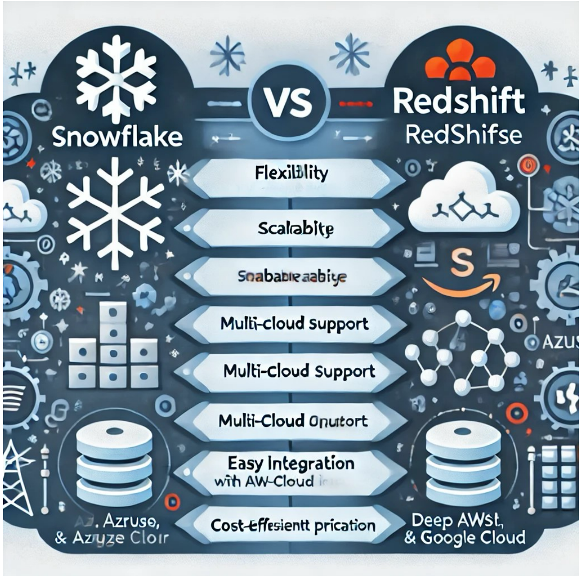
Organizations are increasingly adopting cloud-based solutions to handle the vast amounts of data they generate. With the rapid evolution of cloud technologies, two cloud data warehouses have risen to prominence: Snowflake and Amazon Redshift. Both platforms offer scalable, high-performance solutions to store and analyze data, but each comes with its own set of features, benefits, & limitations. As businesses seek to harness the power of big data, understanding which platform suits their needs is critical.

1.1 Snowflake Overview

Snowflake has rapidly gained popularity due to its unique architecture and ease of use. Unlike traditional data warehouses, Snowflake is built from the ground up as a cloud-native solution, designed to handle a variety of data types, including structured, semi-structured, and unstructured data. Its architecture separates storage and compute, allowing users to scale each independently based on their workload requirements. This means that businesses can optimize their performance without being constrained by the need to scale storage and computing resources together.

One of Snowflake's standout features is its ability to handle semi-structured data, such as JSON, XML, and Avro, without requiring complex transformations. This makes it particularly appealing for organizations dealing with diverse datasets or looking to move beyond traditional SQL databases. Additionally, Snowflake's ease of use and quick setup time allow organizations to hit the ground running, minimizing the need for extensive training or expertise.

© 2018 All Rights Reserved



1.2 Amazon Redshift Overview

Amazon Redshift, a product of Amazon Web Services (AWS), is one of the most well-established players in the cloud data warehouse market. Redshift is based on PostgreSQL and uses a columnar storage model to efficiently store large volumes of data for analytics. It integrates seamlessly with other AWS services, making it an attractive choice for businesses already committed to the AWS ecosystem.

Redshift's architecture is designed for high performance, particularly when running complex queries across massive datasets. While it offers scalability, it does require users to make decisions about compute & storage resources upfront, which can sometimes be a challenge for organizations that experience fluctuating data workloads. However, Redshift's extensive optimization tools, like its workload management features, give users the flexibility to fine-tune performance to meet their needs.

1.3 Snowflake vs. Redshift: What Sets Them Apart?

While both Snowflake and Redshift are highly capable cloud data warehouses, the key differences between them lie in their architecture, pricing models, & ease of integration. Snowflake's decoupled storage and compute architecture make it easier to scale based on demand, while Redshift's deep integration with AWS gives it an edge for businesses already leveraging AWS services. Pricing structures also differ, with Snowflake offering a consumption-based model and Redshift charging based on instance size and usage.

2. Overview of Snowflake & Amazon Redshift

Cloud-based platforms like Snowflake and Amazon Redshift have become dominant players. Each offering has its unique advantages, but understanding which one is right for your organization depends on several key factors. This section will provide an overview of both platforms, highlighting their features, strengths, and differences, to help you make an informed decision. Before 2018, both Snowflake and Redshift had already gained significant

© 2018 All Rights Reserved

traction, but the landscape was shifting, with each platform carving out its niche in the competitive cloud data warehousing space.

2.1 Snowflake Overview

Snowflake is a relatively newer player in the cloud data warehousing market, but its innovative architecture has quickly won over a large portion of the market. Founded in 2012, Snowflake was designed from the ground up for the cloud, allowing it to leverage the elasticity and scalability inherent in cloud computing. Unlike traditional on-premise data warehouses that rely on shared storage and compute resources, Snowflake uses a multi-cluster architecture that separates storage from compute, offering several key advantages in terms of performance, scalability, and cost efficiency.

2.1.1 Data Sharing & Collaboration

Another advantage of Snowflake is its data sharing capabilities. Snowflake allows seamless and secure data sharing between different organizations, departments, or users within the same organization. This is particularly useful for enterprises that need to work with large datasets or collaborate across different business units. Snowflake's Secure Data Sharing feature enables organizations to share live data with partners, clients, or external stakeholders in real-time without moving the data.

Snowflake's data sharing approach is far more intuitive and easier to use compared to many legacy data warehouses, where data transfer often requires complex ETL (Extract, Transform, Load) processes.

2.1.2 Architecture & Design

One of Snowflake's standout features is its unique architecture. The platform is built on top of Amazon Web Services (AWS) or Microsoft Azure, but it doesn't just offer a basic re-skinning of existing technology. Instead, Snowflake employs a multi-cluster shared data architecture, meaning it can scale compute and storage resources independently. This allows users to scale up or down based on needs without impacting the overall system performance.

Another key architectural benefit is Snowflake's automatic management of infrastructure. This reduces the need for manual optimization, a major pain point in traditional data warehousing, where users often had to tweak and optimize settings to ensure efficiency. Snowflake's automatic scaling ensures users only pay for what they need, with no manual intervention required.

2.2 Amazon Redshift Overview

© 2018 All Rights Reserved

Amazon Redshift, launched by AWS in 2013, is one of the most popular cloud data warehousing solutions. It is a fully managed data warehouse service that can handle large-scale data storage and analysis workloads. Redshift is designed to work seamlessly with the AWS ecosystem, offering tight integrations with other AWS services like S3, EMR, Kinesis, and Glue, making it an attractive choice for companies already embedded in the AWS cloud ecosystem.

2.2.1 Pricing Model & Cost Efficiency

Redshift's pricing model is based on the type and number of nodes used for data storage and processing. Customers can choose between dense compute nodes or dense storage nodes, depending on the use case. While this provides flexibility, it also requires more attention from the user, as they must manage the scaling of compute and storage resources to avoid inefficiencies.

Redshift has traditionally been known for its cost efficiency, especially for workloads with predictable usage patterns. It is an attractive option for organizations that have a large amount of data and want to ensure that they are only paying for what they use. However, it can become less cost-effective for ad-hoc querying or fluctuating workloads, as users may over-provision resources or fail to scale down effectively.

2.2.2 Architecture & Performance

Redshift's architecture is built around columnar storage and massively parallel processing (MPP), which is particularly well-suited for performing complex queries on large datasets. The architecture divides workloads into smaller tasks and distributes them across multiple nodes, allowing it to process data much faster than traditional systems.

Unlike Snowflake, Redshift typically requires users to define clusters of resources. This is because Redshift uses a more traditional approach where storage and compute are tightly coupled. As a result, users often need to optimize and configure clusters to ensure performance is up to par, especially as workloads grow and evolve.

2.2.3 Ecosystem & Integrations

Being a part of the AWS ecosystem, Amazon Redshift integrates easily with other AWS services. This makes it particularly appealing for organizations already using AWS for other services like EC2, Lambda, or S3. Redshift also offers strong integration with third-party tools and software, ensuring compatibility with a wide range of data analytics and BI tools.

© 2018 All Rights Reserved

However, one of Redshift's key challenges has been its tight coupling of storage and compute, which, as mentioned earlier, requires more configuration and management from the user.

2.3 Key Differences Between Snowflake & Redshift

While both Snowflake and Amazon Redshift are powerful cloud data warehousing platforms, they differ significantly in terms of architecture, scalability, ease of use, and cost models. To help clarify these differences, let's compare the two platforms across several important dimensions:

2.3.1 Scalability & Flexibility

Snowflake: Snowflake offers automatic scaling, which is ideal for businesses that experience fluctuating workloads. It also allows compute and storage to scale independently, making it an attractive option for organizations with unpredictable data needs.

Redshift: Redshift requires more manual intervention when it comes to scaling, as users need to manage their clusters. Though Redshift offers some flexibility, it is generally better suited for organizations with stable, predictable workloads where scaling is less of a concern.

2.3.2 Architecture

Snowflake: The multi-cluster architecture allows for elastic scalability, independent scaling of compute and storage resources, and automatic management, which simplifies the process for users. Snowflake was built specifically for the cloud, and its architecture reflects this focus on elasticity and performance optimization.

Redshift: Built around a more traditional architecture that combines compute and storage into tightly coupled nodes, Redshift relies on the user to manually scale and optimize clusters to handle larger datasets. While it supports massively parallel processing, users need to manage their own infrastructure more closely than they would with Snowflake.

3. Architecture & Design: Snowflake vs Redshift

When deciding between Snowflake and Amazon Redshift for your cloud data warehouse solution, one of the critical factors to consider is the underlying architecture and design of each platform. Both Snowflake and Redshift offer robust features, but they differ significantly in how they handle data storage, processing, and scalability. In this section, we will break down the architecture and design principles of both platforms and compare their strengths and weaknesses.

© 2018 All Rights Reserved

3.1 Snowflake Architecture

Snowflake is built from the ground up to take full advantage of cloud-native technologies. Its architecture is unique in the world of data warehousing because it separates compute and storage, making it highly scalable and flexible.

3.1.1 Data Storage Layer

The storage layer in Snowflake is designed to be completely separate from compute resources. Snowflake uses a centralized storage system, which is built on top of cloud storage (e.g., AWS S3, Microsoft Azure Blob Storage, or Google Cloud Storage). The data is stored in a columnar format, optimized for analytical queries.

Benefits:

- **Seamless scaling:** Since storage is cloud-native, Snowflake can scale automatically as your data grows. It allows for virtually unlimited storage capacity without worrying about storage limitations.
- **Zero-copy cloning:** Snowflake's architecture supports zero-copy cloning, which means users can clone databases, schemas, and tables without physically copying the data, making it incredibly efficient for testing or data migrations.

3.1.2 Multi-Cluster Shared Data Architecture

Snowflake operates on a multi-cluster shared data architecture, which is a major departure from traditional database systems. This architecture allows multiple compute clusters to access a single shared data store simultaneously without causing contention. The separation of compute and storage means that users can scale each component independently to meet the demands of their workloads.

Benefits:

- **Scalability:** The ability to scale compute resources (virtual warehouses) up or down as needed allows for better performance during high-demand periods, without disrupting other processes.
- **Concurrency:** Because each virtual warehouse operates independently, multiple users can query the same data at the same time without impacting performance.

3.2 Redshift Architecture

Amazon Redshift, while also a powerful cloud-based data warehouse, follows a more traditional design based on the shared-nothing architecture model. It is heavily optimized for high-performance

© 2018 All Rights Reserved

querying over large datasets, but it operates differently than Snowflake.

3.2.1 Columnar Storage

Like Snowflake, Redshift also uses columnar storage, which enables better compression and speeds up query performance, especially for analytic workloads. However, Redshift's approach to storage and computing is more tightly coupled.

Benefits:

- **Query performance:** Columnar storage is ideal for analytic queries, as it allows Redshift to scan only the relevant columns for a given query, leading to significant performance gains.
- **Data compression:** Redshift applies compression algorithms to reduce the amount of data that needs to be stored, which also helps improve query performance by reducing I/O.

3.2.2 Redshift Spectrum

Redshift offers Redshift Spectrum, which allows users to run SQL queries against data directly in Amazon S3, without the need to load the data into the data warehouse itself. This is particularly useful for analyzing large datasets that are not regularly queried or for ad-hoc analysis.

Benefits:

- **Cost savings:** Spectrum allows users to keep data in S3 while still being able to analyze it, which can result in cost savings for infrequently accessed data.
- **Seamless integration with S3:** Redshift Spectrum integrates seamlessly with other AWS services, making it an attractive choice for organizations already invested in the AWS ecosystem.

3.2.3 MPP (Massively Parallel Processing)

Redshift's MPP (Massively Parallel Processing) architecture is what enables it to scale performance. Data is distributed across multiple nodes, and each node processes a portion of the data in parallel. This allows Redshift to execute complex queries quickly and efficiently across large datasets.

Benefits:

- **High-performance parallel processing:** Redshift can handle large datasets and complex queries efficiently by splitting the workload across multiple nodes in the cluster.
- **Scalability:** Adding more nodes to the cluster allows Redshift to scale out and handle increased loads,

© 2018 All Rights Reserved

although scaling requires resizing the cluster, which may involve downtime.

3.3 Comparison of Key Architectural Features

When comparing the architecture of Snowflake and Redshift, there are several key differences that can influence your decision.

3.3.1 Data Sharing & Collaboration

Snowflake: One of the unique features of Snowflake's architecture is its ability to easily share data between different Snowflake accounts. Users can securely share read-only or full access to their data without needing to copy or move the data. This is facilitated by Snowflake's architecture, which is designed for easy data sharing across departments, teams, or even external partners.

Redshift: Redshift doesn't offer the same level of ease when it comes to data sharing. While Amazon Redshift does support Amazon Redshift Spectrum and AWS Data Exchange, sharing data across different accounts or organizations can be more complex and less seamless than with Snowflake.

3.3.2 Scalability

Snowflake: Snowflake's architecture is designed to scale dynamically, with the ability to increase or decrease compute resources (virtual warehouses) on demand. Since compute and storage are decoupled, scaling storage and compute independently ensures that performance remains optimal, even under varying workloads.

Redshift: Redshift requires manual resizing of the cluster to scale up or down. This means that while Redshift can scale, the process can be more cumbersome and may require downtime during resizing. However, it does provide powerful performance for large-scale queries once the cluster is appropriately sized.

3.4 Design Flexibility

Snowflake and Redshift offer different levels of design flexibility, each suited to different use cases.

3.4.1 Redshift's Flexibility

Redshift is highly flexible as well but requires more upfront planning and manual tuning. While Redshift allows users to create highly customized data models and performance optimizations (e.g., using distribution keys and sort keys), this can be more challenging for teams without experience in tuning the performance of the system.

Benefits:

© 2018 All Rights Reserved

- **Optimized performance:** Redshift's customizability lets users optimize the system for specific types of workloads by adjusting things like distribution styles and sort keys.
- **Tighter AWS integration:** Redshift is deeply integrated with other AWS services, making it a great choice for teams already using the AWS ecosystem.

3.4.2 Snowflake's Flexibility

Snowflake's architecture allows for a high degree of flexibility in terms of data modeling and workload management. Because compute and storage are decoupled, users can optimize both independently, resulting in more granular control over performance and cost.

Benefits:

- **Flexible data models:** Snowflake supports both structured and semi-structured data, allowing users to work with JSON, XML, and Parquet without having to transform the data into a rigid schema.
- **Seamless scaling:** The multi-cluster architecture allows for smooth scalability without downtime, enabling users to adjust resources based on workload demand.

4. Performance

When evaluating Snowflake and Redshift, performance is often a critical factor in choosing the right cloud data warehouse for your organization. Both platforms offer strong performance capabilities, but they have different architectures and approaches to query processing, which impacts their performance in various scenarios. This section will delve into the performance aspects of both Snowflake and Redshift, helping you understand the key differences and how they align with specific use cases.

4.1 Query Execution

One of the primary factors affecting performance is how each data warehouse handles query execution. Both Snowflake and Redshift are designed to handle massive data sets efficiently, but they take different approaches in their architecture and processing engines.

4.1.1 Redshift's Query Execution Model

Redshift, on the other hand, uses a shared-nothing architecture. Each node in the Redshift cluster is responsible for processing a portion of the data, and the data is distributed across the nodes based on a distribution key. Redshift uses

© 2018 All Rights Reserved

columnar storage, which is optimized for OLAP (Online Analytical Processing) queries. It can deliver high performance for large-scale data processing and analytics workloads, but it can be less flexible compared to Snowflake when it comes to scaling compute resources independently of storage.

One key aspect that differentiates Redshift is its use of query optimization techniques, such as data compression, sort keys, and distribution keys, to minimize the amount of data scanned during a query. However, the performance of queries can degrade if not properly configured or if the workload is highly variable.

4.1.2 Snowflake's Query Execution Model

Snowflake uses a unique architecture that decouples storage from compute. This means that compute resources are automatically provisioned and can scale up or down based on the needs of a given query. Snowflake's query execution is optimized for parallel processing, allowing for fast execution of queries on large datasets.

When a query is run, it is distributed across multiple virtual warehouses (compute clusters) that work in parallel, ensuring that resources are efficiently used. This results in faster query performance, especially for complex analytics and data transformations.

4.1.3 Key Takeaways on Query Execution

- **Snowflake:** Automatically scales compute resources up or down and is optimized for parallel processing. Great for complex queries and analytics workloads.
- **Redshift:** Uses a shared-nothing architecture and relies on manual optimization (sort keys, distribution keys) to achieve high performance. Well-suited for large-scale batch processing and OLAP workloads.

4.2 Scalability

Scalability is another crucial performance factor, particularly when dealing with growing datasets and fluctuating workloads. Both Snowflake and Redshift are designed to scale, but their approaches vary significantly.

4.2.1 Redshift's Scalability

Redshift's scalability is more manual. While Redshift can scale horizontally by adding more nodes to a cluster, it requires more configuration and management compared to Snowflake. To scale a Redshift cluster, you need to resize the cluster, which can involve downtime. Furthermore, Redshift's scaling process

© 2018 All Rights Reserved

may require rebalancing data, which can be time-consuming and impact performance temporarily.

However, Redshift offers features like Elastic Resize, which allows you to add or remove nodes dynamically, and Concurrency Scaling, which allows you to add additional resources to handle peaks in query demand. This can be beneficial for handling sudden workloads, but it still requires manual intervention to ensure optimal performance.

4.2.2 Snowflake's Scalability

Snowflake's architecture allows for automatic and seamless scaling of compute and storage resources. You can scale compute resources up & down based on demand, without impacting ongoing queries. Since Snowflake decouples storage from compute, storage can scale independently, and there's no need for manual tuning or reconfiguring of the infrastructure.

Snowflake allows you to run multiple compute clusters (virtual warehouses) simultaneously. Each virtual warehouse can be scaled up or down, and workloads can be distributed across different warehouses. This makes Snowflake highly scalable for organizations that need to handle varying query loads and fluctuating data processing requirements.

4.2.3 Key Takeaways on Scalability

- **Snowflake:** Provides automatic scaling of both compute and storage resources, making it easier to handle fluctuating workloads without downtime.
- **Redshift:** Requires manual intervention to scale, which may involve rebalancing data or resizing clusters, potentially leading to downtime.

4.3 Storage Performance

Both Snowflake and Redshift offer columnar storage, which is optimized for analytics workloads. However, the way they manage storage and optimize for performance differs.

4.3.1 Snowflake's Storage Architecture

Snowflake's storage is entirely managed and separated from compute. This allows for automatic scaling & optimization of data storage without the need for manual intervention. Snowflake uses a multi-cluster, shared-data architecture that enables high performance even when data is distributed across multiple clusters. The data is compressed and stored in a highly optimized format that supports fast query performance.

Since Snowflake's storage is scalable and elastic, users can store vast amounts of data

© 2018 All Rights Reserved

without worrying about performance degradation as the dataset grows. Snowflake also benefits from automatic optimization of storage, including compression and pruning, which results in improved storage efficiency and query performance.

4.3.2 Key Takeaways on Storage Performance

- **Snowflake:** Automatically optimizes storage, with no manual intervention required for compression or optimization, resulting in high scalability and performance.
- **Redshift:** Requires more manual tuning and management for storage optimization, but can still handle large datasets effectively with the right configurations.

4.3.3 Redshift's Storage Architecture

Redshift uses columnar storage for efficient data retrieval, and its storage is tightly coupled with compute resources. While Redshift's columnar storage is optimized for high-performance data queries, it requires users to manage storage efficiently, using features like data compression, sort keys, & distribution keys.

One key benefit of Redshift's storage approach is its ability to store and process massive datasets. Redshift's architecture can scale vertically by adding more storage and compute nodes, but this scaling process requires more management and tuning to ensure the system operates efficiently as data grows.

4.4 Concurrency & User Workloads

Handling multiple concurrent users and workloads is a critical performance consideration for data warehouses. Both Snowflake and Redshift have features to manage concurrency, but their methods and effectiveness vary.

4.4.1 Redshift's Concurrency Model

Redshift's concurrency can be more challenging to manage. While it supports Concurrency Scaling, which can add additional resources when needed, Redshift's shared-nothing architecture means that a large number of concurrent queries can cause performance bottlenecks if not properly configured.

To manage concurrency, Redshift requires users to optimize cluster configurations, such as choosing the right distribution and sort keys. Additionally, Redshift can use Concurrency Scaling to automatically add capacity for higher query demands, but this feature may incur additional costs.

© 2018 All Rights Reserved

4.4.2 Snowflake's Concurrency Model

Snowflake handles concurrency well due to its decoupled architecture. Since compute resources (virtual warehouses) are separate from storage, multiple users can access the same data without contention. Snowflake allows you to run different virtual warehouses for different workloads, ensuring that one query does not impact the performance of others.

Each virtual warehouse in Snowflake can run independent queries, and since there is no contention for compute resources, Snowflake can deliver high performance even with many concurrent users.

4.5 Key Takeaways

Both Snowflake & Redshift offer strong performance, but they do so in different ways. Snowflake is designed with automatic scaling and optimization in mind, making it ideal for organizations that require flexibility and ease of use. Its architecture supports high concurrency, efficient storage management, and seamless scaling without the need for manual intervention.

Redshift, on the other hand, offers robust performance for large-scale OLAP workloads but requires more manual tuning and management to achieve optimal performance. It is a powerful solution for organizations with fixed data architectures that do not require dynamic scaling and need fine-grained control over query optimization.

5. Cost Considerations: Snowflake vs. Redshift

When evaluating Snowflake and Amazon Redshift as potential cloud data warehouse solutions, cost is a critical factor to consider. Both platforms offer unique pricing models that cater to different organizational needs & workloads. This section will break down the cost considerations for both platforms, providing a detailed comparison across various cost factors to help businesses make an informed decision.

5.1 Snowflake Cost Model Overview

Snowflake employs a pay-per-use pricing model, which means you only pay for what you consume in terms of storage and compute resources. This flexibility can be beneficial for businesses with fluctuating workloads or those looking for a cost-effective, scalable solution. The pricing is broken down into three key components: storage, compute, and data transfer.

5.1.1 Storage Costs

Snowflake charges for storage based on the amount of data you store, and its pricing is typically lower than Redshift's. Snowflake's storage cost is tiered, which

© 2018 All Rights Reserved

means that the more data you store, the lower the price per unit of storage. However, you should note that this cost can escalate if your data grows quickly, so optimizing your data storage by regularly pruning old or irrelevant data is essential to keeping costs manageable.

5.1.2 Data Transfer Costs

Data transfer costs in Snowflake occur when moving data between Snowflake and external systems or regions. This can add up depending on how often you move large volumes of data, particularly if your data is spread across multiple regions or cloud providers. Snowflake allows for region-specific pricing, which can help reduce data transfer costs if data is transferred within the same region.

5.1.3 Compute Costs

Snowflake's compute costs are based on the number of compute credits you use, which are charged per second. The number of credits depends on the size and type of virtual warehouse you are using (from X-Small to 4X-Large). This granularity allows you to adjust your compute power based on the workload, making Snowflake a flexible option for businesses with fluctuating data processing demands.

5.2 Redshift Cost Model Overview

Redshift uses a more traditional pricing model, where users pay for the provisioned storage and compute resources. While this might offer more predictability in costs for some businesses, it can also lead to inefficiencies and unnecessary costs if the workload is not optimized.

5.2.1 Storage Costs

Redshift's storage pricing is based on the amount of data you store, but it also includes the cost of the underlying infrastructure. Redshift offers a dense storage (DS) option and a dense compute (DC) option, where DS is cheaper for large amounts of data & DC is suited for high-performance use cases. However, Redshift does not offer the same tiered storage pricing as Snowflake, which means that the storage cost is more linear as your data grows.

5.2.2 Data Transfer Costs

Redshift's data transfer costs are somewhat similar to Snowflake's in that they are associated with data movement between regions or between Redshift & external sources. However, Redshift offers a significant advantage when it comes to moving data within the AWS ecosystem, as transfers between AWS services (such as S3, DynamoDB, or EC2) are free or discounted. However, data transfer

© 2018 All Rights Reserved

between AWS and other cloud platforms, or outside the AWS network, may incur additional costs.

5.2.3 Compute Costs

Redshift uses a fixed-cost model where you provision clusters with a specific number of nodes (with either dense compute or dense storage options). This approach results in predictable costs but can lead to over-provisioning. If your workload demands vary, you might be paying for more resources than you actually need. Unlike Snowflake, where compute resources scale automatically, Redshift requires you to scale your clusters manually, which can result in idle time and higher costs.

5.3 Comparative Cost Analysis

When comparing the cost models of Snowflake and Redshift, businesses must consider several factors such as scale, flexibility, and cost control mechanisms. Each platform has its advantages & potential pitfalls, depending on the specific needs of the organization.

5.3.1 Predictability vs. Efficiency

Redshift's pricing model is more predictable, as it is based on fixed resources (nodes and storage). If your data workloads are fairly stable, this predictability can be beneficial in terms of budgeting. However, this stability comes at the cost of efficiency. Organizations may be paying for idle resources, especially if their workloads are irregular.

Snowflake's pricing model, however, might result in cost volatility, as businesses only pay for the compute and storage they use. While this model offers greater efficiency, it can also lead to unpredictable costs during periods of high data processing demand. Snowflake does offer tools for monitoring & managing costs, which can help mitigate this volatility.

5.3.2 Flexibility & Scalability

Snowflake's consumption-based pricing allows for granular control over costs, as businesses can scale compute & storage independently based on workload demand. This elasticity helps avoid unnecessary costs during idle periods, as users can scale down their resources when they are not needed.

On the other hand, Redshift's fixed-price model may lead to higher upfront costs, especially if businesses over-provision their clusters to accommodate peak workloads. While Redshift does allow scaling of clusters, it does not provide the same level of flexibility and automation as Snowflake.

5.4 Hidden Costs & Optimizing Expenses

© 2018 All Rights Reserved

Both platforms come with their own set of hidden costs that businesses need to account for when calculating their total cost of ownership.

5.4.1 Query Costs

Another hidden cost in both platforms is the cost associated with running queries. In Snowflake, users pay for the amount of compute power used during query processing. Complex queries or frequent queries can rack up significant compute costs. Similarly, Redshift users may incur higher costs if they don't optimize queries effectively or if they run queries on under-provisioned clusters.

Both platforms provide tools to help optimize query performance, but it is important for businesses to continually monitor and optimize their usage to prevent excess spending.

5.4.2 Data Transfer Costs

As mentioned earlier, data transfer can be a significant hidden cost for both Snowflake and Redshift, especially if you are moving large volumes of data between regions or cloud providers. Snowflake's costs for data transfer between regions are particularly noticeable when your organization's data is distributed across multiple regions or third-party platforms.

Redshift's data transfer costs are relatively lower if you are operating exclusively within the AWS ecosystem. However, if your organization relies on multiple cloud providers, the cost of transferring data to & from AWS can add up quickly.

5.5 Cost Optimization Strategies

Both Snowflake and Redshift offer various mechanisms for optimizing costs, ensuring that businesses do not overpay for cloud data warehousing.

- **For Snowflake:**
  - Scaling compute resources based on workload: Snowflake allows users to scale compute resources up or down depending on workload demands. This ensures that businesses don't overpay for idle compute capacity.
  - Storage management: Since storage costs in Snowflake are tied to the amount of data you store, regularly purging unnecessary data can help reduce storage expenses.
  - Optimizing queries: Snowflake allows users to monitor queries and optimize them for

© 2018 All Rights Reserved

performance, helping to reduce compute costs during high-usage periods.

- **For Redshift:**
  - Right-sizing clusters: Since Redshift is based on fixed compute and storage resources, it's essential to right-size clusters to match your business needs, avoiding over-provisioning.
  - Leverage AWS ecosystem: If your business is heavily invested in AWS, taking advantage of free or discounted data transfer between AWS services can help reduce data transfer costs.
  - Optimize query performance: Regularly optimizing queries and setting up automated performance tuning can help reduce unnecessary compute costs by minimizing query time & resource consumption.

6.Conclusion

When deciding between Snowflake and Amazon Redshift for your cloud data warehouse, it ultimately comes down to your specific needs and use case. Snowflake stands out for its simplicity, flexibility, & separation of computing and storage, making it ideal for organizations looking for ease of use and scalability without managing complex infrastructure. Its native support for semi-structured data, automatic scaling, and pay-per-use model make it an appealing choice for businesses looking for efficient, cost-effective solutions. Moreover, Snowflake's multi-cloud architecture offers more flexibility, allowing it to integrate seamlessly with AWS, Azure, and Google Cloud. It could be a game changer for companies with a diversified cloud strategy.

On the other hand, Redshift is a powerful and mature platform, especially for organizations already deeply embedded in the AWS ecosystem. It provides excellent performance for complex queries and data processing, & its deep integration with other AWS services offers a robust, well-supported environment. Redshift offers a more cost-efficient solution for companies that prioritize cost management and are already using other AWS services, especially with its recent improvements in performance and pricing models.

© 2018 All Rights Reserved

However, its architecture requires more hands-on management and configuration, which might be a consideration for organizations without dedicated data engineering teams. Ultimately, both platforms offer unique strengths, so your organization's specific requirements, budget, and cloud strategy should guide the choice.

7.References:

1. Dageville, B., Cruanes, T., Zukowski, M., Antonov, V., Avanes, A., Bock, J., ... & Unterbrunner, P. (2016, June). The snowflake elastic data warehouse. In Proceedings of the 2016 International Conference on Management of Data (pp. 215-226).

2. Fernandes, S., & Bernardino, J. (2016). Cloud Data Warehousing for SMEs. In ICSOFT-EA (pp. 276-282).

3. Ferreira, P. J., de Almeida, A., & Bernardino, J. (2017). Data Warehousing in the Cloud: Amazon Redshift vs Microsoft Azure SQL. In KDIR (pp. 318-325).

4. Devarasetty, N. (2017). Scalable Data Engineering Platforms for AI-Powered Business Intelligence. International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence, 8(1), 1-27.

5. Warehouse, C. P. (2001). The Buyers Guide.

6. Yuhanna, N., Leganza, G., & Lee, J. (2017). The Forrester Wave™: Big Data Warehouse, Q2 2017. Adoption Grows As Enterprises Look To Revive Their EDW Strategy, 17.

7. Gade, K. R. (2017). Integrations: ETL/ELT, Data Integration Challenges, Integration Patterns. Innovative Computer Sciences Journal, 3(1).

8. Kurunji, S. J. (2014). Query optimization for cloud data warehouse (Doctoral dissertation, University of Massachusetts Lowell).

9. Nadipalli, R. (2017). Effective business intelligence with QuickSight. Packt Publishing Ltd.

10. Kathiravelu, P., & Sharma, A. (2017). A dynamic data warehousing platform for creating and accessing biomedical data lakes. In Data Management and Analytics for Medicine and Healthcare: Second International Workshop, DMAH 2016, Held at VLDB 2016, New Delhi, India,

© 2018 All Rights Reserved

September 9, 2016, Revised Selected Papers 2 (pp. 101-120). Springer International Publishing.

11. Brito, J. J. (2017). Data Warehouses na era do Big Data: processamento eficiente de Junções Estrela no Hadoop (Doctoral dissertation, Universidade de São Paulo).

12. Aho, M. (2017). Optimisation of Ad-hoc analysis of an OLAP cube using SparkSQL.

13. Sridhar, K. T. (2017). Modern column stores for big data processing. In Big Data Analytics: 5th International Conference, BDA 2017, Hyderabad, India, December 12-15, 2017, Proceedings 5 (pp. 113-125). Springer International Publishing.

14. Wang, J., Baker, T., Balazinska, M., Halperin, D., Haynes, B., Howe, B., ... & Xu, S. (2017, January). The Myria Big Data Management and Analytics System and Cloud Services. In CIDR (Vol. 47, p. 48).

15. Coates, M. (2017). Designing a Modern Data Warehouse+ Data Lake.

16. Gade, K. R. (2017). Integrations: ETL/ELT, Data Integration Challenges, Integration Patterns. Innovative Computer Sciences Journal, 3(1).

17. Gade, K. R. (2017). Migrations: Challenges and Best Practices for Migrating Legacy Systems to Cloud-Based Platforms. Innovative Computer Sciences Journal, 3(1).

© 2018 All Rights Reserved