International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Optimizing Multi-Team Data Sharing: Evaluating Shared Schemas versus Data Lakes

Naveen Edapurath Vijayan

Manager,Data Science, Amazon
Seattle, WA 98765
nvvijaya@amazon.com

Abstract

In today's data-driven organizations, multiple teams often require access to shared datasets for various analytical and operational purposes. Designing data architectures that facilitate efficient data sharing while maintaining performance, security, and cost-effectiveness is a significant challenge. This paper examines the trade-offs between two architectural approaches for data sharing among multiple teams: using a single Amazon Redshift cluster with shared schemas and controlled access, and implementing a data lake architecture where teams own their compute resources but share data through granted access. By analyzing factors such as performance, scalability, cost efficiency, data governance, security, and team autonomy, the paper provides insights into optimizing data infrastructure for organizations. A case study illustrates practical considerations and outcomes of both approaches. Recommendations are offered to guide organizations in selecting the most suitable architecture for their needs.

Keywords: Amazon Redshift, Data Lake, Data Architecture, Data Sharing, Multi-Team Collaboration, Cloud Computing, Data Governance, Scalability, Cost Efficiency

I. INTRODUCTION

The rapid expansion of data volumes, variety, and velocity in modern organizations has elevated data to a critical asset driving decision-making, innovation, and competitive advantage. Departments such as marketing, finance, operations, and analytics rely heavily on data to inform strategies, optimize processes, and uncover new opportunities. Consequently, multiple teams within an organization often require access to shared datasets for reporting, analysis, and application development. However, facilitating efficient data sharing among teams while maintaining performance, security, and cost-effectiveness presents significant challenges.

Organizations must carefully choose an appropriate data architecture that supports multi-team collaboration without compromising on scalability, performance, or governance. Two common approaches have emerged in this context. The first involves using a shared Amazon Redshift cluster, where multiple teams own specific schemas and share data by granting access permissions. The second approach is adopting a data lake architecture, where teams grant access to shared data stored in a centralized repository (such as Amazon S3) while owning their compute resources for data processing and analysis.

This paper explores the advantages and disadvantages of these two approaches in facilitating data collaboration among multiple teams within an organization. By analyzing factors such as performance and scalability, cost efficiency, data governance and security, team autonomy and collaboration, and data integration and processing, the paper aims to provide insights into optimizing data infrastructure. A case study based on a hypothetical large organization's experience is included to illustrate practical considerations. The ultimate goal is to offer actionable recommendations to guide organizations in selecting the most suitable architecture for their needs.

II. BACKGROUND

A. *The Importance of Data Sharing in Organizations*

Data sharing among teams is essential for fostering collaboration, improving decision-making, and driving innovation. In an era where data is increasingly complex and originates from various sources, organizations must ensure that their data architecture enables seamless access and integration. Effective data sharing allows teams to leverage collective insights, avoid data silos, and maintain a unified view of the organization's information assets.

B. *Challenges in Data Sharing*

Despite its importance, data sharing poses several challenges. These include maintaining data consistency and quality, ensuring security and compliance with regulations, managing access controls, and balancing performance with resource utilization. Furthermore, different teams may have varying requirements for data formats, processing tools, and analytical capabilities, complicating the design of a one-size-fits-all solution.

C. *Overview of Amazon Redshift and Data Lake Architectures*

Amazon Redshift is a fully managed, petabyte-scale data warehouse service that enables fast query performance using columnar storage and massively parallel processing (MPP). It is optimized for complex SQL queries and aggregations on structured data, making it suitable for business intelligence and reporting applications. In a shared cluster environment, multiple teams can own specific schemas within the Redshift cluster and share data by granting access permissions.

A data lake, on the other hand, is a centralized repository that allows storage of structured, semi-structured, and unstructured data at any scale. Data lakes are built on scalable storage solutions like Amazon S3 and support a schema-on-read approach, providing flexibility in data ingestion and processing. In a data lake architecture, teams can own their compute resources (such as Amazon EMR clusters or AWS Glue jobs) for data processing and analysis while sharing data through granted access.

III. AMAZON REDSHIFT WITH SHARED SCHEMAS

In a shared Amazon Redshift cluster environment, the organization deploys a single Redshift cluster that serves as a centralized data warehouse. Each team within the organization owns specific schemas within the cluster, containing tables and views relevant to their domain. Data is ingested into the cluster through Extract, Transform, Load (ETL) processes, often using tools like AWS Glue, Informatica, or custom scripts. Teams can share data by granting access permissions to other teams' schemas and tables, enabling cross-functional collaboration.

This approach offers several advantages. The centralized nature of the cluster simplifies data governance and policy enforcement. Uniform data models and schemas facilitate consistency and make it easier to collaborate on shared data models. Redshift's optimized performance for complex SQL queries and aggregations on structured data allows teams to execute sophisticated analytical queries efficiently.

However, there are challenges associated with this approach. Performance bottlenecks can occur due to concurrent workloads from multiple teams, leading to degraded performance, query timeouts, and slow response times. Scaling compute resources requires resizing the cluster, which may cause downtime and cannot be done on a per-team basis. There is limited elasticity, as compute and storage cannot be scaled independently. Resource contention may lead to conflicts among teams, and complex permission structures can be difficult to manage, posing potential security risks.

IV. Data Lake Architecture with Team-Owned Compute

In data lake architecture, data is stored in a centralized repository, such as Amazon S3, in its raw format. This allows for the storage of structured, semi-structured, and unstructured data at any scale. The schema-on-read approach enables teams to define the schema at the time of data processing, providing flexibility in data ingestion and analysis.

Each team owns and manages its own compute resources for data processing and analysis. These resources can include Amazon EMR clusters for big data processing, AWS Glue jobs for ETL tasks, Amazon Athena for serverless querying, or other AWS services. Teams can choose tools and frameworks best suited to their specific workloads, such as Apache Spark, Hive, or Presto.

Data sharing is facilitated through granted access, with teams setting permissions and policies to control who can access their data within the data lake. Robust data cataloging and metadata management, often implemented using AWS Glue Data Catalog, are essential for discovering and understanding data assets.

The data lake approach offers significant advantages. It provides near-infinite scalability of storage with Amazon S3 and allows compute resources to be scaled independently by each team based on demand. This elasticity enables teams to optimize performance and costs according to their needs. The architecture supports diverse data types and advanced analytics, including machine learning and real-time streaming data. Teams have greater autonomy and can innovate more rapidly without being constrained by centralized infrastructure limitations.

However, the data lake architecture also presents challenges. Robust data governance and security measures are required to manage access controls, ensure data quality, and comply with regulations. There is a risk of data silos if teams do not coordinate effectively, potentially leading to inconsistent data definitions and duplication. Managing data consistency and ensuring collaboration among teams necessitates clear policies and communication.

V. Comparative Analysis

A. *Performance and Scalability*

In a shared Amazon Redshift cluster environment, performance is optimized for complex SQL queries and aggregations on structured data. Redshift's columnar storage and MPP architecture enable efficient execution of analytical queries. However, performance bottlenecks can arise when multiple teams run

concurrent queries, leading to resource contention. Query concurrency limits may impact response times during peak usage, affecting productivity.

Scaling the Redshift cluster to accommodate increased workloads requires resizing the cluster, which may involve downtime. This process is not instantaneous and cannot be done on a per-team basis. The inability to scale compute and storage independently limits elasticity, making it challenging to accommodate sudden spikes in workload without overprovisioning resources.

In contrast, the data lake architecture with team-owned compute offers greater flexibility in performance and scalability. Teams can optimize performance based on their specific workloads by selecting appropriate compute resources and configurations. Distributed processing engines like Apache Spark can handle large-scale data processing efficiently. Compute resources can be scaled independently by each team, enabling elastic scaling based on demand. Services like Amazon EMR and AWS Lambda allow for automatic scaling, providing near-infinite scalability.

However, performance in a data lake environment can vary depending on compute configurations, data formats, and processing frameworks. Ensuring consistent performance may require careful optimization and tuning. Nevertheless, the ability to scale resources independently and leverage various tools provides teams with the flexibility to meet their performance needs.

B. *Cost Efficiency*

Cost efficiency is a critical consideration in choosing a data architecture. In a shared Amazon Redshift cluster, costs include compute nodes, storage, and data transfer within the cluster. Organizations may achieve cost savings by using Reserved Instances and committing to specific capacity. However, allocating costs accurately among multiple teams sharing the cluster can be challenging. Overprovisioning to meet peak demands leads to higher costs, while underutilization during off-peak times results in inefficiency. Teams may have limited control over resource usage, impacting their ability to manage costs effectively.

The data lake architecture with team-owned compute offers advantages in cost efficiency. Storage costs are low with Amazon S3's tiered pricing and lifecycle policies. Compute costs are pay-as-you-go, using services like Amazon EMR, AWS Glue, or Amazon Athena. Teams can optimize compute costs by scaling resources up or down as needed, avoiding the need to provision and pay for idle resources. It is easier to attribute costs to specific teams or projects, providing transparency and accountability. Utilizing spot instances and serverless computing can result in significant cost savings.

C. *Data Governance and Security*

Data governance and security are paramount in any data architecture. In a shared Amazon Redshift cluster, centralized control simplifies governance and policy enforcement. Uniform data models and schemas facilitate consistency and make it easier to manage data assets. Redshift supports role-based access control (RBAC), row-level security, and column-level security, providing mechanisms to protect sensitive data.

However, managing granular permissions among multiple teams can be complex. There is a risk of cross-team access if permissions are not carefully managed, potentially leading to unauthorized data access or modifications. Ensuring compliance with regulations may require additional auditing and monitoring efforts.

In data lake architecture, robust data cataloging and metadata management are essential for governance. Tools like AWS Glue Data Catalog help teams discover and understand data assets. Teams must adhere to organizational data standards and policies to maintain consistency. Security is enhanced through fine-grained access control using AWS Identity and Access Management (IAM) policies and AWS Lake Formation. Data can be encrypted at rest and in transit, using AWS Key Management Service (KMS) for encryption keys. Teams can implement security measures tailored to their needs, providing enhanced isolation between teams.

While the data lake approach offers strong security capabilities, it requires careful planning and coordination to ensure that policies are consistently applied and that teams do not inadvertently create security gaps. Regular audits and compliance checks are necessary to maintain a secure environment.

D. Team Autonomy and Collaboration

Team autonomy and the ability to collaborate effectively are important factors in choosing a data architecture. In a shared Amazon Redshift cluster, teams may have limited flexibility due to centralized management. Changes to the cluster, such as updates or maintenance, affect all teams and require coordination. Resource contention may lead to conflicts, and teams may be constrained by the capabilities and limitations of the shared infrastructure.

However, collaboration can be facilitated in a shared cluster environment. Common tools and interfaces simplify cross-team work, and collaborating on shared data models is straightforward when using the same platform. Centralized data governance ensures that all teams adhere to the same standards and practices.

In the data lake architecture with team-owned compute, teams have greater autonomy. They can choose tools and processing frameworks best suited to their workloads, enabling innovation and agility. Teams can scale resources independently and implement changes without impacting others. Data sharing is facilitated through shared storage and agreed-upon data formats.

Collaboration in a data lake environment requires coordination to avoid data silos and ensure data consistency. Establishing shared data standards, governance policies, and communication channels is essential. While teams have the freedom to innovate, they must also align with organizational objectives and collaborate effectively to maximize the benefits of the architecture.

E. Data Integration and Processing

Data integration and processing capabilities differ significantly between the two architectures. In a shared Amazon Redshift cluster, data must be loaded into Redshift using ETL processes. This involves extracting data from sources, transforming it to match the schema, and loading it into the cluster. ETL processes can be time-consuming and complex, especially when dealing with multiple data sources and formats. Redshift is optimized for structured data with predefined schemas, limiting its ability to handle unstructured or semi-structured data without significant transformation.

In the data lake architecture, data can be ingested in its raw format, supporting a variety of data types. The schema-on-read approach allows teams to define the schema at the time of data processing, providing flexibility and reducing the upfront effort required for data ingestion. Multiple processing frameworks are supported, including Apache Spark, Hive, and Presto, enabling both batch and real-time streaming data

processing. This flexibility makes it easier to incorporate new data sources and formats, and to support advanced analytics and machine learning applications.

VI. Case Study: Data Collaboration in a Large Organization

A. *Company Profile*

Consider a multinational corporation with diverse business units, including marketing, finance, operations, and analytics. The organization has multiple teams requiring access to shared datasets for reporting, analysis, and application development. The data environment includes diverse data sources such as customer relationship management (CRM) systems, enterprise resource planning (ERP) systems, web analytics, and Internet of Things (IoT) devices. The organization faces challenges in facilitating cross-team data access while maintaining security and compliance with regulations. There is a growing demand for advanced analytics and machine learning capabilities to drive innovation and competitive advantage.

B. *Initial Approach: Shared Amazon Redshift Cluster*

Initially, the organization deployed a centralized Amazon Redshift cluster as the primary data warehouse. Each team owned specific schemas within the cluster, containing tables and views relevant to their domain. Data was ingested into the cluster through ETL processes, transforming data from various sources to match the predefined schemas. Access to data was controlled through permissions at the schema and table levels, allowing teams to share data by granting access to other teams' schemas.

Over time, challenges emerged with this approach. Performance bottlenecks became apparent as multiple teams ran concurrent queries, leading to degraded performance, query timeouts, and slow response times. Resizing the cluster to accommodate increased workloads required downtime and was not feasible on a per-team basis. Resource contention led to conflicts among teams, and it was difficult to prioritize workloads without impacting others. Managing granular permissions among multiple teams became complex, and there was a risk of accidental data access or modification by unauthorized teams.

C. *Transition to a Data Lake Architecture*

To address these challenges, the organization transitioned to a data lake architecture. They adopted Amazon S3 as the centralized data repository, storing data in its raw format. Data cataloging and metadata management were implemented using AWS Glue Data Catalog, enabling teams to discover and understand data assets. Each team provisioned their own compute resources, such as Amazon EMR clusters, AWS Glue jobs, or Amazon Athena queries, for data processing and analysis.

Data access was granted through IAM roles and policies, with fine-grained permissions controlling who could access specific data sets. Teams could choose tools and processing frameworks best suited to their workloads, allowing for greater innovation and agility. The architecture supported advanced analytics and machine learning initiatives, leveraging services like Amazon SageMaker for model development and deployment.

D. *Benefits Realized*

The transition to a data lake architecture resulted in several benefits. Teams were able to scale their compute resources based on demand, eliminating performance bottlenecks caused by shared compute. The pay-as-you-go model reduced costs, as teams optimized resource usage and avoided unnecessary

expenses. Enhanced team autonomy allowed teams to innovate more rapidly, implementing analytics projects without being constrained by centralized infrastructure limitations.

Data governance and security were improved through centralized data storage with decentralized compute. Fine-grained access controls prevented unauthorized data access, and robust data cataloging ensured that data assets were well-understood and managed. The organization achieved compliance with data protection regulations and enhanced auditing and monitoring capabilities.

E. Challenges Addressed

The organization addressed challenges related to data consistency by establishing data standards and formats to ensure consistency across teams. Regular data validation and quality checks were implemented to maintain data integrity. Collaboration was facilitated through the creation of a cross-team governance committee to oversee data policies and encourage communication. Shared documentation and best practices were developed to support collaboration.

Change management was an important aspect of the transition. The organization provided training for teams to adapt to new tools and processes, ensuring that team members had the necessary skills and knowledge. Incremental migration minimized disruption, allowing teams to gradually adopt the new architecture while maintaining continuity of operations.

F. Outcomes

The outcomes of the transition were positive. Teams reported faster query times and improved analysis capabilities, reducing the time to insights and accelerating decision-making. Advanced analytics initiatives were enabled, including machine learning projects that leveraged unstructured data sources for richer insights. The organization experienced increased productivity and innovation, aligning with their strategic objectives.

VII. RECOMMENDATIONS

A. When to Choose a Shared Amazon Redshift Cluster

Organizations may choose a shared Amazon Redshift cluster when their data environment primarily consists of structured data with established schemas. If the organization prefers centralized control and has limited need for advanced analytics or handling unstructured data, this approach can be suitable. It simplifies data governance and policy enforcement, and collaboration on shared data models is straightforward.

To optimize the shared cluster environment, organizations should implement workload management to prioritize queries and manage concurrency. Regular monitoring of cluster performance and query optimization is essential to maintain performance. Clear governance policies should be established, and permissions should be regularly audited to ensure security and compliance.

B. When to Opt for a Data Lake Architecture

A data lake architecture is recommended for organizations dealing with diverse data types and sources, requiring flexibility in processing frameworks and tools. If teams need autonomy and scalability to innovate and meet their specific workload demands, the data lake approach is advantageous. It supports advanced analytics and machine learning applications, enabling organizations to leverage data more effectively.

Investing in data cataloging and metadata management is critical in a data lake environment to ensure that data assets are well-understood and accessible. Robust security measures should be implemented using IAM policies and AWS Lake Formation to control access and protect sensitive data. Encouraging collaboration through shared data standards and governance policies helps prevent data silos and maintains consistency.

C. *Hybrid Approach*

In many cases, a hybrid approach that combines elements of both architectures may offer the best balance of performance, flexibility, and cost-effectiveness. Organizations can use a data lake for centralized storage of all data, leveraging Amazon S3's scalability and cost efficiency. For teams requiring high-performance SQL querying, dedicated Amazon Redshift clusters can be deployed, possibly using Redshift Spectrum to query data directly in S3 without data movement.

This approach allows teams to choose the tools best suited to their needs while maintaining a centralized data repository. It balances the benefits of both architectures, providing flexibility and scalability while leveraging the strengths of each platform.

VIII. CONCLUSION

Selecting the appropriate data architecture for multi-team data collaboration depends on the specific needs and priorities of the organization. A shared Amazon Redshift cluster offers simplicity and centralized control but can face limitations in scalability and team autonomy. The data lake architecture with team-owned compute resources provides flexibility, scalability, and cost advantages but requires robust governance and coordination to be effective.

Organizations should assess their data types, workloads, team structures, and long-term goals when making this decision. Factors such as the need for advanced analytics, the diversity of data sources, regulatory compliance requirements, and the desire for team autonomy should be considered. By carefully evaluating these factors and involving stakeholders from different teams, organizations can choose an architecture that aligns with their strategic objectives and maximizes the value of their data assets.

In many cases, adopting a hybrid approach that combines elements of both architectures may provide the most balanced solution. This allows organizations to leverage the strengths of each architecture while mitigating their respective limitations. Ultimately, the goal is to enable efficient data sharing, foster collaboration, and support innovation across the organization.

REFERENCES

1. Amazon Web Services. (2018). Amazon Redshift. https://aws.amazon.com/redshift/
2. Amazon Web Services. (2018). Building Data Lakes on AWS. https://aws.amazon.com/big-data/datalakes-and-analytics/what-is-a-data-lake/
3. AWS Whitepaper. (2016). Data Lake Solution on AWS. https://d1.awsstatic.com/whitepapers/aws-data-lake-solution.pdf
4. Inmon, W. H., & Linstedt, D. (2015). Data Architecture: A Primer for the Data Scientist. Academic Press.

5. Dixon, J. (2010, October 14). Pentaho, Hadoop, and Data Lakes. https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/
6. Gartner. (2016). Data Lakes: Deep Insights. https://www.gartner.com/doc/3165317/data-lakes-deep-insights
7. Sawadogo, P. N., & Darmont, J. (2017). On Data Lake Architectures and Metadata Management. Journal of Information and Data Management, 8(1), 1–14.
8. AWS Big Data Blog. (2015). Implementing Workload Management in Amazon Redshift. https://aws.amazon.com/blogs/big-data/implementing-workload-management-in-amazon-redshift/
9. Amazon Web Services. (2018). AWS Glue Data Catalog. https://aws.amazon.com/glue/data-catalog/
10. Amazon Web Services. (2018). Announcing AWS Lake Formation Preview. https://aws.amazon.com/about-aws/whats-new/2018/11/announcing-aws-lake-formation-preview/
11. Inmon, W. H. (2016). Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump. Technics Publications.
12. Hu, H., Wen, Y., Chua, T.-S., & Li, X. (2014). Toward Scalable Systems for Big Data Analytics: A Technology Tutorial. IEEE Access, 2, 652–687.
13. Chen, M., Mao, S., & Liu, Y. (2014). Big Data: A Survey. Mobile Networks and Applications, 19(2), 171–209.
14. Khan, N., Yaqoob, I., Hashem, I. A. T., Inayat, Z., Ali, W. K. M., Alam, M., & Gani, A. (2014). Big Data: Survey, Technologies, Opportunities, and Challenges. The Scientific World Journal, 2014, Article ID 712826.
15. Amazon Web Services. (2018). Amazon EMR. https://aws.amazon.com/emr/