Five Best Practices for Deploying AWS Data Lakes

May 2021

Table of Contents

Introduction

Data lakes provide a cost-effective, centralized repository for both structured and unstructured data. They are used by enterprises of all kinds, supporting a range of analytic applications — from dashboards to big data processing to data science and machine learning. Data lakes are increasingly used for operational requirements also as their capabilities begin to overlap with the traditional data warehouse.

While early data lakes were often installed on premises, today, the action is in the cloud. Cloud data lakes offered by Amazon Web Services (AWS) and others offer many advantages. They are easy to deploy, manage, are highly scalable and provide rich tools to manage and analyze data. Despite these advantages, data lake users still experience challenges related to performance, managing data workflows and sharing data between different tools and AWS accounts.

This paper suggests five best practices to help data lake architects realize an optimal data architecture built on an AWS data lake. It discusses how the Dremio data lake services can complement and extend the capabilities of the AWS data lake. It also describes how Dremio can help data teams to accelerate queries of data sets stored in AWS storage while reducing data engineering costs and simplifying data access for analysts and data scientists.

The Explosive Growth of Data

Today, organizations are awash in data. According to Statista, the amount of data stored by organizations will grow at a CAGR of 26% through 2024.[1] The reasons behind this continued explosion in data are hardly a surprise to data teams. The combination of the internet, mobile technologies and easily accessible cloud computing have created a perfect storm:

- **Interactions of all kinds have moved online** – From online purchases to customer service interactions to virtual tours to social media interactions.
- **Data is vastly easier to capture** – Gone are the days of keying data. With most data already in digital form, storing data is just an API call away.
- **Storage costs have plummeted** – In 1980, a gigabyte of storage cost roughly USD 100K in today's dollars.[2] Today, storing a gigabyte of data costs 2.3 cents per month in the cloud.[3]
- **Tools have improved by leaps and bounds** – Modern analytic and AI tools have made it far easier to extract valuable insights from big data, increasing the incentives for organizations to retain it.

[1] Source: Statista - Calculation based on data growth forecast from 2020 to 2024
[2] Source: Wikipedia - In 1980, an IBM 3380 with 2 x 1.26GB drives cost USD 251,342 in dollars
[3] Based on published Amazon S3 Standard service pricing

The economics have changed dramatically. Data teams once carefully handcrafted database schemas, considering what data to retain in costly enterprise data warehouses. Today data teams are just as likely to decide to store everything in case it is needed in the future.

The Rise of Data Lake Storage

The term data lake was first coined by James Dixon, CTO of Pentaho, in 2011. Unlike a data warehouse where schemas are defined in advance, data is collected in its natural form in a data lake. Much like an actual lake, it is constantly refreshed by multiple data streams, as illustrated in Figure 1.

Data lakes typically employ a "schema on read" approach. Schemas are applied to data files only when they are read. Data in the data lake may be structured (rows and columns from a database), semi-structured (CSV, log files, JSON) or unstructured (email, text, audio/video).

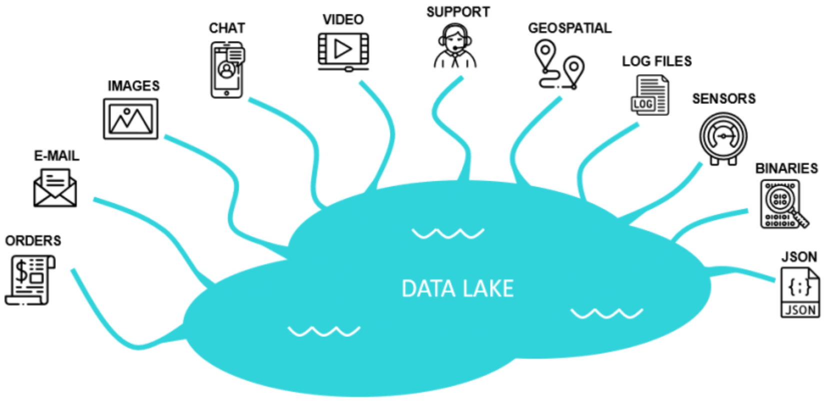

Figure 1 – Data lakes support a variety of structured and unstructured data

Not surprisingly, data lakes are the preferred solution when it comes to storing vast amounts of data. Data lakes are forecast to grow at a CAGR of 27.4 % through 2024, roughly the same pace as data growth overall.[4]

Data Lakes in the Cloud

Early data lakes were built using on-premises Hadoop clusters. These on-premises Hadoop clusters were notoriously difficult to deploy and manage. Capacity in the Hadoop Distributed File System (HDFS) scaled with the number of cluster nodes. The only way to add capacity was to add physical servers, making upgrades and capacity planning a challenge.

Rapid progress in cloud-based services has made it easier to build and maintain data lakes in the cloud. Among the advantages of cloud data lakes are:

[4] Source - Data Lakes Market – Growth, Trends and Forecast (2019-2024)

- **Ease of management** – Customers can quickly deploy large-scale object stores and analytic tools without worrying about on-premises infrastructure.
- **Scalability** – Data teams can start small with only the cloud storage they need and scale seamlessly.
- **Reliability and resiliency** – Customers can spread data across multiple cloud data centers and availability zones and back up data to low-cost archival storage tiers.

Data Lakes on AWS

The AWS Cloud is a preferred platform for implementing secure, flexible and cost-effective data lakes. Amazon provides the building blocks required to help customers quickly ingest, store, process and analyze a variety of data at scale. Foundational data lake cloud services on AWS are the Amazon Simple Storage Service (S3), AWS Glue and AWS Lake Formation. Figure 2 provides a simplified illustration showing how these and other Amazon cloud services fit together.

Amazon also offers tools for data lake query and analysis. Amazon Athena is a serverless environment used to query S3 data lakes using standard SQL. Amazon QuickSight is a serverless, ML-powered BI tool that makes it easier to extract insights from data lakes. AWS data lakes are frequently alongside other Amazon services such as Amazon Redshift (a cloud data warehouse), Amazon EMR (a cloud-hosted Hadoop framework) and Amazon SageMaker (a machine learning platform for data scientists and developers).

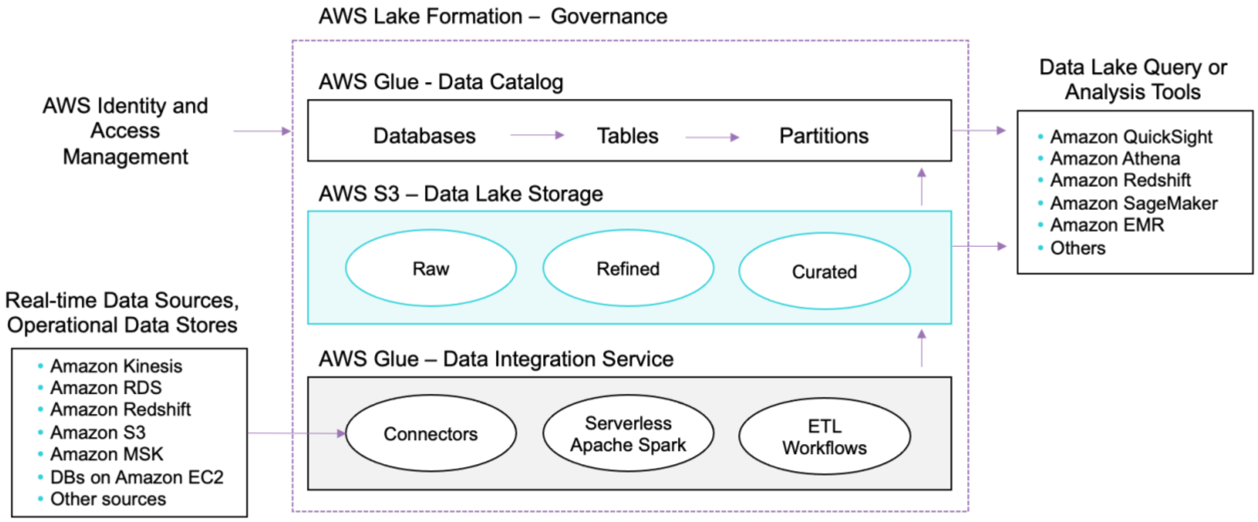

Figure 2 – A simplified view of AWS data lake offerings and how they fit together

Amazon S3

At the heart of the AWS data lake is Amazon S3, a cloud object storage service that offers industry-leading scalability, availability and security. S3 presents a simple, flexible model for storing data in the cloud. While S3 can store any data, tabular data is usually stored using open source formats such as Apache Parquet or the Optimized Row Columnar (ORC). These and other similar S3-friendly file formats have their origins in Hadoop. Data is stored in these optimized formats to provide compression (reducing storage costs on S3) and to support faster, more efficient queries.

AWS Glue

AWS Glue is both a data integration service and a data catalog. Glue discovers and catalogs data from a wide variety of sources, including files stored in S3, Amazon Redshift and Amazon RDS databases. It also provides an ETL engine to cleanse and pre-process data before landing curated datasets in the data lake.

For data in S3 to be useful, it needs to have meaning. Metadata is required that applies structure to raw datasets, including table schemas, field definitions and access control lists. AWS Glue provides this centralized metadata repository via the AWS Glue Data Catalog and provides metadata services for S3 and other data sources. The AWS Glue Data Catalog maintains full compatibility with the Hive Metastore API, widely used in Hadoop environments for compatibility with a broad range of applications and query tools.

AWS Lake Formation

Building on S3 and AWS Glue, Amazon also offers AWS Lake Formation. Lake Formation enables data teams to build data lakes quickly, simplify security management and provide users with self-serve access to data. Lake Formation automates the process of ingesting data from AWS or third-party data stores such as relational and NoSQL databases leveraging the AWS Glue Data Catalog. It uses source crawlers and automated tools for ETL and data preparation tools to ingest data into optimized data lake formats such as Apache Parquet and ORC, where other data engines can easily query it. Lake Formation integrates with AWS Identity and Access Management services, helping organizations define security, governance and auditing policies all in one place rather than setting up policies across multiple services.

Challenges Abound

Despite the rich data lake capabilities in the Amazon cloud, customers often face challenges as they implement cloud data lakes. Among these common challenges are:

- Managing the wide variety of data sources and tools
- Performance-related challenges when querying the data lake
- The proliferation of data copies, aggregation tables and extracts complicating data governance
- Determining a proper data lake topology and sharing data among AWS accounts
- Staying open, flexible and creating a future-proof data lake architecture

While there is no silver bullet to address all these challenges, many can be addressed by following some best practices. Other problems can be avoided by using the right tool for the right job.

Five Best Practices When Deploying AWS Data Lakes

To address the challenges above, the following are some best practices that can help organizations ensure a successful data lake deployment on AWS.

Employ a Data Lake-Centric Design

Most organizations operate a variety of databases to support different data management challenges. For example, transaction-level data may be captured in a relational database. A data warehouse may support reporting and BI applications, and a NoSQL document store may house JSON or XML files.

To explore and analyze data effectively, it is helpful to aggregate data together into a single repository. Ideally, that repository should act as a single source of truth (SSOT).

In modern environments, data can flow in multiple directions. For example, data in a NoSQL database may be loaded into a data lake and combined with other sources for analysis. Data may then be extracted from the data lake and moved to an external data store. Moving datasets is both costly and resource-intensive, however. Data has gravity.

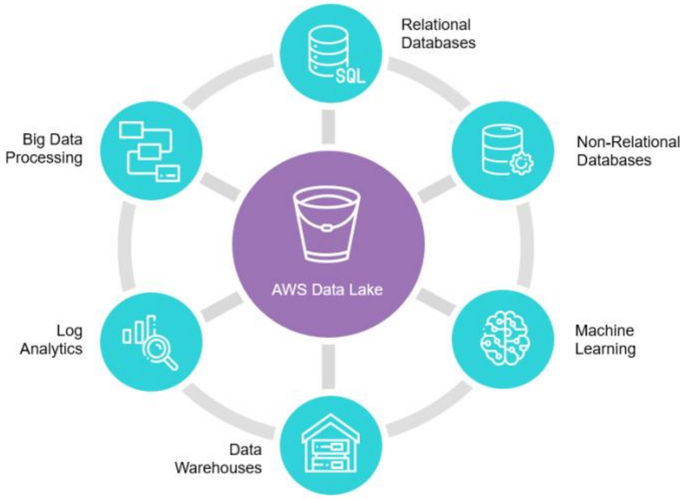

Figure 3 – The data lake as a centralized single source of truth

A good practice is to view the data lake as the central repository for data. For example, data teams should leave data in the data lake where possible and access it directly rather than load data copies from S3 into other services. Users should query the data lake directly using tools such as Amazon Redshift Spectrum or Dremio. This data lake-centric approach is more straightforward and economical since it reduces data replication. It also helps establish the data lake as a single source of truth and helps reduce the need for complex ETL and ELT workflows that are time-consuming to develop and maintain.

Separate Compute from Data

A downside of early Hadoop clusters was that storage and compute capacity scaled in lockstep. The only way to increase data lake capacity was to add compute resources. While Amazon S3 enables storage to scale independently of compute resources, **data and compute** are frequently tightly coupled. For example, Amazon RDS, Redshift or cloud data warehouses such as Snowflake store data in their own native internal formats. This challenge of siloed data environments is illustrated in Figure 4.

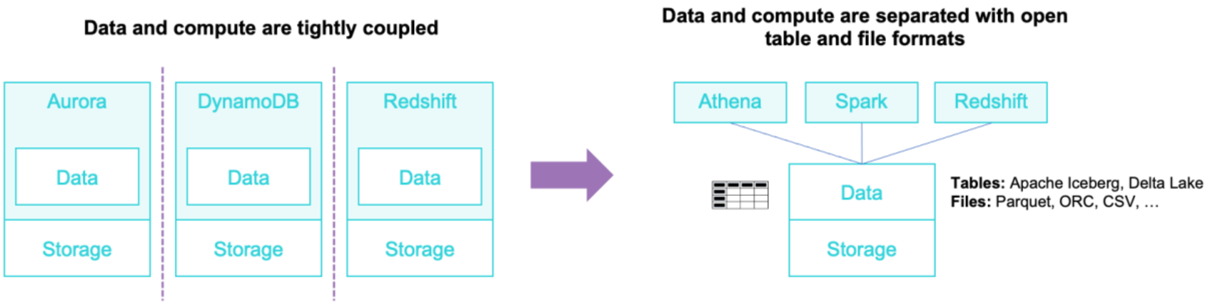
Data and compute are tightly coupled
Aurora
Data
Storage
DynamoDB
Data
Storage
Redshift
Data
Storage
Data and compute are separated with open table and file formats
Athena
Spark
Redshift
Data
Storage
Tables: Apache Iceberg, Delta Lake
Files: Parquet, ORC, CSV, …

Figure 4 – Separate compute and data by using open table and file formats

Until recently, most data engineers viewed this as a necessary evil. However, new open table formats now make it possible to separate data from the compute engines that act on the data.[5] Open table formats such as Delta Lake (developed by Databricks) and Apache Iceberg offer capabilities previously only found in data warehouses. These include atomic transactions, record-level mutations and time travel.

Presto, Snowflake, Dremio and Databricks presently support the Delta Lake table format. AWS supports Delta Lake tables through the AWS Glue Data Catalog using AWS Athena and AWS Redshift Spectrum. Data engineers should consider these open table formats as they present an opportunity to further simplify data lake environments by separating compute from data.

Minimize Data Copies

A frequently overlooked problem in data management is the proliferation of data copies. Organizations copy the same data multiple times, creating performance-optimized copies, personal copies or data copies for offline analysis. An IDC study estimates that organizations, on average, maintain 13 copies of each dataset.[6]

The need to optimize performance or create particular data views are major drivers of proliferating data copies. BI dashboards or reporting applications may require tables with aggregated or pre-sorted data to deliver acceptable performance. In other cases, users may request copies of tables, create OLAP cubes or extract data into external files for offline processing.

[5] Open table formats compared

[6] IT Pro Today article – Too Much Data? Copy—or Copy Data Management

The problem with all of these data copies is that they come at a cost. Not only do they increase storage costs, but data teams spend considerable time developing and maintaining the ETL workflows that support them. Data copies are also a data governance nightmare, particularly in regulated environments.

Fortunately, recent advances in data lake query technologies can dramatically speed up data lake queries and simplify creating intermediate data views. Before using ETL workflows to create query-friendly data copies, data teams should consider modern data engines that can query the data lake directly.

Determine a High-Level Data Lake Design Pattern

Before implementing a data lake, it is essential to consider the organizational structure and how different clients will access data. In some organizations, it may be possible to operate the data lake under a single AWS account using AWS Identity and Access Management (IAM) roles to provide consumers with appropriate access. In other environments, however, data producers and consumers may have different needs.

For example, data consumers such as client departments or business partners may need access to selected data from a centralized data lake. They may also have private applications and data processing requirements better managed in separate AWS accounts. In this case, it may be more appropriate to set up separate accounts for the data producer (the main data lake account) and each consumer, as shown in Figure 5.

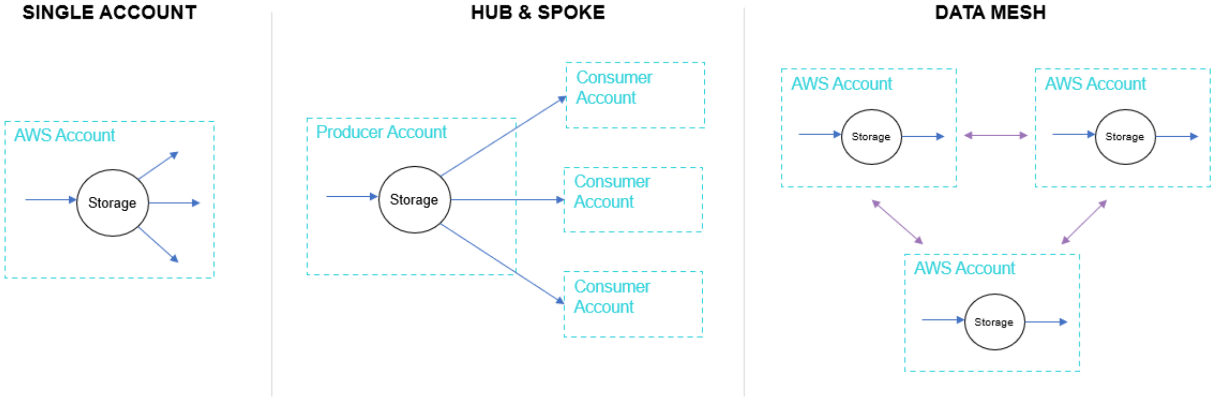

Figure 5 – Common architecture design patterns

In other cases, different organizations may be both producers of data in specific domains and consumers of data in others. In this case, a data mesh topology may be more appropriate. In a data mesh, each data producer retains control over their portion of the data and decides how and with whom to share it. In other cases, organizations may choose to set up a separate account for data lake governance that controls what data views are available to consumers. This approach has the advantage that it separates the roles of data lake management from data lake governance.

Fortunately, AWS Lake Formation and the AWS Glue Data Catalog can support all of these deployment patterns. Data teams can selectively share datasets between AWS accounts, including sharing entire databases, sharing specific tables or only exposing specific columns. Lake Formation allows users to centrally define security, governance and auditing policies in one place and it avoids the need to physically move data between silos.

Stay Open, Flexible and Portable

Finally, as much as it would be convenient to standardize on a single cloud and set of data lake tools, real-world environments are complex. Due to mergers, acquisitions and different departments making independent purchasing decisions, hybrid and multi-cloud environments are a reality for most organizations.

In Flexera's 2020 State of the Cloud Report, 93% percent of respondents indicated that they were implementing a hybrid or multi-cloud strategy.[7]

When designing the data lake architecture, it is worth considering that future requirements can change. It may be necessary to share data with customers and partners in other clouds. A good practice is to store data in open file and table formats and look for solutions that can query data sources across multiple clouds.

The Dremio Data Lake Service

Dremio's data lake service fits seamlessly into the AWS data lake environment. It complements existing AWS services and helps solve common challenges related to data lake implementations.

Dremio supports lightning-fast queries, self-service data access and advanced data governance and security features that complement similar AWS Lake Formation capabilities. Dremio reads data directly from S3 and other sources via the AWS Glue Data Catalog for convenient, centralized metadata management. As illustrated in Figure 6, Dremio complements other AWS cloud services that access the S3 data lake, including Amazon Athena, Amazon Redshift and Amazon EMR.

[7] Source: Flexera 2020 State of the Cloud Report

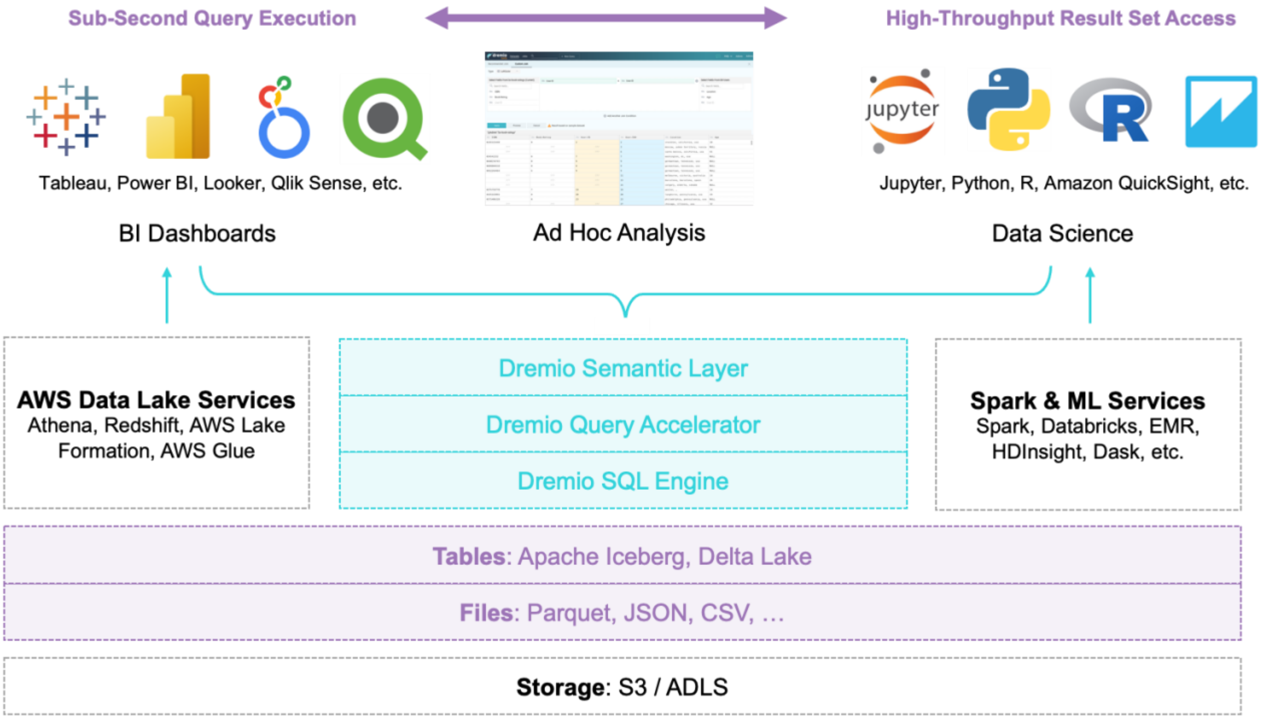
Sub-Second Query Execution
High-Throughput Result Set Access
Tableau, Power BI, Looker, Qlik Sense, etc.
BI Dashboards
Ad Hoc Analysis
jupyter
Jupyter, Python, R, Amazon QuickSight, etc.
Data Science
AWS Data Lake Services
Athena, Redshift, AWS Lake Formation, AWS Glue
Dremio Semantic Layer
Dremio Query Accelerator
Dremio SQL Engine
Spark & ML Services
Spark, Databricks, EMR, HDInsight, Dask, etc.
Tables: Apache Iceberg, Delta Lake
Files: Parquet, JSON, CSV, …
Storage: S3 / ADLS

Figure 6 – Simplified view of the Dremio data lake engine architecture

Lightning-Fast SQL Queries

A key advantage of Dremio over traditional SQL data lake engines is its exceptional query performance. Dremio leverages various open source technologies, including Apache Arrow and Gandiva, to accelerate SQL queries dramatically against S3 data lakes. These technologies combine with Dremio features such as data reflections to deliver between **4-100x the performance** of traditional data lake SQL engines.

Dremio works seamlessly with standard data lake file formats on S3, including Apache Parquet, ORC, Apache Avro and JSON. It also supports multiple metastores, including the Hive Metastore (HMS), AWS Glue and metastores on third-party clouds.

A Self-Service Semantic Layer

In addition to delivering superior performance, Dremio provides a self-serve semantic layer providing multiple virtual views into physical data. Virtual datasets enable data analysts and engineers to manage, curate and share data while adhering to centralized data governance and security policies. This is achieved without the overhead and complexity of copying and moving data. Virtual datasets are fully indexed and searchable. The semantic layer also supports granular role-based access controls and integrates with LDAP, Active Directory and OpenID-based services.

Open Table Formats, Advanced Transactional Capabilities

Open table formats, including open source Apache Iceberg and Delta Lake mean multiple engines (Amazon Athena, Spark, Dremio, etc.) can operate on the same datasets. These table formats bring capabilities to the data lake previously found only in proprietary data warehouses, including:

- Concurrent transactions
- Record-level mutations (updates, deletes, etc.) across large datasets
- Time travel and data versioning

Key Technologies

Dremio has helped pioneer many of the new technologies that make a modern data lake engine possible. Among these technologies are:

- *Apache Arrow* – An open source project enabling columnar in-memory data processing and interchange with distributed, vectorized execution (SIMD or GPU)
- *Arrow Flight* – A parallel zero-copy RPC protocol that exchanges Arrow format memory buffers between a client and a data lake engine with optional parallel high-performance readers
- *Gandiva* – An open source LLVM-based compiler in Apache Arrow that translates queries into vectorized execution kernels for efficient computations
- *Columnar Cloud Cache (C3)* – A real-time, distributed, NVMe-based caching
- *Data reflections* – An optional query acceleration technique in Dremio that transparently re-writes queries to use internal pre-aggregated and sorted materialized views maintained by Dremio, providing a highly optimized physical representation of source data with granular reuse

Dremio Brings New Capabilities to AWS Data Lakes

Dremio integrates seamlessly with S3, AWS Glue and Lake Formation, supporting standard data lake file formats and open table formats. This means that Dremio can run alongside services such as Redshift and Athena, sharing the same physical files and metadata.

Some specific advantages that Dremio brings to AWS data lakes are as follows:

- **Faster queries** – To achieve adequate query performance when using popular reporting and BI tools, customers are often forced to move data into an RDBMS. Dremio achieves performance comparable to a data warehouse directly against the S3 storage with its unique query acceleration technology. Dremio supports any ODBC/JDBC client and provides optimized Arrow Flight-based integrations for popular tools such as Tableau and Power BI to boost performance dramatically. For example, using Dremio's Microsoft Power BI Gateway, Power BI users can directly access S3 data through Dremio datasets without the need to move data between services.
- **A "no copy" strategy avoids ETL/ELT workflows** – Dremio's exceptional performance means that data teams can avoid the need for

performance-optimized copies. This includes intermediate tables, OLAP cubes, data copies and data extracts. Better still, data teams can dispense with the complex ETL/ELT workflows required to build and maintain these intermediate datasets. Like Lake Formation, Dremio provides an internal semantic layer with rich data governance and security features. Users can create shared, customizable, virtualized data views based on the AWS Glue Data Catalog. Creating data views in Dremio can be advantageous in many instances. Dremio data reflections can deliver up to **100x** the performance of traditional data lake queries when querying views based on underlying S3 data.

- **Support for multiple AWS database offerings** – While Dremio is used to query S3 data lakes, Dremio can also access other Amazon and third-party databases. Examples include Amazon Aurora, Amazon RDS, Amazon Redshift, Amazon DocumentDB (MongoDB), Vertica and Snowflake. This means that data teams can join in S3 with data from other sources without the need for ETL. Dremio supports over a dozen data connectors, with many more available at the Dremio Hub.
- **Accelerated data science queries and analysis** – While data science tools often use ODBC or JDBC, these single-threaded record-level query mechanisms are too slow for modern applications. Data science ML tools frequently need to read millions of records to train a single model. To avoid performance penalties, data scientists often extract data and create local copies for offline processing. However, this is a bad practice. Local data extracts break organizational security and governance policies and result in users running models against stale data. Data science users running Jupyter, R or Python applications can take advantage of the Dremio Apache Arrow Flight delivering between **10 and 50x** the performance of pyodbc. This means that data scientists can achieve optimal performance against data lake sources without the need for data extracts that undermine security and data governance concerns.[8]

Dremio is easily deployed via the AWS marketplace and runs on AWS Elastic Compute Cloud (EC2) or Amazon Elastic Kubernetes Service (Amazon EKS). It scales dynamically based on query processing requirements to enable users to minimize costs in the cloud.

Conclusion

For businesses, modern data lakes are essential. They support analytic, BI and data science applications that head to higher quality decisions, efficiency gains and overall improvements in business competitiveness. The AWS Cloud provides an excellent platform for modern data lakes with its scalable and cost-effective S3 object store and rich portfolio of data management, analytics and AI cloud services.

Despite advances in data lake tools, these environments are complicated. Customers can quickly get themselves in trouble with proliferating data copies, challenges meeting performance objectives and complex and costly ETL data pipelines that create a drag on productivity. To avoid these pitfalls, data lake architects can bear in mind some simple best practices. These include:

[8] See article: Eliminating Data Exports for Data Science with Apache Arrow Flight

- Employing a data lake-centric design
- Separating compute from data
- Being mindful of unnecessary data copies
- Selecting a high-level design pattern for data sharing
- Staying open, flexible and portable

Dremio can help data lake architects achieve these goals by making it practical to support a wider variety of analytic, BI and data science workloads directly on the AWS data lake. Dremio can help data teams improve query performance and efficiency, avoid the need for ETL workflows and improve portability and flexibility. It can also help reduce cloud spending while delivering a higher level of service to analysts, BI users and data science teams.

AWS users can easily install Dremio AWS Edition via the AWS Marketplace.

To learn more about Dremio and how it complements AWS data lakes, visit http://dremio.com/aws.

About Dremio Corporation

Dremio reimagines the cloud data lake to deliver faster time to analytics by eliminating the need to copy and move data to proprietary data warehouses, or create cubes, aggregation tables and BI extracts. A self-service semantic layer provides flexibility and control for data architects, and self-service for data consumers.

Founded in 2015, Dremio is headquartered in Santa Clara, CA. Investors include Cisco Investments, Lightspeed Venture Partners, Norwest Venture Partners and Redpoint Ventures. For more information, visit http://www.dremio.com.

Connect with Dremio on GitHub, LinkedIn, Twitter, and Facebook.