HOW SNOWFLAKE'S CLOUD ARCHITECTURE SCALES MODERN DATA ANALYTICS

snowflake®

WHITEPAPER

TABLE OF CONTENTS

DATA AND ANALYTICS SHIFT TO PUBLIC CLOUDS

Data and analytics workloads continue to move in large numbers from private on-premises data centers to public cloud environments. Although 75% of data and analytics solutions are deployed on premises today, the latest figures from IDC predict that the preferred deployment model for Big Data will shift dramatically in the next few years.[1]

By 2023, IDC predicts that 50% of revenue for data and analytics will come from public clouds, representing a 29.7% compound annual growth rate (CAGR). This means public cloud deployments are projected to grow *eight times faster* than on-premises deployments (3.7% CAGR).

Gartner projects an even more aggressive switch to the cloud than IDC, predicting 75% of all databases will be deployed or migrated to a cloud platform by 2022. Further, Gartner predicts only 5% of these workloads will return to on-premises deployments.

What are the primary reasons for moving workloads to the cloud? While it's easy to assume that companies seek to eliminate capital expenditures such as hardware procurement costs, the top reason is to improve **speed, scale, and agility.**

Today's data warehouse managers want to:

- Boost analytics productivity for all users and workloads without resource contention
- Scale instantly through unlimited storage and compute capacity while easily matching business needs
- Onboard and support a diverse set of data without bottleneck delays to develop insights and produce answers from the data as expeditiously as possible.

These requirements are the inspiration behind Snowflake's **Multi-Cluster Shared Data (MCSD) architecture design**, which provides fast analytics performance with unmatched scale and agility across clouds. In combination with dramatically simplified data platform operations, Snowflake enables faster business results.

This white paper explains why the architecture you select for data warehousing and data lakes makes a difference in the agility, performance, and scalability of your data warehouse or data lake. By providing insights into the various architectures, we demonstrate how Snowflake's architecture differs and why Snowflake is the only cross-cloud data platform that is built for the cloud and that scales like the cloud, enabling organizations to be data-driven.

> "Though cost optimization is important, it's not the main reason for moving to cloud services and applications. The fact that we're going to cloud is more about agility, getting the feature functionality you need at the speed you need it. Digital business runs at a much faster speed than [bricks and mortar] business; hyperscale data centers are the only things that can support the speed of digital business, the cooperation required in digital business—it's very difficult to do that on-premises."
>
> **GARTNER**
> CIO Magazine, January 2019

1 Worldwide Big Data and Analytics Software Forecast, 2019–2023. IDC. September 2019.

WHY ARCHITECTURE MATTERS

To appreciate the benefits of a cloud-built data platform, it helps to understand the basics of legacy data warehouse architectures. Regardless of where implemented, on-premises or in the cloud, legacy data warehouse architectures contain physical limitations and complexities inherent to their design that prevent high levels of scalability and agility.

Legacy on-premises architectures

For decades, on-premises data warehouse solutions have been built based on one of two architecture types: **shared-disk** or **shared-nothing architectures.** In particular, on-premises data warehouse appliances and cloud-based data warehouse services frequently use shared-nothing architectures as the basis for their platforms.

Shared-disk architecture

A shared-disk system is a traditional architecture where all compute resources share a common storage structure. Shared-disk databases keep data management straightforward and simple.

System setup:

Shared-disk architectures (Figure 1) store all database objects on a storage structure or device that is accessible from a single database server or, in the case of a cluster topology, from all of the nodes of the database cluster. All changes to the database are written to the shared disk to ensure that the database server or all cluster nodes see a consistent version of the data.

Examples:

- Classic online transaction processing
- Traditional, non-MPP, data warehouses
- Computational clusters

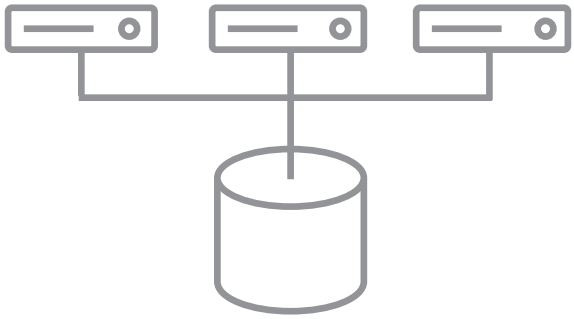

FIGURE 1. SHARED-DISK ARCHITECTURE (CLUSTER TYPE SHOWN)

Challenges:

- Physical factors limit the scalability of this architecture. As the size of the database server or the number of processing nodes increases, the storage device and related storage network become bottlenecks, forcing data processing to slow down to wait for the shared disk to return data.
- When either compute or storage capacity outgrows the limits of a physical storage array, the system must be taken down and swapped for a higher capacity system.
- The database server or cluster operates as a single database object. A database administrator must partition the storage environment to support more than one database/schema or workload.

Shared-nothing architecture

A shared-nothing system is a distributed server, or cluster environment where each node of the cluster contains its own compute and storage resources. This approach localizes and, in theory, speeds up data processing because both the compute resources and the data needed to complete the computations are located within the same node, thereby decreasing data access bottlenecks and processing delays. When the necessary data is not located with the compute resources, the compute resource must spend time fetching the data from other nodes, which creates the bottlenecks and processing delays.

System setup:

This architecture distributes data across all processing nodes in the cluster (Figure 2), with each node holding a subset of the data. This design eliminates the data processing and disk-access bottlenecks associated with a shared-disk solution that often occurs when more than one processing thread or compute node must access the same shared data. In theory, with shared-nothing solutions, the more compute nodes, the more data processing speeds up as multiple nodes operate in parallel to tackle analytic computations.

This type of architecture is also known as a distributed architecture or a massively parallel processing (MPP) architecture. True MPP architectures are constructed of hundreds or thousands of nodes.

Examples:

- Hadoop Distributed File System (HDFS) based solutions
- On-premises MPP appliances from companies such IBM (incl. Netezza), Teradata, Micro Focus, and others
- Cloud data warehouse platform offerings from Amazon, Microsoft, and others

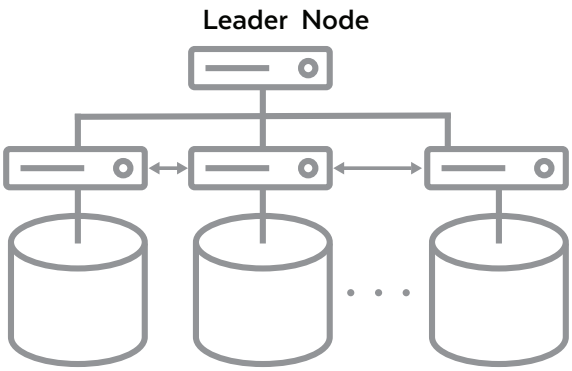

FIGURE 2. SHARED-NOTHING ARCHITECTURE

Challenges:

- Shared-nothing architectures require careful sizing of the hardware infrastructure to deliver the ideal balance of CPU processing, memory, I/O bandwidth, and storage for the expected workloads and needed data storage capacity. These upfront decisions impact long-term performance and scalability, which are hard to forecast with precision. Because of the disruption involved to change and scale configurations, data warehouse managers presize configurations to meet the expected peak demand. This leads to higher upfront costs and larger configurations with under-utilization when data processing is idle or not at peak-demand levels.
- Shared-nothing architectures require distribution of data across processing nodes using manual or dynamic partitioning assignments based on the data needed to process queries. For manual partitioning, a database administrator must execute the partitioning process for both initial deployment and again as the system scales, which is time-consuming and difficult to get right. For dynamic partitioning, often, the algorithm determining the data partitioning is not efficient and leads to an explosion of unnecessary partitions.
- Because compute resources are tied to storage resources, as organizations add nodes to increase storage capacity, they must purchase additional servers. Conversely, when it is necessary to increase computational power, data must be redistributed across all nodes—the existing nodes and the new nodes—resulting in disruption to the environment and increased management complexity.
- A user-specified distribution key (or a default dynamic process) distributes data when it is loaded. Changing a distribution key requires complete redistribution of data across the cluster. This operation is slow and disruptive.
- Particularly for Hadoop-based shared-nothing clusters, the data (especially if in raw form) distributed among the nodes may not be in a queryable format. This inhibits querying the data for analytics. A data engineer or data scientist must transform the data into a format that allows more efficient querying of the data.

Other cloud architectures

Recently, new architectures have appeared on the data warehousing scene. Two such architectures are cloud **serverless query engines** and **cloud data repository platforms.**

Serverless query engines

Serverless query engines operate as a "zero-operations" service, usually as a complement alongside a separate data storage infrastructure. The host cloud platform (for example, AWS or Google) orchestrates and manages all infrastructure for the query engine. Most query engines, however, were originally designed as scan and query engines and not as complete relational database management systems or data warehouse with controllable resources. For example, if the source data loaded into the query engine changes, users may have to manually reload or refresh data into the query engine.

System setup:

Query engines operate alongside data storage or cloud blob storage (data lake) resources (Figure 3). The user manually loads data into the query engine service. The query engine returns the results.

Examples:

- AWS Athena
- Google BigQuery
- Spark-based engines
- Presto-based engines

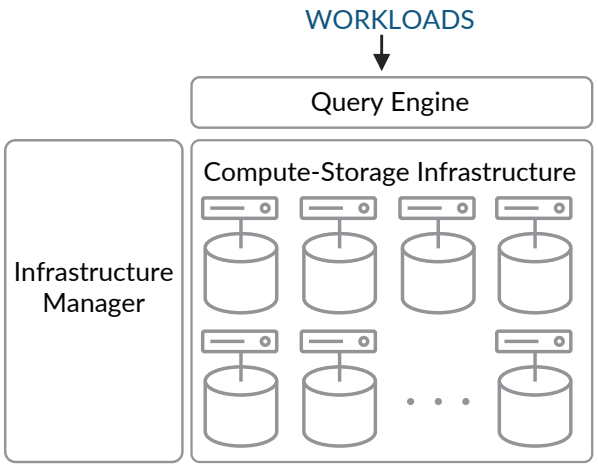

FIGURE 3. QUERY ENGINE ARCHITECTURE

Challenges:

- Depending on the complexity of the query and the data modeling required, the performance for serverless query engines varies widely. In some cases, data modeling is difficult.
- Query engines, generally, are not complete relational database management systems, but capabilities are evolving.
- Users have no control over the resources used to run queries.
- Because resources are predetermined, disruptive scaling or query stalling occurs once compute limits are reached.

Hub-and-spoke data platforms

Most data lake architectures are a hub-and-spoke style data platform design, whereby the storage infrastructure serves as the central point for landing and storing data (Figure 4). A query engine of some sort is then required to execute queries against data in the repository. Further, for the query engine to operate effectively, the data will have to be transformed from a raw data file to a queryable structured table format or a NoSQL/document database engine must be considered for text type data formats such as JSON.

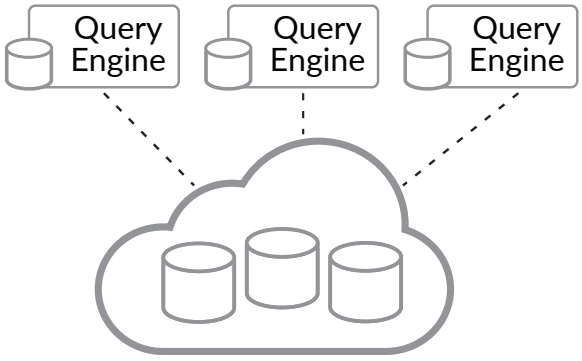

FIGURE 4. CLOUD HUB-AND-SPOKE ARCHITECTURE

System setup:

Hub-and-spoke data platforms include a storage infrastructure (such as HDFS, Amazon S3, or Azure Data Lake Storage) and a companion query tool.

Examples:

- Amazon S3 cloud data storage with AWS Athena serverless query engines
- Google Cloud Datastore with Google Big Query as the query engine

Challenges:

- Raw data in cloud storage must be transformed for querying, or a non-relational querying engine must be considered.
- Data repository platforms may be effective with small, infrequent querying, or single workloads, however, performance decreases as the number of workloads and query complexity increase.

Common architectural shortcomings

Physical scaling limitations are among the biggest challenges for data platforms implemented on premises. For example, there's limited physical rack space to add new server nodes. In turn, server nodes have only a limited number of drive bays available to increase disk storage. Adding new physical capacity is costly and disruptive. As a result, platform managers spend time deleting files and removing analytics data to make room for new data.

In addition, these architectures generally do not natively provide service availability protection, data backup protection and retention, or security. For very large environments, the added cost and effort to protect against service outages discourages platform managers from properly backing up an environment, leaving the data warehouse platform exposed to service outages and creating risk.

Because files generally are stored in their native format, on-premises environments require a significant amount of effort to query and deliver insights from semi-structured data. Data-driven companies require semi-structured and structured data to be joined easily so organizations can achieve more complete relational insights from analytics, such as 360-degree views of customers and business lines.

Simply moving a data platform to the cloud does not solve these challenges. If a data warehouse deployed in the cloud is the same type of platform implemented on premises, the cloud version has the same physical scaling limitations as the on-premises version. Scaling remains limited and disruptive, inhibiting organizations from realizing the agility objectives desired in the cloud.

In the end, other architectural approaches may separate compute and storage and offer scaling capabilities, but only as a bolted-on afterthought. As a result, capacity and automation are limited compared to Snowflake.

SNOWFLAKE—ENGINEERED DIFFERENTLY

Snowflake was founded by a team of experienced database veterans with expertise in relational databases, data platforms, and vector computing. Snowflake founders recognized the challenges of legacy architectures and realized that creating a better solution required rethinking how data platforms and database architectures were built. The founders knew starting with an existing software code base such as Postgres, Hadoop, or any of the popular relational databases at the time, would never work. None of these options were native to the cloud.

That's why the Snowflake team decided to write every line of code anew in the core data platform engine. By building a completely new data platform and relational database management system from the ground up, they could deliver a dynamic infrastructure with instant, disruption-free, scalability and performance-as-a-service at any cloud scale, all at a fraction of the cost of traditional systems.

Snowflake's **Multi-Cluster Shared Data architecture** was born from these ambitions.

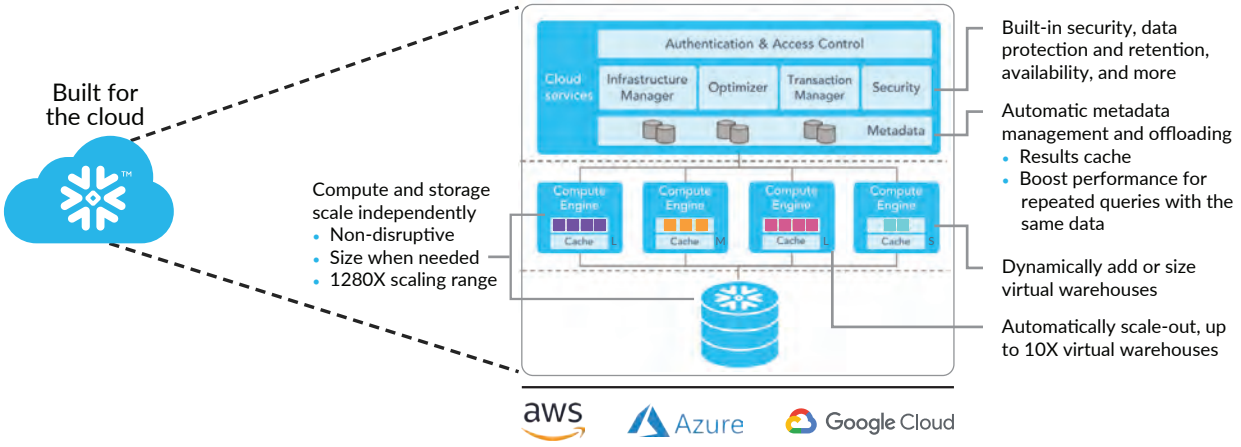

FIGURE 5. SNOWFLAKE MULTI-CLUSTER SHARED DATA ARCHITECTURE

The three-layer design

Snowflake is composed of a three-layer design with separate storage, compute, and cloud services layers. This architecture excels because, while compute and storage resources are physically separate, they are logically part of a single, tightly-integrated data platform system that provides nondisruptive scaling.

Storage

The storage layer performs the following steps when data is loaded:

- Splits the data into micro-partitions, hundreds of thousands of partitions per data file
- Extracts metadata (such as timestamp and min/max values) to enable efficient query processing
- Compresses the micro-partitions to save storage and space costs
- Fully encrypts the data using a secure key hierarchy

Compute

The compute layer is the Snowflake muscle. Compute engines, designed to process large quantities of data quickly and efficiently, perform all data processing. Compute engines do the following when performing a query:

- Retrieve the minimum data required from the storage layer to satisfy queries, as dictated by Snowflake's data pruning algorithm.
- Cache the data and query results locally, delivering significantly faster performance and lowering costs (query results from cache incur no compute charges).
- Unique to Snowflake, multiple compute engines can simultaneously operate on the same data with full system-wide transactional integrity and complete ACID compliance. Across any number of isolated workloads, read operations (SELECT) always see a consistent view of data, and write operations never block readers.

Services

The services layer is the brains of Snowflake. It is constructed of stateless compute resources running across multiple availability zones. This layer provides a highly available and distributed metadata store for global state management and enables Snowflake advanced services such as data pruning, data exchange, and cross-cloud data replication. The services layer also provides security and encryption key management and enables all SQL, DML, and DDL functions.

The services layer is responsible for:

- Authenticating user sessions
- Providing management
- Enforcing security functions
- Performing query compilation and optimization
- Coordinating all transactions

To execute data pruning, for example, the services layer compiles the queries and metadata used to determine which micro-partition columns need to be scanned to complete queries quickly. This ensures only the data needed to complete the query is scanned, resulting in highly performant results.

Because automatic metadata processing is powered by a separate integrated subsystem, the services layer does not need user compute resources to perform statistics gathering or any other metadata operation, ensuring that all user resources are employed on user, not system, data processing.

What a cloud-built architecture means in practice

Whether supporting a few users or thousands of users, Snowflake's cloud-built data warehouse architecture delivers automation and eliminates the physical limitations and the manual work typically experienced with on-premises and other cloud solutions.

With the MCSD architecture, the compute and storage resources layers are scaled independently. This separation enables compute resources to scale without disrupting queries and without redistributing data. And the same is true in reverse—storage resources scale to virtually any capacity without the cost of adding unnecessary compute resources.

Additionally, Snowflake's virtual warehouses, which can be thought of as compute engines, are configurable to automatically scale out up to 10 clusters of the identical virtual warehouse size. When enabled, compute resources automatically suspend when idle or inactive (which also suspends all compute charges) and automatically resume with a new query operation.

Additional benefits of Snowflake's cloud-built data platform include:

- **Semi-structured data:** With one platform, Snowflake provides native support for both structured and semi-structured data such as CSV files, tables, JSON, XML, and Avro, as well as popular data stores such as ORC and Parquet. Upon data loading, Snowflake automatically parses semi-structured data and creates a patented VARIANT data type that can be immediately queried with industry standard ANSI SQL. Query operations with the VARIANT data type enjoy the same performance optimizations as all other queries, ensuring highly-performant query execution of semi-structured data.
- **Data exchange:** Snowflake's advanced and integrated metadata capabilities enable data warehouse operations to provide live, secure data sharing and exchange. This enables collaboration with internal business partners in separate divisions with their own Snowflake accounts. In addition, live data exchange can be offered to outside business partners without the data copying and moving struggles.

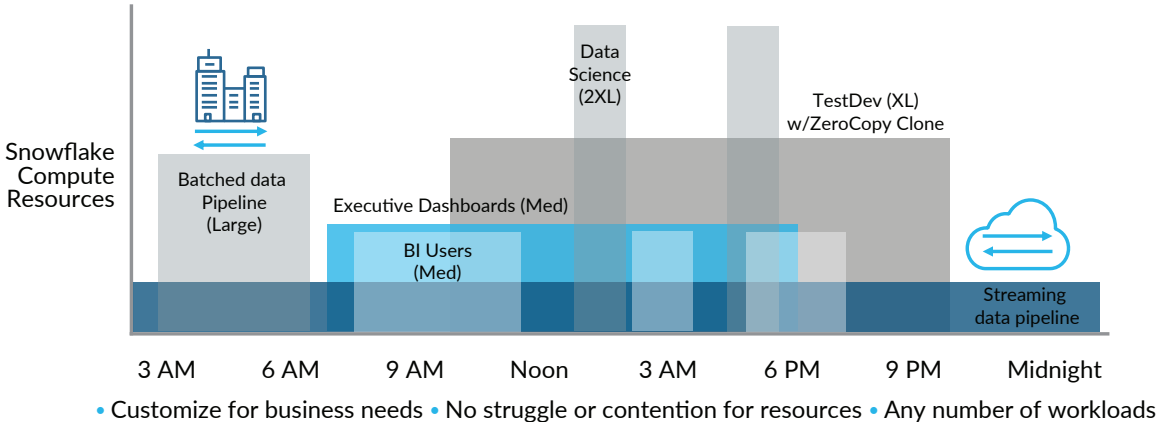

FIGURE 6. ISOLATE AND OPERATE SEPARATE WORKLOADS, CONCURRENTLY

MCSD Cloud Benefits for Data-Driven Organizations

Criteria	Snowflake	Shared Data	Shared Nothing	Hub and Spoke	Cloud Query Engines	Why Important?
Low-latency, high-performance across simple and complex queries	✓	Yes	Yes	No	Limited	Delivers faster business results
Easily join structured and semi-structured data for analytics	✓	Limited	Limited	More complex	Limited	Enables 360° views, multi-channel data analytics
Unlimited query concurrency within a single, integrated platform	✓	Limited	Limited	No	Limited	Eliminates data duplication costs and complexity
Unlimited connections within a single, integrated platform	✓	Limited	Limited	No	Limited	Supports as many accesses to platform as needed, while avoiding data duplication
Multi-statement ACID transactions	✓	Yes	Yes	No	No	Assures pipeline and transactional integrity for single and multi-table inserts
Workload isolation, within the same integrated platform and single source of truth	✓	No	No	No	No	Matches resources to business needs with maximum simplicity
Instantly scale up, down, out or in without disruption	✓	No	Limited	Very Limited	Limited	Delivers fast execution and ensures environment stays up and running when scaling
Simple management, e.g., no indexing, no manual partitioning	✓	No	No	No	Yes	Enables data teams to attain higher strategic focus
ZeroCopy Cloning	✓	No	No	No	No	Simplifies test-dev with full production data, while eliminating storage duplication
Cross-cloud operation	✓	No	No	No	No	Provides strategic and operational flexibility across clouds

- **Automatic data protection and retention:** Integrated cloud services enable transparent provisioning of resources and built-in resilience (for example, up to 90 days data backup and retention and automatic node failure protection). This design spares data platform teams from manually building clusters or extra processes to protect services.

Finally, it's easy to customize Snowflake resources to the specific needs of organizations and businesses.

THE SNOWFLAKE DIFFERENCE—A CROSS-CLOUD DATA PLATFORM

Snowflake's unique cloud-built design and revolutionary architecture provides a single, logically integrated data platform that ties together data storage, analytics operations, and platform services (such as security and data protection and retention) for data warehousing, data lakes, data exchange and other use cases. Organizations of all sizes can now achieve:

- High-performance industry standard ANSI SQL relational database management experiences with structured and semi-structured data
- Support for any number of databases and isolated workloads, along with high levels of query and workload concurrency
- Consolidation of data into a single authority source, upon which all workloads (with the right permissions) can operate with complete read and write ACID integrity

Snowflake's MCSD enables data platform managers to support and easily scale more users, more data queries, and isolated workloads. All compute and storage resources scale instantly and independently, dramatically simplifying data platform management and execution.

With Snowflake, organizations can simplify data architectures, drive down costs, provide easier access to data, and customize computing resources. This results in more productive teams and overall faster time to value from data, which in turn yields faster business results.

Now organizations have a data platform that handles their toughest data struggles today and into the future.

FIND OUT MORE

To learn more about Snowflake customer stories and how a platform built for the cloud takes enterprise data warehousing and analytics operations beyond physical limits and beyond legacy data warehousing approaches, visit www.snowflake.com.

ABOUT SNOWFLAKE

Snowflake is the only data warehouse built for the cloud, enabling the data-driven enterprise with instant elasticity, secure data sharing and per-second pricing, across multiple clouds. Snowflake combines the power of data warehousing, the flexibility of big data platforms and the elasticity of the cloud at a fraction of the cost of traditional solutions. Snowflake: Your data, no limits. Find out more at **snowflake.com**