Table of Contents

Snowflake Pros

Multi-Cluster, Shared Data architecture

Snowflake's multi-cluster, shared data architecture is designed to process enormous quantities of data with maximum speed and efficiency. All data processing horsepower within Snowflake is performed by one or more clusters of compute resources. When performing a query, these clusters retrieve the minimum data required from the storage layer to satisfy queries. As data is retrieved, it's cached locally with computing resources, along with the caching of query results, to improve the performance of future queries.

In addition, and unique to Snowflake, multiple compute clusters can simultaneously operate on the same data while fully enforcing global, system-wide transactional integrity with full ACID compliance. Operations always see a consistent view of the data, and write operations never block readers. Transactional integrity across compute clusters is achieved by maintaining all transaction states within the metadata services layer.

https://www.snowflake.com/product/architecture/

Data Platform as a True Cloud Service

Snowflake's data warehouse is a true SaaS offering.

More specifically:

- There is no hardware (virtual or physical) for you to select, install, configure, or manage.
- There is no software for you to install, configure, or manage.
- Ongoing maintenance, management, and tuning is handled by Snowflake.

Snowflake runs completely on cloud infrastructure. All components of Snowflake's service (other than an optional command line client), run in a public cloud infrastructure.

Snowflake uses virtual compute instances for its compute needs and a storage service for persistent storage of data. Snowflake cannot be run on private cloud infrastructures (on-premises or hosted).

Snowflake is not a packaged software offering that can be installed by a user. Snowflake manages all aspects of software installation and updates.

https://docs.snowflake.net/manuals/user-guide/intro-key-concepts.html#data-warehouse-as-a-cloud-service

Storage Layer: Micro Partitions

All data in Snowflake tables is automatically divided into micro-partitions, which are contiguous units of storage. Each micro-partition contains between 50 MB and 500 MB of uncompressed data (note that the actual size in Snowflake is smaller because data is always stored compressed). Groups of rows in tables are mapped into individual micro-partitions, organized in a columnar fashion.

This size and structure allows for extremely granular pruning of very large tables, which can be comprised of millions, or even hundreds of millions, of micro-partitions.

Snowflake stores metadata about all rows stored in a micro-partition, including:

- The range of values for each of the columns in the micro-partition.
- The number of distinct values.
- Additional properties used for both optimization and efficient query processing.

Micro-partitioning is automatically performed on all Snowflake tables. Tables are transparently partitioned using the ordering of the data as it is inserted/loaded.

The benefits of Snowflake's approach to partitioning table data include:

- In contrast to traditional static partitioning, Snowflake micro-partitions are derived automatically; they don't need to be explicitly defined up-front or maintained by users.
- As the name suggests, micro-partitions are small in size (50 to 500 MB, before compression), which enables extremely efficient DML and fine-grained pruning for faster queries.
- Micro-partitions can overlap in their range of values, which, combined with their uniformly small size, helps prevent skew.

- Columns are stored independently within micro-partitions, often referred to as *columnar storage*. This enables efficient scanning of individual columns; only the columns referenced by a query are scanned.
- Columns are also compressed individually within micro-partitions. Snowflake automatically determines the most efficient compression algorithm for the columns in each micro-partition.

https://docs.snowflake.net/manuals/user-guide/tables-clustering-micropartitions.html#what-are-micro-partitions

Compute Layer: Virtual Warehouses

A virtual warehouse, often referred to simply as a "warehouse", is a cluster of compute resources in Snowflake. A warehouse provides the required resources, such as CPU, memory, and temporary storage, to perform the following operations in a Snowflake session:

Executing SQL SELECT statements that require compute resources (e.g. retrieving rows from tables and views).

Performing DML operations, such as:

- Updating rows in tables
- Loading data into tables
- Unloading data from tables

You can create any number of warehouses within an account to meet the various needs of the organization.

https://www.snowflake.com/blog/support-one-to-one-hundred-data-warehouse-workloads-with-snowflake/

Snowflake outperforms traditional methods for executing data workloads. Compute resources scale linearly in Snowflake, while efficient query optimization delivers answers in a fraction of the time of legacy cloud or on-premises systems. Performance challenges can be addressed in seconds. You can specify the size of a compute cluster based on the performance you initially require. But you can resize at any time and even while a workload is running.

Unlimited Concurrency

True elasticity. Snowflake allows customers to create virtual warehouses (compute engines) which can run multiple workloads against the same data without a contention for resources.

https://docs.snowflake.net/manuals/user-guide/warehouses-overview.html#overview-of-warehouses

Automatic Scaling. With the introduction of multi-cluster DW feature, Snowflake allows customers to dynamically or automatically scale and load-balance increasing/decreasing number of concurrent users (queries).

Multi-cluster warehouses deliver a consistent SLA to an unlimited number of concurrent users. Automatic clustering eliminates manual re-clustering of data when loading new data into a table. With materialized views, users experience improved query performance of workloads composed of common, repeated query patterns. As concurrent workloads increase, Snowflake automatically adds to compute clusters and distributes queries across them, removing the hassle of manually re-clustering data. Clusters pause when the workload decreases. Charges only accrue for active clusters, so you only pay for what you use and by the second. Plus, you can pause compute clusters at any time.

https://docs.snowflake.net/manuals/user-guide/warehouses-multicluster.html#multi-cluster-warehouses

Support for semi-structured data

Data can come in multiple forms from numerous sources, including an ever-expanding amount of machine-generated data from applications, sensors, mobile devices, etc. To support these new types of data, semi-structured data formats, such as JSON, Avro, ORC, Parquet, and XML, with their support for flexible schemas, have become popular standards for transporting and storing data.

Snowflake provides native support for semi-structured data, including:

- Flexible-schema data types for loading semi-structured data without transformation.
- Automatic conversion of data to optimized internal storage format.
- Database optimization for fast and efficient SQL querying.

https://docs.snowflake.net/manuals/user-guide/semistructured-concepts.html#semi-structured-data
https://yuyxg2s4nyr1vz7vr2pfg77x-wpengine.netdna-ssl.com/wp-content/uploads/2015/06/Snowflake_Semistructured_Data_WP_1_0_062015.pdf

Simplicity and Ease-of-use

Snowflake is easy to manage, no need to think about data distribution, statistics, indices, encryption, metadata operations, replication (for disaster recovery) and more.

Snowflake eliminates the administration and management demands of traditional data platforms. Snowflake is a true data platform-as-a-service running in the cloud. With built-in performance, there's no infrastructure to manage or knobs to turn. Snowflake automatically handles infrastructure, optimization, availability, data protection, and more, so you can focus on using your data, not managing it.

Pricing Model

Per-second, usage-based pricing for compute and storage means you only pay for the amount of data you store and the amount of compute processing you use. This translates to no large upfront costs, over-provisioned systems or idle clusters consuming money.

https://docs.snowflake.net/manuals/user-guide/credits.html#understanding-snowflake-virtual-warehouse-storage-and-cloud-services-usage

Support for modern programmability

Snowflake supports multiple ways of connecting to the service:

- A web-based user interface from which all aspects of managing and using Snowflake can be accessed.
- Command line clients (e.g. SnowSQL) which can also access all aspects of managing and using Snowflake.
- ODBC and JDBC drivers that can be used by other applications (e.g. Tableau) to connect to Snowflake.
- Native connectors (e.g. Python) that can be used to develop applications for connecting to Snowflake.
- Third-party connectors that can be used to connect applications such as ETL tools (e.g. Informatica) and BI tools to Snowflake.

https://docs.snowflake.net/manuals/user-guide/intro-supported-features.html#connectivity

Security

Snowflake is secure by design. All data is encrypted in motion, over the Internet or direct links, and at rest on disks. Snowflake supports two-factor and federation authentication with single sign-on. Authorization is role-based. You can enable policies to limit access to predefined client addresses.

Snowflake is SOC 2 Type 2 certified on both AWS and Azure and support for PHI data for HIPAA customers is available with a Business Associate Agreement. Additional levels of security, such as encryption across all network communications and virtual private or dedicated isolation, are also available.

Disaster Recovery: Cross Cloud / Cross Region Database Replication

This feature enables replicating databases between Snowflake accounts (within the same organization) and keeping the database objects and stored data synchronized. Database replication is supported across regions and across cloud platforms.

https://docs.snowflake.net/manuals/user-guide/database-replication-intro.html#introduction-to-database-replication-across-multiple-accounts

Business Continuity: In-Region High Availability

Achieve high availability with Snowflake's scale-out architecture, which is fully distributed across multiple Amazon, Azure and Google availability zones. Snowflake can continue operations and withstand the loss of availability due to hardware failure. The system is designed to tolerate failures with minimal impact to our customers.

The below link talks about AWS but the same concept is applicable to Azure and GCP as well.

https://community.snowflake.com/s/article/snowflake-delivers-high-availability-by-using-aws-availability-zones

Data Sharing

Snowflake's Secure Data Sharing enables you to share the data within your account with other Snowflake users but without having to copy or transfer data from the data provider's account to the data consumer's account.

Instead, you grant secure and curated access to read-only copies of your data. Accounts that receive shared data only pay for the compute resources they use to consume the data. Data shared from a Snowflake data provider can easily be combined with data from the Snowflake data consumer's account without laborious effort or third-party tools. Avoid the burdens and complexities of decades-old email, FTP and EDI technologies with Snowflake. Simply decide what you want to share with your data consumers, and share the data through easy-to-use SQL functions.

https://www.snowflake.com/use-cases/modern-data-sharing/
https://docs.snowflake.net/manuals/user-guide-data-share.html#sharing-data-securely-in-snowflake

Robust Partner Ecosystem

Snowflake works with a wide array of industry-leading tools and technologies, enabling you to access Snowflake through an extensive network of connectors, drivers, programming languages, and utilities, including:

- Certified partners who have developed cloud-based and on-premises solutions for connecting to Snowflake.
- Other 3rd-party tools and technologies that are known to work with Snowflake.
- Snowflake-provided client interfaces such as SnowSQL (CLI), Python, and Node.js, Kafka, Spark, Go, .NET, JDBC, and ODBC (drivers).

The below visual shows a list of some, but not nearly all, 3rd party tools known to integrate with Snowflake

https://docs.snowflake.net/manuals/user-guide/ecosystem-all.html

Great Documentation

Snowflake has easy the best documentation of any technology product currently on the product.

This includes feature documentation, tutorials, hands on labs, videos and more:

https://docs.snowflake.net/manuals/index.html
https://docs.snowflake.net/manuals/user-guide/intro-key-concepts.html
https://docs.snowflake.net/manuals/other-resources.html#useful-topics
https://docs.snowflake.net/manuals/other-resources.html#hands-on-lab
https://docs.snowflake.net/manuals/other-resources.html#getting-started-videos
https://docs.snowflake.net/manuals/other-resources.html#feature-demonstration-videos

Exists as a Generally Available SaaS Today

All of Snowflake's features advertised or documented exist in the product today. When you sign up for a Snowflake account you immediately have access to Data Platform capable of handling Petabytes of data and millions of queries a day.

https://docs.snowflake.net/manuals/user-guide-intro.html#introduction-to-snowflake