snowflake

Snowflake Overview

Snowflake is a **cloud-based** data warehouse created in 2012 by three data warehousing experts who were formerly at Oracle Corporation. Snowflake architecture differs from traditional data warehousing technologies like Oracle, Teradata, SQL Server, and cloud data warehouses like Amazon Redshift and Google BigQuery.

Snowflake is a pay as you go service. It charged users on the **per-second basis** of **computation** or **query processing**. There exist some data warehouses that use either **shared-disk** or **shared-nothing** architecture, but Snowflake uses **hybrid architecture** comprises of both shared-disk or shared-nothing architecture. Snowflake uses a central storage layer for persisting data that is accessible to all compute nodes. Snowflake also processes queries using **Massively Parallel Processing (MPP)** clusters where each node stores a portion of the entire data locally.

Snowflake is a true self-managed service, meaning:

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

Snowflake Architecture

Before we dive into the Snowflake architecture, we first understand shared-disk and shared-nothing architectures.

- **Shared disk Architecture**
  - All computational nodes in this system share the same disk or storage device. Although each processing node (processor) has its memory, all processors have access to all disks. Since all nodes can access the same data, cluster control software is needed to monitor and control data processing. All nodes have a uniform copy of the data when updated or deleted. Two (or more) nodes trying to edit the same data simultaneously must not be allowed.

  - A shared disk architecture is typically well-suited for large-scale computing that requires ACID compliance. Applications and services that require only limited shared access to data and applications or workloads that are hard to split are usually suitable for a shared disk. Oracle Real Database Clusters is one such example of shared disk architecture.

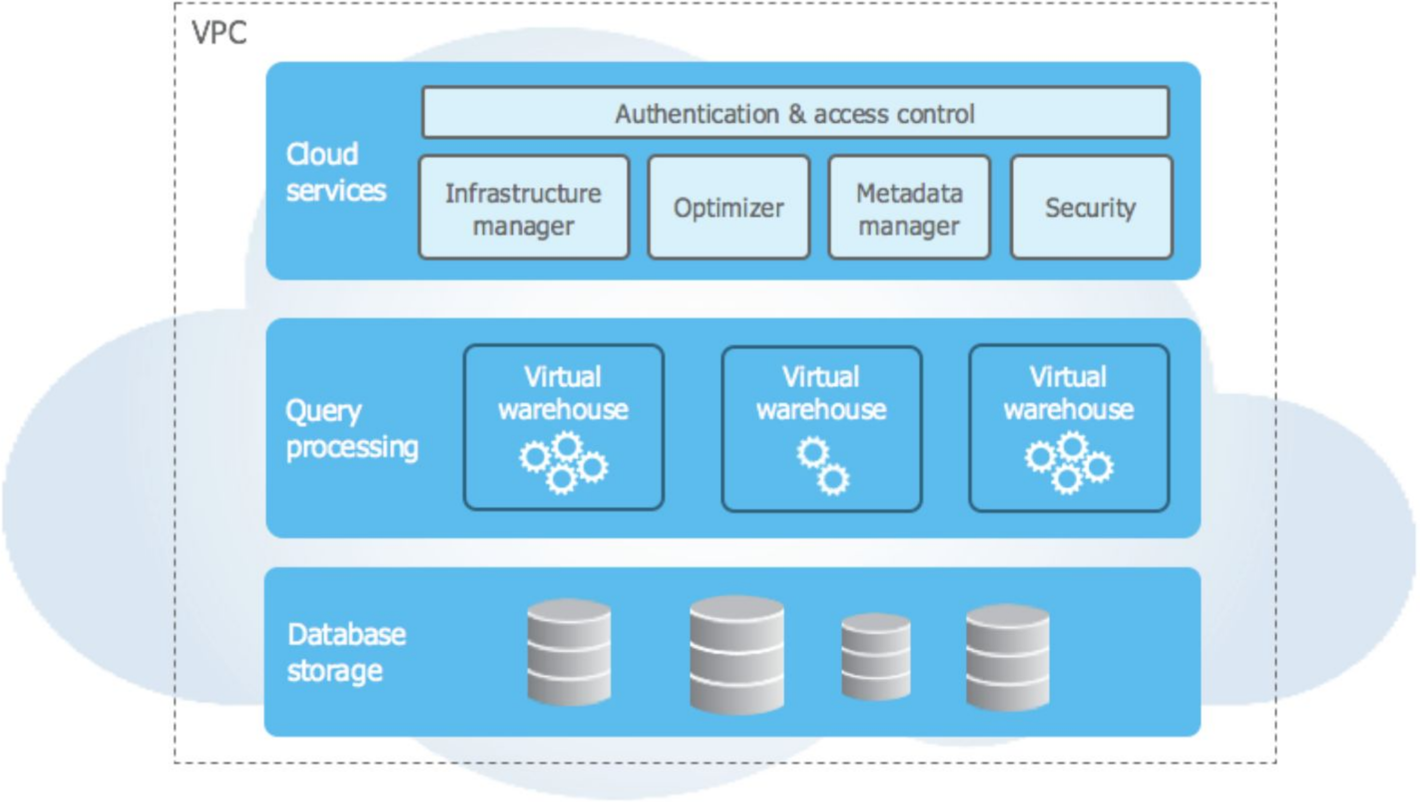
- **Shared-nothing Architecture**
  - Each computing node has its private memory and storage or disk space in a shared-nothing architecture. These nodes can communicate with each other due to interconnected network connections. When a request for processing comes in, a router routes it to the proper computing node for fulfillment. Specific business rules are often used at this routing layer to route traffic efficiently to each cluster node. When one of the computer nodes in a shared-nothing cluster fails, it transfers the processing rights to another cluster node.

  - Due to this change of ownership, there will be no interruption in processing the user requests. A shared-nothing architecture provides the application with high accessibility and scalability. One of the first web-scale technology organizations to deploy shared-nothing architectures, Google runs geographically distributed shared-nothing clusters with thousands of computing units. This is why a shared-nothing architecture is the right approach for a complex analytical data processing system like a data warehouse.

Snowflake Architecture

Snowflake's unique architecture consists of three key layers:

- Database Storage
- Query Processing
- Cloud Services



Snowflake Architecture

Grow Data Skills

- **<u>Database Storage</u>**
  - When data is loaded into Snowflake, Snowflake reorganizes that data into its internal optimized, compressed, columnar format. Snowflake stores this optimized data in cloud storage.
  - Snowflake manages all aspects of how this data is stored — the organization, file size, structure, compression, metadata, statistics, and other aspects of data storage are handled by Snowflake. The data objects stored by Snowflake are **not directly visible nor accessible by customers**; they are only accessible through SQL query operations run using Snowflake.
- **<u>Query Processing</u>**
  - Query execution is performed in the processing layer. Snowflake processes queries using **"virtual warehouses"**. Each virtual warehouse is an **MPP compute cluster composed of multiple compute nodes** allocated by Snowflake from a cloud provider.
  - Each virtual warehouse is an independent compute cluster that does not share compute resources with other virtual warehouses. As a result, each virtual warehouse has no impact on the performance of other virtual warehouses.
  - **What is virtual warehouse? :** 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

Snowflake Architecture

Grow Data Skills

- **Cloud Services** The cloud services layer is a collection of services that coordinate activities across Snowflake. These services tie together all of the different components of Snowflake in order to process user requests, from login to query dispatch. The cloud services layer also runs on compute instances provisioned by Snowflake from the cloud provider.

    - Services managed in this layer include:
        - Authentication
        - Infrastructure management
        - Metadata management
        - Query parsing and optimization
        - Access control

Snowflake Data Storage

Snowflake, a cloud-based data warehousing platform, utilizes a unique architecture that separates its storage and compute layers. This design allows for both high efficiency and scalability.

Storage Layer in Snowflake:

- **Type of Storage:** Snowflake uses a distributed, columnar storage system for data management. This storage system is built on top of cloud storage solutions provided by major cloud service providers like Amazon S3, Google Cloud Storage, and Microsoft Azure Blob Storage.

- **Columnar Storage:** In Snowflake, data is stored in a columnar format, which means that each column in a database table is stored separately. This approach optimizes data compression and enhances the performance of read-intensive database operations, particularly for analytical queries.

- **Cloud Storage Utilization:**
  - By leveraging the cloud providers' robust and scalable storage solutions, Snowflake can efficiently handle vast amounts of data without the need for users to manage physical hardware.
  - Snowflake abstracts the underlying cloud storage infrastructure from the end user, providing a seamless experience regardless of the scale of data.

- **Data Storage and Organization:**

  - Snowflake organizes data into micro-partitions that are internally optimized and compressed.
  - Data is automatically split into these micro-partitions as it's loaded into Snowflake.
  - The system also maintains metadata about the stored data, allowing it to efficiently execute queries.

Snowflake Data Storage

Grow Data Skills

- **<u>Automatic Scaling:</u>** The storage layer in Snowflake can scale up or down automatically based on the volume of data. Users don't need to manually adjust storage; it's handled dynamically by Snowflake.

- **<u>Durability and Reliability:</u>** Data in Snowflake is highly durable and resilient. Cloud storage services used by Snowflake typically replicate data across multiple geographical locations, ensuring high availability and data protection.

- **<u>Cost-Effectiveness:</u>** Snowflake's storage layer benefits from the cost efficiency of cloud storage providers, which often charge based on the amount of storage used. This means users pay for what they use without the need for significant upfront investments in storage infrastructure.

How Snowflake Stores Data?

Grow Data Skills

- **Data Loading:**
  - When data is loaded into Snowflake, it's converted into its optimized, columnar format.
  - Snowflake handles various data formats (like JSON, Avro, Parquet, XML, etc.) and can import data directly from different sources.

- **Data Clustering:**
  - Snowflake organizes data using clustering keys (if defined) to optimize query performance. However, Snowflake's architecture also allows for efficient querying without the need for manual indexing or tuning.

- **Transactions and Metadata:**
  - The system keeps track of all changes through its advanced metadata management, enabling features like time travel, data cloning, and consistent, transactional access to data.

Massive Parallel Processing in Snowflake

Grow Data Skills

- **Data Partitioning:**
  - When data is stored in Snowflake, it is automatically partitioned into micro-partitions that are both columnar and compressed. These micro-partitions are typically a few MB in size.

- **Query Execution:**
  - When a query is executed, Snowflake's optimizer first determines which micro-partitions need to be scanned. This is an efficient process due to the metadata that Snowflake maintains about the data.
  - The query is then broken down into smaller tasks. These tasks are distributed across the nodes within a virtual warehouse.

- **Parallel Processing:**
  - Each node in a virtual warehouse processes its assigned tasks in parallel. The nodes work on different chunks of data simultaneously, significantly speeding up the processing time.
  - Snowflake's architecture allows it to scale horizontally by adding more nodes to a warehouse or even using larger warehouses, thereby increasing the parallel processing capabilities.

- **Result Aggregation:**
  - After all nodes complete their assigned tasks, the results are aggregated.
  - The final result set is then compiled and returned to the user.

- **Dynamic Resource Allocation:**
  - Snowflake can dynamically allocate or deallocate resources as needed. This means that for smaller queries, fewer resources can be used, and for larger, more complex queries, Snowflake can scale up resources to maintain performance.

- **Cache Utilization:**
  - Snowflake caches both the results of queries and frequently accessed data. If a query is repeated or if another query requires a portion of data that has already been processed, Snowflake can deliver results quickly by accessing its cache instead of reprocessing data.