**Snowflake**

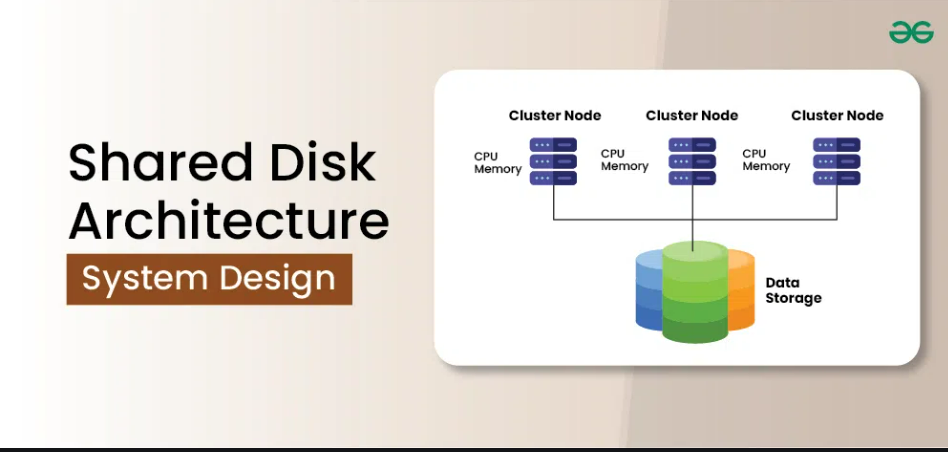
**Pricing in snowflake.**

1. Storage cost: Storage cost is measured as the average amount of data stored in Snowflake monthly.
2. Compute costs are incurred based on the size and duration of virtual warehouses you use for executing queries, data transformations, or running workloads.
3. Cloud services manage end-to-end solution of the user's task. It automatically assigns resources based on requirements of a task. Snowflake provides free usage of cloud service up to 10% of daily compute credits

**Architectures:**

1. **Shared disk architecture:**

distributed computing architecture in which the nodes share same disk devices, but each node has its own private memory and CPU.



**Advantages**:

1. **Ease of Management:**Centralization of storage
2. **Consistency**

**Disadvantages:**

1. **Performance Bottlenecks:** Nodes must wait, because of synchronization
2. **Scalability is limited**

1. **Shared nothing architecture:**

In this each node has its own mass storage as well as main memory. The [processor](https://www.tutorialspoint.com/embedded_systems/es_processors.htm) at one node may communicate with another processor at another node by a high speed interconnection network.

A diagram of a computer server

Description automatically generated

**Advantages:**

1. Scalability: Nodes operate independently, so you can add more nodes to handle increased load without disrupting the system.
2. Performance: Nodes can process their own data without interference, which leads to faster processing times.
3. Fault tolerance: If one node fails, it doesn't affect the functionality of others

**Disadvantages:**

1. Data is distributed across the cluster requires shuffling data between nodes
2. Performance is heavily dependent on how data is distributed across the nodes in the system
3. Compute can’t be sized independently of storage.
4. Snowflake’s architecture is a hybrid of traditional shared-disk and shared-nothing database architectures. Like shared-disk architectures, Snowflake uses a central data repository for persisted data that is accessible from all compute nodes in the platform. But like shared-nothing architectures, Snowflake processes queries using MPP (massively parallel processing) compute clusters where each node in the cluster stores a portion of the entire data set locally.

A blue and white diagram with words

Description automatically generated with medium confidence

1. **Service Layer:**  Which accepts SQL requests from users, coordinates queries, managing transactions and results.  Logically, this can be assumed to hold the *result cache*– a cached copy of the results of every query executed.
2. **Compute Layer:**  Which does the heavy lifting.  This is where the actual SQL is executed across the nodes of a Virtual Data Warehouse.  This layer holds a cache of data queried and is often referred to as Local Disk I/O although this is implemented using SSD storage.  All data in the compute layer is temporary, and only held if the virtual warehouse is active.
3. **Storage Layer:**  Which provides long term storage of results.  This is often referred to as *Remote Disk* and is currently implemented on either Amazon S3 or Microsoft Blob storage.

**Snowflake Cache Layers**

1. **Result Cache:**Which holds the results of every query executed in the past 24 hours. Query Cache is ideal for quick results of repeated queries. [Service layer]
2. **Local Disk Cache:**Queried data is stored onto the virtual warehouse’s SSD.When the query is not identical, result cache cannot retrieve it. Data in SSD cache is stored as long as warehouse is running(until suspended). [Computer layer]
3. **Remote disk:** Actual data on disk [Storage layer]

**Snowflake Performance Summary**

The sequence of tests was designed purely to illustrate the effect of data caching on Snowflake. The tests included:-

* Raw Data:  Including over 1.5 billion rows of TPC generated data, a total of over 60Gb of raw data
* Initial Query:  Took 20 seconds to complete and ran entirely from the remote disk.  Quite impressive.
* Second Query:  Was 16 times faster at 1.2 seconds and used the *Local Disk*(SSD) cache.
* Result Set Query:  Returned results in 130 milliseconds from the result cache (intentionally disabled on the prior query).

Important points to remember:

1. You always need a virtual warehouse to execute queries.
2. Always use limit clause with select \* from queries.
3. During dev activity, always keep auto suspend high
4. Share virtual warehouse when group of users are working on the common tables.
5. Never disable your cloud service layer result cache.
6. Reusing query result in snowflake is free.