Data 603 Technical Paper

Comparison of Cloud Data Warehouses

Amazon Redshift, Google BigQuery, Snowflake

Kam Hin Ho

DATA 603

University of Maryland

November 28, 2020

Abstract

We live in the age of distributed computing, and public cloud platforms have offered unlimited and flexible compute and storage resources for daily demand. With the Software-as-a-Service, such as Google App and Salesforce, the number of users is adapted to the newly cheaper, convenient, and fewer maintenance options. As unstructured data evolves, the demand for data analytics was increased. A traditional data warehouse is not able to handle well. For example, resizing of resources cannot be performed without compromising availability and performance. This means service users typically end up with either over-provisioned or under-utilized expensive resources to accommodate possible peak demand.

The cloud-based data warehouse was introduced as a modern solution for addressing the needs of data processing and reporting which requires flexibility and adaptability in both computing and storage resources. There are a few competitors in the cloud data warehouse market who can make the transition from the traditional star schema-based data warehousing to a modern elastic data warehouse platform possible: Snowflake, Amazon Redshift, and Google BigQuery.

Snowflake is an analytic data warehouse provided as Software-as-a-Service. Snowflake was built by a SQL database engine with its unique architecture for the cloud. Snowflake provides a cloud-based data storage and analytics service, and it allows corporate users to store and analyze data using cloud-based hardware and software. Recently in 2020, Snowflake went public with its IPO listing on the NYSE which was the highest value software IPO. Amazon Redshift is a fully managed, cloud-based, petabyte-scale data warehouse service by Amazon Web Services (AWS) which was initially released in 2012. It is an efficient solution to collect and store enterprise data and enables users to perform analysis by using various business intelligence tools to acquire new insights for business and customers. On the other hand, Google BigQuery is a highly scalable,

serverless multi-cloud data warehouse that is designed for business agility. It is a Software as a Service that supports querying using ANSI SQL with a built-in machine learning capability.

Although the three data warehouses have their similarities, they have the differences that make them unique. A business may have difficulties to choose the best cloud-based data warehouse that can fulfill their needs. This paper will describe and compare the architecture, cloud/on-premises options, storage and compute, Scalability, and security of the three cloud-based data warehouses. The paper will also focus on comparing the databases' maintenance, data access, and pricing as well as application recommendations for the business.

Google Architecture

BigQuery

Google BigQuery is built on top of Dremel technology which has been in production internally in Google since 2006 [4]. Dremel is a query service that allows users to run SQL-like queries against big data and get accurate results in mere seconds [5]. It dynamically executes by dividing the slots to queries on an as-needed basis, maintaining fairness amongst multiple users who are all querying at the same time [6,7]. On the other hand, Google BigQuery relies on Colossus for storage which is the latest distributed file system from Google [7]. Each Google data center has its Colossus cluster, and each Colossus cluster has enough disks to give each BigQuery user thousands of dedicated disks at the same time [7]. In other words, Colossus acts as a distributed storage that handles replication, recovery, and distributed management. As for computation, BigQuery relies on Borg for computation which is Google's large-scale cluster management system [7]. Borg clusters run on a tremendous number of machines and cores which ensure the capacity and speed for the query demands [7]. Besides resource coordination and computing capacity, Google relies on its Jupiter network for managing a large workload that may be throttled by networking throughput [7]. It provides enough bandwidth to allow nearly 100,000 machines to communicate with any other machine at a speed of 10 GB per second [7]. Jupiter allows the user to bypass the traditional approach (separation of storage and compute) and read terabytes of data in seconds directly from storage [7].

Snowflake

Snowflake is designed to be an enterprise-ready service that communicates through RESTful interfaces [2]. Representational state transfer, known as the REST, is a software architectural style that supports the interoperability between computer systems on the internet by allowing requesting systems to access and manipulate textual representations of web resources using uniform stateless operations [1]. Snowflake offers high degrees of usability and interoperability with high availability for the enterprise. Therefore, Snowflake architecture is service-oriented and composed of high fault-tolerant and independently scalable services [2]. Snowflake architecture has three layers: Data Storage, Virtual Warehouses, and Cloud Services. The data storage layer uses a cloud platform, such as Amazon S3, to store table data and query results; Virtual Warehouse layers handle query execution within the elastic clusters of virtual machines [2]. As for the Cloud Services layer, it is a collection of services that manage virtual warehouses, queries, transactions, and all the metadata that goes around the database schemas, access control information, encryption keys, and usage statistics [2].

Amazon Redshift

Amazon Redshift is a fully managed highly scalable data warehouse service in AWS. The cluster is one of the core infrastructure components of the Amazon Redshift data warehouse. A cluster is composed of compute nodes and leader nodes. The leader node will be used to coordinates the compute nodes and handles external communication if a cluster is provisioned with two or more compute nodes [8]. Client applications, like ETL tools and business intelligence reporting tools, communicate with the cluster only with the Leader Node [8]. The leader node handles the communications between the client programs and all communication with the compute nodes. It parses and develops execution plans to carry out database operations [8]. The leader node will compile the code according to the execution plan and distributes the compiled code to the compute nodes along with a portion of the data to each compute node [8].

On the other hand, each computes node has its dedicated CPU, memory, and attached disk storage, and each of them is responsible for the actual execution of queries and have data stored with them [8]. They are responsible for executing the queries and return intermediate results to the Leader Node for aggregating the results [9]. However, there are certain SQL functions in Redshift architecture that can only be executed on the Leader Node, and an error may occur if it references tables that reside on the compute nodes [8,9]. There are two types of Compute Nodes in Amazon Redshift architecture: Dense Storage and Dense Compute. The difference between Dense Storage and Dense Compute is dense storage nodes use Hard Disk Drives (HDDs) to create a large data warehouse with lower cost, and the other uses Solid-State Drives (SSDs) for creating a high-performance data warehouse with a higher price point [9].

A compute node consists of slices, and each slice has a portion of Compute Node's memory and disk assigned for performing Query Operations [8]. The Leader node distributes data and apportions the workload of queries or other database operations to the slices. Then, the slices work in parallel to complete the operation. An even distribution of data enables Redshift to assign workload evenly to slices and maximizes the benefit of parallel processing. Amazon isolated the client applications from its high-bandwidth connection and custom communication protocols to ensure a private and high-speed network communication between the leader node and compute nodes [8]. As for data storage, a cluster contains one or more databases, and user data is stored on the compute nodes [8]. It supports the SQL client communication with the leader node which coordinates query execution with compute nodes.

Cloud/on-premises deployment

All of these databases provides cloud-only options, and it is not supporting on-premise deployment at the moment.

Google BigQuery

Google BigQuery can be deployed within Google Cloud Platform and across other rival clouds, such as AWS and Azure (coming soon) [10,11]. Google BigQuery introduced Omni in July 2020, a flexible multi-cloud analytics solution powered by Anthos. It allows the user to deploy google BigQuery and access rivalry cloud data with SQL without leaving the BigQuery user interface [11]. With Omni, Google BigQuery increases its flexibility on cross-cloud platforms, which may increase the user's favorability to use their data warehouse service.

Snowflake

Snowflake components run completely in a public cloud infrastructure [3]. It can be deployed within AWS, Google Cloud Platform, and Azure. Like Google BigQuery, it provides flexibility for the user to choose their preferred cloud platform.

Amazon Redshift

Unlike Google BigQuery, Amazon Redshift can only be deployed within their own Amazon Web Services, such as public subnets, private subnets, and Amazon S3 [10].

Storage & Compute Scalability

Google BigQuery

Google BigQuery uses Colossus as a distributed file system for storage. Google BigQuery uses the ColumnIO storage format and compression algorithm to optimize and store the data [7]. As a result, BigQuery users will have the optimal way for reading large amounts of structured data. As for computing, Google BigQuery leverages Borg to allocate the computing capacity for Dremel jobs. Borg clusters run a workload across dozens of thousands of machines with hundreds and thousands of cores for handling short-lived latency-sensitive requests, such as Gmail, Google Docs, and BigQuery [15]. Thus, it provides high reliability and availability computing environment for query operations [7]. Since Colossus and Borg are separated, Colossus allows users to scale to dozens of Petabytes in storage smoothly without adding more computing resources and vice versa [7].

Snowflake

Snowflake manipulates the existing cloud platform for storage [2]. When data is loaded into Snowflake, the values of each attribute or column are grouped in a scheme called PAX or hybrid columnar format with aggressive metadata caching [2,10]. As for computing, Snowflake user only configures the size and number of computing clusters, and every computes cluster reads the same data. The compute cluster or the virtual warehouse is an MPP compute cluster composed of several compute nodes assigned by Snowflake from a cloud provider [14]. Each virtual warehouse is an independent compute cluster that does not share compute resources with other virtual warehouses [14]. Thus, it can eliminate the performance reduction from sharing with other virtual warehouses. For scalability, Snowflake offers different warehouse types from small to large which roughly correlate to the number of vCPUs or memory based on the type that the user chooses [4]. Snowflake's scaling policies can adjust the number of clusters according to the running workloads in either standard or economy modes for different usage scenarios [4].

Amazon Redshift

In Dec 2019, Amazon announced its third generation RA3 node type which provides the user the ability to scale compute and storage separately [12]. In the previous generation, DS2 and DC2 nodes had a fixed amount of storage. If the user wants to increase the storage, they are required to adding more nodes to the cluster [12]. The new RA3 brings Amazon Redshift closer to the user experience of Snowflake which allows users to easily decouple compute from storage workload. To achieve that, the user must configure individual compute clusters with a fixed amount of memory, compute, and storage [13].

Security

Google BigQuery

Google BigQuery automatically encrypts all data before it is written to disk. The data will be decrypted automatically when read by an authorized user. Users can choose to use customer-managed encryption keys (CMEK) or the default Google encryption keys for protection [16]. As for default encryption at rest, each BigQuery object's data and metadata are encrypted under the Advanced Encryption Standard (AES) which was established by the U.S. National Institute of Standards and Technology [16]. Google BigQuery also allows the user to encrypt individual values within a BigQuery table by using Authenticated Encryption with Associated Data (AEAD) encryption functions [16]. AEAD encryption function allows the user to create keysets that contain keys for encryption and decryption. Then, the user uses the keys to perform encryption to individual values in a table and rotate keys within a keyset [17]. For Data in transit, Google uses Transport Layer Security (TLS) when transferred between machines [16].

Snowflake

Snowflake uses end-to-end encryption (E2EE) by default for data both in transit and at rest [18]. E2EE is a form of communication in which only end users can read the data [18]. In other words, only the user and the runtime components can read the data, even Snowflake's cloud computing platform or any ISP has no clear access to the data [18]. Like Google BigQuery, Snowflake also provides its default encryption options. Snowflake uses Virtual Private Snowflake (VPS) to store an encryption key in memory on dedicated servers/internal stages [10]. Snowflake provides internal stages where users can upload and group the data files before loading the data into tables. The data files are automatically encrypted by the client on the local machine before the transfer to the internal stage [18]. Then, the data from the stage will be transformed into Snowflake's proprietary file format and stored in the VPC storage container (data at rest) [18]. For query result, it will automatically be encrypted when unloaded to a Snowflake-provided stage for the user to download and decrypts on the client side [18].

Amazon Redshift

In Amazon Redshift, encryptions are optional. Users can choose to use AWS customer-managed key (CMK) or AWS Key Management Service (AWS KMS) encryption to protect data at rest [19]. To let Amazon Redshift automatically migrates the data to a new encrypted cluster, KMS encryption must be enabled [19]. The encryption protects the data blocks, system metadata, and the created snapshots from the encrypted cluster. Though the encryption is optional, Amazon Redshift recommends enabling it for the cluster that contains sensitive data [19]. Depends on the guidelines or regulations, a dataset like payment card industry data must mandatorily follow the specific encryption guidelines for handling the data [19].

Maintenance

Google BigQuery

Google BigQuery partitions and sorts data in the background, and it keeps the user away from the manual operations or optimization. Due to the cost and data scanned, the user is likely to spend more time on optimizing the queries than any other maintenance [10]. The user needs to spend some time maintaining the design of the tables to minimize any excessive scans based on the query patterns, such as ensuring the correct usage of the partition keys, clustering, and range partitioning [10]. Otherwise, Google Query automates much of the toil and complexity associated with setting up and managing the user's data warehouse.

Snowflake

Snowflake only requires the user to choose the size of the warehouse and setting up the scaling and auto-suspend policies [10]. Since snowflake is a SaaS, snowflake handles most of the maintenance and tuning [3]. Unless the user wants to define the clustering keys for optimizing the micro-partitions scan in the larger tables, there is not much maintenance needed for the user [10,20].

Amazon Redshift

Amazon Redshift used to rely on the user to manually schedule some operations, such as VACUUM DELETE, VACUUM SORT, and ANALYZE. In 2019, Amazon Redshift improved these operations, and it can conditionally run in the background [10]. However, if the data is too large to vacuum sort on disk, then the user must resize the cluster or left the data unsorted. Moreover, the query planner still relies on the table statistics heavily, so the user has to keep it updated regularly [10].

Data Access

One of the important functions of data warehouses is the user can access the data easily. Most of the databases have relied on ODBC/JDBC. ODBC is an SQL-based Application Programming Interface (API) created by Microsoft that is used by Windows software applications to access databases via SQL [21]. JDBC is an SQL-based API created by Sun Microsystems to enable Java applications to use SQL for database access [21]. Besides ODBC/JDBC, data warehouses have increases the data access options to accommodate the new forms of data retrieval and manipulation.

<u>Google BigQuery</u>

Google BigQuery can access the data by ODBC and JDBC via Simba drivers, and data can also be access from the BigQuery user interface and its command-line tool. There are several APIs that are available, such as BigQuery connections API for federated sources, BigQuery Jobs API for handling data, and BigQuery Storage API for providing access to BigQuery managed storage [10,22].

<u>Snowflake</u>

Besides the ODBC/JDBC access its command-line tool, Snowflake offers data access via Spark plugin, Kafka, Python, Node.js drivers for specific languages, and Snowsight [10]. Snowsight is designed to support data analyst activities which is the SQL Worksheets replacement. It is not only providing access to the data but also handles the querying and visualizing the data [23]. In the Data Cloud Summit 2020, Snowflake unveiled a new feature called Snowpark, which the company is positioning as a new developer experience for enabling data engineers, data scientists, and developers to develop data routines, such as ETL, data preparation, and feature engineering pipelines [27].

Amazon Redshift

Like Google BigQuery and Snowflake, Amazon Redshift provides ODBC/JDBC access via AWS provided drivers. The user can access the data from the Redshift user interface in the AWS console for some node types, such as ra3.*, ds2.8xlarge, dc2.large [10]. The user can also access the data via their data access API, AWS CLI and the Postgres command line tool [10].

Pricing

Google BigQuery

Google BigQuery storage charges have 2 options: Active and Long-term. The storage charges will be charged as active if the data stored in tables or in partitions that have been modified in the last 90 days [26]. If no data modification has been made in the last 90 days, it will be charged as long-term [26]. As for the query cost, the user can choose between two pricing models: On-demand and Flat-rate. The on-demand model is relatively flexible for the user, which is based on the amount of data processed by each query the user run [26]. In contrast, the flat-rate model is more predictable as the user purchase dedicated resources for query processing and is not charged for individual queries [26].

Snowflake

Snowflake offers a time-based pricing model for its computing resources, in which users are charged for execution time [24]. In other words, the user will still get charged when the warehouse is running even though they are using it for querying. Depends on the plan, the number and size of the warehouses, cloud provider, and geographic location, the cost per credit will vary [10]. Repurchasing storage capacity is also possible for a user who wants to have a more predictable long-term pricing [10].

Amazon Redshift

There are several pricing options for Amazon Redshift: Spectrum pricing, Concurrency Scaling pricing, and Redshift managed storage pricing. Spectrum pricing is the user pay for the number of bytes scanned when running SQL queries directly against the data in the S3 data lake [25]. Concurrency Scaling pricing is charged at an hourly rate depending on the instance type and the number of nodes that the user chooses [24,25]. As for RedShift managed storage pricing, the user only needs to pay for the data that is stored in RA3 clusters independent of the number of compute nodes provisioned per hour [25].

Reference:

1. **https://datavirtuality.com/connectors/integrate/generic-soaprest/snowflake/**
2. **http://pages.cs.wisc.edu/~remzi/Classes/739/Fall2018/Papers/p215-dageville-snowflake.pdf**
3. **https://docs.snowflake.com/en/user-guide/intro-key-concepts.html**
4. **https://panoply.io/data-warehouse-guide/bigquery-architecture/**
5. **https://cloud.google.com/files/BigQueryTechnicalWP.pdf**
6. **https://hevodata.com/blog/google-bigquery-data-warehouse/#a3**
7. **https://cloud.google.com/blog/products/gcp/bigquery-under-the-hood**
8. **https://docs.aws.amazon.com/redshift/latest/dg/c_high_level_system_architecture.html**
9. **https://hevodata.com/blog/redshift-architecture/**
10. **https://poplindata.com/data-warehouses/2020-database-showdown-bigquery-vs-redshift-vs-snowflake/**
11. **https://cloud.google.com/blog/products/data-analytics/introducing-bigquery-omni**
12. **https://aws.amazon.com/blogs/aws/amazon-redshift-update-ra3-4xlarge-instances/**
13. **https://fivetran.com/blog/warehouse-benchmark**
14. **https://docs.snowflake.com/en/user-guide/intro-key-concepts.html**
15. **https://storage.googleapis.com/pub-tools-public-publication-data/pdf/43438.pdf**
16. **https://cloud.google.com/bigquery/docs/encryption-at-rest**
17. **https://cloud.google.com/bigquery/docs/reference/standard-sql/aead-encryption-concepts**

18. https://docs.snowflake.com/en/user-guide/security-encryption.html#end-to-end-encryption
19. https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-db-encryption.html
20. https://docs.snowflake.com/en/user-guide/tables-clustering-keys.html?insvid=175b08f809ae36f5--1605122349117
21. https://www.microfocus.com/documentation/xdbc/win20/BKXDXDINTRXD1.5.html
22. https://cloud.google.com/bigquery/docs/jobs-overview
23. https://docs.snowflake.com/en/user-guide/ui-snowsight-visualizations.html
24. https://medium.com/2359media/redshift-vs-bigquery-vs-snowflake-a-comparison-of-the-most-popular-data-warehouse-for-data-driven-cb1c10ac8555#:~:text=Pricing%20models,-Determining%20which%20solution&text=In%20terms%20of%20pricing%2C%20Redshift,pay%20for%20flat%2Drate%20pricing.
25. https://aws.amazon.com/redshift/pricing/
26. https://cloud.google.com/bigquery/pricing#active_storage
27. https://www.datanami.com/2020/11/17/snowflake-extends-its-data-warehouse-with-pipelines-services/