National College of Ireland

Data Analytics using SaaS, It's Comparison and Improvement techniques

MSc Research Project
Cloud Computing

Nikhil Kumar Singh
Student ID: 19202491

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Nikhil Kumar Singh
Student ID:	19202491
Programme:	Cloud Computing
Year:	2021
Module:	MSc Research Project
Supervisor:	Sean Heeney
Submission Due Date:	16/12/2021
Project Title:	Data Analytics using SaaS, It's Comparison and Improvement techniques
Word Count:	7836
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	16th December 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Contents

Data Analytics using SaaS, It's Comparison and Improvement techniques

Nikhil Kumar Singh
19202491

Abstract

Big-data is no longer a new concept, and the challenges associated with it are quite common, with speed, cost, security, and backups being the most prevalent. Almost every organization that deals with a moderate or large amount of data faces these challenges on a regular basis, which is why I chose this topic for my research. Given that we are in the cloud migration era, I was motivated to investigate how these challenges can be addressed through the use of Software-as-a-Service offerings. I used Snowflake and the provider to access large chunks of data from major cloud storage providers such as Amazon, Google, and Microsoft for the demo purpose. During the implementation, I observed that Snowflake is extremely capable of addressing these issues and includes some truly advanced features. My research focused on Snowflake's security, data retention, and snowpipe features. Additionally, because it is a SaaS, it is extremely user friendly, as all features are accessible via a web browser and do not require installation. Additionally, I used some optimization techniques to boost performance and cut costs. I was able to automate some of the day-to-day tasks using features such as tasks and streams, which is critical for any business that deals with data.

1 Introduction

Big data is a term used to describe large data sets that can't be processed by traditional databases. It's an ambiguous term because it could mean any number of things, including a company's customer database. In recent years, the amount of user data generated have increased exponentially and is expected to grow more in the coming years as well. With this substantial amount of data, there is a requirement of high computing power to process this data to extract content for meaningful purposes. Traditional relational database management systems are ill-equipped for handling big data.

The modern computing architecture has changed to a heterogeneous one with the increasing power of hardware coupled with the capability of the process to work with multiple resources; However, with the recent advancements in cloud computing, organizations prefer to use cloud services to store and process the large datasets as against using the physical on-premise infrastructure.

In the past decade, the number of cloud service providers has also increased and this includes the major IT companies such as Amazon, Google and Microsoft. For choosing the right service provider, it is very important to consider several parameters and compare them based on them. Examples of such parameters are Latency, Pricing, Storage, Uptime, etc.

The large datasets require extremely high computation power and there are costs associated with the same. With so many options available nowadays and all in par with each other, it can be confusing at times to make a decision about which cloud service provider to opt for as there are so many factors that needs to be considered before making the decision, specially the long-term.

1.1 Motivation for the Research:

There are numerous challenges that needs to be addressed when starting with data analytics, specially when it big-data. Very often the engineers/organizations have to go-through lots of hits and trials to find the best solution. I have personally witnessed the change of platform after discovering new challenges which people were unaware initially and it costed a month of rework to the organization and it's team members. There are times when the data engineers looks for techniques to optimize certain aspect's of their work. It's always better if someone has already done all the research and posted the techniques one-place.

Following are some of the major day to day challenges that needs to be addressed and would be covered in this research paper.

1. Data Quality:- When dealing with large set of data or even the smaller ones, the quality of data is affected when it is transferred or loaded from one platform to another. It's a time consuming task to get the data in the desired form on different platforms and lots of manual efforts needs to be made as the commands for such operations are slightly different on different platforms such as oracle, postgresql, snowflake, etc.

2. Data Storage:- Large datasets need large amount of storage this also involves additional storage for the backups. The storage factor is an important concern as their are some major parameters that needs to be considered while choosing the right tools and service providers for data analytics.

3. Validating Data:- The data needs to be validated before going ahead with the visualization and any issues with the quality of data are highlighted in this stage.

4. Accumulating data from different sources:- There are times at which the organization needs to get data from multiple cloud service providers and this could be tricky and time consuming tasks; however with the new advancements in the applications, the scripts

Snowflake is a real software-as-a-service solution. More particularly, there are no hardware's to select, install, configure, or manage (virtual or actual). There isn't much to install, configure, or administer in terms of software.

Since snowflake is relatively new and is very powerful in terms of performance and features, it is definitely worth looking into. In terms of research, i could not find any research paper related to snowflake and hence decided to come up with this research topic so as to do the hard-work of looking into all the aspects of snowflake and come-up with the findings that could help others about the benefits and challenges in using swowflake as SaaS and how does it works best with the cloud storage service providers.

In this paper we are going to use Snowflake, which is a cloud-computing based data warehousing company and is used to work with extremely large datasets. Here we are going to compare the performance for the tool with respect to several cloud service providers, such as AWS, GCP and Azure. We will further look into the optimization and improvement aspects as well..

The purpose of this research is as mentioned below:

1. Compare the performance of AWS, GCP and Azure cloud service providers to store and process large datasets.
2. Latency comparison for retrieving data from the cloud service providers.
3. Latency comparison when the data is modified and when the data is dynamic.
4. Current big-data problems addressed by Snowflake.
5. Cost management with snowflake.
6. Optimization techniques for cost management.
7. Addressing security related concerns.

The whole document is broken into six parts, the first of which is titled "Introduction." In next section "Related Work" we are going to look into some of the research paper's related to big-data analytics and data warehousing techniques and look into the findings of the authors and understand the strengths and limitations of their research.

In the third section we will look into the design aspects of the research where we will look into the pipeline of data ingestion and the services and applications used to load and retrieve data from the cloud service providers. We will also look into the entire lifecycle of data warehousing. Section four will cover the "Implementation" of the design as mentioned in section three. For the implementation of the project several cloud services are applications will be using and shall also be mentioned in the configuration manual.

Section five, will cover the evaluation and results analysis based on the findings that also includes optimization techniques.

Section six is Conclusion and Future work, to conclude the finding and mention what all could be done in future.

2 Related Work

Data analytics is a very important part of the day to day operations these days and all organizations are looking for techniques in which they can process the data faster and cheaper. Since the data is collected from different sources on daily basis and the data is not actually in the desired form, it is important to process the data in the timely manner so that it could be utilized for the daily operations. In this section we will look into the previous research carried out to address the similar problems and discuss the strengths and limitations in brief. We will also look into the usage of cloud services for the storage and processing of data, which would be beneficial for my own research.

2.1 Big Data Analytics using Cloud Computing

In this paper the Khedekar & Tian (2020) has implemented a multi-tenant infrastructure for the processing of big-data on AWS platform. As the critical findings of the research, the author has shared the following results.

- For real-time data, the number of get and put records is constant and proportionate to the time.
- The implementation of multi-tenant infrastructure on cloud platform provided by AWS is efficient in several factors such as cost, time, resources, and people in comparison to the on-premises infrastructure.
- The author also mentions that, with the implementation of the infrastructure on cloud platform, the users can focus on the data analytics rather than the environment, as the environment needs to be setup only once and is very easy to manage.
- The author has mentioned that the prototype setup can be done very easily at very low cost, since the cloud service provides follow the Pay-as-you-go model, in which the user pays only for the services used.
- The cloud platform further provides flexibility in terms of storage and the system specifications. The configuration of the system up-scales or down-scales as per the requirement of the user and the user would only need to pay for the time for which they have used that particular configuration.
- With the traditional on-premises infrastructure, there would be a bottleneck when the storage capacity is reached.

Summarizing the paper, it focuses on the process of the implementation of the multi-tenant cloud infrastructure and the analytics on big-data using the traditional techniques. It mentions the use of technologies such as Server-less cloud computing, Multi-tenancy , Amazon Web Services (AWS), Big-data The strength of the paper being the merits of using the cloud services for storage. The paper doesn't mentions the optimization techniques and the use of new and trending data warehousing techniques.

2.2 Processing of Geospatial data on Cloud

In this paper, Wan et al. (2021) have used AWS cloud services for the implementation of the project and have described the ongoing efforts for the storage, processing, analysis and visualization of the geospatial data. The main purpose of the project was to make the geospatial data available to the regular science users who can manage their own geospatial data and perform actions on the HUBzero platform.

The implementation is divided into two components: the client and the server, and there are many access points to accommodate the community's diverse demands. The deployment of the Hub is done with Cloud ECS. The ECS basically is the virtual infrastructure that consists of all the component's that a typical on-premises system will have, such as CPU's, Memory, OS, etc.

For the storage, Wan et al. (2021) have used Object Storage Service(OBS) for the storage of big-data, The OBS provides unlimited storage for storing massive amount of data securely at a very low cost.Finally, they reviewed the design, implementation, and usage of the Geospatial Hub, which was created to aid scientists from many professions in putting their data and tools online for public sharing. They initiated a demonstration project to illustrate how they might utilize simulation tools and share their results. The paper helps how the data on cloud can be accessed globally. Xin et al. (2019) has demonstrated how cloud services can be utilized for ship big data processing.

2.3 Visualization of data on cloud

In this research paper, Larrick et al. (2020) have shown how using the cloud services for storage helped them to deliver a infrastructure for visualizing terrain tile data within the computer browser, more importantly the tiles could be accessed by multiple users at the same time.

The motive of the research was to visualize data sets in the form of three dimension. To store such large data sets, They employed Amazon EC2 computers to retrieve and decode the terrain data.

They have addressed another challenge to improve the performance of the application by implementation of parallel servers so the url's from multiple users would be redirected to the available servers.

The authors discuss the difficulty of keeping a huge amount of data on a single machine or network and how they overcame it by using Amazon's EC2 service. With the S3 storage, the researchers can access the data from any part of the world.

The researches have also highlighted the challenges of the speed at which such large amount of data is accessed. This is something that would be a part of my research.

In this research, the researches have used Amazon S3 for storage which helps with my

research as well, in my research I would be comparing the other cloud storage options as well.

2.4 Data Access Security Issues

When laying the groundwork for a Big Data cloud computing environment, unauthorized access is a significant danger. This situation is fraught with both internal and external risks. External security concerns result in data loss and damage on the big data cloud computing platform, primarily because users fail to timely enforce security mechanisms when storing data; internal security concerns, or the inability to function in conjunction with big data cloud computing prerequisites, are the result of internal personnel acting improperly. Farsi et al. (2020).

Cloud computing will be a massive and difficult paradigm to cope with in the future. The authors conducted a comprehensive and critical review of the literature on security threats, cloud computing paradigms, and cloud computing security measures on the basis of this study. Additionally, the limitations of current cloud computing research were examined at the time this article was written. Zhong et al. (2019) emphasis on the storage and access of sensitive data on the cloud.

Comprehensive data security solutions, such as encryption and other approaches (that are used to provide cloud security), are critically evaluated in the papers written by their original authors. This program has the potential to pave the way for future research on cloud computing. In general, Farsi et al. (2020) have identified the primary problems associated with cloud data storage security. Such problems would be addressed in my research.

2.5 Data Isolation Security Issues

Data isolation security is crucial in big data cloud computing. It is most prevalent in the sharing operation from an usage aspect. At the present, big data cloud computing clients are predominantly collective, with business organizations accounting for a sizable portion. Resources must always be available for sharing, even more so for government entities Fuguang (2019). Due to the fact that external computers are not needed to transit through the isolation wall in this scenario, encryption is unfeasible. Because the environment is not secure enough, violent attackers may damage and disclose data. The author highlights the importance of security when the data is on cloud.

2.6 Data Destruction Security Issues

To preserve data security and to save storage space in the big data cloud computing environment, some data must be erased. Data leakage and unauthorized use may occur if the time for data destruction is not handled properly or if the data is not completely

deleted Liu et al. (2020). The importance of the data destruction due to several reasons have been highlighted and will also be help me out with my research.

2.7 Data Integrity Security

Aman & Yadav (2019) emphasised that The word "data integrity" refers to the resiliency and trustworthiness of data during the course of its life. It might relate to the state of your data, such as whether it is valid or invalid, or to the process of ensuring and preserving the data's validity and accuracy. The phrase "data integrity" relates to the operation and storage processes' data integrity. In a cloud computing environment with a large amount of data, the back-end application is always configured as dynamic processes. In reality, ensuring data integrity and security is tough due to internal and external attackers. The integrity of data is extremely important and is something that is always a critical part of data analytics. The paper highlights the importance and methods to address such concerns. It is also a major of my research and would be addressed in sections ahead.

2.8 Cloud computing and data security

In this research paper Wang et al. (2021) has highlighted the Data access, data isolation, data integrity, data destruction, data transfer, and data exchange are all covered in terms of data security and privacy control. Finally, a virtualization architecture and accompanying tactics are offered to counter attacks and improve data security in the big data cloud .

For the implementation of the project, the authors have used several cloud computing tools for the processing of big data on cloud. They have used advanced tools such as Map-Reduce, Google Distributed file system (GFS), Hadoop etc. Anwar Hossain (2019) have highlighted the the design and development for enhancing the security of data in cloud computing. Additionally, Yuqing & Mo (2019) also mentions the method for securely securing big data on the cloud also a similar approach has been mentioned by E et al. (2019) for medical purposes and similar concept is mentioned by Suyel et al. (2020).

After highlighting the concern of working with the big data, they have highlighted tool and techniques that would make the processing of big-data efficient and secure. This paper mentions almost all the major challenges related to data analytics and that's a major strength of the paper and I would be addressing most of those in my research.

2.9 Cloud data lake for big data analytics

The term Data Lake is not entirely new and has been around for some time. In this research paper, Zagan & Danubianu (2021) have explained the concept of data lake, their types and the challenges associated.

The authors have also explained the cost's associated with the on-premises and cloud data lakes and that proves why cloud data lakes are better and cheaper and that's the strength of the paper. It doesn't compares the datalake service providers and that is something that I have considered while working on this research. We will also need to come across such challenges and see how those can be fixed in this research paper.

The authors conclude by stating that data lakes must be able to store data in its raw state readily in order to simplify data exploration, automate data ETA management processes, and allow the use of a broad variety of data analysis methods.

2.10 Cloud for Big Data Analytics Trends:

This paper mentions about the features of cloud computing and it's purposes and benefits. Ara & Ara (2016) mention that Cloud computing may offer big data with infrastructure, platform, and software resources as a service, among other things. Organizations that use cloud computing as part of their infrastructure get significant benefits from Big Data. Big data and cloud technologies are converging, making big data analytics on clouds a feasible choice for many organizations. The scope of big data analytics is expanding to encompass information from a wide range of sources Akansha et al. (2020).

Big data may be made up of the vast majority of the highest-volume, fastest-streaming, and most complicated data available Ha-Kyun et al. (2019). The term "Data Analytics as a Service" refers to the use of the cloud for big data analytics. Cloud computing services are used by businesses of all sizes to save capital expenditure since they supply the resources necessary to execute their applications Lidia & Ogiela (2020). One of the goals of my article is to provide an overview of big data and cloud technology, as well as an insight into the interplay of big data and cloud technology, as well as a quick explanation of Data Analytics as a Service trends and limits, among other things. The article discusses the future or trends in the use of cloud computing for big data analytics in the future. The paper is slightly helpful in highlighting the services and benefits of using the cloud services.

3 Research Methodology

The entire process of the research can be divided into the 9 parts and each of them is very important for the research as they serves real-time challenges and are further explained in the following subsections.

3.1 Collection of data:

For this research i have collected data from Kaggle. For the purpose of making the size of data larger, i have used 7 datasets from Kaggle and are listed below. It's to be noted that

i have created this large dataset only for the purpose to demonstrate the implementation of adding data to the cloud storage and access it from Snowflake application and perform analytics on that data. Snowflake also offers sample big-data which could be used to test out the features of the application and I would be using that as well.

1. F1 2020 race data [1]
2. Brazilian E-Commerce Public Dataset by Olist [2]
3. Bitcoin Historical Data [3]
4. The Movies Dataset [4]
5. Trending YouTube Video Statistics [5]
6. 120 years of Olympic history: athletes and results [6]
7. 5000 Movie Dataset [7]

We understand from the above mentioned data sets and different kinds of data sets are used for this research. This is done to make sure that the research is applicable to different kinds of data-sets.

3.2 Loading data into the cloud:

Data entry and data extraction are critical processes for every contemporary data warehouse system. I'll be importing bulk data from on-premises systems and cloud storage, and will give insight into the methods necessary to load streaming data into the SaaS application. The collected data would be uploaded on cloud service providers for storage. In this research I am uploading the identical data on Azure, GCP and AWS and will be sharing the comparison between the three in the later sections. Sangeeta et al. (2020) demonstrates the process to load IoT data into the cloud and it's management.

3.3 Retrieving/Accessing data from the cloud:

Later the data from the cloud service providers is than loaded into snowflake. There are two ways to import data from cloud storage. The exterior stage procedure is advised. An external stage is similar to a virtual stage that exists within Snowflake and references data

[1] https://www.kaggle.com/coni57/f1-2020-race-data
[2] https://www.kaggle.com/olistbr/brazilian-ecommerce?select=olist$_{o}rder_{p}ayments_{d}ataset.csv$
[3] https://www.kaggle.com/mczielinski/bitcoin-historical-data
[4] https://www.kaggle.com/rounakbanik/the-movies-dataset?select=ratings.csv
[5] https://www.kaggle.com/datasnaek/youtube-new
[6] https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results
[7] https://www.kaggle.com/tmdb/tmdb-movie-metadata

stored in a public cloud storage bucket. The reference portion is critical since the external stage does not contain any data; rather, it displays everything in the cloud storage.

Alternatively, you can load data from cloud storage by directly referencing it in your COPY statements.

3.4 Creating a pipeline to load the updated data from the cloud:

It is very important to make the data dynamic. As the organizations deals with updated data on daily basis. The new data is added and processed on daily basis. The idea is that the data is updated daily on the weekends and is pushed to the cloud storage. Further tasks such as data loading, reloading, cleaning, etc. We will look into how this can be done using snowflake in this research which will make such processes automated.

3.5 Analytics of the data:

Extracting meaningful data from the tables is the main task and for that the queries are designed. Once the data is loaded such queries would help extract those data from the tables. Data Analytics is something that is done on the daily basis. Be it, to find more meaningful data or to find or fix the issues with within the current system. It is also important for data visualization, as the data filtered/extracted would be further used with the tools such as PowerBI to present statistical analysis.

3.6 Data Pipelines, Streams and Tasks:

Snowflake, like many other data platforms, provides developers and consumers with tools and abstractions for constructing data pipelines that allow data processing and analysis. In a standard data pipeline, there are several ways to run code, sequence code to execute sequentially, and establish dependencies inside the pipeline and on the environment. Snowflake pipelines are structured utilizing the concepts of tasks and streams. A pipeline enables developers to design a sequence of data processing operations. A task is a logically atomic data process. The other concept of a stream enables intelligent data processing applications by triggering data processing in response to changes in the data landscape.

3.7 Data Retention and Backup:

When dealing with data , weather it's large or small, it is important to take the backups of the data as the loosing of the data could be catastrophic for an organization. With cloud computing in the picture it adds up to the cost substantially, specially when dealing

with the bulk data. Taking the backups of the entire data-set isn't the ideal way and I would be looking into how this can be done in the SaaS application.

3.8 Access and Security management:

Securing data access is critical for any data analytics system. This security consists of two components: authentication (allowing a person to login) and authorization (that is, what objects a connected user has access to). Snowflake enables both discretionary and role-based access control via pre-defined and bespoke roles. Jignesh et al. (2019) demonstrates the encryption techniques that could also be considered for the secuirty of data in cloud.

3.9 Performance and cost management:

After setting up the pipeline for data ingestion and analytics, it is preferable to optimize the model so save resources. In this case by resources, my emphasis is on CPU, time and cost. As part of the research, I would be looking into all the possible techniques by which we can save the resources.

4 Implementation

For the implementation of this research, I have made trail accounts on snowflake, AWS, GCP and Azure. Since I am using the cloud services for only the storage, the credits were enough for the research. I have used Business Critical version of snowflake as it offers all the features.

4.1 Data extraction from cloud

In this research I have loaded the data from cloud storage's such as Amazon S3 bucket, Azure Blob Storage, and GCP and the details steps are mentioned in the configuration manual along with the code snippets. To load data from cloud storage, we'll leverage Snowflake's external stage idea.

There are two ways to import data from cloud storage. The exterior stage procedure is advised. An external stage is similar to a virtual stage that exists within Snowflake and references data stored in a public cloud storage bucket. The reference portion is critical since the external stage does not contain any data; rather, it displays everything in the cloud storage.Alternatively, you can load data from cloud storage by directly referencing it in your COPY statements. Figure 1 shows the architecture to load the data from cloud storage using external stage.

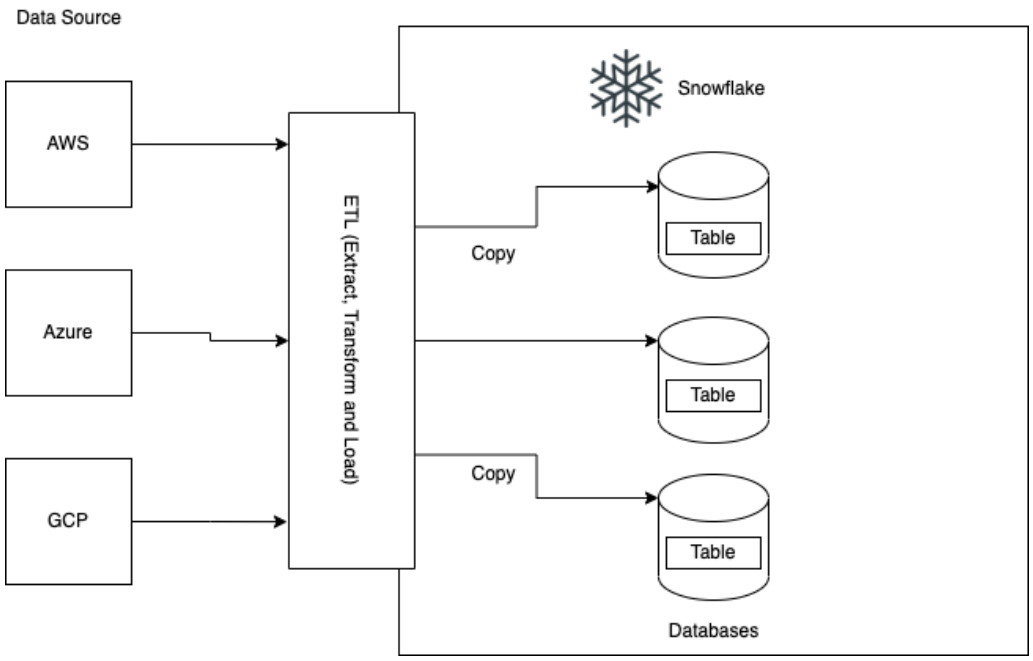

Figure 1: Loading from cloud storage via external stages

To load files from public cloud storage, we begin by designing a file format that specifies that we are loading data in CSV format with an optional header. Following that, we've constructed a stage pointing to s3:/location. Due to the fact that the stage points to public cloud storage, it functions as an external stage. Prior to loading, we attempt to create a list of the files contained in the external stage. If the process succeeds, Snowflake will be able to access our cloud storage data through the external stage. Once everything is configured correctly, all that remains is to execute the usual COPY command to copy the data from the external stage to the target table.

4.2 Copying data into tables:

After the data is loaded into the external stage, it needs to be copied to the tables. For that purpose we need to create file format that specifies the instructions to read the contents from the file. Here we specify the file format like csv or json.

For this project i have uploaded 14.33 GB of data on each cloud storage and have accessed them from snowflake.

Once the file format is specified, we need to create the tables where data would be loaded. We create the SQL statements to create tables and then we use the "COPY INTO" commands to copy the data from the external stage to the tables.

Once the data is copied into the tables, the CRUD operations are performed to extract the meaning-full operations on the data. While doing the crud operations we can see how fast snowflake is when dealing with big data.

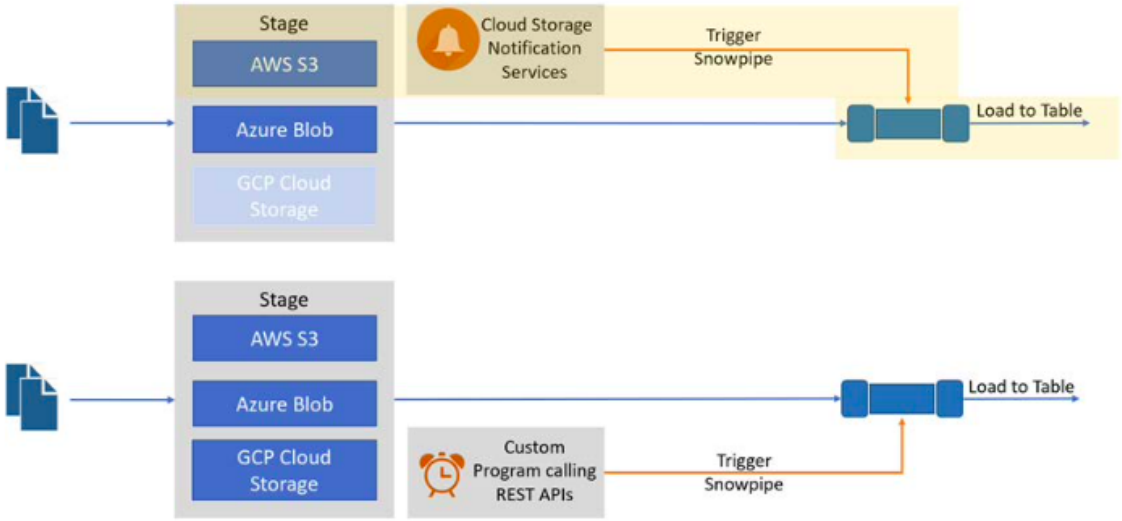

Figure 2: The two methods to trigger Snowpipe

4.3 Real-Time data processing using snowpipe:

Snowpipe enables data to be loaded from files immediately upon their availability in a stage. This enables you to load data from files in micro-batches, making it immediately available to users, rather than manually performing COPY statements on a scheduled basis to load larger batches. Snowpipe automatically loads data from files when they become accessible in a stage. The data is loaded in accordance with the COPY statement specified in a linked pipe.

Snowpipe takes use of the COPY statement, which simplifies the setup process. The majority of the commands and syntax that are compatible with the COPY command are also compatible with Snowpipe. To invoke a Snowpipe, however, extra setting is required. There are two methods for initiating the Snowpipe and thereby loading the data in the S3 bucket. You can either initiate the Snowpipe via a REST API call or configure automatic Snowpipe triggering based on the signal received by cloud storage systems. Each time a new file is created in the cloud storage bucket, we can configure an event to be generated. Figure 2 shows the two methods aforementioned.

The steps for creation of snowpipe are mentioned below (Note: The commands are mentioned in the configuration manual in detail):

- Create a database in which our goal table, Snowpipe, and the stage objects will be stored "CREATE DATABASE Snowpipe;".
- Create the target table into which the data will be loaded via Snowpipe.
- Configure an external stage pointing to the S3 bucket you wish to use.
- Run LIST on the stage to check that the related S3 bucket can be read successfully. There should be no errors returned by the LIST command.

- Create a Snowpipe to facilitate data streaming.
- Although the Snowpipe is established, it will not begin to load data until it is manually activated via a REST API endpoint or until the cloud platform provides an event that can initiate the Snowpipe. Before we move to the AWS console, there is one step that needs to be completed. SHOW PIPES and copy the ARN value displayed in the notification channel column. This ARN value will be used to configure event notification in AWS.
- Proceed to the AWS console to configure an event notification for the S3 bucket, which will trigger the Snowpipe automatically when a new file is created in the bucket. Select the Properties tab for your S3 bucket by clicking on it, then clicking on Events within the tab. On the Events screen, click Add Notification.
- Enter a name for the event on the page for creating new events, such as Trigger Snowpipe or something similar. Select All object creation events in the Events panel, which indicates that the event will be fired whenever a new object is created. Select SQS Queue from the Send to section, click Add SQS queue ARN, and paste the ARN obtained in step 6 into the SQS queue ARN field.
- Continue by saving the event and adding a data file to the S3 bucket.
- Wait a moment, and then conduct a COUNT(*) query on the table via the Snowflake web UI. You will notice that the table has been updated with fresh data.

4.4 Building data pipelines in snowflake:

In this section, the implementation steps for building tasks and tasks and streams and how to use them together to handle complex data processing scenarios.

Snowflake tasks enable the execution of SQL statements on a scheduled basis and can be used to perform mundane activities such as transforming a base table to an aggregate, altering data for reporting, or even processing staged data. When defining a task, you must supply a virtual warehouse name since this will be used throughout the scheduled execution of the operation. Currently, jobs cannot be launched directly; it must always be added to the schedule, which may be programmed to run every x minutes or according to a more elaborate CRON schedule.

Due to the fact that tasks are created in a dormant state by default, they must be resumed before they can be scheduled to run. Only the ACCOUNTADMIN role or another role with the EXECUTE TASK privilege may continue a task. Snowflake schedules the job's execution once it is resumed, and the task history table function allows you to explore the work's execution history.

Additionally, we will connect numerous tasks in a tree structure to create a data pipeline that performs numerous functions as it runs.

Snowflake enables the connection of several tasks via a parent-child relationship. This feature enables the creation of pipelines with many execution steps (The steps for execution is described in the configuration manual)

A successor task can be created for a particular job by giving the task's AFTER configuration. The task indicated in the AFTER config becomes the child task of the task provided in the AFTER configuration. This maintains a hierarchy of tasks. Prior to the child tasks being executed, the predecessor job must be completed.

Additionally, we will demonstrate how to combine the concepts of streams and tasks in order to create a planned Snowflake data pipeline that processes only updated data into a destination table.

Here we see how to combine the concept of change data capture via streams with the idea of tasks to automate the processing of detected changes. We began by configuring a stream to collect only inserts into a staging table. The intention is that data from external sources or via Snowpipe might be added into a comparable staging table in a real-world scenario. Then, on top of the staging table, we developed a stream that maintains track of the rows that are being inserted into the table. Given that this is merely a staging table, we'd like to extract data from it and place it into a more permanent table, which we'll refer to as the target table. A job is built that reads data from the stream and inserts it into the target table using SQL. After that, the task is scheduled to run every ten minutes. New rows are parsed and inserted into the target database as they are added to the stream.

4.5 Data Protection and Security in Snowflake

This is the most critical and important part of the data analytics/engineering in snowflake or any other database management system. Data must be secured at any cost so it doesn't ends up in wrong hands. The best way to implement data security is by not giving everyone access to everything and keep it limited. Figure 3 shows the hierarchy of roles in snowflake.

Following are the steps taken to implement role based access in snowflake.

- Creating custom roles and completing the hierarchy of roles.
- Configuring and designating a user's default role.
- Distinguishing user administration from security and role administration.
- Configuring customized roles for the purpose of controlling access to extremely sensitive data.
- Establishing database hierarchies and roles for development, testing, pre-production, and production.
- Protecting the ACCOUNTADMIN role and its users.

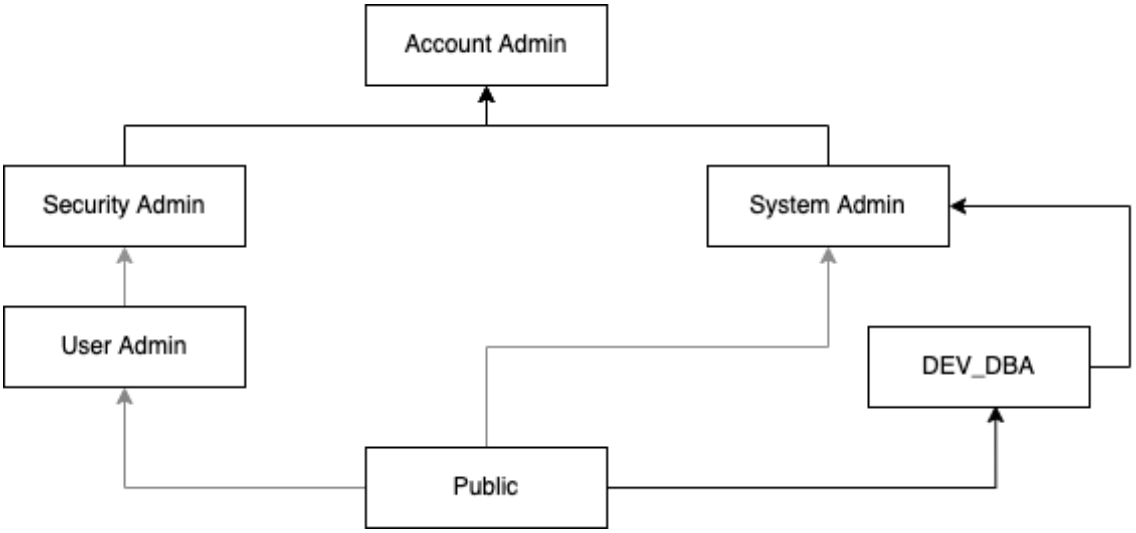

Figure 3: Role hierarchy

The queries for the aforementioned steps are mentioned in the configuration manual.

Snowflake provides industry-leading security precautions to protect your account and users, as well as any and all data stored on Snowflake's infrastructure.Snowflake offers security at various levels and are mentioned below (The features may vary based on the version of snowflake).

- Network/site access.
- User and Group Administration (SCIM to manage user identities and groups (i.e. roles))
- Account/user authentication.
- Security of account objects (Access to all account objects (e.g., users, warehouses, databases, and tables) is regulated by a hybrid model of DAC (discretionary access control) and RBAC (role-based access control).)
- Data security:
  - All ingested data is encrypted using AES-256 strong encryption before being saved in Snowflake tables.
  - Automatic encryption of all files kept in internal stages for data loading and unloading using AES-256 strong encryption.
  - Periodic rekeying of encrypted data.
  - Encryption of data using customer-managed keys is supported.
- Security validations (Few are listed below.):
  - PCI DSS compliance.

- Support for HIPAA compliance.
- Soc 1 Type II and Soc 2 Type II compliance.

Sharing of data in the form of database, tables or views also needs to be in check when dealing with the important data. Snowflake offers techniques to share only what is needed to a specific user without actually sharing all the data. This is different from the user roles, in which the user actions are restricted based on their roles and privileges associated with them. Here the user may have all the access, but only have access to the data that is shared with them.

4.6 Performance and Cost Optimization:

Snowflake has built-in capabilities for optimizing queries and speed through a variety of out-of-the-box features including as caching, auto-scaling, and intelligent table clustering. However, there is always a chance to improve performance by modifying table structures, using physicalization methods, and optimizing your calculations to the utmost. This section will look at some of the approaches that may be used to increase the efficiency and hence the cost of a Snowflake-based data warehouse.

Following are some of the methods that I have used to serve the purpose and are mentioned in detail in the configuration manual.

- **Examining table schemas and determining the appropriate table structure:** Assigning appropriate data types to columns may aid in the table's structure and storage optimization. Using character data types to store numeric and date information consumes more storage and is less efficient during query processing. It is recommended that you accurately input your date and numeric columns. Storage reductions as a consequence might be fairly considerable for big tables.
- **Detecting and resolving query plans and bottlenecks:** It is critical to utilize query profiles to identify time-consuming processes in the execution of your queries.
- **Eliminating wasteful searches through analysis:** The QUERY HISTORY view can be used to discover queries that perform poorly based on a variety of metrics, including runtime, bytes scanned, and so on. Once you've identified the query ID for a problematic query, you can utilize the History tab to open the query profile for that query and examine and fix it.
- **Identifying and minimizing the use of unneeded fail-safe and time travel storage:** Snowflake supports three types of tables: permanent, temporary, and transient. By default, a new table is generated as a permanent table, which includes the Time Travel and Fail-safe features. Both of these functions contribute to storage capacity. When creating a temporary table to store data briefly, it is far preferable to construct it as a transient table, which disables Time Travel and Fail-safe for such tables. It is prudent to monitor the tables in your Snowflake system. Consider tables that have significant values in the Time Travel and Fail-safe bytes columns

but a little value (or even 0) in the active bytes column. In such instance, the table is most likely being utilized as a transient or temporary table and may have been constructed as such.

- **Snowflake-based performance projections:** Snowflake introduces the notion of MVs to facilitate the optimization of various access patterns. MVs enable the separation of table design from developing access patterns.

- **Examining query strategies in order to adjust the grouping of tables:** Snowflake partitions tables automatically, adding new partitions as new data is added to the table. As data is added to the table, the partitions may become no longer infectious, resulting in similar values for a given column being scattered across multiple micro-partitions. The situation described in the implementation distributes the previous 30 days of data over all partitions, necessitating a full table scan every time, even when a WHERE clause is specified. Re-clustering the table redistributes the data so that it is merged into a few partitions for a specific date. When the same query is performed, it now scans a smaller number of partitions, which results in improved query performance.

- **Optimizing the size of virtual warehouses:** Snowflake addresses the issues of large data in a variety of ways. It is scalable horizontally and vertically. Acceptance of a high number of concurrently executed requests necessitates horizontal scaling. A multi-cluster warehouse must therefore be established in this situation. A reasonable inquiry in this scenario would be - how many nodes are necessary in the cluster? The answer is the number of queued or pending requests. If we see that half of the queries entered a waiting state, then the cluster's size should be doubled. It is scalable linearly.

- **Advanced SQL Techniques:** These are the ways by which we can handle the data more efficiently by using advanced technique's. Just like in programming languages we can reduce time and space complexities using several techniques, the same concepts applies here as well. The code blocks are also shared in the artifacts.

4.7 Data Retention:

This is the most useful functionality of snowflake that I came across while working on my research.

Dealing with data problems is never a pleasant task, but it becomes much more so when you cannot tell when the data was modified or whether it was lost entirely. Snowflake's Time Go feature is an amazingly unique method to travel back in time. This chapter discusses the different uses of the Time Travel feature and how it may be used in conjunction with cloning to address typical data loss and debugging difficulties. By default, when you make changes to the data in a table, the previous day's data is kept. However, with Snowflake Enterprise Edition and higher editions, you may adjust the retention period to up to 90 days, enabling you to erase modifications made up to 90 days ago.

Snowflake implements this significant capability, dubbed Time Travel, by maintaining a duplicate of the updated or deleted data for a predetermined length of time. Snowflake retains metadata for each table and object to trace this data copy. When a Time Travel query is executed, the Snowflake query engine may use metadata to seek and retrieve past data. The code of the implementation of this part is shared in the artifacts.

5 Evaluation and Results Analysis

In this section we will look into the results of the research in which we have seen the implementation of processing of big-data by using software-as-a-service (Snowflake).

In this research I have also used services such as Azure, GCP and AWS for the storage of the replica of same dataset. This was done to compare the performance of storage services and what could be the best option for the same.

Some of the finding of my research in relation to the cloud storage for large data-sets are mentioned below.

5.1 Uploading Data-sets:

When comparing the storage containers for data-sets, the Google Cloud Platform (GCP) and Amazon Web Services (AWS) are quiet competent; however, I came across few issues with azure storage. Unlike AWS And GCP, In azure we first need to create a storage account in additional to the regular account, then we need to create a container to store the files and folders. While uploading the files I made on observation that not all files were uploaded. The dataset that I collected has 129 files. While uploading the files, i observed that not all the files get uploaded at once and few files fail to upload. Surprisingly, there was no error message stating the failed uploads, I only came to know after I checked the properties of the folder in azure. At first there were 7 files missing. I tried to upload the remaining files but I had to overwrite all the other 123 files as well. Even after the second attempt the 2 files were missing. In the third attempt, I decided to manually search for the missing two files and I uploaded those to files to the folder. Fortunately, I never faces such issues with GCP and AWS and the uploads were hassle-free.

In terms of accessing the data from Snowflake, there wasn't much difference in terms of speed of accessing the data and were very minute difference and the outcomes were different each time, so probably were dependent on other factors such as the Internet connectivity as well.

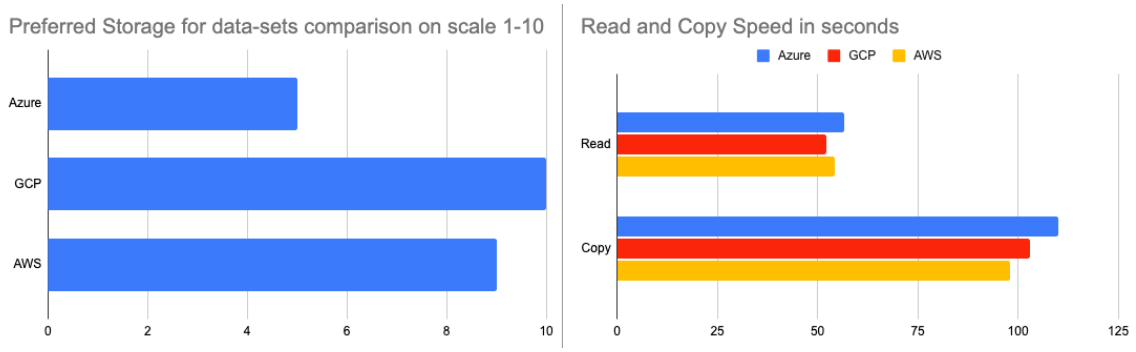

Figure 4: User friendliness and speed comparison.

5.2 Latency and Transfer Speed:

Minimal latency was observed during the implementation on all the 3 cloud storage options. Snowflake was quick to load the data from external stages. To load the entire data-set, the time required was less than 60 seconds. and to copy that data into the snowflake tables, it took about 98-110 seconds. The graphical comparison is shown in Figure 4.

5.3 User Friendliness:

Apart from being efficient and with all the functionalities, It is also important for the service to be user friendly. If the service is very good but it's not user friendly then there won't be many organization or individuals that would opt for that service. I have been using AWS for a while now and have used for numerous applications; however, in this research I have used the AWS, GCP and Azure for the sole purpose of storage the User friendliness has been compared on a scale of 1-10 in Figure 4.

5.4 Performance optimization and cost optimization:

Performance optimization is a very important metric in any field as it saves resources, time and get the tasks done quicker. With the methods mentioned in implementation, I was able to improve the performance substantially. The performance can be also optimized by change the cpu and it scales exponentially. It does adds up the cost significantly, but is helpful at the times when it's important to get the results quicker.

Just like performance, Cost is also a very important factor, and with the techniques that i used in the implementation part, I was able to reduce the expected billing amount substantially lower. With several optimization techniques I was able to reduce the execution time substantially (30-50%). Snowflake itself has it's own mechanisms to optimize the query executions as well.

5.5 Security:

Since security is a very important aspect, I have also focused on several aspects of security in this research paper. It was observed that Snowflake offers security at several layers and that has been discussed in the earlier sections. Finally, it was observed that the security features were up-to the mark and are reliable.

5.6 Data Retention and Backup:

Time travel is quiet a unique feature offered by snowflake and is really helpful, during the implementation, there were numerous occasions when I needed the actions to be reverted. Data once committed or modified in oracle or posgresql cannot be reverted; however, snowflake offers this feature of going back n seconds and get that stage back. This is very critical and useful for organizations as well as individuals.

6 Conclusion and Future Work

In this research paper, we have successfully seen how data analytics can be done using cloud computing and SaaS. The entire infrastructure can be broken into 2 major parts; Data Loading, and Data Processing.

In terms of Data loading, we have seen how the dataset can be uploaded to the cloud storage's such as AWS S3, Azure and GCP and how it can be loading into the snowflake tables using integration and staging. In this part I focused on the two main factors such as User Friendliness and Latency along with the transfer speed. When dealing with large set of data and comparing the performance among the three, I concluded that GCP and AWS were in par with each other. I observed that GCP was more user friendly than AWS and they both were way more friendlier as compared to Azure. When it comes to the performance, I observed that AWS performed better than GCP when copying the data to the snowflake tables which is the most important part and hence, I can conclude that AWS performed best among the three.

After exploring and implementing most of the important features of snowflake application,We have also seen how Snowflake at its advanced features helps access the data quicker. Additionally, there are some more features such as Time travel, Snowpipe, Tasks, Streams, etc. that makes Snowflake stands out and an essential tool for the data analytics. As a conclusion, it is safe to say that Snowflake is a great choice for data analytics in cloud.

In terms of future work, one aspect could be to compare the performance of snowflake with data-bricks and also look into the data visualization aspects of the data and streamlining of the same.

References

Akansha, Gautam, I. & Chatterjee (2020), 'Big data and cloud computing: A critical review'.

Aman & Yadav (2019), 'Security issue in distributedarchitecture of cloud computing in big data aspects', *Journal of Innovation in Computer Science and Engineering* **9(1)**, 53–57.

Anwar Hossain, M. (2019), 'Design and development of a novel symmetric algorithm for enhancing data security in cloud computing'.

Ara, A. & Ara, A. (2016), 'Cloud for big data analytics trends'.

E, Shanmugapriya, R. & Kavith (2019), 'Efficient and secure privacy analysis for medical big data using tdes and mksvm with access control in cloud'.

Farsi, M., Ali, M., Shah, R., Wagan, A. & Kharabsheh, R. (2020), 'Cloud computing and data security threats taxonomy: A review'.

Fuguang, Y. (2019), 'Research on campus network cloud storage open platform based on cloud computing and big data technology'.

Ha-Kyun, Kim, W.-H., So, S.-M. & Je (2019), 'A big data framework for network security of small and medium enterprises for future computing'.

Jignesh, Patel, F., Suthar, S. & Khanna (2019), 'A critical analysis on encryption techniques used for data security in cloud computing and iot (internet of things) based smart cloud storage system: A survey'.

Khedekar, V. & Tian, Y. (2020), 'Multi-tenant big data analytics on aws cloud platform'.

Larrick, G., Tian, Y., Rogers, U., Acosta, H. & Shen, F. (2020), 'Interactive visualization of 3d terrain data stored in the cloud'.

Lidia & Ogiela (2020), 'Cognitive and innovative computation paradigms for big data and cloud computing applications'.

Liu, Qingjie, W., Xiaoying, P. & Zhian (2020), 'Development and application of massive unstructured big data retrieval technology based on cloud computing platform', *Journal of Intelligent Fuzzy Systems* **38(2)**, 1329–1337.

Sangeeta, Gupta, R. & Godavarti (2020), 'Iot data management using cloud computing and big data technologies'.

Suyel, Namasudra, D., Devi, S., Kadry, R., Sundarasekar, A. & Shanthini (2020), 'Towards dna based data security in the cloud computing environment'.

Wan, W., Du, X., Zhao, X. & Yang, Z. (2021), 'A cloud-enabled collaborative hub for analysis of geospatial big data'.

Wang, F., Wang, H. & Xue, L. (2021), 'Research on data security in big data cloud computing environment'.

Xin, Li, J. & Guo (2019), 'Research on ship data big data parallel scheduling algorithm based on cloud computing', *Journal of Coastal Research* **94(sp1)**, 535–539.

Yuqing & Mo (2019), 'A data security storage method for iot under hadoop cloud computing platform'.

Zagan, E. & Danubianu, M. (2021), 'Cloud data lake: The new trend of data storage'.

Zhong, Li, J. & Wang (2019), 'Security storage of sensitive information in cloud computing data center'.