Epidemiology of COVID-19 using Google BigQuery

Dhananjay Narayan (40164521)
Concordia University
Montreal, Canada
dhananjayashwath@gmail.com

Ishika Dhall (40164795)
Concordia University
Montreal, Canada
ishikadhall11@gmail.com

Sarvesh Rayter (40162295)
Concordia University
Montreal, Canada
sarveshrayter0808@gmail.com

Shubham Vashisth (40164794)
Concordia University
Montreal, Canada
shubham.vashisth.delhi@gmail.com

ABSTRACT

The COVID-19 outbreak has affected the lives of billions in numerous aspects. Every day the data related to coronavirus, concerning its social and economic impacts increases substantially. Thus, opening an opportunity for studying the accumulated information using big-data systems. This project, therefore, aims to analyze this big data, comprising the adverse effects of COVID-19 throughout the globe using Google's BigQuery platform. In this report, we discuss the distributed computing aspects of Google BigQuery followed by the discussion on the queried results using the same data system.

1 INTRODUCTION

It comes without a surprise that the regional, as well as global statistics, have played a significant role while encountering the COVID-19 pandemic. The continuous advancements made in the domain of distributed computing and big data have allowed efficient analyzes of such a tremendous amount of data, describing the impacts of COVID-19. Google's BigQuery platform, essentially an enterprise-scale warehousing system, serves as one of the most fitting cloud-based platforms for querying and visualizing large-scale problems such as the COVID-19 analyses. The server less and fully-managed BigQuery platform provides several features that aid in a resourceful, reliable, and user-friendly querying experience to discover the most crucial insights from the data taken into consideration.

This project uses the Google BigQuery system to analyze the COVID-19 Open Data dataset [1] with respect to the number of people tested positive, the number of people getting vaccinated, travel restrictions, drop in mobility, effect on educational institutions, workplaces, and the virus's effectiveness with the variation in temperature and humidity. Another contribution of this report is the discussion on BigQuery concerning its distributed and parallel computing characteristics that are presented in the below sections.

The rest of the report is organized as follows: Section 2 discusses the architecture of BigQuery, Section 3 talks about the distributed design of BigQuery followed by Section 4 which presents the methodology adopted, and Section 5 describes the results. Lastly, the conclusion and future work are presented in Section 6.

2 ARCHITECTURE

BigQuery has a server less architecture that decouples the storage and compute as seen in Figure 1, which are allowed to scale on-demand independent of one another. In a traditional system, both storage and compute are coupled together in the same machine and are a part of the same cluster (on-premises environment). Each node has both storage and compute power. If we are not using the compute power of the node, the cluster would still be running with the same pay scale[2].

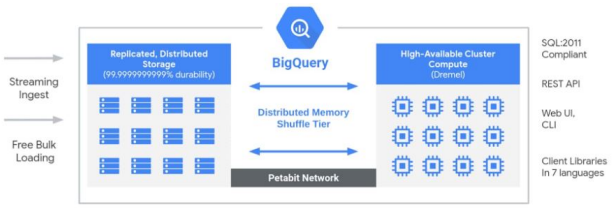

Figure 1: Architecture of Google BigQuery

BigQuery architecture separates the concept of storage and compute. The data can be made available to any compute node very fast using the petabyte network bandwidth of google cloud. The machines can communicate with other machines in the data center over a speed of 10 Gbps[3]. This full-duplex bandwidth means that the locality within the cluster is not important. So in order to just maintain the data, we don't have to power the clusters all the time.

If compute is not being used, we can power it off and just keep the storage nodes running. It significantly reduces the costs incurred as the customers don't have to keep the expensive compute resources running all the time. Owing to this, both storage and compute can be scaled differently, without affecting or slowing down each other. If we are streaming in or bulk loading the data into BigQuery, while parallelly executing heavy queries at the same time, both of the operations can run together without any dependency on each other. This makes BigQuery more economical and scalable.

BigQuery leverages multiple technologies developed at Google, which makes the system more efficient. It is built over the Dremel technology, which is employed in Compute. Dremel is Google's query engine that turns SQL Queries into execution trees that can read data from Google's distributed file system.[4] BigQuery also employs Google's global distributed file storage system, Colossus as Storage. This converts the loaded data into columnar storage. The petabit Jupiter network acts as a bridge for communication between the Colossus storage and the Dremel compute engine, where the Dremel jobs read data from Colossus using this high-speed network. The data moves rapidly from one place to another [2]. BigQuery

makes use of Borg for data processing and fault tolerance. It runs many Dremel jobs over multiple clusters consisting of multiple machines at the same time [5].

3 DISTRIBUTED SYSTEM DESIGN OF BIGQUERY

Whenever dealing with Big data, three major things are to be considered: Variety, Volume, and Velocity. Dealing with a significant amount of data with an extensive variety of data types in several environments, and the rapidity at which the data is generated, collected, and processed can be troublesome. Traditionally, larger sets of data mean longer times between asking a question and getting a solution. It would require proper coordination among the CPUs to commence them at the same time and get an equal amount of work done by them. In such scenarios, servers are more prone to failures. This is where data warehouses like Big Query come into play.

Big Query has the power to provide its user with a humongous number of resources for dealing with such challenges in data processing with 99.999 percent up-time SLA. Working with Big Query involves three primary parts: Storage, ingestion, and querying. All the data is stored in structured tables and due to Big Query's multi-cloud capabilities, standard SQL can be used for easy querying and data analysis [6]. It uses distributed storage of data with replications. Big Query is integrated with the rest of the data analytic platform from Google which allows the streaming of data into Big Query one record at a time. Data can be loaded using the REST API used in Google. Also, the data queries' access can be shared among other users for their insights on the queries. Big Query is powered by multiple data centers each one with hundreds of thousands of cores, petabytes of data storage (Data QnA enables analysis of petabytes of data via BigQuery) and delivering terabytes of data for networking [7]. It is almost impossible to famish Big Query out of its resources as the users' concurrency needs grow.

3.1 Naming

In BigQuery, the identified resources are datasets, tables and views. To help organize and identify these BigQuery resources, we can add labels to them. Labels are the key-value pairs that can be attached to the resources. Once the resources are labeled, we can search for the resources based on these label values. The resources can also be filtered based on their labels. Labels can be based on teams to distinguish the resources owned by different teams. The labels on a resource must meet certain requirements. Each label should be a key-value pair. The keys must have a minimum length of 1 and can never be empty. The values, however, can be empty. The key for a label must always be unique and must start with a lowercase letter. One resource can have multiple labels[8].

3.2 Distributed File System - Colossus

Google's latest-generation system for file distribution which is used by Big Query is called Colossus. Colossus overcomes the limitations of GFS (Google file system) by providing the users with a file system that can work on a global scale and it handles issues with metadata-sharding as well. It is a memory-based file system and Google provides a dedicated colossus cluster to each of its data centers.[2] Each one of these clusters of Colossus has plenty of dedicated disks to be given to all the Big Query users at a time. Big Query provides commendable performance as compared to many in-memory databases due to its unique architecture and Colossus' ultra fast processing. It can leverage much cheaper yet scalable, highly parallelized, durable, and efficient infrastructure. Another property of Colossus is that it can cope with replications and recovery in case of a disk crash. It takes care of distributed data management to avoid a single point of failure. Big Query powers a columnar data storage format called the Capacitor (ColumnIO). Each field in a table of Big Query is stored in a separate Capacitor file. Hence, the compression algorithm to store data in Colossus enables Big Query to achieve a very high compression ratio making it the most optimal way of reading huge amounts of structured data. The Big Query users can scale to dozens of Petabytes of data storage impeccably without hefty payments or penalties of computing resources.

3.3 Borg

A Borg system is a cluster management system at Google that runs a couple of hundred thousand jobs, from several different applications while achieving high utilization by combining overcommitment, admission control, process-level performance isolation with machine sharing, and efficient task-packaging. It is the orchestration system used in BigQuery to make all parts of bigquery work seamlessly together. Additionally, Borg provides the big query with minimized fault-recovery time as it supports high availability applications.

3.4 Parallel query execution using Dremel

Dremel is a large multi-tenant cluster that executes SQL queries. Dremel runs on top of a distributed file system and it provides a highly available cluster compute[4]. Since Big query works in a decoupled manner for data storage and processing using a very powerful petabyte network. Dremel's way of query execution involves turning the queries into execution trees. This execution tree further has two parts to handle the computations and aggregations called the slots (leaves) and the mixers (branches) of the tree, respectively. The hefty lifting of reading data from storage and any other necessary computation is handled by slots of trees and the aggregation part is handled by the mixers. Big Query provides the core set of features which are offered by Dremel to third-party buyers via the Representational state transfer protocol (REST API), including a command-line interface, access control, a Web-based User Interface, etc while maintaining the unparalleled query performance of Dremel. For using the Dremel jobs, Big Query first takes advantage of Borg for allocating the compute capacity then using Jupiter, the Dremel jobs read the data from Google's file system.

3.5 Fault Tolerance and Replication

Fault tolerance of a system is its ability to continue operating as it's intended to even if a failure takes place. Failures can be broadly classified as soft failure(transient) and hard failure (persistent). Soft failure is a failure where the hardware is not destroyed. For example, power failure or machine crash. Hard failure is more severe where the hardware gets destroyed in scenarios such as floods and earthquakes. In either case, BigQuery should never loose the data[9].

Big Query deploys resources across multiple data centres by default and it considers multiple factors of replication in order to attain optimized maximum data durability and service uptime. Big Query does not support cloud-provided replication of data. In case of a machine-level failure, BigQuery continues operating with just a few milliseconds delay. If a dataset is present only in one region, no other Google cloud-based backup is provided to any other region. It can be said that the regional dataset is resilient to soft failures but not to hard failures. For achieving resilience in hard failures, we need to create across-region dataset copies where recovery assurance is necessary. In a multi-region, data is backed up in geographically separated regions. In this way, we can say that the dataset is resilient to both soft and hard failures.[9] The replication need not be done manually. BigQuery automatically replicates data based on the regions that we choose and maintains a 7-day history of change.

BigQuery system is highly available as it is based on highly replicated fault-tolerant architecture for which the details are not made public. It has a monthly up time Service Level Agreement(SLA) of 99.99 percent [10].

3.6 Security

BigQuery has the same security infrastructure as the rest of Google Cloud's services. The data is encrypted both when in rest and transit [11]. Google ensures several measures to maintain the authenticity, integrity and privacy of the data. BigQuery uses Identity and Access Management (IAM) roles and permissions for access control. When a resource is called, BigQuery requires that the user has valid permissions to access the resource. Permissions can be given to granting roles to a user, group or service account.

Bigquery by default encrypts all the data before it is written to the disk. It is then decrypted when an authorized user tries to access it. Google automatically manages the encryption keys that protect the data. However, we can also use customer-managed encryption keys for encrypting the data. To encrypt the data at rest, Advanced Encryption Standard(AES) is used. If the data is in transit, Google cloud uses Transport Layer Security(TLS). The encryption is done before the data is transmitted and then the data is decrypted and verified on arrival at the destination.

4 METHODOLOGY

(1) Creating a GCP account - Open Google Cloud Console and when prompted to sign in, click on Create account to make a new account. After registration, continue to the cloud console and accept the GCP terms and conditions. The dashboard contains all the necessary information about your GCP account. Now create a new project and then by entering BigQuery in the search bar. We will then be redirected to the Big Query API.

(2) Loading the dataset- There are different ways to load the data into BigQuery. We can batch load a set of records or stream individual records or use third-party applications. We can also use other Google services to ingest data. BigQuery also provides public data sets that can also be used. A public dataset is any dataset stored in BigQuery and is made available for the public through the Google Cloud Public Dataset Program. We can access these datasets and integrate them into our projects and applications.

(3) Running and saving Queries- After loading the dataset, we can start querying the table. BigQuery supports two types of queries. In Interactive queries, the execution of queries is done immediately and in Batch queries, BigQuery queues each query and executes them as soon as the idle resources are available. By default, it runs interactive queries. On selecting "Compose new query", a new editor will open wherein we can write the query and execute it to get the results. We can save the queries by clicking on the "Save Query". We can provide a name to the query and later access it under "Saved queries".

(4) Analyzing and Visualizing results- After executing the query, the results can be visualized by using Google Data Studio which is a Google service for data visualization. Selecting "Explore Data", redirects you to Data Studio where results can be analyzed using different graphs and then save the visualization. An interactive report can be created by adding all the visualizations results of queries visualizations combined into one single report.

5 RESULTS

In this section, all the findings that were received from running all relevant queries related to the analysis of data on COVID-19 will be discussed. All the important attributes of the underlined problem have been taken into consideration and snippets of the outputs that were received are showcased here. The queries used in this project address the main problems and shows the effect of the coronavirus disease on all the affected countries in the world. This section aims to show a variety of analysis like the top 10 countries having the highest number of people diagnosed with the disease, a timeline showing vaccination of people all around the world, travel restrictions applied by the government of different countries and many more.

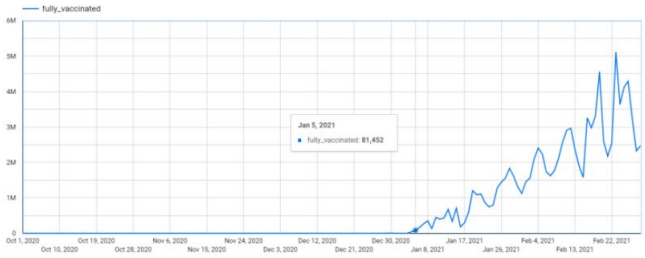

Figure 2: Time-Series graph for the number of people vaccinated from October 20 to March 21

Figure 2 shows the time-series graph for the number of people getting vaccinated during the period of October 20 to March 21. The time series plot shows a drastic shift from December 30 2020 an it kept increasing but soon after as it reached the end of February 2021, a sudden fall can again be seen. From this graph, we can say that the vaccination started in late December 2020 and around mid-February 2021, more than 5M people got vaccinated on a single day.

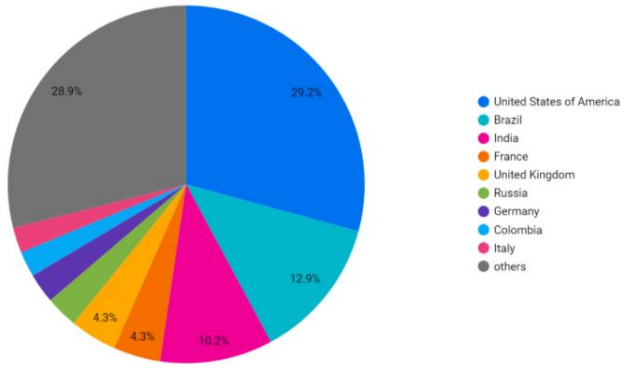

Figure 3: Pie chart of Total New Confirmed cases and percentage of that population

Figure 3 shows a pie chart of new total confirmed cases and the percentage of that population in each county. From the chart we can say that among the sum of new confirmed cases in the world, the majority of them is from the USA with 29.2 percent of total confirmed cases, followed by Brazil with 12.9 percent and India with 10.2 percent.

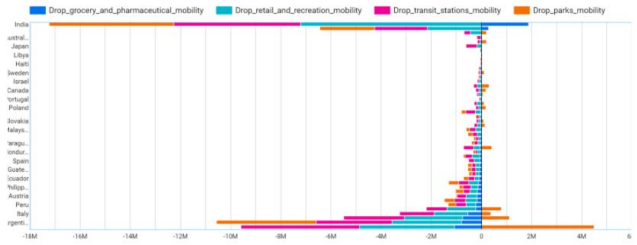

Figure 4: Drop in the mobility of different sectors

Figure 4 shows the drop in the mobility of different sectors during the pandemic. The stacked bar chart shows how the different sectors have suffered a hit due to the pandemic. We can see that India is at the top of the list and has suffered a huge loss in transit stations, retail, recreation and parks mobility due to the lockdown, and on the other hand, it has gained its business in grocery and pharmacy.

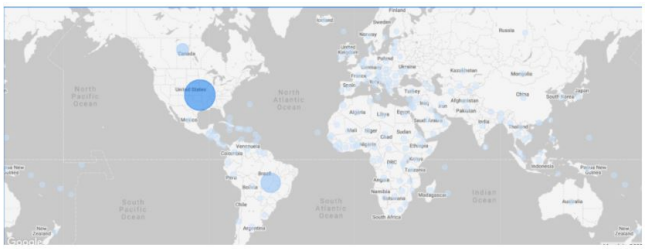

Figure 5: Bubble map showing restrictions on international and domestic travel

Figure 5 shows the restrictions on international and domestic travel in different countries. The Bubble map shows the international and domestic restrictions that were adopted by different countries during the pandemic and we can see from the bubbles that the USA has the most number of international and domestic travel restrictions followed by Brazil.

6 CONCLUSION AND FUTURE WORK

Google's serverless and fully-managed BigQuery platform made it possible to efficiently analyze the COVID-19 Open Data dataset. This enabled the exploration of several areas of society that have been adversely affected by the pandemic. The queried results and their statistical visualization serve as a reliable source to design the counter mechanism for fighting the virus and to perform the necessary resource allocation. This work can be further extended by incorporating the BigQuery ML feature that allows the construction and deployment of several supervised machine learning models as a set of SQL language to make statistically intelligent predictions.

7 REFERENCES

[1] Google Cloud Platform 2021.COVID-19 Open Data. Google Cloud Platform.Retrieved April 20, 2021 from https://console.cloud.google.com/marketplace/product/bigquery-public-datasets/covid19-open-data

[2] Rajesh Thallam. 2020. BigQuery explained: An overview of BigQuery's architecture.Google Cloud. Retrieved April 19, 2021 from https://cloud.google.com/blog/products/data-analytics/new-blog-series-bigquery-explained-overview

[3] Tino Tereshko and Jordan Tigani. 2016. BigQuery under the hood. Retrieved April 19, 2021 from https://cloud.google.com/blog/products/bigquery/bigquery-under-the-hood

[4] Gubarev A. Long J. J. Romer G. Shivakumar S. Tolton M. Vassilakis T. Melnik, S. 2010. Dremel: interactive analysis of web-scale datasets.. In Proceedings of theVLDB Endowment. 330–339.

[5] Panoply .A Deep Dive Into Google BigQuery Architecture. Panoply. Re-trieved April 19, 2021 from https://panoply.io/data-warehouse-guide/bigquery-architecture/

[6] Lopez, G., Seaton, D. T., Ang, A., Tingley, D., and Chuang, I. (2017, April). Google BigQuery for education: Framework for parsing and analyzing edX MOOC data. In Proceedings of the fourth (2017) ACM conference on learning@ scale (pp. 181-184).

[7] Fernandes, S., and Bernardino, J. (2015, July). What is bigquery?. In Proceedings of the 19th International Database Engineering and Applications Symposium (pp. 202-203).

[8] Google Cloud .Creating and managing labels.Google Cloud.Retrieved April 19, 2021 from https://cloud.google.com/resource-manager/docs/creating-managing-labels

[9] Google Cloud.Availability and durability. Google Cloud. Retrieved April 19, 2021 from https://cloud.google.com/bigquery/docs/availability

[10] Google Cloud. BigQuery Service Level Agreement (SLA). Google Cloud. Retrieved April 19, 2021 from https://cloud.google.com/bigquery/sla

[11] Lakshmanan, V., and Tigani, J. (2019). Google BigQuery: The Definitive Guide: Data Warehousing, Analytics, and Machine Learning at Scale. O'Reilly Media.