Project Three Report : BigTable - Google Mail

Team #1

May 5th 2020

CS 464 Intro to Database

TABLE OF CONTENTS

1 Introduction

Cloud Big Table is a massively scalable fully managed NoSQL database service for large-scale applications. A petabyte- scale, low latency, fast, and simple; Big Table is one of the leading database services in the world. It can host enormous applications with the capability to manage, maintain, and expand large amounts of data. Not only is Big Table able to handle so much but it is easily integratable with popular tools that result in a smooth transition to get development teams started.

Gmail's data can be very taxing to maintain and update. With Big Tables' efficient way of storing, allocating, and maintaining data, the possibilities of what can be done may seem endless. Big Tables' storage model is exceptional, allowing us to utilize nodes in a productive manner to retrieve and write into tables that are sharded into blocks to improve the performance of queries. With the amount of data that can be received Balancing the load is essential. Workload is split, signals go off, and data is allocated constantly. Files are being categorized in order for the viewer to have an easy and fast experience with Gmail. Data is always incoming as well as changing, Big Table allows for an easy way to backup data multiple times a day with timestamps in order to have different states of the data. Big Table does not limit developers with its own capabilities but it allows for popular tools to be utilized, whether analytic processing, scalable distributed computing, data processing and much more.

Google's Big Table allows Gmail to store, manage, and rethink what can be done with data. Having an architecture that permits expansive growth while still maintaining the integrity of data is essential in today's market. Big Table fulfills the needs of Gmail.

2 Google Mail : How specific data will be modeled and stored

Bigtable stores data in large tables, each of them are sorted by key and value maps. The table is composed of rows, which usually describe a single entity. Columns which are what *contain individual values for every row. Each row and column intersection contains multiple cells or variants at different timestamps, providing a record of how the data has been altered across time. Cloud-computed big tables are sparse.*

2.1 Storage Model

The table contains a column and qualifiers are used as data. The username is used as a row key. Client requests go through a front-end server before they are sent to the bigtable node. These nodes are organized into a cluster which belongs to a bigtable instance, the container of the cluster. The nodes each handle a subset of requests for the cluster. When adding a node to a cluster, you can increase the number of requests that a cluster can handle which also increases the maximum output of the cluster. By enabling replication with the use of another cluster, you can switch to the other available cluster if one becomes unavailable.

2.2 Storage Sharding

Tables are sharded into blocks of contiguous rows which are called tablets to help balance the workload of queries. These tablets are then stored on Google's file system, "Colossus" in SSTable format. SSTable format is important because it provides a persistent, ordered immutable map from keys to values. Also, all writes are stored in Colossus's shared log when they are acknowledged by Bigtable which provides increased durability.

2.2 Load Balancing

There are zones inside of each cloud bigtable and they are managed by a master process. This master process will balance workload and data volume within each cluster. The master splits the workload by assigning busier or larger tablets in half and merges less-used/smaller tablets together, effectively redistributing them between nodes when needed. If a certain tablet gets a spike in traffic or the performance slows, the master will split the tablet into two and move one of the new tablets to another cluster. Bigtable manages all of the splitting, merging and rebalancing automatically, saving on-site personnel the effort of manually distributing the tablets.

2.3 How Google Mail uses these systems

Many gmail users receive tens to hundreds of emails per day so the priority inbox trys to alleviate such information by learning a user model of importance. It will rank the mail by how likely the user is to act or click on the mail and learn over time. One of the major challenges is accidentally hiding important mail for the user without realizing, so this system is definitely not perfect yet.

3 Utilization of Data

An application like gmail receives a massive amount of information every day. A key aspect of managing a database is being able to organize the data in a way to make it easy and efficient to assess. Working with a system like Big Table leaves us with clear paths in order to organize and access this information. Using NoSQL, the architecture of NoSQL, and many techniques; we are able to manage, maintain, utilize massive information in gmail.

3.1 Big Table Architecture

Cloud Bigtable is what makes gmail run. It's fast, fully managed, and infinitely scalable. Gmail takes in millions upon millions of signals. “Signals include (but aren’t limited to) who the email comes from, what type of content is in the message and how Gmail users have interacted with similar content.[3]” The signals can be a perfect indicator of People you know, promotions, messages from social networks,, updates, and forums.

3.2 NoSQL

Building a database with the main component being the index can be beneficial. As a result the query patterns with a focus on the index can be fast and efficient. Not only is Big Table limited to what it can do but it's able to get extensions in other programs such as Spark SQL which can allow for joins and provisions a Spark DataFrame using relevant ranges from Big table.

3.3 Techniques

With massive amounts of information being stored every second, the architure of Big Table will allow an application like gmail to operate smoothly. Every tuple in this table is sorted by an index. A problem that can arise from the fundamental structure of indexing data is locality of similar types of information. If you want to allocate information of a user within a certain range of indexes this can result in issues. In order to resolve and improve this system, encoding certain aspects of users, regions, or types of data and designating indexes based on the encoding can make it easier to map relations between different tuples in the database. Range Scan can give you information from a starting index to an ending index.

4 NoSQL, its approach and features

NoSQL database provides a way for storage and retrieval of data that is modeled in a way where it's different to its relational database predecessors. NoSQL is mainly used in big data and real-time web apps like social media or email systems. The main drawing point of NoSQL is its simplicity of design and allows for large scale databases that can be scaled to a much larger degree.

4.1 The Approach

NoSQL is used to create large applications that can be scaled to an almost infinite scale. By using clusters and nodes, the data utilization will easily be unstoppable with the correct amount of infrastructure. Each cluster would be located in a single zone. An instance’s cluster package must be in unique zones though. By doing this, you can create an addition cluster at any time in any zone where the cloud bigtable is available. This is how scalability plays a role and why it is so essential for large scale applications. With infinite growth, all you need to worry about is storage which has gotten cheaper over the years. Since storage has gotten cheaper, it has been considered a huge advantage to use NoSQL over traditional means. Storage is one of the only bottlenecks in a NoSQL environment where in a traditional database, relational spaghetti takes place and causes considerable headache for DBAs.

4.2 The Features

- Scalability
  - Bigtable scales directly with the amount of machines inside of the cluster.
  - Cloud Bigtable also does not bottleneck itself when a certain performance threshold is met so you can scale your cluster up to handle a larger influx of read/writes.
- Simplicity
  - Bigtable handles upgrades and restarts autonomously and it maintains high data durability.
  - Replication is easily done by adding a second cluster to an instance, and it automatically starts the process.
  - When you design the table schemas, bigtable will automatically do the rest. Automation is built into this NoSQL architecture.

- Cluster Resizing without maintenance
  - If performance slows, you can increase the size of a cluster for a few hours to decrease the load and then reduce it after the performance stabilizes. This is all done without the need for maintenance.
  - When you do change the size of a cluster, it will balance the performance across all of the nodes inside of that cluster to spread out the tasks.
- Automatic data compression
  - Bigtable will automatically compress data efficiently and effectively when there is low priority for performance.

These are all key features that will help users navigate through an email application. When searching through thousands of emails, the cluster will have to increase in size, balance out the performance and handle the tasks autonomously.

5 Big Table Fulfills Gmail

An email application like gmail, must manage huge amounts of data storage which is necessary to maintain the purpose of the application. With Big table we are able to "scale to billions of rows and thousands of columns, enabling you to store terabytes or even petabytes of data[1]." Not only are we able to store petabytes of data but we are able to read and write at high speeds with minimal latency. In order to maintain an email application like gmail; A lot of infrastructure is required and Big Table fullis the needs of Gmail.

5.1 Demand

An average of about "1.5 billion people use gmail every month,[4]" That is a lot of emails being sent back and forth in a single day. Big Table allows Gmail to sort email based on preferences. With each email that comes in many signals also accompany it. Signals indicate whether the email is to be primary mail from friends or important emails, Social, Promotions, Updates or forums. It Even allows for users to move emails from the folder to another. TensorFlow, a machine learning open source developed at Google, Protects and helps the user navigate possible hundreds of emails a day for one user. An issue that might occur when it comes to sorting incoming emails is sorting out important emails into folders users don't usually check. Big Table also allows retrieval of information to be fast and efficient. Big Table also allows for ease of adding different and new machines to the systems.

5.2 Tools

The architecture that supports Gmail contains a lot of tools that help the administrators navigate and utilize the data in an effective manner. Big Table is sparse, distributed multidimensional sorted map, sharing characteristics of both row-oriented and column-oriented databases. Reading and writing are an essential part of Big Table. There are operations such as put, increment, appdent, conditional updates, bulk import, gets, range scan, filter, and full scan export. All of which are tools to help the programmer understand the data in Big Table. When it comes to the amount of data being updated and changed keeping the data consistent is a difficult job. On one single profile the data can be backed up up to three times per day. “Each row/column intersection can contain multiple cells, or versions, at different timestamps, providing a record of how the stored data has been altered over time[2].” Combining with timestamp causes sequential writes to focus on a single node which preserves the consistency of the data if an issue does happen to occur. Having information is always valuable. Keeping track of clicks is important information for developers to have. With all of these different tools in the system of Big Table, Gmail is best suited for Big Table.

5.3 Expansive

Gmail is continuously growing. The database needs to keep up with constant changes, and updates at an exponential rate. Big Table fullis the needs of Gmail. The scalability and reliability of Big Table is dependable. The simplicity of upgrades and restart autonomously will maintain the data integrity is superb. Not only is maintaining the consistency of data easier but also expanding the space that Gmail will need in the near future. The amount of scalability and high performance makes Big Table a necessity for Gmail.

6 Conclusion

Google’s bigtable is one of the largest structured and defined NoSQL DBMS available right now. It has features that involve record-breaking speeds and performance. It allows for infinite scalability and has high performance autonomous management which lessens the need for DBAs.

Google’s Bigtable has many features and advanced techniques that far surpass other SQL territory. With all of its features and techniques it has amassed over the years of development, it is easily used as a framework for all of Google's services. Over 60 google services use cloud Bigtable infrastructure to house their specified application. With a wide variance in usage and performance needed, bigtable shapes itself to fit the need of any application thrown its way.

With storage becoming cheaper and cheaper, it is more economic to use NoSQL over relational databases due to complexity issues and large data. NoSQL is able to handle large data more effectively by using its built-in performance features such as changing the size of its cluster to overclock the current performance for a couple of hours, or reducing the cluster and taking less resources that could be used in a different cluster. The performance and reliability you get with bigtable, along with all of its features definitely make for the best NoSQL database.

7 Sources

[1]"Overview of Cloud Bigtable | Cloud Bigtable Documentation", *Google Cloud*, 2020. [Online]. Available: https://cloud.google.com/bigtable/docs/overview. [Accessed: 07- May- 2020].

[2]How Gmail sorts your email based on your preferences | Google Cloud Blog", *Google Cloud Blog*, 2020. [Online]. Available: https://cloud.google.com/blog/products/gmail/how-gmail-sorts-your-email-based-on-your-preferences. [Accessed: 07- May- 2020].