RESEARCH ARTICLE OPEN ACCESS

Bigtable: A Distributed Multilevel Map

Amritansh Singh Chauhan [1], Akshita Khatri [2], NikharBhatnagar [3]
Priyanka Sharma [4]
Research Scholar [1] & [2], Assistant Professor [3] & [4]
Swami Keshvanand Institute of Technology
Management &Gramothan, Jaipur
India

ABSTRACT
Bigtable is a compressed, distributed, high performance and proprietary data storage system used for the management of structured data that is designed to scale to a very large size (Petabytes).This database is built on Google File system, Chubby Lock Service, SS Table (log structured storage like Level DB) .This is even now used by a number of Google applications ,such as Web indexing, Map reduce which is often used for
Generating and modifying data stored in Bigtable (Google maps, Google book Search). By introducing this concept of Bigtable (own database) it not only include scalability but even better control of performance characteristics and persistence. The database has successfully provided a flexible, high-performance solution for all of these Google products. The paper has been described by the simple data model provided by Bigtable, its real applications, its design and implementation of the bigtable.

Keywords:- Data models, Real applications, Chubby, pros and con's of Bigtable.

I. INCEPTION

Bigtable development began in 2004[3] .It is designed to scale into the petabyte range across hundreds or thousands of machine and it make easy to add more machines to the system. Bigtable has accomplishd several goals: extensive applicability, consistency, scalability, high accessibility, and high performance. Bigtable is highly used by the evergrowing population which includes the Google projects,Google Analytics, Google Finance, orkutandthe modified search. These products use Bigtable for a diversity of demanding workloads, which range from throughput-oriented batch-processing jobs to latency-sensitive serving of data to end users. Multiple dimensions are assigned to each table (one of which is a field for time, allowing for versioning and garbage collection).The designing is processed by mapping process .It maps two arbitrary string values(row key and column key) and timestamp (hence three-dimensional mapping) into an associated arbitrary byte array.The locations in the GFS of tablets are recorded as database entries in multiple special tablets, which are called "META1" tablets.META1 tablets are found by querying the single "META0" tablet, which typically resides on a server of its own since it is often queried by clients as to the location of the "META1" tablet which itself has the answer to the Question of where the actual data is located. Data is indexed using rows and columns names that can be arbitrary strings. Bigtable also treats data as uninterrupted strings. Clients can control the locality of their data through careful choices in their schemas (bigtable schema parameter).

II. DATA MODEL

The data is sorted, sparse, distributed, persistent multidimensional ,map and time-based which is indexed by a row key, column key, and a timestamp having each value in the map is an associated array of bytes.

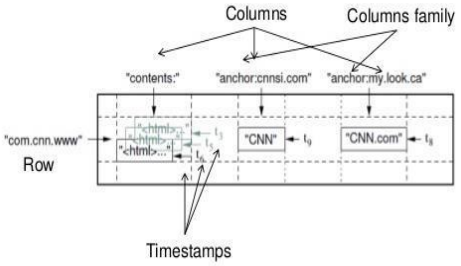

Fig.1 Example of Data-model

The above example shows table that stores Web pages.The row key is the page URL.The column family "contents" has the page contents and the column family "anchor" have the text of any anchors which is referencing the page. Home page of CNN is referenced by both MY-look home pages and Sports Illustrated so the row contains columns named anchor: my.look.ca and anchor: cnnsi.com. Each anchor cell contains one version; the contents columns have three versions, at timestamps t3, t5, and t6

A. Rows

A table is logically split among rows into multiple sub tables called tablets.Each and every row in a table linked with row key of size 64 kilobytes. Every

read or write of data covered by a single row key is atomic.
A tablet is a collection of consecutive rows of a table and it is the unit of distribution and load balancing within Bigtable.Because the table is always sorted by a row, reads of short ranges of rows are efficient: one typically communicates with a small number of machines.

B. Column Families
A column family is a group of columns in a table that are stored as a single key-value pair in the underlying key-value store.Similar type of data is stored in a column family having same properties. First column family is created to store any data. After creating family, we can use any column key within the family. The grouping of data can be done with similar access patterns reduce overall disk access and increases performance.It is our wish that the number of distinct column families in a table be small (in the hundreds at most), and that families hardly change during operation. In contrast, a table can have any number of columns.

The anchor family illustrates the extra hierarchy created by having columns within a column family.
It also illustrates the fact that columns can be created dynamically(one for each external anchor),unlike column families.

C.Chubby
Chubby is used to ensure there is only one active master it is a highly available and persistent distributed lock service that manages leases for resources and stores configuration information.It store the bootstrap location of Bigtable data.It also store Bigtable schema information.

D. Timestamps
Each column family can contain multiple versions of content.Timestamps are of 64-bit integers.They can be assigned by BigTable or client. Either they can be assigned by Bigtable, in which case they represent "real time" in microseconds, or be explicitly assigned by client.Reading column data retrieves the most recent version if no timestamp is specified or the latest version that is earlier than a specified timestamp.A column family can be defined to keep only the latest n versions or to keep only the versions written since some time t.

III. IMPLEMENTATION

Each tablet server manages a set of tablets(in the range of ten to thousands). It handles read and write requests to the tablets it then manages and splits tablets when a tablet gets too large.

In Bigtable clients does not move through the master;they communicate directly with tablet servers for reads/writes.And the internal file is all saved in Google's SSTable which is well ordered and persistent.

A. Tablet Location
A table begins with just one tablet. As the table grows, it is split into multiple tablets .locating rows within a Bigtable are managed in a 3-level hierarchy. The tablet stores the location of all Metadata tablets in a special Metadata tablet.Each Metadata table contains the location of user data tablets.The table is keyed by node IDs and each row identifies a tablet's table ID and end row .A tablet is assigned to one tablet server at a time.Chubby keeps track of tablet servers. As the tablet server starts,it creates and acquires an exclusive lock on a uniquely-named file in a chubby server's directory.

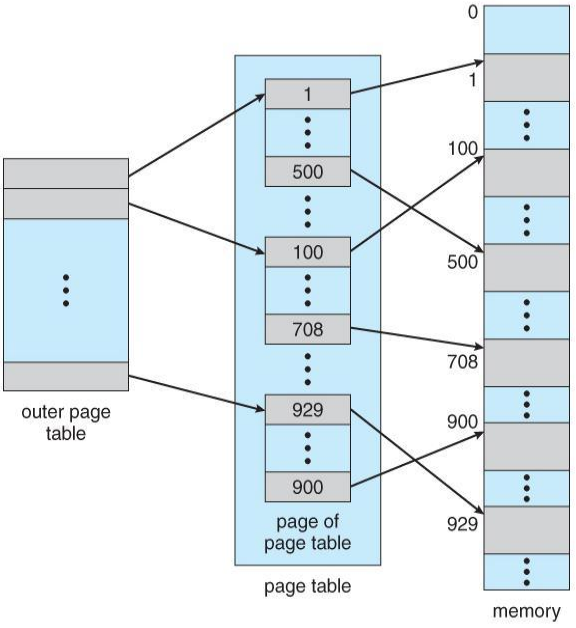

Fig.2 Tablet Location Hierarchy

B. Tablet Assignment
The tablet procedure start by creating and acquiring an exclusive lock on, a uniquely named file when on chubby Master monitors this directory to discover tablet servers.Master is responsible to keep track of the set of live tablet servers, and the current assignment of tablets to tablet servers ,including which tables are unassigned .When the tablet server stops serving the tablet operation it looses its exclusive lock and tries ti acquire the lock on its file as long as the file still exists.And if the file no longer exists the tablet server will never be able to serve again.

C. Tablet Serving

It helps to updates committed to a commit log that stores redo records .The committed updates which were generated recently are stored in memory-mem table. In the given figure 3, the persistent state of a tablet is stored in GFS. The older updates are stored in a sequence of SS Tables.

For the recovering procedure of a tablet, a tablet server reads its metadata from the METADATA table. The operation of read is done to checks the well-formedness of request, and also authorization in chubby file .It then merges memtable and SSTables to find data and then it return data. Operation of write is done at a tablet server, the server checks that it is well-formed and that the sender is authorized to perform the mutation. There is a valid mutation written to the commit log. Group commit is used to improve the throughput of lots of small mutations [4, 5].The write operation writes it to the commit log. After commit,contents are inserted into Memtable.

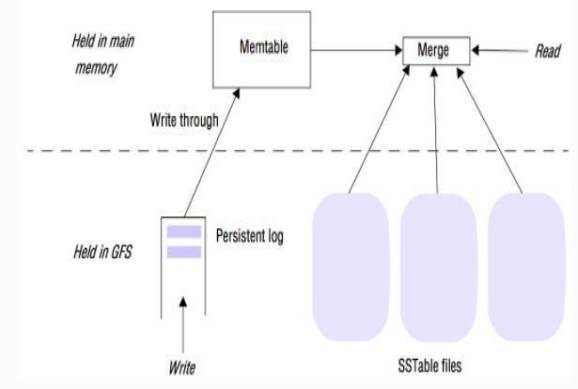

Fig.3.Tablet Assignment of Data

D. Compaction

Bigtable functions periodically it rewrites the tables to remove deleted entries,and to reorganize your data so that reads and writes are more efficient. The control on the size of memtable , tablet log, and SSTable files ,"compaction" is used. It is categorized into three forms.

1. Minor Compaction: It creates a new SSTable. And then it moves the data from memtable to SSTable.
2. Merging Compaction:- It reads the contents of a few SS Tables and the memtable ,and writes out a new SS Table. It merges out with multiple SS Tables and memtable to a single SS Table.
3. Major Compaction: The merging compaction that rewrites all SST ables into exactly one SS Table is called a Major Compaction .It even produces an SS Table that contains no deletion information or deleted data.

4.

IV. REAL APPLICATIONS

The technology which was designed it was intended to be used with petabytes of data. Some of the listed applicationsare:

1. Google Analytics:It is a service that helps webmasters to analyze traffic patterns at their websites. Aggregate statistics are provided such as the number of unique visitors per day and the page views per URL per day, as well as site tracking reports , such as the percentage of users that made a purchase , given that they earlier viewed a specific page. Google has maintained the software as a proprietary, in-house
technology. Bigtable had a large impact on No SQL database design.

2.Google Earth: It is a computer program that renders a 3D representation of earth .It processes a collection of services that provide users with access to high-resolution satellite imagery of the world's surface and its view ,both through the web-based Google maps interface(maps.google.com) and through the Google earth (earth.google.com) custom client software. Users can explore the globe by entering addresses and coordinates, or by using a keyboard ormouse. It provides a series of other tools through desktop application. It allow users to navigate through pan,view and annotate satellite imagery at many different levels of resolution.support inter-cluster replication.Bigtable supports transactions, even cross table transaction. Bigtable index formation is very powerful.Bigtable is only available through PaaS(platform as a service) , like Google App Engine. Bigtable is highly scalable and has been used to host multiple petabytes in a single cell.

V. PRO'S OF BIGTABLE

It is an open source. Also table has replicated storage strategy with algorithms for encryption of data. Big table maintains access control at a column family level. Bigtable contain its own DSL (digital subscriber line) for processing stored data[6]. Correction safety, with the help of CRC checksums. It also support inter-cluster replication.Bigtable supports transactions, even cross table transaction. Bigtable index formation is very powerful.Bigtable is only available through PaaS(platform as a service) , like Google App Engine. Bigtable is highly scalable and has been used to host multiple petabytes in a single cell.

VI. CON`S OF BIGTABLE

Big table is only technically available through google'sPaaS offerings, like Google appengine.It is highly consistent for single-row updates, but offers no consistency guarantees for multi-row updates or cross-table updatesBig table provides a much simpler API than MySQL and Postgre SQL, which leads to no built-in support for secondary indexing and SQL applications are entitled to buildthese features for themselves [7]. The cost is also high to put and gets applications for a big table.

VII. CONCLUSION

Thus we have explained BigTable which is used to store large amount of data by Google. It satisfies goals of high-availability,high performance, massively scalable data storage .It has been successfully deployed in real apps(personalized search, Google maps).It's a distributed systems, designed to store enormous volumes of data. In future we can set up Bigtableas a service to product groups, so that individual groups do not require sustaining their own clusters[8]. As service clusters scale, will must to deal with supplementary resourcesharing issues within Bigtableitself.

REFERENCES

[1] ABADI, D. J., MADDEN, S. R., AND FERREIRA, M. C. Integratingcompression and execution incolumn- oriented database systems. Proc. of SIGMOD (2006).

[2] Chang et al. 2006, p. 3: 'Bigtable can be used with MapReduce, a framework for running large-scale parallel computations developed at Google. We have written a set of wrappers that allow a Bigtable to be used both as an input source and as an output target for MapReduce jobs'

[3] Whitchcock, Andrew, Google's Bigtable, There are currently around 100 cells for services such as Print, Search History, Maps, and Orkut.

[4] DEWITT, D., KATZ, R., OLKEN, F., SHAPIRO, L.,STONEBRAKER, M., AND WOOD, D. Implementationtechniques for main memory database systems. In Proc.of SIGMOD (June 1984), pp. 1.8.

[5] GAWLICK, D., AND KINKADE, D. Varieties of concurrency control in IMS/VS fast path. Database Engineering Bulletin 8, 2 (1985), 3.10.

[6] https://static.googleusercontent.com/media/resea rc h.google.com/en//archive/bigtable-osdi06.pdf

[7] https://www.slideshare.net/tomcythankachan1/g oo gle-bigtable-15370454

[8] http://www.uio.no/studier/emner/matnat/ifi/INF 51 00/h10/undervisningsmateriale/bigtable.pdf