

Google Bigtable

MARK MCCOMBE^{1,*}

¹School of Informatics and Computing, Bloomington, IN 47408, U.S.A.

* Corresponding authors: mmccombe@iu.edu

paper1, March 27, 2017

Google's NoSQL database, Bigtable, is a critical technology in big data for its use internally at Google, as the external service Cloud Bigtable, and for inspiring open source technologies such as Hbase. An overview of Bigtable's storage model and architecture are presented, including available APIs, shell access, and the graphical user interfaces. Performance and security features of Bigtable are discussed along with technologies related to Bigtable. Internal and external use cases involving Bigtable are detailed. Finally, educational resources for learning more about Bigtable are identified.

© 2017 <https://creativecommons.org/licenses/>. The authors verify that the text is not plagiarized.

Keywords: Bigtable, Google, NoSQL, I524

<https://github.com/cloudmesh/sp17-i524/tree/master/paper1/S17-IO-3012/report.pdf>

1. INTRODUCTION

Google Bigtable is a NoSQL database developed by Google, built on several Google technologies, including Google File System, Chubby Lock Service, and SSTable[1]. One of the earliest NoSQL databases, development on Bigtable started in 2004 and Bigtable was introduced to the public in a paper published in 2006 [2]. Bigtable is important to Big Data both for its use internally at Google and externally as Cloud Bigtable, which was made available in May 2015. Google uses Bigtable to power many core Google products, such as Search, Analytics, Maps, Earth, Gmail, and YouTube. [1].

Bigtable has inspired other technologies, notably Hbase [3] an open source distributed, scalable database that was modeled after Bigtable and is typically used along with Hadoop and Hadoop Distributed File System (HDFS) as part of the Apache Big Data Stack.

While Bigtable is a significant technology in Big Data due to its use in Google products and role in the development of other NoSQL technologies, Cloud Bigtable itself is not one of the more popular databases available today. DBEngines ranks Cloud Bigtable only 6 of 9 among Wide Data Stores and 166 of 285 among databases overall for popularity [4].

2. STORAGE MODEL

Bigtable stores data in tables, which are sorted by key/value maps. Tables have rows, typically a single entity, and columns which contain values for the rows. Rows are indexed by a row key. Columns have both a family and a qualifier, which is unique within a family [5].

Figure 1 shows the Bigtable Storage model. It contains four rows (row keys - gwashington, jadams, tjefferson, and wmckinley-

ley), one column family (follows), and four column identifiers (also gwashington, jadams, tjefferson, and wmckinley). Tables in Bigtable are sparse, meaning that a cell will not take up space if it does not contain data (as in the case of jadams/wmckinley). Intersections may contain multiple cells with different timestamps (as in the case of tjefferson/gwashington and jadams/tjefferson) providing a historical record of data in Bigtable [5].

Row Key	Follows			
	gwashington	jadams	tjefferson	wmckinley
gwashington		1		
jadams	1		1	
tjefferson	1	1		1
wmckinley			1	

Fig. 1. Bigtable Data Model [5]

3. ARCHITECTURE

The architecture of Cloud Bigtable is depicted in Figure 2. As shown, client requests come through a front-end server pool and are directed to a Bigtable node (called tablet servers when Bigtable was introduced in 2006). Bigtable nodes are organized into Bigtable clusters, which in turn each belong to a Bigtable instance [5].

Tables in Bigtable are sharded into tablets, which contain blocks of contiguous rows, to balance query workload. Tablets are stored in SSTable format, which provides a map from keys to values,

in Google's file system Colossus and housed in Google's data centers. Each tablet belongs to a specific node [5].

The approach of storing data in tablets rather than rows provides performance benefits and fault tolerance to Bigtable. Because no data needs to be copied, rebalancing tablets between nodes is fast. If a Bigtable node fails, no data is lost and recovery is quick because only metadata needs to be copied to the new node [5].

Bigtable balances data volume and workload across clusters automatically. Bigtable handles this automatically, reducing the administrative effort required [5].

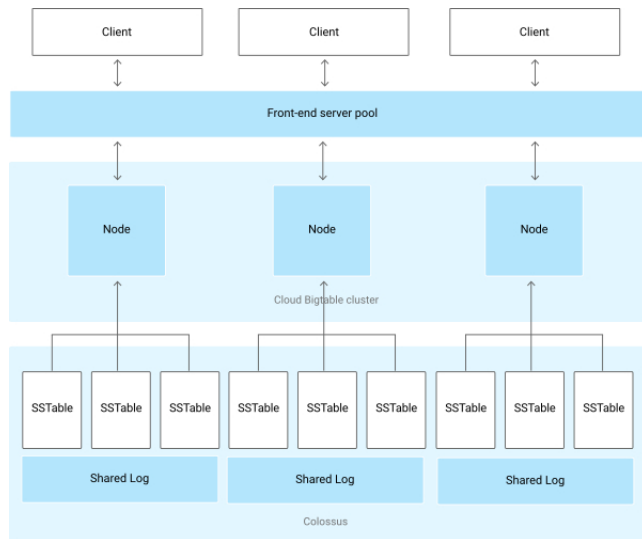


Fig. 2. Bigtable Architecture [5]

3.1. API

Bigtable APIs exist for several languages including HBase Client for Java, Go Client, Python Client, Bigtable-dotnet (.NET), Scio (Scala), and Dataflow Connector (for use in Pipelines) [6].

Bigtable can also integrate with Google Cloud Dataflow, a cloud based, big data programming model, and with Apache Hadoop through the use of Google Cloud Dataproc [6].

3.2. Shell Access

Shell access in Cloud Bigtable can be performed through the HBase Shell [7]. The HBase shell provides a Ruby environment that allows functions in BigTable to be executed on the command line and provides scripting capabilities.

HBase shell commands fall into four main categories [8]. First are general commands. Examples of general commands are version and status. Second are table management commands, which provide functionality to create, drop, and alter tables. The third category is data management commands which include commands like count, get, and delete. The final type of commands enabled by the shell are cluster replication commands, which enable stopping and starting replication and adding and removing peers.

In addition to HBase Shell, cbt, a command line tool written in Go, can be used to perform operations against Bigtable [9].

3.3. Graphical Interface

BigQuery Web UI is a graphical interface, designed to run in Google's Chrome browser, that allows users to interact with

BigTable. Functionality provided by BigQuery Web UI includes the ability to load and export data, to run queries, to create, delete, copy and append to tables, and to view, add, delete and share datasets [10].

4. LICENSING

Bigtable is closed source software and is not available for free use outside of Google. Cloud Bigtable is available for public use on at a cost to the user. The current pricing structure in Figure 3.

Nodes	\$0.65 node/hr (minimum 3 nodes)
SSD Storage	\$0.17 (GB/Month)
HDD Storage	\$0.026 (GB/Month)
Network Ingress	Free
Network Egress	Cross-region and Internet egress rates apply

Fig. 3. Cloud Bigtable Cost Structure [11]

Bigtable has inspired several open source projects which are based on the concepts of Bigtable, notably HBase, Hypertable, and Accumulo (all further discussed in *Bigtable Alternates*). Hypertable is licensed under the GNU General Public License Version 3 while HBase and Accumulo are licensed under the Apache License Version 2.0.

5. PERFORMANCE

According to Google, "Bigtable is designed to handle massive workloads at consistent low latency and high throughput" [11]. While Google does not release performance details of its internal applications, the fact that Google uses Bigtable as the data store for applications that successfully support extremely large data volumes such as Search, Gmail, and Maps supports this claim.

FIS Advanced Technology analyzed the performance of Cloud Bigtable for an application Consolidated Audit Trail (CAT), that will analyze over 100 billion financial market events and store over 30 petabytes of data in the coming years. They reached several conclusions regarding Bigtable's performance capabilities.

- Bigtable was able to handle the demands of the CAT application
- Scaling was linear with clusters of up to 300 nodes
- Data insertion scaled linearly for MapReduce jobs
- No tuning was needed to get sufficient performance from Bigtable

FIS found that Bigtable could write up to 2.7 Gigabytes per second and 10 Terabytes per hour and could process and insert 2.7 million FIX messages per second and 10 billion Fix messages per hour [12].

6. SECURITY

Security in Cloud Bigtable is at the cloud project level. If a user has access to a project, they have access to all tables within the project. Bigtable does not support security at the table, row, column, or cell level [5].

7. RELATED TECHNOLOGIES

7.1. Based on Bigtable

Multiple NoSQL databases have been built based on the specifications presented in the 2006 paper introducing Bigtable. Three important examples are Hbase, Hypertable and Accumulo.

Hbase [3] is the most well known database patterned after Bigtable, HBase is part of the Apache Big Data stack and "provides Bigtable-like capabilities on top of Hadoop and HDFS"[3]. Hbase is written in Java.

Hypertable [13] which is currently sponsored by Baidu, the Chinese search engine, was also inspired by Bigtable's design. It is written in C++.

Accumulo [14] developed by the National Security Agency and contributed to the Apache Software Foundation, extends Bigtable's data model with a new element called Column Visibility. Accumulo is written in Java.

7.2. Bigtable Alternatives

In addition to being a NoSQL database, Bigtable is classified as a wide column store. Popular wide column stores alternatives are Cassandra, HBase, Accumulo, Azure Table Storage, and Hypertable [15].

Bigtable is not well suited for all applications. Google recommends Bigtable for a applications that require "high throughput and scalability for non-structured key/value data, where each value is typically no larger than 10 MB" [5]. Additionally, Google recommends Bigtable for machine learning, stream processing/analytics, and MapReduce operations.[5]

For applications with other needs, Google recommends other databases in the Google suite [5]. For application needing on-line transaction processing (OLTP), Google recommends Google Cloud SQL. For applications requiring online analytical processing (OLAP), Google recommends Google BigQuery. For immutable blobs including images or movies greater than 10 MB, Google recommends Google Cloud Storage. Finally, for structured objects, SQL like queries, and ACID transactions, Google recommends Cloud Datastore.

8. USE CASES

Bigtable is used both internally by Google and externally as Cloud Bigtable by other companies. Use cases of each are discussed below.

8.1. Google Use Cases

Google uses Bigtable internally as the data store for many applications that deal with extremely large data volumes. While Google does not provide the proprietary implementation details of the Bigtable in these applications, their success handling large data volumes is evident. A partial list of applications that utilize Bigtable includes Web Search, Book Search, Search History, Analytics, Maps, Earth, Gmail, YouTube, and Blogger [1].

8.2. External Use Case - CAT Application

As discussed in the *Performance*, FIS Advanced Technology found that Cloud Bigtable was a viable technology for the extreme performance demands of the financial market auditing CAT system [12].

9. EDUCATIONAL MATERIAL

Three key resources exist for learning more about Bigtable. First is the paper *Bigtable: A Distributed Storage System for Structured*

Data introducing Bigtable in 2006 [2]. It contains very detailed descriptions of Bigtable's storage model and architecture. Second is Google's documentation for Cloud Bigtable [16]. The documentation is current and covers all aspects of Cloud Bigtable. Finally, the GoogleCloudPlatform github repository contains many examples of how to use Cloud Bigtable. [17].

10. CONCLUSION

As the the storage layer for Google applications like Search, Gmail, and many others, Bigtable has been one of the most important database technologies in the Big Data Revolution. Based on Google's internal usage of Bigtable and the performance evaluation of Bigtable done by FIS Advanced Technology for the CAT system, it is clear that Bigtable provides excellent performance when processing large amounts of data. With Cloud Bigtable, this performance and scalability is now available to the public.

In addition to Bigtable's own impact, it has inspired other important open source NoSQL datastores, notably Hbase, Accumulo, and Hypertable. These technologies, particularly the Apache Software Foundations, Hbase, have become important players in their own right in the Big Data software stack.

REFERENCES

- [1] Wikipedia, "Bigtable," Web Page, Jan. 2017, accessed 2017-01-29. [Online]. Available: <https://en.wikipedia.org/wiki/Bigtable>
- [2] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, ser. OSDI '06. Berkeley, CA, USA: USENIX Association, 2006, pp. 205–218. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1298455.1298475>
- [3] Apache Software Foundation, "Welcome to apache hbase," Web Page, accessed 2017-02-21. [Online]. Available: <https://hbase.apache.org/>
- [4] DB-Engines, "Google cloud bigtable," Web Page, accessed 2017-02-18. [Online]. Available: <http://db-engines.com/en/system/Google+Cloud+Bigtable>
- [5] Google, "Overview of cloud bigtable," Web Page, accessed 2017-02-15. [Online]. Available: <https://cloud.google.com/bigtable/docs/overview>
- [6] —, "Apis & reference," Web Page, accessed 2017-02-20. [Online]. Available: <https://cloud.google.com/bigtable/docs/apis>
- [7] —, "Installing the hbase shell for cloud bigtable," Web Page, accessed 2017-02-21. [Online]. Available: <https://cloud.google.com/bigtable/docs/installing-hbase-shell>
- [8] Guru99, "Hbase shell and general commands," Web Page, accessed 2017-02-21. [Online]. Available: <http://www.guru99.com/hbase-shell-general-commands.html>
- [9] Google, "cbit overview," Web Page, accessed 2017-02-24. [Online]. Available: <https://cloud.google.com/bigtable/docs/go/cbt-overview>
- [10] —, "Bigquery web ui," Web Page, accessed 2017-02-21. [Online]. Available: <https://cloud.google.com/bigquery/bigquery-web-ui>
- [11] —, "Cloud bigtable," Web Page, accessed 2017-01-29. [Online]. Available: <https://cloud.google.com/bigtable/>
- [12] N. Palmer, M. Sherman, Y. Wang, and S. Just, "Scaling to build the consolidated audit trail: A financial services application of google cloud bigtable," FIS Advanced Technology, techreport, Dec. 2015, accessed 2017-02-224. [Online]. Available: <https://cloud.google.com/bigtable/pdf/FISConsolidatedAuditTrail.pdf>
- [13] Wikipedia, "Hypertable," Web Page, Jan. 2017, accessed 2017-02-24. [Online]. Available: <https://en.wikipedia.org/wiki/Hypertable>
- [14] —, "Apache accumulo," Web Page, Oct. 2016, accessed 2017-02-24. [Online]. Available: https://en.wikipedia.org/wiki/Apache_Accumulo
- [15] DB-Engines, "Db-engines ranking of wide column stores," Web Page, accessed 2017-02-15. [Online]. Available: <http://db-engines.com/en/ranking/wide+column+store>

- [16] Google, "Cloud bigtable documentation," Web Page, accessed 2017-02-12. [Online]. Available: <https://cloud.google.com/bigtable/docs/>
- [17] —, "cloud-bigtable-examples," Code Repository, accessed 2017-02-24. [Online]. Available: <https://github.com/GoogleCloudPlatform/cloud-bigtable-examples>

AUTHOR BIOGRAPHY

Mark McCombe received his B.S. (Business Administration/Finance) and M.S. (Computer Information Systems) from Boston University. He is currently studying Data Science at Indiana University Bloomington.

A. WORK BREAKDOWN

All work on this paper was completed solely by Mark McCombe.