

L^AT_EX Template for Preparing a Paper or Report for I524

JOHN SMITH^{1,2,3}, ALICE SMITH², BRUCE WAYNE¹, AND GREGOR VON LASZEWSKI^{1,*}

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

²School of Science, University of Technology, 2000 J St. NW, Washington DC, 20036

³School of Optics, University of Technology, 2000 J St. NW, Washington DC, 20036

* Corresponding authors: laszewski@gmail.com

project-000, February 25, 2017

This template can be used to prepare a research article for I524. Note that this template can be run from your own T_EX system or within the cloud-based Overleaf system or Sharelatex systems.

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

Keywords: Cloud, I524

<https://github.com/cloudmesh/classes/blob/master/docs/source/format/report/report.pdf>

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]. Development on Bigtable started in 2004. Bigtable was introduced to the public in a paper published in 2006 [2]. Google uses Bigtable internally to power many core Google products, such as Search, Analytics, Maps, Earth, Gmail, and YouTube. In May 2015, a version of Bigtable, Cloud Bigtable, was made available to the public [1].

Bigtable has inspired other technologies, notably Hbase [3] an open source distributed, scalable database modeled after Bigtable 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. DBEngines ranks Cloud Bigtable only 6 of 9 among Wide Data Stores and 166 of 285 among databases overall for popularity [4].

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.

Figure X shows the Bigtable Storage model. It contains four rows (row keys - gwashtington, jadams, tjefferson, and wmckinley), one column family (follows), and four column identifiers (also gwashtington, jadams, tjefferson, and wmckinley). Tables in Bigtable are sparse, meaning that a cell will not take up space (as in the case of jadams/wmckinley) if it does not contain data. Intersections may contain multiple cells with different timestamps

(as in the case of tjefferson/gwashtington and jadams/tjefferson) providing a historical record of data in Bigtable [5].

Row Key	Follows			
	gwashtington	jadams	tjefferson	wmckinley
gwashtington		1		
jadams	1		1	
tjefferson	1	1		1
wmckinley			1	

Fig. 1. Bigtable Data Model [5]

ARCHITECTURE

The architecture of Cloud Bigtable is depicted in Figure X. 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 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, housed in Google's data centers. A tablet belongs to a specific node [5].

The approach of storing data in tablets rather than rows provides performance benefit 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].

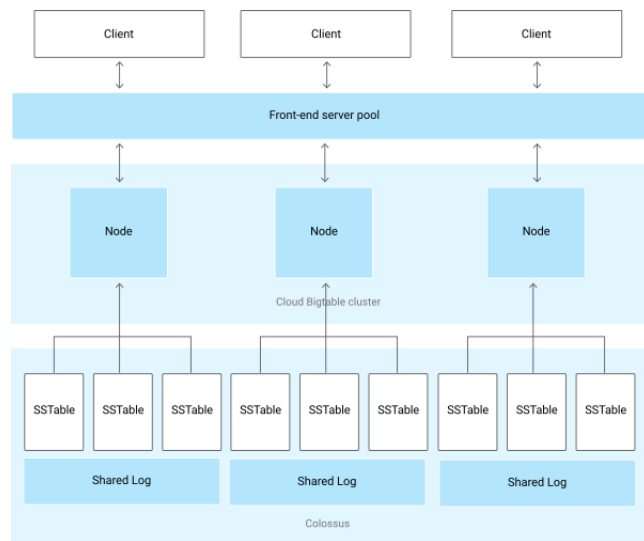


Fig. 2. Bigtable Architecture [5]

API

Bigtable APIs exist for several languages [6]:

- HBase Client for Java
- Go Client
- Python Client
- Bigtable-dotnet (.NET)
- Scio (Scala)
- Dataflow Connector (for use in Pipelines)

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].

Shell Access

Shell access in Cloud Bigtable can be done 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 4 main categories [8].

- General Commands - version, status, etc.
- Table Management Commands - create, drop, alter, etc.
- Data Manipulation Commands - count, get, delete, etc.
- Cluster Replication Commands - start/stop replication, add/remove peer

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

Graphical Interface

BigQuery Web UI is a graphical interface, designed to run in Google's Chrome browser, that allows users to interact with BigTable in various ways [10].

- Load and Export Data
- Run Queries
- Create, Delete, Copy, and Append to Tables
- View, Add, Delete, and Share Datasets

LICENSING

Bigtable is closed source software and is not available for use at no cost outside of Google. Cloud Bigtable is available for public use on at a cost to the user. The current pricing structure is as follows: [11].

- 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

Bigtable has inspired several open source projects based on the concepts of Bigtable, notably HBase. HBase is licensed under the Apache License Version 2.0.

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 its data store for applications that successfully support extremely large amounts of data 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 (GB/Month)
- 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].

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].

RELATED TECHNOLOGIES

Based on Bigtable

Multiple noSQL databases have been built based on the specifications presented in the 2006 paper introducing Bigtable.

- Hbase [3] - 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] - Hypertable, which is currently sponsored by Baidu, the Chinese search engine, was also inspired by Bigtable's design. It is written in C++.
- Accumulo [14] - Accumulo, 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.

Bigtable Alternatives

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

Bigtable is not well suited for all applications. Google recommends Bigtable for 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]:

- Google Cloud SQL - for application needing online transaction processing (OLTP)
- Google BigQuery - for applications requiring online analytical processing (OLAP)
- Google Cloud Storage - for immutable blobs like images or movies greater than 10 MB
- Cloud Datastore - for structure objects, SQL like queries, and ACID transactions.

USE CASES

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

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 is below [1]:

- Search and Search History
- Book Search
- Analytics
- Maps

- Earth
- Gmail
- YouTube
- Blogger

Other Use Cases

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

EDUCATIONAL MATERIAL

There are three key resources for finding information about Bigtable. The 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. The second is the 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].

<https://research.google.com/archive/bigtable.html>

<https://cloud.google.com/bigtable/docs>

<https://github.com/GoogleCloudPlatform/cloud-bigtable-examples>

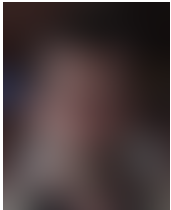
CONCLUSION

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, "cbt 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>

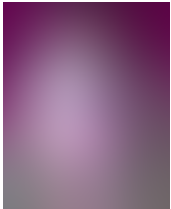
AUTHOR BIOGRAPHIES



John Smith received his BSc (Mathematics) in 2000 from The University of Maryland. His research interests include lasers and optics.



Alice Smith received her BSc (Mathematics) in 2000 from The University of Maryland. Her research interests also include lasers and optics.



Bruce Wayne received his BSc (Aeronautics) in 2000 from Indiana University. His research interests include lasers and optics.

WORK BREAKDOWN

The work on this project was distributed as follows between the authors:

John Smith. Explored the deep mathematical knowledge needed for this paper and taught it to the other authors.

Alice Smith. She explored the world of Oz and was instrumental to work on the deployment of hadoop.

Bruce Wayne. He did not contribute at all to this paper and flew around to save the world.