Google Bigtable

Kasim Mothana
Christian D. Ozoria

What is Bigtable?

* Bigtable is a compressed, highly distributed, high performance data storage system
* Is used by other Google products, including Google Search, Google Analytics and Google Earth, and a part of the Google's Platform as a Service (PaaS)

Why Bigtable?

* Bigtable is primarily designed to scale petabytes of data (for commercial databases such scale is difficult and or expensive to handle) across thousands of commodity machines
  * Scalability is accomplished with the flexibility to add more resources to the system on the fly without the need to reconfigure the system
  * Its model and implementation allow to maintain good performance on such volumes of data
  * The model makes it widely applicable
  * Cost-effective
  * Self-managing

Data Model

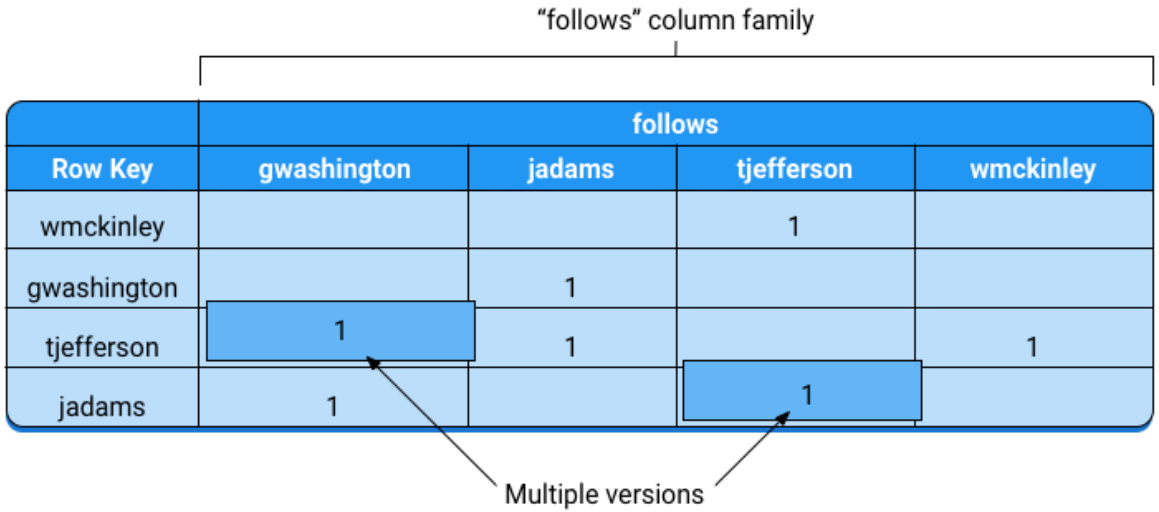
* Bigtable can be loosely compared to a spreadsheet that maintains versions of cells, each with a timestamp. Often it is referred to as a distributed multidimensional sorted map – the map where a row key, a column key, and a timestamp are mapped to a value
* Key model elements:
  * Row key (stored as a string)
  * Column key
  * Timestamp
  * Value that is an uninterrupted array of bytes
* Bigtable is a semi-structured store: for the same row key we can have different columns (similar to the spreadsheet)

Data Model (cont.)

* Storage is organized by row key (ordered alphabetically), column key and timestamp
* Columns are grouped into column families to improve storage characteristics
* Each cell can hold multiple version of data

Example 1

Below is an example of a social network for United States presidents. Each president can follow posts from other presidents. The following shows a Bigtable table that tracks who each president is following on Prezzy:

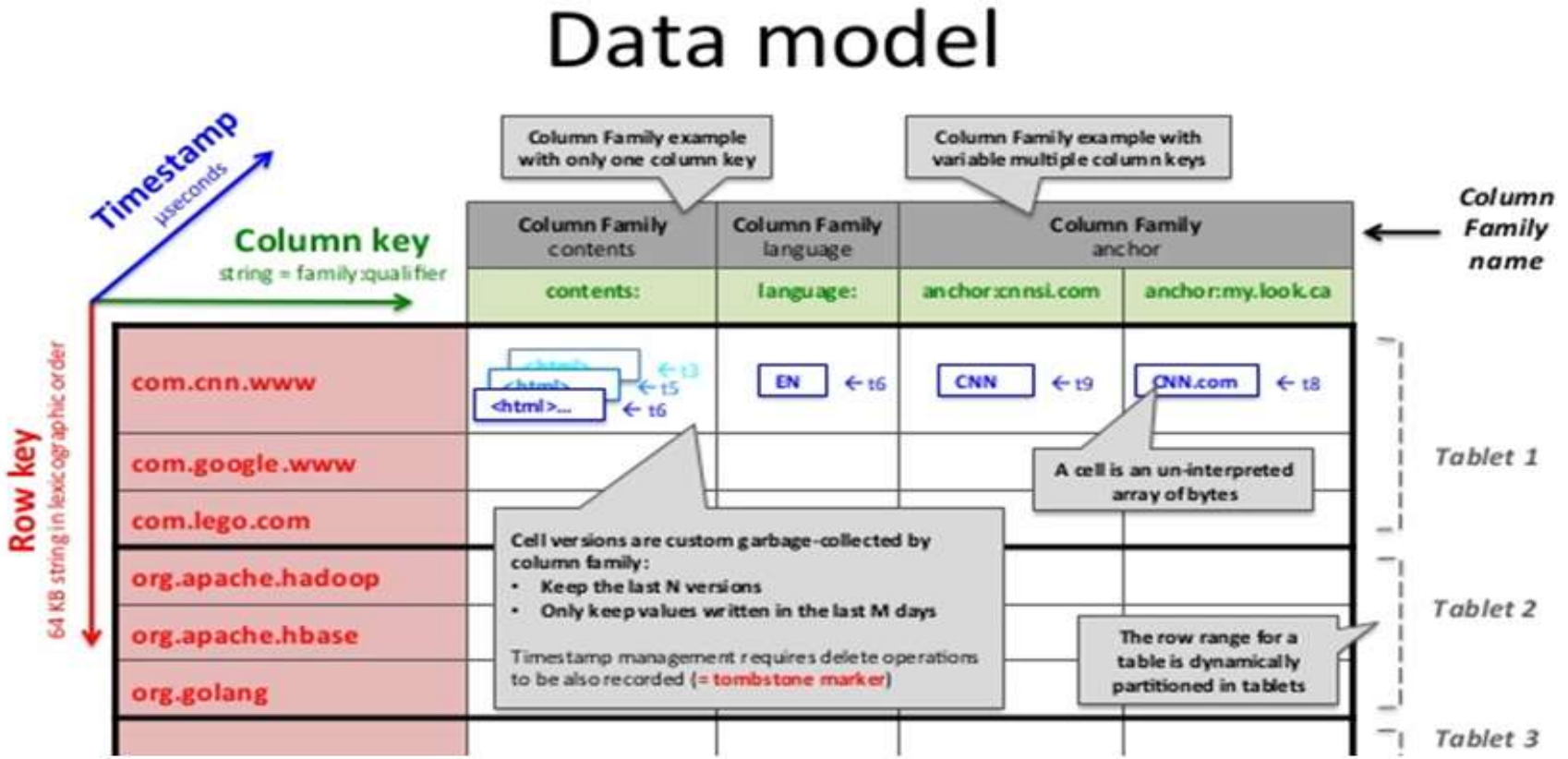


Example1 -- Explanation

* The user name (in this case, the president name) is used as the row key
* This table has one column family containing multiple column qualifiers
* Because new column qualifiers can be added dynamically, it is easy to add new followers
* Illustrates the mapping

  (row:string, column:string, time:int64) → string

Example 2


Data model
Timestamp
μseconds
Column key
string = family:qualifier
Row key
64 KB string in lexicographic order
Column Family example with only one column key
Column Family example with variable multiple column keys
Column Family name
Column Family contents
Column Family language
Column Family anchor
contents:
language:
anchor:cnnsi.com
anchor:my.look.ca
com.cnn.www
com.google.www
com.lego.com
org.apache.hadoop
org.apache.hbase
org.golang
<html>...
t3
t5
t6
EN
t6
CNN
t9
CNN.com
t8
A cell is an un-interpreted array of bytes
Cell versions are custom garbage-collected by column family:
Keep the last N versions
Only keep values written in the last M days
Timestamp management requires delete operations to be also recorded (= tombstone marker)
The row range for a table is dynamically partitioned in tablets
Tablet 1
Tablet 2
Tablet 3

Example 2 – Explanation

* The row key is a reversed URL (explained in the following slides)
* The Contents column family contains the page contents
* The Anchor column family contains the text of any anchors that reference the page
  * CNN's home page is referenced by both the Sports Illustrated and the MY-look home pages, so the row contains columns named anchor:cnnsi.com and anchor:my.look.ca
  * Each anchor cell has one version; the contents column has several versions

Implementation

* The Bigtable implementation has three major components:
  * A library that is linked into every client
  * One master server
  * Many tablet servers

Dynamic Tablet Partitioning

* Data is dynamically partitioned based on the row key; each row range creates a tablet, which is the unit of distribution and load balancing
* Clients can exploit this property by selecting their row keys so that they get good locality for their data accesses. In Example 2, pages in the same domain are grouped together by reversing the hostname components of the URLs: hadoop.apache.com and hbase.apache.com are stored next to each other as org.apache.hadoop and org.apache.hbase

Application: Google Analytics

* Google Analytics is a service that helps webmasters analyze traffic patterns at their web sites
* Tracks website traffic and makes it available to webmasters

Application: Personalized Search

* Personalized Search is a service that records user queries and clicks when using Google search
* Personalized Search stores each user's data in Bigtable. Each user has a unique user id and is assigned a row named by that user id
* Enables a more personalized search experience

Benefits

* Incredible scalability. The Cloud Big Table is designed to scale in direct proportion to the machines in the cluster
* Simple administration. The Cloud Big Table handles upgrades, restarts, and replication transparently

Conclusion

* Bigtable is a distributed system for storing data at Google
* Bigtable clusters have been in production use since April 2005
* In August 2006, more than sixty projects were using Bigtable.
* Users like the performance and high availability provided by the Bigtable implementation

Conclusion cont.

* Users like that they can scale by simply adding more machines to the system as their resource demands change over time
* Several additional Bigtable features, such as support for secondary indices and infrastructure for building cross-data-center replicated Bigtables with multiple master replicas, are in the process of development.

Sources

* Bigtable: A Distributed Storage System for Structured Data http://static.googleusercontent.com/media/research.google.com/en//archive/bigtable-osdi06.pdf
* Under the covers of the App Engine Database.
* http://snarfed.org/datastore_talk.html
* Wikipedia: BigTable
* https://en.wikipedia.org/wiki/BigTable
* Cloud BigTable Beta
* https://cloud.google.com/bigtable/docs/
* Introducing Google Cloud BigTable
* http://googlecloudplatform.blogspot.com/2015/05/introducing-Google-Cloud-Bigtable.html
* CS262B Advanced Topics in Computer Systems. Spring 2009. http://www.eecs.berkeley.edu/~culler/cs262b/summary/bigtable.html
* Tudorica, Bogdan George, and Cristian Bucur. "A comparison between several NoSQL databases with comments and notes." In *Roedunet International Conference (RoEduNet), 2011 10th*, pp. 1-5. IEEE, 2011.
* Lee, Ken Ka-Yin, Wai-Choi Tang, and Kup-Sze Choi. "Alternatives to relational database: comparison of NoSQL and XML approaches for clinical data storage." *Computer methods and programs in biomedicine* 110, no. 1 (2013): 99-109.
* Nayak, Ameya, Anil Poriya, and Dikshay Poojary. "Type of NOSQL Databases and its Comparison with Relational Databases." *International Journal of Applied Information Systems* 5, no. 4 (2013): 16-19.

Sources cont.

* Vicknair, Chad, Michael Macias, Zhendong Zhao, Xiaofei Nan, Yixin Chen, and Dawn Wilkins. "A comparison of a graph database and a relational database: a data provenance perspective." In *Proceedings of the 48th annual southeast regional conference*, p. 42. ACM, 2010.
* Chang, Fay, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. "BigTable: A distributed storage system for structured data." *ACM Transactions on Computer Systems (TOCS)* 26, no. 2 (2008): 4.
* Leavitt, Neal. "Will NoSQL databases live up to their promise?." *Computer* 43, no. 2 (2010): 12-14.
* Stonebraker, Michael. "SQL databases v. NoSQL databases." *Communications of the ACM* 53, no. 4 (2010): 10-11.
* Pallis, George. "Cloud computing: the new frontier of internet computing." IEEE Internet Computing 5 (2010): 70-73.
* Vossen, Gottfried. "Big data as the new enabler in business and other intelligence." *Vietnam Journal of Computer Science* 1, no. 1 (2014): 3-14.
* Zicari, Roberto V. "Big data: Challenges and opportunities." *Big data computing* (2014): 103-128.
* Moniruzzaman, A. B. M., and Syed Akhter Hossain. "Nosql database: New era of databases for big data analytics-classification, characteristics and comparison." *arXiv preprint arXiv:1307.0191* (2013).
* Thankachan, Tomcy. "Google Big Table" November 27th 2012. PowerPoint presentation
* Jacotin, Romain. "Lecture: The Google Big Table" October 9th 2014. PowerPoint presentation