

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is light green. They are positioned diagonally, with the blue one partially covering the green one.

# Cloud BigTable

A large scale Distributed System for  
Structured Data



# Motivation

Built to solve a specific but complex problem.

**How do you store and continuously update petabytes of data, with incredibly high throughput, low latency, and high availability?**

# What is Bigtable

“A BigTable is a sparse, distributed, persistent multidimensional sorted map” [1]

-(row:string, column:string, time:int64) || cell content

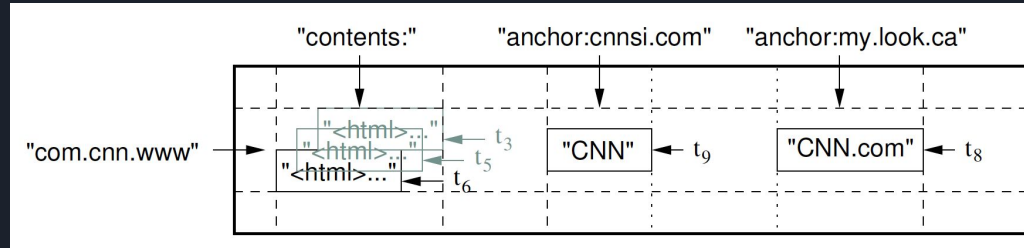


Figure: A slice of an example table that stores Web pages [1]



# Design Goals

The primary use case for Bigtable was the web search index. Bigtable is ideal for applications that need very high throughput and scalability for key/value data. Many of the requirements are related to both performance and scale—which come at the cost of sacrificing many of the nice-to-have features common in modern databases. [2]

- LARGE AMOUNTS OF (REPLICATED) DATA
- LOW LATENCY, HIGH THROUGHPUT
- RAPIDLY CHANGING DATA
- HISTORY OF DATA CHANGES
- STRONG CONSISTENCY
- ROW-LEVEL TRANSACTIONS

# Design Overview

Each row in Bigtable is indexed by its row key. Each cell can contain multiple versions of the same data; these versions are indexed by timestamp.

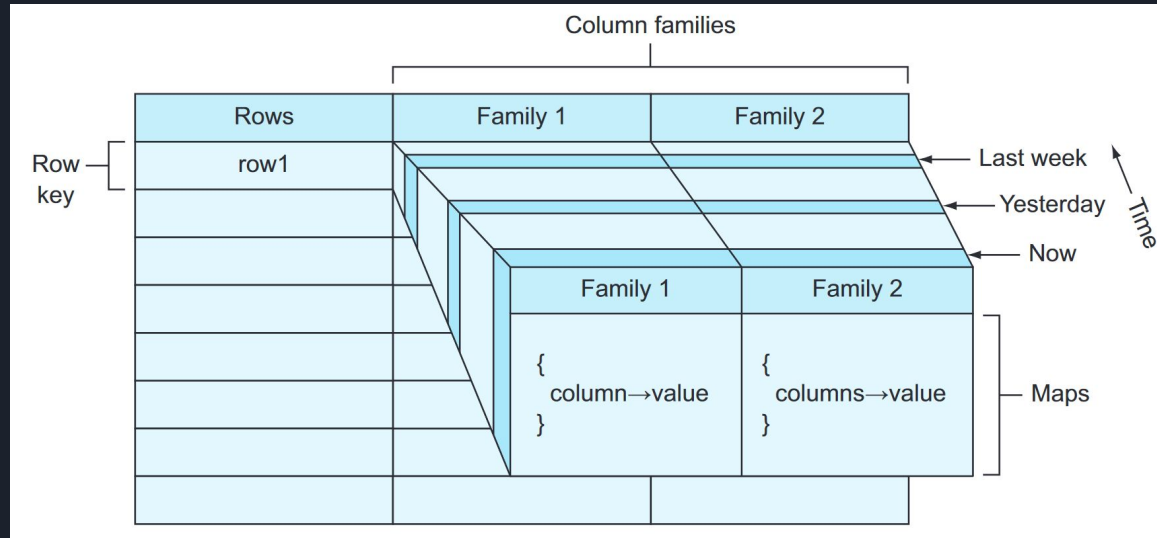



Figure: Bigtable Design Overview [2]



The big key-value store architecture of Bigtable allows it to distribute data across lots of servers while keeping all the keys in that map sorted.

Allows you to do :

- Key Lookups
- Scans over key ranges and key prefixes

Lastly,

**The map is multidimensional.**

**Extra Dimension !! - Timestamp**

Which effectively allows you to go back in time and view data as it was at a previous point. This unique set of features is what makes Bigtable so powerful.

# Infrastructure - Instance

An instance is the container for your data. It has one or more clusters located in different zones, and each cluster contains at least one node.

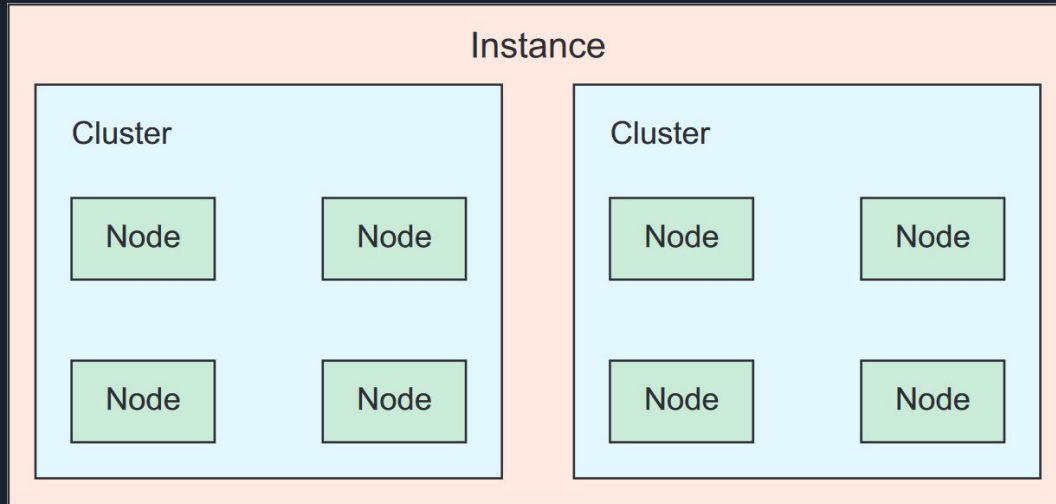


Figure: (Geewax, 2018, 169) Hierarchy of instances, clusters, and nodes

# Infrastructure - Nodes

- Each cluster contains a number of configurable nodes.
- Nodes are computing resources that manage instance data.
- Structure allows to easily balance and redistribute data via tablets

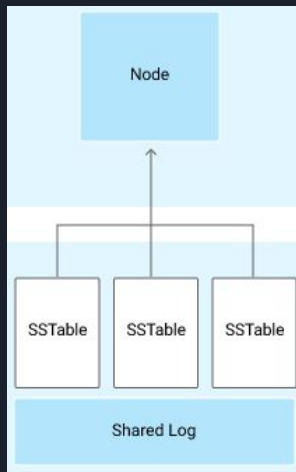


Figure: Node [3]



# Infrastructure - Tablets

Tablets are a way of referencing chunks of data that live on a particular node. Bigtable tablets are stored on the Google file system called Colossus. [3]

Tablets can be split, combined, and moved around to other nodes to keep access to data spread evenly across the available capacity. [3]

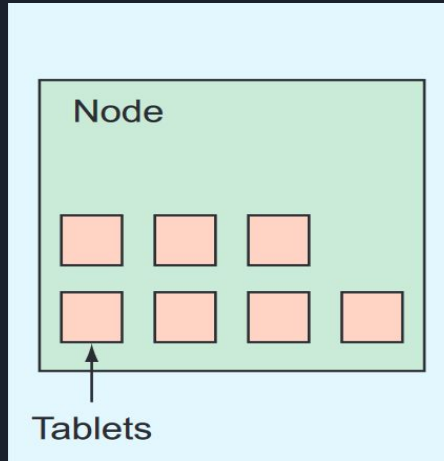


Figure: (Geewax, 2018) Tablets

# Load Balancing

Initially, Bigtable writes data on a single node:

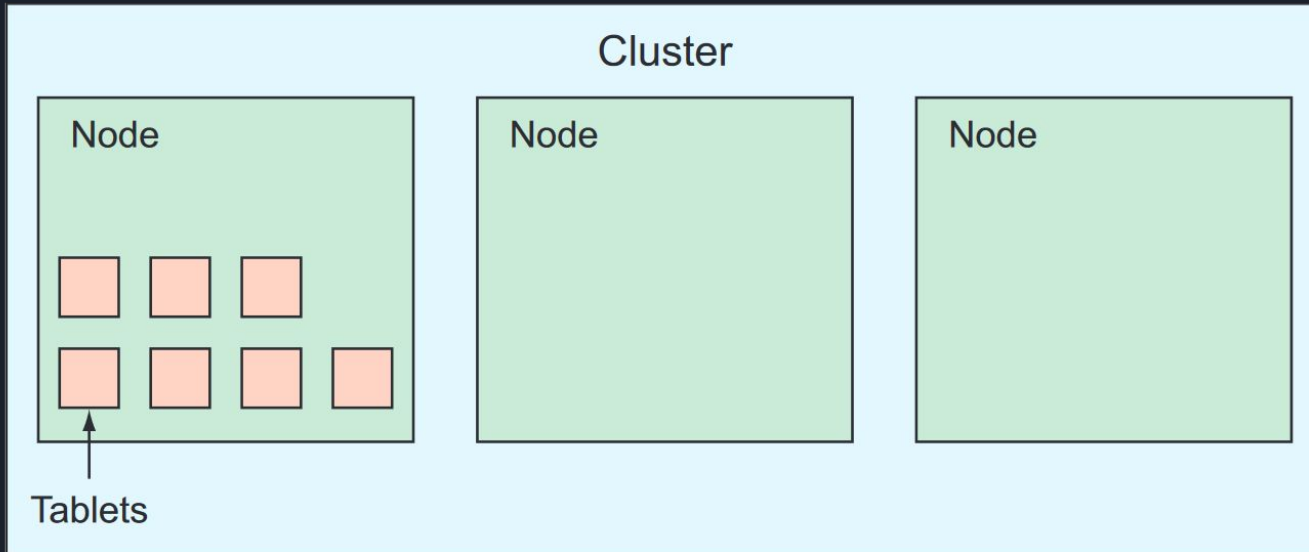


Figure: (Geewax, 2018, 171): Nodes with very high usage will split and redistribute data to other nodes to more evenly spread out compute

Bigtable redistributes tablets to spread data more evenly across nodes

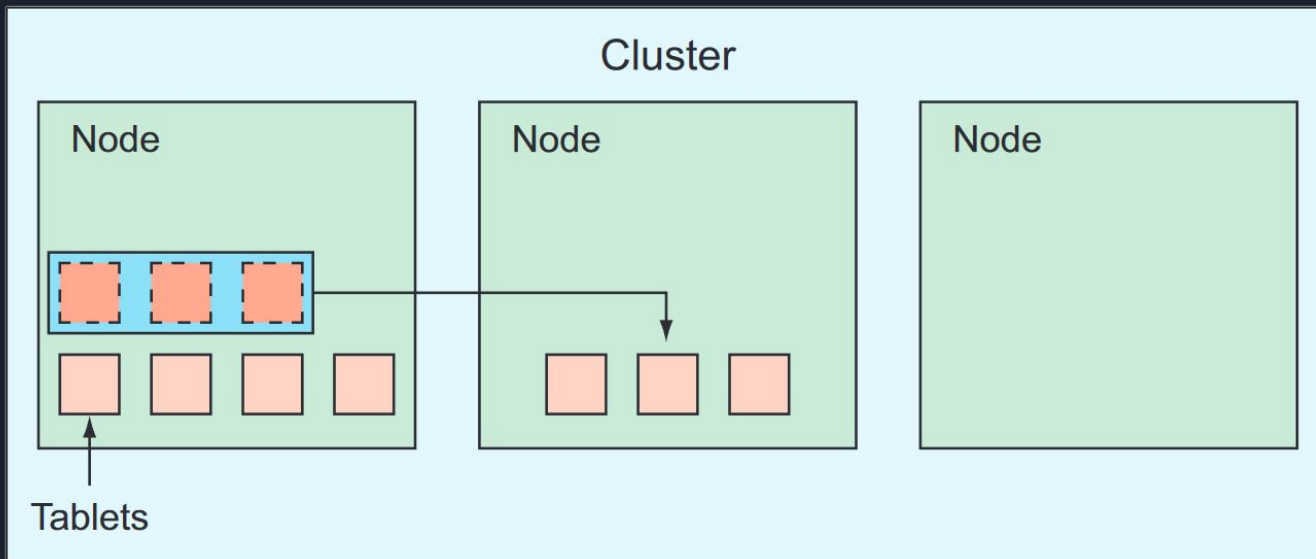


Figure: (Geewax, 2018, 171): Redistribution of tablets among different nodes

It is often the case that some of the data is frequently accessed compared to other. The tablets that stores such frequently accessed data are call hot tablets. In such cases, Bigtable rebalances the load by shifting those hot tablets among other nodes that have more compute availability in order to reduce latency and evenly distribute the requests load.

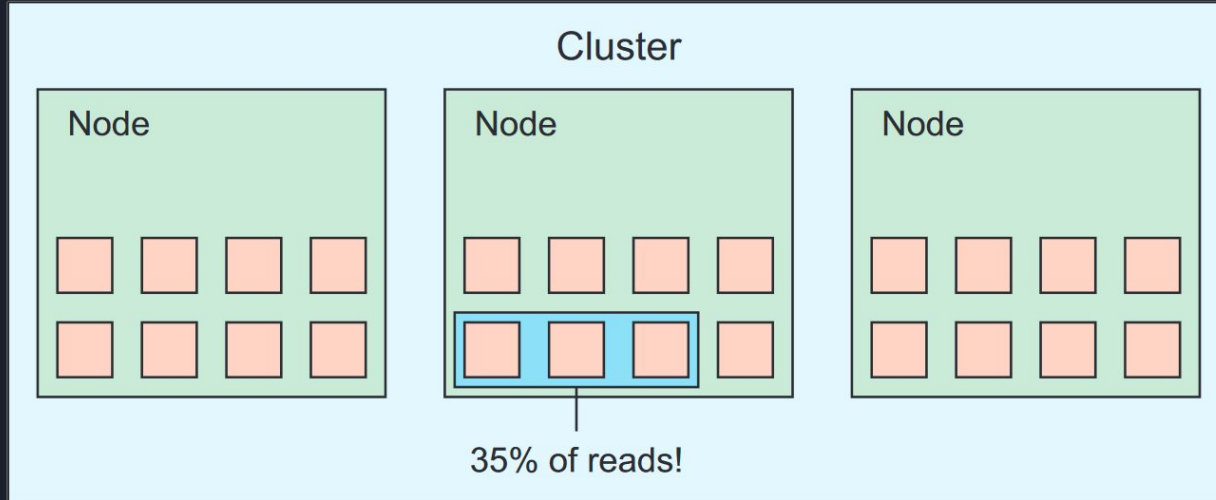


Figure: (Geewax, 2018, 172): Nodes with very high usage will split and redistribute data to other nodes to more evenly spread out compute




# Replication

Replication increases availability and durability of your data by copying it across multiple regions. Bigtable replicates any changes made to your data automatically including the following:

- Modifying existing data in tables
- Adding or Deleting tables and column families

Bigtable treats each cluster as primary, which allows you to perform read and write operations in any cluster. Most common use cases of replication are listed below [4] :

- Improve availability
- Provide near real-time backup
- Ensure data has global presence
- Isolate serving applications from batch reads



## Consistency in Replication

By default, replication for Bigtable is eventually consistent[4].

Although you have to option to choose Read your Writes consistency where all the requests will be routed to one single cluster, while allowing reads from every cluster[4].

Bigtable can also provide strong consistency to ensure that all the applications see your data in same state[4]. This is achieved by choosing Read your Write consistency but restricting all the requests both read and write only towards primary cluster. The other clusters are not to be accessed unless the primary cluster fails for some reason.



# When to use Bigtable

Bigtable comes from performance with both speed and throughput topping the charts.

Aside from the performance, Bigtable acts much like any other key-value store, with almost no structure and little supported query complexity.

Though Bigtable is incredibly powerful, but the lack of common features (such as secondary indexes) tends to be a big drawback. [2]

Due to the absence of SQL queries, joins and multi-row transactions, avoid using Bigtable for an OLAP or OLTP system, instead consider Cloud Spanner or BigQuery as an alternative.




**Whenever you have a large dataset.**

- Typically terabytes or more.
- If the data is only in the gigabyte range (which is typical for a database storing user information), it's better to use other alternatives

**Bigtable is great for usage sustained over a long period of time.**

- A long period of time is measured in hours or days rather than seconds or minutes.
- If Bigtable is used to store and query data only infrequently it's probably better off with some other analytical storage system.





## Extraordinarily high levels of throughput

- Tens to hundreds of thousands of queries every second.
- If the requirement is a few queries per second, it's better to start with another system

You can use Bigtable to store and query all of the following types of data [3]:

- **Time-series data** such as CPU and memory usage over time for multiple servers.
- **Marketing data** such as purchase histories and customer preferences.
- **Financial data** such as transaction histories, stock prices, and currency exchange rates.
- **IoT data** such as usage reports from energy meters and home appliances.
- **Graph data** such as information about how users are connected to one another.



# Summary

- Bigtable is a large-scale data storage system, originally built for Google's web search index.
- It was designed to handle large amounts of replicated, rapidly changing data and can be queried quickly (low latency) with high concurrency (high throughput), while maintaining strong consistency throughout [2].
- Bigtable provides support for automatic load balancing and compaction[3] that is it periodically rewrites your tables to remove deleted entries, and to reorganize your data so that reads and writes are more efficient.
- Bigtable is likely a good fit if you have a large amount of data and primarily access it using key lookups or key scans but not a great fit if you need secondary indexes or relational queries.



# References

- [1] Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., & Gruber, R. E. (2006). Bigtable: A Distributed Storage System for Structured Data. ACM Transactions on Computer Systems, 26.

<https://storage.googleapis.com/pub-tools-public-publication-data/pdf/68a74a85e1662fe02ff3967497f31fda7f32225c.pdf>

- [2] Geewax, J. (2018). *Google Cloud Platform in Action*. Manning.

<https://livebook.manning.com/book/google-cloud-platform-in-action/>

- [3] Google. (2021, March 16). *Overview - Architecture*. Google Cloud.

<https://cloud.google.com/bigtable/docs/overview#architecture>

- [4] Google. (2021, February 18). *Overview of Replication*. Google Cloud.

<https://cloud.google.com/bigtable/docs/replication-overview>