



Partitioning and High-level Architecture

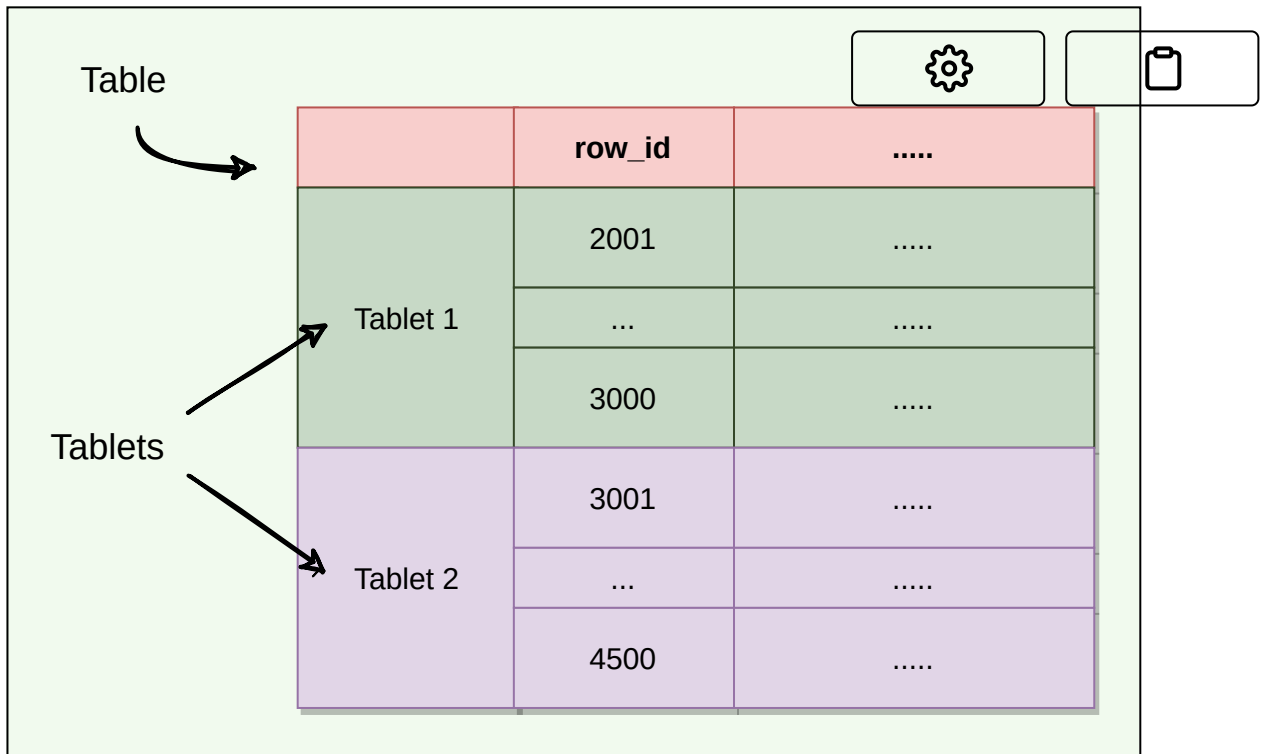
This lesson gives a brief overview of BigTable's architecture and its data partitioning scheme.

We'll cover the following ^

- Table partitioning
- High-level architecture

Table partitioning#

A single instance of a BigTable implementation is known as a cluster. Each cluster can store a number of tables where each table is split into multiple **Tablets**, each around 100–200 MB in size.



- A Tablet holds a contiguous range of rows.
- The table is broken into Tablets at row boundaries.
- Initially, each table consists of only one Tablet. As the table grows, multiple Tablets are created. By default, a table is split at around 100 to 200 MB.
- Tablets are the unit of distribution and load balancing (more about this later).
- Since the table is sorted by row, reads of short ranges of rows are always efficient, that is to say, communicating with a small number of Tablets. This also means that selecting a row key with a high degree of locality is very important.
- Each Tablet is assigned to a **Tablet server** (discussed later), which manages all read/write requests of that Tablet.

High-level architecture#

The architecture of a BigTable cluster consists of three major components:

1. **Client Library:** A library component that is linked into every client.

The client talks to BigTable through this library.

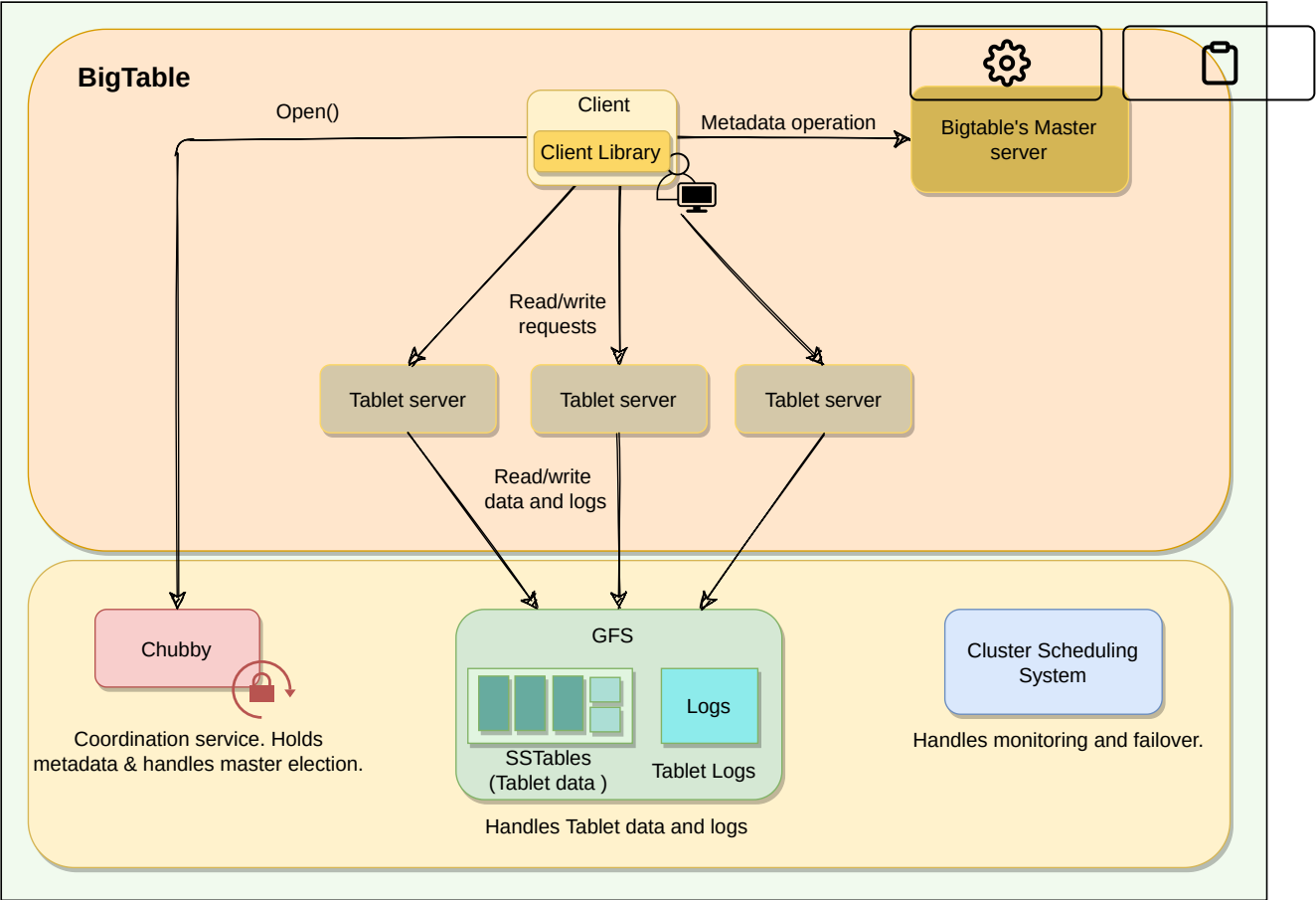


2. **One master server:** Responsible for performing metadata operations and assigning Tablets to Tablet servers and managing them.
3. **Many Tablet servers:** Each Tablet server serves read and write of the data to the Tablets it is assigned.

BigTable is built on top of several other pieces from Google infrastructure:

1. **GFS:** BigTable uses the Google File System to store its data and log files.
2. **SSTable:** Google's SSTable (Sorted String Table) file format is used to store BigTable data. SSTable provides a persistent, ordered, and immutable map from keys to values (more on this later). SSTable is designed in such a way that any data access requires, at most, a single disk access.
3. **Chubby:** BigTable uses a highly available and persistent distributed lock service called Chubby to handle synchronization issues and store configuration information.
4. **Cluster Scheduling System:** Google has a cluster management system that schedules, monitors, and manages the Bigtable's cluster.

Let's understand these components one by one.



High-level architecture of BigTable

← Back

Next →

System APIs

SSTable

☒ Mark as Completed

 Report an Issue