



The Life of BigTable's Read & Write Operations

Let's explore how BigTable handles its read and write operations.

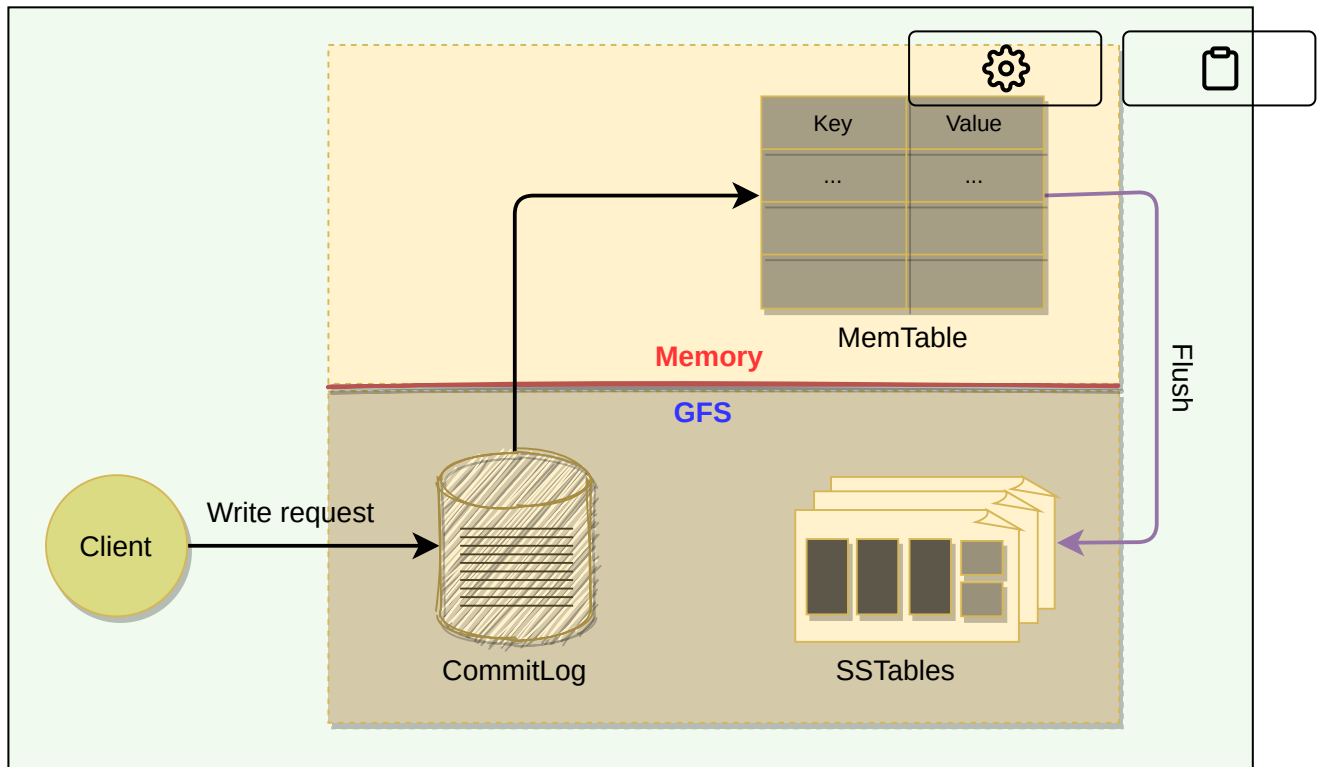
We'll cover the following ^

- Write request
- Read request

Write request#

Upon receiving a write request, a Tablet server performs the following set of steps:

1. Checks that the request is well-formed.
2. Checks that the sender is authorized to perform the mutation. This authorization is performed based on the Access Control Lists (ACLs) that are stored in a chubby file.
3. If the above two conditions are met, the mutation is written to the commit-log in GFS that stores redo records.
4. Once the mutation is committed to the commit-log, its contents are stored in memory in a sorted buffer called MemTable.
5. After inserting the data into the MemTable, acknowledgment is sent to the client that the data has been successfully written.
6. Periodically, MemTables are flushed to SSTables, and SSTables are merged during compaction (discussed later).

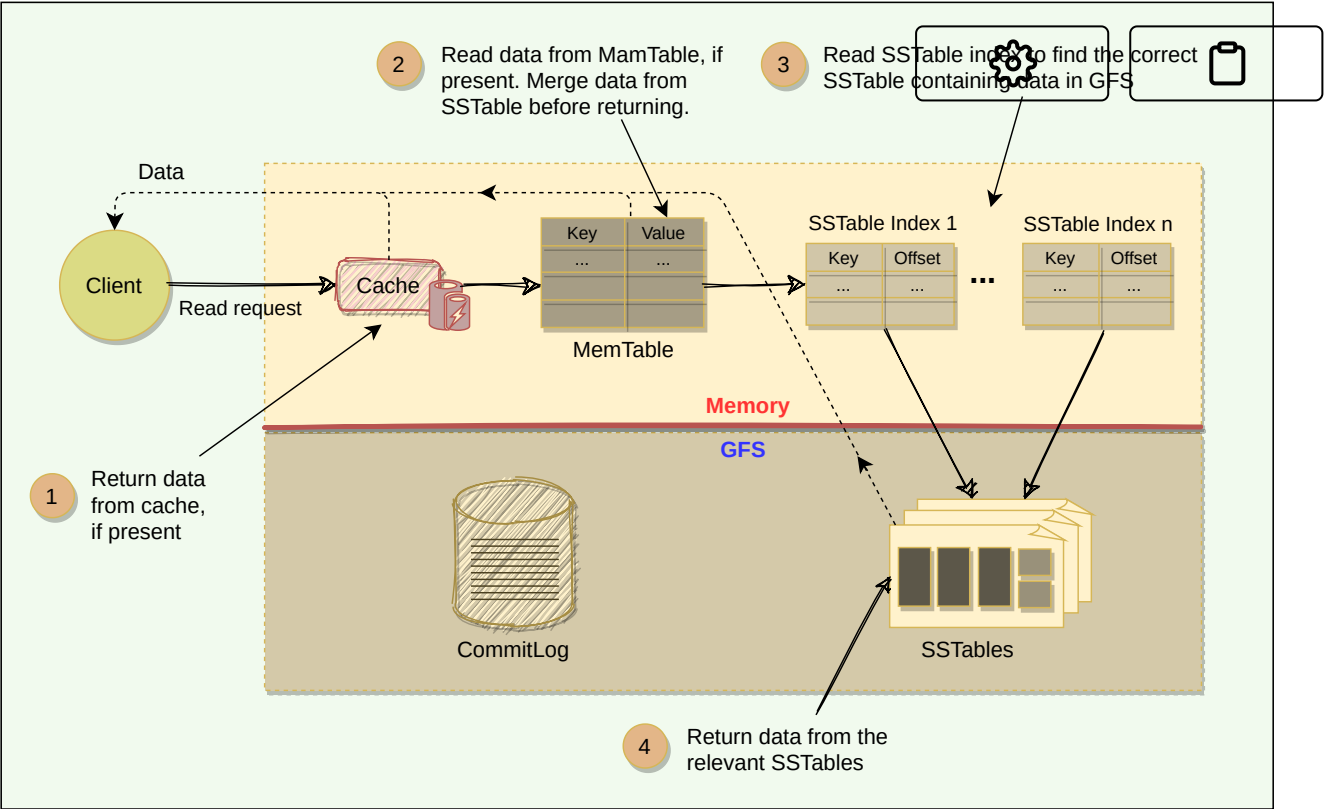


The anatomy of a write request

Read request#

Upon receiving a read request, a Tablet server performs the following set of steps:

1. Checks that the request is well-formed and the sender is authorized
2. Returns the rows if they are available in the Cache (discussed later)
3. Reads MemTable first to find the required rows
4. Reads SSTable indexes that are loaded in memory to find SStables that will have the required data, then reads the rows from those SStables
5. Merge rows read from MemTable and SStables to find the required version of the data. Since the SStables and the MemTable are sorted, the merged view can be formed efficiently.



The anatomy of a read request

← Back

Next →

Working with Tablets

Fault Tolerance and Compaction

☒ Mark as Completed

Report an Issue