Working with Tablets

We'll cover the following

- Locating Tablets
- Assigning Tablets
- Monitoring Tablet servers
- Load-balancing Tablet servers

Locating Tablets#

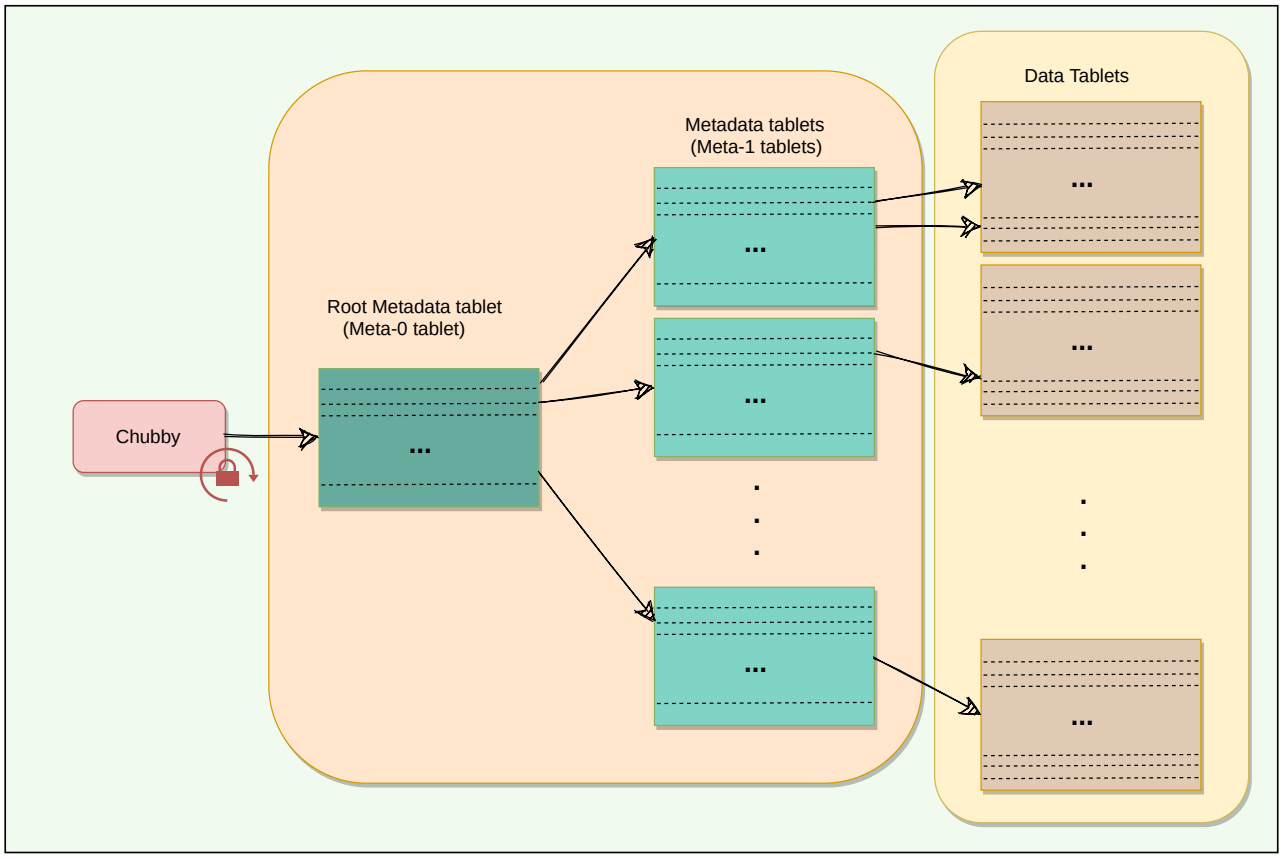
Since Tablets move around from server to server (due to load balancing, Tablet server failures, etc.), given a row, how do we find the correct Tablet server? To answer this, we need to find the Tablet whose row range covers the target row. BigTable maintains a 3-level hierarchy, analogous to that of a B+ tree, to store Tablet location information.

BigTable creates a special table, called **Metadata** table, to store Tablet locations. This Metadata table contains one row per Tablet that tells us which Tablet server is serving this Tablet. Each row in the METADATA table stores a Tablet's location under a row key that is an encoding of the Tablet's table identifier and its end row.

```
METADATA:   Key: table id + end row
          Data: tablet server location
```

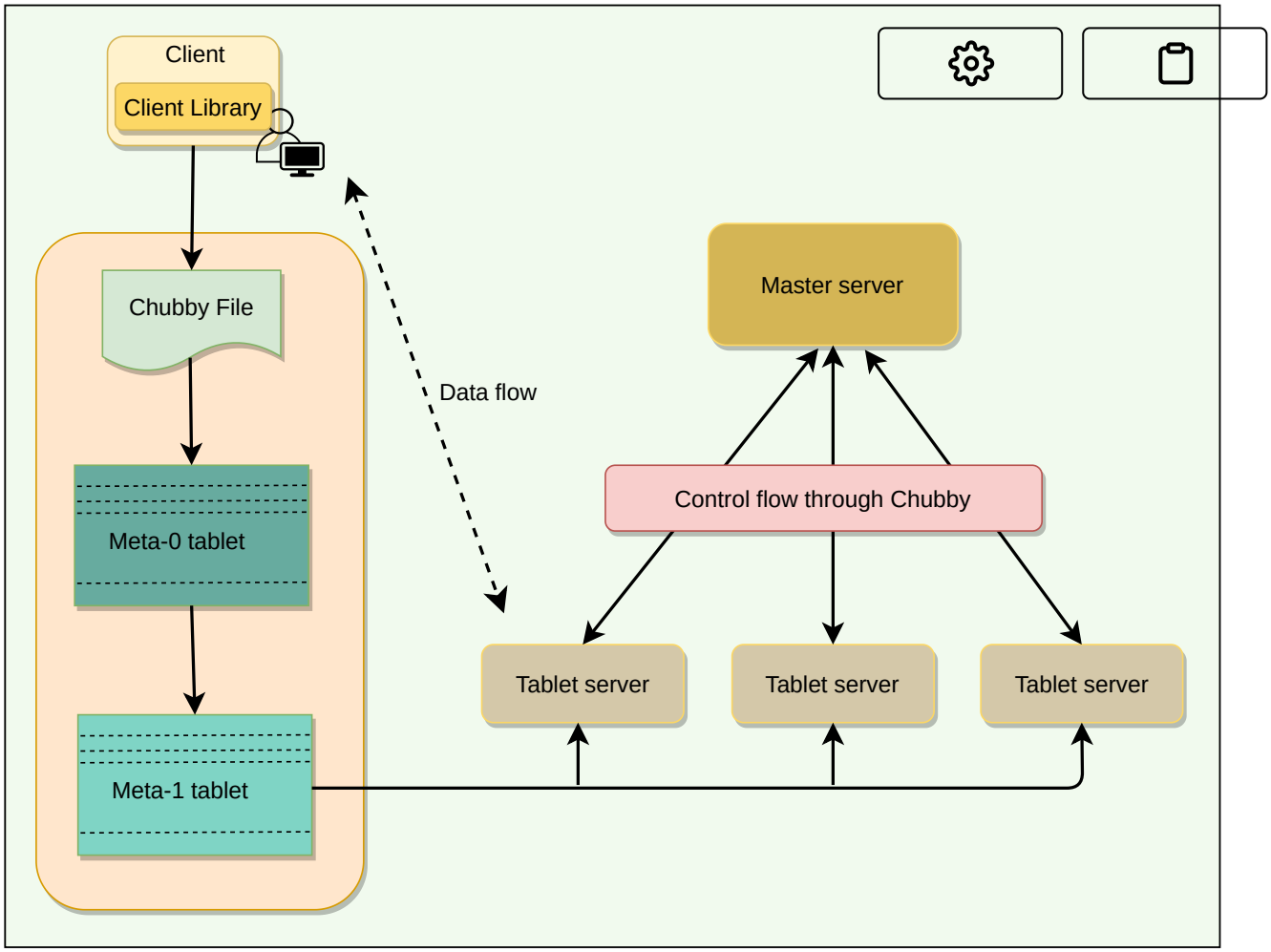
BigTable stores the information about the Metadata table in two parts:

1. **Meta-1 Tablet** has one row for each data Tablet (or non-meta Tablet). Since Meta-1 Tablet can be big, it is split into multiple metadata Tablets and distributed to multiple Tablet servers.
2. **Meta-0 Tablet** has one row for each Meta-1 Tablet. Meta-0 table never gets split. BigTable stores the location of the Meta-0 Tablet in a Chubby file.



Metadata tablets

A BigTable client seeking the location of a Tablet starts the search by looking up a particular file in Chubby that is known to hold the location of the Meta-0 Tablet. This Meta-0 Tablet contains information about other metadata Tablets, which in turn contain the location of the actual data Tablets. With this scheme, the depth of the tree is limited to three. For efficiency, the client library caches Tablet locations and also prefetch metadata associated with other Tablets whenever it reads the METADATA table.



Control and data flow in BigTable

Assigning Tablets#

A Tablet is assigned to only one Tablet server at any time. The master keeps track of the set of live Tablet servers and the mapping of Tablets to Tablet servers. The master also keeps track of any unassigned Tablets and assigns them to Tablet servers with sufficient room.

When a Tablet server starts, it creates and acquires an exclusive lock on a uniquely named file in Chubby's "servers" directory. This mechanism is used to tell the master that the Tablet server is alive. When the master is restarted by the Cluster Management System, the following things happen:

1. The master grabs a unique master lock in Chubby to prevent multiple master instantiations.

2. The master scans the Chubby's "servers" directory to find the live Tablet servers.
3. The master communicates with every live Tablet server to discover what Tablets are assigned to each server.
4. The master scans the METADATA table to learn the full set of Tablets. Whenever this scan encounters a Tablet that is not already assigned, the master adds the Tablet to the set of unassigned Tablets. Similarly, the master builds a set of unassigned Tablet servers, which are eligible for Tablet assignment. The master uses this information to assign the unassigned Tablets to appropriate Tablet servers.

Monitoring Tablet servers#

As stated above, BigTable maintains a 'Servers' directory in Chubby, which contains one file for each live Tablet server. Whenever a new Tablet server comes online, it creates a new file in this directory to signal its availability and obtains an exclusive lock on this file. As long as a Tablet server retains the lock on its Chubby file, it is considered alive.

BigTable's master keeps monitoring the 'Servers' directory, and whenever it sees a new file in this directory, it knows that a new Tablet server has become available and is ready to be assigned Tablets. In addition to that, the master regularly checks the status of the lock. If the lock is lost, the master assumes that there is a problem either with the Tablet server or the Chubby. In such a case, the master tries to acquire the lock, and if it succeeds, it concludes that Chubby is working fine, and the Tablet server is having problems. The master, in this case, deletes the file and reassigns the tablets of the failing Tablet server. The deletion of the file works as a signal for the failing Tablet server to terminate itself and stop serving the Tablets.

Whenever a Table server loses its lock on the file it has created in the "servers" directory, it stops serving its Tablets. It tries to acquire the lock again, and if it succeeds, it considers it a temporary network problem and

starts serving the Tablets again. If the file gets deleted, then the Tablet server terminates itself to start afresh.

Load-balancing Tablet servers#

As described above, the master is responsible for assigning Tablets to Tablet servers. The master keeps track of all available Tablet servers and maintains the list of Tablets that the cluster is supposed to serve. In addition to that, the master periodically asks Tablet servers about their current load. All this information gives the master a global view of the cluster and helps assign and load-balance Tablets.

← Back

Bigtable Components

Next →

The Life of BigTable's Read & Write O...

Mark as Completed

Report an Issue