

Cloud Computing & Big Data: Paper BigTable

1. How is the map in big table indexed?

The big table is indexed by a row key, a column key and a timestamp. The format is: (row:string, column:string, time:int64) → string.

2. Explain with example the illustration in Figure 1 with the role of Row, Column and TimeStamp indices.

Row: the row keys in a table are arbitrary strings. In this case, the row name is a reversed URL “com.cnn.www”. Reversing the URLs allows us to group contiguous rows into groups.

Column: Column keys are grouped into different sets called column families. In figure 1, there are two families. One is contents, the other is anchor. A column key is named using the syntax: family:qualifier, like “anchor:cnnsi.com” or “anchor:my.look.ca”.

Timestamp: Multiple versions of the same data can be put in each cell while they are distinguished by 64-bit integer timestamps. As we can see in figure 1, there are three versions of (com.cnn.www, contents:) data kept in the table using timestamps t3, t5 and t6 respectively.

3. Explain how various Google building blocks: GFS, SSTable, distributed lock service (Chubby) are utilized through an example and explanations.

GFS: BigTable uses Google file system to store log and data files. It provides the cluster management system for scheduling jobs, managing resources, monitoring machine status and dealing with failure to BigTable cluster that operates in a shared pool of machines.

SSTable: The data stored within BigTable is organized by maps that map a key to its value. For example, in figure 1, different columns are mapped to different data cells. The storage of BigTable data is implemented with SSTable which provides a persistent, ordered and immutable map from keys to values.

Chubby: Chubby is a highly available and persistent distributed lock service and it is vital to BigTable since any failure of Chubby will lead to the unavailability of BigTable. The tasks accomplished by Chubby include ensuring only one active master at any time, storing the bootstrap location of data, discovering tablet servers and finalizing table server deaths, storing access control lists and BigTable schema.

4. Explain the key concept illustrated in Fig 4 in your own language.

Figure 4 introduces the hierarchy of storing tablet locations. There are at most three layers. The first layer is the Root tablet whose location is stored in a Chubby file. This Root tablet is the entry of other METADATA tables while itself serves as the first METADATA table. In other words, from Root tablet, we can find the location of other METADATA tables. To make sure that the number of layers won't exceed 3, the Root tablet will always be treated as a whole. The METADATA tablets except Root tablet form the second layer. Each row of the METADATA tablet contains a row key that is encoded from a

tablet's identifier and the end row. From these row keys, we can locate the UserTablets which make up of the third layer of this hierarchy.

5. What is the role of bloom filter?

Bloom filter is applied to reduce the number of disk seeks. Ordinarily, if bloom filters are not used, a read operation has to read all SSTables that make up the tablet's state. When the required SSTables are not stored in the main memory, we have to access disk and retrieve it. On the other hand, if we create a Bloom filter for SSTables in a particular locality group, it allows us to check whether an SSTable (possibly) contains data for a given row/column pair. In this way, we can substantially decrease the number of disk access at the cost of a little storage for bloom filters.