Bigtable is a sparse, distributed, persistent storage multidimensional sorted Map. The index of the Map is the row keyword, column keyword, and timestamp; each value in the Map is an unresolved byte array. Bigtable has the following characteristics: wide applicability, scalability, efficient processing performance and high reliability.

The keywords of row in bigtable can be arbitrary strings, and the read and write operations of each row are atomic; the keywords of row in Bigtable are stored in lexicographic order, the rows in the table can be dynamically partitioned, each partition is called tablet, and the tablet is the smallest unit of data distribution and load balancing. Because row keys are stored in lexicographic order, querying with row keywords as the criteria queries are milliseconds.

Bigtable provides customers with a simple data model. With this model, users can control the size of data to read, it is dynamically set rather than static. Users can decide where and how to store the files, the data will be marked with the location of storage, and the name can be arbitrarily decided by the user. Bigtable treats the stored data as strings, but it does not interpret or modify these things, leaving it to the client program. The client program associates the stored data with these strings in a certain way, and these can be controlled by the user. Finally, BigTable can also set where the data is stored, such as on hard drive or in memory.

Bigtable's function implemented relies on three main parts: one that can connect to each user's library files, one main server that controls the others, and many directory servers to allocate the storage of files. The master server is responsible for assigning directories to directory servers, detecting directory server changes, balancing directory server load, garbage collection of Google's file system, and controlling changes to different column families. Each directory server manages a set of directories. The directory server handles read and write requests for the directory on the server, which means different files corresponding to different directories, and will also split the directories that are too large beyond the storage limit into multiple directories to store data. A Bigtable cluster stores a large number of tables. Each table consists of a set of directories, each of which contains a certain amount of stored data. Initially, each table contains only one directory. As the data in the table increases, it will automatically split into multiple directories for storage, and the default can reach 1GB per table.

To improve the performance of read operations, the Tablet server uses the policy of L2 caching. The scan cache is the first-level cache, which mainly caches the Key-Value pairs obtained by the Tablet server through the SSTable interface; the Block cache is the second-level cache, which caches the Blocks of SSTable read from GFS. Scan caching is very efficient for applications that frequently read the same data repeatedly, and block caching is more useful for applications that frequently have to read data near the data they just read.

The designing of BigTable meets the needs of most big data programs, which breaks the structured storage of relational databases. It can be deployed on thousands of servers, store petabytes of data, and provides a solid theoretical foundation and success stories for the rapid development of the entire Internet industry.