Bigtable: A Distributed Storage System for Structured Data

Kartikeya Upasani (kuu2101)

Paper Review, 2 October 2016

1 Motivation

With the large and growing number of internet users, the large volumes of data produced by each user, and the increase in importance of data analytics, there is a need to design special techniques that can help deal with these problems of scale.

2 Goal

Developing a system that can manage large volumes of structured data with low latency and high throughput.

3 Key Idea

Using a client-master-tabletserver architecture that stores structured data in a distributed fashion, while separating metadata and paraphernalia from the data serving process.

4 Approach

Structured data is stored in a tabular fashion with rows and columns being referenced by strings. The data is stored persistently in chunks called tablets on tabletservers that use Google File System for storage. Versions of data across timestamps can be maintained as required. The master server maintains the locations of tablets and routes clients to the requested tablet. Thereafter, the client and tabletservers communicate directly. Metadata is stored as a table too. The log of newer changes to the data and metadata is maintained in memory (*memtable*) and the log of older changes is stored persistently (*SSTables*). *Chubby* is a distributed lock control service that helps deal with concurrency. Several optimizations are performed: contiguous rows are stored on the same tabletserver, tablets are split up or duplicated for load balancing, *bloom filters* are used for fast existence checks in tables.

5 Results

A setup of N clients and N tabletservers was tested for request of R rows for a wide-range of values of N (1 to 500). It was observed that aggregate throughput increased linearly with N, while performance did not.

6 Conclusion

BigTable is successful in efficiently storing large volumes of structured data and serving it in large scale. It is deployed in production at Google.

7 Comments

The paper comprehensively covers design aspects of BigTable, experiments performed for testing performance, and the cases where it is deployed in Google. It is worth noting that the design choices made in the paper are heavily centered around simplicity, which seems to be important in a concurrent system such as this.