BigTable and Spanner

BigTable is a distributed storage system for managing structured data that is designed to reliably scale to petabytes of data and thousands of machines. BigTable resembles a database in many ways but it does not support a full relational data model, instead it gives users dynamic control over data layout, format and whether to serve data out of memory or from disk; and allows them to reason about locality properties of the data represented in underlying storage. Data is indexed using row and column names that can be arbitrary strings. BigTable also treats data as uninterpreted strings, although users usually serialize various forms of structured and semi-structured data into these strings.

BigTable is a sparse, distributed, persistent multi-dimensional sorted map. It is indexed by a row key, column key and a timestamp. Timestamp is a 64-bit integer and is by default a real time value in ms. Row keys are arbitrary strings and can be at most 64 KB in size although are typically 10-100 bytes. Every read and write under a single row key is atomic. Data is lexicographically ordered by row key. Each row range is called a tablet. Columns keys are grouped into column families, which are the basic unit of access control. Each cell can contain multiple versions of the same data in decreasing order and by default is limited to three. BigTable has automatic garbage-collection to collect stale versions of the data. Each tablet is represented by SSTable, which is a persistent, ordered immutable key-value map. GFS is used to store logs and data in SSTable file format, and each SSTable is represented by a set of blocks associated with indices. BigTable uses API that contains functions for creating and deleting tables and column families, changing clusters, tables and column family metadata. Users can read, write, delete values in the table, use look up values from individual rows, columns or iterate over a subset of the data from the table.

BigTable architecture consists of a master, that assigns tablets to tablet servers. The master is responsible for tablet management, handle schema changes and garbage collection of files in GFS. The tablet server handles read/write to the tablets. It has a Chubby server which used Paxos to elect master from the 5 active replicas. The namespace consists of directories and files which acts as locks. Chubby client registers use events for notification of changes. There is a cluster management system that manages job schedule, resource management on shared machines, failure recovery and monitors machine status.

When the memtable size reaches a threshold, memtable is frozen and converted to an SSTable and written to GFS and then a new one is created. A merging compaction can be performed that rewrites all SSTables into exactly one SSTable. SSTables generated are immutable and enables Bigtable to split tablets quickly.Users only need to communicate with the master to find relevant tablet server and cache this data till they are notified the data is stale.

Spanner is Google’s distributed database which was designed to be highly accessible as well as globally scalable. Like other distributed datastores covered in this class, Spanner is version based and replicates data in a process called sharding in response to failures as well as to balance load. The Spanner database provides some unique features. First of all, unlike other distributed systems, Spanner’s data replication can be configured in great detail. Unlike in older systems learned in class which aimed to be 100% transparent, Spanner allows the user to specify exactly what data is stored/replicated in which data center as well as how many replicas are maintained. Secondly, Spanner provides externally consistent reads and writes which allow it to support consistent backups, executions and schema updates at global scale.

As Spanner is meant to scale globally, there is a need to separate the application into subsystems. Spanner achieves this by specifying three control layers a *universe, zone,* and, *spanserver.* A Spanner deployment is called a *universe* and there aren't many of them at once. Spanner then divides itself into *zones*, each containing hundreds to thousands of *spanservers*. A zone can be seen as a BigTable server and are administrative deployments where data can be replicated. A spanserver is what holds the data called *tablets*. These tablets are a bag of key-value-timestamp mappings, the latter most field making it able to be more like a multi-version DB than a key-value store.