Cassandra – Review

Yixing Chen, yc3094

The paper *Cassandra - A Decentralized Structured Storage System* introduced Cassandra – a distributed storage system for managing very large amounts of structured data spread out across many commodity servers, while providing highly available service with no single point of failure.

Cassandra aims to run on top of an infrastructure of hundreds of nodes (possibly spread across different data centers). At this scale, small and large components fail continuously. In many ways Cassandra resembles a database and shares many design and implementation strategies therewith, Cassandra does not support a full relational data model; instead, it provides clients with a simple data model that supports dynamic control over data layout and format. Cassandra system was designed to run on cheap commodity hardware and handle high write throughput while not sacrificing read efficiency.

Compared to P2P storage systems that only support flat namespaces, distributed file systems typically support hierarchical namespaces. Update conflicts are typically managed using specialized conflict resolution procedures. There are many other file system and storage system, such as Farsite, Coda, GFS, Bayou, Ficus, Dynamo and so on.

A table in Cassandra is a multi-dimensional map indexed by a key. The key is atomic per replica and the columns are grouped together into sets called family. Rows are organized into tables; the first component of a table's primary key is the partition key; within a partition, rows are clustered by the remaining columns of the key. Other columns may be indexed separately from the primary key. In order to scale incrementally, Cassandra can dynamically partition the data over the set of nodes in the cluster. In order to achieve high availability and durability, Cassandra uses replication. Each data item is replicated at N hosts. Failure detection is used to avoid attempts to communicate with unreachable nodes during various operations in Cassandra.

The architecture of a storage system that needs to operate in a production setting is complex. In addition to the actual data persistence component, the system needs to have the following characteristics; scalable and robust solutions for load balancing, membership and failure detection, failure recovery, replica synchronization, overload handling, state transfer, concurrency and job scheduling, request marshalling, request routing, system monitoring and alarming, and configuration management. The most important core distributed systems techniques are partitioning, replication, membership, failure handling and scaling.

The Cassandra process on a single machine is primarily consists of partitioning module, the cluster membership and failure detection module and the storage engine module. The Cassandra system indexes all data based on primary key. The data file on disk in broken down into a sequence of blocks.

In the process of designing, implementing and maintaining Cassandra, the writers learn any important lessons. For example, don’t add any new feature without understanding the effects of its usage by applications. Most problematic scenarios do not stem from just node crashes and network partitions.

The paper demonstrated that Cassandra has provided its scalability, high performance, and wide applicability. Cassandra can support a very high update throughput while delivering low latency. In the future, Cassandra may involve adding compression, ability to support atomicity across keys and secondary index support.