Google File System

Sirui Tan st2957

Google File System (GFS) is a distributed file system designed to meet the rapid growth of Google's data processing needs. GFS is built upon a cluster of many unreliable inexpensive servers and is able to recover from possible failures. It is designed to store a modest number of large files and mostly process three primary file system operations: large streaming reads, small random reads and data append. GFS is designed for multiple concurrent write processes and maintains a high sustained bandwidth.

A GFS cluster consists of a single master and multiple chunkservers. It can serve multiple clients at the same time. Files in GFS are divided into fixed-size data chunks which are identified by globally unique 64 bit chunk handles. All data chunks are replicated and stored among chunkservers while the master stores solely metadata of files, including file namespace, map from files to chunks and location of each chunk's replicas. The separation of metadata and file data separates data and control flow, which is advantageous for using bandwidth wisely. The master node keeps file namespaces and file-chunk mappings in memory for efficiency and replica locations temporarily for synchronization purposes. It maintains operation logs to keep track of file changes and checkpoints for swift recovery.

GFS uses leases to maintain a consistent mutation order across replicas of a chunk. When a client wants to mutate on a chunk (e.g. write some new data), it starts by asking master the location of primary who holds the lease. Then client pushes data to all replicas followed by sending write request to primary. The primary then assigns a serial number to the request and forward it to every other replica. All replicas then mutate themselves in the order of request serial number. Atomic record append is almost identical to lease-based mutation with only a little extra logic at the primary. Mutation guarantees bandwidth usage by pipelining data transfer sequentially. GFS also provides a snapshot operation which makes an instantaneous copy of a file or directory tree.

GFS represents its namespaces as a lookup table mapping file name to metadata. Each node in the namespace tree is associated with a read-write lock for safe concurrent mutations in the same directory. Master node create a new chunk replica for three reasons: chunk creation, re-replication and rebalance. Master chooses where to place the replica so that chunkservers with more disk space, longer idle time and higher priority (in situations of re-replicates) are chosen, and replicas can be scattered apart. GFS master will not delete a file immediately, but postpone the deletion to a later regular scan of namespace. Chunkservers behave similarly with respect to orphan chunks.

As a highly distributed system of unreliable nodes, GFS maintains reliability using two mechanisms: fast recovery and replication. All nodes in GFS are designed to restore state and therefore able to recover instantly. Both chunks and mater states are replicated to ensure fast recovery at contingency. Each chunk is checksummed in case of possible data corruption.

Although faced with some disk and OS-related technical issues, GFS has successfully used by Google as a storage platform for dealing with web-scaled problems.