The Google File System – Review

Yixing Chen, yc3094

The paper *The Google File System* introduced a scalable distributed file system for large distributed data-intensive applications, that is Google File system. It provides high performance, scalability, reliability and fault tolerance on large number of inexpensive hardware and mass clients.

The google File System (GFS) is designed to meet the rapidly growing demands of Google's data processing need, it not only share same goals as previous distributed file system, but also has radically different points. For example, component failures are the norm rather than the exception, files are huge by traditional standards. Multi-GB files are common, most files are mutated by appending new data rather than overwriting existing data, and co-designing the applications and the file system API benefits the overall system by increasing our flexibility.

The architecture of GFS is separated as three different parts: master, chunkservers and clients. The master maintains all file system metadata, chunkservers store chunks on local disks as Linux files and read or write chunk data specified by a chunk handle and byte range, and GFS client code linked into each application implements the file system API and communicates with the master and chunkservers to read or write data on behalf of the application. All interactions among these 3 roles are needed to implement mutations, atomic record append and snapshot to minimize the involvement of master.

The GFS has high fault tolerance, it can use fast recovery (master and the chunkserver can re-store their state and start in seconds no matter how they terminated), chunk replication (each chunk is replicated on multiple chunkservers on different racks) and master replication (the operation log and checkpoints are replicated on multiple machines) to ensure the high availability. Besides, checksum is used to guarantee the data integrity.

Then the writers proved the GFS performance by implementing tests. The data they used are two clusters: one is the data for research and development, which is small reads, while the other one is for production data processing which is long sequential reads through entire files. The testing results show that GFS has great performance in different situations, which is stated by 3 aspects: micro-benchmarks, real world clusters and workload breakdown.

In my opinion, it's a great paper to introduce the Google File System to us. It illustrate its design, architecture, fault tolerance, measurement, test and experience clearly and easy to understand, we not only learn the features of GFS, but also learn many mechanism and possible problem in designing a file system, which may provide valuable experience for our future studying and working. Even though the paper is very old, its viewpoints and details can still enlighten us these days.