Paper review [1]

Yu SU 55323707

1. The problem

The paper tried to design a new file system, Google File System, to fit the rapidly growing demand of data processing need. In detail, there are four main problems. First, the system needs to handle the inevitable and numerous component failure; second, it should fit the huge size files; third, the system can both append and overwrite the data; last, it should be flexible for applications.

2. Challenge

There are some challenges of the system to solve the problem. First, the system needs constant monitor and the prompt recovery; second, the size of the file would be too large to handle and at the same time, small file could also be processed; third, how to reduce the workload in the operation of system is quite a challenge because the system needs to read both large and small file and write both in appending and overwriting; the last challenge is the performance of the system. The system must efficiently handle the concurrently appending, which means the sustained bandwidth is more important than the low latency.

3. key insight

There are many talented designs in the system design. First is the about the interface. To minimize the master's involvement in all operations, the system set the lease mechanism, which could maintain a consistent mutation order across replicas to minimize management overhead at the master. And in order to getting more efficient network, the system decoupled the flow of data from the flow of control. What's more, the system also provides an atomic append operation which is very useful in the distributed applications. The snapshot operation is also a novel design, which could make the creation of branch copies is quick.

Second is the architecture. The GFS cluster consists of a single master and multiple chunkservers and is accessed by multiple clients. The design of single master vastly simplifies the system and make it possible to make sophisticated chunk placement and replication decisions using global knowledge.

Third, there are some little novel design. For example, the system increases the chunk size which could extremely reduce the workload of interaction and network overhead. What's more, the master stores three major types of metadata, which could make the recovery more efficient. Besides, GFS uses some simple and efficient implementation to make the co-designing application more flexible.

4. Limitation

The system pays too much attention on the large size file and there is not optimal for small sized files. Only one master maybe causes a bottleneck problem. And the operation of system is also not efficient enough.

5. Future work

The system should try to solve the bottleneck problem by optimizing the master node, make the system more efficient.

[1] Ghemawat, S., Gobioff, H., & Leung, S. (2003). The Google file system. *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles - SOSP 03.* doi:10.1145/945449.945450