The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
Google*

Gabrielle Romano

11/11/2013

Main Idea

- The paper describes the design, operation and performance of the GFS (Google File System)
- The authors of the paper describe the ways in which the GFS implement the different tasks while managing a large amount of big data
- The GFS is a scalable, and fault tolerant distributed file system that runs on inexpensive commodity hardware.
- The GFS is used for large distributed data-intensive applications

Implementation

- The GFS cluster consists of one master server and many chunk servers that can be accessed by multiple clients
- Files are divided into fixed size chunks and are stored on local disks as Linux files
- The GFS uses a large chunk size coupled with lazy space allocation to avoid wasting space
- The master handles all the file system metadata (the file chunk namespaces, the mapping from files to chunks, and the locations of each chunk’s replicas)
- The operation log contains a historical record of critical metadata changes, files and chunks are identified by the time they were created
- When mutation (modification) is preformed on a replica, leases are to be used to maintain consistency among the other replicas.
- The GFS utilizes automatic record appends to allow the clients on different machines to append to the same file simultaneously
- The master creates, re-replicates and rebalances chuck replicas to spread replicas across racks, minimize the impact of failures and account for better load balancing and disk space.
- Garbage collection allows the GFS to keep the file after it has been deleted for three days by renaming it to a hidden name
- The GFS relies on fast recovery and chunk replication to account for its faulty hardware and increase reliability
- In order to preserve data integrity every chunk server uses check summing to detect corruption
- Servers generate agnostic logs , record events that can be diagnosed with the use of diagnostic tools

Analysis

- Since The GFS is run on cheap commodity hardware, it is prone to failures that might lead to corrupt data; however, since hardware malfunctions are considered normal, Google was able to adequately prepare strategies to counteract the failures
- The GFS is like other file systems in which it must effectively maintain performance, reliability, scalability, and availability for their clients applications
- However, The GFS is also a unique system in that it conforms to Google’s need to process large amounts of big data

Advantages/Disadvantages

- GFS is redundant, files are duplicated three times
- Master operations are fast due to the metadata being stored in memory
- GFS garbage collection is reliable by cleaning up replica messages and allows files to be viewed after they are deleted (for three days)
- GFS is mostly used to manage large files
- However GFS does not adequately manage small files
- Since GFS is implemented on commodity hardware it is prone to component failures
- Many of the GFS disks did not respond due to mismatches in the older Linux drivers, they did however responded reliably to the more recent ones.

Real-World Use Cases

- AFS, xFS Frangipani and Intermezzo provide caching in comparison to the GFS
- The GFS is similar to the NASD prototype in which commodity machines were used
- GFS data is stored across servers to ensure fault tolerance similar to xFS and Swift
- Many distributed file systems unlike GFS do not have centralized servers

Real-World Use Cases (cont.)

- Storage
  - “Used Space” consists of all replicas (three files)
- Metadata
  - Googles servers have 50 to 100 MB of metadata which allows the server read the data in a few seconds
- Appends versus Writes
  - Record appends are used more than writes