The Google File System

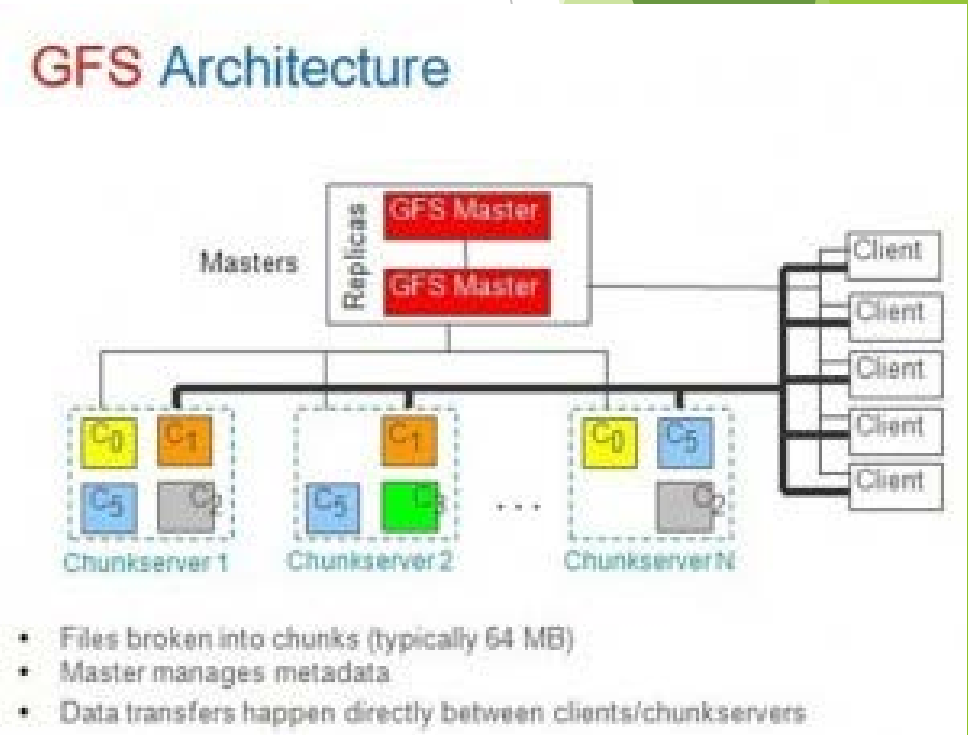
Paper by: Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Presented by : Demetrius Williams

Date: 11/21/2013

Main Idea: How GFS is the Best

- Greatly separates itself from previous distributed file systems
  - Design, reliability, performance, scalability, and availability
- Advanced technological environment
  - Put with the latest in technology to create new features
- Massive storage facilities
  - Data clusters with hundreds of terabytes on over 1000 machines
  - Storage is always running smooth and never at capacity



Implementation of GFS

The Google File System has been implemented to meet the rapidly growing demands of Google's data processing needs

- Implemented as a user space library
- High performance to a large number of clients
- Fault tolerance while using inexpensive commodity hardware

Analysis

- The google file system shows a radical difference from other File systems out right now; showing efficient ways to divide data and utilizing replication by storing each chunk multiple times. The implementation was good by making it userspace thus increasing fault tolerance. Google has made an ideal file system to best fit what they are doing. That being said, the GFS was built for a specific set of applications and would not be good as a general purpose file system.

Advantages and Disadvantages

pros

- GFS consistency model supports highly distributed applications well and is easy and efficient to implement
- High availability through the hundreds of servers in a cluster
- Atomic Record Appends
- Snapshot operation to make quick copies of huge sets of data
- Chunk replica placement policy

cons

- GFS is built from inexpensive commodity components that often fail
- Garbage collection: after a file is deleted, GFS does not immediately reclaim the available storage.
- No use of cache for data
- Bottleneck

Real-World Use Cases

For GFS clusters, cases are set up where a single task consist of many processes on many machines reading and writing files simultaneously.

Case 1

- A cluster can be used within google for research and development by over 100 engineers.
  - A task is initiated by a human user and can use anywhere between a few megabytes up to many terabytes.
  - The cluster then transforms the data and writes the results back into the cluster

Case 2

- A cluster can be used for production data processing.
  - This can be a long task that will continuously generate and process multi-terabyte data sets
  - Human intervention on occasion