The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. SOSP'03, October 19–22, 2003, Bolton Landing, New York, USA.

Nick DiCamillo
22 Nov 2013

Main Idea

- The Google File System (GFS) is a distributed file system designed by Google's observations of their workload both current and anticipated
- The main goals of the GFS are performance, scalability, reliability, and availability
- The paper discusses:
  - The file system interface
  - Many aspects of the design
  - Measurements of the system

How Idea is Implemented

- GFS consists of thousands of storage machines built from inexpensive hardware and accessed by client machines
- Files are organized hierarchically in directories and identified by path-names.
- A GFS cluster consists of a single master and multiple chunkservers and is accessed by multiple clients
  - The master maintains all file system metadata
  - Chunkservers store and manage each 64-bit chunk on local disks
  - Clients interact with the master for metadata operations, but all communication goes directly to the chunkservers

Analysis

- The GFS is good because:
  - It is designed for Google
    - Can meet all data and storage needs
  - A file is never deleted just relocated elsewhere
  - Updating or editing files is easy
  - Google runs billions of searches a day, without a good file system it would not be possible to run daily operations
  - Low cost

Advantages

- GFS has snapshot and record append operations
  - Snapshot creates a copy of a file or a directory tree at low cost
  - Record append allows multiple clients to access the same file at the same time while guaranteeing the atomicity of each
- A Large chunk size offers three advantages
  - Reduces clients need to interact with the master
  - Reduce network overhead
  - Reduces the size of the metadata stored on the master
- The chunk replica placement policy serves two purposes
  - Maximize data reliability and availability
  - Maximize network bandwidth utilization
- The GFS does not fully delete files

Disadvantages

- There are problems caused by application bugs, operating system bugs, human errors, and the failures of disks, memory, connectors, networking, and power supplies.
- Since the systems is built from inexpensive parts, failures in them often occur.
- Hot spots did develop when many requests to a single file were made at the same time.
- Multiple chunk replicas lead to duplicated data

Real-World Use Case

- Cluster A is used regularly for research and development by over a hundred engineers
  - A typical task is initiated by a human user and runs up to several hours
  - It reads through a few MBs to a few TBs of data, transforms or analyzes the data, and writes the results back to the cluster
- Cluster B is primarily used for production data processing
  - The tasks last much longer and continuously generate and process multi-TB data sets with only occasional human intervention
- In both cases, a single "task" consists of many processes on many machines reading and writing many files simultaneously