

The Google File System
and

A Comparison of Approaches to Large-Scale Data Analysis

By Michael McGinnis

Due: 12-12-2014

Main Ideas

- A redesign of traditional file systems.
- Anticipate component failures and treat them as normal occurrences and not exceptions
- Optimize for large files instead of small files
- Use a single master and several chunkservers.
- High sustained bandwidth is more important than low latency.
- Use a large chunk size to avoid wasting space due to fragmentation.
- Allows multiple masters multiple operations to be active at once and user locks over individual regions.
- Only reclaims physical storage during regular garbage collection.
- High availability through fast recovery and replication.
- System based around fault tolerance.

How idea is Implemented

- Because the system is designed to anticipate failure it is built from inexpensive commodity components that often fail. It constantly monitors itself, and must recover quickly.
- The single master has a minimal involvement to ensure it never becomes a bottleneck. Client read and writes are not done through the master.
- Data is stored on a minimum of three different racks to ensure no loss of data.
- To fully utilize each machine's network bandwidth, the data is pushed linearly along a chain of chunkservers rather than distributed in some other topology such as trees. This ensures the data is transferred as fast as possible.
- When a file is deleted, the master logs it; then it is moved to hidden name where upon a masters regular scan of the file system it is removed.

My Analysis

- Upon initial review this seems like a very good idea.
- By using cheap and inexpensive components you can buy a much larger volume of hardware for the same price.
- The data can then be cheaply backed up, and if any hardware component fails it is fine since it is cheap and the data is backed up.
- Based on what we learned on space allocation, using a large chunk size seems very smart because less time will be spent reallocating space and defragmenting.
- In todays age of rapidly growing file sizes, optimizing for large files instead of small seems like an incredibly smart solution on googles part.

Comparison

- According to the comparison document, the google file system is capable of scaling up to 1000s of nodes but the superior efficiency of modern DBMs alleviate the use such massive hardware.
- The technologies developed in the last 25 years has granted large performance to both database systems.
- In testing DBMS-X was 3.2 times faster than MR and Vertica was 2.3 times faster than DBMS-X.
- There is not a substantial difference in performance between the systems in the testing.

Advantages and Disadvantages

➤ Advantages:

- Single Master
- Slightly more reliable due to numerous backups and ease in which backups are created.
- Better management system
- Extremely cheap
- High availability

➤ Disadvantages:

- Poor performance on small reads/ writes
- Would probably not work well general datacenters
- Is not actually for sale
- Uses a relaxed approach which would not be acceptable for enterprise work.