The Google File System

Aaron Kippins

11/25/2013

Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. "The Google file system." ACM SIGOPS Operating Systems Review. Vol. 37. No. 5. ACM, 2003.

Brief Overview Of The Paper

This Article Discusses The Use Of The Google File System

- The Google File System is a scalable distributed file system for majorly distributed data-intensive applications.
- It provides it’s users with an efficient manor of organizing their files to suit their storage needs.
- It provides great tolerance for errors while running on inexpensive service hardware.
- It also delivers high aggregate performance to a large number of clients.

Implementation of the Idea

Interface	Architecture	Single Master	Chunk Size	Metadata
GFS provides a familiar file system interface We support the usual operations to create, delete, open, close, read, and write files.	A GFS cluster consists of a single master and multiple chunkservers and is accessed by multiple clients	Having a single master vastly simplifies our design and enables the master to make sophisticated chunk placement and replication decisions	Chunk size is one of the key design parameters. We have chosen 64 MB which is much larger than typical file system block sizes	All metadata is kept in the master's memory
GFS has snapshot and record append operations also Snapshot creates a copy of a file or a directory tree at low cost	Files are divided into fixed-size chunks for reliability, each chunk is replicated on multiple chunkservers	Clients never read and write file data through the master. Instead, a client asks the master which chunkservers it should contact	A large chunk size offers several important advantages. Like reducing clients' need to interact with the master	Master operations are very fast so periodic scanning is used to implement chunk garbage collection
Record append allows multiple clients to append data to the same file concurrently while guaranteeing the atomicity of each individual client's append	Each chunk is identified by an indisputable and globally unique 64 bit chunk handle	the client typically asks for multiple chunks in the same request and the master can also include the information for chunks immediately following those requested	a large chunk, a client is more likely to perform many operations on a given chunk, it can reduce network overhead by keeping a persistent TCP connection	Its simpler to request the data from chunkservers at startup, and periodically thereafter so for this reason the master does not keep a persistent record of which chunkservers have a replica of a given chunk.
	The master maintains all file system metadata	This extra information sidesteps several future client-master interactions at practically no extra cost	At 64 MB the size of the metadata stored on the master is reduced	

Analysis of the ideas and its implementation

Design

The design of the Google File System is very efficient, well organized, simplistic to it's users, and very reliable in the way it stores and backs up it's data

Interactions

The minimization of referencing the master increases the fluidity and speed of the design. There is no constant need to read from the drive since the system reviews the system at start up and keeps a cache of data with all of the changes made

Reliability

The system also places great focus on keeping the data constantly backed up, thought the users data isn't stored directly on the master drive it is stored on 3 chunkservers so the user always access has 3 copies of their data

Pros & Cons

Advantages	Disadvantages
Spatial Availability • Having an allotted 64MB of chunk size allows users to be less reliant on the master • It also allows the user more freedom to store as much they please Lazy Garbage Collection • After a file is deleted it isn't immediately wiped from physical storage. There is a window that allows it to be recovered.	No Optimization For Smaller Files • A small file consists of a small number of chunks, perhaps just one. The chunk servers storing those chunks may become hot spots if many clients are accessing the same file • It seems as if the system should have the ability to store smaller files as easily as it does the larger files

Real World Uses

There are 2 Clusters used within Google itself

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.