THE GOOGLE FILE SYSTEM

BY: STEVEN O’MEARA

NOVEMBER 16TH, 2013

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

MAIN IDEA OF THE PAPER

- The google file system papers main idea is to show the reader how the google team uses and implements their widely used google file system. They go through how each component specifically works with the system as well as how they tackled problems that arose as they were implementing the system for wide scale use. They are trying to build a large data database that allows multiple users to access and append data as they see fit, most of the time simultaneously. They use low end machines to keep the cost down as well as making the system scalable and use redundancy to allow the system to be reliable and available at all times.

HOW THE IDEA WAS IMPLEMENTED

Assumptions

-Google used assumptions to not only challenge themselves but build the system so that it would be able to deal with problems normal to a big data system.

-**The assumptions include**:

-The idea that the system should be self monitoring.
-The system would be focused around big data reads.
-The file system should have logical flow for many clients appending the same file.
-High sustained bandwidth is more important then low latency.

Interface

-The files are organized in a hierarchy with directories and identified by pathnames.
-Supports all usual operations. (Create, delete, open, close, read and write.)
-The interface also supports two more important operations:

-Snapshot: creates a copy of a file at low cost
-Record Append: Allows multiple clients to append at the same time while guaranteeing the atomicity of each append.

Architecture

- The main overview of the Google Files System(GFS) is that is uses a single master file as a main place for all applications to reference, and then puts all of the files into a 64mb chunk and onto chunkservers.

Master: It is used as a directory, it sends the client to the chunk server that the file they are looking for is located.
-The client sends a request with the file name and chunk index, the master will return the handle as well as the location of the file and chunkserver.

Metadata

-The master stores three different types of metadata. They include file and chunk namespaces, mapping from files to chunks and locations of each chunk's replicas. The first two are kept persistent by logging mutations in the operations log.

Metadata Cont.

In-Memory Data Structures

- Since metadata is stored in memory, master operations are fast.
- Allows the master to periodically can through its entire state.

Metadata Cont.

Chunk Locations
-The master polls chunkservers for their location on startup and sends heartbeat messages for updates.
-This allows the master and chunkserver to be in sync as the chunkserver leaves and joins the cluster.

Metadata Cont.

Operation Log
-Contains metadata changes and is central to the Google File System.
-It is replicated on multiple remote machines and responds to client operation only after flushing both locally and remotely.

Architecture Cont.

-**Chunk Size**: One of the key components of the Google File System, each chunk is stored as a Linux file on a chunkserver.
Makes use of lazy space allocation to avoid wasting space due to internal fragmentation.
Chunk size has its benefits, for one it allows the client to only interact with the master once as it reads/writes on the same chunk, it reduces network overhead by keeping a persistent TCP connection to the chunk server over a long period of time, and it reduces the size of the meta data stored in the master.

ANALYSIS OF THE IDEA AND ITS IMPLEMENTATION

-Since it's google, I shouldn't have expected anything less than exceptional work done on the Google File system. I believe that the main reason why they had so much success with this large file system is the fact that they included assumptions. Google is a long standing corporation that has learned a lot from their various ventures and they used those past experiences as a part of the process of this system. Using assumptions they put down the focus of the system as well as getting around problems that will most likely arise from implementing a file system this large.

-They had a great idea of keeping the main focus of the system to be the large files rather then the small files. They allow small files into the system but they were not optimized in the system in the least.

-Keeping only meta data in the master file was also a great idea. It would be too much information to store all of the data on one master drive since this is entirely a big data system. Using the master file as a directory rather then a storage location was one of the most intelligent things that google could have done.

ADVANTAGES AND DISADVANTAGES

Advantages

-Since the chunkservers are located on cheap Linux machines and are replicated on at least 3 different machines it makes the system both reliable and fairly cheap .

-Since the system was built around assumptions it avoids most of the problems that a file system of this scale would usually have. The google team tackled a lot of usual problems from the beginning making the system require little to no human fixes during its implementation.

-The master is self monitoring and self fixing, if it runs into a problem it can deal with it accordingly with fixes that are already in the system.

-Setting up an operation log was also a huge benefit to the system as a whole as it can keep track of critical metadata changes, this is greatly important to the master as it needs a way to track these changes to it can update itself.

Disadvantages

- Since the architecture of the entire system is built on low end equipment then the system will be just as fast as its slowest component. Though using all these low end machines is cost effective, it doesn't give the speed that a system built off high end machines would

- Right now the entire system is built around only google related applications, if someone is to attempt to use the system with a non-google application there may be major errors.

- The system is built around large data, though it allows small data into its system it could have errors as, as per my reading, it is not as closely regulated as big data. It seems to be entirely overlooked.

REAL-WORLD USE CASES FOR THE GOOGLE FILE SYSTEM

- The Google File System is already widely in use. It is connected to most of googles applications. Google uses the file system in two clusters, which they call Cluster A and Cluster B.
- Cluster A is in regular use and is focused around research and development for over a hundred engineers. In this cluster a task is initiated by a user and runs for several hours
- Cluster B is used for the production of data processing. These tasks last longer than Cluster A tasks and they continuously generate and process multi-TB data sets with little to no human intervention.