The Google File System.

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

By: Tori Palazzolo

November 25th 2013

Main Idea

- The sole purpose of the Google File System is to support the massive amount of information that google receives, and process it in a coherent matter.

- The file system contains many different storage systems, and are accessed by client machines. Within the Google File System many components are not reliable 100 percent of the time, however since there are so many it is not an issue if one breaks down.

- GFS unreliability needs there to be constant monitoring, to detect any errors or make any necessary recoveries.

- Another issue the GFS runs into are handling each files which contains numerous amount of content. When you are managing billions of objects within the files, each I/O operation and block sides, have to be relooked at to see that the file is functioning properly.

- Most files receive numerous amount of data, therefore most files are mutated to avoid overwriting the existing data.

- With new data coming in it is important to focus on performance optimization and atomicity. Focusing on those issues will allow the file system to run sufficiently.

- When designing applications Google must focus following API guidelines, which allows them to be flexible. The consistency within the file allows the model to be simplified without burdening the applications.

- Also GFS clusters each have a different purpose, with different amount of storage, so clients can access them continuously throughout the day.

Implementation of the GFS.
The overall goal of the GFS is to ensure that large amounts of data is organized appropriately, ensuing that the users will have reliable access to their data.

- The Google File runs on Linux, which has the advantage of being accessed by multiple users.

- The GFS codes is implemented within the file systems API. This allows the servers to communicate and read or write data for the application.

- Files are divided into fixed sized chunks, in order to sort the size of the data. Chunk servers store chunks, than organize the size of the data by byte range.

- The master maintains all the metadata access control information, and the mapping from files to chunks and their current location. The master also communicates with the chunk server in order to give the chunk necessary instructions.

- GFS files are organized in a hierarchy data structure, which is than located within the directories and identified by the path names.

- GFS has snapshots that record operations. These snapshots than create a copy of this file, or create a directory tree. When the snapshots are recorded it allows multiple clients to get data where it needs to go, while also guaranteeing atomicity of the data.

- To ensure that data remains in tack. Each piece of information that a client sends to the file system is stored three times, to ensure that data does not get lost.

My Analysis

Each day many users use Goggle to access information, read emails, and store important documents. Google is the most popular search engine in the world, meaning that this large amount of data, must be available all of the time, with limited amount of bottlenecks within the system.

- Google serves millions of people each day, meaning that people are constantly accessing and storing data. This numerous amount of data encouraged google to create a file system, that will separate data into chunks on numerous amount of servers, so the clients can have easy access, and to ensure data is not lost.

- Goggle needed to create a way to steer away from a system that depended entirely on the hardware, which allowed them to create a system that will constantly be available to users.

- Many of Goggles applications are created with this system, allowing Goggle software, such as google docs, Gmail and goggle drive to be accessed by millions of people everywhere.

- This Goggle File system, is a great system for large amount of data. What the system primarily focuses on, is taking large data and separating it into smaller chunks. By separating the data it keeps the system from overloading, and allows the users to have easier access to the data.

Advantages And Disadvantages.

Advantages

- Large Chunk Sizes reduces the clients need to interact with the master chunk, since the same chunk requires only one initial request.

- A large chunk client is more likely to preform many operations within the chunk. It reduces network overload by keeping a constant connection with the TCP, over long periods of time.

- Many chunk replicas that are not known to the masters is "garbage." The Goggle File System has a garbage collection, which deletes files without any precautions to the data.

Disadvantages.

- Since there is numerous amount of data being used across the Google File System, one disadvantage are the delays, from the overload of data, which can halt Goggles ability to perfect the system, since storage is so tight.

- Small files consist of small chunks, therefore it can pose a problem when many people are trying to access that same file. Many of these chunk servers were overloaded by hundreds of simultaneous request.

Real World Use-Cases.

- The GFS, is a system that is specially made for large amount of data.

- The GFS can be used for any system that has numerous amount of data that people must access and store.

- Such companies like IBM, Microsoft and Apple, can use similar designs, since millions of people are accessing and storing information on their servers every day.

- Today many technology users need consistent access to their data at all time. A client wants to know that they can log onto Facebook, or google something anytime anywhere. Therefore they are constantly accessing and storing data on servers.

- Large scale companies like Google need to ensure their clients that they are reliable, so people continue to use their site and or software.

- Many large scale companies need to find ways to organize the large amount of data that they have, so that the servers do not get overloaded, break down, and cause issues to the overall system.