The Google File System & A Comparison of Approaches to Large-Scale Data Analysis

Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung

Google+

Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J, Abadi, David J. DeWitt, Samuel Madden, Michael Stonebraker

Megan Crane

5/9/2014

Main Idea

- The main idea of the first paper is to describe and explain the process of creating and implementing the Google File System (GFS)
- The paper explains how the GFS provides fault tolerance while running on inexpensive commodity hardware and delivers high aggregate performance to a large number of clients
- The paper also mentions that the GFS met all of its storage needs and is widely deployed within Google as the storage platform for the generation and is used in the processing of data used by the developers’ service as well as research and development efforts that require large data sets

Implementation

- The GFS was created to include a familiar file system interface but does not implement a standard API such as POSIX
- Along with supporting usual operations such as create, delete, open, close, etc., GFS also has snapshot and record append operations.
  - These are useful for implementing multi-way merge results and producer-consumer queues that many clients can simultaneously append to without additional locking; tools invaluable in building large distributed applications
- The GFS also has a single master that simplifies the design and enables the master to make sophisticated chunk placement and replication decisions using global knowledge
  - The masters involvement is minimized in reads and writes to ensure it does not become a bottleneck

My Analysis

- I think the concept behind creating a simpler filing system that better supports large systems was a good idea.
- The developers of GFS recognized some core issues with earlier file system designs and worked to analyze these issues and create something more efficient and effective. I believe recognizing these issues was a huge positive step towards creating a program that has been so successful.
- Finally, I was very impressed that the developers have already implemented their system and it has been adopted into Google as well as other systems due to the fact that it performs its objectives so well.

Comparisons between Ideas and Implementations

- After going through the two papers, I believe that both studies were conducted thoroughly with a lot of evidence supporting the conclusions that the researchers arrived at.
- Both papers had a clear idea of what they were trying to explain and went through many different tests to ensure that the information that was being outputted was relevant and accurate.

Advantages and Disadvantages

- The studies done in the second paper showed that Parallel DBMSs have a faster run-time than MapReduce, which Google has its own version of. This suggests that a Parallel DBMS would have the advantage of making the GFS run more quickly.
- However, I would say that even though the MapReduce was proven to be slower than Parallel DBMSs, I still believe MapReduce is a better program to go with when dealing with systems like the GFS as MapReduce is better equipped to deal with errors and crashes efficiently which is an important aspect when dealing with a large-scale file system. A little speed in searching can be sacrificed to have a program that runs smoothly, speed can be implemented later.