The Google File System

A Comparison of Approaches to Large-Scale Data Analysis

One Size Fits All- An Idea Whose Time Has Come and Gone

Brian Tracey

14 March 2016

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

Pavlo, A., Paulson, E., Rasin, A., Abadi, D. J., Dewitt, D. J., Madden, S., & Stonebraker, M. (2009). A Comparison of Approaches to Large-Scale Data Analysis. Proceedings of the 35th SIGMOD International Conference on Management of Data - SIGMOD '09.

Stonebraker, M. (Adapter). (n.d.). One Size Fits All- An Idea Whose Time has Come and Gone [Video file].

THE GOOGLE FILE SYSTEM

- Google created The Google File System (GFS) to handle large scale data processing on low cost hardware.
- The Google File System uses a separate file design system in order to meet the high demand of users performing performing multiple tasks at once.
- GFS is also scalable for data distributed intensive applications.
- It also handles failures extremely well due to its' constant monitoring, replication of data and automatic recovery.

IMPLEMENTATION OF GFS

Interface	Architecture	Single Master	Chunk Size	Metadata
Supports usual operations to create, delete, close, read and write files	GFS cluster consists of a single master and multiple chunkservers and is accessed by multiple clients	Enables the master to make chunk placement and replication decisions	Choose a large chunk spot	All metadata is kept on the master's memory
GFS has snapshot and record append operations	Files are divided into fix sized chunks	Minimalize its involvement	Large chunk spot reduces client interaction	Periodic scanning is used to collect chunk garbage
	Each chunk is identified by a 64 bit chunk handle		It reduces network overhead	
	The master maintains all the file system metadata		It reduces the size of metadata on the master	

ANALYSIS OF GFS

- The reliability, the design and the interactions allow for the success of the Google File System.
- The constant backing up and storing data on three separate chunkservers allows for reliable data.
- The design, being that it is efficient, well organized, and reliable is a major reason for GFS success.
- Since the master does not have to be referenced every time, the GFS can run fast and smoothly.

A Comparison of Approaches to Large-Scale Data Analysis

- A Comparison of Approaches to Large Scale Data Analysis looks in depth at the two different approaches: MapReduce and Parallel DMBS.
- MapReduce is a more simple approach that focuses on key/value data pairs. The "Map" function reads a set of records from an input and outputs it in a the form of new key/value pairs whereas the "Reduce" combines the records and places it in an output file.
- Parallel DBMS is executed by the fact that most tables are partitioned over nodes in a cluster and that the system uses an optimizer that translate SQL commands into a query demands split upon nodes.
  - So if a file size is too large, parallel DBMS will split it up in similar nodes.

IMPLEMENTATION OF LARGE DATA APPROACHES

	Schema Support	Indexing	Programming Model	Data Distribution	Flexibility	Fault Tolerance
MapReduce	Free structure, but must be built into map and reduce programs	Too simple to provide built in indexes	Uses high level languages that shares codes	Has to perform tasks manually	Full generality and flexibility	Adept to handling node failures
Parallel DBMS	Requires data to fit into rows and columns	Uses hash or B tree indexes to speed up data and supports multiple indexes per table	Have to state what you want in programming languages such as SQL	Uses knowledge of data distribution ad location to their advantage to balance computational workloads and minimizing the amount of data transmitted	Has insufficient expressive prowess	Restarts in the event of a failure

Analysis of Large Data Approaches

- In conclusion, after comparing the two approaches to large scale data analysis, the parallel DMBS was more advantageous in performance compared to MapReduce.
- MapReduce, however, was easier to setup and was more extensible compared to DBMS.
- MapReduce also handles failures better, but at a cost of performance.

One Size Fits All- An Idea Whose Time Has Come and Gone

- The main idea of Stonebraker's paper was that there was not enough storage in one database to hold all relevant data.
- Stonebraker focused on finding the fastest and most efficiency of finding data in a DMBS.
- Vendors will shift to column stores rather than row stores, which, in return, will be two times as fast in returning data.
- NoSQL market will grow with the increase of Big Data. NoSQL has no standards since it does not follow the traditional relational model.
- Stonebraker includes other types of markets including the streaming market, complex analytics and graph analytics.
- There is a lot of opportunity for growth in the future to increase efficiency, speed, and reliability.

One Size Fits All- An Idea Whose Time Has Come and Gone

- The Google File System is an adapted version of the MapReduce system. As Stonebraker stated, "Not one size fits all." As in the MapReduce system cannot be used for all systems, but companies like Google broke it down and amended it to their specific needs. This is advantageous to Google, since they have an extremely efficient way of searching through data on inexpensive hardware.
- There are disadvantages, however. The GFS's chunkservers can develop hotspots. Google still needs to work on a type of privacy that will allow clients to use other client data, which reduce overload. This would reduce record stores and Big Table clones as stated in the video.