Big Data Paper Summary

By Juan S. Vasquez - 3/14/16

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. *The Google File System*. 2003.

Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden, Michael Stonebraker.
A Comparison of Approaches to Large-Scale Data Analysis. 2009.

The Google File System - Main Ideas

- The GFS demonstrates qualities required for supporting large-scale data processing workloads on commodity hardware
- Reexamined traditional file system assumptions in order to radically alter the design
- Fault tolerance and diagnosis provide the greatest challenges

How Are the Main Ideas Implemented?

- Separation of file system control from data transfer
  - File system control passes through the centralized master, which has reduced involvement in common operations, preventing a bottleneck
  - Data transfer passes directly between chunk servers and clients
- Treat component failures as the norm rather than the exception
- Optimize for bigger files that are mostly appended to and read
- Extend and relax the standard file system interface to improve the overall system

Analysis of Implementation

For the authors, finding exactly what was needed from their new file system design began with describing key observations. These ‘assumptions’ provided a path for the authors to follow, as well as their goals. Their implementation of the system addressed all of their key observations.

This was a successful reexamining of the traditional file system assumptions while focusing on addressing day-to-day processing needs for complicated distributed systems with existing commodity components.

Comparison Paper - Main Ideas

- MapReduce (MR) vs. parallel SQL database management systems (DBMS)
- Both cluster computing and parallel database systems come with interesting tradeoffs; there are performance differences in various aspects of the task processes
- Analysis of both MR and DBMS architecture, accessibility, extensibility, and performance

How Are the Main Ideas Implemented?

- Experiments conducted revealed DBMS as having a significant performance advantage over MR
  - Performance issues attributed to superior minimization of hardware failure
- MR was found to be more accessible (setup, usage) and cheaper than DBMS
- MR lacks schema, an architectural difference between it and DBMS that is likely to remain in the long run and provide important consequences such as parsing at run time instead of at load time, compression issues, etc.

Analysis of Implementation

The authors seeked to describe and compare the computing models of MapReduce and DBMS, as well as evaluate the systems in the categories of performance, development complexity, accessibility, cost, and extensibility, among others. The testing conducted answered these questions, as well as speculation about the future of these paradigms.

Comparison of Ideas/Implementations

After reading both papers, it seems clear that the Google File System takes more after the MapReduce computing model that was analyzed in the Comparison paper. Performance seemed to be the most important factor in the Comparison paper, while fault tolerance and diagnosis seemed to be a bigger driving factor in the GFS paper. Overall, both implementations provided satisfying conclusions to the queries posed.

Stonebraker Talk - Main Ideas

- Relational Database Systems were once thought to be potentially universal, one-size-fits-all
- Different markets (Data Warehouse, OLTP, NoSQL, Complex Analytics, Streaming, Graph Analytics) have a huge diversity of engines, and traditional row stores are good in none of them
- “Elephants” (legacy vendors) will try to adapt, preserving their customer base and maintaining market share

Advantages/Disadvantages - Google File System

Advantages

- Relatively cheap and accessible computing model
- Higher failure tolerance

Disadvantages

- Because components often fail, a higher priority must be placed on fault tolerance and diagnosis, creating an impact on performance
- Lacks schema