

Big Data Paper by Dio Minott

3/3/17

Bibliography: “The Google File System” by Sanjay Ghemawhat, Howard Gobioff, and Shun-Tek Leung. “A Comparison of Approaches to Large-Scale Data Analysis” by Andrew Pavlo, Erik Paulson, Alexander, Daniel J. Abadi, David J. Dewittt, Samuel Madden, and Michael Stonebraker. Michael Stonbraker’s ICDE 2015 “10 Year Test of Time”

“The Google File System”

The Google File System is a system that can handle large data processing workloads that can be ran on a hardware. The main idea of this system is to be able to handle large sets of data that is unable to fit on a single commodity hard disk. It also became a cheaper way towards using a commodity hardware. Also it optimize large files that are appended and read.

GFS implementation

Replications of chunks using the chunk server improves the bandwidth for read access patterns. A prefix compression and table lookup is used in order to decrease the time on chunk server translation. A simpler and more reliable approach for this system is having some lazy garbage collection which improves latencies of each chunk of data. Block check summing for consistencies of chunks. Also high aggregate throughput is achieved by separating file system control which passes through the master from data transfer, which passes between the chunk server and clients.

GFS Analyze

- The file system will perform poorly on small files which can cause metadata pressure on the master can increase. Also the master is only used to only establish some type of connection when actual data transfer doesn't involve the master. GFS takes many lazy approaches hence lazy garbage collection and snapshot.

“A Comparison of Approaches to Large-Scale Analysis”

The main idea is to compare two different approach when dealing with large scale data. One approach is MapReduce which is programming model and a implementation for processing and generating big data sets with parallel distributed algorithm on a cluster. Another approach is a parallel Database Management System. Both systems divides data sets into partitions, which are located to different nodes to achieve parallel processing.

Implementation

- The map reduce is broken down into two steps which are map and reduce. MR inputs key values into a set of pairs and uses a reduce function that groups all the values that belong to the same key and produces an output key value set by merging them together. Parallel DB systems are much easier to use than hadoop. DBMS uses b-tree indices to speed the execution of selection operations, uses novel storage mechanism, has the ability to work directly on compressed data, and has sophisticated parallel algorithms querying large amounts of data.

Analyzing

MapReduce was designed fairly simple, but yet not all applications can run on this framework. Also, MR model on used on thousands of node is a forced solution that wastes a vast amount of energy. SQL is still a very powerful tool to use today but can be difficult for some users to use depending how large the scale is.

Comparison

- Both articles presented different approach when dealing with big data. The Google file system seems to be less efficient compared towards using a parallel database of MapReduce. While the Google file system article talks about being able to run large data that's unable to fit on hard disk, the comparison article focuses mainly on time vs. space.

Stonebraker Talk

DBMSs added abstract data, Referential integrity and triggers. The SQL specs grew from 100- 1500 pages within a couple of years, which was an attempt to make DBMS universal. Basically one size does not fit all. There are a list of markets that are not going for anything. For example, data warehouses, NoSQL, OLTP database etc. There is a wide diversity of engines. It's a great time to be a database researcher. Now is a great opportunity for new ideas and we expect to see many new implementations in the future. Elephants are trying to adapt without losing market share.

Advantages and Disadvantages

- The advantage of Stonebraker talk is it encourages database researchers to create many implementations. The market will be great for researchers in the future. Getting rid of useless applications like flash.
- Disadvantages is many vendors will have a difficult time staying current.