Big Data Paper Summary on MapReduce by Jonathan Spence 3/7/2017

- MapReduce paper - http://labouseur.com/courses/db/papers/dean.osdi04.mapreduce.pdf
- Comparison paper - http://labouseur.com/courses/db/papers/pavlo.sigmod09.comparison.pdf
- Stonebraker talk - http://kdb.snu.ac.kr/data/stonebraker_talk.mp4
- MapReduce Google's slides - https://research.google.com/archive/mapreduce-osdi04-slides/index.html
- MapReduce Wikipedia - https://en.wikipedia.org/wiki/MapReduce

Mapreduce Main Ideas

- Mapreduce is designed to process large amounts of data. It identifies similar data, and then combines that similar data.
- It does this over distributed computing network, coordinating the work automatically between hundreds or thousands of servers (nodes).
- A user/programmer does not have to do extra work to use the distributed computing, as it is done automatically, therefore it is easy to use.

How MapReduce is Implemented:

- The model is split into two main functions: mapping and reducing.
- The Map function takes an input pair consisting of a key and a value, and returns an intermediate key and value pair.
- The Reduce function takes the intermediate key and value pairs, and merges similar values.
- In the example from the paper of counting the occurrences of words in a file, the map function would note each occurrence of a word, and create a key and value pair for each one. The reduce function would group all of the words with the same key, counting the number of occurrences of a word in a file.

Analysis of Idea and Implementation

MapReduce is very cool in its ability to handle the distribution of large tasks of work over hundreds of machines, with no extra consideration needed by the user.

- Google’s implementation uses (or used) a combination of the Google File System along with MapReduce, similar to the use of Hadoop File System and MapReduce in Hadoop.
- MapReduce is not primarily a DBMS.

Main Ideas of Comparison Paper

- There are two approaches to large scale data analysis: MapReduce and parallel database management systems
- The paper argues that the while MapReduce may appear to provide a simple way to do complicated distributed tasks, parallel database management systems can be used to accomplish the same tasks.
- The two parallel database management systems tested, DBMS-X and Vertica, provided better performance than Hadoop (open source software which uses MapReduce). Though Hadoop was notably easier to set up and use than the parallel dbms solutions

How a Parallel DBMS is Implemented

- Parallel DBMS's have standard relational tables and SQL.
- In a parallel dbms, tables are partitioned over the nodes in a cluster, and the database uses an optimizer which translates the SQL code from the user into a query which executes over the multiple nodes.
- Similar to MapReduce, a user can interact with the parallel database at a high level, and not have to worry about the specifics of the distributed computing.

Analysis of the Ideas and Implementations in a Parallel DBMS

- Parallel database management systems seem to be more fully featured and advanced than MapReduce.
- A parallel dbms is capable of performing distributed computing tasks dealing with large amounts of data, just like MapReduce.
- Whereas MapReduce is not grand in its scope as a model, a parallel dbms contains both the mechanisms for database management as well as distributed data computing and management.

Comparison of Ideas and Implementations in the Two Papers

- In comparison to MapReduce, it is much more structured when dealing with pieces of data and information.
- A parallel dbms is first and foremost a database, therefore its first job is to specify the rules for the organization of the data.
- MapReduce does not specify these rules, because it is not first and foremost a database. My understanding is that what it emphasizes is its ease of use and its focus more on the physical side, handling hardware failures and the physical storage of the data over hundreds or thousands of nodes..

Main Ideas of Stonebraker talk

- From the inception of relational databases management systems they were perceived to be the answer to any and all database needs. Not much advancement occurred in the field because the thought was that relational databases were the best solution for anyone's needs.
- There are now a variety of database solutions in place to better achieve specific tasks, such as for data warehouses or online transaction processing, or analytics
- There are a variety of different database management systems using different that exist and are being created. They are disrupting the field of database management systems.

MapReduce in context of Stonebraker talk

- The relational database management system is great, but that is not where data storage technology ends, as the relational model does not handle the logistics of dealing with massive amounts of data.
- MapReduce is one such model that was created in this field to fit a specific need, as Google's needs required a model which placed more emphasis on automating the interaction between warehouses of servers and the data they store than on proper database rules.
- While the "one size fits all" idea is dead, it appears that Stonebraker and others argued in their paper that a parallel DBMS was a better performing solution than MapReduce.