Big Data Summary

-Hive

-A comparison of approaches to a large scale data analysis

-Michael Stonebraker's ICDE 2015

A presentation by Ronald Cavaliere 3/7/17

The Main Idea Behind Hive

As a result of modern data set sizes and increasing expenses in the business intelligence industry, there was a dire need for a higher level of Hadoop. Hadoop is an open source, MapReduce implementation system being used in order to store and analyse these large data sets for big companies like Yahoo, and Facebook. What Hive does is add that extra layer on top of Hadoop. It gives Hadoop another necessary level to control this data by supporting SQL like languages. Writer's using Hive can turn SQL queries into MapReduce functions making it easier for everyone in the end.

The Implementation of Hive

Hive is the best friend of those who are not familiar with MapReduce structures such as Hadoop. It gives them the chance to write queries in a language that is very similar to SQL, HiveQL, and still have them executed in Hadoop. Hive is built by the following components which all have specific purposes.

➔ **Metastore**

Stores system catalog and metadata about tables, columns, etc.

➔ **Driver**

Oversees the lifecycle of a HiveQL query as it goes through the entire process

➔ **Query Compiler**

Turns HiveQL statements into MapReduce jobs.

➔ **Execution Engine**

Executes jobs created by the Query Compiler.

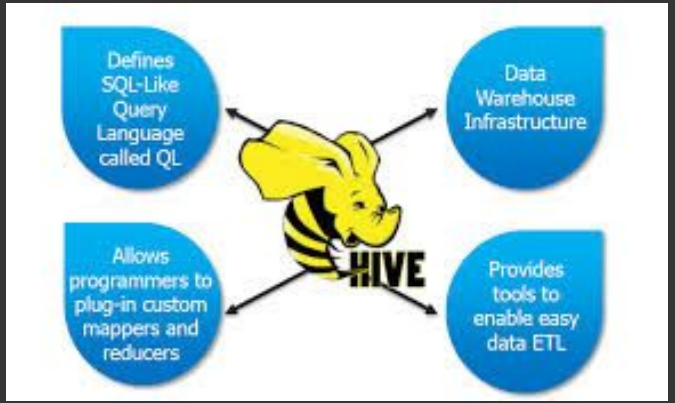
➔ **HiveServer**

Is the reason why Hive can intermingle with other applications.

The Analysis of Hive

Hive makes the lives of end users, especially those who were not familiar with map-reduce structures, much simpler than it once was. A task that used to take more than a day to complete for end users can now, because of Hive, be completed within a couple of hours. The main reason behind this was the lack of familiarity of modern query languages such as SQL. Hive has brought the wonders of using tables in relational database models to the “world of Hadoop”.

Using Facebook as an example, Hive has succeeded in accomplishing its mission. As the article states on page 1004, Hive has cut costs tremendously for Facebook. It has given them the chance to provide data processing services to engineers and analysts in a way the traditional processes could not. It was much easier and cost-efficient. On top of this, Facebook strongly believes that Hive shows the potential of growth and long term success for the future.



A Comparison of Approaches to Large-Scale Data

This paper analyzes and compares the structures and ideas behind the MapReduce model for big data analysis and the traditional framework of database management systems. The authors perform this comparison by researching different performance measures and the development complexity of both models. Overall, the main idea behind this paper is to take the advantages of both models and suggest possible implementation concepts for future systems.

MapReduce is a data processing model for large data sets that is continuing to gain popularity. Based on what is inputted and the desired task at hand, MapReduce will return a set of records.

Implementation of the Comparison

The comparison of both models was accomplished by performing 5 different tasks, all which were averages of three testings. The 5 tasks were as follows:

Data Loading- The relationship between nodes used and seconds to load were directly related. As nodes increased, the time to load did as well. This test shows Hadoop is the worst (figure 3).

Selection Task- Both DBMS out performed Hadoop again (figure 6).

Aggregation Task- This was used to evaluate when nodes exchange intermediate data to compute the final value. Both DBMS outperformed Hadoop for the third time (figure 8)

Join Task- This was assigned to ensure the systems were able to handle complicated tasks for large data sets. Once again both DBMS outperformed Hadoop by a large margin (figure 9).

UDF Aggregation Task- This was used as one of the 5 tasks in order to compute the inlink count for separate documents within the set. Unlike the others, one of the DBMS and Hadoop performed consistently however as the number of nodes increase the time of the return is slower because of the large number of output data (figure 10).

Figure 3

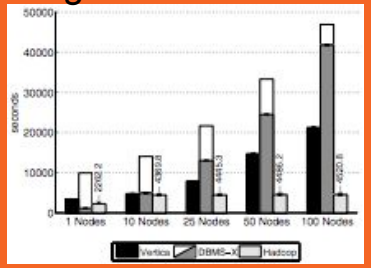


Figure 6

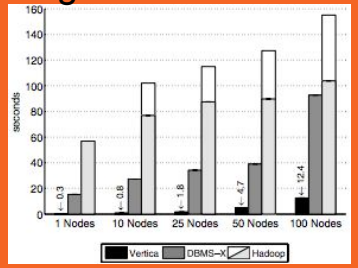


Figure 8

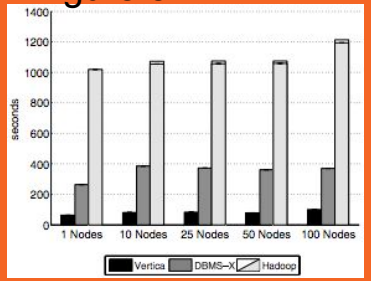


Figure 9

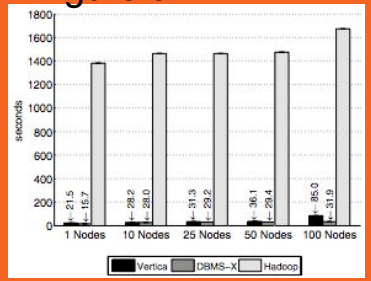
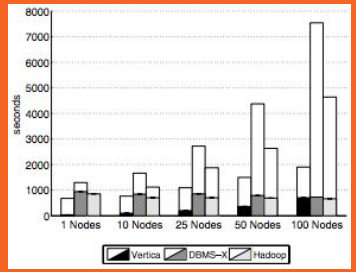


Figure 10



Analysis of the Comparison

After analyzing all of the tasks the authors found that both of the DBMS models had an overwhelming advantage over Hadoop. Averaging the three scores for all five of the tasks at 100 nodes, DBMS-x was faster than the MapReduce model by 3.2 times. Vertica was faster than DBMS-x by 2.3 times the amount. What this means is that both parallel database systems would not use as much energy as the MapReduce model. This is a result of their use of clustered indexes. Also, it is safe to say that these results are because of the development of different technologies over the years. MapReduce is relatively new therefore will take time to catch up to speed.

Comparing Hive...

- Hive was created...
  - In order to improve the performance of the newly structured Hadoop MapReduce system.
  - Ease the lives of end users and experienced SQL users.
  - It worked for facebook by saving tons of time and money for their data processing systems.
- Since MapReduce models are very complex sometimes, they are difficult to update and reuse.

To the Comparison Paper.

- Analyzed three different systems
  - DBMS-x
  - Vertica
  - MapReduce
- MapReduce returned the slowest times out of all three after multiple tasks.
- This paper had the objective of comparing all three systems to make suggestions on what future systems should consist of.

Stonebraker's Discussion

- ➔ New systems that handle big data sets faster are taking over for DBMS.
- ➔ Row stores are slower than column stores.
- ➔ Data scientists will replace business analytics as complexity continues to increase.
- ➔ SQL is getting slower and slower
- ➔ Just like the comparison paper, Stonebraker expects there to be a lot of new implementations to future systems.

Hive: Pros and Cons

Pros

- ➔ Similar to SQL so code that was already written can be reused.
- ➔ Storage and executing queries is handled by Hadoop so professionals don't have to spend time worrying about that.
- ➔ Professionals can manipulate data sets.
- ➔ The structure is clear. Jobs are broken down specifically.

Cons

- ➔ Hadoop is much slower than other data processing systems therefore it's a con that Hive only works for Hadoop.
- ➔ Hive does not have update, insert, or delete operations therefore it is not fully capable of doing what SQL can do.