Hive A Petabyte Scale Data Warehouse Using Hadoop
And
A Comparison of Approaches to Large-Scale Data Analysis

Joseph Gust
3/7/2017

The main idea of this paper was the creation of HIVE, an open-source data warehousing solution built on top of Hadoop, which is an open-source implementation of map-reduce. Hadoop could be difficult to use, which was one of the main reasons for the creation of HIVE.

In order to make HIVE easier to manage, they created a new language, HiveQL. HiveQL borrows heavily from SQL, but it’s meant to maintain the increased extensibility and flexibility from hadoop.

The comparison paper attempts to determine the pros and cons of using different map reduce methods vs. parallel SQL DBMSs. It takes in factors such as the speed for initially loading the data and speed it takes to complete various tasks.

They ran various tests to check the speed of initially loading the data, and running several tasks. Hadoop was quicker when it came to loading that data, but significantly slower at task execution. When they used a select statement, Hadoop took significantly longer than the parallel SQL DBMSs. When asked to perform an aggregate function, Hadoop was left in the dust once again. For performing a join statement, Hadoop was vastly slower than the other two, but for the UDF Aggregation task Hadoop was faster than DBMS-X and much slower than Vertica. So despite Hadoop's excellent speed for initially loading the data, it's much slower than the other two when it comes to task execution.

As we can see from the comparison paper, the Hadoop(map-reduce) was much faster at loading the data than the parallel SQL DBMS counterparts. On the other hand Hadoop was much slower than the other two when it came to task execution.

The Hive paper showed how they've gone about trying to make hadoop easier to manage and how they've tried to improve the efficiency of Hive. The comparison paper showed some of the downfalls of hadoop, as well as its improvements vs parallel SQL DBMSs.

The Stonebraker talk was primarily about how traditional Relational Databases are losing their effectiveness. It used to be “one size fits all,” but Stonebraker believes it’s now “one size fits none.” Also, companies are moving from row stores to column stores since they’re quicker and more efficient. He goes over some NoSQL databases that are becoming popular, showing that we’re moving away from the classic relational database that has been used for decades.

The HIVE paper goes over a form of map-reduce that's well adapted for the needs of large companies such as facebook. As we know from the stonebraker talk, relational databases can't always be the solution, so it's great that companies are starting to diverge from the classic SQL DBMSs. Perhaps in the near future we can come up with a database model that better suits our needs.