

“Hive – A Petabyte Scale Data Warehouse Using Hadoop”

By: Robert Coords

Date: May 7, 2015

Bibliographic Data

- “Hive – A Petabyte Scale Data Warehouse Using Hadoop”
 - <http://www.labouseur.com/courses/db/papers/thusoo.icde2010.hive.pdf>
- “A Comparison of Approaches to Large-Scale Data Analysis”
 - <http://www.labouseur.com/courses/db/papers/pavlo.sigmod09.comparison.pdf>
- Michael Stonebraker on his 10-Year Most Influential Paper Award
 - http://kdb.snu.ac.kr/data/stonebraker_talk.mp4

Hive's Main Ideas

- “Hive – A Petabyte Scale Data Warehouse Using Hadoop” is about why Hive was designed, the design choices made by the Hive team, and how Hive functions when it is implemented.
- Facebook adopted Hadoop to provide additional scalability. However, their end users had trouble with Hadoop, as it did not support SQL and forced these users to write complex map-reduce programs to perform simple tasks.
- Hive was designed by Facebook to fix these issues, and was built on top of Hadoop to allow scalable analysis of big data.

Hive's Implementation

- Hive's implementation removed the complexity facing end users, while allowing scalable analysis of big data through Hadoop.
- Hive added HiveQL, fixing the issue of Hadoop's lack of SQL support and making Hive more accessible to end users.
- Map-reduce functions were created/ performed by compiler/ execution engine whenever possible, making programming easier for end users.
- Hive was built on top of Hadoop, allowing Hive to keep Hadoop's extensibility and flexibility.

Analysis

- Hive was designed by Facebook to make life easier for the end user, and they seem to have succeeded. SQL is a language that nearly every programmer knows, and adding HiveQL (which is based on SQL) can only make Hive (and Hadoop) more accessible. The benefits Hive receives from Hadoop when doing scalable analysis on large data sets is also great.
- One negative design feature I noticed regarding Hive is its limitations regarding inserts. This comes from Facebook's culture, where new data is placed in a partition and joined to the rest of the database every hour or day. Hive does not support the commands 'INSERT INTO', 'UPDATE', and 'DELETE' because of this. However, I believe that, as Facebook grows and Hive is used more often, these commands will be added and supported.

Main Ideas from “A Comparison of Approaches to Large-Scale Data Analysis”

- The paper compares the MapReduce paradigm for large-scale data analysis to the traditional, parallel Database Management System. The comparison is based mostly on system architecture and performance.
- From an architecture perspective, MapReduce works better on smaller projects, due to its design choices.
- The parallel Database Management Systems they tested were able to outperform the dominant MapReduce system, Hadoop.
- Although both systems had strengths and faults, the authors seemed to prefer the parallel Database Management System model.

Implementation Comparison

7

Parallel Database Management System (DBMS)

- DBMS follows the relational paradigm for structuring data.
- DBMSs use hash or B-trees for indexing. Adds an overhead cost while loading data.
- DBMS uses SQL for querying data.
- DBMS has poor fault tolerance. When part of a transaction fails, the entire transaction must restart.

MapReduce (MR) Paradigm

- MR systems can use any schema to organize data, or give the data no structure.
- MR programmers create the system's indexes. Allows MR to load data faster.
- MR programmers write low-level algorithms for queries.
- MR has good fault tolerance. When part of a transaction fails, that part is re-executed on a separate node.

Analysis of Both Approaches

Parallel Database Management System (DBMS)

- DBMS seems to be a better fit for business environments since it provides structure with the relational paradigm, has built-in indexes that will not vary from programmer to programmer, and uses SQL instead of forcing the programmer to write low-level algorithms for queries.
- DBMS's ability to out perform MR proves that it will continue to be commonly used.

MapReduce (MR) Paradigm

- MR has some interesting ideas, but its faults outweigh its strengths.
- The authors noted that MR was easier to set up and use than a DBMS. Although, this would benefit the user, I believe many customers would choose DBMS over MR, because of MR's inferior performance.
- I believe that MR's lack of SQL support also acts as a deterrent for future users.

Comparing the Two Papers

9

“Hive – A Petabyte Scale Data Warehouse Using Hadoop”

- Hive was built to correct many of the issues with MapReduce explored in the other article.
- Hive incorporated HiveQL, which added SQL support to MapReduce.
- Hive’s compiler helped remove the need to write low-level algorithms when querying a MapReduce system.
- Hive is a middle ground between MapReduce and parallel database management systems. This may help Hive surpass both system’s popularity in the future.

“A Comparison of Approaches to Large-Scale Data Analysis”

- Concepts from parallel database management systems can be seen in Hive. HiveQL is one example. Another is how Hive system catalog is stored in a relational database management system.
- MapReduce does not provide a schema for structuring data. This slowed down MapReduce in performance tests. Hive uses a schema similar to a relational database management system, which may improve Hive’s performance.

Main Ideas from Stonebraker Talk

- Michael Stonebraker's talk was about how traditional Relational Database Management Systems are no longer the answer for data storage.
- Traditional Relational Database Management Systems use row stores.
- Stonebraker says that row stores will either be replaced by column stores or a different paradigm.

Advantages and Disadvantages of¹¹ Hive in context of Stonebraker Talk and “A Comparison of Approaches to Large-Scale Data Analysis”

- Hive's advantages over MapReduce are that it implements SQL through HiveQL, making it easier for the user to write queries.
- Hive's disadvantage next to parallel Database Management Systems (DBMS) is that it is built on top of Hadoop. Given that Hadoop could not out perform the DBMSs, Hive is probably slower than the DBMSs as well, because Facebook did not try to improve Hadoop's speed with Hive.
- Hive's schema is based on the Relational Database Management System model, which Stonebraker said will soon be replaced in his talk. This is a disadvantage.
- Hive probably uses a row store, which Stonebraker said will soon be replaced by column stores. This is another disadvantage.