Big Data Paper Summary

Hive – A Petabyte Scale Data Warehouse Using Hadoop

Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Ning Zhang, Suresh Antony, Hao Liu and Raghotham Murthy

Facebook Data Infrastructure Team

A Comparison of Approaches to Large-Scale Data Analysis

Andrew Pavlo Erik Paulson Alexander Rasin Brown University University of Wisconsin Brown University pavlo@cs.brown.edu epaulson@cs.wisc.edu alexr@cs.brown.edu Daniel J. Abadi David J. DeWitt Samuel Madden Michael Stonebraker Yale University Microsoft Inc. M.I.T. CSAIL M.I.T. CSAIL dna@cs.yale.edu dewitt@microsoft.com madden@csail.mit.edu stonebraker@csail.mit.edu

Matthew Rahtelli

15 March 2016

Hive - A Petabyte Scale Data Warehouse Using Hadoop

- Substancial companies like Facebook, Yahoo, etc. were encountering a problem with large scale data processing infrastructure taking days to complete with the traditional RDBMS storage
- The companies were able to utilize “Hadoop”, an open source project, already processing data on a petabyte scale, which reduced data processing to a matter of hours.
- After implementing Hadoop, another significant problem occurred. It would take end users hours or days to write programs for simple analysis.
- To combat this final problem, Hive, (an open-source data warehousing solution), was created to bring familiar concepts and a language like SQL to run on top of Hadoop.

How the Idea is Implemented

- The data is stored in three different types: tables, partitions and buckets. Tables are stored in the directory. Partitions of a table are stored in a sub-directory. Buckets are stored in a file within the partition's or table's directories.
- Hize supports the following datatypes: Primitive - (integers, floats, strings), and complex types - (maps, lists, structs).
- The language of HiveQL brings traditional SQL constructs and enables end users to start to have a Command Line Interface right away.
- Hive compiles the queries into map-reduce programs which are able to be handled by Hadoop.

Analysis of Implementation

- These companies encountered a large scale data problem that was of dire need to fix. Hive enables end users to be able to migrate to this format easier.
- Keeping the traditional constructs of SQL but also having some of its own was a great way to solve the problem of end users taking days to write programs.
- Having the queries compile into map-reduce programs seems to be the best fix since Hadoop supports it.
- Keeping the same types of data storage and types again makes it easier for end users to migrate over and understand more about what Hive is.

Comparison of Approaches to Large-Scale Data Analysis

- There are two architectures: MapReduce (MR) and Parallel Database Management Systems (DBMS.). It compares the architecture between these two different systems of data analysis.
- In terms of preformance, Parallel DBMS can outpreform MapReduce. However, MapReduce is easier to implement and work on smaller types of projects since preformace is less of an issue due to scale.
- Parallel DBMS seemed to be the best system to go with because there are always going to be problems with every system, and one is always going to be better than the other in some aspect.

How the Idea is Implemented

- MapReduce does not need any structure or organization while Parallel DBMS requires that data fits into a relational paradigm.
- Fault Tolerance: MapReduce is more adept to dealing with failures and can automatically restart the task. Parallel DMBS restarts completely in the event a failure occurs.
- The language that Parallel DBMS uses is SQL which is familiar by many end users, but for MapReduce, the users must use other programming and will be a pain to upkeep for maintenance.
- There is more flexibility in MapReduce but that comes with more upkeep of the system. The standard Parallel DBMS with SQL gives less flexibility for accessibility.
- DBMS indexing takes the form of Hash or B-Tree and MapReduce does not provide indexes and relies on the end user to create it.

Analysis of Implementation

- DBMS is better with data structure so the end user is able to understand the data or information so you are able to make use of it and analyze it.
- MapReduce is better with fault tolerance because the whole query does not have to be restarted, only the specific section that failed.
- DBMS uses SQL which is familiar by many end users and less upkeep.
- DBMS provides the indexes which is something that does not have to fall on the end user to create which is better in the long run in terms of focusing on other issues and preformance.

Comparison of Ideas and Implementations of Both Papers

- Hive utilized the open source project of Hadoop, the pre-existing data processing open source project. It sped up preformance, was easy to use, and easy to implement.
- DBMS or DBMS is a great system, better than MapReduce or MR, and can be seen in Hive as an example. Instead of a parallel DBMS, Hive utilizes a relational DBMS which is must faster.
- MapReduce has no flexibility and requires more upkeep in terms of being able to analyze data. MR does not provide indexes and leaves that upon the user. Hive seems to be the most logical choice for performance and accessibility.

Main Ideas of the Stonebreaker Talk

- The initial answer to databases was Relational Database Management Store (RDBMS) which was realized by 2005, that it was wrong route for storage.
- By 2005, C-Store was being worked on with column store which looked nothing like row stores, (one size did not fit all).
- This creates great new opportunites for data, most large scale vendors will transition to using this type of DBMS, and how column stores are going to replace the original row stores.

Advantages and Disadvantages of the Main Idea of the Chosen Paper in the Context of the Comparison Paper and the Stonebraker Talk

- Advantages:
- Hive uses the language similar to SQL called HiveQL which makes it easier for end users to understand and transition easier.
- The talk showed that things are always going to change as time passes and Hive leaves less reliance on the user and more capability in the system itself.
- Disadvantages:
- Hive is built on top of Hadoop and is not its own system. This probably affects performance issues.
- Hive uses the RDBMS model which in one of the papers is one of the slower models. In the talk, Stonebraker mentioned that RDBMS is making its way out and that markets will convert to better or the newer model.