Hive: A Petabyte Scale Data Warehouse Using Hadoop

A Comparison of Approaches to Large-Scale Data Analysis

Michael Stonebraker on his 10-Year Most Influential Paper Award at ICDE 2015.

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

Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden, Michael Stonebraker

By: Trevor Pirone
October 29, 2017

Hive: Main Idea

- Hive is an open-source data warehousing solution built on top of Hadoop.
  - Hadoop is a popular open-source map-reduce implementation used by companies to store and process extremely large data sets on commodity hardware.
- Hive is supported by HiveQL, a similar language to SQL, which enables users to plug in custom map-reduce scripts into queries.
- HiveQL queries are compiled into map-reduced jobs and exceuted by Hadoop.
- Hive's model is difficult to maintain as programmers and developers have to write their own programs.
  - Not only is it hard to maintain and reuse the programs, it can be hard to teach new users to operate with Hive.
- Hive has its own system catalog called *Metastore*.
  - It contains schemas and statistics.
    - They are useful in data exploration, query optimization, and query compilation.

Hive: Implementation

 Hive has seven main components:
- Metastore – contains the system catalog, the schemas, and metadata about tables, columns, statistics, and partitions
- Driver – Responsible for managing the lifecycle of a HiveQL statement as well as maintaining session handles and statistics
- Query Compiler – The component that manages, compiles and processes HiveQL statements
- Execution Engine – Executes the tasks put forth by the query compiler (i.e. HiveQL statements) and interacts with Hadoop
- HiveServer- It provides an interface and a JDBC/ODBC server which allows other applications to be integrated with Hive
- Client components such as CLI, web UI, and JDBC/ODBC driver
- Extensibility Interfaces – SerDe interface which is based in Java, ObjectInspector which works with SerDe to get types from a jar file, and the User Defined Function and User Defined Aggregate Function interfaces for custom functions

Hive: My Analysis

 Hive is very similar to traditional databases, but has restrictions because it is built on Hadoop and MapReduce.

 Hive only accepts a subset of SQL as valid queries.

 It is good that Hive is open-source and is being used by big companies such as Facebook.

 A nice support range of primitive and complex types, but there is definitely more types that should be added in future development.

 I think that Hive is a good idea. Very modern thinking approach, but it needs some more development before it can be optimized and used as much as traditional database systems.

A Comparison of Approaches to Large-Scale Data Analysis: Main Idea

- The authors evaluate two paradigms, MR and DBMS, for large-scale data analysis.
- MapReducing, in comparison with DBMS, consists only of two functions, called Map and Reduce, that are written by a user to process key/value data pairs.
- Each paradigm undergoes a series of tests and the results are analyzed.
  - Some test categories included data loading, task execution, selection task, aggregation task, and join task.
- The results showed that DBMS outperformed MR in almost every single category that was tested and analyzed.

A Comparison of Approaches to Large-Scale Data Analysis: Implementation

- Top conduct the experiment, Hadoop was chosen to represent the MR paradigm while DBMS-X and Vertica were used for the DBMS paradigm.
- The authors performed a series of tests to compare the architectural elements and performance benchmarks of each paradigm.
  - Tests in flexibility, fault tolerance, benchmark testing, and load times were all conducted in the experiment.
- DBMS ends up outperforming MR in every test category except for Grep Tasks.
- DBMS had better overall performance because in contrast to MR's use of two function to compute data, DBMS uses index systems which is a lot more efficient.

A Comparison of Approaches to Large-Scale Data Analysis: Analysis

- As mentioned before, the authors concluded through a series of tests that DBMS performs much better than MR.
- They went on to explain why the results favored DBMS, but they provided positive insight on MR.
  - MR was easy to configure, install, and set up. In comparison, the DBMS configurations were difficult to set up and Vertica was temperamental to certain system parameters.
  - MR does a superior job of minimizing the amount of work that is lost when a hardware failure occurs at a cost in performance, however.
  - MR is more extensible. Neither of the parallel systems tested did a good job on the UDF aggregation tasks.
  - MR has a much lower upfront cost and does not require as much manpower to operate.
- I learned a lot from reading this paper. Although I have experience working with DBMS, I would like to try and experience querying on MR.

Comparison of Hive and A Comparison of Approaches to Large-Scale Data Analysis

- Hive is based off of Hadoop which uses an MR paradigm.
- In comparison to traditional DBMS, Hive and Hadoop are looking to improve on making the older systems better.
- As of right now, MR systems are still behind in terms of overall performance when compared to DBMS.
- Businesses may look to switch from DBMS in the future if MR improves its performance and maintains a low cost.

Michael Stonebraker on his 10-Year Most Influential Paper Award at ICDE 2015: Main Ideas

- Stonebraker mentions a "One size fits all." in the context of relational databases.
  - Relational database model systems were built with the intention of possibly becoming the ultimate solution to managing database systems.
  - Unfortunately, there is no existing relational database table that fits the needs of everything existing in the universe.
- The Relational Database Model is the most widely used model and is adopted by many companies.
  - There exists other models and paradigms such as Map-Reduce.
  - However, big companies will not abandon relational models until the competitors can begin to compete with the relational model in terms of performance and adaptability.
- Major vendors are moving to column stores which are two orders of magnitude faster than row stores.
- Today, there are a wide range of engines available for use!

Advantages vs. Disadvantages
Hive vs. Comparison Paper and Stonebraker Talk

 Hive is open-source meaning that anyone in the community can contribute to improving its flaws.

 It has a fully incorporated failure model more sophisticated than DBMS.

 Hive, running off of Hadoop, has an upfront cost advantage.

 It can read and compile from files.

 Not created for large business use (Business Logic).

 Performance issues due to Hadoop (slow query execution times).

 Not as flexible as DBMS.

 Cannot insert data into pre-existing tables.