Hive - A Petabyte Scale Data WareHouse Using Hadoop

Facebook Data Infrastructure Team. Hive - A Petabyte Scale Data Warehouse Using Hadoop. Facebook. Web. 19 October 2016

A comparison of Approaches to Large-Scale Data Analysis

Pavlo, Andrew, Erik Paulson, Alex Rasin, Daniel Abadi, David DeWitt, Samuel Madden, Michael Stonebraker. A Comparison of Approaches to Large-Scale Data Analysis Web. 19 October 2016

Michael Stonebraker's ICDE Talk About his "10 Year Test of Time"

Nicholas Barranco
October 19th, 2016

What is Hive?

- Due to the rapid growth in which data sets are measured in, the process for warehouse solutions is impractical as well as expensive.
- Hadoop is a popular map-reduce implementation, and is open-source, the downside to is that it requires a lot of upkeep ultimately requiring more time by programmers to perform maintenance.
- Hive is another open source data warehouse implementation that uses Hadoop.
- Hive uses HiveQL, which is similar to other SQL languages.
- It compiles into map-reduce jobs, which then run on Hadoop.
- Hive uses an implementation of Serialization/Deserialization (SerDe) java interface when the user associates the provided one to a table.
- It contains: Metastore, a system catalog.
  Driver, manages the lifestyle.
  Query Compiler, the component that compiles the HiveQL
  Execution Engine, the part that runs tasks created by the compiler;
  Hive Server, that has a thrift interface and a JBDC/ODBC server and helps integrate Hive amongst other applications.

Implementation of Hive

- Facebook uses, with 5TB (15 after replication) of compressed data added daily.
- Runs on Hadoop, so it is not fully distinctive.
- Hive can take implementation of SerDe java interface.
- Due to Hadoop’s questionable efficiency, Hive was well-accepted after implementation.
- Uses HiveQL, which is similar to any other SQL languages. Very similar and almost identical in some ways.
- Open-Source project that was easily adapted to and is a work in progress.

Analysis of Hive

- Much more efficient than previous process. Cheaper and uses less man power.
- Lack of inserting into pre-existing tables to be a problem, but has apparently not caused an error yet.
- Great support amongst SerDe, File Formats, and Data Storage.
- Likeable if you like the SQL like language.
- Its also a great effort towards moving towards more productive and cost efficient methods in the future, it is not the end all be all.

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

- It's primarily about MapReduce(MR) which is a method that is used regularly when it comes to managing Big Data. MapReduce allows the input of data sets to be stored to a partition which are then deployed to each node via a file distributer.
- The paper is trying illustrate the difference of the MapReduce approach to Business Intelligence while comparing to other similar systems that were also made for this reason.
- Database Management Systems (DBMS) is another alternative to MapReduce and has a few benefits over it.
- The biggest benefit DBMS has over MR is that DBMS has support for standard relation tables and SQL (Structured Query Language)
- The biggest downside of DBMS over MR is its higher upkeep in both manpower and cost.
- The paper concludes that DBMS performs significantly better than MR, with DBMS-X and Vertica it performed twice as fast than Hadoop did.

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

- To test the systems the writers compare the two systems through a series of tests that would test the Flexibility, Performance, and Fault Tolerance of the Systems.
- DBMS-X and Vertica were used for the DBMS testing.
- Hadoop was used for MR testing.
- The results of the test as shown in the paper shows that's DBMS dominates MR in almost every regard beating Hadoop in almost everything but in the Grep Tasks testing.
- Through the tests we are shown that DBMS is much more reliable in most regards when compared to MR because DBMS uses an indexing systems while MR has to perform two tasks of Mapping and Reducing.

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

- The tests that were chosen by the writers were chosen because for MR to perform the tests it would need both the functions of MR to perform.
- Ultimately to make it a fair demonstration it would need to use both functions to be useful
- In the end the results of the table show that MR is outclassed by DBMS in almost every test.
- This is because DBMS executed queries by having the nodes scan local tables and extract the necessary fields.
- Those queries that are executed are then merged, without using a much resources as MR.

Comparing the Two Papers

Hive - A Petabyte Scale Data Warehouse Using Hadoop

- When comparing both Hadoop and Hive you have to remember that one is built off of the other.
- The paper discussed how the improvements on the new systems better the old system and it flaws.
- It also made clear that it was an improvement of an older system through usage and syntaxes.

A Comparison of Approaches to Large-Scale Data Analysis

- Compares two similar systems for Business Intelligence and Big Data analysis.
- Went over the advantages of the systems and testing them through the theory of their systems and various testing.
- Discussed the results of both systems and went over the various differences between the systems and the performance of the systems when used in the various tests.

Stonebreaker Talk - Main Ideas

- Relational databases are supposed supposed to be “One Size Fits All” according to Stonebreak who says the idea has “come and gone”
- Originally RDBMS provides abstract data types, referential integrity, as well as triggers so it could be the universal solution to database models.
- There was no place for streaming applications within the traditional row-store of RDBMS implementation.
- Ultimately was impossible to follow the “One Size Fits All” based upon the 3 examples in Stonebreaker’s paper.
- One Size Fits None: “DB2, Oracle, Sequel Servers are good for nothing.”

  -*Data Warehouse Markets* are moving towards Column Stores because they are two orders of magnitude faster.

  -*OLTP Market* are generally moving towards main memory deployments with lightweight transactions which have been proven to be useful for the OLTP markets.

  -*NoSQL Market* have around 100(or so) vendors. With a plethora of data models and architectures. (Compared to a potpourri) With no standards at all.

Stonebreaker Talk - Main Ideas Cont.

- One Size Fits None: “DB2, Oracle, Sequel Servers are good for nothing”

  -*Complex Analytics*: Business intelligence(BI)using data warehouse. Capable of performing basic and standard data analytics. Regressions, Data Clustering Predictive Models, the BI of the future are defined through Arrays and will rarely include tables at all.

  -*Streaming Market* is not based upon the traditional row stores. In general OLTP generally will have a greater market share in this area. Can add streaming to an OLTP engine much easier than adding persistence to a streaming engine.

  -*Graph Analytics Market* simulates a column store architecture or array matrix. Generally traditional Row stores are not ideal for this.
- There is a **“YUGE”** diversity of engines all of which are specialized in a unique way. Traditional row stores doesnt meet any of these criteria.
- NVRAM, Big Main Memory, Processor diversity (Nvidia GPU,s Numa, Xeon/Phi) Higher Network Speeds LLVM, Vectorization are advancing at an astonishing rate and allows for new architecture to emerge that will take advantage of the new technologies.

Advantages and Disadvantages of Hive

.Advantages:

- Failure model incorporated.
- Easy join functionality as well as multiple uses for one system.
- Compiles and reads from files.
- Constantly evolving since it’s a work in progress.

Disadvantages:

- Very Strict
- Only Structured Data
- Not for Business Logic
- Inability to insert into pre-existing tables but contains other methods to compensate.

SEE YOU SPACE COWBOY...