Big Data Paper Summary

Connor Sargent May 7th 2015

Hive – A Petabyte Scale Data Warehouse Using Hadoop

A Comparison of Approaches to Large-Scale Data Analysis

Pavlo, Andrew, Erik Paulson, Alexander Rasin, Daniel J. DeWitt, Samuel Madden, Daniel J. Abadi, and Michael Stonebraker. "A Comparison of Approaches to Large-Scale Data Analysis." (2009): n. pag. Web. 7 May 2015.

Thusoo, Ashish, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Ning Zhang, Suresh Antony, Hao Liu, and Raghotham Murthy. "Hive – A Petabyte Scale Data Warehouse Using Hadoop." (2010): n. pag. Web. 7 May 2015.

Main Idea of the Hive

- Hive is a solution to the map-reduce implementation problem that occurs in Hadoop when dealing with processing large data sets on commodity hardware.
- Hive supports queries expressed in a SQL-like declarative language - HiveQL, which are compiled into map-reduce jobs that are executed using Hadoop.
- Hive stores data in tables similar to traditional databases and supports the common primitive types as well as several complex types such as associative arrays, lists, and structs.
- It uses Hive Query Language(HiveQL) which is similar to SQL, but with several extensions, enabling anyone familiar with SQL to begin querying the system immediately.
- The Hive extensions support analysis expressed as map-reduce programs by users and in the programing language of their choice.
- There are some limitations to Hive, including how inserts are done, where inserting into an existing table or data-partition is not supported and will overwrite all existing data.
- Limitations of Hive have not shown many issues because queries can be expressed as equi-joins and the frequency in which new data is loaded means new partitions of tables are inserted daily or hourly regardless.

Implementation of Hive

- Hive implementation is fairly simple because of its similarities to traditional databases, as well as HiveQL's similarity to SQL.
- Hive is implemented in Facebook to enable an extremely large workload on the Hadoop cluster in Facebook because of the simplicity with which adhoc analysis can be done.
- Hive consists of 5 main components.
  - Metastore. stores the system catalog and metadata about tables, columns, and partitions.
  - Driver. manages the life cycle of a HiveQL statement as it moves through Hive.
  - Query Compiler. Compiles HiveQL into a directed acrylic graph of map-reduce tasks.
  - Execution Engine. Interacts with Hadoop to execute the tasks produced by the compiler in proper dependency order.
  - Hive Server. Provides a way of integrating Hive with other applications.
- The large amounts of data that are processed, which is about 75 TB of data per day, consists of adhoc queries and dashboard reports.
- Hive enables this much data processing because of the simplicity in which adhoc analysis can be done on the Hadoop cluster.
- Challenges with the system include sharing the same resources as adhoc users and reporting users, causing untuned data to consume valuable cluster resources.

Analysis of Hive

- Hive is an open source data warehousing solution designed to appeal to programmers that are already comfortable using SQL
- It is able to handle extremely large amounts of data, which is imperative for a company like Facebook, and provided processing and analytics for the data.
- Hive supports easily running analytical queries in big data.
- It has some limitations in how the data can be manipulated, but compensates with the more streamlined use in big data analytics.
- Large scale reporting of data using a query language that is already known, since HiveQL and SQL are very similar, is done through compiling Hive queries into map-reduce jobs that are executed using Hadoop.
- Hive is used as an improvement to Hadoop’s map-reduce implementation for large scale data processing without the hassle of using a new query language, or having programmers create custom programs that are difficult to maintain and reuse.

Main ideas of comparison paper

- Map- reduce and Parallel DBMS’s are similar, but differ in several key areas.
- Map-reduce is very simple in comparison to parallel DBMS’s because it only consists of two functions, map and reduce. The map function reads a set of records from an input file, does any desired filtering and transformations, and then outputs in the form of new key or value pairs.
- Parallel DBMS’s support standard relational tables and SQL, enabling data to be stored on multiple machines. Parallel execution is enabled by most tables being partitioned over the nodes in a cluster, then the system uses an optimizer that translates SQL into a query plan whose execution is divided amongst multiple nodes.
- Hadoop outperforms DBMS-X and Vertica in load time due to the fact that each node is simply copying each data file from the local disk to the local HDFS, and distributing two replicas to other nodes in the cluster
- DBMS-X load times do not decrease as less data is stored per node, but execution time of the second phase of the loading process is cut in half when twice as many nodes are used to store the data.
- Hadoop has the slowest task execution in comparison to Vertica and DBMS-X, caused by it’s start up costs dominating the execution time.
- Hadoop and DBMS-X perform relatively the same with increased amount of data processing, but still slower than Vertica because of its ability to become more effective as more data is stored per node.

Implementation of the ideas

- Hadoop is able to be installed easily and run jobs with little effort due to installing requiring minimal steps. The steps include setting up data directories of each node and deploying the system library and configuration files.
- DBMS-X Has a straightforward installation process, but was difficult to configure in order to start running queries due to the fact that each nodes kernel was configured to limit the total amount of allocated virtual address space.
- DBMS-X showed to be ineffective for adjusting memory allocations, as well as being easily able to lock the user out withotu a failsafe mode to restore to a previous state.
- Vertica was also relatively easy to install, but there is no explicit mechanism to determine what resources were granted to a query, nor is there a way to manually adjust per query resource allocation.
- Hadoop is easier implementation in proper installation and configuration for immediate use.
- DBSM’s excel in the fact that the tuning that is needed is mostly done prior to query execution, and certain parameters are suitable for all tasks. Hadoop requires more system tuning as well as well as individual task tuning to work well with the system.
- The parallel DBMS’s aid the tuning process, but Hadoop requires trial and error tuning.
- Hadoop falls short in the categories regarding execution speed, loading times, and general performance in comparison to database management systems, but does succeed in ease of setup and use.

Analysis of ideas and implementations

- Hadoop may fall short in performance speed and in some functionality issues regarding tuning, but has picked up in popularity because of its upfront cost advantage from being an open source product along with its ease of use.
- Database systems are tolerant to a wide variety of software failures, but map-reduce does a better job in minimizing the amount of work that is lost in the event of a hardware failure.
- The ability to minimize hardware failure that Hadoop possesses is at the cost of performance, which is due to the cost of materializing the intermediate files between the map and reduce phases.
- The architectural differences between map-reduce and database systems regarding parsing cause parsing to occur at run time, while database systems are able to perform parsing at load time. This makes compression less valuable in map-reduce and causes a portion of the performance difference between the two classes of systems.
- Map-reduce systems include higher level interface systems such as Hive and Pig to improve upon the parts of the foundation that are lacking in comparison to parallel database systems.
- Parallel databases are looking to improve upon ease of use, where map-redue is looking to improve upon performance and functionality.
- Both classes of systems lack in certain areas and are making strides to improve, moving both towards eachother as they progress.

Comparison of both papers

- Hive contributes to the map-reduce database system and improves upon some of its functions.
- Map-reduce already succeeded in ease of use and retention of work during hardware failure, but lacked in performance and execution speed. Hive contributes to improving performance for larger scale data in order to make analytical queries easier to perform, while still using similar query language to SQL to ensure the programmer can effectively jump right in to writing queries in HiveQL.
- Parallel database systems implementation is more difficult at startup, and with large scale data performs as well as map-reduce without the use of Hive, shown by the tables in the comparison paper. While hive may not necessarily improve the speed of map-reduce query execution, it does make it easier to use with big data, which improves upon the strengths of map-reduce.
- The implementation of Hive into Hadoop is also relatively easy, similar to the implementation of Hadoop itself, making the map-reduce database system type more attractive for companies to use.
- The implementation of parallel database systems may be more difficult at startup, but can offer better performance speeds if that is what the user prioritizes in their database. The speed of performance is a result of the development of parallel databases systems over the course of many years, ability to operate directly on compressed data, and sophisticated parallel algorithms for querying large amounts of relational data.
- Hive and Hadoop together create a very efficient database system that can deal with big data and perform many analytical queries for the respective data at an efficient speed, without requiring a completely different query language to do so.

Analysis of Stonebraker talk

- The main idea of the talk is that one size fits none, different types of database systems are needed for different purposes.
- Many systems are becoming obsolete such as oracle and SQL server.
- All major vendors are going to be column stored rather than row stored, because column stores are much faster than row stores.
- Business analysts are predicted to be replaced by data scientists over time, while SQL analytics while phase out and analytics will be defined on arrays rather than tables.
- Business intelligence products are predicted to phase out SQL analytics to define data on arrays rather than tables. SQL is proving to be very slow with arrays, which will lead to its phase out.
- OLTP engines have a greater market share of streaming because of the ease of adding streaming to an OLTP engine.
- Moving to new technology causes trouble for users of older technology to not lose market share
- DBMS research was stagnant for many years, where the one size fits all mentality existed, but now there are many new ideas coming in to database implementations

Hive comparison to paper and talk

- Hive’s main purpose is to improve upon the already existing Hadoop map-reduce database system.
- Advantages to Hive include easy handling of data analytics using SQL based programming code.
- The disadvantages of Hive in regards to the Stonebraker talk is the fact that it is continueing with the same technology and database ideas that he considers obsolete, and will be phased out in the next decade.
- In regards to the comparison paper, Hive is an improvement upon what is an already viable and prefered method of data warehousing for use with big data and big data analytics.
- If what is suggested in the stonebreaker talk is true, Facebook and other big companies that use Hive and Hadoop to manage their data will look to switch to array defined data rather than the table defined data that Hive and Hadoop currently operate on.
- Array database systems can be run on SQL, but will run very slow, which may lead to users of Hadoop map-reduce or parallel database systems to transition from their current systems.