Big Data Paper Summary

HIVE – A Petabyte Scale Data Warehouse Using Hadoop

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

A Comparison of Approaches to Large-Scale Data Analysis

By: Andrew Palvo, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden, Michael Stonebrake

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

By: Zachary Tsouprakos

3/28/2016

HIVE Stacked On Hadoop

- Data that is being collected and analyzed by large companies is growing exponentially, thus resulting in traditional data warehouse solutions becoming very expensive
- Hadoop, a relatively new open-source map-reduce implementation is being used by major companies such as Yahoo, Facebook, and others.
  - Hadoop allows large sets of data to be processed, and is a solution to the problem addressed above
  - Hadoop allows data to be stored at a petabyte scale, all on commodity hardware
- The map-reduce programming model is very difficult to maintain because custom code is needed to be developed for specific jobs
  - This is not easy for end-users
  - This lacks the expressiveness of SQL
- Hive is an open-source data warehouse solution that is stacked on top of Hadoop, used to manipulate all of the data stored using Hadoop

HIVE, How Is It Being Used

- As previously stated, Hive is an open-source data warehouse solution that is stacked on top of Hadoop
  - HiveQL is the query language used to manipulate the data being stored.
  - HiveQL supports queries similar to SQL, which are compiled into map-reduced jobs, executed using Hadoop
  - HiveQL supports tables with primitive types, collections of arrays and maps, and more
  - Similar to SQL, there is a system catalog that contains schemas and statistics (Metastore)
- HIVE is used to avoid the map-reduce language for jobs being executed in Hadoop
- HIVE allows users to use the expressiveness of SQL on the data stored within Hadoop

Analysis of HIVE

- HIVE allows a solution to the tedious and difficult map-reduce programming model
- This software allows end users to take data manipulation, storage, and analysis into their own hands
- Whether it is partitioning existing data or creating new tables, HiveQL gives the end-user control
- HiveQL is also very powerful because it allows analysis expressed as map-reduce programs in the programming language of the end-users choice
  - This allows complex logic expressed in map-reduce programs to be executed into HiveQL queries effortlessly
- Hive is great solution to managing the data stored in Hadoop, and even though it is still in the works, it has proven its usefulness and power

A Comparison of Approaches to Large-Scale Data Analysis

- MapReduce (MR) is a large-scale data analysis computing model, which is the new "hype" in the big data world, which is part of Hadoop
- Parallel Database Management Systems
  - These are the traditional databases that have been around for 20+ years
  - These databases work along side one-another, unlike Hadoop/ MR(one system)
- This paper compares MapReduce and parallel database systems
  - Both models are evaluated based on development complexity and performance
  - Multiple tasks are ran on both models, and the performance of the two models are compared

MR vs Parallel DBMSs

- 5 Tasks were executed on Hadoop MR, Vertica, and DBMS-X (Vertica and DBMS-X being the parallel DBMSs)
  - The tasks that were conducted showed several different statistics
  - Each task was executed in three different scenarios
    - Some scenarios used as little as 1 node, where the biggest cluster was up to 100 nodes
- From software configuration, and storage testing, to analytical tasks, the power of these tools were tested and the results were quite shocking
- EX: One test was was taken directly from the original MapReduce paper
  - This task specifically tests the scanning abilities for the programs
  - Each program needed to scan through a 100-byte data set, searching for a specific pattern

Analysis of MR vs. Parallel DBMSs

- It was seen that the parallel DBMSs outperform Hadoop MR on all levels of cluster scaling (up too 100 nodes)
- Hadoop MR showed to be the slowest, compared to Vertica being the fasted on all cluster scaling and DBMS-X coming in second after Vertica
  - An advantage of MR that was seen was uploading data to be stored did take significantly quicker in comparison to the parallel DBMSs
- Another factor that was seen throughout this paper was the fact that MR executes a complete table scan, where the parallel DBMSs can take advantage of the clustered indexes, reducing time
- As more data and nodes were added to the trials, Hadoop MR showed to be affected the most
  - This is due to the start-up costs as more nodes are added to the cluster, which proportionately results in larger fraction of query time compared to the other two models

Comparison of ideas and Implementation

- The HIVE paper took a different approach of implementation
  - This was more of a using and testing approach to better Hadoop
  - Hive was stacked on Hadoop and used to manipulate data
  - The idea was to have a query language similar to SQL for users to have an easy transition to HiveQL
  - HiveQL was the solution for avoiding the difficult MapReduce code that was hard to maintain and reuse
- Comparing MapReduce to two parallel DBMSs resulted in shocking results
  - It was seen that the two parallel DBMSs out-performed MR
  - The parallel DBMSs and MR were given 5 tasks to perform
    - The parallel DBMSs out-performed MR in all tasks including analytical tests
  - Some of these tasks were jobs such as data storage, data scanning, and aggregating values

Stonebreaker Talk: Main Ideas

- One size fits none!
  - Traditional row stores do not do the job anymore and are obsolete!
- Changing storage to main memory
  - Placing all data in main memory is more effective and cheap
  - This results in light transactions, along with different SQL techniques compared to row storage
- All major vendors have now or will have column stores
  - These column stores are"2 orders of magnitude" faster than row stores
- Complex Analytics
  - Column stores, array stores, or another potential statistical package will take this market (as well as the graph analytics market)
  - Row stores (SQL) perform these analytic algorithms too slowly for today's expectations
- New ideas
  - The "bottleneck" seems to be at networking, higher speed networks will potentially fix this problem
  - Non-volatile RAM will get rid of flash-memory
    - Processor diversity will increase due to main memory databases growing exponentially
- What about the "elephants"
  - The traditional developers or innovators will need to transition while trying to not lose market share
  - Stonebreaker believes that SQLServer and Hekaton will be the legacy vendors that lead this transition

Advantages & Disadvantages of HIVE

Being Compared to The Comparison of Large Scale Data Analysis and Stonebreaker's Talk

- Advantages:
  - Hadoop is a column storing database which results in faster execution than a row store database
  - HiveQL alleviates the struggle of learning MR, and provides the users with a language that has the expressiveness of SQL
  - HiveQL allows SQL users to adjust to this query language quickly, resulting in Legacy Vendors having users adapt to this relatively new engine, potentially resulting in lower loss of market share
  - Using Hive, Hadoop has the potential to take over markets such as graph analysis and complex analytics
- Disadvantages
  - MapReduce is very inefficient in comparison to parallel DBMSs
    - Statistics show that parallel DBMSs produce faster analytical tasks compared to MR
    - There is also a very steep learning curve when dealing with MR, which results in wasted time from users learning MR
  - Hadoop and Hive is also inefficient because it does not split its tasks amongst databases (it does use multiple nodes but is only one "system"), resulting in tasks being completed at a slower rate
    - Having an updated version would help this problem but make it difficult for the original version to be converted to this new system
    - Stonebreaker speaks about how you can have specific databases for certain tasks