Megan O'Halloran
March 3, 2016
CMPT308 – Big Data Summary

Hive – A Petabyte Scale Data Warehouse Using Hadoop
A Comparison of Approaches to Large-Scale Data Analysis
Michael Stonebreaker on his 10-Year Most influential Paper Award

BIBLIOGRAPHY:

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 IEEE 26TH INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE 2010)* (2010): N. PAG. WEB.

PAVLO, ANDREW, ERIK PAULSON, ALEXANDER RASIN, DANIEL J. ABADI, DAVID J. DEWITT, SAMUEL MADDEN, AND MICHAEL STONEBRAKER. "A COMPARISON OF APPROACHES TO LARGE-SCALE DATA ANALYSIS." *PROCEEDINGS OF THE 35TH SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA - SIGMOD '09* (2009): N. PAG. WEB.

MICHAEL STONEBRAKER ON HIS 10-YEAR MOST INFLUENTIAL PAPER AWARD AT ICDE 2015. PERF. MICHAEL STONEBRAKER. *MICHAEL STONEBRAKER ON HIS 10-YEAR MOST INFLUENTIAL PAPER AWARD AT ICDE 2015*. N.P., N.D. WEB. <HTTP://KDB.SNU.AC.KR/DATA/STONEBRAKER_TALK.MP4>.

Hive – A Petabyte scale Data Warehouse Using Hadoop

Main Idea

- Hive was created to provide a simpler query capability while utilizing Hadoop's ability to handle petabyte scale data warehousing
- The idea of Hive is to use familiar concepts of tables, columns, partitions, combining ideas from SQL and Hadoop
  - Uses similar language from SQL while "maintaining the extensibility and flexibility that Hadoop enjoyed."
- Hive is used by a wide variety of applications
  - Facebook

The Implementation of Hive

- Hive supports the major data types and more complex types:
  - integers, floats, doubles, and strings
  - Maps: similar to associative arrays, Lists: lists columns of tables that use SerDe, Structs: physically groups a list of variables with a name under a block of memory
- Hive query language – HiveQL
  - Some parts SQL and some extensions the developers found useful
- Hive components:
  - Metastore: stores the system catalog and metadata about tables
  - Driver: manages the life-cycle of a HiveQL statement, the sessions activity and it's statistics
  - Query Compiler: compiles the HiveQL statements into a graph of map/reduce tasks
  - Execution Engine: executes the tasks produced by the compiler in order of dependency
  - HiveServer: provides a fast interface, a JDBC/ODBC server and a way to incorporate Hive into other applications
  - Client Components: such as the CLI, the web UI, and the JDBC/ODBC directory
  - Entensibility Interfaces: includes SerDe, ObjectInspector interfaces and UDAF interfaces that allow users to create/declare functions

My Analysis of Hive's ideas and Implementation

- HiveQL is very easy to understand:
  - As a beginner in SQL I understood all the examples presented in the article
  - Uses commands exactly like, and very similar to, SQL
- Hive and Hadoop are primarily used in Facebook
  - It can take in large amount of data and store it
  - Facebook takes in large amounts of data an hour, whether it's for advertisers or Facebook's Lexicon
- Features in Hive are still being enhanced such as: the JDBC/ODBC drivers and the HiveQL syntax
- Implementation was a little difficult to understand
  - Hive is very reliable
  - The main building blocks of Hive are thorough

A Comparison of Approaches to Large-Scale Data Analysis

Main Idea

- The description and comparison between Map-Reduce (MR) and Database Management System (DBMS)
  - Hadoop representing Map-Reduce, which was used to create Hive
  - DBMS-X which represents a parallel SQL DBMS
  - Vertica to represent DBMS designed for large data warehouses
- Vertica ran best on all of the tests, where Hadoop had the most problems
  - The main issue with Hadoop was that it's loading speed was significantly slower
  - For some tests, queries needed to be modified to be run on Hadoop, where Vertica was able to accept all queries fast and easily

The Implementation of the Comparison Between Hadoop, DBMS-X, and Vertica

- Seven tests were used on Hadoop, Vertica and DBMS-X
  - A description of how these tests were run was presented, as well as their performance and an explanation as to why they may had done well or failed
  - These tasks were given to test:
    - Task Start-up
    - Compression
    - Loading and Data Layout
  - Each time a new test was ran, a little bit more of information was added to see the effects on the frameworks
- DBMS did run better overall

My Analysis of the Comparison Paper

- Both Database Systems appeared to have had a better performance then Hadoop, which was used to represent MR
  - DBMS-X ran 3.2 times faster than Hadoop
  - Vertica ran 2.3 times faster than DBMS-X
- MapReduce is capable of accepting large amounts of data, but I was surprised to learn that the processing time was very slow in comparison to the Database Management Systems that were tested in this paper
- After analyzing the paper, I found both DBMS may possibly have the following advantages:
  - B-tree indices: increase the speed of execution time for select statements
  - Novel storage mechanisms: allow command and control of memory
  - Aggressive compression techniques: ability to operate and compress commands
  - Sophisticated algorithm techniques: capable of reading through large amount of data quickly, and efficiently

Comparing the Two Papers

- Hive is a framework that uses Hadoop as well as SQL
  - The syntax was very easy to understand
  - Easy set-up
- The runtime is an issue that would need to be taken care of on the side of Hadoop
- DBMS was very fast and can store large amounts of data
  - The syntax of DBMS was also very easy to understand because they were based on SQL syntax as well
- Overall, DBMS is the better way to go
  - The syntax is simple
  - Has a fast execution time
  - Can read data and store it at the same time
- The only disadvantage I can see with DBMS is the set-up time

Michael Stonebraker on His 10-Year Most Influential Paper Award

Main Idea

- In his paper, he talked about how RDBMS was supposed to be the answer to everything
  - Complex data types, Referential Integrity, Triggers
  - RDBMS was going to be universal: "one size fits all"
- RDBMS is not the answer
  - Used C-store; looked nothing like rows
  - SQL servers were very slow
  - Used arrays not tables; very inconvenient
  - Had no data management
- DBMS is the way to go
  - Capable of doing more in the future than it does now because it is constantly being worked on
    - Information can be stored on NVRAM
    - Work on high speed networks
    - New implementations are constantly being added

Advantages and Disadvantages of Hive in Context

- Advantages based on both papers and the talk:
  - Very easy to understand and code
  - Very easy to set-up and implement into a system
  - Provide support to debug systems
  - Can check user code and can determine whether queries are evaluated correctly
- Disadvantages based on comparison paper and talk:
  - Very slow execution time
  - Contains complex analytics
  - High maintenance
  - Not universal, does not work on all applications
  - Does not hold as much data as DBMS
  - The difficulty and annoyance of having to reuse code on different data sets
  - "No explicit representation of the schema of data used"