A Comparison of Database Systems

- *Hive – A Petabyte Scale Data Warehouse Using Hadoop* (Thusoo, Sen Sarma, Jain, Shao, Chakka, Zhang, Antony, Murthy, Murthy) 2010
- *One Size Fits All- An Idea Whose Time Has Come and Gone* (Stonebraker, Cetintemel) 2005
- *A Comparison of Approaches to Large-Scale Data Analysis* (Pavlo, Paulson, Rasin, Abadi, DeWitt, Madden, Stonebraker) 2009

Iglika Hadjiyska, March 15, 2016

Hive – A Petabyte Scale Data Warehouse Using Hadoop

Main idea:

- Previously, RDBMS used too much data and took too much time
- Facebook switched to Hadoop
  - it was open source, offered scalability, executed tasks faster
  - confusing and required special programs
- Hive was created
  - a data warehouse built on top of Hadoop
  - runs on MapReduce
  - uses Hadoop File System as storage

Hive – A Petabyte Scale Data Warehouse Using Hadoop

Implementation:

- Traditional database model: stores data in tables, rows and columns, and supports primitive and complex data types
- Hive query language: format follows SQL closely with certain limitations
  - format on joins is slightly different and streamlined
  - Inserts are not possible, will overwrite data instead
- Data storage in HDFS:
  - Table: stored in a directory in HDFS
  - Partition: part of a table stored in the subdirectory of a table's directory
  - Bucket: stored in a file on a table's directory (or partition's directory if table is partitioned)

Hive – A Petabyte Scale Data Warehouse Using Hadoop

Implementation (continued):

- File formats: specifies how the records will be stored in a file (text, binary, files stored as columns, however the user likes)
- Building Blocks:
  - Metastore: systems catalog of Hive
  - Driver: manages lifecycle of HiveQL statements, maintains session handle and statistics
  - Query Compiler: compiles data into directed graph of tasks
  - Execution Engine: executes tasks from compiler
  - Hive Server: provides thrift interface, integration of Hive with other applications
  - CLI and other various interfaces

Hive – A Petabyte Scale Data Warehouse Using Hadoop

Analysis:

- Hive works well for Facebook's needs:
  - Huge data loads must be organized in some way and without wasting too much time or space
  - It optimizes the tasks needed to be run, however minute, so that they are done faster
- It also still utilizes parts of RDBMS, which helps familiarity:
  - HiveQL mirrors SQL closely and its differences barely affect any data inputted
  - Metastore allows for storage of metadata which is essential for the query compiler and execution engine
- This would not be ideal for systems that wouldn't use RDBMS

A Comparison of Approaches to Large-Scale Data Analysis

Main Idea:

- MapReduce vs. Parallel DBMS: is there a clear function for MapReduce that cannot be achieved by parallel DBMS?
- MapReduce
  - Easier to load, start up
  - Has better fault tolerance
  - “Schema Later” paradigm
  - Less confusing than using SQL
- Parallel DBMS
  - Close to 2 times faster
  - Less energy needed

A Comparison of Approaches to Large-Scale Data Analysis

Implementation:

- Both systems were tested on several tasks:
- Grep: finding a three letter pattern in set of 100 byte records
  - Loading and Execution time
  - DBMS performed better and faster
- Analytical: HTML documents processing
  - Loading, Selection, Join
  - Aggregation, UDF Aggregation
  - DBMS did better yet again

A Comparison of Approaches to Large-Scale Data Analysis

Analysis:

- Interesting study still proves that RDBMS have a place
  - While MapReduce might be open source and easier to understand and load, it still didn't execute most tasks any better than parallel systems would have
  - However, Hadoop would have its place i systems that need fast load times and simple processing, rather than the repeat access given with RDBMS

Hive vs. Approaches to Large-Scale Data

Ideas:

- While RDBMS took too much data, some of its features were still implemented into the hybrid Hive model
- Hive is very much an amalgamation of MapReduce and DBMS

Implementation:

- While *Approaches* actually tests the task execution of both systems, *Hive* just described the functions which mirrored RDBMS closely
- Hive seems to solve many of the problems that MapReduce had

One Size Fits All- An Idea Whose Time Has Come and Gone

Ideas:

- One size fits none: RDBMS have become obsolete in numerous markets
- Most markets (data warehouse, complex analytics and graph analytics) use column stores
- Transaction processing (OLTP) needs little memory, doesn't need heavyweight row stores
- There are many other options (JSON stores, Big Table stores, etc), excel with features that row stores cannot help with (simulating graphs, data management, streaming engines, etc)

Hive comparison

Advantages:

- Hybrid of both DBMS and MapReduce: manages to mitigate many of the problems found in both
- Perfect for Facebook's data analytics needs

Disadvantages:

- Despite its hybrid model, it is still only good for certain tasks
- As shown through Stonebraker's talk, Hive would not be helpful for markets that would require column store or certain kinds of graphing