


# Big Data Paper Summary

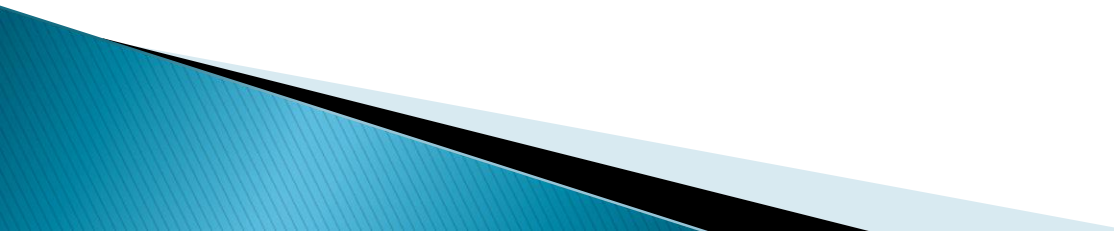
Brian Dones  
Database Management  
Professor Labouseur

May 8, 2015

# Main Idea: HIVE

- ▶ The vision of HIVE was to bring the familiar concepts to Hadoop while still maintaining Hadoop's extensibility and flexibility
  - ▶ Designed to be able to process extremely large data efficiently
  - ▶ Hadoop was difficult for end users because end users would have to write map-reduced programs for simple tasks.
  - ▶ Hadoop lacked expressiveness of common popular query languages resulting in excessive work for end users for simple analysis.
- 

# Implementation of HIVE

- ▶ HiveQL runs onto of Hadoop
  - ▶ Hive supports queries expressed in a HiveQL, which are compiled into map-reduce jobs that are executed using Hadoop
  - ▶ Hive utilizes the concepts of tables, columns, partitions and a subset of SQL to the unstructured world of Hadoop
  - ▶ Hive includes a system catalog, Metastore, that contains schemas and statistics, which are useful in data exploration, query optimization and query compilation
- 

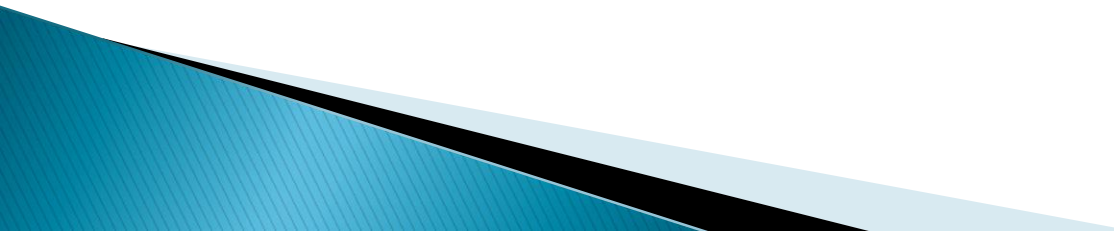
# Continued

- ▶ Hive supports all major primitive types such as integers, floats, doubles and string as well as more complex types such as maps, lists and structs
- ▶ Hive allows users to extend the system with uniquely-defined types and functions
- ▶ The logical data units in Hive are tables
- ▶ The primary data units and their mappings in the Hadoop File System are:
  - Tables – A table is stored in a directory in HDFS


# Continued... yet again

- Partitions – A partition of the table is stored in a sub-directory within a table's directory
- Buckets – stored in a file within the partition's or table's directory depending on whether the table is a partitioned table or not
- ▶ Hive has the following components that are the main building blocks:
  - Metastore
  - Driver
  - HiveQL
  - Query Compiler
  - Execution Engine
  - HiveServer
  - Clients
  - Extensibility Interfaces


# Analysis

- ▶ Utilizing SQL-like language structure and organizing data into tables, column, and partitions is more efficient than writing unique map-reduce code
  - ▶ Hive can support complex types and allows for nested complex types so experienced programmers may have more tools at their disposal for database effectiveness
  - ▶ Inserts in Hive will overwrite the data in other tables that exist which is great for data integrity
  - ▶ Hive's structured query language is similar to the universal standard features of structured query languages with some unique features of its own
- 

# Comparison

- ▶ Since we run Hive on top of Hadoop, we can utilize the nice uniformity and organizational qualities traditional SQL languages provide
  - ▶ Hadoop is very effective and efficient in data loading speeds – this is especially important when dealing with exponentially growing sizes of data within database nowadays
  - ▶ Hadoop lacks performance and reliability in querying, join tasks and aggregation tasks. This is a potentially huge issue because while we may be able to work with large scale data quickly, this leaves potential for incorrect and inconsistent data. Its great that Hadoop is fast, but it is better to get the right answer in a slower but feasible amount of time as opposed to the wrong answer very quickly.
- 

# Main Ideas: Stonebraker talk

- ▶ The Relational Database model was the one solution to everything related to storing data but in recent years we see that the relational Database model is not the one answer to everything out there
  - ▶ A faster solution than rows stores in data warehouses are column stores
  - ▶ Complex Analytics, Streaming, and Graph Analytic Markets have been shown to be examples of traditional database models not working for today's needs
  - ▶ The idea of “one size fits all” was completely off and, overall, the database model was far from what we wanted as an answer
- 



# Advantages and Disadvantages

## Advantages:

- Hive is faster at loading data which is very beneficial in larger quantities of data
- Hive is more reliable
- SQL-like structure that supports relational data models using tables and columns

## Disadvantages:

- Slower in pattern-matching, aggregation, and selection tasks
  - Does not support Insert into, Update and Delete
- 