Big Data Paper Summary

Hive - A Petabyte Scale Data Warehouse Using Hadoop

A Comparison of Approaches to Large-Scale Data Analysis

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

BRANDON TRADITI

DATABASE MANAGEMENT

10/30/17

Main Ideas of Hive

- "Our vision was to bring the familiar concepts of tables, columns, partitions and a subset of SQL to the unstructured world of Hadoop, while still maintaining the extensibility and flexibility that Hadoop enjoyed."
- An SQL based open-source data warehousing solution built on top of Hadoop built by Facebook Data Infrastructure Team.
- Use Hadoop/Hive cluster to run multiple jobs varying in applications from business intelligence to machine learning.

Implementation of Hive

- Data Model- Structures Data into commonly used database concepts such as tables, columns, rows and partitions. Supporting primitive types like integers, floats, doubles, strings, maps, lists and structs.
- Query language is similar to SQL, comprises of a subset of SQL along with extentions.
- Use in Facebook- warehouse consists of 700TB of data.
- "In general, the system had enabled us to provide data processing services to engineers and analysts at a fraction of the cost"

Analysis and Implementation

- Hive applies another layer to Hadoop that allows users to efficiently fulfill jobs in different applications.
- Eliminates end users having to take days to write programs in Hadoop.
- Similar structure to SQL, allowing users with SQL experience to easily understand and use Hive.
- Keeping traditional database concepts like tables, columns, rows and partitions allows familiarity for database users.

Main Ideas of Large Scale Analysis

- Evaluate and compare both MapReduce and SQL database management systems in terms of performance and development complexity.
- Understand the differences between MapReduce and Parallel database in the approach to preform large-scale data analysis.
- Present the results of running a benchmark on a 100-node cluster.

Implementation

- MapReduce-Simplistic, Consisting of only two functions: Map and Reduce.
- Map function reads a set of “records” in the input file, filters or transforms these inputs, and outputs the records as new key pairs.
- Reduce function processes or combines the records assigned to it and then writes an output.
- Parallel DBMS-most tables are partitioned over the nodes in a cluster and the system uses an optimizer that translates SQL commands into a query plan.

Analysis and Implementation

- Comparing the new and old concepts of this system within the same benchmark shows a great advantage/disadvantage scale between the two.
- Final comparisons between Hadoop, DBMS-X, and Vertica.
- Grep task-scan through a data set of 100-byte records looking for a 3 character pattern.
- Multiple Tasks ran on each system including data loading, selection, aggregation and join.
- In conclusion, parallel database structures outperformed MapReduce.
- Vertica was 2.3 times faster than DBMS-X while DBMS-X was 3.2 times faster than Hadoop.

Comparisons of Both Papers

- Both of papers touch on the topic of a MapReduce system.
- Hive- builds on top of Hadoop to attempt to make it much more efficient and user friendly .
- Tests ran on Parallel systems vs MapReduce show that MapReduce is far less efficient.
- Both papers touch on the system architecture or design of MapReduce systems while running queried commands on them.

Main points of Stonebraker

- History of RDBMS and its push for universality
- One Size fits none
- Data Warehouses
- Transaction Processing
- NoSQL
- Data scientists
- Analytics: complex and graphical

Advantages and Disadvantages

- Advantages of Hive:
- Shows the process of how to improve an already mainstream system.
- Dives deep into the system architecture and diversification of the system.
- Disadvantages of Hive:
- Doesn’t show comparisons of the MapReduce system vs its competitors on the market.
- Doesn’t show much of the big picture or the industry, only shows a small subsection.