Hive – A Petabyte Scale Data Warehouse using Hadoop

A Comparison of Approaches to Large-Scale Data Analysis

Stonebraker Talk

By Pearl Amin

October 27th, 2017

Thusoo, Ashish, et al. "Hive-a petabyte scale data warehouse using hadoop." Data Engineering (ICDE), 2010 IEEE 26th International Conference on. IEEE, 2010.

Pavlo, Andrew, et al. "A comparison of approaches to large-scale data analysis." *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. ACM, 2009.

Hive – A Petabyte Scale Data Warehouse using Hadoop

Hive is an open-source data warehousing solution built on top of Hadoop

Hive supports queries expressed in an SQL-like declarative language (HiveQL) which are complied into map reduce jobs that are executed using Hadoop

It is a work in process and is being actively worked on by Facebook and other contributors.

How the Idea is Implemented

Hive structures data into tables, columns, rows and partitions

It supports all the major primitive types: integers, strings as well as more complex types like maps and lists

Apart from some restrictions, HiveQL has extensions to support analysis expressed as map-reduce programs in their choice of language

The works: HiveQL is complied into mapreduce jobs that are executed using Hadoop

Facebook is using Hive and Hadoop for its flexible infrastructure that scales up while being cost effective with the increasing amount of data being generated

My Analysis of the Idea and its Implementation

I believe that Hive is a great idea because it allows users, such as myself, who know basic SQL, to create queries in MapReduce more easily

It will help streamline the processes of using MapReduce which, now, is very slow and not as user-friendly as other systems

Its implementation is still a work in progress, only accepting a subset of SQL as valid queries but it is being worked on

The Main Ideas of the Comparison Paper

- This paper compares Hadoop, DBMS-X, and Vertica
- It concludes that Hadoop is easy to setup in comparison to DBMS-X and Vertica
- However, Hadoop proved to be very slow despite its potential ability
  - Example: If a MR system needs 1,000 nodes to match the performance of a 100 node parallel database system (Vertica/DBMS-x) –making it more likely that a node will fail

How the ideas are implemented

- The ideas are implemented by conducting different experiments to help compare these three choices and their tradeoffs
- The results of these experiments concluded:
  - Possibly, the systems would all have the same relative performance on 1,000 nodes
  - Hadoop – easy to set up and use but has large startup costs and can require more programmer work
  - Vertica – installing is straightforward but erratic with certain system parameters
  - DBMS-X – difficult to configure but required a lot of oversight by the vendor to perform well and was quick

Analysis of Ideas and Implementation

- These experiments helped with understanding the pros and cons of all of these systems in different situations
- Post-examination, we are able to determine what we can do to improve these systems for future use
- These systems are not perfect, Hadoop has the most potential for improvement
  - Improvements are actively in the works by using systems like Hive and Pig that run on top of MapReduce
- There is no one best solution, they all have their tradeoffs

Comparison of Hive and A Comparison of Approaches to Large-Scale Data Analysis

- Hive runs on top of MapReduce
  - Allows end users to use Hadoop more easily if they are not familiar with MapReduce
- Writing map-reduce programs for simple tasks was very difficult
  - Example: Getting the average
- Hive was inspired by popular query languages like SQL (expressive) that help users analyze data more productively
- In the comparison paper, it stated Hadoop had the potential for so much more than:
  - Minimize the amount of work lost during hardware failure
  - Being easy to get started with (*beware of long term maintenance over time)
- Both papers wanted increased productivity, efficiency and the use of an SQL-like language

The Main Ideas of the Stonebraker Talk

- DBMS Research dead in the 80s-90s
- Until the 2000s, the idea was that RDMS was the answer to all (one size fits all)
  - In 2015, they realized one size does not fit all
    - It fits none
- Columns are the future (they are faster)
  - Rows are becoming obsolete as markets expand
- Huge increasing diversity of engines
- SQL will lead the way (speed)
  - Make legacy vendors adapt or die
- SAP system called HANA runs on top of Oracle storage
- Future possibilities:
  - SAP will not support Oracle or will run faster on HANA
  - SAP will be primarily in the database business
    - Oracle and SAP will begin to compete
- Great time to be a DBMS Researcher!

Pros/Cons of Hive in the context of the comparison paper and Stonebraker talk

- Pro: Hive is a great innovative solution that allows more users to use systems like Hadoop for providing data summarization, query and analysis
- Con: Hive is not developed enough to support everything that SQL currently can
- Con: Hive is very slow, and Stonebraker stated that we need increased speed, which is very problematic for future growth
  - In the comparison paper, Hadoop was consistently very slow compared to the other systems
  - We are moving toward columns and away from rows