A Paper on Hive and its comparison with DBMS

-Vishnu Meduri,

5/7/2014.

Bibliography:

1. http://labouseur.com/courses/db/papers/thusoo.icde2010.hive.pdf
2. http://labouseur.com/courses/db/papers/pavlo.sigmod09.comparison.pdf

Main Idea:

The Hive is a project developed to design a data warehouse at Facebook to support large data storage and managing it.

Because of the increasing amounts of data to be stored a traditional RDBMS will not be able to handle the data in the required time.

And introducing a totally new system will make it difficult for the end users to adapt to it. So keeping the SQL quires like interface to interact with the system but changing the implementation is the idea of developing HiveQL.

The underlying implementation of HiveQL depends on Hadoop which is an open-source project. Hadoop is a framework that can be used to store large amounts of data.

Implementation of the HiveQL:

The HiveQL is designed to be easy to use for users familiar with SQL. So the language for querying the data warehouse is almost like SQL.

The SQL queries are then complied into map-reduce jobs which can be handled by Hadoop which is the underlying platform of Hive.

The file system of Hadoop is the Hadoop distributed File System.

Hadoop is designed and developed in Java.

So basically Hive supports the data types that are supported by SQL and also has some of its own native data types that it can use.

Hive can also store data in the form tables as seen traditionally in a SQL based platform.

My Analysis:

The idea of retaining the use of SQL syntax quires is a very good perspective instead of using a totally different language for the same purpose of querying a data warehouse.

And the implementation of compiling the quires in to a form that Hadoop can understand seems to be the easiest way to plugin the platform.

As more amounts of data is being generated it is inevitable to migrate to a more sophisticated system which can handle the data more efficiently thus a system like HiveQL is being developed.

Comparison:

When comparing the Architectural elements between Hive and a parallel DBMS. The Hive which is based on Hadoop is better at supporting the schema.

Indexing is also a very useful feature in Hadoop which is not quite useful when using DBMS which decides the index depending on the query.

The execution strategy in DBMS seems to be better than the one used in Hadoop.

There is less flexibility in only using the SQL for accessing and querying and that's why the use of user-defined querying and map-reduce jobs are more efficient at this.

The performance of analytical tasks is compared and in all the tasks mostly Hadoop has an upper hand as it depend on Map-Reduce jobs than the DBMS.

Though it takes some time for starting up the HDFS when compared to DBMS but it also reduces the space required by the large data sets and also better at recovering from faults.

Advantages:

Installing Hadoop is very simple.

HDFS can make accessing data from large data sets faster than traditional RDBMS.

Hadoop has a more fault tolerant system.

The flexibility of having user-defined functions and data types.

Hadoop comes with a web interface to browse the file system.

Disadvantages:

Loading of Data takes more time than in RDBMS.

It is difficult to design query into the Map-Reduce paradigm with aggregations that can be used more efficiently in SQL.

The Hadoop system is not as robust as RDBMS.

Indexing is not built-in for Hadoop and must be implemented by programmers and this is not easy to accomplish.