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

Kevin Clark
5/9/14

Main Ideas of the First Paper

- Data generated in companies like Facebook and Yahoo was growing big, very quickly so they needed a form of infrastructure that could scale with the data and perform daily processing
- Hive is an open source query language that is very similar to SQL except runs more efficient with big data
- In order to ensure that data is backed up, Hive replicates data and stores it temporarily when doing queries
- It stores the data into folders that the user can create based on the data
- Users can specify what kind of format they want their Hadoop files to be
- Users can also use regular expression to parse columns out from a row

How the mains idea are implemented

- Facebook Data Infrastructure Team created Hive in order to make it easier for users so they didn't have to spend hours to write simple queries
- Hive accepts a subset of SQL as valid queries
- Hive backs up data by copying the data before the query is ran and putting it on temporary storage
- Hive uses tables, partitions, and buckets to store data logically and users can create partitions and buckets by using certain commands. These commands then separate the data based on what the user puts in the command and creates separate files for the data
- Hive doesn't impose restrictions on file input formats. It has its own "Stored As" clause
- Regular expression is useful for custom serialization/deserialization

Analysis of main idea and implementation

- Hive met the needs of Facebook by combining the ease of use and familiarity of SQL with the unstructured and difficult to query Hadoop
- Hive architecture is very customizable which could be a bad thing as data grows more and more there isn’t as many standards that users need to follow
- There is no insert, update, or delete, only insert overwrite. This means users don’t have to worry about locking but the lack of insert could pose a problem
- The data storage is very logical and well made, as the data is just stored in different, specific directories created by the query that the user creates
- Regular expression can be a very powerful tool and it was a good idea for them to include that as an option

Comparison of Hive and Comparison Paper

- Hive is a language for Hadoop which is the open-source version of MapReduce that Facebook uses
- Codasyl was the low-level language that was used for MapReduce until Facebook Data Infrastructure team came up with Hive
- Hive and MR are much better at handling failures of execution than DBMS as the result files are streamed one by one as opposed to a transaction. So if a node fails then you don't have to restart the entire query
- File Input for Hadoop yields better performance than serialization data and DBMS
- The queries code for Hadoop used in the comparison paper were mostly written in Java as opposed to Hive (which is more user friendly for Hadoop)
- Hive takes away a big advantage of SQL engines because it is easier to code in and takes much less code to run tasks

Advantages and Disadvantages of Hive

- Compared to a DBMS, if sharing is needed between two programmers then it is harder because each programmer must know what indexes can be used, and how the data is structured. If it were one programmer than it would be better because it is so flexible
- You only need Hive when using a MapReduce system, which a lot of companies don’t need because it is processing extremely large amounts of data (Petabytes)
- The start-up costs of Hadoop makes running tasks much slower than if using DBMS
- Hadoop and Hive are easier to install but tuning the system and tuning each individual task to work with the system took longer
- Extra tools need to be developed in Hive for Hadoop as opposed to DBMS which has a lot of tools