Big Data Paper

By: Shannon Brady

PAPER ONE:

Tan, Junjay, et al. **“Choosing a Cloud DBMS: Architectures and Tradeoffs.”** *Proceedings of the VLDB Endowment*, vol. 12, no. 12, 2019, pp. 2170–2182., doi:10.14778/3352063.3352133.

PAPER TWO:

Thusoo, Ashish, et al. **“Hive - a Petabyte Scale Data Warehouse Using Hadoop.”** *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, 2010, doi:10.1109/icde.2010.5447738.

Paper One: **Main Ideas**

- Apache Hive is an open-source data warehouse system designed for querying large data systems in the Hadoop Distributed File System. A data warehouse provides a central store of information for the purposes of data management and analysis.
- Hive was developed to manage petabytes of data in a way that is both structurally flexible and cost effective. It was developed in 2010 and is used by large companies such as Facebook and Yahoo.
- The three main features of Hive are data summarization, data analysis, and data query.
- Hive converts SQL-like queries into MapReduce jobs for efficient execution and rapid processing of extremely large data sets.
- The query language utilized by of Hive is HiveQL
  - HiveQL: an SQL-like interface that is used to query data stored in databases. In addition to containing a subset of SQL, HiveQL maintains many traditional features of SQL, including all the simple primitive types (integers, floats, doubles, and strings) and well-understood relational database concepts, such as tables, columns, rows, and partitions.
- The way in which Hive stores and queries data closely resembles traditional relational databases; however, since Hive is built on top of Hadoop, it has to adhere to the rules set by the Hadoop framework.

Paper One: **Implementation**

- Hive and Hadoop were designed with the intent of creating an infrastructure that is both capable of scaling up in a cost effective manner and flexible in catering to the needs of a diverse set of applications and users.
- Hadoop, however, was largely inefficient alone due to the fact that it lacked the expressiveness of popular query languages such as SQL and was not easy for end users to understand, especially those unfamiliar with MapReduce. Hive was created as an extension to Hadoop in order to address these issues.
- Today, Hive is used extensively for different kinds of data processing, such as batch processing, at many large companies, including Facebook. Hive is also used for a wide variety of tasks, ranging from simple summarization jobs to complex machine learning algorithms. The system is also used by those with a wide variety of skills levels, with new users being able to use Hive with only a short amount of training.
- At Facebook, Hive provides data processing services to engineers and analysts at a fraction of the cost of more traditional warehousing infrastructure.
- Hive allows users to simultaneously access data while maintaining a much faster response time than most other types of queries. Hive is also highly adaptable in regards to adding more commodities easily in response to adding more clusters of data, all without any drop in performance.

Paper One: **Analysis**

- Due to its SQL-like features, Hive has a distinct advantage of deploying high-speed data interpretation while managing large datasets distributed across multiple locations.
- **Additional Advantages:**
    - Can quickly handle petabytes of data using batch processing.
    - Any problem that can be solved with MapReduce can be expressed in SQL.
    - Table structures are similar to tables in a relational database
    - Provides a familiar, SQL-like interface that is accessible to non-programmers and those unacquainted with MapReduce.
    - easy to distribute and scale based on company needs.
- **Disadvantages:**
    - Open-source, so can have degree of inconsistency with updates or support
    - Accepts only a subset of SQL
    - Not designed for Online Transaction Processing

Paper Two: **Main Ideas**

- When moving analytic (OLAP) applications to the cloud, there are a variety of DBMS setups to choose from. As more OLAP workloads begin to make this transition, DBMSs have changed from a shared-nothing design with locally attached storage to a hybrid design, ultimately leaving many advantages and trade-offs to consider.

- This study's main intent is to "provide users insights on how different DBMS and cloud configuration perform for a business analytics workload". The TPC-H benchmark is used to assess six popular OLAP DBMS applications, including Redshift, Redshift Spectrum, Athena, Presto, Hive, and Vertica, to find out which set up works best and the drawbacks involved.

- **Main findings:**
  - Cheap remote shared object storage (S3) provide greatest cost savings when compared to remote block stores (EBS), which are commonly used for DBMS storage in shared-nothing architecture
  - Physically attached local instance storage provide faster performance than EBS
  - Caching from remote object stores to node block storage is disadvantageous in cold start cases.
  - General use column store formats (ORC) provide flexibility for future system optimization when compared to formats used by shared-nothing DBMS. This provides more options to optimize performance if workload changes
  - Most systems gain from cluster horizontal scaling
  - Systems that use general data formats are most compatible with other systems

Paper Two: **Implementation**

- Design choices and trade-offs are grouped into three categories:
  - External storage, Query executors, and DBMS as a service offerings

- The tests were done using 1000 scale factor TPC-H data. Each of the systems tested begins from a cold start unless stated otherwise.

- The experimental results focus on six areas of comparison:
  - **Query Restrictions**
    - Spectrum and Athena could not run the full TPC-H query suite.
  - **System Initialisation Time**
    - Redshift had the highest initialization time at 12 minutes, with most systems having initialisation times in the range of 5-7 minutes. Athena does not require initialization before running a query, since it is an always-on service.
  - **Query Performance**
    - Measured from both warm and cold caches
    - Most systems have comparable performance with the exception of Redshift and Vertica on cold cache
    - Redshift offers the best performance for frequent querying
  - **Cost**
    - A combination of compute costs, storage costs, data scan costs, and software license costs
  - **Data Compatibility With Other Systems**
    - Systems that use general data formats (ORC on S3 and HDFS), such as Hive, Presto, and Vertica, are most compatible with other systems
  - **Scalability**

Paper Two: **Analysis**

- Choosing the optimal configuration and utilizing the full capabilities of DBMS-as-a-service can provide significant cost and performance benefits.
- Redshift appears to have the strongest performance if kept running and with a warm cache.
- Athena also a good option, given its strong query performance, cost-effectiveness, and the fact it is an always-on service
- **Overall advantageous design choices include:**
  - Prioritizing low-cost object stores, such as S3, for data storage
  - Using locally attached instance storage over EBS since they are faster and more cost effective
  - Using portable data formats that provide flexibility when switching to other systems for different workloads
  - Make features that benefit subsequent runs optional, such as query precompilation and caching data to faster storage

Paper One vs. Paper Two

- In regards to the various OLAP DBMSs and six areas of comparison discussed in *"Choosing a cloud DBMS: architectures and tradeoffs"*, Apache Hive does not appear to be among the top performers.
- Based on the data from the study, Redshift, for example, tends to be faster, cheaper, and easier to maintain.
- Apart from the graphical evidence provided, Hive was not often explicitly discussed in the relevant summaries. Rather, the focus of the paper often remained on Redshift, Athena, and Vertica. This may point to Hive's less than remarkable capability.
- Upon further research, it became clear that since 2015 Hadoop has lost significant relevance.
- According to Bobby Johnson, who helped run Facebook's Hadoop cluster, the fact that Hadoop is still around is a "historical glitch." He believes that Hadoop is "ill-suited for running interactive, user-facing applications." At the Hive layer, Johnson says that there remains a degree of usefulness, however he believes that there are far better alternatives (Woodie).
- Given this greater context, it makes sense why the optimistic discussion of Hive in *"Hive - a Petabyte Scale Data Warehouse Using Hadoop"* does not match the manner in which it was analyzed in the comparison paper, *"Choosing a cloud DBMS: architectures and tradeoffs"*.
- Hive may have been initially a hopeful solution to the issues posed by Hadoop, however it appears that over time its impacts have lost both relevance and significance.

Stonebraker Talk: **Main Ideas**

"One size fits none"

- Stonebraker uses this phrase to refer to the fact that there is not one database architecture that can fit the multiple distinct use cases relevant today.
- For decades the main database model was a relational database, specifically a row-store implementation. Regardless of the shape or function of the data in the application, the data was modeled as relational. Instead of the use case driving the requirements for the database, it was the other way around.
- Stonebraker points out that over the last 20 years, there has been a huge demand for alternative data management solutions. He claims that commercial world will continue to fracture into a collection of independent database engines.
- Stonebraker identifies the most efficient solution in most cases to be a column-store implementation, with row stores, he believes, in the process of becoming obsolete.
  - Column-stores are two orders of magnitude faster than row-stores.

Stonebraker Talk: **Main Ideas**

- Stonebraker often refers to "the legacy vendors" as "elephants" whose RDBMS product offerings together take up the majority of the current market for data management software. Over time, he claims, they will inevitably lose market share as new ideas/solutions become more prevalent.
- He cites number of specialized database markets where traditional RDBMS products are considering modern solutions, such as transaction processing and streaming services.
- Stonebraker also predicts data science will supersede business intelligence in years to come.
  - Data scientists are trained to work with arrays rather than tables and are capable of creating predictive models.
- Finishes the talk acknowledging that time will tell how accurate these predictions; no one knows for sure what the data management market will look like in the following decades. Since future is unknown, new spaces must be explored to see what is most successful.

Advantages and Disadvantages

- **Main reasons for choosing Hive:**
  - Provides an SQL interface operating on the Hadoop Distributed File System
  - Reduces the complexity of MapReduce
  - A horizontally scalable database
  - Supports processing and analysis of structured and semistructured data
  - HiveQL makes it easier for developers who have RDBMS backgrounds to build and develop faster performing data warehousing framework.

- However, the future probably is likely bleak for Hive as part of Hadoop.
  - When considering Hive in the overall context of the Hadoop framework, there does not appear to be much indication of lasting relevance.
  - MapReduce is considered largely outdated, with better alternatives in use today, such as Apache Spark.

- As stated by Stonebraker himself at the 2019 MIT Citi Conference
  - "Essentially, all of you drank the Hadoop Kool-Aid. And I heard this all the time, starting around 2011, eight years ago. Google says Hadoop is the answer. Hadoop, by the way, means MapReduce, the open source version of it written by Yahoo-- MapReduce. And MapReduce is not good for anything. I can state that categorically," ("Michael Stonebraker").

Additional Resources

Colyer, Adrian. “Choosing a Cloud DBMS: Architectures and Tradeoffs.” *The Morning Paper*, 25 May 2020, blog.acolyer.org/2019/08/30/choosing-a-cloud-dbms/.

“Michael Stonebraker - 2019 Citi-NY.” *ILP*, 6 Nov. 2019, ilp.mit.edu/watch/michael-stonebraker-2019-citi-ny.

Pei, Liquan. *Apache Software Foundation*, Confluence, 6 Sept. 2020, cwiki.apache.org/confluence/display/Hive/Home.

Woodie, Alex. “Hadoop Has Failed Us, Tech Experts Say.” *Datanami*, 13 Mar. 2017, www.datanami.com/2017/03/13/hadoop-failed-us-tech-experts-say/.