A REVIEW ON PERFORMANCE EVALUATION OF SQL-ON-HADOOP SYSTEMS

Mansi Bhadoria[1], Dr.Sachin Goyal[2], Dr.Ratish Agarwal[3]

[1]DDIPG Scholar, [2,3]Assistant Professor
Department of Information Technology
UNIVERSITY INSTITUTE OF TECHNOLOGY,
RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

Abstract: **The quantity of data which is generated by various industries for business intelligence is emerging speedily, and to analyze these enormous quantity of complex facts is an expensive task by traditional analytical techniques.apache Hadoop is an open source solution for storage along with processing these large amount of complicated facts, Hadoop is being used in many companies like Facebook, Yahoo, etc. Apache hadoop employs HDFS to store up the data and mapreduce framework for processing facts on commodity hardware. In spite of this, the mapreduce framework supports java programming model which is very minimal level and developers needs to write complex and custom programs to deal with the records and these programs is troublesome to write, preserve and reprocess. In the view of this paper, we proposed apache Hive which is an open source data warehouse analytical solution integrated upon the hadoop. Hive supports SQL identical to declarative language entitled as HiveQL whose queries are compiled by hive compiler and then converted into mapreduce jobs which are achieved by means of hadoop. Within this paper we can also discuss the significance of SQL-on-hadoop system.**

Keywords: **Hadoop, Hive, SQL, Mapreduce, SQL-on-Hadoop**

I. INTRODUCTION

Hadoop could be a widespread open supply computer code framework that enables the distributed process of massive scale information sets [1]. It employs the MapReduce paradigm to divide the computation tasks into elements that may be distributed to a goods cluster and so, provides horizontal scaling [2-9]. The MapReduce functions of Hadoop manipulates (key,value) pairs as format. The input is retrieved in chunks from Hadoop Distributed file system (HDFS) and appointed to 1 of the mappers which will method information in parallel and turn out the (k1,v1) pairs for the reduce step. Then, (k1,v1) try goes through shuffle section that assigns constant k1 pairs to constant reducer. The reducers associate the pairs with constant k1 values into teams also implements aggregation operations.

HDFS is that the underlying file storage system of Hadoop. As a result of its simplicity, reliable, fault-tolerance along with potency Hadoop has gained important support from each business and academia; but, there square measure some limitations in terms of its interfaces and performance [10]. Querying the information with Hadoop as during ancient RDBMS infrastructure is one in every of the foremost common issues that Hadoop user face. This affects a majority of users, an agency don't seem to be conversant in the inner details of MapReduce jobs to extract data from their information warehouses.

Hadoop Hive is associate degree open supply SQL-based distributed warehouse system that is planned to resolve the issues mentioned higher than of by providing associate degree SQL-like abstraction together with Hadoop framework for querying information hold on during a cluster [11].
When users demand to acquire acquaintance from each MapReduce and SQL, mapping SQL statements to MapReduce tasks will become awfully tough job. This job is done by Hive by decoding queries to MapReduce jobs; in this manner, make use of the scalability of Hadoop whereas presenting a well-known SQL abstraction. These attributes of Hive create it an acceptable tool for information warehouse applications wherever giant scale information is analyzed, quick response times don't appear to be needed, and there's no ought to update information oftentimes [4].

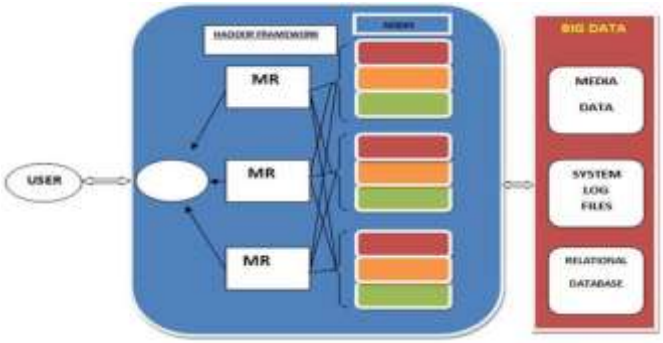

Fig. 1 Hadoop framework

Nearly all information warehouse applications are enforced maltreatment SQL-based RDBMSs, Hive take down the barrier to traversing these applications to Hadoop, thus, those already grasp SQL will simply employ Hive. Since Hive relies on a question-at-a-time model and processes every query severally, supplying multiple queries in shut measure decreases performance of Hive as an outcome of its implementation model. From this angle, it's vital to notice , no study, to date, that includes the Multiple-query improvement (MQO) technique for Hive to scale back the whole accomplishment phase of a batch of queries [1].

II. LITERATURE REVIEW

In [1] Hadoop is currently the factual customary for storage and treating huge knowledge, not just for unstructured knowledge however conjointly for a few structured knowledge. Consequently, providing SQL analysis practicality to the large knowledge resided in HDFS becomes countless and a lot of vital. Hive could be a foremost setup that assists SQL-like study to the facts in HDFS. In contrast, the efficiency of the primitive design of Hive isn't adequate. This results in the short occurrence of dozens of SQL-on-Hadoop systems that try and support cooperative SQL question process to the information hold on in HDFS. This paper first offers a concise technical view on modern struggle of SQL on-Hadoop systems. Then we incline to scrutinize and judge the functioning of 3 representative SQL-on-Hadoop systems, supported the TPCH benchmark. In keeping with the results, we tend to show that such systems will profit a lot of from applications of the many parallel query process techniques that are wide studied within the ancient massively data processing databases.

During this paper, they first review struggles of SQL on Hadoop systems in recent past. Then we incline to inspect 3 representative systems mistreatment the TPC-H benchmark. We discover that by applying progressive question process techniques (such as columnar storage, MPP design, be a part of optimization, and vectorized question execution) that are extensively studied by info community for several years; SQL-on-Hadoop systems will for the most part improve their performance. It's expected that with a lot of advanced parallel info techniques applied, the performance of SQL-on-Hadoop systems are often more improved. Providing high performance SQL analysis practicality to the information hold on in HDFS can attract a lot of and a lot of users to use SQL on Hadoop systems for interactive analysis as an alternate of proprietary DBMSs.

In [2], fast question within the huge knowledge is vital for mining the precious is data to boost the system performance. To attain this goal, analysis establishments and net firms develop three-type script question tools that are severally Hive supported MapReduce, Spark SQL supported RDD and Impala primarily based distributed question engine .In this paper, we tend to compare three-type question tools in many ways that. First we tend to analyze the impact of the file format for the question time, and that we conduct that compression will scale back the number of information, thus on improve the question time. It's the most effective option to take RCFile compressed by Snappy for Hive, and it's the most effective option to take Parquet for Impala.
Moreover, Impala holds the quickest question speed in comparison with Spark SQL and Hive. Second we tend to discuss that the file format impact on both memory in addition to the central processor unit. Impala taken Parquet prices the smallest amount resource of central processor and memory. Impala taken the file format of Parquet show sensible performance .So we tend to commit to appraise Impala and Parquet. Then we discover Parquet generated by completely different question tools show different performance. Finally, we discover the question velocity of Impala taken the file format of Parquet generated by Spark SQL is that the quickest. Consequently, appropriate to employ impala for fast question.

In [3], this paper set forth how to regulate the definite job performance development of Spark SQL and Hive, and make an analogy of them concurrently. First, with the help of ten SQL queries, comparison among Hive and Spark SQL will be determined. By inspecting the signification of distinct compression strategies and file formats on the performance in diverse query types, we presume that Parquet is supported by Spark SQL more effectively, while it does not exhibit evident advantages for Parquet in Hive as in Spark SQL. Snappy has a better outcome on the intermediary data compression, and in regard to ORC, Parquet integrated with Snappy has the outstanding performance. Second, in Hive, we will alter the default configuration, remodel the number of Map Reduce, optimize the join strategy; dispose of the effects of data skew, making Hive performance enhanced by 10% to 75% or more depending on the workload types. Also, we optimize Spark SQL through the improvement of parallelism and join methods.

In 1980's The (Multiple Query Optimization) MQO problem was proposed and come up with an optimal global query plan adopting the MQO was shown to be NP-Hard problem [13]. Since then, RDBMSs and data analysis applications have ventured a substantial amount of work. In a study during devising multiple queries to establish the aggregate of intra-operator parallelism in parallel databases considering the factors like memory usage, I/O load variables and CPU utilization to deprecate the overall execution interval of declustered join procedures. Beynon suggested a proxy- established infrastructure for managing data profound applications [15]; however, compilation of distributed cache servers obtainable at multiple back-ends is to be expected scalable than this infrastructure. A data incorporation structure that cut down the communication expenditure by a numerous query reconstruction algorithm is put forward by [16].

IGNITE [17] are meaningful studies that put to work the micro machine theory for query operators to drain the total execution time of a query set. A unique MQO structure is advised for the current SPARQL query engines. For gigantic data analysis, a cascade-style optimizer can be preferred for Scope, Microsoft's system.
In recent past, a substantial load of research and commercial venture has centred on merging MapReduce and structured database technologies. Primarily, by two ways we can either adds up MapReduce features to a parallel database or summating database technologies to MapReduce. The second way is more engaging as no broadly obtainable open source parallel database system exists,

whereas MapReduce is accessible as an open source project. Additionally, MapReduce is assisted by a profusion of free tools along with having support and cluster availability. A SQL abstraction across MapReduce platform is provided by Hive, Pig, Scope, and HadoopDB projects to enlighten the programmers with complicated queries. SQL/MapReduce is a modern project that accounts MapReduce to handle user-defined functions (UDF).

In recent past, intriguing studies for unstructured data have been achieved manipulate MQO to MapReduce frameworks for example MRShare treats a bunch of input queries as a distinct query. An optimization problem established on MapReduce cost model is entitled as the optimal clustering of queries for execution. The tentative outcomes noted for MRShare determine its efficiency. In spite of some foremost MQO studies to cut down the execution time of MapReduce-based single queries, to our observation there is no study coinciding to ours that is associated to the MQO of Hadoop Hive by employing *insert query* records.

III. OVERVIEW OF REPRESENTATIVE SYSTEMS

Hive, an open source venture essentially established at Facebook, was the elementary SQL-on-Hadoop come out with an SQL-like query language, designated as HiveQL, and utilizes MapReduce run-time to execute queries. Hive put together HiveQL queries into a chain of map reduce jobs. As SQL-on-Hadoop achieved acknowledgment, MapReduce allotted run-time did not fix out the demanded response times, as a result of high latency in dispatching map reduce jobs. To discuss this subject, Hive is reassigned to a distinct run-time, Tez [10], which can drive DAGs as a specific job, ccompressing the latency in launching jobs. For the time being, the Facebook team advanced a second SQL-on-Hadoop exhibit, known as Presto, which calls a convenational MPP DBMS run-time as a substitute of MapReduce.

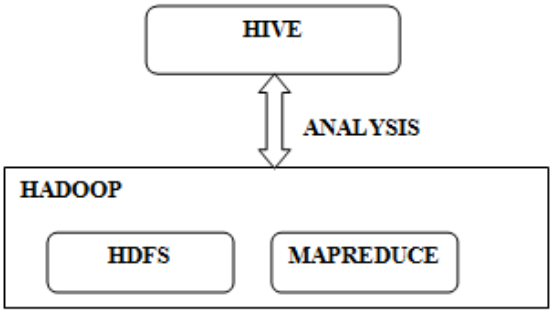

Figure2. Workflow Diagram

Hadapt, which is a prolonged HadoopDB research project, was the early marketable SQL-on-Hadoop offering. File-oriented HDFS storage formats with DBMS-oriented storage, inclusive of column-store data layouts were altered by Hadapt and HadoopDB. Hadapt authorizes two disparate queries run-times: a MapReduce-placed run-time (like the initial Hive run-time) for long queries that depend upon fault tolerance, and a collective MPP run-time for smaller queries.
.
Spark is adaptable with Hadoop data as it is speedy, cluster computing engine and attempts to discuss the demerits of MapReduce. Spark is adopted by three distinct systems as their runtime for SQL handle: SQL Shark, Hive on Spark, and Spark. Apparently, for interactive SQL query processing, the latency of launching jobs for each query is toshardo pricey; there was an alteration to shared-nothing database architectures for SQL processing upon Hadoop data. Hadapt, all employ MPP architectures make use of all Impala, Big SQL, Drill as well as Presto where sequentially acknowledges SQL queries in addition to long-running process coincide with DataNodes on every node in the cluster.

Cloudera Impala is a fully-integrated open-source MPP SQL query engine. On contrary, other systems (often branches of PostgreSQL), Impala is an entirely new engine. Impala reads nothing but disk bandwidth and is generally able to overload all attainable disks. A main attribute of Impala is that it manipulates LLVM to induce code at runtime to accelerated frequently executed code paths.

IBM Big SQL influences contemporary relational database technology of IBM, to develop classic SQL queries on the top of HDFS data, acknowledging all generic Hadoop formats, other than familiarizing any appropriate formats. Big SQL 3.0 allots the identical catalog and table definitions with Hive utilizing the Hive Metastore. Big SQL manipulates prominent query rewrite modifications that are indicated for complex nested decision assist queries. It accounts improved data statistics and a cost-based optimizer to adopt the finest query execution plan. Scheduler services get together by Big SQL that authorizes HDFS blocks to database workers for dealing on a query by query basis.

Apache Drill is an open-source venture that intent at arranging SQL-alike declarative managing on the top of self-specifying semi-structured data. It concentrates on evaluating data without establishing an organized outline or building tables in a catalog similar as Hive MetaStore. It glides queries over files, HBase tables besides detects data when reviewing input data. For each certain portion of data, it distinguishes its schema, establishes an in-memory columnar representation, and induces specific code for processing.

As such, it can coordinate data cluster with variable schemas. Numerous SQL-on-Hadoop systems influences in current use relational database technology.

A substantial grouping of SQL-on-Hadoop be composed of systems that contribute some level of SQL groundwork over HBase data. Auto-sharding and fail over technology is HBase granted for computing tables over multiple servers. It also facilitates updates, functioning over the HDFS, which itself does not endorse updates. HBase balances out to petabytes of data effortlessly over a bunch of commodity hardware. Splice Machine [12] accommodate SQL support over HBase data employing Apache Derby, concentrating on both operational along with analytical workloads. HBase coupled with Hadoop alter the run-time and storage system of Derby. Splice Machine influences the Derby compiler stack to initiate execution plans whatever come to HBase servers (Like SQL stored techniques) which provides the means for promoting computations down to every region (shard) of HBase.
Phoenix implements SQL querying on the top of HBase supported by an embeddable JDBC driver constructed for remarkable proficiency and read/write process. It remodels SQL queries into execution layout possessed of HBase scans. Custom filters and Coprocessors are influenced chronological to boast accomplishment. Phoenix grants secondary indexes in addition to essential medium for joins, both of which are challenging to get with HBase.

IV. CONCLUSION

Hive is a work in progress. It is an open-source venture, and is being actively worked on by Facebook as well as several external contributors. At present, HiveQL admits simply a subsection of SQL as authentic queries. We are at work in the direction of constructing HiveQL incorporate SQL syntax. Hive aids queries conveyed in a SQL-similar declarative language - fasciHiveQL that are compiled into mapreduce task which are accomplished by means of Hadoop. Additionally, HiveQL facilitates users to connect custom map-reduce scripts within queries. In this paper we can also explain why performance of SQL-on-hadoop system is desirable.

REFERENCES

[01] Xiongpai Qin, Yueguo Chen*, Jun Chen, Shuai Li, Jiesi Liu, Huijie Zhang, “The Performance of SQL-on-Hadoop Systems: An Experimental Study” in IEEE, 2017 IEEE 6th International Congress on Big Data.

[02] Xiaopeng Li, Wenli Zhou, “Performance Comparison of Hive, Impala and Spark SQL” in 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics.

[03] Man Zhang1, Fang Liu, Yutong Lu, Zhiguang Chen, “Workload Driven Comparison and Optimization of Hive and Spark SQL’ in 2017 4th International Conference on Information Science and Control Engineering.

[04] Swathi Prabhu, Anisha P Rodrigues, Guru Prasad M S & Nagesh H R, “Performance Enhancement of Hadoop MapReduce Framework for Analyzing BigData”, IEEE 2015, 978-1-4799-608S-9/1S.

[05] White Paper Big Data Analytics Extract, Transform, and Load Big Data with Apache Hadoop-Intel Corporation.

[06] Improving MapReduce Performance Using Data Prefetching mechanism in heterogeneous or Shared Environments Tao gu,Chuang Zuo,Qun Liao , Yulu Yang and Tao Li, International Journal of grid and distributed computing (2013).

[07] “Improve the MapReduce Performance through complexity and performance based on data placement in Heterogeneous Hadoop Cluster” Rajashekhar M. Arasanal, Daanish U. Rumani Department of Computer Science University of Illinois at Urbana-Champaign.

[08] Improving MapReduce Performance in Heterogeneous Network Environments and Resource Utilization, Zhenhua Guo, Geoffrey Fox IEEE (2012)

[09] Improving MapReduce Performance Using Smart Speculative Execution Strategy Qi Chen, Cheng Liu, and Zhen Xiao, Senior Member, IEEE 0018-9340/13/$26.00 ©
2013 IEEE

[10] S. Khalil, S. A. Salem, S. Nassar and E. M. Saad, “Mapreduce Performance in Heterogeneous Environments: A Review”, International Journal of Scientific & Engineering Research, vol. 4, no. 4, (2013).

[11] J. Xie, S. Yin, X. Ruan, Z. Ding, Y. Tian, J. Majors, A. Manzanares and X. Qin, “Improving MapReduce Performance through Data Placement in Heterogeneous Hadoop Clusters”, IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), **(2010)** April 19-23: Arlanta, USA.

[12] X. Zhang, Z. Zhong, S. Feng and B. Tu, "Improving Data Locality of MapReduce by Scheduling in Homogeneous Computing Environments", IEEE 9th International Symposium on Parallel and Distributed Processing with Applications (ISPA), **(2011)** May 26-28: Busan, Korea.
[13] Finkelstein S (1982) Common expression analysis in database applications In: Proceedings of the 1982 ACM SIGMOD international conference on, Management of data. ACM, pp 235–245

[14] Roy P, Seshadri S, Sudarshan S, Bhobe S (2000) Efficient and extensible algorithms for multi query optimization. ACM SIGMOD Rec 29(2):249–260

[15] Beynon M, Chang C, Catalyurek U, Kurc T, Sussman A, Andrade H, Ferreira R, Saltz J (2002) Processing large-scale multi-dimensional data in parallel
and distributed environments. Parallel Comput 28(5):827–859

[16]Chen G, Wu Y, Liu J, Yang G, Zheng W (2011) Optimization of sub-query processing in distributed data integration systems. J Netw Comput Appl 34(4):1035–1042

[17] Lee R, Zhou M, Liao H (2007) Request Window: an approach to improve throughput of RDBMS-based data integration system by utilizing data sharing across concurrent distributed queries In: Proceedings of the 33rd international conference on, Very large data bases. VLDB Endowment, pp 1219–1230

[18] Harizopoulos S, Shkapenyuk V, Ailamaki A (2005) QPipe: a simultaneously pipelined relational query engine In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data. ACM,
pp 383–394