International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Business Intelligence on Hadoop (BIoH)

Rakesh Kumar Sahoo[1], Kaumod Mishra[2]

Senior Technology Architect, Infosys Limited

Technology Lead, Infosys Limited

Abstract: ***This document gives insight into BI/SQL query processing for analytics over Hadoop data. Lately SQL on Hadoop or BI-On-Hadoop has gained significant acceptance in the Big Data Market. Out of all the solutions providing SQL over Hadoop, Hive is the most popular access mechanism on top of Big Data system as most of the BI tools support it. But is it the right and performant access mechanism on Hadoop. In this research, detailed analysis and experimental results, we evaluate various access mechanisms using BI tool on top of Hadoop which includes Hive, Drill, Presto, Spark, Tez and Jethro. Various features, advantages and disadvantages of each access mechanism is described.***

Keywords: BI on Hadoop, Hadoop

1. Introduction

The data volumes are exploding and more data has been created in the past two years than in the entire previous history of the human race.

In a newly released forecast, International Data Corporation (IDC)[11] projects that the worldwide "**Big data technology and services market growing at a compound annual growth rate (CAGR) of 23.1% over the 2014-2019 forecast period with annual spending reaching $48.6 billion in 2019**".

As data grows, it becomes difficult for organizations to store, integrate and access. That's because traditional data warehousing methods are not capable to handle the growth of data. Enterprise looked at Data lakes to resolve the issue, however the assumption was that business users will discover, apply or reuse the structure they need when they're doing their explorations. However directly accessing the data from Hadoop still remains a potential bottleneck wrt. Performance. To overcome this potential performance issue Enterprises have adopted hybrid architecture where Hadoop being considered as the storage and processing layer and only the useful information is moved into traditional database/Appliance (like Oracle, Teradata, DB2, Netezza, Vertica).The BI happens on traditional database/Appliance. But the hybrid architecture requires multiple hop of the same data (from source system to data lake to Traditional DB), requires additional storage space (Storage on Hadoop and Traditional DB), additional investment (Traditional DB apart from Hadoop)

In case the access mechanism is more robust with respect to variety of query, performance and integration with BI tool can be seamless then hybrid architecture will converge into Hadoop only architecture.

In our paper we have explored and benchmarked various access mechanism which can offer better performance on top of Hadoop environment. We have tried to do performance benchmarking using multiple access mechanism using the same data and Same BI tool.

2. Comparative Analysis -BIoH

In this paper, we focus on the BI on Hadoop with connectivity from BI Tools to Hadoop and investigate the performance. Highlighting their different design trade-offs through detailed experiments. The Various tool analysed are Hive, Drill, Presto, Spark, Tez, Jethro.

3. Access Patterns Using Hive

Apache Hive, the data warehouse system for Hadoop, enables data summarization, querying, and analysis of data by using HiveQL (a query language similar to SQL). Hive uses predefined schema of hive metadata to query data without Java or Map reduce knowledge. It is one of the first querying mechanism from Hadoop using SQL like interface. Hence lot of BI tools have adopted connectivity to Hive to access data from Hadoop.

Access from BI tool using Hive uses the following framework depicted in the diagram below –

- BI tools connect through ODBC/JDBC driver to Hive and issues HiveQL Query
- Hive parses and plans the query. The query is compiled , optimized and executed
- Query is converted into map reduce jobs
- Map Reduce is run on Hadoop and result set is sent back to BI tool.

Hive using JDBC\ODBC connects connect to most of the BI Tools like Tableau, Excel, Micro Strategy, Tibco Spotfire, Qlikview, Business Objects and Analytical tools like R, SAS.

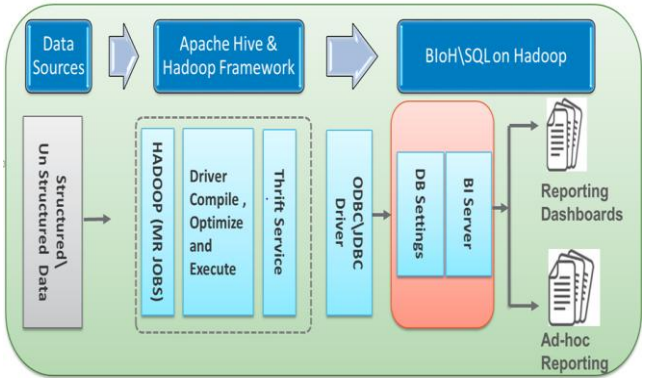

Figure 1: Access on Hive (High Level Architecture)

Advantages of using Hive SQL Solutions is most of the BI tools have predefined connector already available. Hive QL is an extensible solution as it can handle data variety and as volume grows, machines can be added, without a considerable reduction in performance.

Hive is considered one of the first tools installed on almost all Hadoop installations. Hive is very easy to install and doesn't need much of infrastructure to install. However queries run on Hive are very slow and that's the primary reason enterprise are looking for other tools for fast\Interactive SQL Solutions on Hadoop.

Disadvantages with Hive is its response time. Though access mechanism through hive is good for batch processing, it is too slow for interactive analysis.

Hortonworks has been working on the development of Apache Tez as a new back-end for Hive to provide rapid response time's currently not possible using Map Reduce framework

4. Interactive Reporting on Hadoop –DRILL

Apache Drill provides a low latency SQL solutions that can analyze data over distributed networks. Drill is huge in terms of its scaling capabilities, and can process petabytes of data across many thousands of servers, which makes it an ideal solution for Big Data SQL Querying and Analysis.

Apache Drill is similar to Google Dremel except in an open source format, with increased functionality to support a variety of data formats, data sources and query languages.

Access from BI tool using Drill uses the following framework depicted in the diagram below –

- The Drill client issues a query using is a JDBC, ODBC, command line interface or a REST API.
- The Drill bit then parses the query, optimizes it, and generates a distributed query plan that is optimized for fast and efficient execution.
- Zookeeper sends the query to Drill bit that accepts the query.
- The Drill bit schedules the execution of query fragments on individual nodes according to the execution plan.
- The individual nodes finish their execution and return data

Drill using JDBC\ODBC connects to most of the BI Tools like Tableau, Qlik, MicroStrategy, Spotfire, SAS and Excel to interact with non-relational databases and Analytical tools like R, SAS.Developers can leverage Drill's simple REST API in their custom applications to create beautiful visualizations

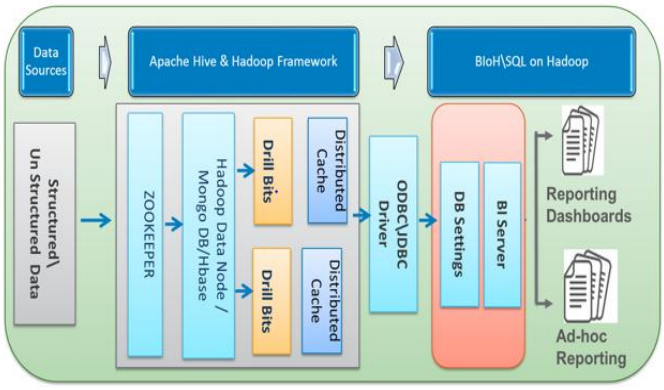

Figure 2: Access on Drill (High Level Architecture)

Advantages of Drill SQL Solutions is its ability to bring all Sql landscape and performance of the RDBMS to Hadoop scale data without compromising on the flexibility of Hadoop/NoSQL systems. Furthermore it offers a flexible, low-latency SQL engine, a requirement in big data analytics .The biggest advantage of drill is it's schema less i.e. No schemas or ETL or No schema management needed.

The differentiating factor with Drill is the ease in the creation of User defined functions through an easy to use, high performance Java based API.

Disadvantages of drill are it can't support nested Data in Map, Array, and repeated scalar types in GROUP BY or ORDER BY clauses or in a comparison operator

Latest version of Drill is out of alpha (version 1.2 as of October 2015), so it's certainly becoming more stable as time progresses.

5. Access Analysis on Presto

Presto is an open source project designed by Facebook .Presto is a distributed SQL query engine for running interactive analytic queries against petabytes of data. Facebook designed presto with the objective to create a Big data Sql solution which can create queries with response similar to that of commercial data warehouses

Key Highlights of presto are its custom query execution engine, query execution in memory to avoid unnecessary I\O. It Supports the ANSI SQL standard, including complex queries, aggregations, joins, and window functions. Presto has a special Tableau Web API to lets users run queries from Tableau against Presto. Presto also has unique data security feature by requiring no data to be stored on any device.

Access from BI tool using Presto uses the following framework depicted in the diagram below –

- Clients submit SQL statements
- Sql get parsed and planned
- Parallel tasks for SQL are scheduled to workers which jointly process rows from the data sources.

The Final data set gets are returned to the client.

Presto allows maintaining leverage among distributions like Hortonworks and Cloudera while allowing portability among them and therefore BI tools supported by Presto don't depend

on the underlying Hadoop distributions. Presto support all BI Tools via the ODBC and JDBC drivers Tableau, Excel, MicroStrategy, Spotfire, Qlikview, and Business Objects.

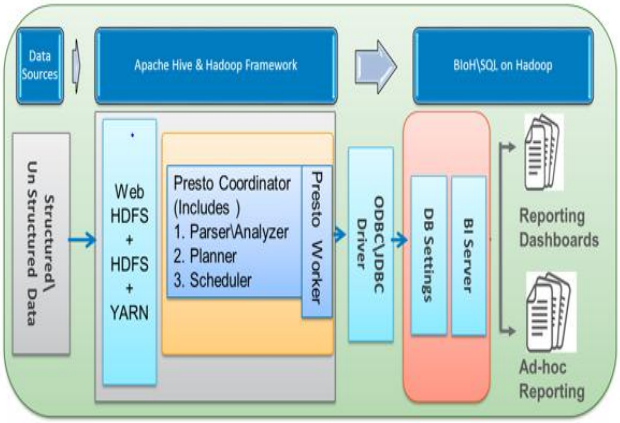

Figure 3: Access on Presto (High Level Architecture)

Advantage with presto is it's feature to combine data from multiple sources, allowing for analytics across your entire organization .Presto works great with petabytes of data and can give Response times ranging from sub-second to minutes.

Presto query execution is extremely fast because of in-memory processing and in turn proper CPU optimizations, this translates to highly interactive query performance

The main **disadvantage** with Presto is its limitation to join. It also lacks the capability to write back the output data to tables and can only be streamed to Client BI systems.

6. Access Analysis on Jethro

Jethro is an acceleration layer that speeds up BI dashboards & visualizations. Jethro can speed up the queries to 500%-5000% without forfeiting any data when compared with traditional Map Reduce jobs in flexibility or data freshness.

Jethro Data uses the advantage of full indexing and a columnar structure to get good response on big data queries. To Meet Concurrency requirement, nodes in Jethro are highly elastic, allowing it to easy scale-out. Jethro's index and column files in Propriety format are stored as standard files on HDFS or Amazon S3 and benefit from their native scalability and high availability. Jethro leverages its indexes to surgically fetch only the required data and thus reducing the load on the shared Hadoop cluster. Works on ANSI SQL commands, Extend the system by writing connectors for it using its service provider interface

It is compatible with all Major BI Vendors like Tableau, Qlik, and home-grown SQL-based SaaS BI dashboards. BI tools connect to Jethro Data using JDBC or ODBC Driver. The driver automatically load-balances SQL statement across all Jethro Data hosts.

Access from BI tool using Jethro uses the following framework depicted in the diagram below –

- Clients submit SQL statements
- Sql get parsed and indexed
- SQL is run using the ODBC\JDBC Connectors using the Jethro Acceleration Layer which process rows from the data sources.
- The Final data set gets are returned to the client.

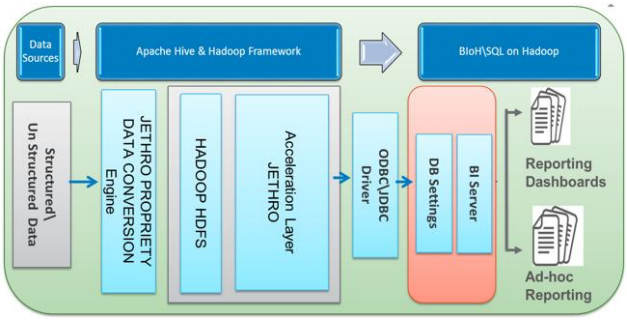

Figure 4: Access on Jethro (High Level Architecture)

Advantages with Jethro are its data formats which are even faster than the generic formats like ORC and Parquet however the downside is that all files will have to be converted to the vendor's proprietary format. Its propriety format uses the multi-threaded processing, and vectorization of queries to boost performance. Jethro processes using less of I/O load on HDFS and thus offloading huge percentage of SQL processing from Hadoop and which helps cluster to load balance between online users and batch processing

Jethro runs machines which are higher-end hosts and are configured with – with extra memory and CPU cores to support faster query performance using Jethro's local SSD for caching. Jethro is new to the market when compared with competitor, but has made its mark in BIoH or Sql on Hadoop Solutions

Disadvantage with Jethro is its mandatory File data conversion to Jethro's proprietary format

7. Access Analysis on Spark

Apache® Spark™ was developed in the AMPLab at UC Berkeley. As per Apache "**Apache® Spark™ is an open-source cluster computing framework with in-memory processing to speed analytic applications up to 100 times faster compared to technologies on the market today**". Apache Spark can help reduce data interaction complexity, increase processing speed and enhance mission-critical applications with deep intelligence.

Apache Spark is popular for its simplicity in creating algorithms that use data from Variety of data and process them. Spark was elevated to a top-level Apache Project in 2014 and continues to expand today.

Key Highlights of Apache's Spark project is for real-time, in-memory, parallelized processing of Hadoop data .Apache's SQL supports an array of advanced data analytic operations including SQL queries, and procedures pertaining to complex analytical functions

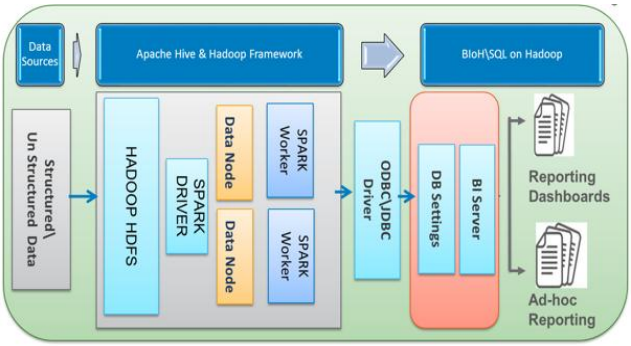

Figure 5: Access on Spark (High Level Architecture)

Access from BI tool using Spark uses the following framework depicted in the diagram below –

- Clients submit SQL\Hive QL statements using the command-line or over JDBC/ODBC.
- Sql get optimized
- SQL is run using Spark Engine (Spark Driver, Data Node and Spark Worker).
- The Final data set gets are returned to the client

Advantages of Apache spark is the Reduced Disc read and Write that it achieves through the Resilient Distributed Dataset (RDD), which allows it to store data on memory and hence reduce effort on data processing factor. Spark has Unparalleled fault tolerance capability to ensure recovery of sql data from all kinds of failure.

Disadvantages of Spark is it Consumes lot of Memory and also it is still working at the bugs from previous releases like on SQL Optimizer and Improvements in interactive query response time

8. Access Analysis on Hive on Tez

Apache Hive on Tez provides a framework that is highly extensible which helps in data-processing tasks across both Batch processing tasks and also for interactive queries. Coordinated by Apache Yarn in Hadoop. Apache Tez generalizes the standard Map Reduce program to a more powerful framework by creating a process to run complex DAG (directed acyclic graph) of tasks for a single job. DAG (Directed Acyclical Graph) workflow allows, parallelization, data routing, and DAG component reuse within and between computational executions to meet requirements of faster query response times and extreme throughput at petabyte scale

Key highlights of Apache Tez Apache is its additional framework which can be used for building high performance batch and analysing huge volume of data and processing applications, coordinated by YARN in Apache Hadoop. This helps in faster query response, while retaining MapReduce's ability to scale to petabytes of data. Other key feature is its customizable execution architecture which helps users to express complex computations on Sql and also in dynamic data performance optimizations

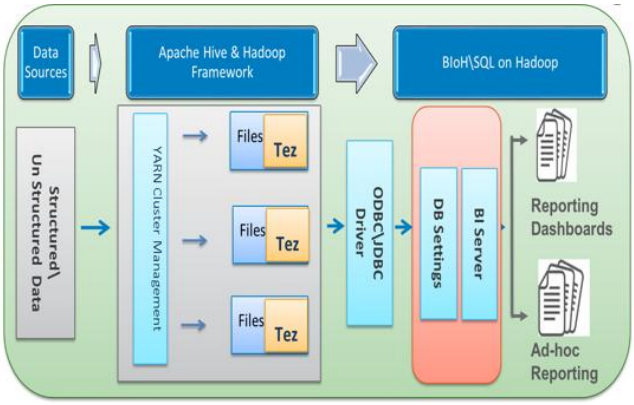

Figure 6: Access on Apache Tez (High Level Architecture)

Access from BI tool using APACHE Tez uses the following framework depicted in the diagram below –

1) Clients submit SQL statements using the command-line or over JDBC/ODBC.
2) Uses YARN Cluster Management to for firing SQL to available nodes
3) Sql get planned and optimized
4) SQL is run using Tez Engine comprising of
   a) DAG : Data processing
   b) Vertex: Defines user logic , resource and Environment
   c) Edge: Defines connections between different components.
5) The Final data set gets are returned to the client

Advantage of APACHE Tez (open source project) is faster data processing over Map Reduce under Hadoop's traditional data-processing languages. Additionally Tez provides a rich set of SQL features such as analytic functions, query optimization, and standard data types such as timestamp etc. The query performance for Apache Tez is its Engine which preparer better query plan in comparison to Map reduce.

Disadvantage with Tez is it not meant directly for End users and needs developers to build end user Sql Solutions

9. Benchmarking of various BIoH solutions

We took sales Transaction file of size 200 GB each (in CSV format). We used the same data type for each column in order to eliminate the impact of diverse data types on scan performance. We used a Data Generation tool to create such a huge volume of data. Thus, datasets with arbitrary Random values would not be able to exploit this feature BIoH \SQL on Hadoop Analysis

1) 10 Million sales transactions
2) 200 GB of Data for Analysis
3) Benchmarking by End Users using Self Service BI tool Tableau
4) Access Mechanism Evaluated by
   - Drill 1.5.0
   - Hive 2.3.4.0
   - Presto 0.141
   - Hive 2.3.4.0-3485
   - Jethro 1.4.3

Using this Data we created a common dashboard as shown below

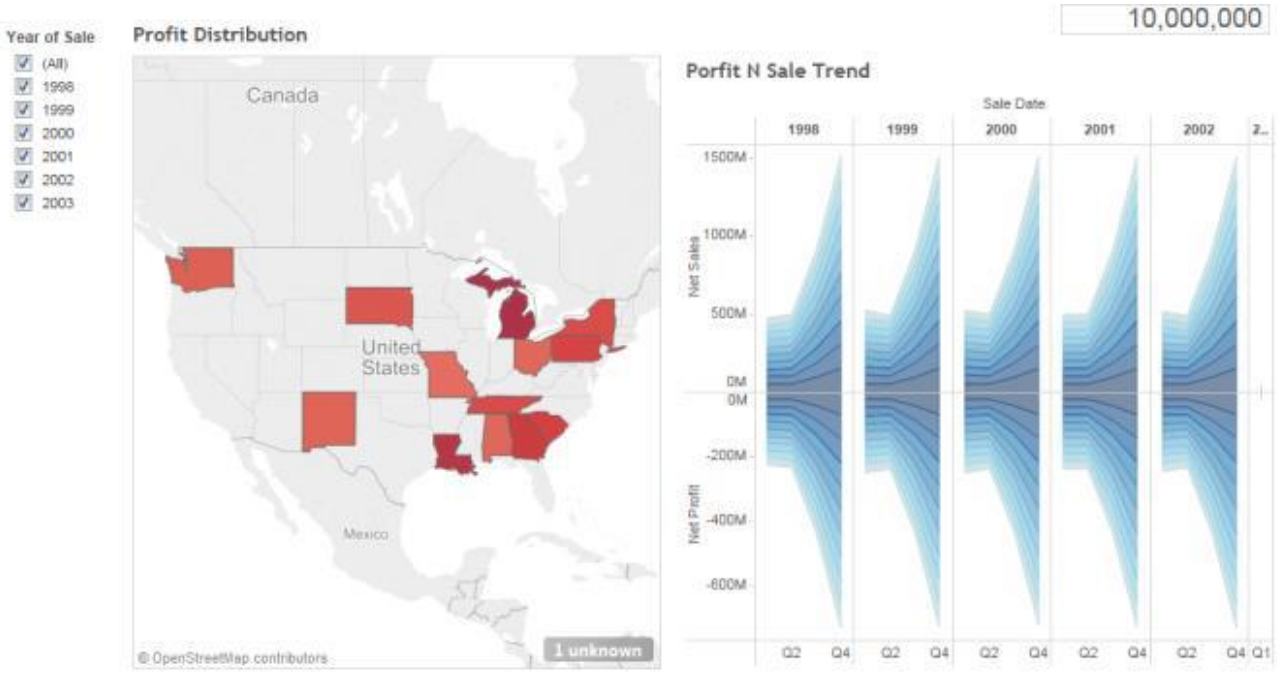

Figure 7: Dashboard on Tableau

We captured the response time for two important criteria

Tableau: Response time and Filter Response time.

Figure 8: Query Response time of BIoH Solutions

Response time for the filter to work

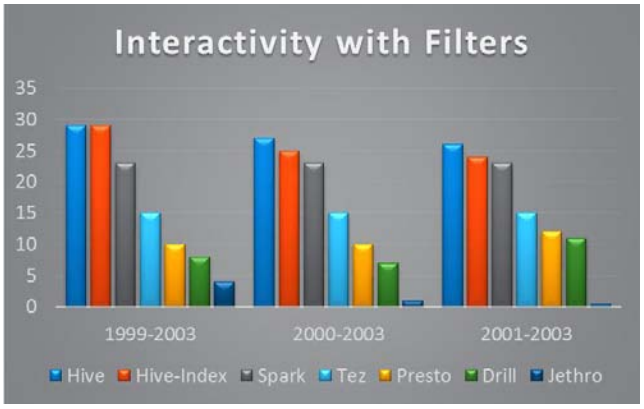

Figure 9: Report Filter Response time of BIoH Solutions

10. Conclusion

Increased need to analyze huge volume of data has fueled the need for solutions Like BIoH or SQL-on-Hadoop. In this paper, we conducted an experimental evaluation of Hive, Drill, Presto, Jethro and Tez among the various BIoH or Sql on Hadoop systems. We find that Jethro and Presto were giving response time for queries on run time much better than others

Hive on Map Reduce pays the overhead of scheduling and doesn't match the performance needed for analyzing the data interactively. However, both Hive-MR and Hive-Tez are CPU-bound during scan operations, which negatively affect their performance.

The performance for Jethro was the best in term of query response time and the difference is attributed to Jethro's very efficient indexing and I/O sub-system and to its pipelined query execution which resembles that of a shared nothing parallel database.

References

[1] "Hadoop Releases".Apache.org. Apache Software Foundation. Retrieved 2014-12-06.
[2] "Hadoop Releases". Hadoop.apache.org. Retrieved 2016-01-31.
[3] "Welcome to Apache Hadoop!" http://hadoop.apache.org/. Retrieved 2015-12-16.
[4] HDFS-Architecture https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
[5] Apache Hive https://hive.apache.org
[6] Apache Drill http://drill.apache.org/
[7] Apache Tez http://tez.apache.org/
[8] Presto https://prestodb.io
[9] Jethro http://jethro.io/
[10] Spark : https://spark.apache.org
[11] IDC: http://www.idc.com/
[12] Forbes:http://www.forbes.com/sites/bernardmarr/2015/09/30/big-data-20-mind-boggling-facts-everyone-must-read/#73bbf85e6c1d
[13] Sql on Hadoop: Secret- http://conferences.oreilly.com/strata/stratany2013/public/schedule/detail/31731

[14] Hive and Big SQL Performance Test on Hadoop : https://developer.ibm.com/hadoop/2015/10/23/hive-and-big-sql-performance-test-update/
[15] Getting Started with Interactive SQL-on-Hadoop : http://conferences.oreilly.com/strata/big-data-conference-ca-2015/public/schedule/detail/39010