Performance evaluation sql-on-hadoop: a case study of Hortonworks and Cloudera

M. Faridh Ronianto\textsuperscript{1}, Ibnu Asror\textsuperscript{2}, Sidik Prabowo\textsuperscript{3} and Fajar Arief Nugraha\textsuperscript{4}
Telkom University, Bandung

E-mail: \textsuperscript{1}faridhronianto@student.telkomuniversity.ac.id, \textsuperscript{2}iasror@telkomuniversity.ac.id, \textsuperscript{3}pakwowo@telkomuniversity.ac.id, \textsuperscript{4}fajar@telkom.co.id

Abstract. Currently companies in the world have focused on the Big Data business which has become an invaluable tool in assisting business processes and data analysis. SQL-on-Hadoop is a small part of the Big Data Platform that has been developed to date. Our research implements Big Data Platform on Cloudera and Hortonworks using TPC-H Benchmarks on SQL-on-Hadoop systems and evaluates the characteristics and performance of query processing machines in each scenario applied to each platform. We focuses on evaluating the two Big Data Platforms, Cloudera and Hortonworks, to determine the advantages and disadvantages of each platform based on the TPC-H Benchmark that has been recognized as a Decision Support System to compare the two with four different scenario that run on the same configurations. The results obtained are Cloudera with Impala can process queries with a ratio of up to 41x faster than LLAP-Tez and 200x faster than Hive-Spark.

1. Introduction
Increased data will affect processing and access to data, especially in its distribution. The amount of data will affect the speed and accuracy of processing large data that will be needed [1]. Big Data is a term that refers to a data set or combination of data sets whose size (volume), complexity (variability), and growth rate (speed) make them difficult to store, manage, process, or analyze with conventional technologies and tools such as Management Systems National Database (RDBMS) and Desktop Statistics or visualization packages, when needed to make it useful [2].

The company has now increased the use of the platform as a platform for processing such huge data by implementing Big Data. Apache Hadoop [3] is the right solution to answer the problem. However, using just one platform is not enough to meet the needs of complex companies. The Data Platform is a type of solution that combines the features and capabilities of several applications and utilities such as Cloudera [4] and Hortonworks [5]. One type of solution given is that Big Data processing using complex queries will affect the results of execution time. Both platforms have different query processing scenarios to speed up execution time.

Therefore, research is needed to evaluate how the performance between the two platforms is on the storage side and processing on Big Data when using a SQL-on-Hadoop system. The author aims to evaluate the efficiency of several query execution engine scenarios between two Big Data Platforms by explaining each machine execution scenario such as storage type and the
difference in execution time of each machine. By using TPC-H Benchmark [6] to test the system using Cloudera and Hortonworks as a basis for testing so that the results of this study will be compared with subsequent research on the SQL-on-Hadoop system.

2. Related work

In the study [7], the authors compared the advantages and disadvantages of query optimizer and query engine execution in Big Data Frameworks such as Impala, Drill, Phoenix, and Spark SQL. The author compares the scale factors and database storage formats when added to each system. Micro benchmarks and TPC-H benchmarks are used to evaluate the system. As a result, Impala is superior to other systems with a scale of 4.41x - 6.65x in text format.

In the study [8], the authors evaluated the performance of Big data systems in OLAP-style and used TPC-H Benchmarks to get results from four Big Data Platforms, namely Hive, Spark SQL, AsterixDB and System-X (parallel commercial RDBMS). This research produces no system, storage format, and variants that will give the best performance on the query. However, the different systems show different enhancement behaviors.

In research [9], the author focuses on three levels, distributed file systems, NoSQL, and SQL-like components on Big Data Ecosystems Hadoop. At the first level there is HDFS, Ceph, Luster and concludes that HDFS is more suitable for Big Data. On the second level there is HBase and MongoDB where memory is limited and MongoDB efficiency is lower than HBase while “reading” performance is better than HBase. At the third level there are Hive, Pig, and Phoenix using the TPC-H Benchmark which results in the overall appearance of Pig being lower than two opponents.

In this study [10], the authors conducted an experiment on the performance of the SQL-on-Hadoop system which began with a brief technical review of the latest efforts of this system. After that, the author provides comparative performance of three system representations namely Hive, Impala, and Spark SQL in SQL-on-Hadoop based on TPC-H Benchmark with the conclusion that by providing high performance SQL analysis functionality for data stored in HDFS will attract more and more users to use SQL-on-Hadoop for interactive analysis as an alternative to exclusive DBMS.

In the study [11], the authors analyzed the performance of the SQL-on-Hadoop Framework based on the column database system by applying the TPC-Benchmark to measure and analyze with Hive, Impala, and Tajo. This study uses two large file formats namely ORC and Parquet along with various other performance effects. The results show that Impala outperforms Hive and Tajo by 5x and 10x when the database workload matches its memory.

In this thesis research I will do research with reference to [11], [12], and [8] by comparing the query performance of SQL-on-Hadoop execution machines for both Big Data Platform based on TPC-H Benchmark.

2.1. Sql-on-hadoop systems

There are many RDBMS systems that are present today with SQL support to analyze data and provide interactive responsibility. Here are some reasons why SQL runs on the Hadoop system [10].

- Cost effective. Hadoop can run in large groups of commodity hardware and has better costs than MPP databases (massive parallel processing) that require high-end servers that are expensive and cannot be made up to thousands of nodes.
- High I/O bandwidth. With large volumes of data is processed in memory and the rest is stored on the hard drive. Distributing I/O to larger clusters is one of the benefits of MapReduce.
• Supports complex analysis by providing SQL capabilities, Machine learning, and Data mining functionality

2.2. Tpc-h benchmark
TPC-H Benchmark is an analytic systems industry standard that distributes its specifications for public and can run on any supported database environment [13]. In this research we use TPC-H as the database workload system that can be thought for represent the typical workload business users inquiring about the performance of their business in the company. Its consist of many ad-hoc queries and data modifications that have been chosen to have a road industry relevance.

2.3. Text and orc format
The text format is the famous input / output format used in Hadoop. In Hive if we define a table as text it can load data from CSV (Comma Separated Values), separated by Tabs, Spaces, and JSON data. This means the fields on each record must be separated by commas or spaces or tabs or maybe JSON data (JavaScript Object Notation). By default, if we use the text format, each line is considered as a record. While ORC is a data column format designed for Hadoop workload. ORC is optimized for reading large streams, but with integrated support to find the required lines quickly. Saving data in column format allows the reader to read, decipher, and process only the values needed for the query [14].

3. Proposed method
3.1. Hardware Configuration
To obtain results, we will use standard development configuration using one server or can be called a node with specifications a Intel Nuc 2.3GHz 4-core processor, 32 GB RAM, and 1 TB SATA SSD. We use a single node cluster because the database that is accommodated is quite adequate and can run well and efficiently in this study for make a relevance database that simply run in industry.

3.2. Software configuration
For software configuration we use a dual boot operating system consisting of CentOS 7 version Gnome desktop for our Cloudera Distribution including Apache Hadoop (CDH) with version 5.14.2 and Ubuntu Desktop 16.04 Version desktop Gnome for Hortonworks Data Platform (HDP) version 2.6.4. To support SQL-on-Hadoop research, it only focuses on services that support SQL such as MapReduce v2, Hive, Impala, and Spark-SQL with the default versions of CDH and MapReduce v2, Hive 2.1.0, Tez 0.7.0, Spark-SQL 2.2.0, and Low Latency Analytic Processing (LLAP) on HDP. This is because the TPC-H database workload can simply work and run with those services and achieve the purpose of our research.

3.3. System architecture
In this research, CDH and HDP will be carried out by applying 2 layers, namely Data Storage and Data Process Layer. In figure [1] shows CDH consists of Hadoop Distribution File System (HDFS) as a Storage Layer and in the Data Process Layer using Other Resource Management (YARN) combined with MapReduce, Hive, Impala, and Spark-SQL. In figure [2] shows HDP which has the same principle, but there is data on the Data Process Layer that combines YARN with MapReduce, Hive, Tez, LLAP, and Spark-SQL.
3.4. Database workload
In this study using the famous TPC-H workload to evaluate SQL-on-Hadoop systems such as Hive, Impala, SparkSQL, Tez, and LLAP that will be carried out in this download. This benchmark consists of 8 tables and 22 ad-hoc read-only queries without refresh streams. The author combines a database available on TPC-H with a scale factor of 100 GB. Research is limited to 100 GB due to the constraints imposed by Impala that require a suitable collection of workloads into memory. However, this database is sufficient to provide a significant level of view to evaluate the advantages and disadvantages of this system [11].

4. Experiment result and analysis
To achieve the purpose of our research we will conduct four different scenarios in both platform to see how are the database workload handled our TPC-H Queries. For First scenario, there are only MapReduce framework used in both CDH and HDP to see the performance. Next Second is a little different where at CDH we used Hive-on-Spark and HDP use Hive-on-Tez and how the performance going. Third which use same framework Spark-SQl but at different platform. Finally the last Scenario we put the two service that is special to both platform CDH with Impala and HDP with LLAP. The performance of the execution time and how the platform complete all the query ad-hoc will be the parameter of our evaluation.

4.1. First scenario

![Figure 1. CDH layer.](image1)

![Figure 2. HDP layer.](image2)

![Figure 3. Hive-MapReduce on HDP and CDH](image3)
In figure 3, describes the first scenario that Q6 with the ORC format type on CDH is the most optimal with 328 seconds compared to HDP with 495 seconds. This happens because the query only runs with 62 Map phases and 1 Reduce phase to run 1 job sort. In contrast to Q21 with the ORC format on CDH the longest execution time is 16,871 seconds with a total of 13 jobs with each phase approximately 62 Map phases and 264 phases Reduce, while HDP is 14,615 seconds due to this query having a Query Cost of 17%. In text format, the time produced is much slower due to the larger number of sizes that cause the Map and Reduce phases, and the Job increases. Overall in figure 10, there is no significant difference between Hive on MapReduce HDP which is 1.2x faster than CDH.

4.2. Second scenario

In figure 4, the second scenario as a whole shows that Hive-Tez and Hive-Spark are much faster than the previous scenario because the two platforms have different technology characteristics with MapReduce that processes queries more efficiently in this study. The fastest execution time lies in Hive-Tez with Q6 ORC format of 36.34 seconds and text format 191.54 seconds. This query requires 1 job with 18 Map phases and 1 Reduce phase, while in Hive-Spark with 59 seconds ORC and 492 seconds text format. Inversely proportional to Q7 the text format on Hive-Tez becomes the longest time with 6223 seconds with 46 Map phases and 2 phases of Reduce and Query Cost of 5%, whereas in the Hive-Spark it is much faster reaching 2282 seconds. Overall in figure 10, Hive on Tez HDP is 1.4X faster than the Hive-Spark CDH, and 5.1X is faster than the Hive-MapReduce ORC and 7.6X formats in the text format.

4.3. Third scenario

In figure 5, explains the third scenario with Spark-SQL a little slower than the previous scenario because in Spark-SQL all queries are divided into 1 job, but have different stages depending on the query executed which will be processed by the executor of each platform. This can be seen from the fastest execution time located on HDP in Q11 ORC format of 63 seconds with 400 stages of read shuffle, while the text format is 111 seconds which is slightly faster than CDH.
with 98 seconds in ORC format and 184 seconds in text. Inversely proportional to Q17 is the highest query with 4580 seconds in text with 708 stage read shuffle and ORC equal to 2986 seconds, while CDH can process the query with only 629 seconds at ORC and 699 seconds in text. Overall in figure 10, Spark-SQL with ORC format on HDP is 1.3X faster than CDH and 1.2X faster on Spark-SQL with text format.

4.4. Fourth scenario

Figure 5. Spark-SQL on HDP and CDH

Figure 6. LLAP-Tez on HDP and Impala CDH
In figure 6 describes the fourth scenario with LLAP and Impala far faster than the execution time of the previous 3 scenarios. This can be seen from the fastest execution time on LLAP HDP with ORC format with only 11 seconds and Impala CDH with 1 second. This occurs because LLAP runs work fragments using executors which basically "replace" each daemon that will be used by one task at a time. It is different from Impala who owns State store which divides Query Planers, Query Coordinators, and Query Execution Engines above HDFS where all of them have supported Massively Paralel Processing (MPP). Inversely proportional to Q18 is the highest query with 1899 seconds, while Impala CDH can process the query with only 9 seconds. Overall in figure 10 Impala on CDH is 41X faster than LLAP-Tez on HDP.

4.5. Analysis

![Figure 7. The result comparison between CDH](image)

![Figure 8. The result comparison text to orc format](image)

![Figure 9. The result comparison between HDP](image)

![Figure 10. The result comparison between CDH and HDP](image)

Experiments have been done with some of the above scenarios and we have found new things where impala compared to LLAP in Hortonworks has a far better performance when faced with a dataset that is suitable for our research at figure 7 can be seen in the comparison between platforms running on CDH with the largest ratio of Impala with 200X faster than Hive-Spark. Figure 9 is a platform comparison that runs on HDP with the largest ratio of Hive-Tez with ORC 11X format faster than MR-Hive. Figure 8 is the overall comparison of text format changes to the ORC with the largest value in LLAP-Tez with 5.7X faster. Figure 10 is a comparison between the two platforms with Impala 41x faster than LLAP in the ORC format.

5. Conclusion

Two Big Data Platforms have been implemented on the sql-on-hadoop system based on TPC-H Benchmark with their respective configurations. The results of the experiment and analysis
have been carried out and the results of the comparison of performance evaluations are obtained with different characteristics and execution times of each platform. The results of the analysis carried out with both Big Data Platforms can successfully run on a single node cluster. Based on our test results obtained MR-Hives on both CDH and HDP produces the longest time in processing the query. Furthermore, it is continued by the Hive-Tez and Hive Spark platforms, and SparkSQL with performance results that are not much different with a comparison value of up to 1.5X. However, a significant difference occurred in the Impala CDH and LLAP-Tez HDP platforms which resulted in much better performance. In our case, Impala can operate 41X better than LLAP-Tez on HDP followed by 141X faster than Spark-SQL, and 200X faster than Hive-Spark. For further research we will conduct more dataset, more format compression data, and any other sql-on-hadoop framework in line.

References
[1] Navya Francis and Sheena Kurian K. “Data Processing for Big Data Applications using Hadoop Framework”. In: International Journal of Advanced Research in Computer and Communications Engineering 4.3 (2015), pp. 1–4.
[2] Yassir Samadi, Mostapha Zbakh, and Claude Tadonki. “Comparative study between Hadoop and Spark based on Hibench benchmarks”. In: Cloud Computing Technologies and Applications (CloudTech), 2016 2nd International Conference on. IEEE, 2016, pp. 267–275.
[3] Apache Hadoop. HDFS Architecture Guide. URL: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
[4] Cloudera. Machine Learning, Analytics, Cloud - Cloudera. URL: https://www.cloudera.com/
[5] Hortonworks. Data Management Platform. URL: https://hortonworks.com/
[6] TPC. TPC - Benchmark. URL: http://www.tpc.org/default.asp
[7] Yueguo Chen et al. “A study of SQL-on-hadoop systems”. In: Workshop on Big Data Benchmarks, Performance Optimization, and Emerging Hardware. Springer, 2014, pp. 154–166.
[8] Pouria Pirzadeh, Michael Carey, and Till Westmann. “A performance study of big data analytics platforms”. In: Big Data (Big Data), 2017 IEEE International Conference on. IEEE, 2017, pp. 2911–2920.
[9] Qing Liu et al. “Big Data Management Performance Evaluation in Hadoop Ecosystem”. In: 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM). IEEE, 2017, pp. 413–421.
[10] Xiong pai Qin et al. “The Performance of SQL-on-Hadoop Systems-An Experimental Study”. In: Big Data (BigData Congress), 2017 IEEE International Congress on. IEEE, 2017, pp. 464–471.
[11] Awais Mehmood et al. “Performance analysis of shared-nothing SQL-on-Hadoop frameworks based on columnar database systems”. In: Innovative Computing Technology (INTECH), 2016 Sixth International Conference on. IEEE, 2016, pp. 128–133.
[12] Xiaopeng Li and Wenli Zhou. “Performance comparison of hive, impala and spark sql”. In: Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2015 7th International Conference on. Vol. 1. IEEE, 2015, pp. 418–423.
[13] HammerDB. Apache ORC - High Performance Columnar Storage for Hadoop. 2018-10-19. URL: https://www.hammerdb.com/docs/ch09s01.html
[14] Apache ORC. Apache ORC - High Performance Columnar Storage for Hadoop. 2018-8-16. URL: https://orc.apache.org/