https://dx.doi.org/10.4314/dujopas.v8i3b.10
ISSN (Print): 2476-8316
ISSN (Online): 2635-3490
Dutse Journal of Pure and Applied Sciences (DUJOPAS), *Vol. 8 No. 3b September 2022*

Evaluation of Open-Source Tools for Big Data Processing

[1]Umar Suleiman Ahmad, [2]Abubakar Muhammad Miyim, [3]Muhammad Salisu Ali
[1]Department of Computer Science,
Federal University Dutse,
Nigeria.

[2]Department of Information Technology,
Federal University Dutse,
Nigeria.

[3]Department of Cyber Security,
Federal University Dutse,
Nigeria.

Email: umar.sahmad@fud.edu.ng

Abstract
Every day, large terabytes of data repository are being generated which comes mostly from modern information systems, new technologies, Internet of Things (IoT) and cloud computing. With the ever-expanding number of alternatives, the choice of picking machine learning tools for big data to analyse such volume of massive data can be difficult and so necessitates exertions at various stages to excerpt information meant for decision making. As big data analysis is currently the latest researchable area of interest, this paper therefore intended to aid researchers understand machine learning and focus on exploring the impact of open-source tools for the processing of big data. Machine learning was used to analyse three open-source tools of Hadoop, Spark and Presto These open-source tools were evaluated by considering scalability, fault tolerant and latency as the metrics. While Presto as a tool for big data analytic was discovered to be efficient and fast in processing huge data, spark plays greater role in precision and Hadoop was found to be the best in fault tolerance. In conclusion, the paper furnishes the platform with various steps to explore big data that could open latest sphere of research development.

Keywords—MapReduce; Hadoop; Spark; Presto; K-Means; Structured Data; Unstructured Data

INTRODUCTION
The fast transformation of the digital technologies and the huge data generated from multiple sources has brought about the growth of big data. This development has led to the discoveries of how bulky and complex datasets are collected from different sources. The collection of such datasets requires so much energy in undertaking the task when applying conventional database management tools or data processing platforms. This type of datasets is grouped into either structured, semi-structured or unstructured format in big volume of petabytes and more. In literatures, it is defined as quantities in terms of Volume, Velocity and Variety (3Vs) and Volume, Velocity, Variety and Veracity (4Vs) presenting the characteristics of big data as depicted in figure 1. The 3Vs here is driven and refers to volume which stands for the big data

**Author for Correspondence*

that are being created daily, velocity relates to speedy generation of data and collated for analysis while many issue information about data types. The fourth V however, stand for accuracy that combines availability and accountability. However, the key objective of big data analytics is the processing of large volumes datasets, velocity, variety, and veracity using different techniques for computational intelligence (Kakhani *et al*., 2015). Figure 1 refers to the brief definition of big data that is problem specific that assist in decision making, comprehensive findings and optimization while being inventive and worthwhile.

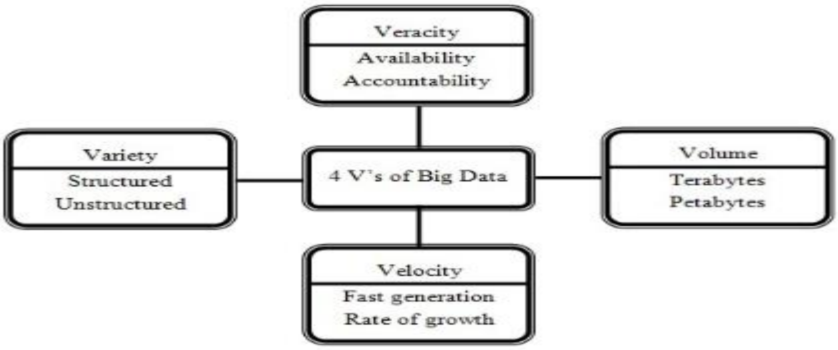


Figure 1: The 4V's of Big Data (Source: Snehalata *et al*., 2014).

Taking a look at the perspectives of information and communication technology (ICT), big data seems to be a robust push for the next generation of information technology industries (Jin *et al*., 2015), which here refers to big data, cloud computing, internet of things (IoT) and social business. In essence, it is the data warehouses that have been managing the large datasets by extracting knowledge from such available data. It is clear that data most mining approaches lack the ability to successfully handle enormous datasets. The main setback in analysing big data includes the lack of coordination between database systems and the tools for the analysis (data mining and statistical analysis). The challenge arises as a result of performing knowledge discovery while finding its practical applications. Additionally, studying the complex theories of big data leads to understanding of vital characteristics as well as complex patterns generation in big data analytics. Its representation gives a better knowledge that could guide the design of computing models and algorithms for big data (Jin *et al*., 2015). However, the interest of both the academia and the industry are in propagating the findings of big data, only and not interested in the usefulness of the analysis.

From the review, most of the studies in this field attempt analysing and assessing some open-source tools for big data analytic using traditional methods. Some of the researchers (Aye, 2013) compared Hadoop and GlusterFS and arrived at solving the problem of volume and variety issues but unable to solve issues on velocity. Researchers like Gureev, (2018) did compared three of the tools: Hive, Spark and Presto on ORC, Parquet, LLAP and Tez functionalities in tackling the problems encountered with data size and velocity. Consequently, it is understood that most of the issues encountered in handling big data are the data size and processing speed of such data.

From the related works presented and analysed in this research, it is clear that no research work of this nature, to the best knowledge of the researcher have been conducted on scalability, fault-tolerant, query response time and data management using Hadoop, Spark and Presto. It is against this backdrop that the research chooses the title of the thesis as; Evaluation of Open-Source Tools for Big Data Processing. The focus is to determine which among these open-source tools (Hadoop, Presto and Spark) stand out to be the best in terms of scalability, fault-tolerant and query response time, speed. This research intends to solved velocity issue as well as serve either directly or indirectly in choosing the best data warehouse and to appropriately eradicate the gap of some missing data.

The primary focus of this work was to make comparative analysis among the three open-source tools (Hadoop, Spark, and Presto) for big data analytics in terms of their query response time, ability to scale and input/output process. Therefore, the dataset used in this work is from Facebook online free download and Amazon.

LITERATURE REVIEW

There exist a lot of big data analytics open-source tools to process and analyse huge data and some of which are here discussed with some techniques for analysing big data with emphasis on three emerging open-source tools namely: Hadoop, Spark, and Presto (Chenga, 2015). Majority of these tools focuses on batch and stream processing, as well as interactive analysis as most of these tools are built on Apache Hadoop. Venkatesh & Ahamed (2017) discussed data streaming applications as tools used in real-time analytics in big data. Though most of the massive streaming platforms include storm and splunk where the ones for interactive analysis permit users interact straight with their own analysis in real time. Some of the tools for big data analytics have been debated by such authors like Funde *et al.*, (2019) and Nikita, (2018). A typical workflow of big data analytics highlighted by Hong et al., (2019) is depicted in Figure 2.

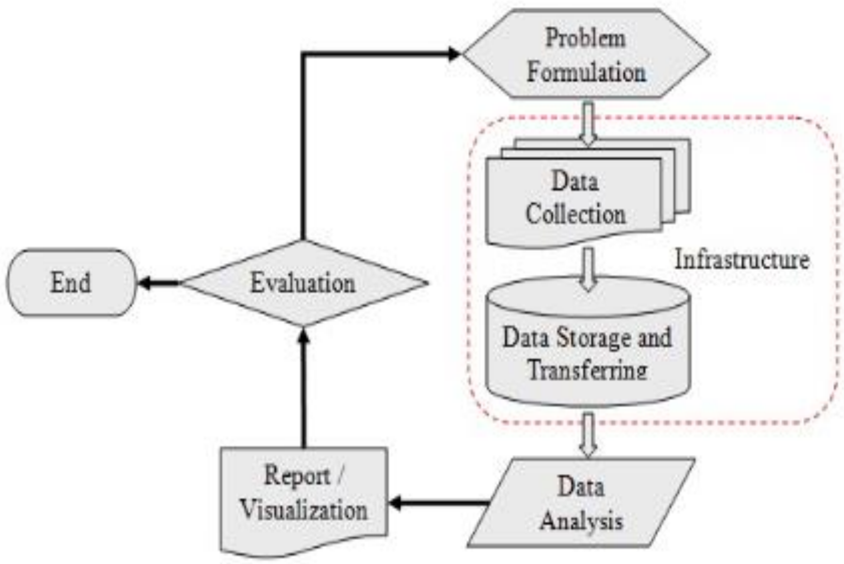


Figure 2: Workflow of Big Data Analytics (Hong *et al.*, 2019)

K-Means Clustering Algorithms

This algorithm happens to be one of the least difficult solo clustering algorithms that gears towards the well-known assembling issues (Oyelade *et al.,* 2010). Such technique pursues effortless and easy path classify known information index (dataset) via a certain cluster (*k* groups) of rooted a priority. The principal idea here focuses on distinguishing *k* with all the groups. These directions are to be placed in a shrewd track due to the different locations that brought such diverse result and it's therefore, better to ignore them. Next move is to consider each of the directions towards allocating generated information collection for closer partnership with the group. Since the new *k* centroids have been re-configured as bury focal point of the next clusters due to previous action, there is the need to couple similar information index that resembles the new one (Haidari, 2019). As a result of the group creation, it could be seen that the k focus' changes position bit-by-bit till no more changes are experienced.

Finally, the computational centre that limits target capacity is known as squared blunder job as expressed by Arage *et al.,* (2018) and given in the equation 1.

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{ci} \quad (||\mathrm{Xi} - \mathrm{Vj})^2 \qquad (1)$$

With, '||xi- vj||' as the Euclidean distance between xi and vj, while 'ci' represents information centres (data centres) in i^{th} cluster. Also given is 'c' as the number of group Centres

Let X = {x_1, x_2, x_3, … … ..., x_n} be the information and V = {v_1, v_2, …., vc} the centres.
1) Random selection of 'c' group focuses.
2) Calculate the separation between every data point and data centres.
3) Assign information/data to the focal group that are least of all the group focuses.
4) Re-compute the new focal group using equation 2 below:

$$vi = (\frac{1}{ci}) \sum_{j=1}^{ci} Xi \qquad (2)$$

5) Re-compute the distance between every data point and the newly acquired data centres.
6) If no data point is reassigned, then stop, else re-hash from stage 3 (Arage *et al.,* 2018).

Apache Hadoop/MapReduce

The most formidable open-source tool for big data analytics is Apache Hadoop and MapReduce (Avish *et al*., 2018). It comprises of Hadoop kernel, MapReduce, Hadoop distributed file system (HDFS) and apache hive. The component responsible for the processing of big datasets is the Map reduce programming model and exists based on the phenomenon of divide and conquer method. Hadoop is executed on master and slave nodes with master node using map step technique to distribute sub-problems to the slave nodes. The method allows the master node to put together in reduce step, the outputs of all the sub problems making Hadoop and MapReduce to form a strong big data problem solving software framework as described by Aritha, (2018) and Deshai *et al*., (2018).

Presto

The work of Saradevi *et al*., (2016) opined that R is an extension of a runtime language for managing distributed task where parallel execution and data distribution are added its functions. The runtime is responsible for managing the memory, schedule data partitioning as well as fault tolerance as the master is in control of the general accomplishment. It accomplishes the distribution tasks of the program across multiple slave processes as given by Raghav *et al*., (2019). While R, an array- based condition which gives a collaborative domain to evaluate data, likes translated contingent execution if, loops (for, while, repeat), and uses array managers written in C, C++ and FORTRAN for better execution.

Apache Spark

One single open-source tool that was built with processing speed considered in mind, is the Apache spark analytics. It was developed as an open-source tool in 2010 by UC Berkeleys AMPLab with the characteristic of accepting other applications of java, Scala and python. Furthermore, a part from MapReduce operations, it also accepts queries from SQL, data streaming & processing and Machine Learning (Armbruty *et al*., 2015). For additional functionality, spark rides on Hadoop distributed file system (HDFS) infrastructure for the enhancement of its functionality. There exist cluster manager and slave nodes and is applied at the execution point of the spark cluster where cluster manager distributes resources while the slave nodes process the data as tasks. This provides assistance when deploying the spark applications on Hadoop clusters. Figure 3 depicts a typical architecture of Apache Spark below:

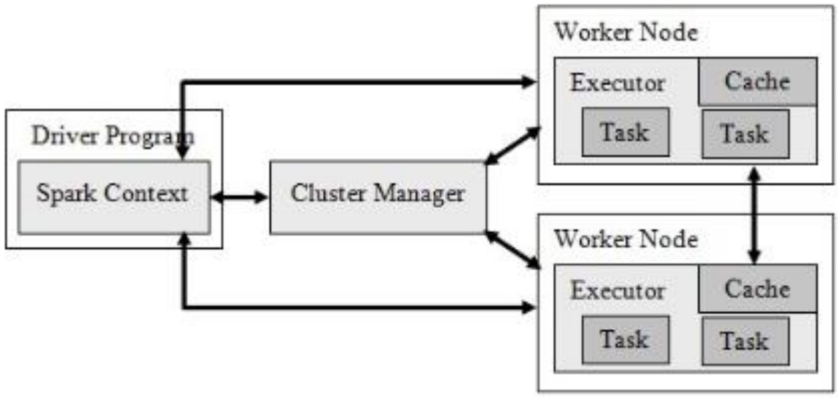


Fig. 3: A Typical Spark Architecture (Armbruty et al., 2015)

- Spark is a resilient distributed datasets (RDD) tool that includes stores data and deliver fault tolerance with no duplication as it supports powerful computing, improved speed and resource utilization.
- User is capable of running Java, R, Python, or Scala programming languages because of its higher-level libraries meant for state-of-the-art analytics. As a result of its standard libraries, productivity increases and eventually merge to produce complex workflows.
- Spark is capable of handling applications fast up to 100 times and process 10 times faster with Hadoop cluster on disk due to the reduced number of read or write operations.
- It runs on java virtual machine (JVM) environment but written using Scala programming language as it supports other applications such as python and R.

Storm

For huge data streaming process, there is no better distributed and fault tolerant real-time computing system other than Storm. In variance with the batch processing functionality of Hadoop, this platform is designed for real time processing only. The platform has indistinguishable cluster to that of Hadoop with simple mode of operations, scalable, aggressive performance and fault-tolerant. Unlike Hadoop cluster, Storm cluster is made up of two nodes; master and slave nodes, where both the master and worker nodes implement closely similar but different roles of nimbus and supervisory respectively. The nimbus takes charge of code distribution all over the storm cluster, schedule and assign job to slave nodes as well as observe the entire system.

Apache Drill

One of the big data interactive distributed systems is the Apache drill that has the flexibility of supporting different types of query languages, data formats and data sources. This platform has been designed to process nested data and to increase the processing capability of up to 10,000 servers or more to reach petabytes of data in seconds. Apache make use of Hadoop Distributed File System (HDFS) to store data and MapReduce for performing batch analysis.

Jaspersoft

Another big open-source data analytic platform tool is the Jaspersoft software package meant to produce reports directly from databases. Its scalability comprises of fast data visualization on well-known storage platforms, like Cassandra, MangoDB, Redis, etc. It has the capability of exploring big data easily without extracting, transforming and loading (ETL) data. Additionally, it has the ability of generating strong hypertext markup language (HTML) reports straight from big data store without ETL requirement and share it within or outside user's organization.

Splunk

Splunk as a platform known for intelligence and as a real time processing of data streams, is developed to explore machine generated big data from industries and businesses (Zhou, 2015).

Such data are gathered from cloud technologies and big data which in turn assist users search, monitor, and analyse the data generated by the system from web interface. These outcomes are intuitively displayed graphically, in form of reports or as generated alerts. Splunk has peculiarities as it differs from other stream processing tools in structured indexing unstructured, system created data, real-time search, results analysis report, etc. One of the keys focuses of Splunk is to produce metrics for applications, system problems diagnostics and provide infrastructures for information technology as well as support business intelligence operations.

METHODOLOGY

The selected open-source tools for big data analytics include the following, three primary layers of Hadoop, Presto, Spark and MapReduce. Furthermore, other software's were considered in this paper such as java JDK, Scala, winutils, Anaconda, pycharm python, K-mean HMR, and lastly Standard K-Means as found in Ketu, (2020). To gain a varied analysis, it has to consider several data sizes as follows 62MB, 1GB with a single node, 1GB with two nodes, 2GB with three nodes and monitored the performance in terms of the time taken for clustering as a requirement using K-Means algorithm.

K-MEANS Algorithm

In this research k-means (KM-HMR) algorithm was chosen in order to simplify the use of various volume of dataset which gives easy structure and arranged the dataset in cluster to determine the accuracy of the analysis. The Algorithm: refer to KM-HMR, a MapReduce execution of K-means which form clusters quicker than typical K-means clustering procedures. Big datasets were given as input, separated into portions and kept in the HDFS where the chosen K data points and objects given as primary cluster centers, updates the nucleus of all cluster by cluster accomplished many replications. The MapReduce receive series of files encompassing prime cluster centers as key value (k, v) pairs. In this stage, the gap amongst data entities and every cluster was calculated while considering the Euclidean space. Traditional K-means clustering algorithm was found to be suitable for application to small, moderate and not vey massive but structured datasets. Once the volume of the datasets is huge and are unstructured, processing and result generation of such takes significant period. The new KM-HMR accomplishes the aim of treating huge quantities of data in parallel with the MapReduce programming model as depicted in figure 4.

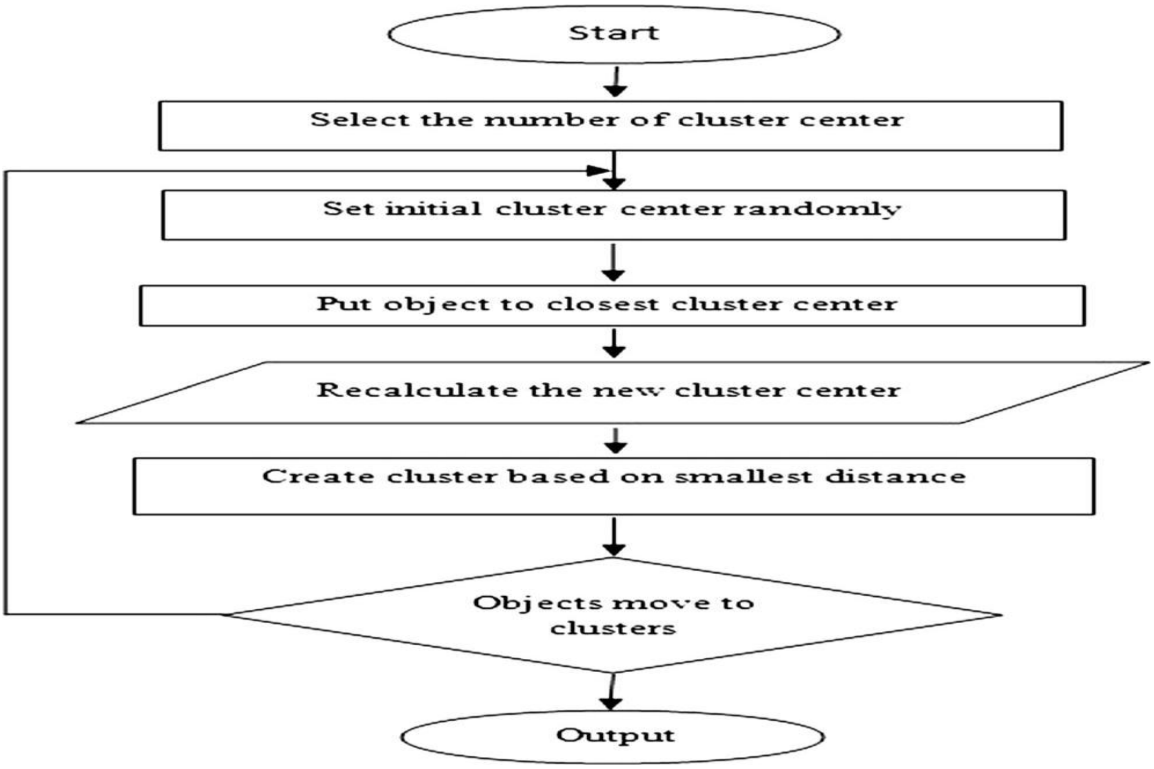


Figure 4. Flow chart of the Algorithm

Pseudo code for KM-HMR Algorithm

```
Input:
        O: {o1, o2, o3, …on}; set of objects/entities to be clustered
        K: K clusters (number of clusters)
        Mni: Maximum number of iterations
Output:
        Final output clusters
        KM-HMR (data)
        NI      0
                for each datapoint dϵD do
                        IC      select (K, D)
                input(D)
        write (IC)
                OC    IC
        while (true)
call to job.mapper( )
call to job.reducer( )
        NCV = read ()
                // repeat until convergence
if update ((NCV, OC) > 0)
        OC = NCV
                Else
                        update NCV to result
                NI++
        result = read ()
```

The objects that belong to the same cluster are sent to reduce phase. the reduce phase calculates the new cluster centroids for the next MapReduce job. The overall flow of KM-HMR. Cluster centroids produced at the end of an initial iteration are stored in an old cluster file and are tested for the appearance of new cluster centroids with each iteration. When new cluster centroid values are obtained, new cluster centroid values are updated in a new file and the number of iterations is increased by one. This process is repeated until no more changes in cluster centroid values are found, and this state is referred to as convergence. The final output clusters are stored in a result file.

Data Gathering

Data collection could be defined as a process where information is accurately gathered and measured on targeted metrics in a systematic fashion, which then enables answering pertinent enquiries and evaluates end results. It is then necessary that the process for conventional data collection is maintained as the spelled-out accuracy and succeeding decisions build on reasons expressed in findings through data are valid.

RESULT AND DISCUSSION

The comparative analysis of Apache Hadoop, Spark, and Presto considered three parameters vis-a-viz, scalability, latency and fault tolerance on a dataset that allow clustering using the K-Means algorithm. The following results for comparison are shown in the tables 1, 2 and 3. The study carefully observed all executions processes for different data sizes in order to get an accurate result. The analysis using the two algorithms with regard to clustering were repeated several times. The tools were also compared based on the various data sizes with regard to the response time during the processing i.e., retrieval and sending the data. The tables (1-3), show how the tools responded to the experiments for the three open-source tools based on some parameters and characteristics.

Hadoop

Table 1. Response Times queries of Hadoop

S/N	Datasets	Time(s)	Latency	Iterative	Total
1.	100GB	2.6	1.1	3.0	6.7
2.	150GB	3.0	1.4	3.1	7.5
3.	200GB	3.5	1.9	3.3	8.7
4.	300GB	4.0	2.3	3.5	9.8

Spark

Table 2. Response times queries of Spark

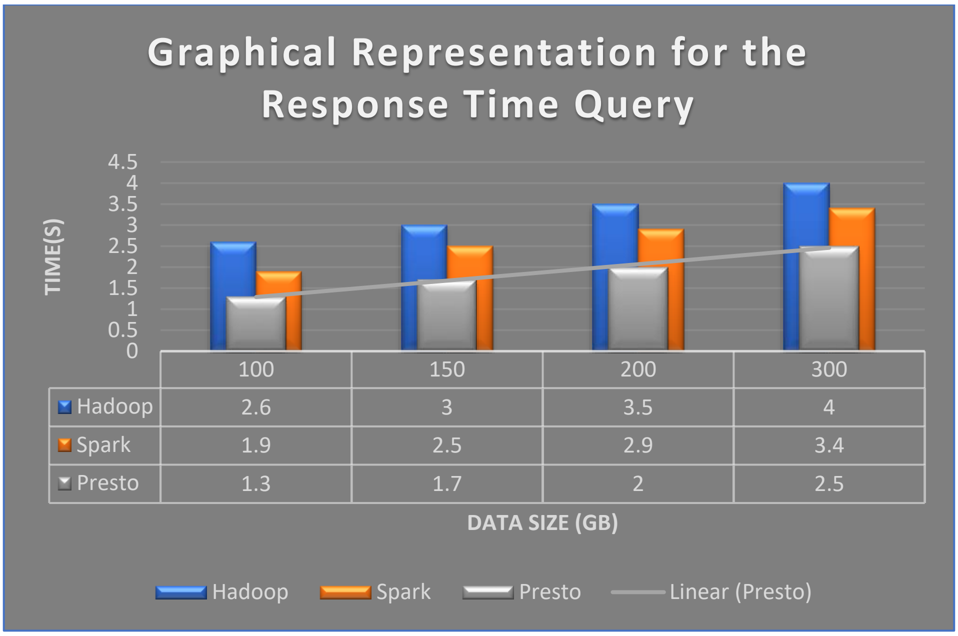
S/N	**Datasets**	**Time(s)**	**Latency**	**Iterative**	**Total**
1.	100GB	1.9	0.81	2.9	5.61
2.	150GB	2.5	1.05	3.0	6.55
3.	200GB	2.9	1.23	3.2	7.33
4.	300GB	3.4	1.65	3.4	8.45

Presto

Table 3. Response time queries of Presto

S/N	**Datasets**	**Time(s)**	**Latency**	**Iterative**	**Total**
1.	100GB	1.3	0.48	2.5	4.28
2.	150GB	1.7	0.71	2.7	5.11
3.	200GB	2.0	0.96	2.9	5.86
4.	300GB	2.5	1.09	3.1	6.69

Therefore, the results shown in three tables 1 – 3 above are interpreted presenting the query time responses of the three open-source tools and depicted in Figure 4. From the experiments, Hadoop's job processes request from users by repeatedly scanning the data with while performing other jobs. Such phenomenon gives Hadoop the advantage to allow more nodes to be added. Moreover, full scanning of the data becomes necessary if processing of the data sub-set is required and only then, does the strategy for the data access becomes a constrain. There exist other unique approaches to confirm unquestionable actions that could be speedily achieved. Many a times, unique data access procedure re-assures every query of various data access policies with Hadoop being enhanced for end-to-end requirements and only approve different data with no competence for update or otherwise.



	100	150	200	300
Hadoop	2.6	3	3.5	4
Spark	1.9	2.5	2.9	3.4
Presto	1.3	1.7	2	2.5

Figure 5: Graphical Representation of Response Time's Queries of Researcher

It is shown in Figure 5 that, Spark has the capability to query huge datasets and used 300GB of data packet to analysed. The result further shows the data for the experiment used 100GB to 300GB instances with 4 cores and 4 GB of RAM each. Then run probes to explore views of whole pages, pages with titles identical to titles partly similar and each of the queries were able to scan the entire input data. Even at 300 GB of data, queries on Spark took some seconds faster than an order of magnitude when compare with on-disk data.

However, Presto looks faster than Hadoop and Spark with regard to this result in figure 5. The same dataset was applied for presto from 100GB to 300GB and the analysis were carried out as with the other tools. One feature with presto that upgrade its performance is, the distributed SQL query engine that run interactive analytics queries against data source of all sizes. From Figure 4 and 5, there is clear indication that presto has the minimum query time and low latency as compared to Hadoop and spark.

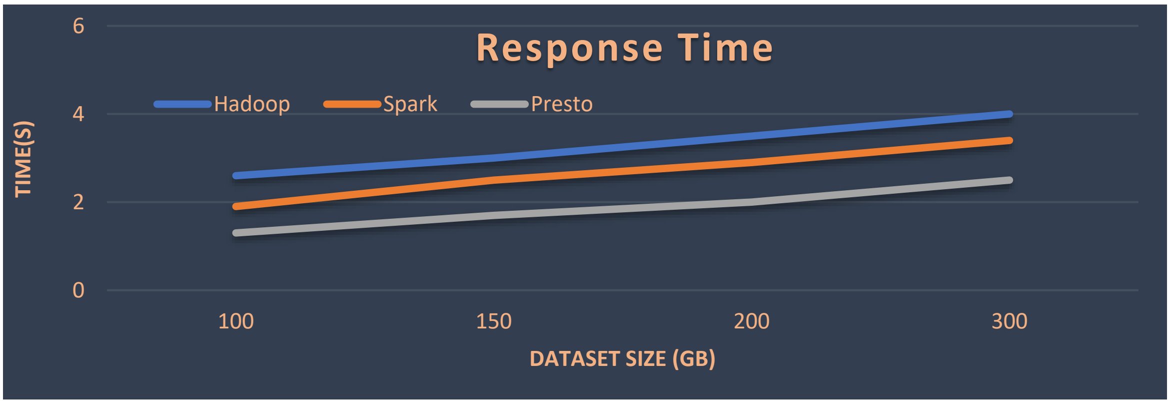


Figure 6. Response Time of the Tools of Researcher

Figure 6. clearly shows the difference between the three open-source tools in terms of latency, speed, processing time, and scalability. Considering the time, it takes to process dataset of different volume for each open source in the graph, which implies that a dataset of 100GB was processed in 2.6 seconds by Hadoop, 1.9 seconds by Spark, and 1.3 seconds by Presto. Therefore, the time it takes the tools to store and retrieve a volume of different dataset differ and obviously indicated that in terms of speed, latency, query response, and scalable. It is evident that presto showed the best execution time, while Spark is presented as the second best based on this work, thereby demonstrating the strength of presto's low latency with high speed among the tools.

PERFORMANCE EVALUATION

Using the evaluation standards, a comparison of the three open-source tools is shown in Fig. 6. The paper tried to evaluate the performances of the three open-source tools in order to determine which one of them is better in terms of fault tolerance, latency and scalability. These tools were assessed using the datasets given earlier as inputs.

The performance evaluation of the tools considered in this work is presented in figure 6. The graph shows that scalability/adaptability is viewed as the capacity to include more equipment (scale up or scale out) to improve the limit and execution of a framework. Virtual Clusters of Hadoop and Spark had got ninety-two percent, as well as hundred percent for Presto which implies that the framework has capacity to include nodes inside one cluster in each device with no overhead. Therefore, with regard to this study and due to the experiment carried out Hadoop and Spark open-source tools were found to be scalable as it attains a 100% scalability. In terms of fault tolerance, the work considers the probabilities of disappointment in the framework and give a high appraising in case of failures. This empowers the impartial examination between inconsistent frameworks with fault-tolerance and solid hardware that are could not adapt well to internal failure components. It is clear from the figure 6 that virtual cluster for Hadoop recorded a hundred percent (100%) fault-tolerant as the process reinstates itself the moment it crashes in the middle of its execution. Nevertheless, in Spark, this feature is absent and it commences the process from the very start due to rise in time complexity, but used spark streaming framework for stream processing of fault-tolerant to handle big data velocity. This led to Spark having eighty percent (80%), Presto got sixty-five percent (65%) because of Single Point Failure in Query Execution that could occur once any host quits the query execution.

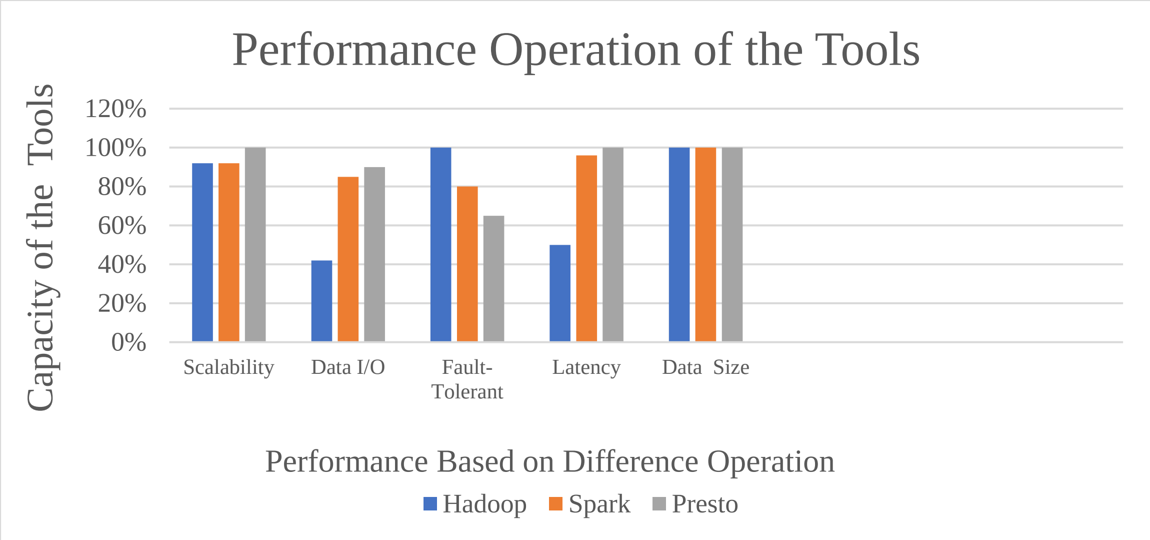


Figure 7. Comparative Ranking of Three Open-Source Tools

For Latency, the time it takes for dataset to be stored or retrieved from data warehouse was considered important and so the attention of how long it takes for application or user to save or recover source data from data warehouse console/dashboard was given priority. With regard to this research the latency of the open-source tools categorically shows that Hadoop response time for a data size of 100GB is lowest when compare with the other tools due to its self-reinstated process that the Hadoop possesses as strength. Therefore, Hadoop is discovered to have the highest latency, while Spark and Presto respectively.

To justifiably rank the open-source tools, a modified rating of such tools in accordance with the selection criteria is necessary and is presented in figure 7. While a number of other tools are available, little literatures to extensively evaluate them exist. In conformity with the previous statement, the choice of machine learning was for user specific application. Due to speed and scalability of numerous datasets, Machine Library (MLlib) is a good option when selecting algorithms. Furthermore, the design for real time streaming, speed and scalability, presto is found to go well with MLlib.

CONCLUSION

The current happening today with unprecedented access to huge data requires synergy of tools necessary to process, analyse and store such volume of data. It is difficult therefore for such massive data to be properly analysed without a careful and pragmatic choice of the correct tools to execute. The open-source tools are to facilitate the importance of learning tasks many of which are gaining ground to becoming more resourceful. This paper therefore, discussed three of the open-source data analytics tools vis-à-vis, Hadoop/MapReduce, Spark and Presto ecosystems that provides conditions that make machine learning as a tool for the analytic domain. The work adopted K-means as framework for clustering because of its importance to data analytics. The alternative to these instruments is mostly linked to purpose for which they being utilised for users. Domains like healthcare frequently generate varying datasets that call for blending of batch and streaming processing, where Spark was found to be the right choice. In this study, the Hadoop/ MapReduce, offers support for iterative tasks, while presto was adjudged to be fast in data processing and offers the best with regards to real time stream processing. The choice of these open-source tools will surely provide solutions to the large datasets.

REFERENCES

Anitha Patil. (2018). Securing MapReduce programming paradigm in Hadoop, cloud and big data eco-system. Journal of Theoretical and Applied Information Technology, 96(3); 756-766.

Arage, C.S., Gaikwad M.P., Tadasare, R., & Bhutra, R. (2018). Analyse Big Data Electronic HealthRecords Database using Hadoop Cluster. International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue 03., pp448-449.

Armbrusty, M., Reynold S. X., Cheng L., Yin H., Davies L., Joseph K., Bradleyy, X., Mengy, T. K., Michael J. F., & Ali G., Matei, Z. (2015). Spark SQL: Relational Data Processing in Spark. International conference on management of data. (15); 1383–1394

DOI: https://doi.org/10.1145/2723372.2742797.

Avanish Singh., P. Gouthaman, Shivankit Bagla., & Abhishek Dey. (2018). Comparative Study of Hadoop Over Containers and Hadoop Over Virtual Machine International Journal ofApplied Engineering Research, 13(6); 4373-4378.DOI:10.1109/IGARSS.2014.6946698.

Chenga, E., Liya Maa., Adam Blaissea., Erik Blaschb., Carolyn Sheaffb., Genshe Chenc., Jie Wua., & Haibin Linga. (2015). Efficient Feature Extraction from Wide Area Motion Imagery by MapReduce in Hadoop, *ACM*, 1-9.

Deshai, N., Venkataramana, S., & Varma, G. P. S. (2018) Big Data Hadoop Map Reduce Job Scheduling. A National level Conference on Current Trends of Information Technology, 6(1); 103-114.

Funde, S., Karale, S., & Bharate L. (2018). Survey of Big Data Security. International Journal of Current Engineering and Scientific Research, 5(2); 27-29.

Haidari, S., Alborzi, M., Radfar, R., Afsharkazemi, M. A., & Ghatari, A. R. (2019). Big Data Clustering with Varied Density Based on MapReduce. Journal of Big Data 6(77); 1-16. https://doi.org/10.1186/s40537-019-0236-x

Hong L., Mengqi L., Wang R., Peixin Lu, Wei L., & Lu L. (2018). Big Data in Health Care: Applications and Challenges. Data and Information Management, 1-23. DOI: https://doi.org/10.2478/dim-2018-0014

Jin, X., Wah, B. W., Cheng, X., and Wang, Y. (2015). Significance and challenges of big data research,Big Data Research, 2(2), pp.59-64.

Kakhani, M. K., Kakhani, S. and Biradar, S. R. (2015). Research Issues in Big Data Analytics, International Journal of Application, Innovation in Engineering & Management, 2(8), pp.228-232.

Ketu, S., P. K. Mishra., & S. Agarwal. (2020). Performance Analysis of Distributed Computing Frameworks for Big Data Analytics: Hadoop Vs Spark, 24(2); 669–686. DOI: 10.13053/CyS-24-2-3401.

Nikita Gureev. (2018). Hive, Spark, Presto for Interactive Queries on Big Data. Degree Project in Information and Communication Technology, second cycle, 30 credits Stockholm, Sweden.

Oyelade, O. J., Oladipupo, O. O., & Obagbuwa, I. C. (2010) Application of k-means clustering algorithm for prediction of students' academic performance. International Journal of Computer Science and Information Security, (7)1; 292-295.

Raghav, S., Martin, T., Dain, S., David, P., Wenlei, X., Yutian, S., Nezih, Y., Hoazhun, J., Eric, H.,Nileema, S., & Christopher, B. (2019). Presto: SQL on Everything. IEEE International Conference on Data Engineering, 1, 1802-1813 DOI : 10.1109/ICDE.2019.00196

Saraladevi, B., Pazhaniraja, N., Paul, P.V., Basha, M.S.S., Dhavachelvan, P., 2015. Big Data and Hadoop-A Study in Security Perspective. Procedia Computer Sci. 50:(596–601).

Zhou, W, X. (2015). Performance Comparison of Hive, Impala and Spark SQL. International Conference on Intelligent Human-Machine Systems and Cybernetics - IEEE Computer Society, 1:(418-423).