International Journal on Emerging Technologies 11(2): 198-204(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

Comparative Performance Analysis of Apache Spark and Map reduce using K-Means

E. Laxmi Lydia[1], R.M. Vidhyavathi[2], Irina Pustokhina[3] and Denis Alexandrovich Pustokhin[4]

[1]Professor, Department of Computer Science and Engineering,
Vignan's Institute of Information Technology (A), Visakhapatnam (Andhra Pradesh), India.
[2]Assistant Professor, Department of Bioinformatics, Alagappa University, India.
[3]Associate Professor, Department of Entrepreneurship and Logistics,
Plekhanov Russian University of Economics, Moscow, Russia.
[4]State University of Management, Moscow, Russian Federation, Russia.

(Corresponding author: E. Laxmi Lydia)
(Received 04 December 2019, Revised 30 January 2020, Accepted 05 February 2020)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: All around the globe, computer science grabbed its interest inBig Data that has developed extremely high with its continuous raise of data generation in social media and the aim for industrial and borstal Hercules institutions to facilitate additionalinvestigation of their knowledge. This paper provides a deliberate correlation regarding two Apache frameworks such as Apache HadoopMapReduce and Apache Spark (advanced). The two frameworks present a design structure to partition the tremendous data to appropriate information. Although these preferences rely on the BigData objective, individual achievement conflict from collective perspective utilization and its implementation purpose. Such an idea alternates the two commendable analyses for the variation and selection of BigDatain dynamic possibilities. In this paper, we contradict the mentioned two designed framework structures together and offereffectivenessin evaluating throughhandling anapproved machine learning approaches for data assembling using K-Means. From the beginning, the observationsof this workingpaper determinethe relative performance measuresand approximating specifications for MapReducesuch asvelocity, throughput, and dynamism consumption of energy.

Keywords: Big Data, ApacheHadoop, Apache Spark, Apache Mahout, Machine Learning, HDFS, MapReduce, MLib, K- Means.

I. INTRODUCTION

Resilient distributed Datasetsare imposed from memory enclosed by many different questions beyond any necessity of replication.Significant to the renovating of the misplaced information was setback with the appropriate origin. Respective RDD will recognize and collect the performed work from various data sets to regenerate itself (such as a map and sometimes join). RDDs authorize Spark to overwhelm the actual design by involving various passes over analytics [16] up to 100x. These RDDs were used to reinforce an extensive combination of continual computations, along with data mining perceptive and a completely useful SQL generator Shark [8]. Fig. 1 testifies the Hadoop Ecosystem System.

MapReduce is a distributed manipulating approach that performs designed models using Java. It is designed depending upon two primary jobs to be performed such as map (also known as mapper) and reduce (also known as reducer). Mapper schedules the data accordingly and support a similar process for scheduling of other data, where unique data segments are partitioned into tuples represented as pairs of key-value. Likewise, reducer scripts the inputs from a mapper and accompanies data tuples (key-value pairs) within a limited provision of key-value pairs of data. As per the series of mapper the continuous job process is maintained through reducer. The evident supportiverole of MapReduce work on entire data but hard to proportion data composition over abundant nodes. Subsequent primitives for the data preparation through Map Reduce acknowledged as mappers and reducers.

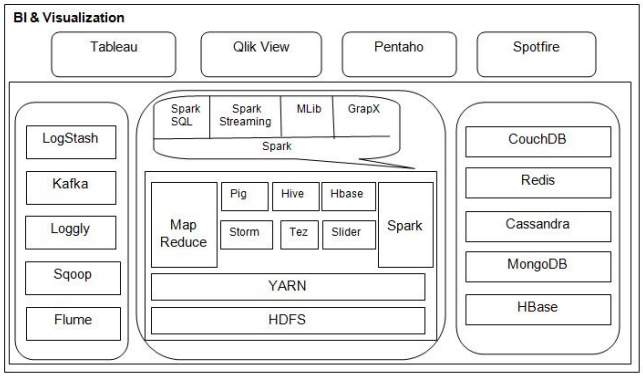

Fig. 1. Hadoop Ecosystem System.

Within the delay interval of time, the data is processed using mappers and reducers. Composing the architectural design structure for Map Reduce, the balanced application was proposed to work over hundreds of, thousands of, and many more machines (nodes) that form a cluster [14]. This can be maintained and improved through a configuration setting of nodes. Programmers have gained great profit using the Map Reduce model, as it is the basic versatility is that attracted them [10]. The process provides an equal number of cluster outcomes, for all the documents of mappers and reducers [11, 12]. The major two versions of hadoop are 2.x and 3.x.The fault tolerance in Hadoop2.x handlesthrough replication and Hadoop3.x through erasure coding,to balance the dataHadoop2.x usesHDFS balancer, Hadoop3.x uses HDFS disk balancer. Hadoop2.x uses tenthousand nodes for processing, Hadoop3.x uses more than ten thousand nodes for processing.

Following are the generalized steps for the MapReduce mechanism:

–Most essentially, the residence of the data is transferred from the MapReduce to the computers depending on the prototype.

–It is designed to function under three different phases. They are map phase, shuffle phase, and reduce phase.

–The incoming input data is processed usingthe mapper phase in file or directory pattern. This data is stored and managed in HDFS. It is processed line by line by generating many smaller chunks of data using the mapper function.

–The output from the mapper phase is considered in the reducer phase with the mixture of the shuffle phase. A new collection of results are obtained after the reducer phase and stored in HDFS.

–Clusters are maintained by the servers and allocated with tasks from applicable jobs from the MapReduce model.

–The designed model handles the overall functioning of the work progress of data by granting tasks, verifying, replacing within the cluster nodes.

–Network traffic is reduced by performing the data computations on local disks of nodes.

–The actual outputs of every task from the reducer are assembled and transmitted to the Hadoop server.

The major functional performance of MapReduceis based on <key, value>pair. Every single job is executed by setting<key, value> pairs and presentsa new output set of <key, value>pairs of the activities, probably of specific varieties.

The framework is designed to maintain in a serialized manner with the key and the value classes. Therefore, it is easy to perform the writable interface. The simplified sorting of key classes carries out a writable-comparable interface with the designed framework.The key-value pairs are written in the form of <k1,v1> from input and <k2,v2> from mapper function to reducer function and <k3, v3> from reducer function to output.

	Input data	Output data
Mapper function	<k1, v1>	list <k2, v2>
Reduce function	<k2, list (v2)>	list <k3, v3>

Limitations of MapReduce and Spark: Mapreduce support only batch processing files and not suitable for Real-time data processing, moreover small files are not as block size in Hadoop is developed to store large files consumes more time in processing, Aapche spark lacks file managing system, iterative processing as the data needs immense rearranging of data, very few algorithms and has manual optimization

However, this paper considers Data Set that assimilates sensor data of 1240 MB size saved all the time and assimilates right ascension and declension that valuates of the distinctive records. A variety of sample data records are enclosed and originated as follows:

– Data format with Date
– Data format with Name of the Device
– Data format with the Device ID
– Data format with Device Status
– Data format with Latitude in decimal
– Data format with Longitude model record

The main objective of this paper is to provides the Spark growth in business with the confirmed and authorized structure for a comprehensive statistical application that works with Big Data processing.

II. LITERATURE REVIEW

Big data is an environment where large volumes of data have been analyzed, consistently extort information using uniquely designed open source applicational software solutions like Apache Hadoop [1]. It handles and scrutinizes Big data. In the year 2005, a software developer named Doug cutting, who was an employee of a yahoo search engine, got a chance to extend web crawler. The designed framework was noticeable with two internal foundations of processing and storage titled as Hadoop Distributed File System [2] and the MapReduce [3]. The scheme to organize all the files is restored by Google. This system [4] is flexible, dynamic, beneficial, and replicative with the established quantity of data at any number of distinct nodes that form a cluster known as GFS. A refinement to the Google system is a master-slave architecture that consists of master nodes and slave nodes, represented as Namenode and Datanode. These nodes consist of original statistical information. For the backup of the data, the replication factor was assigned to the system based on the necessity of the client; otherwise, it is by default represented as three. Secondly, the most immediate and authorized part of Hadoop is MapReduce. This is structured to process the imitated data complementary in a distributed manner using two major functions, map, and reduce. The functional performance of map function undergoes partitioning of data based on the dataset to Map is the stage which is actualized to conveyed segments of a dataset to various other “mappers” that manage data in parallel to permit the opportunity to the estimation of Big Data kernel [18, 19].

This will concede to the sorting and shuffling of the data, that comes from the mappers. Further, it leads the data to reduce the phase. The reducer will aggregate the data and represent them to the result that overcomes the elemental concerned description.

Estimating the time and number of resources utilizied by minimizing big data systemsthat has deficiency while progressing. These can be scheduled using Apache Spark and Hadoop processing [5].

Despite the fact, the dynamic computational [17] nature of the Big Data, a model has been designed with proper production and substantial dispensation of Apache Spark, that presents a straightforward approach to reduce the code lines undertaking and

providing improved achievement of many circumstances relevant to BigData. Spark yields a possible choice to MapReduce, inclusive of inquisition SQL that progress with shark and machine learning repositories known as MLib. When compared to the effectiveness and work progress spark is not remarkably identical to MapReduce. Nevertheless, in the meanwhile, it is subservient on the compulsion of parallelism, the description of subjective concerns in locating and the existence of assets. Apache Spark initiated as an exploratory prolongs at UC Berkeley in the AMPLab and this held the objective to describe the programming pattern that supports a much more voluminous grade of operations in particular to MapReduce [6], during the time managing its mechanical fault tolerance. Spark [9] suggests a rumination termed as Resilient distributed Datasets [7] to sustain these demands productively.

A scalable spark approach to cluster data using parallelism (MapReduce algorithms) with efficient big data processing [20]. Density based soft clustering was Implemented Clustering algorithm for spark computational model. Big Data classification and Deep data characterization was proposed based on the Apache Spark and Deep Learning.

A. Methodology

Distributed Cluster Computing using Apache Spark: Large management corporations analyze their voluminous data sets with broad utilization of Hadoop. Hadoop is designed for an unambiguous and uncomplicated programming model referred to as MapReduce, and it authorizes a total estimated provision that is resourceful, resilient, fault-tolerant and profitable. This leads us to handle and assure high-speed comprehensive datasets and also waiting time of the program and generated queries. Apache Software Foundation conferred Spark to advance the speed up of Hadoopdata processing enumerated programming mechanism. A prevailing condemnation against Spark is not an accommodate translation of Hadoop and also not, mostly, susceptible to Hadoopupon the determinant that it has its specific private cluster management. Hadoop framework is analyzed as the only address to carryout Spark. It exploits Hadoop as a factor of double manner - one is a repository and the other is the formulation. As spark considers it retain cluster managementestimation, it handles Hadoop for arsenal justification as it continues.

Apache Spark is uniquely instantaneous cluster management exploration, predetermined for abruptcomputation. It is contingent upon HadoopMapReduce and stretchesthe MapReduce design toyield its effectiveness and exploit it for higher categorization of forecasting, that integrates collective catechism and surgemanipulation. The fundamental focal point of spark relies on main memory cluster management that frames the qualifying expedite of an application. Distant numerous workloads that carry batch processing applications, repetitive mechanisms,streaming, and correlative queries are assured to examine. Other than the processing workloads, it diminishes the management of maintaining disconnected equipment.

Progression of ApacheSpark: MateiZaharia designs Spark in the connectivity of stretching out the Hadoop's components in 2009 located in UC Berkeley's AMPLab. The software has been available open-source by the grant provided by BSD in 2010, in 2013, it was accustomed to the Apache programming foundation, and in 2014 apache foundations were reached to the dominant level through Apache spark.

Major Characteristics features of Apache Spark:

Versatility: Spark is easy to write large applicational operations using different programming languages such as Scala, Java, Python. This assists application designers to discover and operate their applications on inherentlyrecognizable programming terminology and make it frameeasy parallel apps. Above 80 high-level manipulators were collected and specified as built-in functions. This can be implemented convertible to interrogate data inside the shell excessively. Representation of Python API for Word Count in Apache Spark.

```
datafile = spark.textFile("hdfs://....")
datafile.flatMap(lambda line: line.split()).map(lambda word: (word,1)). reduceByKey(lambda x,y: x+y)
```

Associations of data cascading, complex analytics, and SQL: In supplement of transparent operations such as "map" and "reduce", Spark encourages queries generated from SQL, data flow, and different composite analytics comparative to machine learning and graph design out-of-the-box. On top of that, the end-user can associate all these effectiveness coherently in anoriginal workflow system.

Speed: One of the most prominent features of spark applications is its velocity 100x faster (in memory) and 10x quick even when consecutively runs on disk. Spark is potential with desired features to perform read/write minimum operations on the disc that shows factors of maintaining time consumption of processing. Data is accumulated all the intermediate processing of input info in memory. Spark's idea on Resilient Distributed Dataset (RDD) is handled, which grants memory precisely regarding the storage of data. The carry-through of data to its disc is required.

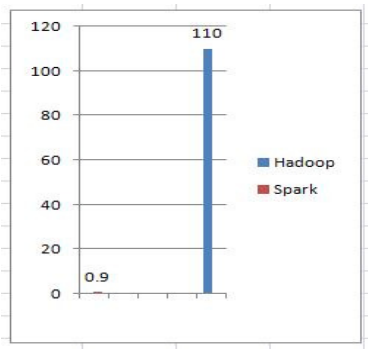

Fig. 2. Running time in seconds forHadoop and Spark.

Runs Everywhere: Spark manages and performs its operations on Hadoop, Cloud, and Standalone or with Mesos. The differing sources from different users that use spark include Cassandra, HDFS, HBase.

Acceptance of Spark in Hadoop: The subsequent built-in process of spark in Hadoop is represented in the following diagram in three different ways:

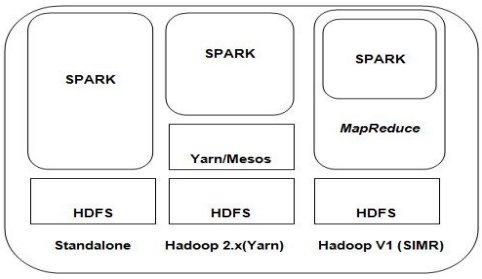

Fig. 3. Built-in process of spark in Hadoop.

Following are the three procedures of spark for its formation are made clear below:

– Standalone- The major functionality employed from the spark is independent and conqueredits place from the Hadoop distributed file system. and its space capacity is designatedconsidering HDFS, specifically. This contributes jobs on clusters [15] to run next to each other using Spark and MapReduce.

– Hadoop Yarn- Hadoop Yarn categorizes, spark to operate on Yarn beyond any preinstallation process or origin access recommended. It supports to organize spark into a Hadoop environment. It permitsalternative segments to function on top of the stack.

– Spark in MapReduce (SIMR)- Spark jobs are initiated to standalone deployment using MapReduce. Service through SIMR can give direct access to the user beyond any administrative access.

Spark Components: The various Spark components are described in the following Fig. 4.

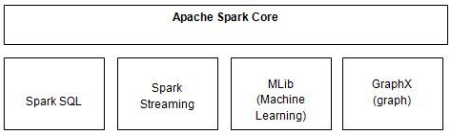

Fig. 4. Various Spark Components.

Apache Spark core- The basic functionality to perform approximate execution for spark platform is built upon Apache spark core. This also determines the computations using In-Memory and external storage systems with applicable datasets.

Apache Spark SQL- The preeminent component that resides above to apache spark core is apache spark SQL which includes aSchemaRDD by initiating new data abstraction. This determines the effective maintenance of data structures such as structured and semi-structured data.

Apache Spark Streaming- Data analytical streaming got more influenced by apache-spark core through spark streaming with the fastest scheduling potentiality. Data is being devoured using mini-batches and implements the data processing of mini-batchesthrough RDD(Resilient Distributed Datasets).

Apache Spark MLlib- MLlib(Machine Learning Library) components determines the importance of distributed memory-based spark architecture in distributed machine learning establishment.Thiswas measured by benchmarks for the opposition of Alternating Least Squares (ALS). Compared to Hadoop apache mahout Spark MLlib is nine times is faster.

Apache Spark GraphX- Spark with its graphical interface determines distributed graph-processing design using the GraphX component. A pregel abstraction API for graphical computations were implemented through user-defined graphs. Moreover, it determines revamped runtime for this abstraction.

Characteristic essential elements between map reduce and Spark: Most extensive architectures are designed based on storage capacity and storage of data. ApacheSpark collects the data in-memory at the same time Hadoop collects the data on disk. Hadoop exploits replication to manage fault tolerance directed through spark. On the other side spark handles discrete data stockpiling prototype, with reduced fault tolerance from a sharp approach, the network I/O is minimized.

– Use of Machine Learning Iterative Algorithms

– Associative Data Mining and Data Processing techniques.

– Spark performs 100x speedy than Hive witha fully Apache Hive-Compatible data ware housing system.

– For the Stream processing alerts, accumulation, analysis live streams were identified through Fraud detection and prepared through Log processing.

– Data is retrieved and blended from diversified sources using sensor data processing. The in-memory dataset got accessible with a straight forward process.

Performance assessment in Hadoop and Apache Spark: The performance of Hadoop is estimated by the removal of many reads and writes to discussing the MapReduce framework. Whereas the performance of sparkis estimated by the execution of batch jobs which is 10 to 100 times quicker than the Hadoop. Hadoop provides MapReduce with two extensive distributionon performing tasks using mapper and reducer and secures the data residing in the disc. Jobs that bounce back in the MapReduce framework with repetitive occurrences in processes doesnot affect the memory storage of Hadoop batch processing.

b) Though the information is recommended with suggestions of Hadoop spark Resilient Distributed Datasets provides, afford memory storage and defend it to the encircled computations and are mandatory. This won't prescribe the sorting of integratedinterferences that will keep the procedure back off. Thus, the approximate task implementations in spark machines aremore abrupt than HadoopMapReduce with the memory applicational operations.

Effective management in Hadoop Map Reduce and Spark: Effective management is natural for data processing and associates utilized streamline infrastructure. Currently, Hadoop spark is easy to understand and implement streaming, batch processing and machine learning approaches all in relatedclusters.

HadoopMapReduce helps to create records that lead to the development of huge segments of the extended Hadoop applications to explore answers for reported queries and achieve key measurements consecutively. It also works with an alternative framework that handles stream processing. Subsequently, the combinations will supervise all computational models.

The complexities are distributed with probable stream and assembled machinesto actualize to solve the progress, distribution, and aid of the Hadoop spark applications. Whereas spark has imaginable control over huge varieties of workloads. Suppose, different workloads have a similar process, spark provides a secure communication applying restricted applicable Map Reduce.

Stream processing in the real-time mechanism at Spark and Map Reduce: In real-time Hadoop uses batch processing as a straightforward process to

execute data whereas spark uses stream processing for reasonable progress of data.

Among high software operations, spark streaming contributes data with analytical solutions and graph processing. Apache spark connects the data to every distant element in graph processing. Apache spark produces cooperative usage of built-in libraries in machine learning code, which helps Hadoop cluster data to be organized accurately.

Storage (caching) in Spark and Map Reduce: Spark maintains and ensures the latency computations by storing the intermediate outcomes through appropriate shared jobs over its memory. Altogether, MapReduce shares its jobs through disc arrangement. Hadoop Spark is deliberately resolution by bringing immense advisability in comparable composite Hadoop Map Reduce pipelines.

Utilization of Spark and Map Reduce: Spark code is written in a compactable way than the Hadoop Map Reduce code. An example of spark MapReduce shown in the below image isthe word count program written in spark and Hadoop Map Reduce code. By this Hadoop Map Reduce, it is identified that code is more lengthy and verbose.

Rationale for the selection of Spark: Spark exploits the advantage of RDD that acknowledges users to stockpile data on memory and persist it by approving its requirements. This approves an excessive increment in batch processing job assessment (more than 100 times that of Map Reduce).

In terms of cache data, Spark alsopermits us toaccumulate the data in memory, which shows productive iterative algorithms, for instance, employed applicable machine learning parts. Non-cyclic data rely on customary Map Reduce and DAG motors applications for problem frequent occurrences. These applications with specified problematic jobs require continuous execution, analyze data through a distributed file system of Hadoop, and consist of proper storage. Every executable evolution of data brought in with critical cost stacking and duplicated back to the storage.

Data is transmitted in chunks; sparks obligates such considerable data information using stream processing and regulates it. Online machine learning handles the processes and promotes an imperative case for perpetual observations in necessary business organizations and various corporations.

MapReduce is precisely not extravagant for applications that contribute to the distribution of low-latency data implementing multi-pass over multiple parallel transactions. These specified Map Reduce applications are altoge thertypical in the check-up, and build:

– Algorithms that enforce Iterative models, besides, to plenty of machine learning models and graph models related to Pagerank.

– Loading data traversing through cluster into RAM and challenge it over again with interactive data mining.

– Managing of Streaming applications with combinative specific over time.

Applicational Performance of Machine Learning and K-Means Approach:

Presentation to machine learning: Advanced new patterns and information from different sources lead to generating precise codes using artificial intelligence that takes over all machine-learning approaches by developers. Machine learning permits reorganization of consistent data and formulates it for analysis, implements selected model, flourish model to analytical model and train it to run to know the accurate scores of the data [14].

Composition of the K-means approach for clustering data: Clustering of data can be achieved using different approaches; on the most prominent approach for effective data, clustering is K-means with the non-hierarchical approach. The data is been collected through the client to cluster for proper utilization and framing of identical groups. The composition of the K-means approach concentrates on centroids where centroid attracts the closest and similar data from the dataset.The procedure for the grouping is estimated by reducing the overall squared partitions of the input data and the assigned centroids. There are contrasting ways to preferential centroid, nevertheless, an arbitrary allotment is promoted as a rule. Following is the procedure [14] to run k-means :

– The user initially prefers to choose ‘n’ number of clusters and defines the system to randomly choose centroids ‘k’, where the number of clusters and centroids are equal.

– Every individual data element from the prescribed dataset isaccredited to nearby centroids. This process is defined as a Cluster assignment.

– The computational process to assign data points to clusters is to find the average of alldata points of the centroid and represented as a new value to the appropriate centroid. This process is defined asthe Centroid movement.

– Determine and estimate the sum of the square of the distance from a particular centroid and change its value based on the obtained value. Continue to process 2 and 3 stepsprior to threshold values no less than or equal to 0.01 as a choice when the number of iterations that ranges to maximum iterations defined to get contended.

Correlation between Apache Spark and Hadoop Map Reduce: With a definite term objective to show up aselection about the feasibleresemblance of apache spark and Hadoop map-reduce. With these designed frameworks, datasets are employed to implement a k-means approach for clustering.

III. RESULT AND DISCUSSION

Accomplishment of performance analysis and description: Considered sensor dataset has been implemented k-means in Apache Spark and Hadoop Map Reduce. Table 1 and 2 are the outcomescompared obtained from the k-means approach results. To improvethediscrete analysis this paper worked on, single node consisting of 64MB, 1240 MB and two nodes of 1240 M and observed the performance with respect to clustering time by means of our concerns through K-Means approach. Following are the considered machine configurations for evaluation:

– The memory of 4GB RAM

– Operating System consisting of Linux Ubuntu

– Hard Drive with 500 GB.

The outcomes obtained distinctly exhibit that the operations of spark resulting in vigorously more advanced with continuous time. Every dataset benchmarks the outcomes that degrade the time with acceptable thrice when differed by Map-reduce. Although the outcome has minor inconstancy on account of the peculiar way of the K-Means method and prevents the significance of evaluation to a broad standard.

Table 1: Obtained time for Spark (MLib) for distinct dataset size using K-means.

Dataset size	nodes	Time(s)
62 MB	1	18
1240MB	1	149
1240 MB	2	85

Table 2: Obtained time for MapReduce (Mahout) for distinct dataset size using K-means.

Dataset size	nodes	Time(s)
62 MB	1	44
1240MB	1	291
1240 MB	2	163

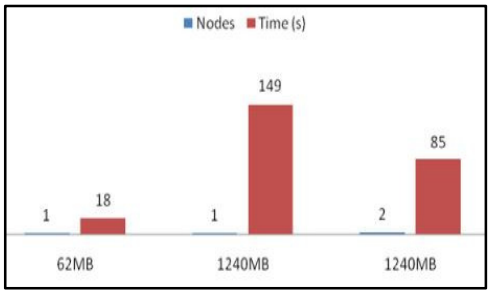

Fig. 5. Obtained time results for Spark (MLib) for distinct dataset size.

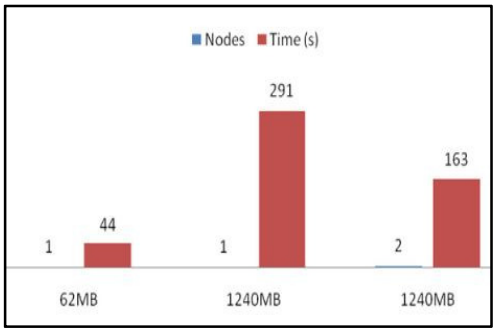

Fig. 6. Obtained time results for MapReduce (Mahout)for distinct dataset size.

Figs. 5 and 6 present the performance of graphical analysis, Fig. 5 illustrates the fastness ofa spark than map-reduce dealing with the datasets, nodes, and time. The emerging outcomes obtained through apache-spark analysis are a very powerful challenger to Hadoop Map Reduce and deliver transformation by considering in-memory processing.

IV. CONCLUSION

This paper administers the design and works analysis of apache spark and MapReduce by permitting additional specifications of the k-means algorithm. K-means minimizes the squared error function which enhances the throughout analytical process of both frameworks, the retrieved outcomes of the determination prove that Spark is a remarkablyreliable contender and choose to understand the advantage by implementing in-memory processing. By the observations of Spark's competency to observe cluster manipulations, streaming, and machine learning. A glance at Spark shows its growth in business with the confirmed and authorized structure for a comprehensive statistical application that works with Big Data processing.

V. FUTURE SCOPE

Despite various machine learning algorithms that work on Mahout are straightway implementations of MapReduce. Spark's persistent renovation and accumulation end-user source have managed mahout to support spark for their source structure restoration of Map Reduce for their outlook applications. Spark contributes in-memory handling of data to progress the processing speed. This is one of the diversified occurrences point to spark is justifying out to gain control over Map-reduce.

ACKNOWLEDGEMENT

Dr. RM. Vidhyavathi sincerely acknowledge the financial support of DST-FIST[SR/FST/LSI-667/2016(C)], DST-PURSE[SR/PURSE Phase 2/38(G)] and MHRD-RUSA 2.0[F.24/51/2014-U, Policy (TNMulti-Gen), Dept. of Edn. Govt. of India] for the infrastructure facilities available in the Department of Bioinformatics, Alagappa University.

Conflict of Interest. Nil.

REFERENCES

[1]. Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The hadoop distributed file system. In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)* (pp. 1-10).

[2]. Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified data processing on large clusters, *In OSDI'04: Sixth Symposium on operating system design and implementation,* 137-149.

[3]. Ghemawat, S., Gobioff, H., & Leung, S. T. (2003). The Google File System, *In 19th symposium on Operating Systems Principles,* 29-43.

[4]. HortonWorks documentation 2014 http://docs.hortonworks.com/HDPDocuments/HDP1/HDP-1.2.4/bk-getting-started-guide/content/ch_hdp1_getting_started_chp2_1.html

[5]. Lee, J., Kim, B., & Chung, J. M. (2019). Time Estimation and Resource Minimization Scheme for Apache Spark and hadoop Big Data Systems with Failures. *IEEE Acess*, 7, 9658-9666.

[6]. Zaharia, M., Chowdhury,M.,Das,T., Dave, A., Ma, J., McCauley, M., Franklin, S.,Shenker, S., & Stoica, I. (2011). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. *Technical Report UCB/EECS-2011-82, EECS Department, University of California, Berkeley,* 1-14.

[7]. Xin, R., Rosen, J., Zaharia, M., Franklin, M. J., Shenker, S., & Stoica, I. (2013). Shark: SQL and Rich Analytics at Scale, *SIGMOD,* 1-12.

[8]. Singh, A., Mittal, M., & Kapoor, N. (2018). Data Processing Framework using Apache and Spark Technologies in Big Data, *In Book: Big Data Processing using Spark in Cloud,* 107-122.

[9]. Spark Internals - Spark Summit 2014 http://spark-summit.org/wp-content/uploads/2014/07/A-Deeper-Understanding-of-spark-internals-Aaron-Davidson.pdf

[10]. Spark Job Flow - Databricks https://databricks-training.s3.amazonaws.com/slides/advanced-spark-training.pdf

[11]. AaronDavidson, http://www.cs.berkeley.edu/~kubitron/courses/cs262a-F13/projects/reports/project16_report.pdf, Andrew Or. Optimizing Shuffle Performance in Spark. Technical Report

[12]. Machine Learning, Wikipedia, 2014 http://en.wikipedia.org/wiki/Machine_Learning
[13]. Machine learning with Spark - Spark summit 2013 https://spark-summit.org/2013/exercises/machine-learning-with-spark.html
[14]. Lydia, L. E., Govindaswamy, P., Lakshmanaprabu, S. K., & Ramya, D. (2018). Document Clustering based on Text mining K-means algorithm using Euclidean distance similarity, *Journal of advanced research in dynamical & control systems*, *10*(2), 208-214.
[15]. Lydia, L. E., & Ramya, D. (2018). Text Mining with Lucene and Hadoop: Document clustering with updated rules of NMF Non-Negative Matrix Factorization. *International Journal of Pure and Applied Mathematics*, *118*(7), 191-198.
[16]. Muruganantham A., Nguyen, P. T., Lydia, L. E., Shankar, K., Hashim, W., & Maseleno, A. (2019). Big Data Analytics and intelligence: A perspective for Healthcare. *International Journal of Engineering and Advanced Technology, 8*, 861-864.
[17]. Chen, Z., Xu, G., Mahalingam, V., Ge, L., Nguyen, J., Yu, W., & Lu, C. (2016). A cloud computing based network monitoring and threat detection system for critical infrastructures. *Big Data Research*, *3*, 10-23.
[18]. Celli, F., Cumbo, F., & Weitschek, E. (2018). Classification of large DNA Methylation datasets for identifying cancer drivers. *Big Data Research, 13*, 21-28.
[19]. Subbu, K. P., & Vasilakos, A. V. (2017). Big Data for Context-Aware Computing - Perspectives and Challenges. *Big Data Research, 10*, 33-43.
[20]. Behrooz, H., & Kiani, K. (2019). A Big Data driven distributed density based hesitant fuzzy clustering using Apache spark with application to gene expression microarray. *Engineering Applications of Artificial Intelligence, Elsevier*, *79*, 100-113.

How to cite this article: Lydia, E. L., Vidhyavathi, R. M., Pustokhina, I., and Pustokhin, D. A. (2020). Comparative Performance Analysis of Apache Spark and Map reduce using K-Means. *International Journal on Emerging Technologies*, *11*(2): 198–204.