International Journal of Industrial Eng. & Production Research (2020) pp.

Volume , Number , 2020

International Journal of Industrial Engineering & Production Research

Journal Website: http://ijieen.iust.ac.ir/

Research on Advanced Streaming Processing on Apache Spark

K.V.K.Sasikanth, K.Samatha, N.Deshai, Dr B.V.D.S.Sekhar, Dr S.Venkatramana

A.K.V.K.Sasikanthr, *Department of CSE, GITE, Rajahmundry, A.P, India*
K.Samatha, *Department of CSE, JNTUK, Kakinada, A.P, India*
N.Desai, *Department of IT, SRKREC, Bhimavaram, A.P, India*
Dr B.V.D.S.Sekhar, *Department of IT, SRKREC, Bhimavaram, A.P, India*
Dr S.Venkatramana, *Department of IT, SRKREC, Bhimavaram, A.P, India*

KEYWORDS

Big Data,
Hadoop,
HDFS, Map
Reduce,
Apache Spark,
Processing,

ABSTRACT

The Today's digital world computations are tremendously difficult and always demands for essential requirements to significantly process and store enormous size of datasets for wide variety of applications. Since the volume of digital world data is enormous, this is mostly generated unstructured data with more velocity at beyond the limits and double day by day. In last decade, many organizations have been facing major problems to handling and process massive chunks of data, which could not be processed efficiently due to lack of enhancements on existing and conventional technologies. In this paper address, how to overcome these problems as efficiently by using the most recent and world primary powerful data processing tool, which is hadoop clean open source and one of the core component called Map Reduce, but which has few performance issues. This paper main goal is address and overcome the limitations and weaknesses of Map Reduce with Apache Spark.

1. Introduction

The Today's digital world significantly enhances emerging technologies, new mechanism, techniques, tools so media such as social networks generate huge data and which size rises exponentially in each year. The IDC research forecasts that by 2020 the total data could increase 50 times, mostly driven by different types of sensors, medical equipment, banking, twitter, Face book, Google etc. Ninety percent of the last decade applications are generated unstructured data, such as documents, mail, images etc. Nearly similar to three twitter messages per minute for a period of 26,976 years actually [1]. In regards, the overall number of servers running data stores worldwide could increase tenfold during the next decade. Big data become hot topic at present decade, which seems to be an enormous size, which is majority of the unstructured data, typically never, be processed by using conventional computational tools and methods as efficient, scalable, cost effective and reliable manner. In digital world, the term big data is the latest era described for immense structured, semi, unstructured data sets that are handling with conventional data processing tools, and methods but which is insufficient and poor performance.

* **Corresponding author.**
Email: *sasikanth@giet.ac.in,* ***Email:*** *desai@srkrec.ac.in*

Big data significantly support to efficiently analyze the in-depth concepts for the better intelligent decisions and strategic taken for the development of the organization. Big data distinctly address with characteristics six V's volume, velocity, veracity, variety, value, variability and validity [2].

2. Background Challenges

Major limitation from existing approaches because process only on low volume of datasets, which could be hosted only by standardized servers of the database, or until the data processor is limited. However, it is really an exhausting environment to process such information using a single data base capacity because when it comes to managing massive amounts of extensible data. Map Reduce needs a more time to perform map and reduce tasks thereby raising the latency. Reduces data processing velocity and increasing the delay during more distributed

and processed the large data is across the cluster in Map Reduce [3].

Major problem: This is the real time computation therefore which is more essential to complete the job without delaying the next step or actual deadline of completion.

Critical path issue: Internet world computations are highly demands real time manner to complete each job significantly without halting the next step or exact deadline of completion. Therefore, if one computer slows down the actual work, all the other work will be delayed.

Problem of reliability: Typically every machine working with part of the data. However, which is a major challenge to handle this disaster.

Equal splitting problem: splitting the data into slightly smaller pieces in order for every computer to just use a portion of the data. This means that the data can be split up in such a way that no such single machine has been overwhelmed or used.

Single splitting can fail: we could not calculate the outcome when a device fails work to produce the total output. A framework should guarantee the systems achieve more fault tolerance ability.

Result aggregation: This mechanism could be apply to aggregate the outcome produced from final output by the each device, since these are the difficulties we need to deal with separately while just processing massive data sets only in parallel just using conventional methods [4].

3. Map Reduce

Google's has introduced a latest tool to significantly solve existing technology difficulties by Map Reduce which is simple, open source, more distributed and parallel processing framework. This is a first and most latest digital world framework, widely used to process incredible volumes of world data on a large commodity cluster regarding reliability, high fault tolerance, great scalability and more reliable manner. In December 2004, Google issued a journal on Map Reduce. The great benefit of Map Reduce is that multiple computer nodes are easy to modify data processing [5]. Apache Hadoop Map Reduce is being use extensively in the past decade for parallel computing of massive-size data; this is becoming de facto a benchmark in these areas. Map Reduce perform two different tasks, Map and Reduce, which takes place entirely after the completion of the mapped phase. The map task, which is actually read and extremely processed a data block at intermediate level outputs for producing key value pairs. The reducer receives from several map jobs the key value pair then added to a smaller set of multiples or key-value pairs (the final output) by means of intermediate datasets (intermediate key-value pair) as shown in Fig.2.

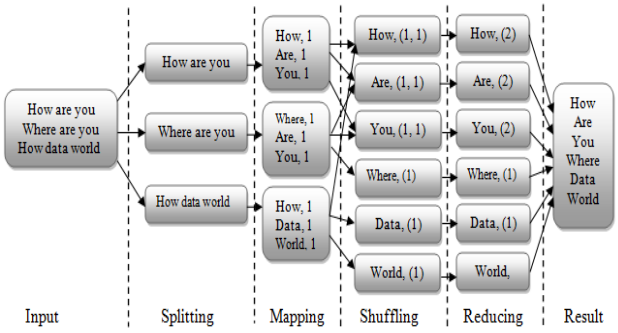

Fig.1.MpaReduce word counting

Features of Map Reduce

Parallel Processing: Typically split the job between several nodes on Map Reduce moreover, every node operates parallel mostly with part of the job. Map Reduce follows divide and conquer strategy, which enables us to manipulate data with multiple machines [6]. Actually, the time taken to extremely process the data is significantly reduced by a number of machines rather than by one machine in parallel.

Table 1: Map phase with key and values

	Input	Output
Map Phase	<key1, value1>	list (<key2, value2>)
Reduce phase	<key2, list(value2)>	list (<key3, value3>)

Fault Tolerance: Hadoop extremely controls the faults with the help of replication factor. Whenever user store a particular file in hadoop storage component HDFS, which is partition the file into number of blocks and distribute data blocks over the various machines in HDFS cluster. In addition, to generate the default replica value 3 of each block is on other cluster machines. If one machine in the cluster could fails during critical circumstances. Therefore, the user could gain data from other machines [7].

Scalability: Typically, hadoop has one of the major strength was scalability. Therefore, very simply added new nodes with there are no downtime. Hadoop supports horizontal scalability therefore latest nodes are added on the fly manner to the machine. In Apache hadoop, every application can run significantly on more than thousands of nodes.

Reliability: In hadoop, the whole data become more reliable which is stored on the cluster of machines regardless of machine failure because replication mechanism can support to gain same data from different place. Therefore, if any of the nodes fails, then also we can store data reliably.

High Availability: During the more copies of datasets, therefore actual data is more available and easily access in even though hardware face failures. Therefore, any device goes down but our required data could be retrieved from another way.

Data locality: The major limitation of Hadoop has more crossing-switching system traffic from the enormous quantity of data. Therefore tremendously beat this problem, Data Locality came into reality. Hadoop can support to move the computation very closely tied with real data, which actually resides on the cluster node. Therefore, which extremely reduce network congestion

but to enlarge the system throughput.

Algorithm for counts the appearance of each word in a set of documents:

Step1: function map task (String name, String document):
Step2: for each word w in document:
Step3: emit (w, 1)
Step4: function reduce task (String word, Iterates partial Counts):
Step5: sum = 0
Step6: for each pc in partial Counts:
Step7: sum += pc
Step8: emit (word, sum)

Lack of real-time processing, because it could not continuous employ every aspect by Map Reduce. While your intermediary process mostly require to respond to each other (each jobs run in separation).Typically, processing needs huge data could be shuffle across the network. Whenever you require handling streaming processing with Map Reduce is highly difficult. Because Map Reduce is most excellent suitable to batch process enormous amounts of data [8].

Some major issues on hadoop:

Small files, processing speed very low, high latency, less security, poor real time stream processing, efficiently support up to batch processing only, more uncertainty, line of code is complex, no mechanism of caching, difficult ease of use, generally vulnerable, lack of delta iterations, poor interactive processing, lack of in memory and graph processing [9]. Even though its programming interface is low. Along with the continually rising actual size of data-disseminated programming existing models including Map Reduce and its open source application Hadoop face performance problems. Apache Spark was created to address the problems and drawbacks of Map Reduce.

4. Apache Spark

Apache Spark is really a fastest general-purpose cluster-computing model, which is more distributed, parallel and completely open-source. It was originally developed in 2009 and opened in 2010 as an Apache project in the UC Berkeley's AMPLab [10, 11, 12]. Spark offers more interface mostly with underlying data parallelization and fault tolerance for programming on whole clusters. Spark has significantly proven incredibly popular and which is more widely used by many real world big corporations for enormous, multi- peta to zetta byte data storage and analysis computations. This has fairly been since lighting fast and in-memory processing. Last year, Apache Spark place a world witness by implementation a benchmark examination involving sorting 100 terabytes of data in 23 minutes - the previous world record of 71 minutes could held by Hadoop. Spark Core, majorly at the centre of a project providing decentralized functionality, programming and transmission, offer programmers a significantly faster and flexible effective alternative to Map Reduce. Developers from Spark conclude that when properly processed massive data in memory approach it can work effectively, and extremely lighting faster means 100 times faster than Map Reduce also 10 times faster than disk. Spark simply provides more scalability, more fault tolerance, reliability, and several other features [13, 14, 15].

Spark and its RDDs did reveal in 2012 while the response to weaknesses in the Map Reduce cluster computing model, which makes a selective straight dataflow construction on distributed applications: Map Reduce applications gather input data during the disk source, then map a responsibility over the data, degrade the outcome of the map task, and tremendously store reduce task outcome on disk. Spark's RDDs could perform similarly to working set to distributed applications, which contribute an (intentionally) reduced pattern of distributed oriented shared memory. Apache Spark has complete service of architectural establishment with resilient distributed dataset (RDD), which support extremely read-only operation on number of data items.

circulated across the cluster, which significantly implement in a fault-tolerant manner.

The Data frame API could be generated as a notion on top of the RDD. In apache, Spark the starting interface that is especially an application-programming interface (API) called the RDD. Spark strongly supports memory processing to enhance the performance of applications for big data analysis, but it could also execute traditional disk-based treatment whenever data sets are far too large for the system memory available. The RDD is specifically designed so that customers can cover up a great deal of the computer complexity. It aggregates data and partitions it across a whole server cluster where it could be calculated, migrated or simply run via an analytical paradigm in another data store [16].

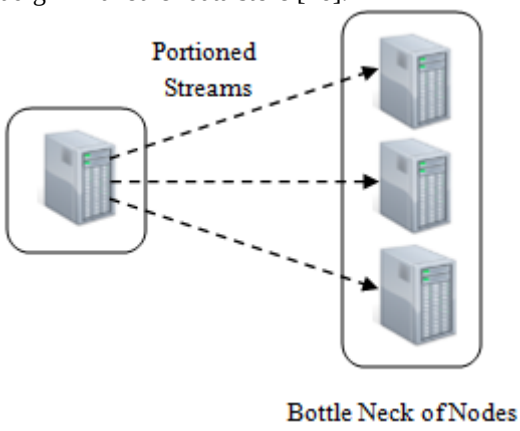

Fig.2.Conventional System with Static Scheduling

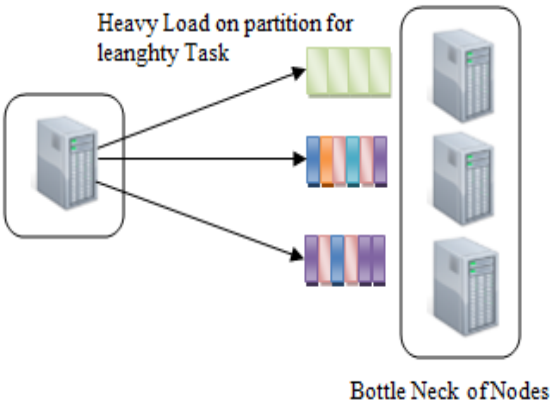

Fig.3.Spark Streaming with Dynamic Scheduling

In the Conventional approach utilized by many systems, but distributions are larger computationally exhaustive than the others, during the nodes are assigned statically to process that partition based on batch processing but introduce a major issue is a bottleneck also slow down the entire process. Overcome this issue to use Spark streaming as shown in Fig.1, 2. The Apache Spark most significant benefit of utilizing is speed. Digital world Applications could operate 100X active in-memory and 10X quicker during working on disk drive. Compare with Hadoop, spark keep the intermediate outcomes in-memory (rather than disk). Spark significantly overcomes the number of disk I/Oís. The Fig.4 reveals a flash of the logistic regression in Hadoop and Spark and they denote starkly separated by the operating times.

Spark greatly supports the tremendously deploying of iterative-based techniques, which encourage their data set various times during a loop service, and especially the data analysis with complete interactive manner. Spark achieve low latency of those applications could be degraded by various request of dimension compare with Apache Hadoop Map Reduce execution. Between the sorts of iterative techniques are the train-based methods for machine learning operations, which made the fundamental incentive for improving Apache Spark. It can also easily handle large batch and real-time world analytics, significant data processing tasks and interactive queries, and machine learning. Apache Spark accomplishes higher performance for both batch and streaming information, just using a state-of - the - art DAG scheduler, a query optimizer, and a psychical execution engine. Apache Spark always needs a cluster based administrator and the latest parallel and more distributed storage segment. During batch control, spark recommends standalone mode, Hadoop YARN, or Apache Mesos. During propagated accommodation, Spark makes interface among an extensive diversity, including Hadoop Distributed File System (HDFS), and a custom solution could be executed. Spark further establishes a pseudo-distributed restricted mode, normally utilized simply for improvement or examination objectives, where disseminated accommodation is not required and the local file operation can be utilized alternately in such a situation, Spark is operated on a single machine including one executor through one CPU kernel.

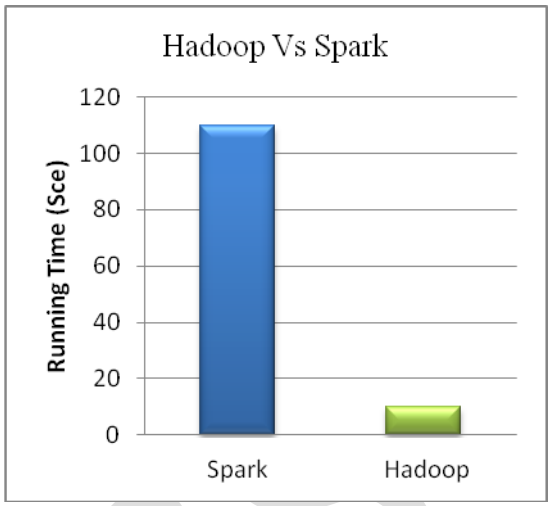

Fig.4.Hadoop and spark Running Time

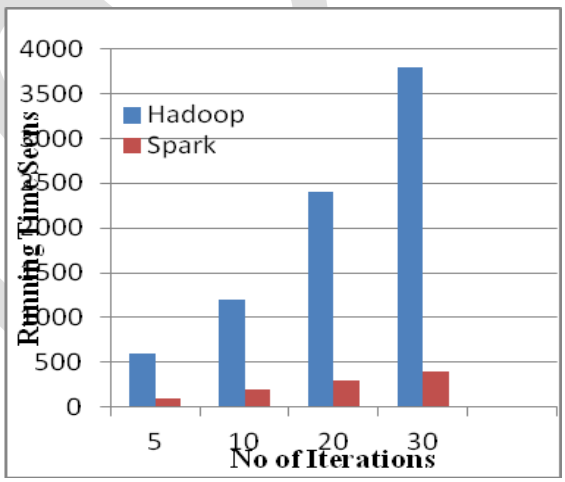

Fig.4.Hadoop and spark Running Time

Spark libraries

The Spark Core engine processes mainly provide high level API and actually support a similar set of related data-management and analysis tools. In addition to spark core, a new package of most popular code libraries to be used in data analysis software programs also actually comes with an apache spark environment. Spark SQL allows users to query stored data in different applications in the relevant SQL language [10, 12]. Spark streaming can simply build an application to evaluate and present information even in real-time.

	Hadoop Map Reduce	Spark
Processing	Batch	Micro Batch, Stream
Speed	Slow	Faster than MR
Operators	NA	Time-based
Windows	No	Yes
Storage data	HDFS	In-Memory
Latency	High	Low
Fault tolerance	High	High , RDD DAG
Performance	Slow	High than MR
Remove duplicate	High	Process records exactly
Iterative data flow	Chain of states	Cyclic data flow DAG
Scalability	Incredible up to 10,000	High cluster of 8000
Visualization	Low	High, need RAM
Recovery	high fault tolerant	RDD DAG
Abstraction	NO	Spark RDD Data stream
Easy to use	Difficult	Easy
Realtime analysis	No	Good
Scheduler	Fair, capacity	Own flow scheduler
SQL	Hive	SSQL Hive, FDSL
Catching	Not	Yes
Hardware	Commodity h/w	Mid to high level h/w
Machine learning	Mahout	Mlib
Line of code	1,20,000	20,000
Deployment	Fully distribute mode	Standalone on mesos/YARN

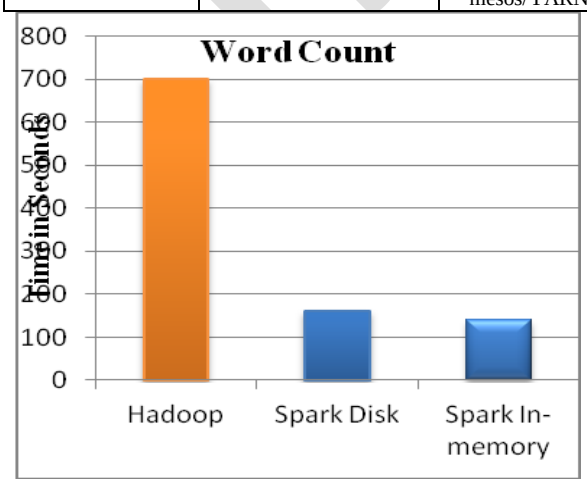

Fig.5.Hadoop and spark Word Count Time

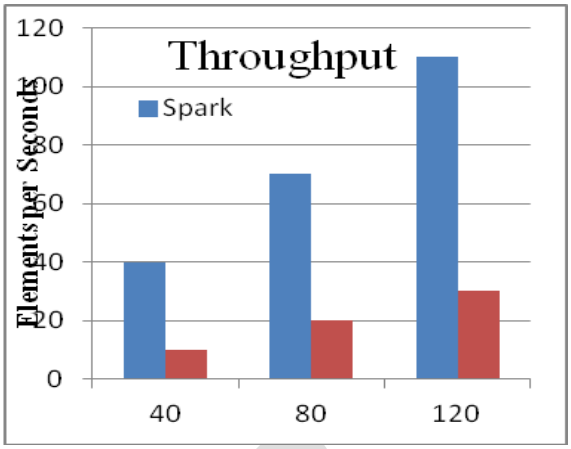

Fig.6.Hadoop and spark Throughput

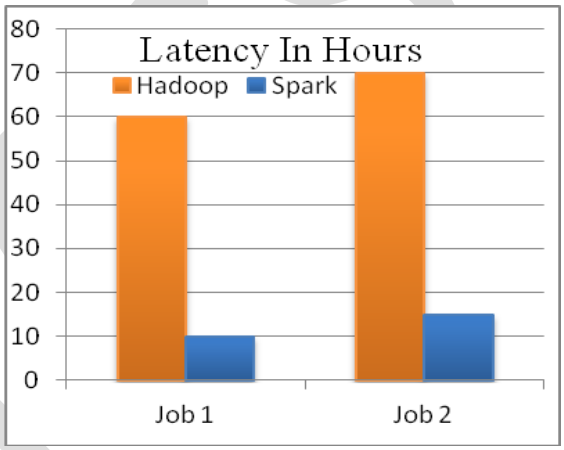

Fig.7.Hadoop and spark latency

MLlib is a device trying to learn code library that allows customers to use advanced mathematical operations on spark information and create new applications for those analysis. GraphX-which is a graph-parallel numerical computation online tool actually built-in library. Spark typically provides more than 80 operators, which simply make parallel applications easy to develop. Although from the Scala, Python, R and SQL shells then you really could use it interactively.

Spark enables a pack of libraries, along with SQL and Data Frames, MLlib, GraphX, and Spark Streaming. In almost the same application, you can dynamically incorporate these libraries. Then you really can continue to run Spark mostly on EC2, Hadoop YARN, Mesos, or Kubernetes by using its independent cluster mode [14, 16, 17 ,18, 19]. HDFS, Alluxio, Apache HBase, Apache

Table 1: Sample document with Customer Tweets

Hive and hundreds and hundreds of many other sources of data have access to data. Swift Processing, dynamic in nature, more in-memory computation in spark, reusability, fault tolerance in spark, real-time stream processing, lazy evaluation in apache spark, support

multiple languages, active, progressive and expanding spark community, support for sophisticated Analysis, integrated with hadoop, spark graphX, cost effective manner.

5. Citations and References

Today's digital world follows most centralized analytics engine for large-scale data processing. Present internet applications achieve more speed, storage and stream processing across the huge data as reliable manner during Apache Spark framework. Spark main goal to overcome the performance issues, limitations and weaknesses of Map Reduce. Because it provides good latency, throughput, fast execution, stream processing.

6. Citations and References

[1] Prathyusha, Rani., Yiheng, L., "Data analysis using hadoop MapReduce environment," IEEE International Conference on Big Data, 2017, pp.4783-4785.

[2] Bichitra, M., Srinivas, S., Ramesh, Kumar, S.,"Architecture of efficient word processing using Hadoop MapReduce for big data applications," International Conference on Man and Machine Interfacing (MAMI),2015, pp.1-6.

[3] Sheoran, D., Malathi, K., Kumar, Senthil., " Map reduce scheduler: A bird eye view ," International conference of Electronics, Communication and Aerospace Technology (ICECA), Vol.1, 2017, pp.213-217.

[4] Hisham, M., Stephane, M.,"Enhancing MapReduce Using MPI and an Optimized Data Exchange Policy" 41st International Conference on Parallel Processing Workshops 2012, pp. 11-18.

[5] Jia-Chun, L., Fang-Yie, L., Ying-ping, C.," Impacts of Task Re-Execution Policy on Map Reduce Jobs ,"The Computer Journal,Vol.59, 2016, pp.701-714.

[6] Maedeh, S., Nishant, S., Kumar, S.," Hadoop-MapReduce: A platform for mining large datasets " 3rd International Conference on Computing for Sustainable Global Development (INDIA Com), 2016, pp.1856 -1860.

[7] Akhil, G., Bharti, G., " Study on emerging implementations of Map Reduce ," International Conference on Computing, Communication & Automation, 2015, pp.16-21.

[8] Joshua, S., Jonathan, V., Enyue, L.," Analyzing Patterns in Large-Scale Graphs Using MapReduce in Hadoop ," SC Companion: High Performance Computing, Networking Storage and Analysis, 2012, pp.1457-1458.

[9] Deshai, N., et al Performance and Cost Evolution of Dynamic Increase Hadoop Workloads of Various Data centers, Springer Nature Singapore, Vol.105, No.1, 2019, pp.505-516.

[10] Deshai, N., "A cross study on apache hadoop and yarn", International Journal of Engineering & Technology, Vol.7, No.4, 2018, pp.4850-4855.

[11] Deshai, N., "Study with Comparing Big Data Handling Techniques using Apache Hadoop MapReduce Vs Apache Spark", International Journal of Engineering & Technology,Vol.7, No.4, 2018, pp.4839-4843.

[12] Deshai, N.,et al," Big Data Challenges and Analytics Processing Over Health Prescriptions", Jour of Adv Research in Dynamical & Control Systems,Vol.15, No.1,2017, pp.650-657.

[13] Deshai, N.,"Big Data Hadoop MapReduce Job Scheduling: A Short Survey",Information Systems Design and Intelligent Applications, 2019, pp.349-365.

[14] Deshai, N.,"A Study on Big Data Hadoop Map Reduce Job Scheduling, 'International Journal of Engineering & Technology"Vol.7, No.3.31, 2017, pp.59-65.

[15] Deshai, N.,"An advanced comparison on big data world computing frameworks",Journal of Physics: Conference Series,Vol.1228, No.1,2019, pp.12003-12011.

[16] Deshai, N.,"MLlib: Machine Learning in Apache Spark', International Journal of Recent Technology and Engineering (IJRTE),Vol.8, No.1S3, 2019, pp.45-49.

[17] Deshai, N.,"Protect Internet from Intrusion with Advanced Spark Framework, International Journal of Recent Technology and Engineering (IJRTE), Vol.8, No.1S3, 2019, pp.186-190.

[18] Deshai, N.,"Processing Real-World Datasets Using Apache Hadoop Tools",International Journal of Recent Technology and Engineering (IJRTE), Vol.8, No.1S3, 2019,pp.209-213.

[19] N. Deshai,"A Study on Big Data Processing Frameworks: Spark and Storm",Springer, Vol.160, 2019, pp.415-424.

DNO	Text	Class
1	I liked this Product	+
2	I Hated this Product	-
3	A great Product ,Good Product	+
4	Poor Quality	-
5	A great Product, Good Quality	+

Table 2: Document with Positive and Negative Words

DNO	I	Liked	This	Product	Hated
1	1	1	1	1	1
2	1		1	1	
3				2	
4					
5					

Table 3: Document with Positive and Negative Words

DNO	A	Great	Poor	Quality	Good	Class
1	1	1	1	1	1	
2	1		1	1		
3				2		1
4						
5						1

Table 4: Document with Positive Words

DNO	I	Liked	This	Product	Hated	Class
1	1	1	1	1	1	+
2				2		+
3						+

Table 5: Document with Positive Words

DNO	A	Great	Poor	Quality	Good	Class
1						+
2	1	1			1	+
3		1		1	1	+

SA plan to find out the approach of the speaker or a writer in terms of subject or the taken as a whole relative polarization of a text in order to implement this project I have used naive bayes algorithm [6]. Let us try to understand a ways with an example. Here I have five documents with movie reviews of which three are positive and two are negative. if we observe there are total ten unique words in these documents. Here I have listed out all the unique words and the frequency of each word in a document as shown in Table1.

Here Table1 has a document with 5 customer tweets, also it has total 10 unique words are [I, Liked, This, Product, A, Great, Hated, Good, Quality, Poor]. Finally this document evaluated with number of unique words and the estimate the frequency of each word.

Let's look at the probability of a positive outcome. These are the documents with positive reviews. So the probability of positive is three over five that is 0.6. now we will have to calculate the probability of each word being positive the formula to calculate this is NK which is the frequency of the word or the quantity of time the word happen N which is the total number of + ve words or -ve words vocabulary is the total amount of unique words while testing if we get an unknown word we use NK equals 0 and find its probability being both +ve and -ve. This is how we calculate probabilities using neighbor ways.

Positive =3/5=0.6 Compute the Document Step by Step using Naïve Bayes Algorithm:

Step1:P(I/+);P(Liked/+);P(This/+);P(Product/+);P(Hated/+);P(A/+);P(Great/+);P(Poor/+);P(Quality/+);P(Good/+);P(Class/+);

Step2: P (WK/+) = (Nk+1)/n+|vocabulary|

Step3: Nk: How many times word K happened in these cases (+)

Step4: N: How many words in (+) case: 14

Step5: Vocabulary: overall distinctive Words

Step6: P (WK/+) = (nk+1)/n+|Vocabulary|

Whereas test for unidentified words we utilize nk=0 and finds its Probability being both +ve and -ve.

The flow chart of the entire project:

First step: is to train the classifier with a trained data with labels that is positive or negative depending upon the review.

Second step: is the test classifier the

Third step: is the get sentiment of a given sentence or the word.

Now let us see each of the mining time in the trained classifier. Stop words is a list of neutral words like nouns articles prepositions etc. we eliminate these stop words and for the remain terms regarding on the label. We divide reviews into +ve and -ve and find out the probabilities of every word and dump into pickle file Test classifier. We present input as the test data again remove the stop words from the testing data and for each of the remain words. We verify if the word is in the training data or not if yes we get the probability related with it which could be positive or negative if not, we compute the probability based on a naïve bayes algorithm [7]. After repeating this for all the words we add these probabilities individually for both positive and negative along with prior probabilities. Now if the positive probability is higher we say it is positive or who say is negative.

1. Sentiment Analysis with Apache Spark

Sentiment analysis is described as the process of obtaining techniques for the identification and extraction of data from unstructured data through NLP and text analysis. The study of emotions has been used in projects. This facilitates decision-making, where it allows making the right choice by gathering the thoughts of people through their feedback and comments. Of example, on the basis of customers ' feedback, several retailers sell latest products or develop their new ones [8]. It's also used to assess assumptions for politicians in the election based on an individual's feelings. The following paragraphs address essential principles relevant to the study of emotions.

2. Spark-based Sentiment Analysis

The uniqueness of large data introduces the latest difficulties to sentiment analysis. Generally, the massive size, variety and velocity of data are generated from various digital world applications. Due to this reason, more requirements are huge demands to utilize advanced big data and Hadoop frameworks for effective sentiment analysis [23,24,25]. Various types of research suggested improving the effectiveness of sentiment analysis to extracting people feelings using big data frameworks. In recent research base on machine learning, lexicon-based and hybrid methods. In machine learning methods including Naïve Bayes, support vector machine classifier. From those methods, the Naïve Bayes was the frequently utilized technique due to its more precision flows in contrast beside the extra methods [26,27,28]. The scalability on Naïve Bayes to efficiently handling huge twitter datasets has been evaluated on a Spark framework. Spark Streaming denotes an extension of the heart of Spark API that facilitates more scalable, large throughput, higher fault-tolerant stream processing across the digital world application live data streams. Spark Streaming could be utilized to a stream of live data and processing with more real times. While it arrives in Real-Time Analytics across world datasets, Spark Streaming offers a particular platform to ingest data to significantly achieve

quick and live processing [29]. Data Streaming is an essential technique for transporting data so that this could be prepared as a constant and endless stream. Streaming technologies are converting major leading and more demand with the extension of the digital world. Now we could use Apache Spark Streaming to streaming the real-time data from multiple applications like Twitter, Stock Market and Geographical Systems and play great analytics to the success of markets.

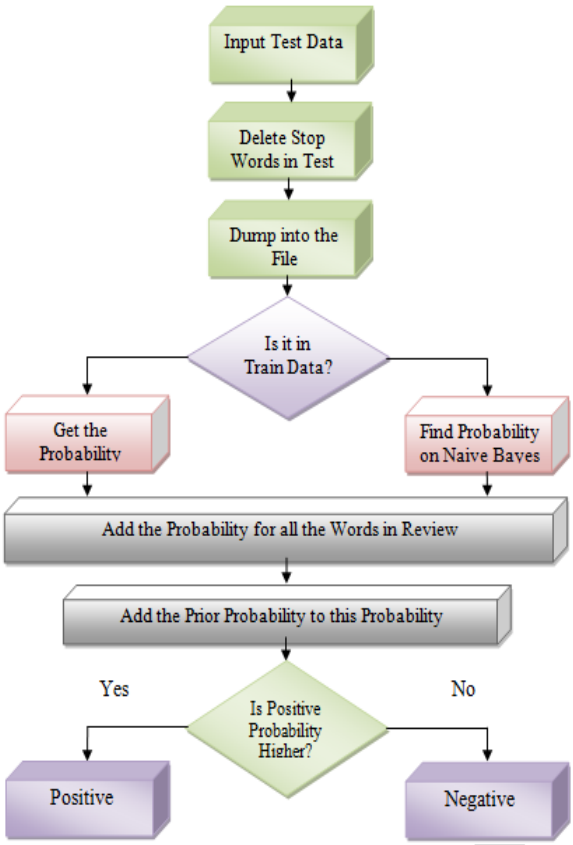

Fig 2.Train Naïve Bayes Classifier

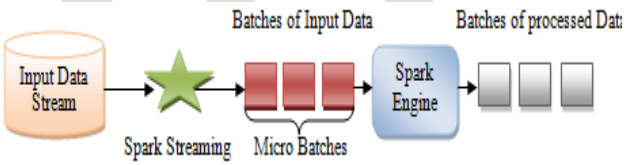

Fig 3.Apache Spark Stream Processing

The fundamental stream element is a stream which is mostly a sequence of RDDs to efficiently process the real-time data across digital world applications as shown in figure 3.

Spark Streaming Features

Scaling: It could simply increase to 100 of nodes and more nodes.
Speed: This is easily attained low latency.
Fault Tolerance: which becomes the more capability to powerfully regain from failures?
Integration: Spark combines easily batch, micro-batch also real-time processing applications.
Business Analysis: This implies completely supported to trace the activities of each customer in market aspects [30].

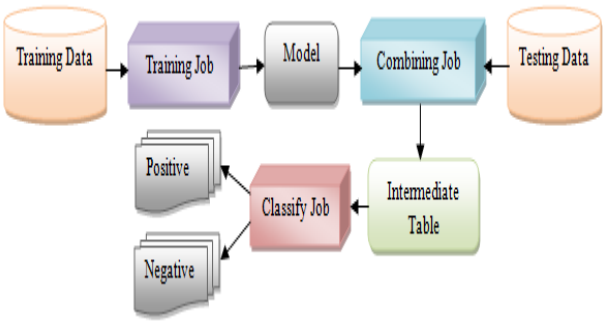

Fig 4.Naïve Bayes Evaluation system on Apache Spark

"We aim to deduce the user's feeling by evaluate their tweet text. In Fig 4 we apply Navie Bayes Evolution System for handling the world large twitters dataset and easily achieve customers feeling and opinions with the help of some essential actions for efficient sentimental analysis as shown in Figure. Initially twitter data collecting, perform some preprocessing aspects, stored as a training data for training job then built the Navie Bayes classifier model based on top of Apache spark framework the with high scalability [9,10,11,12]. Combing the each job and find each job accuracy with best classifier model and prepare a testing data for finding the best classifier base on precision and sentimental like positive and negative opinions [31]. The components meant the controller of the workflow (WFC), the compiler of content, the terminals of the customer and the locator of the output. The group's rates have been as follows: first, the WFC framework used to develop the framework to conduct the learning job as shown in Fig 4. A combiner has been used to integrate the testing data with just the standard, resulting in intermediate outcomes. The reports became eventually listed by measuring every report's probability. The entire system overall precision is about 82 percent.

Variety of Naive Bayes (NB) and Support Vector Machines (SVM) become commonly seen as basic classification strategies, but its efficiency varies greatly based on the framework configuration, functionality used throughout different jobs and repositories [13, 14, 15].

They demonstrate that: I the addition of term token features that offers consistent performance on sentiment analysis activities; (ii) for short fragment sentiment functions, NB is doing well again rather than SVMs (while the reverse results persists with lengthy files); (iii) a basic yet modern SVM variant uses NB log-count proportion as feature level performs consistently well in both functions and databases [16, 17, 18, 19]. MNB is generally superior and other steady than multivariate Bernoulli NB, and the gradually more recognized outcome that binaries MNB is improved than normal MNB.

For the SVM, x (k) = ˆf (k), and w, b are achieved by reducing
wT w + C X I max(0, 1 − y (i) (wTˆf (i) + b))2
We observe this L2-regularized L2-loss SVM to work the most reliable and L1-loss SVM to be less stable.
While this seems very strong for lengthy texts, we find that interpolation between MNB and SVM performs excellently for all documents and we report results using this model:
w0 = (1 − β) ¯w + βw
Wherever w¯ = ||w||1/|V | is the mean importance of w, and β ∈ [0, 1] is the insertion parameter. This insertion can be seen as a model of regularization: imagine NB except the SVM is really trusting.

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

$$P(c|X) = P(x_1|c) * P(x_2|c) * \ldots * P(x_n|c) * P(c)$$

$$c^* = \arg\max_c P(c|d)$$

P (c|x) Posterior Probability.
P (x|c) Likelihood.
P (c) Class Prior Probability.
P (x) Predictor Prior Probability.

3. Experimental outcome

The intention about this testing is to evaluate the precision from popular MLlib's model Naive Bayes, support vector machine. Regarding accuracy to determine the complete routine of each classifier [21, 22, 23, 24, 25]. Because the precision is an excellent measure to choose which classifier is the best to handle large database. The precision metric is illustrated as follow: *AAAAAAAAAAAAAAA* = (*TTTT* + *TTTT*)/ (*TTTT* + *TTTT* + *FFFF*+ *FFFF*) Where TP, TN, FP, and FN are true +, true -, false + and false -, correspondingly. The estimated outcomes are shown in table 2. As shown from table2, the most excellent presentation was accomplished with Support Vector machine behind with Naïve Bayes classifier behind with logistic regression classifier.

In the area of machine learning and particularly the issue of numerical identification, a confusion matrix, also recognized as an error matrix, is a particular table structure which enables representation of an algorithm's output, generally a supervised learning model (generally considered a match matrix in unsupervised learning). Every row of the matrix describes the circumstances in a projected class whereas each column describes the circumstances in an actual class (or vice versa).[21, 26, 27 ,28,29, 30, 31, 32, 33] The named derives from the fact that it allows it easier to see whether the method confuses two categories (i.e. generally misrepresenting one class as the other).

$$\text{classify}(f1, \ldots, fn) = \text{argmax}\, p(C = c) \prod_{i=1}^{n} p(Fi = fi \backslash C = c)$$

Table 6: Evolution Result

Classifier	Accuracy
Naïve Bayes	**87%**
Support Vector machine	**85%**

4. Conclusion

The whole paper brought new observations by introducing various classified techniques, Naïve Bayes and Support vector machine to characterize opinion of huge-scale information utilizing Spark's MLlib. Since it is versatile to deal with a large amount of data, Apache spark machine library (MLlib) has been used. The tests were performed using dataset of Amazon reviews comprising four million ratings, 50,000,000 training reviews and 40,000 test feedback. Several pre-processing procedures have been implemented to clear up and get ready the information for classification. Two classifiers, naïve Bayes and Support vector machine were measured in terms of effectiveness. Test results revealed a higher performance of the naive bayes classifier than any of the other classifications.
Cotter, N.E., Guillerm, T.J., " The CMAC and a Theorem of Kolmogorov," Neural Networks, Vol. 5, 1992, pp. 221-228.

References

[1] Kaur, H., V. Mangat, V., Nidhi, N., "A Survey of Sentiment Analysis techniques," International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2017, pp. 921-925.

[2] Liu, B., Blasch, E., Chen, Y., Shen, D., Chen, G., "Scalable Sentiment Classification for Big Data Analysis Using Naïve Bayes Classifier," in 2013 IEEE International Conference on Big Data, USA, 2013.

[3] Edison, M., Aloysius, A., "Concepts and Methods of Sentiment Analysis on Big Data," International Journal of Innovative Research in Science, Engineering and Technology, Vol. 5, No. 9, 2016, pp. 16288-16296.

[4] Patel, P., Prabhu, G., Bhowmick, K., "A Survey of Opinion Mining and Sentiment Analysis," International Journal of Computer Applications, Vol. 131, No. 1, 2015, pp. 24-27, 2015.

[5] Madani, Y., Bengourram, J., Erritali, M., Hssina, B., Birjali, M., "Adaptive e-learning using Genetic Algorithm and Sentiments Analysis in a Big Data System," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 8, 2017, pp. 394-403.

[6] Biltawi, M., Etaiwi, W., Tedmori, S., Hudaib, A., Awajan, A., "Sentiment Classification Techniques For Arabic Language: A Survey," in 2016 7th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2016.

[7] Madani, Y., Mohammed, E., Jamaa, B.,"A Parallel Semantic Sentiment Analysis," in 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), Rabat, 2017.

[8] Appel, O., Chiclana, F., Carter, J., Fujita, H., "A hybrid approach to the sentiment analysis problem at the sentence level," Knowledge-Based Systems, 2016, pp. 110-124.

[9] Biltawi, M.,Al-Naymat, G., Tedmori, S., "Arabic Sentiment Classification: A Hybrid Approach," in 2017 International Conference on New Trends in Computing Sciences, Amman, Jordan, 2017.

[10] Huq, M., Ali, A., Rahman, A., "Sentiment analysis on Twitter data using KNN and SVM," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 6, 2017, pp. 19-25.

[11] Jianqiang, Z., Xiaolin, G., "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis," in IEEE Access, 2017.

[12] Madani, Y., Erritali, M., Bengourram, J.,"Sentiment analysis using semantic similarity and Hadoop MapReduce," Knowledge and Information Systems, 2018, pp. 1-24.

[13] Haddia, E., Liu, X., Shi, Y., "The role of text pre-processing in sentiment analysis," in Information Technology and Quantitative Management (ITQM2013), 2013.

[14] Hassan, S., Miriam, F., Yulan, H., Harith, A., "On stopwords, filtering and data sparsity for sentiment analysis of Twitter," LREC 2014, Ninth International Conference on Language Resources and Evaluation, 2014, pp. 810–817.

[15] Singh, T., Kumari, M., "Role of Text Pre-processing in Twitter Sentiment Analysis," Procedia Computer Science, 2016, pp. 549-554.

[16] Sharif, W., Samsudin, N., Deris, M., Nase, R., "Effect of Negation in Sentiment Analysis," in 2016 Sixth International Conference on Innovative Computing Technology (INTECH), Dublin, Ireland, 2016.

[17] Saif, H., He, Y., Fernandez, M., Alani, H., "Semantic Patterns for Sentiment Analysis of Twitter," in International Semantic Web Conference, 2014.

[18] Tartir, T., Abdul-Nabi, I., "Semantic Sentiment Analysis in Arabic Social Media," in Journal of King Saud University – Computer and Information Sciences, 2017.

[19] Biltawi, M., Etaiwi, W., Tedmori, S., Shaout, A., "Fuzzy based Sentiment Classification in the Arabic Language," in Intelligent Systems Conference 2018, London, UK, 2018.

[20] Chauhan, V., Shukla, A., "Sentimental Analysis of Social Networks using MapReduce and Big Data Technologies," IJCSN International Journal of Computer Science and Network, Vol. 6, No. 2, 2017, pp. 120-130, 2017.

[21] Parveen, H., Pandey, S.,"Sentiment Analysis on Twitter Data-set using Naive Bayes Algorithm," in 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Bangalore, India, 2016.

[22] Ramesh, R., Divya, G., Divya, D., Kurian, M., Vishnuprabha, V., "Big Data Sentiment Analysis using Hadoop," IJIRST – International Journal for Innovative Research in Science & Technology, Vol. 1, No. 1, 2015, pp. 92-96.

[23] Deshai, N., "Performance and Cost Evolution of Dynamic Increase Hadoop Workloads of Various Data centers," Springer Nature Singapore, Vol.105, No.1, 2019, pp.505-516.

[24] Deshai, N.,"A cross study on apache hadoop and yarn," International Journal of Engineering & Technology, Vol.7,No.4, 2018, pp.4850-4855.

[25] Deshai, N.,"Study with Comparing Big Data Handling Techniques using Apache Hadoop MapReduce Vs Apache Spark," International Journal of Engineering & Technology,Vol.7, No.4,2018, pp.4839-4843.

[26] Deshai, N.," Big Data Challenges and Analytics Processing Over Health Prescriptions" Jour of Adv Research in Dynamical & Control Systems,Vol.15, No.1,2017,pp.650-657.

[27] Deshai, N.,"Big Data Hadoop MapReduce Job Scheduling: A Short Survey",Information Systems Design and Intelligent Applications, 2019, pp.349-365.

[28] Deshai, N.,"A Study on Big Data Hadoop Map Reduce Job Scheduling," International Journal of Engineering & Technology, Vol.7, No.3.31, 2017, pp.59-65.

[29] Deshai, N., "An advanced comparison on big data world computing frameworks", Journal of Physics: Conference Series,Vol.1228, No.1, 2019, pp.12003-12011.

[30] Deshai, N.,"MLlib: Machine Learning in Apache Spark," International Journal of Recent Technology and Engineering (IJRTE),Vol.8, No.1S3, 2019, pp.45-49.

[31] Deshai, N.,"Protect Internet from Intrusion with Advanced Spark Framework, International Journal of Recent Technology and Engineering (IJRTE),Vol.8, No.1S3, 2019, pp.186-190.

[32] Deshai, N., "Processing Real-World Datasets Using Apache Hadoop Tools," International Journal of Recent Technology and Engineering (IJRTE), Vol.8, No.1S3, 2019, pp.209-213.

[33] Deshai, N.,"A Study on Big Data Processing Frameworks: Spark and Storm," Springer, Vol.160, 2019, pp.415-424.

[34] Zhou, K., Doyle, J.C., & Glover, K., Robust and optimal control, Prentice-Hall, Englewood cliffs, NJ, 1996.