IJAER/July- August- 2021/Volume-10/Issue-1 ISSN: 2278-9677

International Journal of Arts & Education Research

"A CRITICAL ANALYSIS OF MAP-REDUCTION RESULTS FROM HUGE DATA PROCESSING EFFICIENCY"

Ms. Manju Sharma

Ph.D. Scholar

Department of Comouter Science

Malwanchal University Indore (M.P.).

Dr. Ajay Agarwal

Supervisor

Department of Comouter Science

Malwanchal University Indore (M.P.).

ABSTRACT:- The challenge that has to be answered is how enormous volumes of dispersed data can be processed swiftly while maintaining acceptable reaction times and repeatability at the lowest possible cost. The use of multiprocessor computing in some kind of a virtualized environment is among the most effective methods for the handling of massive amounts of data. The MapReduce algorithm is at the heart of cloud computing, which is a paradigm for distributed computing that focuses on the processing of huge datasets on either a network of accessible computer nodes. Mapper is a programming model that Google presented to the rest of the planet in 2004; this is capable of running on a huge cluster of computers and has a high degree of scalability. That is a methodology that has great performance and is used to tackle issues with large-scale datasets. In a MapReduce calculation, gigabyte and terabyte amounts of item data are processed on multiple processors. Dremel is the system that Google employs to index websites. The primary objective of this system is to analyze huge amounts of data throughout parallel while they are being maintained on such a computer system. In this paper, a method is shown that uses the Software tool to handle difficulties associated with the processing of huge files in a multiprocessor fashion while dealing with a large group of workstations. It is the foundation for using the cloud based paradigm as both a new capacity industrial standard for computation.

KEYWORDS:- *Map reduction, Data efficiency techniques etc.*

The chip multiprocessing is becoming more popular of its ability to execute data-parallel programs in groups on such a single console that has a large number of cores. The programming technique known as Mapper is used in order to write code for clusters. It does this by finding solutions to issues that involve vast amounts of data being processed in parallel. The processing of multi-core central processing unit systems and mobile processors takes place on simultaneous platforms. The variations in the environments of clusters

versus multicores are due to the existence of new model regions, and these differences provide opportunities to enhance the speed of MapReduce using multicore. Granite tiles is a technique that use the "tiling approach" to divide a big MapReduce task into a handful of discrete sub-jobs and then iteratively processes each of the anti - anti. Tiled-MapReduce is a kind of iterative Elasticsearch. The potential of such computational resources is harnessed via the Mapreduce - based paradigm, which shows promise for application on multicore systems. Vinyl tile is an extension of the conventional Reduce phase that processes the extracted features of all rounds rather than just the data packet. This is accomplished by processing the tiled representation of the information. It is possible for the outcome of the Feature extraction step to be compatible with both the outcome of the Online phase in general. Change the name of the Feature extraction step inside one subjob towards the Integrate phase so that it may be differentiated from the TiledMapReduce stage, which is the final Reduction phase. Some processing processes that are involved in Tiled-MapReduce are shown in the bottom portion of Figure 11. The top section of the diagram illustrates the general program execution of either a Tiled-MapReduce task as well as the construction of Vinyl tile runtime. Every Granite tiles job is involved inside this mr dispatcher procedure. This job initializes und configures the scheduler based on the parameters and software applications (e.g., available resources). This dispatcher will then create N employees and attach each of those workers to something like a core of the CPU. Additionally, an iterative portion split is performed on incoming data inside the Incarnation Window by the coordinator. The size of this window was dynamically modified based on the realtime settings. The piece of information will be divided farther into M parts, which will ultimately result in M challenge being created. During the Map tasks, a worker will choose a paper has outlined from the repetition window anytime it is unoccupied. This will then run the Map function that was supplied by the developer, which will then process the incoming data and construct intermediary digital certificates. In order to enter a shared key into the intermediate storage, which is structured as a Md by R grid of bins, this emit intermediate procedure, which is given by the software, will be executed. R relates to the amount of Reduce jobs. During the Combining phase, the Employee will choose one of the Reduce jobs at a time and then use the combine code that was supplied by the coder in order to process one of the columns in the Intermediary Buffer. Each I by R grid of buckets makes up the iteration buffer's basic structure, so where's the overall number of times the buffer will be iterated. Because all of these sub-jobs have been completed, the Workers will call any Reduce function that was supplied by the coder in order to perform the final Reduction operation here on data that is included inside every column of both the loop buffer. Mentioning the emit procedure causes the final outcome of a keyword to be appended to the end of the buffers when the Eliminate unnecessary is used. In the end, the outputs of each Reduce job are combined and arranged in a common Output Buffer before being sent on.

The information on the many elements of Granite tiles, and its benefits and drawbacks, may be found in Table 6.

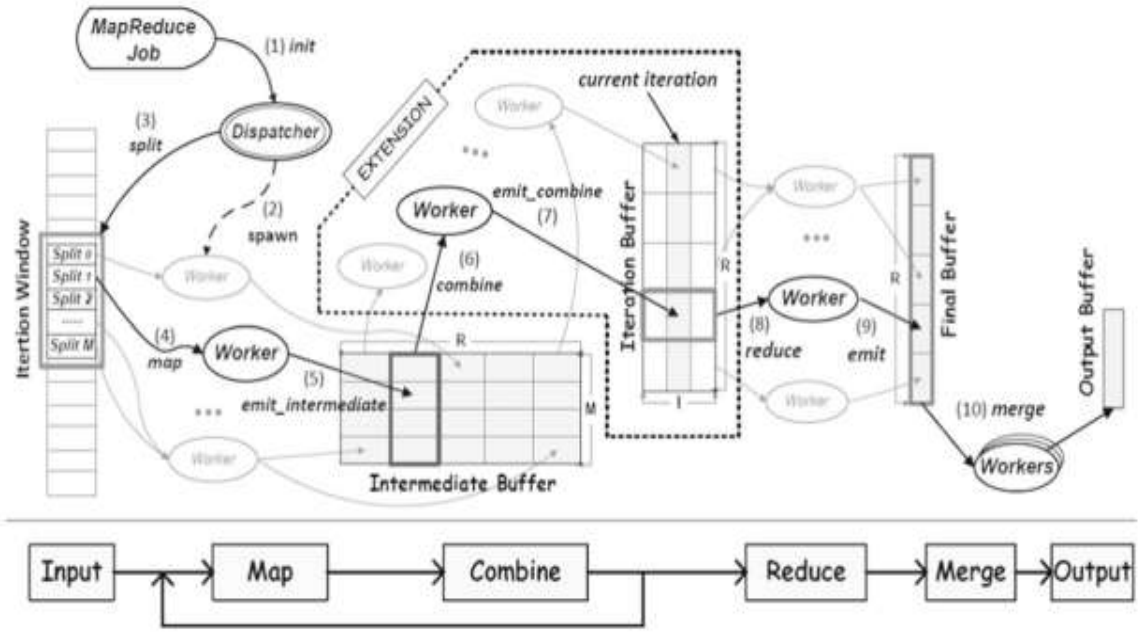

Figure 1: the workflow of Tiled-MapReduce

Pros	Cons
It provides better data locality and task parallelism.	Tiled-MapReduce investigates the potential use of the tiling strategy in the single-node version of MapReduce.
Tiled-MapReduce explores several optimizations, it improves the memory and saves up to 85% memory.	–

Table 1: the pros and cons of Tiled-MapReduce

Twister

The Mapreduce - based approach has been instrumental in the successful development of a great deal of data related software. MapReduce queries to parallel computer communities make huge use of such a ease of the scripting language as well as the excellent efficiency of the care and their convenience. In order to develop Lava runtime, Twister specifies a set of improvements that may be made to the instruction set and its architecture. These improvements are focused on the many scientific approaches that are built on

MapReduce. Several Map/Reduce activities that are controlled with all of these multiple opposite sorts of business intelligence tools are represented in Figure 12. Those tasks could be used to pack (read) any statistics that is present at the Convolution and Pooling tasks. For instance, during the normal Map task of both the computing, data item pairs are referred to this as data points, and the dynamic data that is already loaded into memory results in the production of a collection of outputting (key, value) combinations. In addition to that, this has a reducing step that may be skipped called "combine," and its purpose is to integrate the outcomes of the Detection phase into such a numerical result. Because the user software and the merged operation are carried out in the same process area, the result of the combination operation may be made immediately available to the running process. The benefits and drawbacks of playing Twister, in addition to its particulars, are outlined.

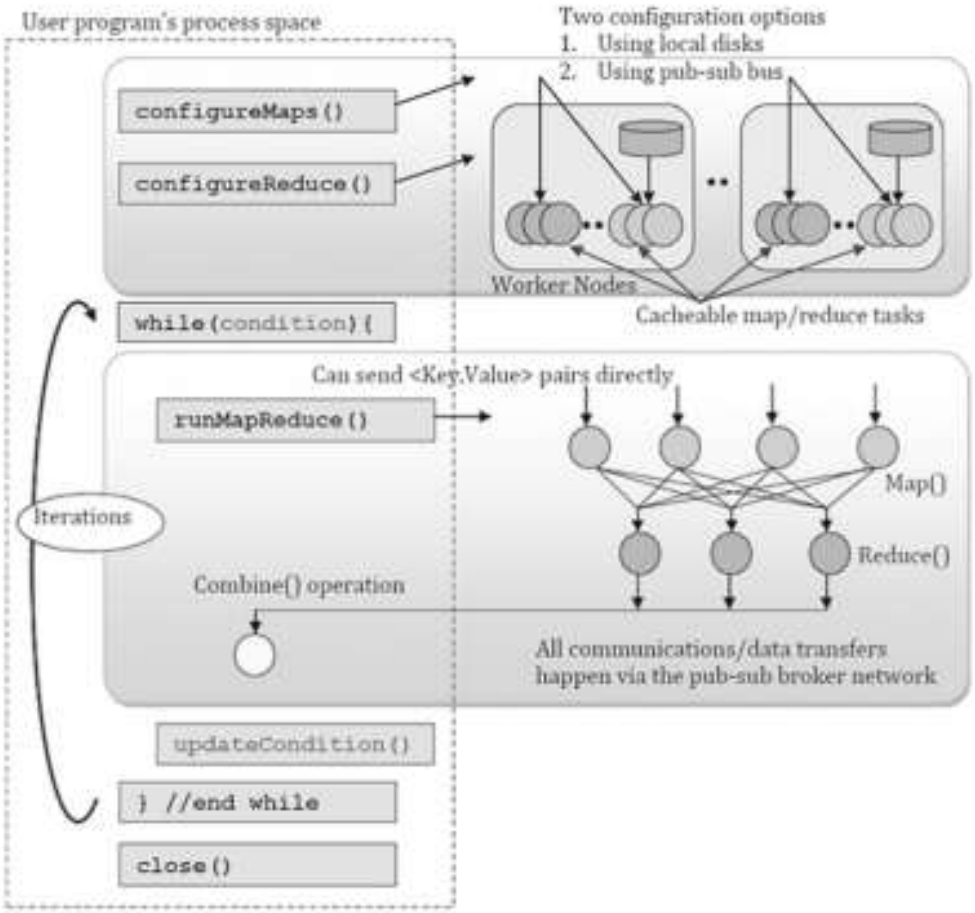

Figure 3: The workflow of Twister programming model

Pros	Cons
Twister proves to be an efficient support for Iterative MapReduce computations (extremely faster than Hadoop or Dryad/DryadLINQ).	In Scheduling Task, Google's MapReduce and Hadoop use a dynamic scheduling mechanism which is more efficient than Twister.
It provides features to support MapReduce computations like distinction on static and variable data.	A possible disadvantage of this approach is that it does require the user to break up large datasets into multiple files.
It combines varied phases to collect all Reduce outputs.	–

Table 4: The pros and cons of Twister and its components

Sector and Sphere

Sector plus Sphere seem to be two crucial criteria for excellent functioning of cloud service. This sector can handle data across dispersed data centers. Likewise, the Sphere offers user-defined procedures (UDFs) upon data both inside and between cloud services. A Location UDF executes basic MapReduce- type programming inside the sphere. Helix as a Dremel runtime system allows distributed data store, distribution, and computation across massive large clusters processors across a given or numerous data centers. It section contains three characteristics, including safe, highly efficient and scalable hdfs. The sphere delivers Sector files upon that storage nodes via easy coding, which runs quicker than Hdfs. But in the opposite hand, Twister as a coding standard significantly supports iterative Roadrunner computations quickly. Twister provides exceptional performance comparing with other comparable implementations like as DryadLINQ but also Hadoop for very huge data parallel workloads.

Table 5 contains the details about various aspects of Sector and Sphere as well as its pros and cons.

Pros	Cons
Sector manages the large distributed datasets with high-reliability, high-performance I/O, and a uniform access. The sphere is used to simplify data access, increase data I/O bandwidth and exploit wide-area, high-performance networks using the sector-distributed storage system. Sphere presents a very simple programming interface by hiding data movement, load balancing, and fault tolerance.	Sector breaks up large datasets into multiple files and uses a device to complete it. Sector assumes that the user to develop code for working with large datasets is as complicated as the user to split a large dataset into multiple files if required.

iMapReduce

iMapReduce seems to be a customized Hadoop command line tool which enables users to describe the recursive computation using the isolated Map reduce functionalities. It can be used to run the repeated processes based on grid environment. The typical properties of repeating methods are obtained in this fashion and it gives the created capability for these aspects. It displays the continual tasks to decrease the job/task startup cost, provides efficient data storage to eliminate the moving of static data around tasks, and permits delayed Map project implementation when it might be feasible. Figure 8 displays the data stream in the Reaper implementation here on left edge. The Mapper should load its input data via DFS even before function begins. Then, it generates the intermediary public keys because of Reducing functional operation upon that data packet and it leads the result of this cycle, which itself is written onto DFS. The identical operation repeats in the following iteration when the Mapper loads the iterative methods data form DFS again.

Additionally, the recurrent DFS does have a costly trying to load. Hadoop offers location optimization that lowers the distant communication. The very same tasks are conducted at each repetition. It indicates that the very same Data transformation functions are executed. iMapReduce employs Map/Reduce jobs permanently. That would be the Plot operations in Chart tasks continuing working until the iterations is done. Further, iMapReduce allows the Dampener result to be transmitted to the Mapping for the following round iteration. This data flow for iMapReduce is displayed on the left. This dashed line shows the data importing from DFS occurs once during the setup phase and the resultant data are saved to DFS exactly once because the iteration finalizes. Moreover, Table 9 offers facts regarding many features like iMapReduce but also its advantages and downsides.

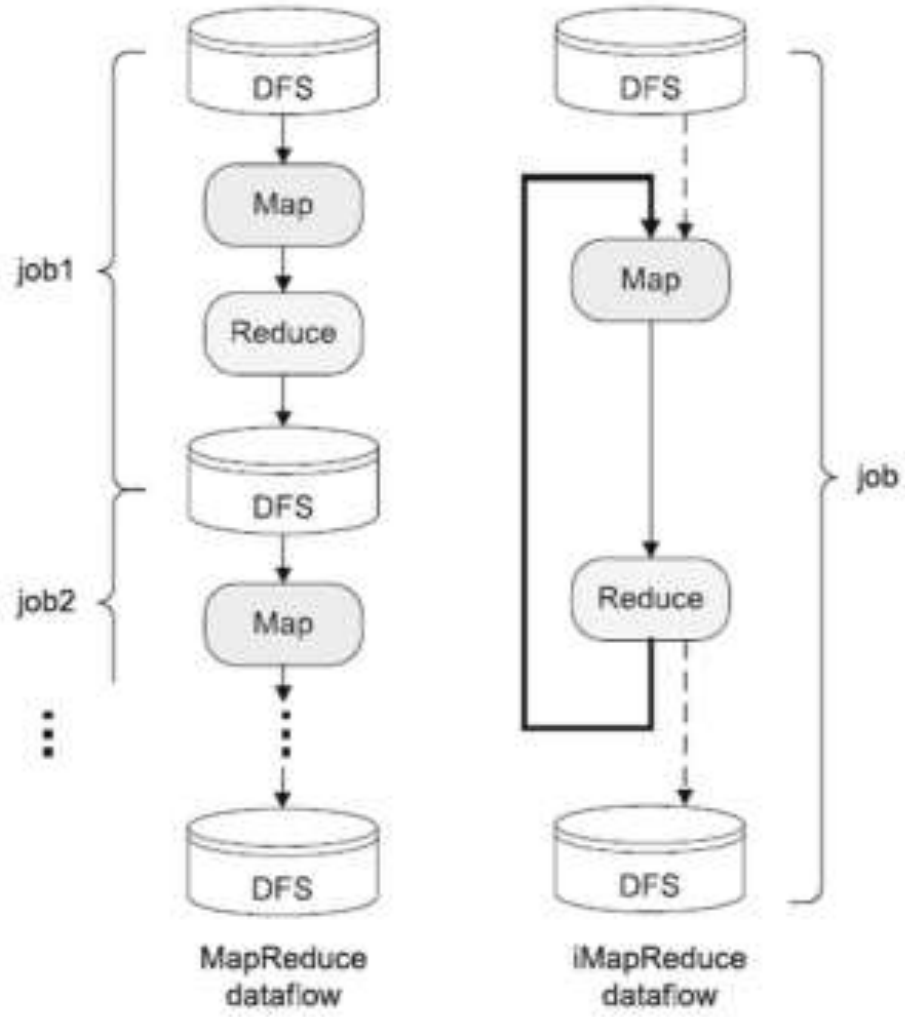

Figure 6: The workflow of MapReduce and iMapReduce

Pros	Cons
iMapReduce results in the context of various applications, show up to 5 times of faster operation compared with traditional Hadoop MapReduce.	iMapReduce is not suitable for more iterative computations.

Table 7: The pros and cons of iMapReduce and its components

MapReduce Applications

The MapReduce program enhances the performance of numerous data parallel activities. The Mapper is the primary factor in very many application areas and it may boost system concurrency. It draws great attention, for information and matrix multiplication applications on machines clusters. It really is utilized as a fast distributed computing tool for varying problems, e.g., searches, clustering, logs, various forms of join procedures, matrix multiplication, template matching, and effects of complex networks; it enables researchers to examine in diverse fields. MapReduce would be used for several more big data implementation such as simple text mining, evolutionary computation, k-means clustering technique, DNA fragment, transportation infrastructure, Healthcare related technologies, Fuzzy inference classification schemes, heterogeneous situations, cuckoo search, recurrent neural network, Random Forest, power proportionally, Smartphone Sensor Readings, web 2.0 and many. Hence, in this part suggests the quick assessments of these implementations and a short glance at different fields of MapReduce implementations.

Distributed Grep

Distributed Regex is a program that looks through plain-text data files for lines that match a predicate. It may search through a vast file with a certain pattern that you provide. A site administrator has to scan the web service logs in line with a certain pattern in order to discover which articles are the most often sought. Therefore order to demonstrate how Layout works within the context of this scenario, Figure 14 provides an illustration of a relatively common case. A line od data is provided to both the Map phase in Figure 14, and then each Mapper takes its own line of material and separates it into phrases. The term and the neutral point are both produced by the Mapper in the form of a (keyword, value) pair. With in Reducer, individual keys are categorized into groups, and the data of keys with comparable characteristics are accumulated. As a result, the aggregation stage for keys is formed by the Transformation function.

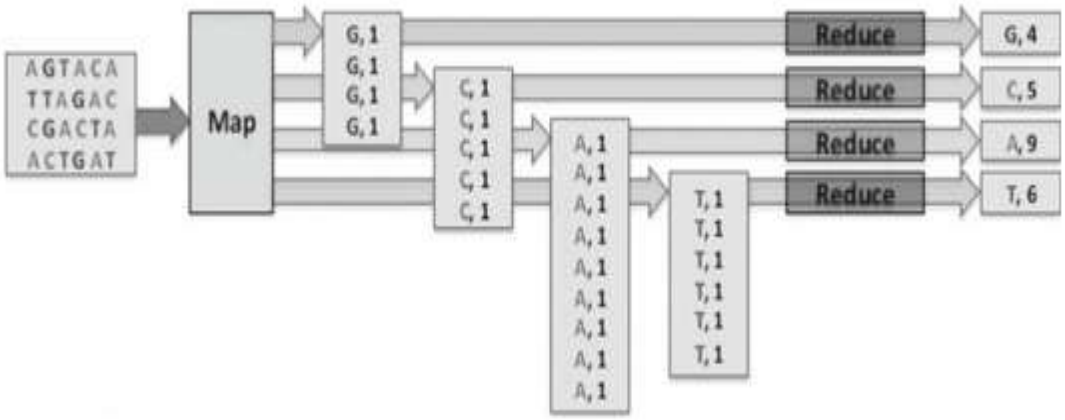

Figure 8: The overall MapReduce distributed Grep process

Word Count

Manuscript is regarded as MapReduce program which tallies the instances of every word in massive textual information and pulls a tiny quantities of information from such a vast dataset. As all the routes are autonomous of one another and, all Mapmakers run concurrently. The word frequency procedure takes done in three ways: Analyzer, Shuffle and just a Retarder. In Mapper, initially the file format is divided into syllables and afterwards make adjustments to name - value pairs using these phrases— the key always being the term itself and values '1'. For example, take the statement "Book Penn Apple Apple Pad Pen Ball Throw Book" in Bitmap the phrase would be separated into words but from the first model . it contains as < Ball, 1>. That after Map process is finished, the bodies of data packet is moved from Morph to Reducer with in shuffle step. The data transmission via shuffling occurs from the Transporter disks but instead of their main memory and the intermediary result will indeed be processed by the keywords to put the pairings with much the same values together. Using Reducer, the keywords are grouped along, and the numbers for comparable keys are merged. Because there are just one pairing of comparable keys 'Book' and values for all these variables would be combined so the result keyvalue pairings would be it would yield the frequency of occurrences of each item in the source and Reducer generates an aggregating phase for keywords and presents the final outcome as Exhibit. 15 clearly illustrates the word frequency process.

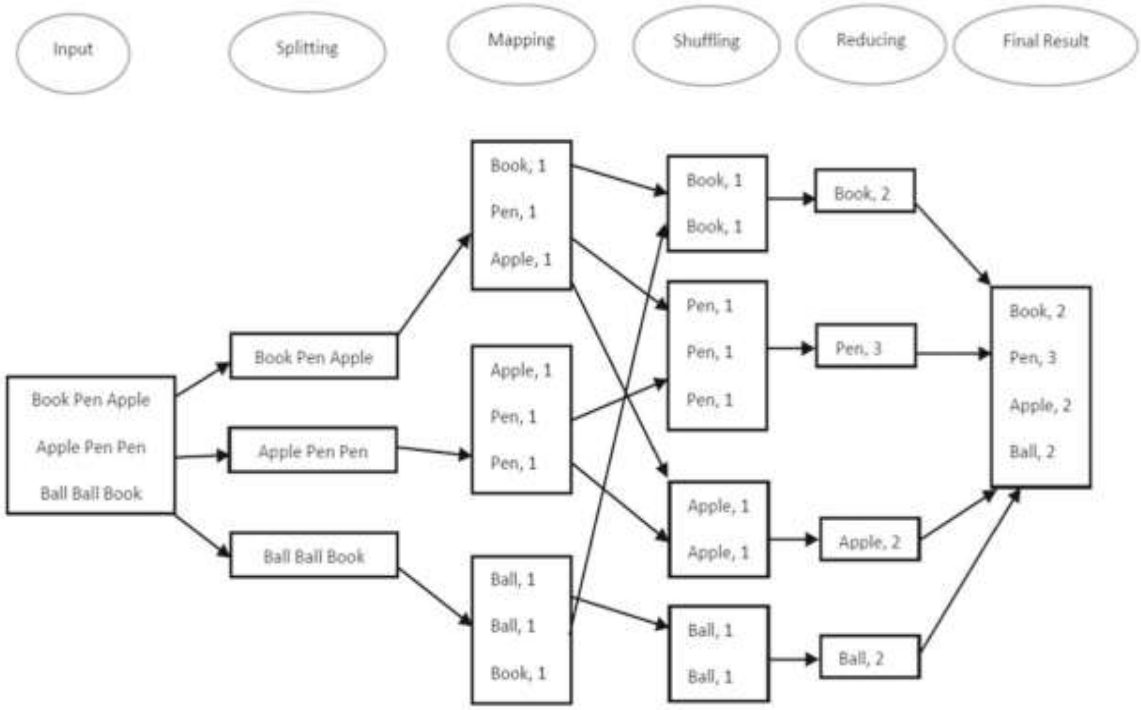

Figure 9: The overall MapReduce word count process

TeraSort

• The production of the data is handled by the Map/Reduce software known as Termagant.

• TeraSort is used to apply in order that is being supplied, and then Plot is used to apply in order together into complete order.

• TeraValidate is indeed a Map/Reduce application that enables the sorted product to be validated and provides this capability.

• Application of biotechnology is a normal Plot type; however, it does not have a custom couple of years ago. A custom mapper employs an ordered list of N-1 sampling keys to specify the key region for each Reduction. TeraSort not have a custom shortened version. To be more explicit, it transmits all of the values, including the samples [i-1]. < =key

Inverted and Ranked Inverted Index

An inversion index that is used to record a mapping across content, like words or integers, to its place in some kind of a data item or a page. An inverted entry may also be used to hold information about the relationship between the two. The input for a Jacquard application is indeed a list, and the output is a word-to-document index. In its most basic form, the Mapper does an analysis on each document and generates a series of pairings. The Reducer takes all possible pairings for a certain word, sorts those document IDs that correspond to those pairs, and then produces a pair consisting of the words "phrase" and "list (article ID)". Use of all the output combinations, a straightforward inverse index is developed. This calculation has to be improved so that it can track down specific word places. 3 The World ranking game is one form of Mapper application. This process took a vocabulary and the number of times those words appear in each document, and then it generates lists those documents that include those words in decreasing order of occurrence. In addition to this, it makes lists of folders that include the supplied words arranged in descending order by frequency based on their occurrences in the supplied word lists together with the percentages of terms with in file called.

Term-Vector

When doing an analysis of a patient's significance in relation to an inquiry, it is helpful to have a definition of the terms that appear the most often in the host. This map phase operation produces tuples in which the recognize the role is also a pair of the type. When the phasing is reduced, the words that have a frequency that is below threshold are deleted. It then sorts the remainder of the lists taking into consideration the number, and it outputs packets of the type.

The following is an example of a phrase vector:

[("word1", 8), ("literary tradition", 4), ("word3", 7)]

Each word vector is represented by a Map task in this example. After that, the Refine phase obtains from many Instances and Map a collection of term arrays for the specified host, which are as follows:

("hostX", [("word1", 8), ("focus of interest", 4), ("word3", 7)])

in comparison to a variety of different categorization techniques. Implementations that are quick, scalable, and simultaneous have a bearing on big information as a result of the methodologies. Dremel is the most effective strategy for achieving this objective. In the context of big data, the use of a Classifier is helpful in handling the approach of handling unbalanced datasets. In particular, oversampling, price learning, and -15

have been applied to massive data by utilizing Lava. As a direct result of this, these methods are equipped to successfully handle datasets that are as vast as is required in order to accurately identify the disadvantaged class. The speed, robustness, and adaptability of the Classification Model make it an ideal candidate and used as the foundation for the comparative.

Spark

Spark is a framework for cluster computing that allows applications that operate with part of the process and has scalability and having various qualities that are comparable to those of MapReduce. By using an abstraction that is referred to as graph database data, it really is utilized to keep MapReduce's scaling and fault - tolerant in check (RDDs). The RDD is just a collection of elements that could only be accessed, and it is partitioned using a series of computers that could be recreated in the event that one of the partitions is deleted. The development of a text set of components that is preserved in memory between iterations and allows fault resolution is the basic concept behind Spark. This concept is referred to as a graph database dataset, or Offering a wide variety for short. Spark offers three different forms of basic data representations for use while programming cluster nodes: distributed data (Database management system (rdbms), broadcast values, and capacitors are the names of the two limited types of sharing variables that are provided. These concepts have the potential to provide certain difficulties due to the limitations they impose on current computation frameworks, such as iterative optimization calculations.

Extreme Learning Machine

Supervised learning machine, often known as ELM, has been gaining traction in a number of different study domains under the MapReduce architecture. Applications are regarded as being among its characteristics owing to the fact that it converges quickly and has strong generalization performance. Several variations of ELM, including basic Sem, ELM that is based on randomised hidden layer given trait, incremental Oak, kernel-based Oak, and others, have been suggested for use in real world applications. Another of the upgraded extreme learning machines algorithms, known as online serial machine learning techniques (OS-ELM), online sequence supervised learning device (OS-ELM) supports online sequence learning well. It investigates the dependence linkages between the matrix computations of OS-ELM and suggests a parallelism OSELM that is built on MapReduce called POS-ELM. On the other hand, it is capable of managing extremely large-scale training datasets when used in applications involving big data. Flexible El is just a novel framework with elastic exceptional learning machines that is built on MapReduce. It was developed to address the scarcity of ELM systems whose capacity for teaching is inadequate for frequently

updated size of the training datasets. The massively rising amount of testing phase in applications that use massive learning causes the central Sem with cores to experience a significant increase in the amount of memory that is used by huge matrix operations. Additionally, it has an important communication cost, therefore prevents certain matrix operations from being directly used to communicated cloud based models like Dremel. This is because of the nature of the distributed software model. An application of ELM that makes use of kernels and Dremel is known as Distributed Positions with respect ELM (DKELM).

DNA Fragment

A Dna includes the genetic codes of a creature. To grasp the organism's construction, it is important to accomplish the whole genetic code and knowing the structural features of a genes. This sequence assembly technique is the cornerstone of Genomic dna data acquisition. Fragment assembling is amongst the most critical challenges in sequence building. Algorithms for Plasmid dna assembly utilizing de Bruijn graphs have been extensively utilized however these algorithms demand a substantial amount of storage and operating time to just be constructed. De Bruijn method suffers from losing information. Three main aspects must be taken to create a features row parallel method, including wanting to avoid diagram division which could also end up causing an adverse effect on transcriptional results, having overcome memory limitations through large sequence legislature by distributing d'une Bruijn chart, as well as putting Tensorflow to be using, is just an optimal solution for features row algorithm. Hadoop separates de Hack tool graph partition and the use of this structure and also building a features row parallel method, which can substantially increase the computation productivity by eliminating the memory restrictions of the useful guidance.

Mobile Sensor Data

Mobile devices are filled with a set of sensors, such as a gyroscope, a strong magnetic gauge, and also a cabin pressure meter. These sensors play a part of obtaining situational data about the person, such as their position, condition, and so forth. Providing the recovered data, inside which data gets withdrawn from mobile devices and stored with in cloud service, is among the major chores that need to be done. For the purposes of teaching and recognizing human actions based on gyroscope sensor information, the exploitation of parallel processing with Dremel in the clouds is used. This method is based via classifier which can readily scale in both efficiency and durability. The purpose of this is to identify and forecast the patterns of human behavior. In order to achieve the training objectives, it is important that information be collected in a manner that is as realistic as feasible. Data acquired in labs do not, by general, represent the patterns of activity that individuals utilize in their everyday lives. In addition, collecting training examples

for these human motion predictive modeling is a challenging process, particularly when it incorporates integrated sensors like the gyroscope found in portable devices. This makes the task more difficult. Smartphones need to have their accelerometer sensors configured in order to combat these issues. It is utilized for the purpose of gathering data in real-world settings and maintaining training repositories, both of which are performed in the internet utilizing distributed processing (the Dremel framework). Every user contributes to the collection of actual accelerometer data, that is then utilized by the algorithms cloud - hosted for the objectives of machine learning algorithms respectively.

Social Networks

The E.g. Facebook Service (SNS), which is enjoying ever-increasing levels of popularity, has also seen an exponential development in the range of statistical research that have been conducted in this dynamic subject. Due to the lightning-fast expansion of the internet, the time-honored approaches to data analysis cannot be used in the modern day. Hadoop Map function is capable of solving the challenge of big complex networks by harnessing the power of several computers, and each is skilled in the handling of enormous amounts of data pertaining to social networks. When doing a series of studies on huge social networks, comprising multiple distributions such as design and radius, Hadoop is used in lieu of several conventional techniques of data analysis. This social network, which may be thought seen as a graphs depicting the ties and interactions that exist among the persons of people, plays an essential part in the dissemination of information, opinions, and influence between all of those members. The present algorithms of the influence global optimum (IMP) for such a social media platform are not effective enough to deal with the social networks that exist in the actual world. This is because the IMP is just so important. It is recommended that MapReduce / Apache be parallelized in order to manage large social networks using Hadoop also as base or alternative Map Reduce implementation.

CONCLUSION:- Analysing academic feedback by an accurate algorithm for pre- processing Academic Feedback Data using big data analytics is very much warranted. The competence of the existing porter's stemming algorithm can be improved by adding dictionary for evaluation. Compared to Lovin's Stemmer, Paice / Husk Stemmer and Dawson stemmer, Porter's stemming algorithm seems to be an unfussy algorithm. To avoid over stem and to improve the competence of the porter's algorithm, a dictionary is used to compare the stemmed words. The stemmed words are again checked in the dictionary and the output is displayed. It was observed that the efficiency of porter's stemming algorithm is improved after adding dictionary for checking the stemmed words. The Under Stemming and Over Stemming Indexes are metrics

of specific errors that occur during the implementation of a stemming algorithm. According to these metrics, a good stemmer should produce as few under stemming and over stemming errors as possible.

A DURKAR framework was developed and tested for analyzing the feedback received from various academic stakeholders. The objective of the proposed framework DURKAR is achieved by the following proposed algorithms:

1. HC-SWA (Hybrid Classifier - SentiWord Analysis)
2. HMP-SWA (Hybrid Map Reduce – Senti Word Analysis)

The Hybrid Classifier HC-SWA is for Senti Word Analysis achieves an accuracy of 83.25% and the Hybrid Map Reduce HMP-SWA is for SentiWord Analysis achieves an accuracy of 98.0 %. These proposed algorithms are experimented with real time data set collected from the schools in and around Tamil Nadu and Pondicherry. These schools run SUITS (School – University – Tie-up – Scheme) programme through IECD, an autonomous skill development institute of Bharathidasan University.

The feedback results with respect to factors such as Overall Support from the University, Quality of Teaching / Learning materials, Communication processes, Training Programme and Examination procedures will help the IECD to take managerial decisions for enhancing the quality of SUITS programme at three levels (Management, Teacher and Students). Feedback was collected from the stakeholders in the academic years 2014 – 2015 and 2015 – 2016 for analysis. The experimental results show that the proposed algorithm and framework performs well and achieved high accuracy.

REFERENCES:-

1. *A.Moturi, C., & K. Maiyo, S. (2012). Use of Mapreduce for Data Mining and Data Optimization on a Web Portal. International Journal of Computer Applications, 56(7), 39–43. https://doi.org/10.5120/8906-2945*
2. *Abdullah M Alghamdi, & Fahad Alghamdi. (2019). Enhancing Performance of Educational Data Using Big Data and Hadoop. International Journal of Applied Engineering Research, 14(19), 3814–3819.*
3. *Ahamad, D., Akhtar, M., & Hameed, S. A. (2019). A review and analysis of big data and mapreduce. International Journal of Advanced Trends in Computer Science and Engineering, 8(1), 1–3. https://doi.org/10.30534/ijatcse/2019/01812019*

4. *Aher, S. B., & Kulkarni, A. R. (2015). Hadoop MapReduce: A Programming Model for Large Scale Data Processing. American Journal of Computer Science and Engineering Survey, 3(1), 1–10. https://pdfs.semanticscholar.org/5cfd/131b43c6369f36396a32e64a209a77e3ec1b.pdf*
5. *Ahmadvand, H., & Goudarzi, M. (2017). Using data variety for efficient progressive big data processing in warehouse-scale computers. IEEE Computer Architecture Letters, 16(2), 166–169. https://doi.org/10.1109/LCA.2016.2636293*
6. *Ahmed, N., Barczak, A. L. C., Susnjak, T., & Rashid, M. A. (2020). A comprehensive performance analysis of Apache Hadoop and Apache Spark for large scale data sets using HiBench. Journal of Big Data, 7(1). https://doi.org/10.1186/s40537-020-00388-5*
7. *Daneshyar, S. (2012a). Evaluation of Data Processing Using MapReduce Framework in Cloud and Stand - Alone Computing. International Journal of Distributed and Parallel Systems, 3(6), 51–63. https://doi.org/10.5121/ijdps.2012.3605*
8. *Daneshyar, S. (2012b). Large-Scale Data Processing Using MapReduce in Cloud Computing Environment. International Journal on Web Service Computing, 3(4), 1–13. https://doi.org/10.5121/ijwsc.2012.3401*
9. *Elsayed, A., Ismail, O., & El-Sharkawi, M. E. (2014). MapReduce: State-of-the-Art and Research Directions. International Journal of Computer and Electrical Engineering, 6(1), 34–39. https://doi.org/10.7763/ijcee.2014.v6.789*
10. *Engineering, C. (2013). Hadoop File System and Fundamental Concept of Mapreduce Interior and Closure Rough Set Approximations. International Journal of Advanced Research in Computer and Communication Engineering, 2(10), 3960–3963.*
11. *Feller, E., Ramakrishnan, L., & Morin, C. (2015). Performance and energy efficiency of big data applications in cloud environments: A Hadoop case study. Journal of Parallel and Distributed Computing, 79–80, 80–89. https://doi.org/10.1016/j.jpdc.2015.01.001*
12. *Jasim Hadi, H., Hameed Shnain, A., Hadishaheed, S., & Haji Ahmad, A. (2015). Big Data and Five V'S Characteristics. International Journal of Advances in Electronics and Computer Science, 2, 2393–2835. https://www.researchgate.net/publication/332230305*
13. *Johnson, A., Havinash, P. ., Paul, V., & Sankaranarayanan, P. . (2015). Big Data Processing Using Hadoop MapReduce Programming Model. International Journal of Computer Science and Information Technologies, 6(1), 127–132. www.ijcsit.com*
14. *K.Madasamy, & M.Ramaswami. (2017). Performance Evalution of Word Frequency Count in Hadoop Environment. International Journal of Innovative Research in Science, Engineering and*

Technology, 6(6), 11. https://doi.org/10.15680/IJIRSET.2017.0606186

15. *http://www.indianjournals.com/ijor.aspx?target=ijor:iitmjmit&volume=9&issue=1&article=004*
16. *Ravichandran, G. (2017). Big Data Processing with Hadoop : A Review. International Research Journal of Engineering and Technology, 448–451. www.irjet.net*
17. *Sajwan, V., & Yadav, V. (2015). MapReduce : Architecture and Internals. 4(5), 2013–2016.*
18. *Santosh Kumar, J., Raghavendra, B. K., Raghavendra, S., & Meenakshi. (2020). Performance evaluation of Map-reduce jar pig hive and spark with machine learning using big data. International Journal of Electrical and Computer Engineering, 10(4), 3811–3818. https://doi.org/10.11591/ijece.v10i4.pp3811-3818*
19. *Selvaramalakshmi, P., Ganesh, S. H., & Tushabe, S. F. (2016). An Improved Performance Evaluation on Large-Scale Data using MapReduce Technique. 5(12), 92–102.*