Cluster Serial Analysis of Data Management System over Enormous Dataset with Map Reduce

Uma Mahesh Kumar Gandham, Dr P Suresh varma

Abstract— Big data if used suitably is able to bring huge benefits to the commerce, knowledge and community. Create data repositories range from terabytes to pet bytes in size. This requires performing deep analysis in excess of huge information repositories poses an important confront to existing statistical software and data management systems. Big data analysis needs synthesis of technique for data with those of machine learning. The design and development of the system contain someway evolved separately for transactional and periodical analytical processing. Such a system-level separation has resulted in troubles such as data originality as well as serious data storage space redundancy. With advent of Big Data analysis propose and to meet this challenge, a new algorithm called CSADMS (Cluster Serial Analysis of Data Management System) has been work mechanism on theory implementation above datasheet with map reduce. The proposed algorithm two different phases namely selection of the Selection of optimal map reduce and algorithm implementation of CSADMS. The algorithm has been tested with the Apache Hadoop.

Index Terms— Mapreduce, data management, Enormous dataset, Cluster, Synthesis, big data, DBMS

1 INTRODUCTION

A penalizing research area such as big data analysis where close teamwork connecting users, application area expert, designer, and software developers is required to understand well-organized system. To developing the cluster based data management system without sacrificing the performance. Large information is a compilation of big datasets that cannot be process by customary compute technique. Difficult of these datasets involve a variety of tools, technique and frameworks to procedure big data relate to information formation, storage space, recovery and examination so as to is amazing in terms of quantity, diversity, and pace. You can study additional about Big Data, Hadoop and Map decrease aptitude. [1]t.

The denote of big data the quantity, font, or cipher on operation be perform by a processor, which may be amass and transmit in the shape of electrical signal and evidence on attractive, optical, or automatic footage medium. Such a information is so big and composite that none of the customary data management tools are able to store it or procedure it efficiently. [6]. Many researchers have been done by the community of the big data for developing the map reduce analysis, Scheme developer will just decide a data layer protocol that suits the target application well rather than designing a totally new data management system.[7].

These algorithms have been tested and developed for the apache hadoop tool and brief storage analysis Adaptive Hadoop Distributed File system. Given the maturity of the available big data and that Enormous Dataset with Map Reduce is still a cornerstone for the Data stock Exchange applications. Hence a new algorithm called CSADMS (Cluster Serial Analysis of Data Management System) which combines the privacy mechanism for making big data in map reduce in arrangement data. This algorithm has been developed for the big data networks employing the multiple cluster links.[4]

CSADMS: It mechanism consists of map shrink theory it is multidisciplinary field which combine the Data organization folder scheme and Adaptive Hadoop Distributed File scheme action length of with the Structured and unstructured data analysis machine learning process. The tool Apache Hadoop is a frame work used to develop data processing application which is execute in a spread computing environment. As Big Data tends to be distributed and unstructured in nature HADOOP clusters (CSADMS) are best suited for analysis of Big Data. Since, it is processing logic (not the actual data) that flows to the HADOOP clusters can easily be scaled to any extent by adding additional cluster nodes, and thus allows for enlargement of Big Data. Also, scale does not need modification to application logic. This idea is called as Real time data region concept which helps increase efficiency of Hadoop based applications

2 RELATED WORK:

Chaokun Wang: et.al [1] Proposed a work belongs to near copy reimbursement many application e,g online news selection over the web by keyword search. The purpose of this demo is show the design and implementation. Hung-chih :[2] Proposed Map-Reduce is a indoctrination replica that enable easy growth of scalable similar application to procedure vast amount of data on large clusters of commodity machines this model facilitates parallel completion of many real-world tasks such as data processing for look for engines and machine learning.

Kyong-Ha Lee: et.al [3] Proposed a famous similar data dis-

- *Research Schalor, AKNU University and working as Asst.Professor in the dept of CSE, GIET college of engineering, rajahmnudry mail id is: umamahesh.gnadam@gmail.com*
- *Professor & dept of Computer Science in AKNU, University College of Engineering Adikavi Nannaya University Rajamahendravaram – 533296 Andhrapradesh , India mailvermaps@yahoo.com*

pensation tool Map Reduce is gaining significant impetus from both manufacturing and academic world as the quantity of data to examine grows rapidly. Robson L. F. Cordeiro at.al :[4] Proposed the Best of both Worlds – BoW method, that automatically spots the block and choose a good plan. Our main aid is: we propose BoW and carefully derive its cost functions, which dynamically choose the best plan.

2.1 Problem Identification

Unstructured data formation due to this imbalanced storage and retrieve problem occurs in existing methodology (i.e.) any data with unidentified form or the arrangement is confidential as unstructured data. In adding to the size being vast, unstructured data poses manifold challenges in conditions of its dispensation for derive value out of it. Typical instance of unstructured data is, a mixed data source contain a mixture of simple text files, images, videos etc. Now a day organization has wealth of data obtainable with them but regrettably they don't know how to derive value out of it as this information is in its uncooked shape or formless arrangement.[6] One of the best examples unstructured data obtained in output returned by Google search. The volume of big data itself is related to a size which is enormous. Size of data theater extremely vital role in formative value out of data. Also, whether an exacting data can in fact be careful as a Big Data or not, is needy upon volume of data. Hence, 'Volume' is one quality which needs to be careful while dealing with 'Big Data'. [10]

3 Proposed Map Reduce Algorithm

Big data is a procedure of any data that can be store, access and process in the form of fixed format is termed as a 'structured' data. In excess of the period of time, aptitude in computer science have achieve better achievement in rising techniques for working with such kind of data (where the format is well known in advance) and also derive value out of it. The worldwide in sequence Map reduce algorithm mechanism as shown in the below figure. The algorithm uses the segregate cluster data from the huge data node and then aggregates the data by using some aggregation algorithms such as middle move toward, namely CSADMS (Cluster Serial Analysis of Data Management System) this aggregated data is to classify in structure manner the normal file system by selecting the efficient path [12].

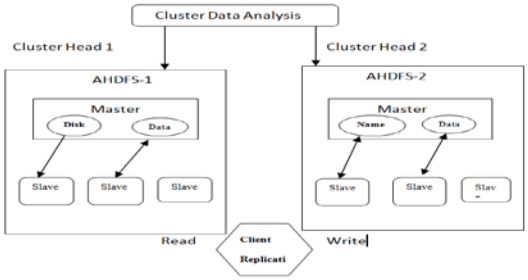

Fig: 1 Architecture of the Map Reduce AHDFS (Adaptive Hadoop Distributed File system) overview

4 Adaptive Methodology for Map Reduceitations

In this research work, the storage efficient of big data is mainly analysis by use of Map Reduce it's most suitable for processing of huge data. In Hadoop is capable of running Map Reduce program return in java language mostly. The performing of large scale data analysis using multiple machines in the cluster. Work is followed by two phases Segmentation Map Phases and Reduces phases. Their works like tree mechanism structure it classify the group of word and split finally merge the words. [13]

Step 1: Mechanism for Segmentation Map Phases for selection

Step 2: Mechanism for Reduces phases

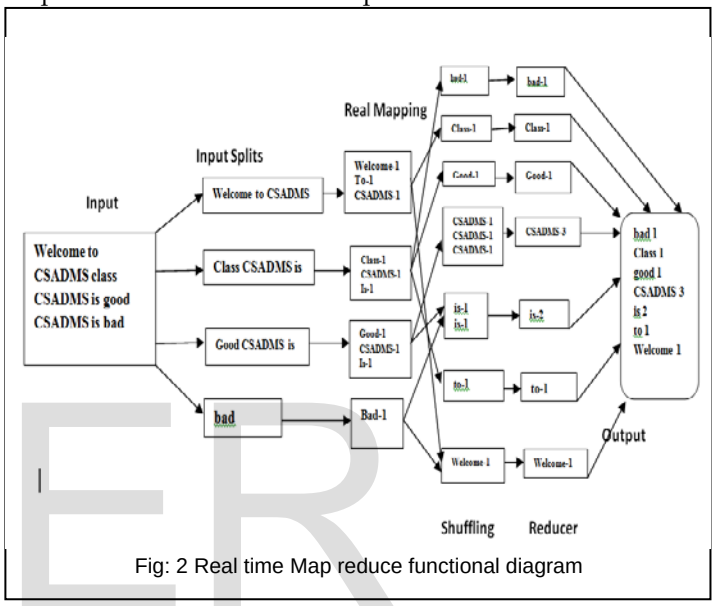

Fig: 2 Real time Map reduce functional diagram

The projected plan of map reduces as per contribution information the group of clustering is classified or split up into input splits i.e... contribution to a chart Reduce job is divided into fixed-size pieces called input splits contribution tear is a chunk of the input that is inspired by a solitary chart Separate the process due to compress the size the overlap data is avoided in this methodology after splitting real mapping is take over in this process This is very first phase in the implementation of map-reduce program. In this stage data in each split is passed to a mapping function to produce production values. Inside our instance, job of map stage is to add up number of occurrence of each utterance from contribution split and get ready a list in the shape of <word, frequency>[17]

Shuffling: This phase consume output of map phase. Its task is to merge the pertinent account as of Mapping phase output. In our example, same words are clubbed jointly along with their respective frequency [15].

Reducing: In this stage, production principles from shuffle stage are aggregate. This stage unites values from shuffle phase and income a solitary production value. In short, this phase summarize the total dataset. In our example, this phase aggregate the values from Shuffling phase i.e., calculates total occurrence of each words [18].

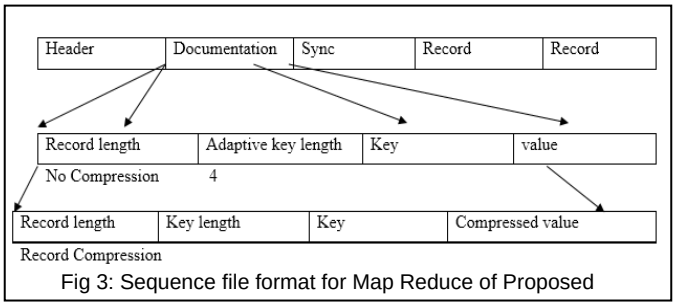

Fig 3: Sequence file format for Map Reduce of Proposed

A series file consists of a header follow by one or more records the first three bytes of a series folder are the bytes SEQ which acts as a the telepathic figure, follow by a solitary byte on behalf of the explanation figure. The slogan contain additional fields, including the person's name of the input and value education, density details, user-defined metadata, [12] and the sync marker. Keep in mind that the sync indicator is second-hand to let a person who reads to go with to a evidence edge from any put in the file. Each file has a arbitrarily make sync marker, whose worth is store in the slogan. Sync marker appears flanked by minutes in the series file. They are intended to incur less than a 1% storage space space in the clouds, so they don't of need appear flank by every pair of minutes (such is the case for short records).[17]

Large Data chart task is wrought for each tear which then execute map cause for each evidence in the tear. It is forever helpful to have manifold splits, since occasion taken to procedure a split is little as contrast to the occasion in use for indulgence of the entire input. When the split are lesser, the dispensation is better load balanced since we are dispensation the splits in similar. Though, it is also not attractive to have splits too little in size. At what time split are too small, the excess of association the split and chart task formation begins to control the total job implementation time.[18]

Future for the majority jobs, it is improved to make split size equal to the size of an HDFS chunk (which is 64 MB, by default).completion of map tasks results into writing output to a local floppy on the own node and not to AHDFS.[21]

Cause for choose local disk over AHDFS is, to avoid duplication which takes place in case of AHDFS store process. Map output is middle output which is procedure by decrease everyday jobs to create the final output.

On one occasion the job is total, the map manufacture can be frightened absent. So, storing it in AHDFS with duplication becomes excess. In the occasion of node failure before the map output is enthused by the decrease job, Hadoop reruns the chart job on one more node and re-creates the map output. Decrease task work on the thought of information area. Output of every map task is fed to the decrease task. Map production is transferred to the machine where reduce task is organization. On this mechanism the output is compound plus then approved to the consumer separate decrease purpose. [22]
Different to the map output, reduce output is store in AHDFS (the first replica is stored on the local node and other replicas are stored on off-rack nodes).

5 Design and Implementation

CSADMS Data honesty in AHDFS: AHDFS clearly checksums all information on document to it and by non-payment verify checksums when understanding data. A separate checksum is wrought for each io.bytes.per.checksum of data. The default is 512 bytes, and since a CRC-32 checksum is 4 bytes long, the storage space in the clouds is less than 1%.Data nodes are responsible for confirm the data they take release of previous to store the information and its checksum. This is pertinent to information that they get release of from customers and as of other information nodes throughout repetition. A client writing data send it to a tube of data nodes (as explained in Chapter 3), and the previous data node in the tube verify the checksum. If it detects an error, the client receives a Checksum exemption, a subclass of IO exemption, which it is theoretical to grip in an application-specific way; for example, by retrying the procedure. [17]

Hadoop Local File System: The Hadoop Local File scheme performs client-side check summing. This income that when you mark a file called filename, the file system client clearly creates a hidden file, .filename.crc, in the same index containing the checksums for each chunk of the file. Like AHDFS, the chunk size is forbidden by the io.bytes.per.checksum possessions, which defaults to 512 bytes. The large piece size is stored as metadata in the .crc file, so the file can be read back properly even if the setting for the chunk size has distorted. Checksums are recognized when the file is read, and if an error is detected, Local File System throws a Checksum exemption.

Pattern conf =...
File System fs = new RawLocalFileSystem();
fs.initialize (null, conf);

The components of the Hadoop CSADMS Design flow is shown in figure 4.

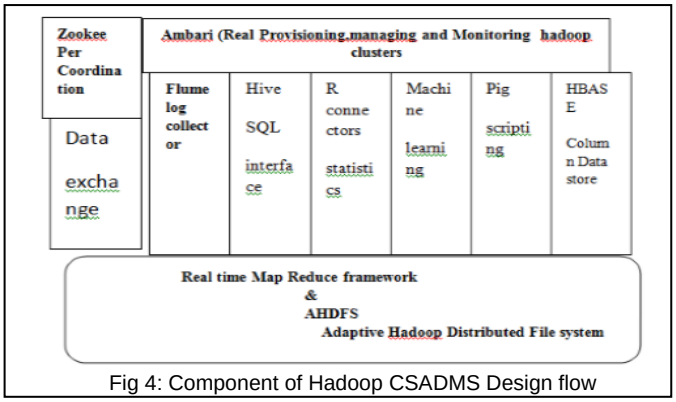

Fig 4: Component of Hadoop CSADMS Design flow

Hadoop MapReduce : Map Reduce is a computational replica and software structure for script application which is run on Hadoop. These Map Reduce program are able of dispensation huge data in similar on large come together of calculation nodes.

AHDFS (Adaptive Hadoop Distributed File System): HDFS takes care of storage space part of Hadoop applications. Map Reduce application put away data from HDFS. HDFS creates manifold replica of information block and distributes them on labor out nodes in come together. This distribution

enables reliable and very rapid computation. [16]

Though Hadoop is best documented for Map Reduce and its dispersed file system- AHDFS, the word is also used for relations of connected project that fall beneath the umbrella of discrete calculate and large-scale data dispensation. Other Hadoop-related project at Apache comprise are Hive, HBase, Mahout, Sqoop , Flume and ZooKeeper. The proposed algorithm is listed below

Algorithm 1:
Initial phase clustering ----
Population initialization:

```
1: map (real const Key&key,/* stud _id*/
2:     real const value& value /*stud_info */{
3:     stud_id = key;
4:      dept_id= value.dept_;
5:    /* calculate information using stud_info*/
6:   output_key = ( dept_id, stud_id);
7:  output_value = (bonous);
8:  Emit (output _key,output _value);
9:    }
```

After in receipt of the contribution split each mapped forms the first inhabitants of persons. Each person is a genetic material of size . Every segment of the material is a Centroid. Cancroids are randomly selected data points from the received data split.

For every data point in each chromosome clustering is performed. For this data tip in the conventional data put assigned to the come together of the closest centroid.

Others are crowded into a dept info "value." One instance inquiry is to join these two datasets and calculate student in order Before these two datasets are joined in a merger, they are first process by a couple of mappers and reducers.[18]

Algorithm 2 second phase:

```
1: map (value const Key, /*dept_id*/
2:        const value &value /*dept_info*/){
3:   dept_id= key;
4:   information_ adjustment  = value.info_adjustment;
5:  Emit ((dept_id), (info_adjus));
6 :}
7: map reduce (const key&key,/*(dept_id,emp_id)*/
8: cons value iterates&value
9: /* an iterate is information collection*/ {
10: info_sum=/*sum of info for each  student*/
11 :}
12 :}
```

Purpose, computer function, divider selector, and configurable iterate. We will use the employee-bonus example to explain the data and control flow of this framework and how these mechanisms collaborate.

The merge purpose (merger) is like map or reduces, in which developers can implement user-defined data processing logic. As a call to a map function (mapper) process
A key/value pair, and a call to a reduce purpose (reducer) processes a key-grouped value compilation, a merger processes two pairs of key/values, that each comes from a visible source. [17]

Covering Clustering: Covering clustering is an algorithm view for joint use with one more clustering algorithm whose straight usage may be not levelheaded due to the large size of a data set. Using covering clustering the data is first partition into overlapping covering using a cheap distance metric. The data is then clustered using more traditional clustering algorithms such as K-means which are extra often than not additional expensive due to more luxurious distance measures. Cover clustering allows these cheap distance comparisons to be made only flanked by member of a particular canopy rather than across all members of the data set. Beneath sensible supposition of the cheap distance measure it is possible to decrease computational time over a traditional cluster algorithm without a important loss in clustering accuracy. Hence, canopy clustering can be cautious a divide-and-conquer heuristic in the sense that it allows an optimal solution to be reach fast which may or may not be the best solution. However, the divide and-conquer characteristic of awning sets. [19]

Steady flow Greedy adaptive Agglomerative Clustering: Agglomerative clustering build the favored clusters from lonely data substance. A Greedy move toward is to unite the two clusters with the most similarity at each step. This process is frequent until either the preferred figure of clusters is achieve or until the resulting clusters all meet some predefined trait. The next figure demonstrates an agglomerative clustering. The progressive merging cluster is shown in figure 5.

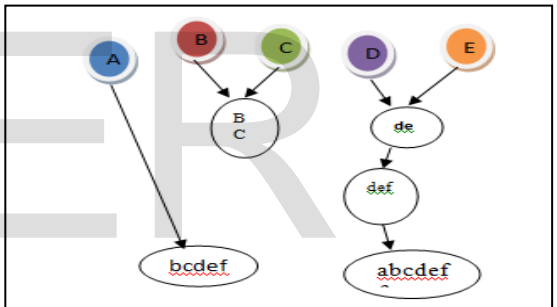

Fig 5: progressive meging cluster

Cluster {a}, {b}, {c}, {d}, {e}, and {f}. The primary pace is to make a decision which article to come together into a cluster. In the insatiable move to the clusters can be complex based on which are closest to each other base on the distance gauge. For example in 11 the first round of amalgamation, the new clusters {b,c} and {d,e} may be formed. This continue until the final cluster {a,b,c,d,e,f} has been created [15]. The flow chart of proposed algoritham is shown in figure 6.

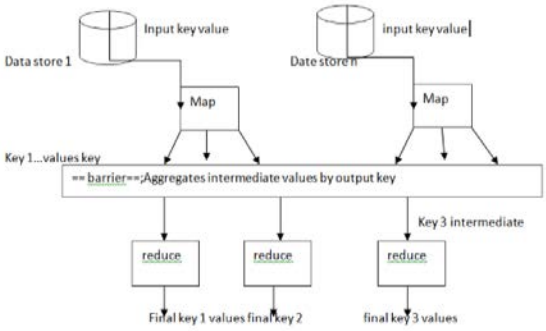

Fig 6: Flow chart of the proposed method

6 SIMULATION RESULTS

Testing Big Data request is additional a corroboration of its data indulgence quite than hard the person skin of the software manufactured goods. When it comes to big data difficult, presentation and functional testing is the key. In Big data difficult QA engineers corroborate the winning treat of terabytes of data by product come together and other helpful mechanism. It stresses a high level of difficult skills as the indulgence is very fast. Dispensation may be of three types. The setup for the implementation is shown in figure 6.

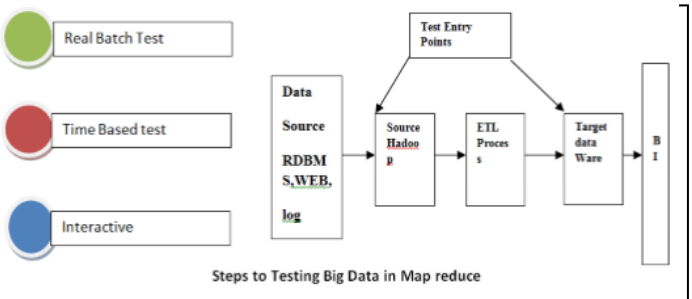

Fig 7: Test setup for big data analysis

- Adaptive Data consumption and Throughout:

In this phase, the tester verifies how the quick system can eat data from a diversity of data source. Difficult engage recognize unlike message that the line can procedure in a given time border. It also include how fast data can be insert into fundamental data store for example placing rate into a Mongo and Cassandra database.[1]

- Real time Data Processing:

It attach verify the speed with which the question or map decrease jobs are carry out. It also includes hard the data treat in division when the basic data store is busy inside the data sets. For example running Map Reduce jobs on the underlying HDFS.

- Sub-Component presentation:

These systems are made up of a variety of mechanism, and it is necessary to test each of these plans in separation. For example, how fast communication is indexed and inspired, map reduces jobs, query appearance, search, etc.

Valuable Architecture Testing

Hadoop process very large quantity of information and is very set aside strong. Hence, architectural hard is very important to make sure achievement of your Big Data system. Poorly or rude prospect scheme may lead to look nastiness, and the system might fail to meet the compulsion. At least, look and Failover test armed forces be supposed to be complete in Hadoop environs. Appearance difficult includes tough of job completion time, memory utilization, data throughput and a like system metrics. As the cause of Failover test repair is to verify that data treat occur completely in case of stop ready of data nodes. [5]

Step 1: Innovative information performance corroboration

The primary step of large data hard also referred as pre-Hadoop stage involves procedure validation. In order as of a variety of basis like RDBMS, weblogs, communal media, etc. be hypothetical to be authenticate to make certain that right data is pull into system difference basis data with the data pushed into the Hadoop system to make sure they match confirm the right data is take out and laden into the right HDFS site Tools like Talend, Datameer, can be second-hand for in order staging corroboration. [6]

Step 2: CSADMS Map Reduce corroboration

The second step is a justification of "Map Reduce". In this phase, the confirm the deal logic corroboration on each node and then authenticate them after companionship against various nodes, make sure that the Map Reduce procedure works correctly Data aggregation or division rules are put into do on the data Key worth pairs are make authenticate the information after Map Reduce procedure .[8]

Step 3: Production Validation stage

The final or third stage of Big Data firm is the production corroboration procedure. The output data records are creation and ready in the way of be inspired to an EDW (Enterprise Data Warehouse) or any other scheme base on the force.

Activities in third stage includes

1. To create certain the modification set of laws are correctly practical
2. To make sure the information truthfulness and engaging data load into the target system
3. To confirm that there is no data dishonesty by evaluate the goal data with the HDFS file system data[9]

New Network Topology in Hadoop

Topology (Arrangement) of the scheme, have an effect on look of the Hadoop cluster when size of the hadoop cluster grow. In adding to the appearance, one also wants to care about the high ease of use and treatment of failure. In order to achieve this Hadoop cluster pattern makes use of network topology typically, network bandwidth is a important factor to think while form any network. However, as gauge bandwidth could be hard, in Hadoop, system is stand for as a hierarchy and coldness flank by nodes of this tree (number of hops) is careful as important factor in the prototype of Hadoop cluster. At this time, coldness between two nodes is equivalent to figure of their coldness to their closest common forebear. [22]

Hadoop cluster consists of data center, the rack and the node which in fact executes jobs. Here, data center consists of racks and rack consists of nodes. Network bandwidth available to processes varies depending upon site of the process. That is, bandwidth available become lesser as we go away from-

- Modus operandi on the like node
- Dissimilar nodes on the parallel rack
- Nodes on nothing like racks of the same data center
- Nodes in dissimilar data middle

Priority based Checksum file system

Limited File System uses Checksum File System to do its labor, and this collection of student makes it easy to add check summing to other (non check summed) file systems, as Checksum File System is just a covering approximately File System. The universal phrase is as follows:

File scheme unprepared Fs =...

File scheme check summed Fs = new Checksum File System(rawFs);

The essential file scheme is call the uncooked file sys-

tem, plus strength be get back by the get RawFileSystem() technique on Checksum file scheme. Check sum File scheme has a small figure of additional useful technique for ready with checksums, such as get Check sum File () for in receiving of the trail of a checksum file for any file. Make sure the guarantee for the others. If an error is detect by Checksum File System when interpretation a file, it will call its repor tCheckSumFailure() technique. The non-payment conclusion does not no matter which, but LocalFileSystem move the criminal file and its checksum to a side directory on the similar machine called bad_files. Administrator should every so often make sure for these dreadful records and get achievement.

Adaptive big data Solidity

Folder width brings two major reimbursements: it decrease the space required to amass files, and it data move obliquely the system or to or from disk. When trade with large volumes of data, both of these savings can be significant, so it pays to cautiously consider how to employ density in Hadoop.

Present are a lot of different thickness format, tools and algorithms, each by means of dissimilar independence. Table lists some of the additional normal ones so as to can be second-hand with Hadoop.

TABLE 1
A SUMMARY OF COMPONENT FORMATS

Compression format	Tool	Algorithm	File name
Real DEFLATE	N/A	DEFLATE	.deflate
View Gzip	Gzip	DEFLATE	.gz
Tar bzip2	*bzip2*	bzip2	*.bz2*
LZO	*Lzop*	LZO	*.lzo*
LZ4	N/A	LZ4	*.lz4*
Snappy	N/A	Snappy	*.snappy*

All density algorithms show a space/time trade-off: earlier density and decompression speed more often than not approach at the price of smaller space investments. The gear scheduled in Table 4-1 typically give some manage over this exchange at hardness time by contribution nine unlike option: –1 means optimize for pace, and -9 means optimize for space. For example, the following power creates a opaque file file.gz using the best density method:

gzip -1 file

Software installation Apache Hadoop tool

Start Hadoop for the time format HDFS system…$HADOOP_HOME/bin/hdfs namenode –format. The Apache Hadoop Tool in Linux Setup is shown in figure 8.

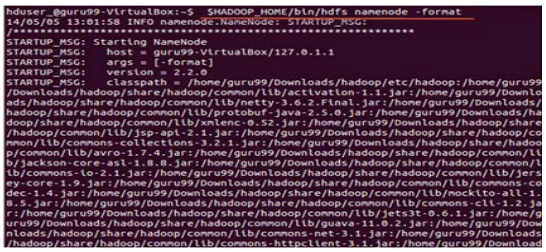

Fig. 8: Apache Hadoop Tool in Linux Setup

The software tools is installed to analysis the big data map reduce factor in different kind of applications...Output analysis of Map Cluster analysis is shown in figure 9

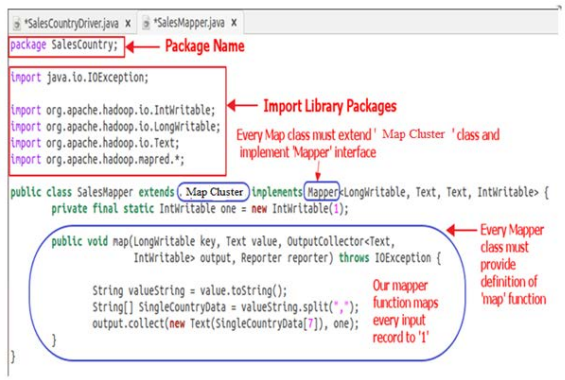

Fig 9: Output analysis of Cluster Mapping

The requirement of name under package of class, sales country is name of package the class meaning extend in Map reduce implements Mapper<LongWritable,Text,Text,IntWritable>

{

public void map(LongWritable key,

Text value,

OutputCollector<Text, IntWritable> output,

Reporter reporter) throws IOException

This is Mapped class in Cluster data analysis in Serial manner by use of Cluster Serial Analysis of Data Management System it must implement in Map per interface. The cluster sample of the group data is shown in figure 10.

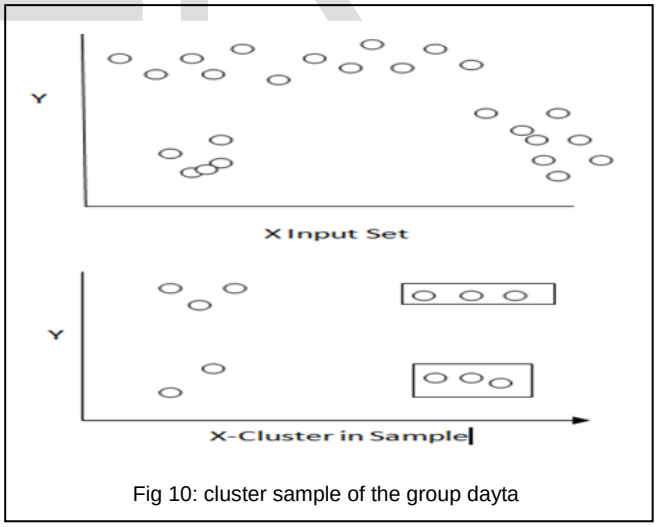

Fig 10: cluster sample of the group dayta

Simulation result: Linear scale a map reducers which act like normal split the disc or node in huge big data classifier come on to the reducers the mechanism which implementation in - Cluster Serial Analysis of Data Management System over Enormous Dataset which ensure the data in the overall file size increase the cluster head is selected and map reduce mechanism take place with high throughput allocations

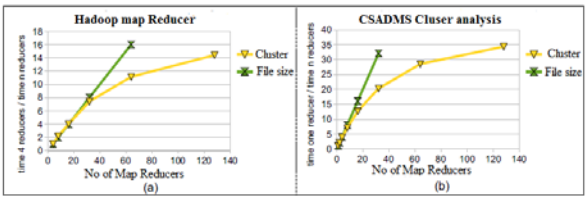

Fig. 10. Generated simulation Result.

7 Conclusion

Map Reduce is a potential explanation to generous out problems between large amounts of data. Mainly for difficulty that can just be divided into autonomous sub everyday jobs that can be solved in similar. Hadoop is an open source MapReduce conclusion feature in this study. Hadoop be hypothetical to be careful only for non-time receptive daily jobs that can be lot process. An instance is sure type of data clustering, such as the data clustering presented in this report. As well-known the most demanding part with scheming MapReduce programs is usually formative inputs, outputs and expresses a difficulty in MapReduce terms. By achieving this fraction by scheming algorithm called Cluster Serial Analysis of Data Management System CSADMS in map reduce methodology over big data.

References

[1] Chaokun Wang "MapDupReducer: Detecting Near Duplicates over Massive Datasets" Copyright 2010 ACM

[2] Hung-chih Yang " Map-Reduce-Merge:Simplified Relational Data Processing On large Clusters" 2007, Beijing, China.

[3] Kyong-Ha Lee" Parallel Data Processing with MapReduce: A Survey December 2011

[4] Robson L. F. Cordeiro " Clustering Very Large Multi-dimensional Dataset With Map Reduce" 2011, San Diego, California, USA.

[5] Sudipto Das" Ricardo: Integrating R and Hadoop" June 6–11, 2010, Indianapolis, Indiana, USA

[6] Yu Cao" Es2: A Cloud Data Storage System for Supporting Both OLTP and OLAP china

[7] Apache. Hadoop. http://lucene.apache.org/hadoop/, 2006.

[8] A. C. Arpaci-Dusseau et al. High-Performance Sorting on Networks of Workstations. In SIGMOD 1997,pages 243–254, 1997.

[9] E. A. Brewer. Combining Systems and Databases: A Search Engine Retrospective. In J. M. Hellerstein and M. Stonebraker, editors, Readings in Database Systems, Fourth Edition, Cambridge, MA, 2005. MIT Press.

[10] L. Chu et al. Optimizing Data Aggregation forCluster-Based Internet Services. In PPOPP, pages 119–130. ACM, 2003.

[11] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In OSDI, pages 137–150, 2004.

[12] F.N. Afrati and J.D. Ullman. Optimizing joins in a map-reduce environment. In Proceedings of the 13th EDBT, pages 99–110, 2010.

[13] A. Ailamaki, D.J. DeWitt, M.D. Hill, and M. Skounakis. Weaving relations for cache performance. The VLDB Journal, pages 169–180, 2001

[14] D. DeWitt and M. Stonebraker. MapReduce: A major step backwards. The Database Column, 1, 2008.

[15] A. Abouzeid et al . HadoopDB: An architectural hybrid of MapReduce and DBMS technologies for analytical workloads. Proceedings of the VLDB Endowment, 2(1):922–933, 2009.

[16] A. Floratou et al . Column-Oriented Storage Techniques for MapReduce. Proceedings of the VLDB, 4(7), 2011.

[17]] A. Matsunaga et al . Cloudblast: Combining mapreduce and virtualization on distributed resources for bioinformatics applications. In Fourth IEEE International Conference on eScience, pages 222–229, 2008.

[18] J. Gray et al. Scientific data management in the coming decade. SIGMOD Record, 34(4):34–41, 2005.

[19] M. Isard et al. Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks. In EuroSys, 2007.

[20] R. L¨ammel. Google's MapReduce Programming Model – Revisited. Draft; Online since 2 January, 2006; 26 pages, 22 Jan. 2006.

[21] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. SIAM J. Sci. Comput., 16(5):1190–1208, 1995

[22] J. Canny. Collaborative filtering with privacy via factor analysis. In SIGIR, pages 238–245, 2002.

[23] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. R. Bradski, A. Y. Ng, and K. Olukotun. Map-Reduce for machine learning on multicore. In NIPS, pages 281–288, 2006.