Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 04, April -2018

# ANALYSING BIGDATA WITH CONTEXT TO HASH PARTITION USING METADATA

[1]Mrs.K.Shruthi, [2]Mr.B.Vijay Kumar

[1]*M. Tech Student, Department of CSE, MallaReddy Engineering College, Maisammaguda, Dhulapally, RangaReddy, Telangana, India.*

[2]*Associate Professor, Department of CSE, MallaReddy Engineering College, Maisammaguda, Dhulapally, RangaReddy, Telangana, India.*

**Abstract** -*Streaming data analysis has pulled in consideration In different applications like monetary records, information investigation, and so forth. Such kind of utilizations require nonstop stockpiling of expansive measure of information in information distribution center while at the same time giving brisk reaction time to the questions against the information that is put away in the framework. The span of getting information shifts relying upon kind of information required from the framework.. This paper presents the performance estimates in terms of MySQL Partition, Hive partition-bucketing and Apache Pig framework. In this paper, big data eco systems and comparative performance analysis of frequently used data retrieval techniques such as MySQL, Hive and Pig are described. From the work introduced in the paper, it is presumed that the execution time for removing information turns out to be vast with development in information estimate, especially if there should arise an occurrence of MySQL. When contrasted with MySQL, Hive and Hive takes less time and give better outcomes.*

***Keywords*** *:- Big Data; Hive; MySQL; Pig; Partitioning; Bucketing; Hadoop framework.*

## I INTRODUCTION

Now days increasing demand of storing a large amount data in digital form is quiet challenging task. In Big information stockpiling, substantial measure of copy information is available. In extensive organizations or huge organizations substantial measure of information is prepared inside seconds. This large amount of data may be in the unstructured form without any format or media. This unstructured data may contain duplicate data used at multiple times so to identify duplicate data and create unstructured data into structured data format is a testing undertaking. To handle this kind of challenging task various authors provided different kind of mechanism like whole file chunking, content defined chunking, and fixed size chunking.

In whole file chunking, whole file is taken as chunk and produces hash values to find Duplicate data. Data may be duplicate within file if whole file chunking is used then duplication can be detected with in files. And to produce hash values for whole file it may take more computation time.

On the other hand, content defined chunking is based on variable size chunking. In this Content defined chunking file is divided into the blocks of the data and then hash values are produced from these blocks to detect duplication id the blocks. To discover indistinguishable lumps or pieces in content characterized lumping system is exceptionally troublesome assignment.

In Fixed size chunking mechanism, file is divided into fixed size chunks and then produces hashes to find fixed size duplicate chunks. In fixed size chunking there are fixed size chunks are created but when there is some changes in data then there may be a problem boundary shift problem.

To handle the duplicate data in big data is a challenging task various authors provided different kind of mechanism like whole file chunking, content defined chunking, and fixed size chunking.

In whole file chunking, whole file is taken as chunk and produces hash values to find Duplicate data. Data may be duplicate within file if whole file chunking is used then duplication can be detected with in files. And to produce hash values for whole file it may take more computation time. On the other hand, content defined chunking is based on variable size chunking.

## II RELATED WORK

Previously a scientist developed a prototype system that is content based file chunking which consist of two subsystems: one is CPU chunking subsystem furthermore, other is GPGPU subsystem. This framework will choose which subsystem would utilize pieces. They analyzed different de-duplication techniques and compared these techniques and concluded that variable size data de-duplication is very efficient from other techniques.

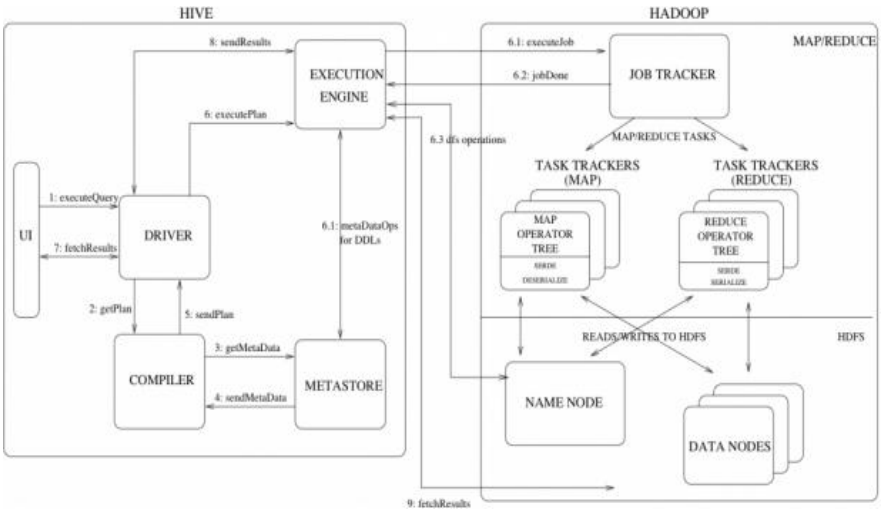
He developed a data reorganize method that is ReDedup it works to address data fragmentation problem and reallocate files and places them on disk. They explained about clustering architecture with several storage nodes for data

de-duplication. In this design, there was an expulsion information repetition at document level and piece level and look at for copy lumps in all hubs in the meantime.

There are distinctive methods for characterizing and contrasting BigData and the customary information, for example, information estimate, substance, accumulation and handling. Huge information has been characterized as substantial informational collections that can't be handled utilizing conventional preparing strategies, for example, BigData is either a social database (Structured, for example, securities exchange information or non-social database (Semistructured or Unstructured, for example, web-based social networking information or DNA informational indexes [6]. The 4V's of BigData are 1) Volume of the information, which implies the information measure. Some of organizations' information is about Zetabyte. 2) Velocity, which implies the rapidly at which the information is created. 3) Varity of the information, which implies the information shapes that distinctive applications manage, for example, grouping information, numeric information or double information. 4) Veracity of the information, which implies the vulnerability of the status of the information or how clear the information is to these applications [4]. Distinctive difficulties in BigData have been talked about in past research [2] and they are portrayed as specialized difficulties, for example, the physical stockpiling, that stores the BigData and lessen the repetition. Likewise, there are numerous difficulties, for example, the way toward removing the data, cleaning information, information incorporation, information accumulation, and information portrayal. Since BigData has these issues, it needs such a domain or structure to work through these difficulties. Hadoop, which works with BigData sets, is a structure that most associations use to process BigData to conquer information challenges.

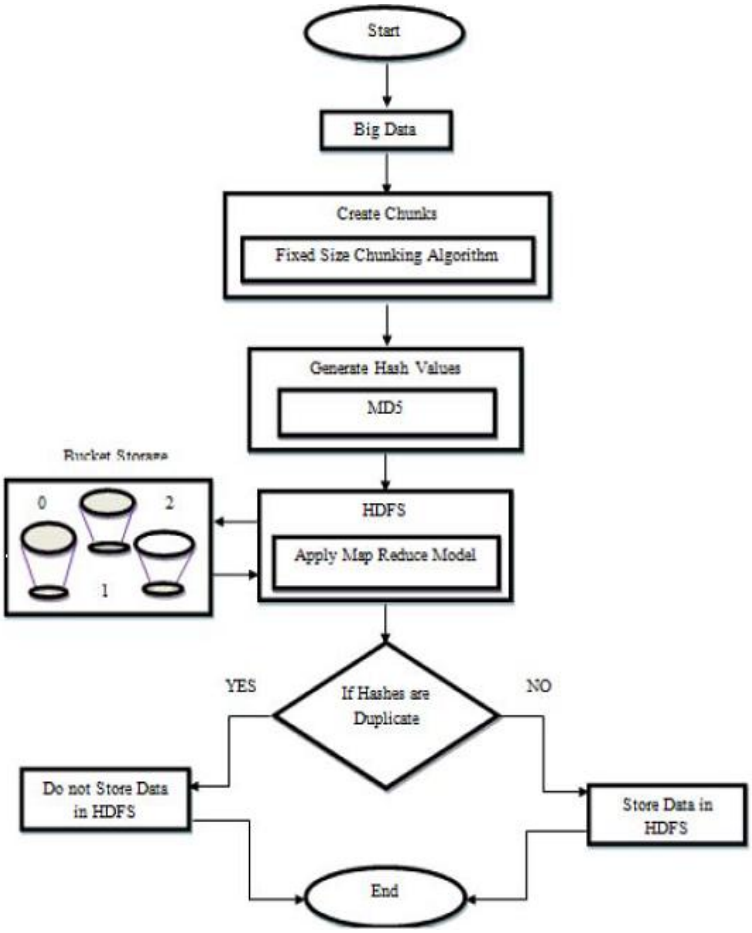
Hadoop is an Apache open-source programming framework that is created in Java for passed on limit and spread taking care of. It gives answers for BigData planning and examination.
It has a record framework that gives an interface between the clients' applications and the neighborhood document framework, which is the Hadoop Distributed File System HDFS. Hadoop disseminated File System guarantees dependable sharing of the assets for effective information examination [1]. The two primary segments of Hadoop are (I) Hadoop Distributed File System (HDFS) that gives the information dependability (circulated capacity) and (ii) MapReduce that gives the framework investigation (appropriated handling) [1] [2]. Contingent upon the decide that "moving computation towards data is more affordable than moving data towards figuring" [2], Hadoop uses HDFS to store immense data. MapReduce gives stream scrutinizing access, runs endeavors on a gathering of centers, and gives a data directing structure to a passed on data accumulating system [3]. MapReduce calculation has been utilized for applications, for example, creating seek records, archive bunching, get to log investigation, and distinctive different sorts of inform exam. "Form once and read-many" is an approach that licenses data records to be created only once in HDFS and subsequently empowers it to be scrutinized numerous conditions over with respect to the amounts of doled out occupations [2]. The pieces are then composed and copied in the HDFS. The squares can be copied various circumstances in light of a particular esteem which is set to 3 times of course [5]. In HDFS, the bunch that Hadoop is introduced in is separated into two principle parts, which are (I) the ace hub called NameNode and (ii) the slaves called DataNodes. In Hadoop cluster, single NameNode is responsible for general organization of the record system including saving the data and directing the occupations to the fitting DataNodes is store related data. DataNodes encourage Hadoop/MapReduce to process the occupations with spilling execution in a parallel preparing condition [1, 4].



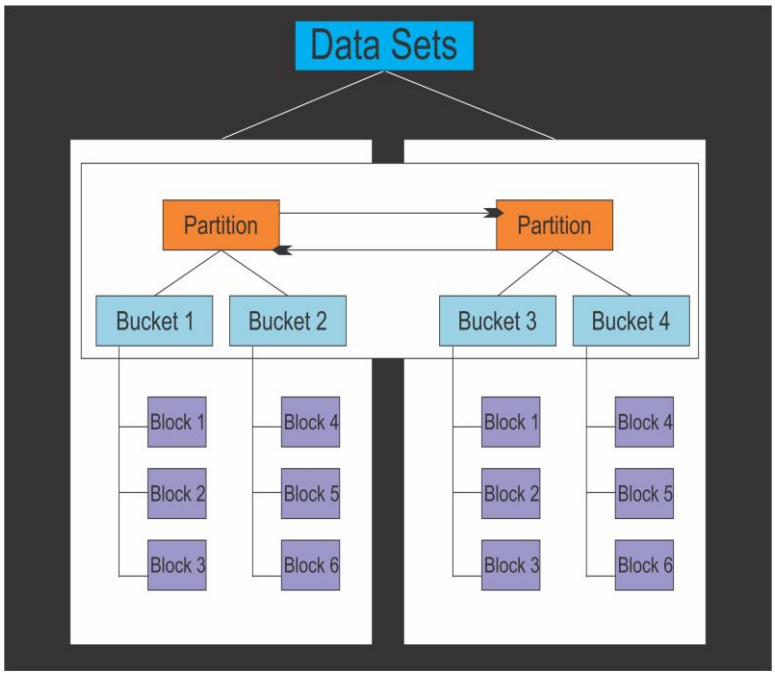

Hive is an information distribution center foundation instrument to process organized information in Hadoop. It dwells over Hadoop to compress Big Data, and makes questioning and investigating simple.

**III DESIGN OF THE WORKFLOW:**




Here first collect real dataset from DATA.GOV. Now divide real data into different chunks. To do this task apply fixed size chunking algorithm. In fixed chunking algorithm initialize the number of chunks and size of chunks is to be generated for example size of 64 MB. It indicates file is divided into various chunks of size 64MB.




Dividing is the way toward figuring out which reducer occasion will get which middle of the road keys and values. Every mapper must decide for the greater part of its yield (key, esteem) sets which reducer will get them. It is vital that for any key, paying little heed to which mapper example produced it, the goal parcel.

## IV METHODOLOGY

Create a dataset
Start working on dataset
Create a partation table based on existing table

p(a)=p(b)<p(a)

Load data from the original table

p(b)⇔p(a)

Then create the schema which is stored in the warehouse

s(x)<-p(b)

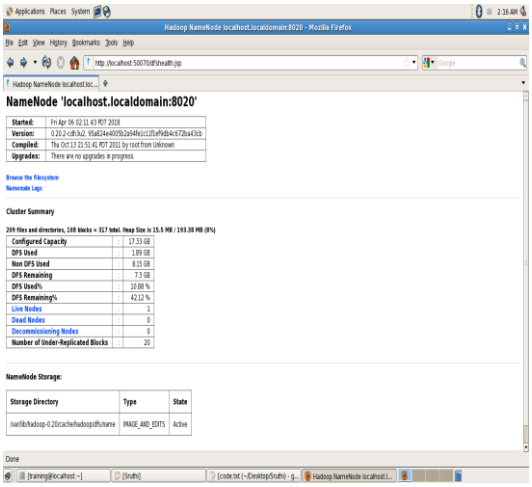
x implies schema from warehouse

Transform the schema based on the requirements

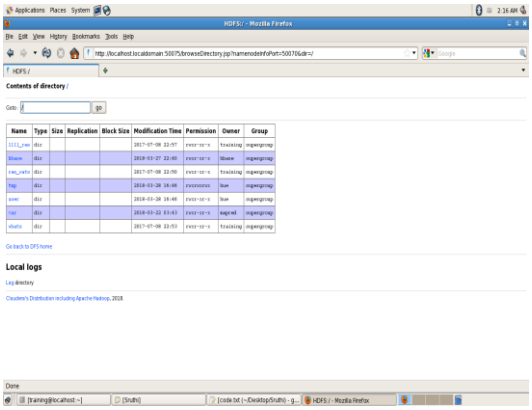
f(x)=s(x)<f(p)

Overwrite the problem function and keep the result in the HDFS file

## V EXPERIMENT RESULTS

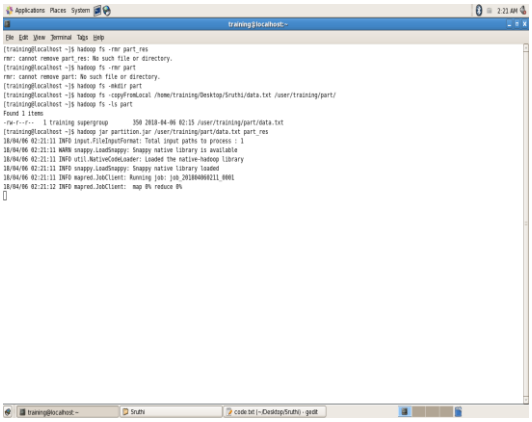



data to hadoop cluster

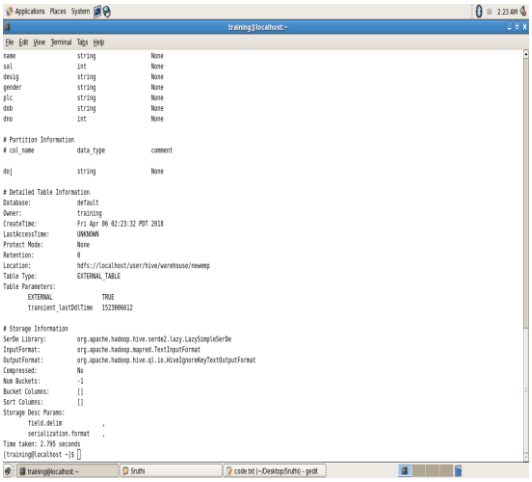



data for partition

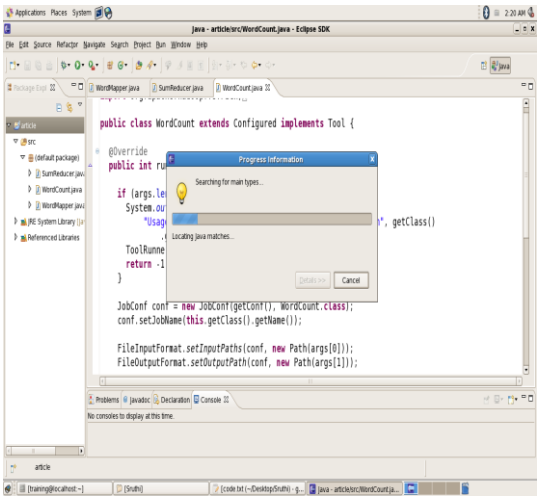
 



Executing the MapReduce Partition



Describing the Hive Partition



Partition of data in Process

## VI CONCLUSION

In big data storage data is too large and efficiently store data is difficult task. To solve this problem Hadoop tool provides HDFS that manages data by maintain duplication of data but this increased duplication. To efficiently storing the data and

deduplicate the data, the solution is bucket based technique. In proposed technique different buckets are used to store data and when same data is accessed by map reduce i.e., already stored in bucket then that data will be discarded so this technique definitely increases efficiency of bigdata storage.

Results shows that in proposed mechanism deduplication ratio is high, data size reduction is high hash time and chunk time is low as compare to existing fixed size chunking technique.

## VII FUTURE ENCHANCEMENT

In future, we will continue working on it and refine results with low computation time also. We propose new mechanism in which all modules are combined like chunking, deduplication and hashing that can find more duplicate content and remove them in proper manner with less time duration.

## VIII REFERENCES

[1] Qinlu He, Zhanhuai Li and Xiao Zhang, "Information Deduplication Techniques", 2010 International Conference on Future arrangement Technology and Management Engineering, IEEE 2010, pp. 430-433.

[2] Won, Lim and Min, " MUCH: Multithreaded Content-Based File Chunking". Journal Transactions on Computers, Journal 2015, pp. 1-6.

[3] Zhi Tang and Youjip Won, "Set out: A Deduplication-Aware Detection and Elimination Scheme for Reduction with Low Overheads", IEEE Transactions on Computers, Journal 2015, pp.1-14.

[4] Bin Lin, Shanshan Li, Xiangke Liao and Jing Zhang, "SecDep: A User-Aware Efficient Fine-Grained Secure Deduplication Scheme with Multi-Level Key Management", Journal 2015, pp. 1-4.

[5] Wen Xia, Hong Jiang, Dan Feng and Lei Tian, "Multithread Content Based File Chunking System in CPU-GPGPU Heterogeneous Architecture", 2011 First International Conference on Data Compression, Communications and Processing, IEEE 2011, pp. 58-64.

[6] E. Manogar and S. Abirami, "A Study on Data Deduplication Techniques for Storage", 2014 Seventh International Conference on Advanced Computing(lCoAC), IEEE 2015, pp.161-176.