Identification of Duplicated Data by using Fixed Size Chunking Algorithm

B. Vijay Kumar Naik[1] Mrs. C. Hemavathy[2]
[1]Student [2]Assistant Professor
[1,2]Department of Computer Applications
[1,2]KMM Institute of PG Studies, Tirupati, India

Abstract— From the past few years, there has been a fast progress within the cloud and big data, with the increasing type of firms using the resources from the cloud, it is important for shielding the information from completely different users, that are exploitation centralized resources. Every second millions of information is being generated because of the use of various new technologies like IOT and device. So it's very troublesome to store and handle such great deal of data. Many enterprise organizations unit investment numerous money to store such huge data for backup and disaster recovery purpose. But ancient backup resolution does not provide any facility of preventing the system from storing duplicate data, that may increase the storage value and backup time that in turn decreases the system performance. Fixed size chunking algorithm in data De-duplication is that the resolution for such disadvantage. It is a replacement rising technique that eliminates the duplicate or redundant data and stores entirely distinctive copy of data. So it reduces the storage utilization and worth of maintaining redundant data.

Key words: Big data, Cloud computing, Data De-duplication, Storage Optimization, Stages in de-duplication

I. INTRODUCTION

In the existing condition due to increasing request of internet empowered gadgets we are getting sizable degree of records. Presently days, records develops speedy from severs social locales and media utilized by humans businesses like Face-book, twitter satellites, Airplanes, stock selling and so on. They all are creating parcel of facts every second and all records put away and related to cloud. Investigation and getting ready of such sort of data is so difficult. It finally ends up hard to manner this large measure of records utilizing close by Database the board apparatuses or normal statistics copying with application.

These days, because of the exponential development being used of growing innovation like dispensed computing and sizeable information, records development price is also increasing quickly. Consistently a big number of data is being created in mild of the usage of diverse new improvements like IOT and Sensor. Henceforth it is extraordinarily checking out to save and address such enormous degree of facts. Numerous Enterprise Institutions are contributing lots of coins to store such big records for reinforcement and for calamity healing cause. In any case, standard reinforcement arrangement does no longer deliver any workplace of retaining the framework from setting away replica statistics, which expands the capability price and reinforcement time, which thusly diminishes the framework execution. Information De-duplication is the solution for such difficulty. It is any other developing method which dispenses with the reproduction or repetitive data and stores simply one in every of a kind duplicate of statistics. Thus it decreases the ability utilization and price of preserving up repetitive facts. Today, endeavour agencies in addition to a standard character need their facts to be remained cautious. Thus they store there facts on severs spots. Putting away information on various spots consequences in high measure of capability utilization. Another problem might be fiasco that can be happened because of feature purpose or counterfeit cause; consequently every person desires their touchy records to be covered and at ease. We cannot think little of such long haul popularity when you consider that sensitive facts are important to be safeguarded. Providing Information on de-duplication for such troubles; it disposes of excess information and simply store considered one of kind facts.

De-duplication is an excellent process that for the maximum element facilities to spare garage room with the aid of expelling repetitive duplicates of statistics. The technique of de-duplication includes dividing of statistics into squares of settled or variable size. At that point supply every lump with a singular hash esteem determined by using MD5 or SHA1. A query manner is then pursued to examine the extra lump by means of contrasting the hash esteems and right now placed away hash esteems. After correlation, if the piece isn't always observed, the lump report is refreshed with the brand new records, else reference is made pointing the present day information. De-duplication may be accomplished on 3 dimensions named as whole document stage, square measurement or byte level.

A. File level de-duplication

In document stage piecing or complete record lumping thinks about an entire document as a piece, in place of breaking files into distinctive portions. In this technique, just a unmarried report is made for the overall document and the equivalent is contrasted and the as of now positioned away entire files. As it makes one report for the entire report, this system shops much less wide variety of document esteems, which thusly spares area and allows save more list esteems contrasted with exclusive methodologies. It continues a strategic distance from maximum extreme metadata question overhead and CPU use. Additionally, it lessens the listing query system and additionally the I/O venture for each piece. In any case, this method comes up brief when a touch little bit of the report is changed. Rather than registering the document for the modified parts, it figures the listing for the complete file and moves it to the reinforcement place. Consequently, it affects the throughput of the de-duplication framework. Particularly for reinforcement frameworks and expansive facts that change mechanically, this system is not appropriate. Whole record is taken into consideration as a solitary lump and the hash an incentive for complete document is produced.

B. Block Level De-duplication

In Block degree de-duplication performs extra fined de-duplication via setting apart each report into squares, evacuating statistics extra at rectangular size.

C. Byte Stage De-duplication

In Byte stage de-duplication analyzes data lumps byte-via-byte and performs assessments for replica portions all the extra exactly.

D. Storage Optimization Techniques

Essential stockpiling is over the top expensive need within the present time of superior world, however ability is maximum crucial for ventures as well as for home clients too. Essential stockpiling is additionally called as degree 1 stockpiling thru which we will keep our records ideally. There are one of a kind enhancement structures given via the vendors, as skinny provisioning, clones, depictions, pressure and de-duplication. Be that as it is able to, amongst these which stockpiling streamlining technique is higher is an inquiry before the IT section. We will see quick advantages and disadvantages of each one of these potential enhancement systems.

1) Compression:

Compression is one of the important stockpiling advancement systems. Pressure is utilized to shop the facts all of the more efficaciously so as the most excessive information can be put away in limited capacity. It is likewise utilized for data transmission streamlining crosswise over machine. Pressure method evacuates the twofold measurement extra facts from the information obstructs with a view to spare storage room. There are again techniques for stress loss and lossless strain. In lossless strain whilst any document is packed its fine will live identical and through decompressing particular record may be gotten all matters considered, but in loss it'll expel the data for all time. Compression does no longer require exertion for reproduction data, it's going to keep the facts no matter duplication of data.

2) Thin Provisioning:

It is the technique applied for adequately dispensing the garage room to spare the data. It facilities round apportioning circle area for all various depended clients. For that it's going to do not forget the base necessity of ability want of every purchaser. Thin provisioning works in shared capability circumstance wherein it will distribute the fact square progressively at something point there is want of ability or out of capability circumstance emerges i.e. It's miles unadulterated on interest system. It maintains up the pool of unfastened space in light of which it's going to accomplish higher capacity use share. In normal provisioning for each application it will allot a few extra measure of potential limit, which cannot be utilized by a few different utility. Subsequently extra frequently than not it results in wastage of bodily reminiscence pointlessly .But inside the event of skinny provisioning it's going to evacuate such extra paid stockpiling limit and assign correct degree of capability required. If there should be an occurrence of more garage room require it'll gradually introduced to the present day joined capability framework.

3) Snapshot:

Snapshot innovation is one of the common information stockpiling advancements these days. Previews are perused just duplicates of facts which can be treasured for information protection as well as for replication. Most sellers make use of this innovation at running framework to get to records at application layer correctly. Preview implies at some random time of length recording situation of the ability devices. Thus it very well may be moreover recuperated on the season of unhappiness. There are special methods to actualize preview innovation counting on dealers and situation approach may be resolved. A few sellers can also make use of examine simply depiction and a few may make use of writable preview innovation. Duplicate on-compose, Redirect-on-compose, Split mirror those are a part of the strategies in depiction innovation. In which replica in compose takes preview of metadata of the first information where divert on compose, composes just modified statistics rather than taking replica of finish unique facts. In spite of the truth that it gives statistics protection execution can be problem on this innovation.

4) Clones:

Clones and previews are to a few diploma may seem to be comparative and may confound merchants however there's comparison in them. Cloning implies making a correct comparative replica of the digital device where depiction makes a delta file, which permits you to rollback a digital system. Previews and clones are comparative in nature yet both have numerous characteristics and approach of employments. Clone VM is a unique of generation VM, which include IP deal with, DNS name, and so forth. Depiction is a "rapid go back" spotlight, if something turns out badly. In any case, notwithstanding the whole thing you require high-quality antique reinforcements on different capability.

Over all stockpiling streamlining strategies make use of one of a kind techniques to productively save vast degree of information in confined circle area quite simply and less stockpiling prerequisite. Be that as it could, above systems don't deal with the reproduction statistics; it's going to save the extra statistics as it's miles which hence require more storage room. Thus information de-duplication is applied.

E. Data deduplication

Information de-duplication is some other rising manner which is applied to take out excess records and shops just one of a type duplicate of the data. Consequently it will likely be in price of higher stockpiling usage and effective gadget to deal with comparative facts. For example, count on there may be one e mail framework in which there are a hundred prevalence of precise 1 MB file connection. At the point whilst the e-mail framework is subsidized up without de-duplication it'll require 100MB of ability. Be that as it may, if de-duplication is hooked up on email framework, only a unmarried event can be put away in the beginning and after that consequent occurrences might be given reference pointer to the first case spared. In such a manner, the interest for storage room is decreased from 100MB to 1MB. There are distinct Stages in De-duplication:

1) The lumping/blocking approach isolates the widespread facts report into little pieces called lumps or squares.

2) Hash calculation is attached to create an interesting hash identifier of each datum square.
3) At the point whilst new statistics square wishes reinforcement, it is going to be contrasted and as of now put away hash identifier.
4) On the off threat that coordinate located, reference pointer will be given and replica data square is erased, else it will save the excellent identifier and information rectangular.

The factor in time at which the de-duplication calculation is achieved is referred to as the De-duplication timing. The planning of the calculation dependably positioned a tremendous trouble on how a great deal time it desires to carry out facts de-duplication and the dimension of mastering the calculation reflect on consideration on the brand new file information. Regarding the making plans, the de-duplication scenario may be characterized into: disconnected de-duplication and on-line de-duplication.

F. Offline De-duplication

This situation will become a critical aspect whilst the information de-duplication is completed disconnected. For this situation, all facts is first composed into the capacity framework first and de-duplication is achieved later. The best gain of this advent is that after the de-duplication technique is continued, the framework has a static angle of the complete report framework and has a complete getting to know quite an awful lot every one of the information it strategies and may virtually decorate the de-duplication talent. Yet, the execution can be minimal mild for the reason that it might evaluation the report information with all facts put away at the plate. Once extra, the facts kept in touch with the potential framework must be bunched till the following planned de-duplication time. This makes an un-dismissible postponement among while the statistics is composed and when area is reproduced via expelling copy facts.

G. Online De-duplication

When contrasted with Offline statistics de-duplication, the web one is done because the information is being composed to plate. The number one favourable role of this situation is this takes under consideration set off space recuperation. Be that as it is able to, there may be an expansion in compose inertness since the compose is obstructed until all excess record records is worn out.

II. LITERATURE SURVEY

Darrell D. E. Long [1] In this we use File level de-duplication, Extreme binning uses file similarity. First, it chooses minimum hash index value of particular file as its characteristic fingerprint using border's Theory. Then it transfers the files to the same de duplication server to de-duplicate. here we generate single hash value for entire file sometimes the hash functions consider spaces also at that time it gives different values for same data.

Guanlin Lu [2] In this we Used metadata information of different levels in the I/O path such that more Meaningful data Chunks can be generated in the process of file partitioning in order to achieve interfile level de-duplication.

Ravinder Singh [3] In this we use Fixed size chunking algorithm. If file size 1GB, then name node will apply create chunk of files and transfer chunks to the secondary data nodes where de-duplication is performed.

Qing Liu [4] in this They have used Map-reduce technique for parallel de-duplication framework. Index table is distributed in each node which is stored in lightweight local MySQL databases.

Shengmei Luo [5] In this we use Block level chunking(Super chunk) algorithm. It uses efficient data routing algorithm which is based on data similarity, hence reduces the network overhead to identify target storage location. It uses multiple storage data node for parallel de-duplication.

Naresh Kumar [6] Buckets are used to store the Hash value of blocks. Map reduce technique applied to compare hashes stored in bucket with incoming hash of block.

III. PROPOSED ALGORITHM

A. Fixed-Size Chunking Algorithm:

Fixed size chunk approach components files into similarly measured pieces. The piece limits rely on balances like 4,8,16kb, and so forth. This approach effectively comprehend troubles of the report degree piecing method: If a huge record is modified in just multiple bytes, simply the modified lumps must be re indexed and moved to the reinforcement location. Be that as it could, this method makes more lumps for larger record which calls for extra room to store the metadata and the ideal opportunity for question of metadata is more. As it elements the report into settled length, byte shifting issue occurs for the changed record. On the off threat that the bytes are embedded or erased on the record, it adjustments all ensuing piece function which leads to replica file esteems. Hash impact is probably going to arise on piecing method by way of making identical hash an incentive for numerous lumps. This may be worn out by using a tiny bit at a time exam which is steadily unique, yet requires more opportunity to take a look at the statistics.

1) Step 1: start
2) Step 2: read the entire file
3) Step 3: divide the entire file into blocks based o length
4) Step 4: find the hash values for each block
5) Step 5: repeat the for loop for entire table, if the same hash value is exist then delete that block if not exist store that block of data. This procedure is repeated for all the blocks of file
6) Step 6: end

IV. RESULT AND ANALYSIS

```
user@node:~$ start.all.sh
this script is depressed. Instead use start-dfs.sh and start-yarn.sh
19/01/22 15:20:04 warn util.NativeCodeLader:unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
staring namenodes on[localhost]
localhost: namenode running as process 2626. Stop it first.
localhost: datanode running as process 2752. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0 secondary namenodes running as process 2969. start it first
19/01/22 15:20:09 WARN util.NativeCodeLoader: unable to load native-hadoop library
for your platform...using builtin-java classes where applicable
starting tarn deamons
resourcemanager running as process 3216. stop it first.
localhost: nodemanager running as process 3257. stop it first.
user@node:~$ cd Desktop
```



```
user@node:~$ start.all.sh
this script is depressed. Instead use start-dfs.sh and start-yarn.sh
19/01/22 15:20:04 warn util.NativeCodeLader:unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
staring namenodes on[localhost]
localhost: namenode running as process 2626. Stop it first.
localhost: datanode running as process 2752. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0 secondary namenodes running as process 2969. start it first
19/01/22 15:20:09 WARN util.NativeCodeLoader: unable to load native-hadoop library
for your platform...using builtin-java classes where applicable
starting tarn deamons
resourcemanager running as process 3216. stop it first.
localhost: nodemanager running as process 3257. stop it first.
user@node:~$ cd Desktop
user@node:~/Desktop$ hadoop fs -put de.csv/user/
```

```
user@node:~$ start.all.sh
this script is depressed. Instead use start-dfs.sh and start-yarn.sh
19/01/22 15:20:04 warn util.NativeCodeLader:unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
staring namenodes on[localhost]
localhost: namenode running as process 2626. Stop it first.
localhost: datanode running as process 2752. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0 secondary namenodes running as process 2969. start it first
19/01/22 15:20:09 WARN util.NativeCodeLoader: unable to load native-hadoop library
for your platform...using builtin-java classes where applicable
starting tarn deamons
resourcemanager running as process 3216. stop it first.
localhost: nodemanager running as process 3257. stop it first.
user@node:~$ cd Desktop
user@node:~/Desktop$ hadoop fs -put de.csv/user/
19/01/22 15:21:36 WARN util.NativeCodeLoader: unable to load native-hadoop library
for your platform...using builtin-java classes where applicable
user@node:~/Desktop$ hadoop fs -put de.txt/user/
```

```
user@node:~$ start.all.sh
this script is depressed. Instead use start-dfs.sh and start-yarn.sh
19/01/22 15:20:04 warn util.NativeCodeLader:unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
staring namenodes on[localhost]
localhost: namenode running as process 2626. Stop it first.
localhost: datanode running as process 2752. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0 secondary namenodes running as process 2969. start it first
19/01/22 15:20:09 WARN util.NativeCodeLoader: unable to load native-hadoop library
for your platform...using builtin-java classes where applicable
starting tarn deamons
resourcemanager running as process 3216. stop it first.
localhost: nodemanager running as process 3257. stop it first.
user@node:~$ cd Desktop
user@node:~/Desktop$ hadoop fs -put de.csv/user/
19/01/22 15:21:36 WARN util.NativeCodeLoader: unable to load native-hadoop library
for your platform...using builtin-java classes where applicable
user@node:~/Desktop$ hadoop fs -put de.txt/user/
19/01/22 15:21:59 WARN util.NativeCodeLoader: unable to load native-hadoop library
for your platform...using builtin-java classes where applicable
user@node:~/Desktop$ hadoop fs -ls /user/
```

```
drwxr-xr-x   - user supergroup          0 2015-11-30 04:42 /user/flume
-rw-r--r--   1 user supergroup      95718 2019-01-21 15:21 /user/itinerary.csv
-rw-r--r--   1 user supergroup        537 2019-01-17 10:40 /user/kmeans.csv
-rw-r--r--   1 user supergroup       4900 2019-01-12 10:53 /user/kmeans1.csv
drwxr-xr-x   - user supergroup          0 2019-01-22 12:53 /user/output
-rw-r--r--   1 user supergroup      28372 2019-01-17 15:37 /user/pattern.csv
drwxr-xr-x   - user supergroup          0 2015-11-30 04:06 /user/sample
-rw-r--r--   1 user supergroup      62716 2018-12-27 16:17 /user/stu.csv
drwxr-xr-x   - user supergroup          0 2019-01-19 14:56 /user/te
-rw-r--r--   1 user supergroup    8582428 2019-01-03 17:53 /user/te.csv
-rw-r--r--   1 user supergroup   16987816 2019-01-03 10:29 /user/tee.csv
drwxr-xr-x   - user supergroup          0 2019-01-19 10:53 /user/text
-rw-r--r--   1 user supergroup        718 2019-01-22 12:33 /user/text.txt
drwxr-xr-x   - user supergroup          0 2019-01-19 11:44 /user/text1
-rw-r--r--   1 user supergroup       5389 2019-01-22 12:33 /user/text1.txt
drwxr-xr-x   - user supergroup          0 2019-01-19 11:45 /user/text3
drwxr-xr-x   - user supergroup          0 2019-01-19 11:45 /user/text4
drwxr-xr-x   - user supergroup          0 2019-01-19 10:38 /user/textfiles
-rw-r--r--   1 user supergroup      10760 2019-01-03 11:19 /user/tte.csv
-rw-r--r--   1 user supergroup    8293954 2019-01-08 10:43 /user/ttt.csv
-rw-r--r--   1 user supergroup        142 2018-12-26 15:59 /user/w.csv
-rw-r--r--   1 user supergroup     364902 2018-12-27 15:26 /user/w1.csv
-rw-r--r--   1 user supergroup    3115249 2018-12-21 18:19 /user/worker.csv
user@node:~/Desktop$ hadoop jar de.jar com.driver /user/de.csv /user/out
```

```
user@node: ~/Desktop
-rw-r--r--   1 user supergroup     364902 2018-12-27 15:26 /user/w1.csv
-rw-r--r--   1 user supergroup    3115249 2018-12-21 18:19 /user/worker.csv
user@node:~/Desktop$ hadoop jar de.jar com.driver /user/de.csv /user/out
19/01/22 15:48:26 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
19/01/22 15:48:27 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0
:8032
19/01/22 15:48:27 WARN mapreduce.JobSubmitter: Hadoop command-line option parsin
g not performed. Implement the Tool interface and execute your application with
ToolRunner to remedy this.
19/01/22 15:48:27 INFO input.FileInputFormat: Total input paths to process : 1
19/01/22 15:48:27 INFO mapreduce.JobSubmitter: number of splits:1
19/01/22 15:48:27 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_15
48131987901_0013
19/01/22 15:48:28 INFO impl.YarnClientImpl: Submitted application application_15
48131987901_0013
19/01/22 15:48:28 INFO mapreduce.Job: The url to track the job: http://node:8088
/proxy/application_1548131987901_0013/
19/01/22 15:48:28 INFO mapreduce.Job: Running job: job_1548131987901_0013
19/01/22 15:48:33 INFO mapreduce.Job: Job job_1548131987901_0013 running in uber
 mode : false
19/01/22 15:48:33 INFO mapreduce.Job:  map 0% reduce 0%
19/01/22 15:48:40 INFO mapreduce.Job:  map 100% reduce 0%
```

```
42,MALE,white
42.FEMALE,white
43,FEMALE,white
44,FEMALE,white
43,MALE,white
44,MALE,white
```

```
user@node: ~/Desktop
                Reduce input records=400
                Reduce output records=400
                Spilled Records=800
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=61
                CPU time spent (ms)=1250
                Physical memory (bytes) snapshot=434647040
                Virtual memory (bytes) snapshot=1701335040
                Total committed heap usage (bytes)=294649856
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=6016
        File Output Format Counters
                Bytes Written=7941
K
user@node:~/Desktop$ hadoop jar de.jar com.driver /user/de.txt /user/out1
```

Clustering techniques have been widely adopted in many real world data analysis applications, such as customer behavior analysis, targeted marketing, digital forensics, etc. With the explosion of data in today's big data era, a major trend to handle a clustering over large-scale datasets is outsourcing it to public cloud platforms. This is because cloud computing offers not only reliable services with performance guarantees, but also savings on in-house IT infrastructures. However, as datasets used for clustering may contain sensitive information, e.g., patient health information, commercial data, and behavioral data, etc, directly outsourcing them to public cloud servers inevitably raise privacy concerns.
we propose a practical privacy-preserving Kmeans clustering scheme that can be efficiently outsourced to cloud servers. Our scheme allows cloud servers to perform clustering directly over encrypted datasets, while achieving comparable computational complexity and accuracy compared with clusterings over unencrypted ones. We also investigate secure integration of MapReduce into our scheme, which makes our scheme extremely suitable for cloud computing environment.Thorough security analysis and numerical analysis carry out the performance of our scheme in terms of security and efficiency.Experimental evaluation over a 5 million objects dataset further validates the practical performance of our scheme.

V. CONCLUSION AND FUTURE SCOPE

Every second millions of information is being generated due to the utilization of assorted new technologies like IOT and sensor. Then it's really troublesome to store and handle such large amount of data. Many enterprise organizations are finance lots of money to store such huge information for backup and disaster recovery purpose but ancient backup resolution does not offer any facility of preventing the system from storing duplicate information, that may increase the storage worth and backup time that in turn decreases the system performance. Fixed size chunking algorithm in data de-duplication is that the resolution for such draw back. It's a replacement rising technique that eliminates the duplicate or redundant info and stores exclusively distinctive copy of information. Hence it reduces the storage utilization and worth of maintaining redundant information.

REFERENCES

[1] Deepavali Bhagwat ,Kave Eshghi,Darrell D. E. Long," Extreme Binning: Scalable, Parallel Deduplication for Chunk-based File Backup", IEEE 20O9.
[2] C. Liu, Y. Lu, C. Shi, et al.,"ADMAD: Application-driven metadata aware deduplication archival storage System",in Proc. 5th IEEE Int. Workshop Storage Netw. Archit. Parallel I/Os, 2008,
[3] Shamsher Singh ,Ravinder Singh," Next Level Approach of Data Deduplication in the Era of Big DatA", IEEE 2017.
[4] Qing Liu, Yinjin Fu, Guiqiang Ni," Hadoop Based Scalable Cluster Deduplication for Big Data",IEEE 2016.

[5] Shengmei Luo, Guangyan Zhang, Chengwen Wu," Boafft: Distributed Deduplication for Big Data Storage in the Cloud", IEEE 2015.
[6] Naresh Kumar , Rahul Rawat , S. C. Jain." Bucket Based Data Deduplication Technique for Big Data Storage System", IEEE 2016.

