International Journal of Engineering & Technology, 7 (3.27) (2018) 252-257

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Massive Volume of Unstructured Data and Storage Space Optimization- a Review

Ranjeet V. Powar[1*] B. Arunkumar[2]

[1]*Research Scholar, Department of Computer Science and Engineering, Karpagam Academy of Higher Education, Coimbatore.*
[2]*Associate Professor, Department of Computer Science and Engineering, Karpagam Academy of Higher Education, Coimbatore.*
**Corresponding author E-mail:ranjeet.powar@gmail.com*

Abstract

Nowadays the volume of digital data generated and used by enterprises is increasing at an enormous rate. The survey says that more than 80% of data that were generated in the last two years are unstructured in nature. Hence storage space requirement for storing this big volume of unstructured data is very high. It has gained attention to large-scale storage systems. Deduplication is a space efficient method mainly used to solve storage space optimization problem. This paper focuses on the effect of massive volume of unstructured data and review various storage optimization techniques and survey of various storage types. In addition, it elaborates specific challenges with regard to storage optimization using deduplication and technology that handles a huge amount of unstructured data.

Keywords: *Deduplication, storage optimization, data compression, unstructured data.*

1. Introduction

In today's Computer era, data arrive from different sources and they are in a variety of forms such as structured, semi-structured and unstructured. Different sectors like manufacturing, business, science and sensors at different places generate a huge amount of data every day. Due to the use of Internet of Things and social networking sites, the collection of unstructured data grows rapidly. However, the growth of the data volume increases the burden on computing as well as storage infrastructure.

There are a lot of big data examples [7][29]: More than 4.15 billion active users of Facebook are generating social interaction data, Wal-Mart's transaction database contains more than 3 Petabytes of data and more than 5 billion people use their mobile phones for calling, texting and browsing websites. This growth of data from Terabytes (TB) to Petabytes (PB) increases pressure on the storage devices.

Peoples uploads photos, videos, documents and text friends on internet. Most of the time, relatives, common friends and colleagues of the same department upload same files on internet. Hence the enterprise data centers receive a large amount of redundant copies of data. Due to full system backups, shared documents and redundant files from users needs a lot of storage space at data centers. Again it consumes a large number of network resources due to the transmission of huge volumes of duplicate data over the network.

Data deduplication and data compression are storage optimization techniques. Data deduplication saves a huge amount of storage space by removing redundant data more efficiently and cost effectively to achieve a better deduplication ratio. Data deduplication is a process where a single copy of data is maintained against redundant copies of data on data server. Data compression is a technique which differs from deduplication by reducing the file size. The data compression reduces the size of a file but can't removes its redundant copies.

2. Unstructured Data

Unstructured data are not organized and do not have fixed data model. The term unstructured data refers to information that does not fit in traditional databases. It is either textual or non-textual data. Text file, social media data, website contents, text messages on mobiles, images, audio, video files, business application data are some examples of unstructured data. Weather data, military movements, atmospheric data, surveillance photos and videos, traffic and large number of sensor data are examples of machine generated unstructured data. IDC [7] estimates that more than 80% of organization data are unstructured in nature and are growing at the rate of 60% per year. Nowadays unstructured data have becomes a big challenge for organizations--the challenge of storing, maintaining and analyzing these data with optimum resources. Again, redundant copies of the same data on social media and internet, increase the burden on storage devices. For example, if one popular video of 100MB size is uploaded on social site by 1000 users all over the world, then it will take 1TB size on disk. If replicas are maintained for data availability then 3TB size on disk is needed to store 100MB size of a video file.

Sources of Unstructured Data

Today we live in the information age. Voluminous data has been generated from different sources however the internet of things and social media sites are generating are cord count of unstructured data and transactions. The data which do not fit in traditional database systems, a free flow data, are denoted as unstructured data. Unstructured data may contain text and multimedia files. Consider the following survey: YouTube [14] has over 1 billion users and People upload 100 hours of video per minute on it. Every day those users watch a billion hours of videos and generate billions of views. The statistics are increasing every

year and now it is 50% greater than previous year. More than 100 TB of data are uploaded daily on Facebook [16]. The site has maintained more than 300 billion photos which are rising every month by 7 petabytes. Instagram [30] users share more than 95 million posts and like 4.2 billion posts per day. Over 1 billion accounts have been created and more than 2 million search queries in minute every day on Google.

As per survey 2500 thousand Terabytes of data have been generated daily and more than 80% of total data all over world have been generated last year alone. It is not easy to measure the total amount of data stored electronically. However IDC [7] estimates that universe data were near about 5 Zeta bytes in 2014 and forecast that 44 Zeta bytes of would be generated by 2020.

3. Storage Space Optimization Techniques

This section provides the classification of traditional data compression and data deduplication approaches.

3.1. Data Compression

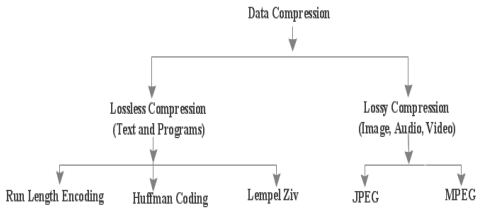


Fig. 1: Classification of data compression

Data Compression [10] [12] is broadly categorized into lossless and lossy compression techniques as shown in fig 1.In lossy compression, it does not generate original data after decompression but regains near match of data. In lossless compression, it regains original data without any loss of data after decompression. However data compression technique saves storage space on disk by reducing data size. The compression ratio of lossy compression is better than lossless compression technique. The lossy compression technique is mainly used to cut down the size of images, audio and video files before storing to storage disk. The lossless compression technique can be used to compress text and binary files. Sometimes it can be used to compress images also in digitized data. Data compression saves storage space and transmission costs. However compression and decompression require extra processing overhead.

A. Lossless Compression

Various Lossless data compression algorithms are used and the main methods are run length encoding, Huffman coding and Lempel-Zev algorithm. JPEG and MPEG are the main methods used in lossy data compression.

1. Run Length Encoding (RLE)

In RLE, repeated strings in file are replaced by single string and counter that is equal to the number of times it is repeated. It is simple and useful data compression method when the file contains a large number of repeated data values in sequence. It can be useful on small images like icons and animated files.

2. Huffman Coding

In lossless data compression, Huffman code is a particular type of optimal prefix code. The Huffman coding assigns code-words to input data blocks. It assigns small code-words to high probability input block and large code-words to low probability input block. A Huffman encoder takes a fixed-sized character block as the input and produces variable length block of characters. Today Huffman coding is used in other compression methods such as PKZIP algorithm and multimedia codec's. This is called intra-file compression.

3. Lempel-Zev Algorithms

Lempel–Zev is a universal lossless data compression algorithm. It encodes fixed-length 8-bit data as a code to repeated data pattern in file. This method splits the input data content into non-overlapping blocks and constructs a dictionary of blocks. It is also called dictionary-based encoding. As the algorithm performs best on data with repeated patterns, it gives little compression for initial parts of a message and maximum compression ratio when message grows. This is also intra-file compression.

B. Lossy Compression

JPEG and MPEG are the examples of lossy compression methods.

JPEG

JPEG compression method is used for image or picture compression. It divides whole image into small blocks and converts it into a set of numbers to identify redundancies. It decreases possible image colors which reduces the quality of image but improves network bandwidth efficiency. As this method loses color quality, it is used when color noting in not essential.

MPEG

MPEG compression method is used for video compression. Video is a series of images in frames which runs fast enough to appear as moving. The technique of JPEG for compressing single image is not sufficient for a series of still images in frames for video compression. In MPEG, all other images, excluding base image, are encoded by calculating the difference between images.

3.2. Data Deduplication

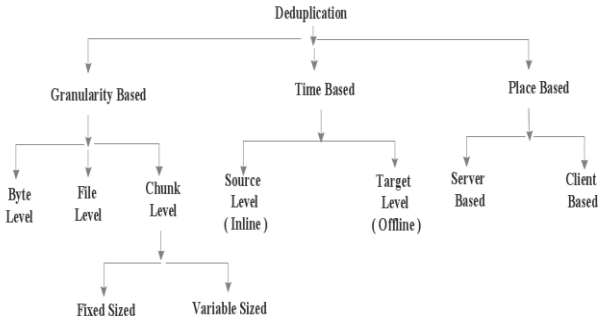


Fig. 2: Classification of data deduplication

Data deduplication [1][19] [20] is a compression technique that optimizes storage space by minimizing the amount of unwanted duplicate data. In normal condition, an email server requires200MB storage space when the same copy of the text file, which is of size 5 MB, is present in 40 people's Inbox. Data deduplication saves storage space by storing a single copy of file attachment and keeps reference of it to all other duplicate copies. These techniques are used to avoid transfer of duplicated data between the host and storage server, thus improving bandwidth efficiency. In general, data deduplication identifies redundant chunks of data and keeps just one copy of these repeated chunks.

The section furnishes detailed classification of data deduplication techniques.

Data deduplication is classified based on granularity, deduplication time and deduplication place as shown in fig 2. Three main components of deduplication process are chunk generation, hash code computation and indexing. In chunk generation process file data are divided into unit of data called chunk. To identify redundant data, hash key of chunk is computed, compared and hash key of unique chunk is stored as an index.

A. Based on a Granularity of Data

Data deduplication can be either at file-level, chunk-level or byte-level based on granularity of data.

1. File Level Deduplication

The file level deduplication process is shown in fig 3. In file level deduplication, an entire file is compared based on hash code of file to eliminate redundant copies of files from storage devices. For the first time, file and hash code of file are stored on disk and index respectively. For every new file, hash code is computed and checked for duplication. If hash code of the file exists in the index then that file is reported as duplicate and only entry in file is updated. Only one copy of redundant copies of file is stored on the disk. File level deduplication saves significant storage space if more redundant copies are to be stored. It requires less computation and index overhead. However, it shows less deduplication ratio compared to other deduplication techniques.

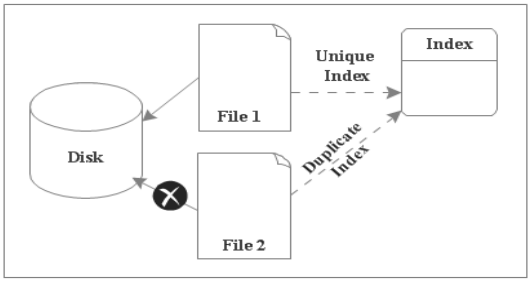


Fig. 3: File level deduplication

2. Chunk Level Deduplication

File level deduplication has limitations to identify redundancies within files. Chunk level deduplication [20] is further subdivided into fixed sized and variable sized chunk deduplication.

Fixed Sized Deduplication

In fixed sized chunk level deduplication, file is divided into fixed sized chunks (i.e. 4k, 8k, 16k, 32k etc.) and compares index of chunks to identify redundant chunks. The chunk level deduplication process is shown in fig 4. It is a simple and faster deduplication technique. However, small updates in either of similar files cause issues in identifying redundant chunk using fixed sized chunking deduplication. It has offset-shifting problem.

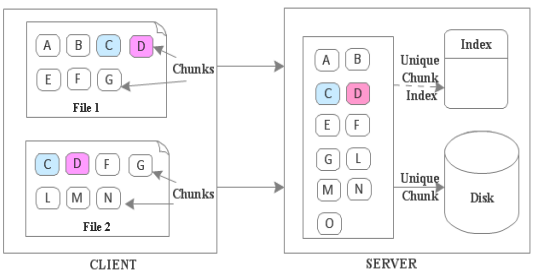


Fig. 4: Chunk level deduplication

Chunk level deduplication is further divided into fixed sized and variable sized chunk deduplication.

Variable Sized Deduplication

Variable sized chunking is also called content-defined chunking. It overcomes offset-shifting problem in fixed sized chunking deduplication. It saves more storage space by identifying more redundant data in similar files.

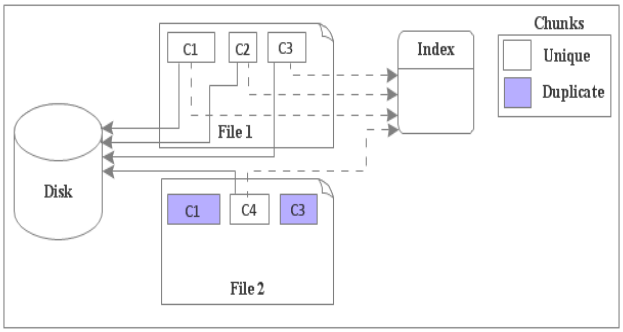


Fig. 5: Variable sized deduplication

In fig 5. file1 is the original file and file2 is similar to file1 with some modifications. Three chunks C1, C2 and C3 of file1 are unique chunks and stored on the disk. When file2 is to be uploaded then C1 and C3 chunks of file2 are identified as redundant chunks and only C4 chunk, the unique chunk are stored on the disk. Variable sized chunk deduplication can identify more duplicates than fixed sized chunk deduplication. However, it requires more processing and index overhead.

Byte Level Deduplication

In byte-level deduplication [20], duplicate detection is at the byte level. This method identifies the maximum possible duplicates and saves more storage space as compared to other deduplication techniques. A byte-level comparison is simple but requires high computation overhead.

B. Based on the Timing of Deduplication on Data

There are two methods namely source (inline) and target (offline) data deduplication based on timing of deduplication on data.

1. Inline Deduplication

In inline deduplication, redundant copies of data are eliminated or files before it are stored on disk. Fig 6.explains how inline deduplication works. When a user on client machine uploads data file, deduplication process reads data and checks for duplicates. Single copy of data and indexes are saved on storage devices.

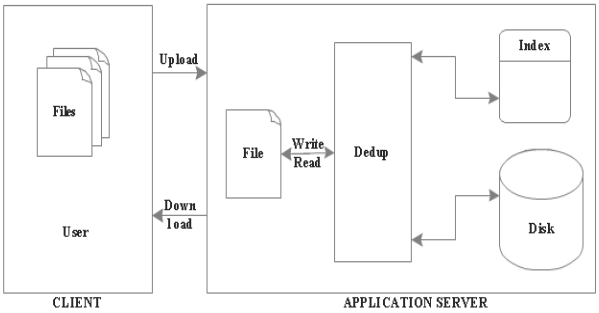


Fig. 6: Inline deduplication

Inline deduplication removes data redundancies on write path, thus saving storage space as well a network bandwidth without incurring extra space overhead. However, this technique requires more time to upload data.

2. Offline Deduplication

In target (offline) deduplication, redundant copies of data of files are eliminated after data are stored on disk. Fig 7. shows how offline deduplication works. When a user uploads data files, data files are stored on storage disk without deduplication. This technique runs deduplication on stored data at disk and compares every data with other data on disk.
Offline deduplication does not require extra time while uploading data, but requires extra disk space. However, this technique consumes extra network bandwidth to upload all data files along with redundant copies of data.

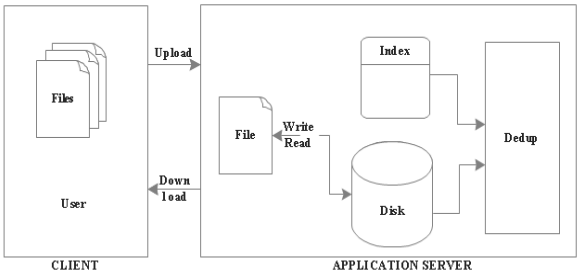


Fig. 7: Offline deduplication

C. Based on the Placement of Deduplication Process is Performed

There are two methods named as Server Based (Target) Deduplication and Client Based (Source) Deduplication.

Server Based Deduplication

Fig 8. shows working of server based deduplication technique where client machine sends data files to server through client application. On server, files are divided into chunks and hash code is computed for each chunk. New chunk hash code is compared with indexes of previously stored chunks in index file. Only unique chunks and its hash code are stored on disk and index file respectively.
Server based deduplication removes significant redundant data but consumes more network bandwidth to transfer all data to server. In addition to that, server requires extra computation and memory overhead for indexing all data.

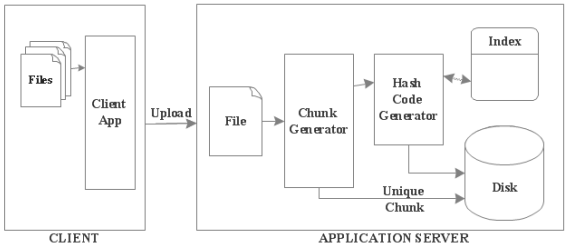


Fig. 8: Server based deduplication

Client Based Deduplication

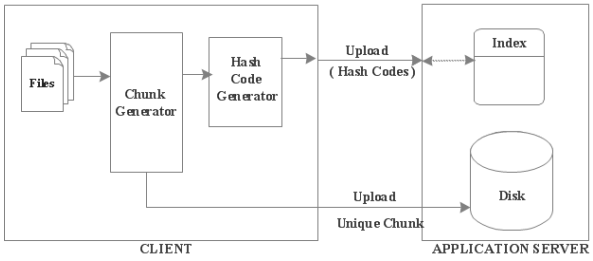


Fig. 9: Client based deduplication

In Client based deduplication, the deduplication process runs on client. As soon as user uploads file, the deduplication module on client splits the file into chunks and computes hash codes. It checks new hash codes into index maintained at client and only hash code of non-redundant chunks are sent to server. Fig 9 shows how client side deduplication works to eliminate redundant data among clients with the help of server. The hash code of chunks is compared at server to identify duplicate chunks and unique chunk indexes are informed by server to client. Then client sends unique chunks to server to store on disk.
Client based deduplication removes significant duplicate data before transferring to server. Only redundant chunks among clients are sent through network occupies extra bandwidth. The client requires extra computation and memory overhead.

4. Key Issues in Deduplication

Deduplication can be considered as a process of eliminating redundant copies of same data. The data which are to be uploaded are compared with the data which are already stored in storage devices and duplicates have been removed. In the design of deduplication system, different key issues are identified as challenges.

Challenge 1: Granularity

All deduplication systems create chunks of data which are to be compared and removed after being identified as redundant. The main challenge in storage optimization system due to granularity is overhead on the storage system. The smaller chunk size saves more space on storage devices but leads to bigger index structures which are unable to fit in main memory. Index files are too large due to relatively small chunk size and this is one of the storage issues. Hence it needs to store on disk and this leads to another issue. The storage of index files on disk decreases IO performance while using deduplication.

Challenge 2: Computation Overhead

File level deduplication runs faster due to no overhead of partitioning file. But data storage efficiency is achieved more when data are partitioned into chunks. The new data chunk is compared with existing data chunks to identify and remove duplicates. The hash based chunking technique is used to achieve better performance of deduplication system. In hash-based deduplication, a hash value of each data chunk is calculated and stored as a fingerprint of a chunk. The computation cost of creation of chunk and generation of fingerprint for large volume data is high. Hence, chunk level deduplication is more CPU intensive than file-level deduplication.

Challenge 3: Indexing

In deduplication system, unique fingerprints of every chunk are to be saved in chunk index. When the size of a chunk is small, a larger number of chunks are created and as a result, a larger number of fingerprints are created.
When chunk index is large, searching newly arrived chunk fingerprint for the duplicate chunk is a time consuming job and degrades the overall performance of deduplication system. It is also difficult to keep large chunk index file into primary memory. Thus, there is a need to store on disk and it increases a lot of IO operations.

Challenge 4: Restore

The deduplication system removes redundant data and stores only unique data. Then it is necessary to restore data whenever requested by the user.
The continuous chunks in each file are physically stored in different data nodes. The file restore is one of the vital processes in a whole deduplication system.
Extra metadata and integrity of metadata need to be maintained to correctly restore the data file. The chunk fragmentation also degrades the restore performance.

Challenge 5: Security and Privacy

Deduplication based storage systems face issues of security and privacy. Different users encrypt data with their private keys and create different cipher-texts which make deduplication impossible across different users. The client uses the fingerprint as proof of ownership for the entire file to prove the ownership of a file. Then the privacy issue is, the attacker who is familiar with the proxy fingerprint of the file can prove ownership to the storage server and download the entire file.

5. Storage Types

Each storage type has its own environment and requirements. The design features of storage optimization systems change significantly with storage type being used.

5.1. Backup and Archival Storage

There is a large quantity of duplicates in secondary storage systems, such as backup and archival storage [31]. The data redundancy in backup and archival storage ranges from 35% to 80% [7]. This type storage requires temporary storage devices to store a huge amount of backup data, which leads to the problems of storage cost and power consumption. The storage optimization systems save storage space and reduce hardware cost with some additional computation and I/O overheads.

5.2. Primary Storage

Primary storage [32] is used as container for online and transaction data, which are accessible to end users. There is a need to design primary storage system in such a way that optimal performance can be achieved. The computation and index overhead of deduplication technologies impact on performance. The survey says that more than 40% of data are redundant copies of previous data. Primary storage is relatively more expensive. Due to its higher cost, the deduplication system is even more critical.

5.3 Random Access Memory

The amount of digitized data increases at an enormous rate and perhaps doubles every year. As data volume increases, the index file becomes large and search time for duplicate check also increases. Sometimes index files are too large to fit in random access memory.

5.4. Solid State Drive

Recently, Solid State Drive (SSD) becomes more popular in data centers because of its better IO performance and low energy requirement. SSD has less storage capacity than traditional hard drives. In big data environment, SSD can be used as better option of storage to reduce the size of datasets using deduplication.

5.5. Cloud Storage

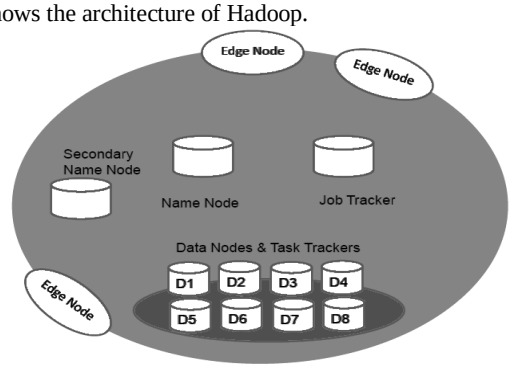
The Big data [33] concept is gaining importance in recent years and it demands data storage requirements much beyond the capabilities of a traditional standalone storage system. In order to extend the storage capacities at a large-scale, a popular approach is to move all the data to the cloud storage system.

6. Technology to Process Unstructured Data

Nowadays processing and managing unstructured data are problems for any organization. Different tools can be used to handle the massive volume of unstructured data. These tools are Big Data tools, Business Intelligent software, Data integration tools, and search and indexing tools. In BI, a broad category of analytics and reporting tools helps the organizations to make business decisions on structured and unstructured data. Data integration tools merge structured and unstructured data for analytics by a single software. Search and indexing tools collect the required information from unstructured data. Apart from all these tools, Hadoop [25][27] is de facto big data tool for storing and managing a huge amount of unstructured data.

6.1. Hadoop

The rapid growth of unstructured data creates challenges before IT industries. The ample of newly generated business data have challenge of storing and retrieving efficiently. Data scientists have one more challenge of processing and analyzing stored data effectively. Hadoop is an open source framework that efficiently handles structured as well as unstructured data. Hadoop [18] is developed from Google's MapReduce and Google's file system. It uses commodity hardware to store and manage a large volume of unstructured data and scale up to thousands of machines without limits. Hadoop is cost effective as compared to other legacy software systems. Hadoop is simple, accessible, robust and scalable distributed programming framework. Apache Hadoop has two main components MapReduce and Hadoop Distributed File System (HDFS). The processing part of Hadoop is called MapReduce programming model and storage part is called HDFS. Fig. shows the architecture of Hadoop.



Hadoop architecture

Apache Hadoop consists of five demons that are Namenode, Secondary Namenode, Jobtracker, Datanode, and Tasktracker.

- Namenode: It is the master demon which contains location address for every data block on Datanode. It only stores metadata of data stored on HDFS.
- Secondary Namenode: It is a backup node for Namenode and can be used in case of failure of Namenode.
- Jobtracker: It keeps track of running MapReduce jobs and sends work to Tasktracker nodes in the cluster.
- Datanode: It is a slave node and contains actual data. It periodically sends heartbeats to Namenode to inform his aliveness.
- Tasktracker: It is demon which is running on Datanode. It keeps track of operations on Datanode and communicates with Jobtracker through heartbeat messages.

A). MapReduce

MapReduceis certainly not the first programming model to process distributed data in parallel paradigm. It provides a very important tool for handling a huge amount of data. It is a data processing model consisting of two processing components that are called mappers and reducers. [34] It divides large problems into small parts and assigns them to separate workers. These workers are called mappers. Intermediate results of each worker are combined and sorted as per business logic using reducers. The map and reduce phases of MapReduce framework process a large amount of data on clusters of Data nodes.

B). HDFS

HDFS is responsible for storing and maintaining data on a cluster of low cost commodity hardware in Hadoop. Every data file is divided into the block of 64mb before storing on HDFS. The blocks of a data file are replicated on different Data nodes for the purpose of availability of data in case of failure of any of Datanode in a cluster. HDFS allows Hadoop to store different kind of data like structured data, unstructured data, image, audio and video.[35]

7. Conclusion

This paper illustrates storage space saving methods and tools to handle massive amounts of unstructured data. It also focuses on the difference between data compression and data deduplication. Data deduplication is a more proficient duplicate data elimination technique than other techniques. Now industries have widely accepted the data deduplication and their benefits in storage optimization. Finally, the paper addresses key issues due to the massive growth of unstructured data on data deduplication in storage systems and overview of Hadoop as tool to process and manage a huge amount of structured as well as unstructured data.

References

[1] He Q, Li Z & Zhang X, "Data deduplication techniques", *IEEE International Conference on Future Information Technology and Management Engineering (FITME)*, (2010), pp.430-433.

[2] Chen CP & Zhang CY, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data", *Information Sciences*, Vol.275, (2014), pp.314-347.

[3] Kulkarni P, Douglis F, LaVoie JD & Tracey JM, "Redundancy Elimination within Large Collections of Files", *USENIX Annual Technical Conference, General Track*, (2004), pp.59-72.

[4] Michael K & Miller KW, "Big data: New opportunities and new challenges [guest editors' introduction]", *Computer*, Vol.46, No.6, (2013), pp.22-24.

[5] Shoro AG & Tariq RS, "Big data analysis: Apache spark perspective", *Global Journal of Computer Science and Technology*, (2015).

[6] Manyika J, Chui M, Brown B, Bughin J, Dobbs R, Roxburgh C & Byers AH, "Big data: The next frontier for innovation, competition, and productivity", *McKinsey Global Institute*, (2011).

[7] Turner V, Gantz JF, Reinsel D & Minton S, "The digital universe of opportunities: Rich data and the increasing value of the internet of things", *IDC Analyze the Future*, (2014).

[8] Nguyen TH, Shirai K & Velcin J, "Sentiment analysis on social media for stock movement prediction", *Expert Systems with Applications*, Vol.42, No.24, (2015), pp.9603-9611.

[9] Salomon D, *Data compression: the complete reference*, Springer Science & Business Media, (2004).

[10] Reghbati HK, "Special feature an overview of data compression techniques", *Computer*, Vol.14, No.4, (1981), pp.71-75.

[11] Boldi P & Sebastiano V, "The web graph framework I: compression techniques", *Proceedings of the 13th international conference on World Wide Web*, (2004).

[12] Sethi G, "Data Compression Techniques", *International Journal of Computer Science and Information Technologies*, Vol.5, No.4, (2014), pp.5584-6.

[13] Chen M, Shiwen M & Yunhao L, "Big data: A survey", *Mobile networks and applications*, Vol.19, No.2, (2014), pp.171-209.

[14] Statistics, YouTube, YouTube Inc., (2016).

[15] Hess B & Virginia T, "Educating Consumers through Social Media", *Consumer Interests Annual*, Vol. 60, (2014).

[16] Brain, Statistic, "Facebook statistics", *Retrieved March*, Vol.17, (2014).

[17] Brain, Statistic, "Twitter statistics", *Statistic Brain* (2014), http://www.statisticbrain.com/twitter-statistics/.

[18] White, Tom. *Hadoop: The definitive guide*, O'Reilly Media, Inc, (2012).

[19] Bigelow SJ & Hawkins J, Data deduplication (Intelligent compression or single-instance storage), (2008).

[20] Matze JEG, "System and method for data deduplication", U.S. Patent No. 8,205,065, (2012).

[21] Venish A & Siva Sankar K, "Study of chunking algorithm in data deduplication", *Proceedings of the International Conference on Soft Computing Systems*, (2016).

[22] Shin Y, Dongyoung K & Junbeom H, "A survey of secure data deduplication schemes for cloud storage systems", *ACM Computing Surveys (CSUR)*, Vol.49, No.4, (2017).

[23] Sharma S & Mangat, V, "Technology and trends to handle big data: Survey", *IEEE Fifth International Conference on Advanced Computing & Communication Technologies (ACCT)*, (2015), pp.266-271.

[24] Bhadani AK & Dhanya J, "Big Data: Challenges, Opportunities, and Realities", *Effective Big Data Management and Opportunities for Implementation, IGI Global, Pennsylvania, USA*, (2016), pp.1-24.

[25] O'Malley, Owen, "Terabyte sort on apache Hadoop", *Yahoo*, (2008), pp.1-3.

[26] Zakir J, Tom S, and Kristi B, "Big Data Analytics", *Issues in Information Systems*, Vol.16, No.2, (2015).

[27] Cohen J & Subatra A, "Towards a more secure apache hadoop hdfs infrastructure", *International Conference on Network and System Security*. Springer, Berlin, Heidelberg, (2013).

[28] Frank S, "Scalable block data storage using content addressing", U.S. Patent No. 9,104,326, (2015).

[29] Yaqoob I, "Big data: From beginning to future", *International Journal of Information Management*, Vol.36, No.6, (2016), pp.1231-1247.

[30] Smith C, "By the numbers: 160+ interesting Instagram statistics", *Retrieved February*, Vol.11, (2016).

[31] Meyer DT and William JB, "A study of practical deduplication", *ACM Transactions on Storage (TOS)*, Vol.7, No.4, (2012), pp.1-14.

[32] Nam YJ, Dongchul P & David HCD, "Assuring demanded read performance of data deduplication storage with backup datasets", *IEEE 20th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems* (MASCOTS), 2012.

[33] Wallace G, "Characteristics of backup workloads in production systems", FAST. Vol.12, (2012).

[34] G, Abikhanova, A Ahmetbekova, E Bayat, A Donbaeva, G Burkitbay (2018). International motifs and plots in the Kazakh epics in China (on the materials of the Kazakh epics in China), Opción, Año 33, No. 85. 20-43.

[35] A Mukanbetkaliyev, S Amandykova, Y Zhambayev, Z Duskaziyeva, A Alimbetova (2018). The aspects of legal regulation on staffing of procuratorial authorities of the Russian Federation and the Republic of Kazakhstan Opción, Año 33. 187-216.