International Journal on
“Technical and Physical Problems of Engineering”
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

March 2022 | Issue 50 | Volume 14 | Number 1 | Pages 196-203

ENHANCEMENTS AND AN INTELLIGENT APPROACH TO OPTIMIZE BIG DATA STORAGE AND MANAGEMENT: RANDOM ENHANCED HDFS (REHDFS) AND DNA STORAGE

M. Sais N. Rafalia J. Abouchabaka

Department of Computer Science, Computer Research Laboratory (LaRI), Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco, manar.sais.lari@gmail.com, arafalia@yahoo.fr, abouchabaka3@yahoo.fr

Abstract- The evolution of mobile technology, the popularization of tablets and smartphones, the daily data generated by industries, large organizations, and research institutes, and the rapid growth of social media have all created unprecedented amounts of data. Different types of data with different levels of complexity are being generated at different speeds. The term big data has emerged from the huge increase in data storage and processing requirements. However, the use of traditional databases poses potential problems for processing and storage of this huge data and he has no ability to withstand the weather, and we are in dire need of new technology to store and manage our voluminous data in a good way. in order to respond to this demand a next-generation of storage and management systems are designed and implemented to this large data. These current technologies (Hadoop and MySQL) had a great deal of credit in addressing the problem of storing large data, but nevertheless it faces some challenges and cannot keep pace with this increasing exponentially of data. This is what prompted us to make more improvements to the current technology like the random enhanced HDFS system (REHDFS), as well as to search for more efficient and permanent storage solution like DNA synthetic to face the future generation of data. The objectives of this paper are to present the two future storage solutions with the advantages and values-added by both approaches.

Keywords: Big Data, Data Storage, Hadoop, NoSQL, HDFS, REHDFS, DNA Synthetic.

1. INTRODUCTION

Since the origin of mankind, man has never stopped leaving written traces, they come from rock art, the first form of writing known to mankind 40,000 years ago. This art has consisted of engraving and painting on walls. Remains of the cave of Lascaux, Combarelles are still not kept. Writing is the most important means of communication that allows us to transmit, share our remains, our knowledge, and the whole experience of the history of Man and his environment to future generations. Today, in the 21st century, we are overwhelmed by a mass of data with a constant and growing flow.

The rise in digital storage is increasing at an exponential rate, but traditional storage media's capacity is insufficient., posing a major challenge for today's data centers and storage technologies. To meet the increasing demands for important data storage and processing, advance of big data platforms had resulted in creation of different tools, products, and database systems [1-3].

Data storage encompasses all media able of storing numerical data. This holds true for entirely forms of media. Hard disks, floppy disks, and even SSDs are examples of media. It is free to use for personal purposes (documents, music, photos, etc.) [4]. Increase of big data [5] has resulted in a rapid increase in global data volumes, creating new opportunities in a variety of fields like scientific research, finance, business, and medicine. We would be able to perform prediction or trend analysis, establish profiles, anticipate hazards, and follow events in real time. The fundamental challenge with big data is locating technology that could store and analyze enormous volumes of data so that information can be examined and extracted. Several solution suppliers provide out-of-the-box large data management solutions [6-11].

For Big Computational and Storage Systems, Hadoop has become the de facto industry standard [12]. Hadoop is based on a distributed data system that saves data in a specified format and a parallel processing idea that uses a cluster of computers to operate the Map-Reduce processing paradigm [13]. With a simple computational model, the framework is extremely scalable and fault resistant. In recent years, NoSQL technology has gained in popularity [14], and data storage differs from standard SQL databases, with each form of NoSQL database having its unique data storage needs. The NoSQL is a new class of data storage systems that support Big Data, many of which sacrifice query power and consistency guarantees to provide horizontal scalability and superior availability for relational databases. The work [54] compares traditional data storage and technologies, and summarizes the various future challenges facing these technologies.

Although all these developments and efforts have been made to invent methods and technologies to solve the problem of big data management and storage, all these existing technologies are still incomplete and still need to be improved, but in the future, when faced with such a large amount of data, they will lose their effectiveness. For example, the Hadoop HDFS system lacks several features. It only allows users to perform sequential operations, and does not allow random operations. In order to save time and efficiency, there is a great concern to look for a solution to these problems, either by developing and adding a smart touch to the available technologies (Hadoop and REHDFS) or by discovering solutions efficient in another domain for example chemistry (synthetic DNA).

The paper is devoted to the analysis of two approaches that have the objective of improving the process of storage and management of Big Data. The primary solution is an improved HDFS (REHDFS) that investigates various block placement and read strategies, supplying a scalable and a load-based block connection strategy to outperform other strategies. More reading and random writing features depending on the primary solution is an improved HDFS (REHDFS) that investigates various block placement and read strategies, supplying a scalable and a load-based block connection strategy to outperform other strategies. More reading and random writing features depending on optimistic and pessimistic models the other solution is DNA strands, which have demonstrated unbelievable storage power of several G bits in a perfectly consistent manner and could be the answer to store a highest amount of data in a minimum amount of space.

This paper's sections are organized as follows: The first part provides an overview of the research. Section 2 gives an overview of huge data, these storage issues, and Hadoop's representation (HDFS) technology. Section 3 presents REHDFS (Random Enhanced HDFS) as an enhanced solution for a simple HDFS system and its added value. Section 4 serves to introduce synthetic DNA as novel type of big data storage and a encouraging solution to efficiently preserve our data in long term with a comparison of DNA storage and REHDFS system.

2. RELATED WORK

The volume of enormous and complicated data is gradually expanding with the growth of Internet usage and the huge growth of the information business. Data generated has increased exponentially, with large data volumes and different structures, far exceeding the capacity of traditional storage devices. The operation of identifying, storing, and processing unstructured data has become a major challenge [15]. The Big data challenge has become more complex with noisy, sparse and heterogeneous data. As the need for improved technologies that could also efficiently process big volumes of data in a short amount of time grows, so does the desire for innovative systems that can store vast amounts of data.

The paper [16] have developed a big data storage optimization method for device health monitoring based on the Hadoop cloud computing platform. This research aims to provide a consistent hashing algorithm across multiple copies to ensure the relevance of monitoring data, and to use adaptive searchable encryption schemes for storage optimization. This study results of indicate that this approach to safeguarding equipment monitoring data can be effectively used in practice to guarantee the security of Ex equipment monitoring data.

Another study [17] describes new Hadoop Archive, which is centered on Hadoop Archive (HAR) and intends to enhance the accuracy of obtaining minor files in HDFS and improve metadata memory usage. In addition, it extends the functionality of HAR to allow the insertion of additional files into existing archive files.

The authors of [18] suggests an optimum file placement method based on Hadoop Statistical Workload Injector for MapReduce (SWIM). The process utilizes real workloads to evaluate a technique for optimal file placement in storage in terms of improving I/O performance in Hadoop. Various I/O scenarios for some SWIM jobs are examined. Then look at the I/O patterns of a few SWIM jobs. In order to meet storage needs, researchers have offered a new storing method data. One of the new techniques of entering data in DNA is a process known as genetic data storage. The principle objective of this study [19] is introducing DNA as an excellent data storage method and to address the two main problems associated with it. The first one is the extraction of genomic data, which is a very tedious process, although improvements can be expected. The second is the cost factor, as such technology can be very attractive, so the cost will increase.

Researchers in [20] conducted multiple experiments to estimate the artificial aging of DNA. They stored information in encapsulated DNA fragments with error-correcting codes, the aging process is then accelerated by exposing the mixture to very severe conditions (such as high temperature). By tracking the results of the kinetic degradation over time, the researchers were able to recover the original information. According to this experiment, artificial aging is equivalent to 2000 years in Central Europe.

3. BACKGROUND

3.1. Big Data

Big data necessity profitable and new methods of information processing in order to provide improved understanding, production, and automating of decision-making procedures. Big data has immense promise in industry and it has the ability to permanently alter the way businesses make choices and perform research in a variety of fields [22].

The amount, diversity, and velocity of big data are increased. Data that is large in volume, arrives quickly, and is diverse in kind, comprising both organized and unstructured data [23]. The quantity of data from various input resources that rises each second is referred to as data volume [25]. Variety refers to all sorts of data that have experienced significant changes in classical structured data analysis needs as part of the decision-making and understanding process. The pace during which data is created and processed is referred to as velocity.

Data storage is the process by which a computer system archives, organizes, and shares the information that makes up the things we use every day; it provides a physical space for all applications to store and access all their data. This data will be used by different types of applications, including standalone desktop applications, web applications, mobile applications, etc. [25]. There are many methods or technologies that can store, manage and access this data. These methods or technologies will continue to evolve with new inventions and technological breakthroughs.

In recent years, the amount of data generated daily by industries, networks, e-commerce, large organizations, and research institutes has increased at an alarming rate. Storing, managing and retrieving this big data is the most important task, not only for analysis purposes, but also in compliance with laws and service level agreements for data protection and retention. The challenges related to big data storage and processing challenges include [26]: data capture, data storing, data search, data share and data analysis.

3.2. Hadoop Distributed File System

Apache Hadoop is the most prominent free and open-source system in data storage, processing, and analytics sector due to its easy and cost-effectiveness. Hadoop is composed of 2 main components: an HDFS storage component and a MapReduce processing component.

HDFS is an excellent Big Data storage solution. Its capabilities include the potential to save terabytes or even petabytes of data due to its enormous capacity and dependability. Combining this approach with YARN improves the HDFS Hadoop cluster's data management capabilities, allowing it to handle large amounts of data effectively. This section details the HDFS file system and compares it to the improved REHDFS file system, as well as its additional features.

3.2.1. Functionality and Architecture

We need a robust file system in the digital age to efficiently store created data and huge files. Faster data transport implies that dispersed systems can be used more effectively. Hadoop Distributed File Solution (HDFS), which runs on commodity hardware clusters, become the more popular high bandwidth, efficiency system for continual Big Data storage because of its availability [27]. HDFS is used to store any type of dataset that is required to run multiple user products to make business intelligence choices [32].

A master server and multiple slave servers are set up in HDFS (Data nodes). The master server is a name node that manages the file system name-space and client access to different files stored in the file system, whereas the slave server is made up of data nodes which actually store the data files. The design is fault-tolerant and scalable [28]. The core method for HDFS fault tolerance is data replication, which allows each block of data to be duplicated across several Data Nodes (replication factor by default is 3).

Each HDFS file is made up of blocks (block size by default is 64 MB). When an HDFS client requests that a file be opened in write mode, the NameNode creates a block with a unique ID and selects which DataNodes will host a copy of the block. A pipeline is formed by the Data Nodes [29]. The client pushes the data to the next Data Node in the pipeline after writing the block of data to the first Data Node. Bytes are delivered through the pipeline as a series of packets. After the acknowledgement is received in the pipeline, it is written to the Data Node. When all replicas have been written properly, the client requests that the NameNode write the next block.

3.2.2. HDFS Difficulties

Users can contribute files to HDFS on a regular basis. If a client wants to update a single byte in a file, they must first create a new file, then replace the old one. Consider the case below: A development team should be able to handle a high number of project files. These files can be modified by any project member and should be easily accessible and fail-safe, therefore maintaining consistency is essential. The members of the team employ a range of file processing software. Standard I/O operations such as write(), read(), seek(), and others are supported by the project's technologies. For file management, random read and compose routines are required [30].

4. PRACTICAL STUDY OF REHDFS

During large-scale data processing, random queries become increasingly important. Unfortunately, random write operations have many drawbacks in HDSF while ensuring data consistency. To solve this issue, we try to provide a solution that improves HDFS performance while writing files randomly [31].

In this section we present the improved HDFS system (REHDFS) with its architecture, its additional functionality and also the two models to operate the arbitrary write operation (pessimistic and optimistic).

4.1. REHDFS Design and Features

The REHDFS is shown in Figure 1. The main components of this architecture are almost the same as HDFS, except that new component are added to meet new functions (lock/validation manager and cache module).

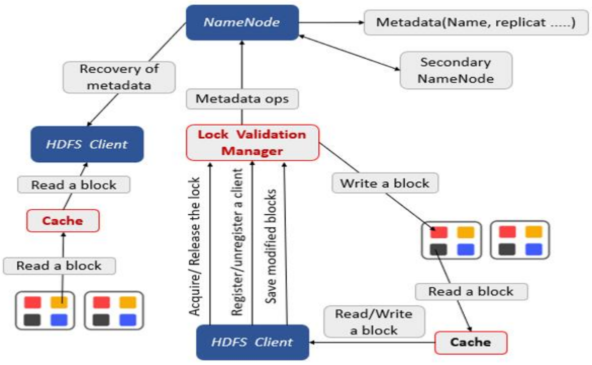

Figure 1. Extended HDFS architecture based on RMI

A new component named Lock/Validation Manager was introduced to the HDFS design to facilitate random writing operations. For random write operations, the component provides two models (optimistic and pessimistic). Difference between the two systems is the lock acquisition. Clients need first obtain a lock from the lock/validation manager when using the pessimistic model. The optimistic technique, on the other hand, allows the client to change file cache blocks without acquiring locks. The next part [30] will go into the specifics of these two models.

The client retrieves a block from the data node, which is stored in the cache module to make retrieving the block easier. If the client's requested block is in cache, it is given to them. Otherwise, the block is cached, and the client receives a copy from the data node that hosts it. The cache block relating to the alteration is updated when a client updates a specific segment of a file. In addition, the cache module keeps track of which blocks have been modified by the client.

4.2. Arbitrary Write

For the basic HDFS system, storing a file is as follows, the file must first be partitioned into one or more blocks, with each block duplicated over several data nodes. These replicas complicate random write operations, which HDFS does not handle. Through two additional templates, REHDFS uses two models for implementing random writes. Optimistic or pessimistic approaches to random writing might be used.

The FLPM (File Level Lock with Pessimistic Model), State diagram of a file in the pessimistic model is presented in Figure 2.

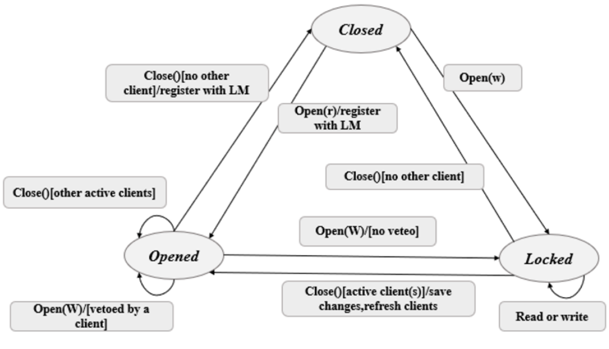
Closed
Close()[no other client]/register with LM
Open(w)
Open(r)/register with LM
Close()[no other client]
Close()[other active clients]
Open(W)/[no veteo]
Opened
Locked
Open(W)/[vetoed by a client]
Close()[active client(s)]/save changes,refresh clients
Read or write

Figure 2. Pessimistic Model diagram

In pessimistic design, we can find a file in three states: closed, open and locked. Each file starts in a closed state. If a client wishes to modify or write to a file, it must obtain the agreement of all clients operating on the file and registers with the lock manager (LM) [30]. If one of the clients rejects the request with a veto, the write operation fails and the status of the file becomes locked. The file's state switches to closed (since no other client reads it) or open (when another client reads it) when the client who holds the lock requests that it be closed () (when the file is read by client). The close () request from the lock management owner transmits an update request to all clients.

When a client tries to make a file, it first tries to register with lock management system. The manager examines the file to determine if it is locked by some other user. If this is not the case, the open request will be fulfilled.

As a leader, the lock manager is responsible for responding to client requests, creating lists of active clients, issuing locks to requesting clients, deleting clients after the lease expires, logging changed blocks, and notifying active clients of changes [30]. Optimistic Modeling with File-Level Consistency:

The state diagram of the optimistic model displays the client's four states: inactive, registered (with validation manager), update (with reading block), and attempting to save the updated block.

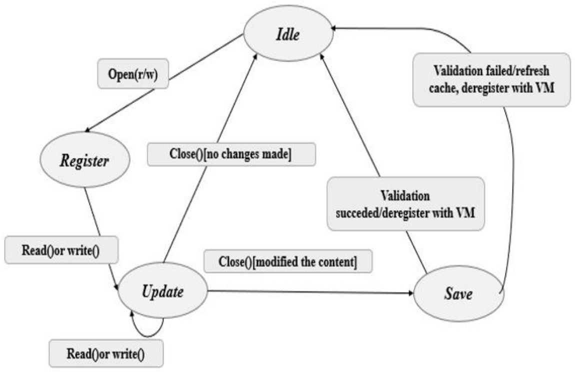
Idle
Open(r/w)
Validation failed/refresh cache, deregister with VM
Register
Close()[no changes made]
Validation succeded/deregister with VM
Read()or write()
Close()[modified the content]
Update
Save
Read()or write()

Figure 3. Diagram of the Optimistic Model

A file's first state is inactive. If a user opens a file, it goes into backup mode, logs a callback to manager Verification (VM), and gets a unique timestamp from the VM. When the client receives the block, it enters the updated state and alters it as needed. The validator manager verifies the changes after closing () the user from such an update phase to an inert phase (if the client hasn't modified any blocks) or a save state (if client has altered one or more blocks). The validation succeeds and the changed block is saved [30] if a client has lowest timestamp value or if no other user reacts on the file.

In REHDFS, a single component named Lock-Validation Manager handles both the validator manager and the lock manager.

4.3. FLPM vs. OMFC

To make modifications to a file using pessimistic model (FLPM), user must first get a lock on the file. The deadlock problem is experienced in this architecture while a set of customers desires to alter a collection of files and requests locks. In optimistic model (OMFC), on the other hand, any clients with authorization to alter blocks can do so in their own caches, and the client's changes with the earliest timestamp are stored. When numerous clients are executing on the same file, the OMFC model performs better than the FLPM model.

5. DNA SYNTHETIC AS A DATA STORAGE MEDIUM

The quantity of data created today greatly surpasses the storage capability of our technology, and the world is on the verge of a data storage disaster. Despite significant advancements in classical data storage technologies, the advent of energy conservation and Big Data platforms concerns provide new difficulties to the storage industry [33]. Researchers have developed a novel data storage approach termed “genetic data storage” to answer this pressing demand for new storage technologies. Deoxyribonucleic acid (DNA) has shown to be a viable Big Data storage medium due to its high density, massive store capacity, and long-term stability [21] and [34]. The predicted information density of four nucleotides (adenine (A), thymine (T), cytosine (C), and guanine (G) [35]) is roughly 1018 B/mm^3, and the storage capacity is double binary.

5.1. The DNA Molecule

Deoxyribonucleic acid, or DNA, is a naturally occurring storage molecule that contains genetic information. It is the recommended resolution for processing vast volumes of data because of its enormous sequence, which is compacted into an exciting task [36]. The DNA sequence holds the expression information of many species [37] and [39], and its storage system is nearly identical to that of a digital CD, with information stored in the ground and mines indicated by 0 and 1 in the spiral footprint. [38] clearly shows potentiality of DNA like a hard disk.

In creatures, DNA has the same structure, consisting of two strands coiled in a helix. DNA double helix is made up of two strands, each of which is made up of nucleotides. Sugar (deoxyribose), phosphoric acid, and nitrogenous bases are the three constituents that make up each nucleotide. Adenine, which could be marked (A), thymine, which could be marked (T), cytosine, which could be noted (C), and guanine, which could be noted (G) are four types of nitrogenous bases with the chemical characteristics of bases [40].

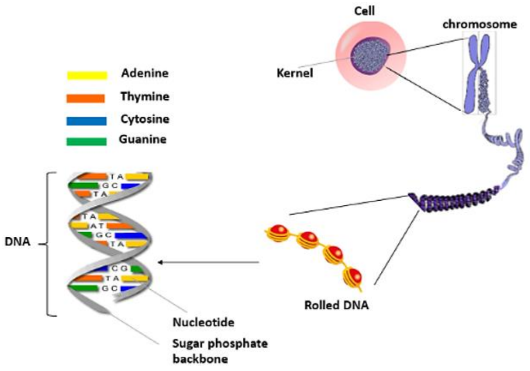

Figure 4. DNA molecule and double Helix structure

5.2. Data Storage on Synthesized DNA

Computer data is stored in binary form as 0 and 1 [51-53], but the information in DNA is stored as four basic components, not binary files. These components are adenine, thymine, cytosine and guanine, which are labeled A, T, C and G [41]. In most cases, the technique of saving digital data in DNA is carried out. in stages to transform that data to make it more suitable for DNA storage.

An encoder converts a binary string into a DNA oligonucleotide, a DNA synthesizer creates a strand that encrypts the data to be saved in DNA, a DNA sequencer reads the strand, and a decoder converts DNA Strings back into numerical data [42, 43]. The essential processes for saving and recovering numerical data to / from DNA storage are presented and detailed in this section.

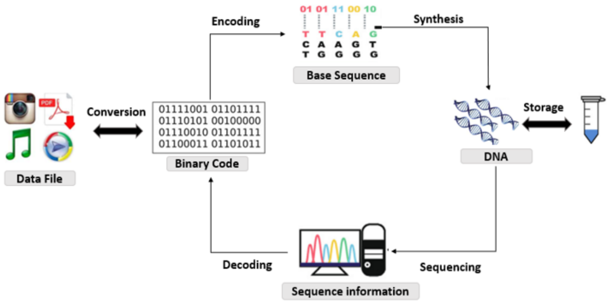

Figure 5. Overview of DNA data storage system

Data encoding into the DNA sequence: The first stage in storing DNA data is encoding. It turns readable or visible materials into a binary language comprising 0 and 1, and then uses computer techniques to translate that binary information into DNA nucleotide sequences utilizing the four bases (A, C, G, and T) instead of 0 and 1. Every base could be characterized by two bits, resulting in four distinct options and 16 potential DNA base pair combinations, for example (AT is 00, GC is 01, TA is 10, and CG is 16). [44] for 11.DNA synthesis (writing): After encoding binary data and producing DNA sequences, they should be written to DNA molecules, often in the form of 100 to 200 NT strands. Chemically produced single-stranded DNA sequences are possible [43, 44]. Each nucleotide is joined to the nucleotide next to it, according with numerical sequence data. Artificial DNA synthesis, on the other hand, has a 99 percent efficiency and a 1% mistake rate. This code-based protocol takes text files with lines and decodes the data into digital files that may be saved.

- DNA Sequencing: We presume that a tube has millions of distinct DNA strands. All of the DNA must be sequenced in order to retrieve the digital data and return it to its original form. The method of reading DNA sequences to digital sequences is known as DNA sequencing. Extract droplets from DNA tube and using PCR to increase the targeted DNA strand as the first step in reading the targeted DNA strand. In PCR, we need to introduce a certain primer pair and use it to repeat the DNA string [45].
- Decoding Information: Decoding is the final step in DNA storage. The sequence is generated and sent back to the decoder, which uses another computer algorithm with reverse coding function to decode the DNA sequence into binary language

5.3. DNA Storage Technology’s Potential

As previously said, we are confronted with massive data storage issues as the world's data rises dramatically. DNA can be employed as a prospective solution to these storage challenges because of its high density, replication efficiency, endurance, and long-term stability [46]. In 2011, IBM completed the construction of a complete data center with a storage ability of around 100 PB. Due to its high density, DNA can store a big quantity of data in a little amount of area as a data storage medium. At its theoretical maximum, one gram of DNA can store 200 PB of data, which is over double the capacity of IBM's entire data center [47].

DNA medium can store information for a long period due to its excellent resilience. Scientists read the DNA of a horse which had lived for 700,000 years in 2013. This DNA has been frozen in less-than-ideal conditions, yet it is still robust enough to be sequenced completely. DNA is stable at a broad range of temperatures, ranging from -800 to 800 °C [48]. As a result, DNA medium can sustain data integrity for an extended period of time.

5.4. DNA Storage Technology's Challenges

DNA may become a possible and promising medium for numerical data storage due to its unique features when compared to standard media [49]. However, Before DNA can be monetized, there is still a long way to go. High prices, low throughput, restricted access to data storage, short synthetic DNA fragments, and synthesis and sequencing error rates are only a few of the issues we confront [50].

5.5. Big Data Storage Options in the Future: Synthetic DNA or the REHDFS System

Synthetic DNA can address rising data storage demands since it has a large capacity, is nearly indestructible, and is energy efficient. It becomes the greatest contender to tackle future big data storing difficulties as a result of these properties. The table below compares the two systems (REHDFS and DNA) on a number of criteria and shows that DNA store is the better option.

Table 1. Comparison of the REHDFS storage and DNA storage

System / Properties	Synthetic DNA	REHDFS system
storage architecture	DNA data storage architecture is based on nucleotide sequences	HDFS clusters could include at least one name node, with storage spread over numerous data nodes
Information density	Density of DNA is about ten million times greater than that of the best traditional systems. DNA can in principle store half Zo of information per gram (g)	storage capacity can thus reach several petabytes
Longevity	The longevity of DNA is approximately ten thousand times that of traditional media. DNA molecules over 560,000 years old have been analyzed from historical samples	As a midrange hardware the system commonly has a lifespan of three to five years

6. CONCLUSION

The globe has seen rapid data expansion in recent years. The difficulty of large data storage is getting increasingly difficult with the introduction of big data, mobile apps, social media, and megadata analysis programs. IN order to properly determine the growth rate, the appropriate storage device must be selected. Capacity, performance, throughput, cost, scalability, and reliability are important factors in selecting an ideal storage solution system, as Big Data storage and management methods can significantly affect the entire organization. Among the solutions we need to focus on now are improvements to current technology, as well as finding more efficient alternatives solutions to address future Big Data storage challenges.

The REHFD (Random Enhanced HDFS) is one of the enhancements to HDFS. With the new components, REHDFS is able to perform other functions, such as random read and write operations, based on the two pessimistic and optimistic models detailed in this paper.

The second solution proposed in this paper is DNA storage technology. We can store big volumes of data in a little amount of space using persistent information storage systems. DNA data storage became one of the many forward techniques for long-term information storage due to its extraordinarily high density and long-term preservation. Unlike conventional storage, which limits the format in which encoded data can be stored. Two biological restrictions apply to the stored data in DNA storage system.

ACKNOWLEDGEMENTS

This work is supported by National Center for Scientific and Technical Research (CNRST), Rabat, Morocco.

REFERENCES

[1] B. Vladimir, T. Vladimir, N. Evgeny, "Comparative Characteristics of Big Data Storage Formats", Journal of Physics Conference Series, Vol. 1727, pp. 012005, January 2021.
[2] G. Alexander, I. Dmitry, N. Evgeny, "The Dataset of the Experimental Evaluation of Software Components for Application Design Selection Directed by the Artificial Bee Colony Algorithm", MDPI Journals, Vol. 5, No. 3, pp. 59, July 2020.
[3] G. Petushkov, "Evaluation and Reliability Prediction for Highly Reliable Software and Hardware Systems: The Case of Data Processing Centers", Russian Technological Journal, Vol. 8, pp. 21-26, Russia, March 2020.
[4] C. Min, M. Shiwen, L. Yunhao, "Big Data: A Survey", Mobile Networks and Applications, Vol. 19, No. 2, pp. 171-209, 2021.
[5] P. Russom, "Big Data Analytics," TDWI best Pract. report, fourth Quart., Vol. 19, No. 4, pp. 1-34, 2011.
[6] R. Menon, "Cloudera Administration Handbook", Packet Publishing, pp. 220-254, 2014.
[7] R. TRM, "HortonWorks Data Platform New Book", Horton Works Book, pp. 01-14, 2015.

[8] T. Dunning, E. Friedman, "Real-World Hadoop", O'Reilly Media Inc., pp. 53-72, 2015.

[9] Q. Dino, N.E. Arias, B.G. Pablo, F.C. Rodrigo Ceron, C.H. Luis Carlos, J. Peng, F.L. Franz, M. Peter, M.P. Ichsan, W. Joanna, W. John, "Front Cover Implementing an IBM InfoSphere BigInsights Cluster using Linux on Power", IBM redbooks promotions, pp. 125-129, 2015.

[10] Pivotal HD, "Pivotal HD Enterprise Installation and Administrator Guide", GoPivotal Inc, pp. 59-80, 2014.

[11] D. Sarkar, "Pro Microsoft HDInsight", Berkeley, CA Press, pp. 13-22, 2014.

[12] E. Shahverdi, A. Awad, S. Sakr, "Big Stream Processing Systems: An Experimental Evaluation", 35th International Conference on Data Engineering Workshops (ICDEW), pp. 53-60. 2019.

[13] K. Ouaknine, M. Carey, S. Kirkpatrick, "The Pig Mix Benchmark on Pig, MapReduce, and HPCC Systems", IEEE Int. Congr. Big Data, pp. 643-648, 2015.

[14] A.B. Moniruzzaman, A.S. Hossain, "NoSQL Database: New Era of Databases for Big data Analytics-Classification, Characteristics and Comparison", International Journal of Database Theory and Application, Vol. 6, No. 4, pp. 1-13, 2013.

[15] R. Chandrima, P. Manjusha, S. Siddharth, "A Proposal for Optimization of Data Node by Horizontal Scaling of Name Node Using Big Data Tools", The 3rd International Conference for Convergence in Technology (I2CT), pp. 1-6, 2018.

[16] H. Yin, D. Li, Z. Sun, T. Lin, "The Research on the Optimization of Data Storage of the Explosion-Proof Equipment Status Monitoring Based on Cloud Platform", 3rd International Conference on Information Science and Control Engineering (ICISCE), pp. 716-719, Beijing, China, 2016.

[17] V. Chatuporn, N. Natawut, "Improving Performance of Small-File Accessing in Hadoop", The 11th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp. 200-205, 2014.

[18] N. Makoto, K. Joichiro, L. Gil Jae, F. Jose, Y. Saneyasu, "File Placing Location Optimization on Hadoop SWIM", 6th International Symposium on Computing and Networking Workshops (CANDARW), pp. 516-519, 2018.

[19] J.P.L. Cox, "Long-Term Data Storage in DNA", TRENDS Biotechnol., Vol. 19, No. 7, pp. 247-250, 2001.

[20] R.N. Grass, R. Heckel, M. Puddu, D. Paunescu, W.J. Stark, "Robust Chemical Preservation of Digital Information on DNA in Silica with Error-Correcting Codes", Angew. Chemie Int. Ed., Vol. 54, No. 8, pp. 2552-2555, 2015.

[21] C. Luis, N. Jeff, S. Karin, "Molecular Digital Data Storage Using DNA", Nature Reviews, Genetics, Vol. 20, No. 8, 2019.

[22] J. Kamal, E. Fathy, A. Abdullah, "A Framework to Secure Big Data Storage", Journal of Computational and Theoretical Nanoscience, Vol. 14, pp. 5600-5605, 2017.

[23] P. Ghosh, J. Moorthy, "Big Data and Consumer Privacy", The Journal for Decision Makers, Vol. 40, No. 1, pp. 74-96, 2015.

[24] J. Kamal, E. Fathy, A. Abdullah, "A Framework to Secure Big Data Storage", Journal of Computational and Theoretical Nanoscience, Vol. 14, pp. 5600-5606, 2017.

[25] T. Subash, "Big Data Storage Analytics", International Journal of Computer Trends and Technology, Vol. 51, pp. 68-76, 2017.

[26] R. Lisbeth, R.E. Cristian Aaron, S.C. Jose Luis, C. Jair, G.A. Jorge Luisa, A.H. Giner, "A General Perspective of Big Data: Applications, Tools, Challenges and Trends", The Journal of Supercomputing, Vol. 72, No. 8, 2016.

[27] M.P. Neha, H. Mosin, D.S. Parth, "Improving HDFS Write Performance Using Efficient Replica Placement", The 5th International Confluence the next Generation Information Technology Summit, pp. 36-39, Noida, India, 2014.

[28] L. Bastien, "HDFS: Definition, Advantages and Disadvantages of Apache Hadoop System", Lebigdata Magazine, pp. 1-4, France, June 2021.

[29] K. Shvachko, H. Kuang, S. Radia, R. Chansler, "The Hadoop Distributed File System", IEEE 26th Symposium on MSST, pp. 1-10, May 2010.

[30] R.C. Veerabhadra, "REHDFS: A Random Read/Write Enhanced HDFS", Journal of Network and Computer Applications, Vol. 103, pp. 85-100, 2018.

[31] P.D. Gudadhe, A.D. Gawande, L.K. Gautham, "Enhance the Performance of Hadoop Distributed File System for Random File Access Using Increased Block Size", ACM SIGCOMM, Vol. 9, pp. 63-74, 2009.

[32] D.S. Kumar, M.A. Rahman, "Simplified HDFS Architecture with Blockchain Distribution of Metadata", International Journal of Applied Engineering Research, Vol. 12, No. 21, pp. 11374-11382, January 2017.

[33] A. Siddiqa, A. Karim, A. Gani, "Big Data Storage Technologies: A Survey", Frontiers Inf. Technol. Electron. Eng., Vol. 18, No. 8, pp. 1040-1070, 2017.

[34] G.M. Church, E.M. Rubin, S. Kosuri, "Next Generation Digital Information Storage in DNA", Science, Vol. 337, No. 6102, p. 1628, 2012.

[35] H.M. Kiah, G.J. Puleo, O. Milenkovic, "Codes for DNA Sequence Profiles", IEEE Trans. Inf. Theory, Vol. 62, No. 6, pp. 3125-3146, January 2016.

[36] L. Dixita, G. Manish, "Natural Data Storage: A Review on Sending Information from Now to then via Nature", arXiv:1505.04890 [cs, math], May 2015.

[37] Y. Cevallos, T. Luis, D. Inca, S. Nicolay, S. Ivone, Z. Amin, A.G. Guillermo, "On the Efficient Digital Code Representation in DNA-Based Data Storage", The 7th ACM International Conference on Nanoscale Computing and Communication, No. 18, pp. 1-7, New York, USA September 2020.

[38] R. Deaton, M. Garzon, R.C. Murphy, J.A. Rose, D.R. Franceschetti, S.E. Stevens, "Reliability and Efficiency of a DNA-Based Computation", Phys. Rev. Lett., Vol. 80, No. 2, p. 417, 1998.

[39] S. Manar, R. Najat, A. Jaafar, "Synthetic DNA as a Solution to the Big Data Storage Problem", Journal of Theoretical and Applied Information Technology, Vol. 99, No. 15, August 2021.

[40] M. Sais, N. Rafalia, J. Abouchabaka, "Synthetic DNA as a Solution to the Big Data Storage Problem", J. Theor. Appl. Inf. Technol., Vol. 99, No. 15, pp. 3912-3922, 2021.
[41] Y. Erlich, D. Zielinski, "DNA Fountain Enables a Robust and Efficient Storage Architecture", Science, Vol. 355, No. 6328, pp. 950-954, 2017.
[42] L. Andreas, S. Paul, W. Antonia, Y. Eitan, "Coding over Sets for DNA Storage", IEEE Transactions on Information Theory, Vol. 66, No. 4, pp. 2331-2351, 2020.
[43] S. Lichun, H. Jun, L. Jing, C. David, "DNA and the Digital Data Storage", Health Science Journal, Vol. 13, No. 3, pp. 8, 2019.
[44] C. Bancroft, T. Bowler, B. Bloom, C.T. Clelland, "Long-Term Storage of Information in DNA", Science, Vol. 293, No. 5536, p. 1763, 2001.
[45] F. Sanger, A. Coulson, "A Rapid Method for Determining Sequences in DNA by Primed Synthesis with DNA Polymerase", Sel. Pap. Frederick Sanger Comment, Vol. 94, p. 382, 1996.
[46] G.M. Church, Y. Gao, S. Kosuri, "Next Generation Digital Information Storage in DNA", Science, Vol. 337, No. 6102, p. 1628, 2012.
[47] O. Lee, et al., "Random Access in Large-Scale DNA Data Storage", Nature Biotechnology, Vol. 36, No. 3, pp. 242-248, 2018.
[48] A. Fatima, H. Ikram, A. Haider, L.A. Tahir, "Trends to Store Digital Data in DNA: An Overview", Molecular Biology Reports, Vol. 45, No. 5, pp. 1479-1490, 2018.
[49] S.M.H.T. Yazdi, Y. Yuan, J. Ma, H. Zhao, O. Milenkovic, "A Rewritable, Random-Access DNA-Based Storage System", Sci. Rep., Vol. 5, No. 1, pp. 1-10, 2015.
[50] M. Blawat, et al., "Forward Error Correction for DNA Data Storage", Procedia Computer Science, Vol. 80, pp. 1011-1022, 2016.
[51] A. Moumen, A. Lakhdar, K. Mansouri, "Elastoplastic Behavior of Polybutylene Terephthalate Polyester Bio Loaded by Two Sustainable and Ecological Fibers of Animal Origin with two Numerical Methods", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 46, Vol. 13, No. 1, pp. 29-37, March 2021.
[52] A. Lakhdar, A. Moumen, K. Mansouri, "Study of the Mechanical Behavior of Bio Loaded Flexible PVC by Coconut and Horn Fibers Subjected to Aging", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 46, Vol. 13, No. 1, pp. 75-80, March 2021.
[53] H. Chaiti, A. Moumen, M. Jammoukh, K. Mansouri "Numerical Modeling of the Mechanical Characteristics of Polypropylene Bio-Loaded by Three Natural Fibers with the Finite Element Method", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 49, Vol. 13, No. 4, pp. 45-50, December 2021.

BIOGRAPHIES

Manar Sais was born in 1996 in Fez. She received his Master's degree in computer science, big data cloud computing from Ibn Tofail University, Kenitra, Morocco. She is a Ph.D. student in Computer Research Laboratory (LaRI) of the same university. Her research interests include big data, data storage, cloud computing, distributed computing.

Najat Rafalia was born in Kenitra, Morocco, 1968. She has obtained three doctorates in Computer Sciences from Mohammed V University, Rabat, Morocco by collaboration with ENSEEIHT, Toulouse, France, and Ibn Tofail University, Kenitra, Morocco. Currently, she is a Professor at Department of Computer Sciences, Ibn Tofail University, Kenitra, Morocco. Her research interests are in distributed systems, multi-agent systems, concurrent and parallel programming, communication, security, big data, and cloud computing.

Jaafar Abouchabaka was born in Guersif, Morocco, 1968. He has obtained two doctorates in Computer Sciences applied to mathematics from Mohammed V University, Rabat, Morocco. Currently, he is a Professor at Department of computer Sciences, Ibn Tofail University, Kenitra, Morocco. His research interests are in concurrent and parallel programming, distributed systems, multi agent systems, genetics algorithms, big data and cloud computing.