Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 201 (2022) 746–751

Procedia Computer Science

www.elsevier.com/locate/procedia

The 3rd International Workshop on Statistical Methods and Artificial Intelligence (IWSMAI)
March 22 – 25, 2022 Porto, Portugal

Intelligent Approaches to Optimizing Big Data Storage and Management: REHDFS system and DNA Storage

Manar Sais[a,]*, Najat Rafalia[a], Jaafar Abouchabaka[a]

[a]*Computer Research Laboratory LaRI, Faculty of Sciences Ibn Tofail University, Kenitra, Morocco*

Abstract

In the last few years, we have witnessed unprecedented growth in data and gigantic amounts of data are being produced every day. By 2020, the amount of information we want to store it will be around 44 trillion gigabytes. on the one hand the data volumes continue to grow at an even higher speed, However, our traditional databases are limited in the storage and processing of this large and complex data and we do not have a reliable physical storage medium that can withstand the weather. On the other hand, the term Big Data is now the new natural resource and current analysis architectures face much greater challenges in terms of scalability, rapid ingestion, performance, processing and storage efficiency. In order to cope with these massive and exponentially increasing amounts of heterogeneous data generated more and more quickly, many researchers believe that they have found the solution to this problem, either develop and add an intelligent touch to the available technology (REHDFS), or have discovered solution effective in the field of chemistry, DNA for example. The objectives of this paper are to present the two future storage solutions with the advantages and values-added by both approaches.

Keywords: Storage; Big Data; REHDFS; DNA.

1. Introduction

Global digital content is growing rapidly and is expected to reach 35 zettabytes over the next decade. A huge amount of data is generated by social networking websites/applications such as Facebook, Twitter, and WhatsApp,

* Corresponding author. Tel.: + 212-689-506-933;
E-mail address: manar.sais@uit.ac.ma

10.1016/j.procs.2022.03.101

as well as the growing number of sensors used. The vast amount of data continuously collected from these sources contains emerging market/environment trends. If most organizations have access to information about emerging trends, they can make informed decisions. Organizations that traditionally used data mining tools to analyze structured historical data are now investing heavily in developing tools to analyze recent, unstructured, and rapidly growing data to understand popular market trends and portend customer behavior. The insights gained from using these tools allow them to improve decision-making processes, rectify current processes, and adjust policies to better compete in the marketplace. These storage and analysis requirements are a challenge for traditional data storage tools and a huge challenge for relational databases. Therefore, the evolution of big data storage and analytics tools is highly desirable.

This paper is dedicated to the analysis of two approaches to improve the storage and management process of Big Data. The first solution is an enhanced HDFS (REHDFS), which explores different block placement strategies, providing a scalable architecture and a load-based block access strategy. Moreover, two models one pessimistic and the other optimistic implement the random read and write features. The second solution, DNA strands [5], has demonstrated an amazing ability to store thousands of gigabytes completely reliably and may be the solution for storing maximum data in minimum space.

2. Related work

The volume of vast and complicated data with various structures is gradually expanding, far beyond the capacity of traditional storage devices, thanks to increased Internet traffic and the rapid development of the information business. With noisy, sparse, and heterogeneous data, big data storage has become a major concern. New technologies that can store massive volumes of data are likewise in high demand [7].

On Hadoop Statistical Workload Injector for MapReduce, the authors of [8] offer an efficient file placement approach (SWIM). The method uses real workloads to thoroughly analyze a method for efficient file placement in storage in order to increase I/O performance in Hadoop. Various I/O scenarios for some SWIM operations are investigated. Then look at various SWIM tasks' I/O patterns.

Researchers have devised a novel method of storing data in order to address storage needs. One of the novels approaches to encoding information in DNA is a procedure known as genetic data storage. The main goal of this study [11] is to introduce DNA as a good data storage technique and to overcome the two major issues that it has.

3. Intelligent approach for optimal storage and management of big data: The REHDFS system

Hadoop Distributed File System (HDFS) is a highly scalable and fault tolerant data storage system. It provides cost-effective and reliable storage capacity and supports up to hundreds of nodes in a cluster. With this capacity, HDFS can accommodate large amounts of data (i.e., stored files can exceed 1 TB) and can handle both structured and unstructured data [10]. The content stored in an HDFS system is divided into blocks that are replicated across multiple data nodes to improve the throughput and access time, and make the system highly fault tolerant. One limitation of HDFS is that it only allows users to perform sequential reads and write, not random reads and writes. This section focuses on presenting an improved HDFS (REHDFS), with new components added to the HDFS architecture, the system explores different strategies for placing and accessing data blocks, as well as pessimistic and optimistic models for performing random read and write functions [1].

This part is dedicated to representing the HDFS system and its additional features compared to the traditional HDFS system and also presents the optimistic and pessimistic models used to achieve the random write operation.

3.1. HDFS: architecture and functionality

The open-source Hadoop is one of Google's implementations of the solution. Leading technology businesses such as Facebook, Amazon, Twitter, and others use it. Hadoop consists of two parts: a storage system called Hadoop Distributed File System (HDFS) and a processing system called MapReduce [10].

One of the fundamental components of the Apache Hadoop framework, especially its storage system, is the distributed file system (HDFS), which can store and retrieve files in record time. The HDFS storage system is very suitable for Big Data because of its massive capacity and reliability.

HDFS is a master-slave architecture that employs clusters to store and manage large amounts of data. A cluster consists of a master node (NameNode) that oversees file system operations and many of slave nodes (DataNodes) that manage and coordinate data storage across several compute nodes. Hadoop uses data replication to assure data availability [2].

3.2. The features added by the REHDFS system

HDFS divides each stored file into blocks and stores each block on many data nodes to ensure fault tolerance. When creating new files, HDFS utilizes the default block size and lets users to alter it. When implementing the HDFS architecture, different block placement strategies can be tested. HDFS systems, on the other hand, do not support random read and write operations. The new features provided by the REHDFS system:

- Block placement strategies.
- Choosing the block size.
- Block caching and retrieval strategies.
- Data Node selection strategies.
- Random read and write

3.3. REHDFS Architecture and Features

The REHDFS architecture is shown in the figure above. Its main components are a name node, a set of data nodes, a client module, a cache module, and a lock / validation manager.

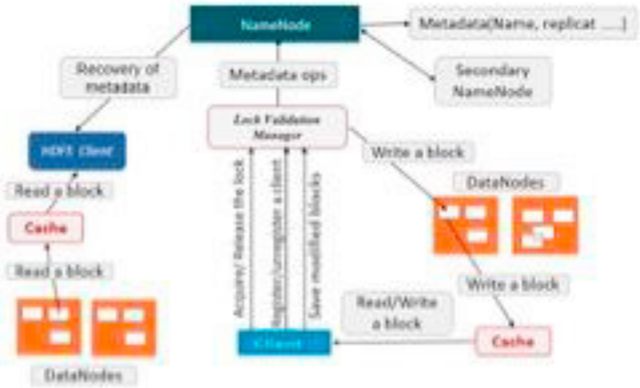

Fig. 1. Extended HDFS architecture based on RMI

In order to achieve additional functionality and support random write operations, a new component called Lock / Validation Manager is added to the HDFD architecture, which supports two models for the implementation of the operation of random writing, one pessimistic model and another optimistic.

3.4. Random Write: File Level Lock with a Pessimistic Model (FLPM)

As illustrated in figure 2, a file in a pessimistic model can be in one of three states: closed state, open state, and locked state.

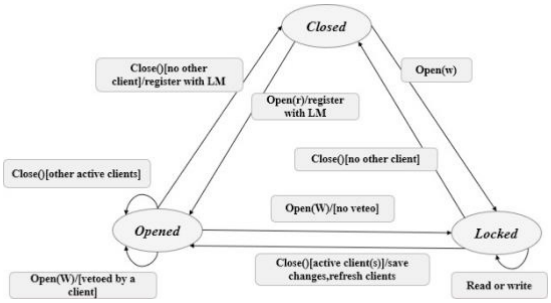
Closed
Close()[no other client]/register with LM
Open(w)
Open(r)/register with LM
Close()[no other client]
Close()[other active clients]
Open(W)/[no veteo]
Opened
Locked
Close()[active client(s)]/save changes,refresh clients
Open(W)/[vetoed by a client]
Read or write

Fig .2. State diagram of a file in a pessimistic model.

In the pessimistic model, each file starts with a closed state. The state of a file changes from closed to open (locked) when a file is opened by a client, and the client is registered by the lock manager (LM). The figure shows the file state changing based on client actions and requests [1].

3.5. Random Write: Optimistic model with file level consistency (OMFC)

As shown in figure 3, a client of the optimistic model can be in one of four states: inactive, register (register with the validation manager), update (by modifying the read blocks), and record (trying to save the modified blocks). the following diagram describes the states of a file in the optimistic model.

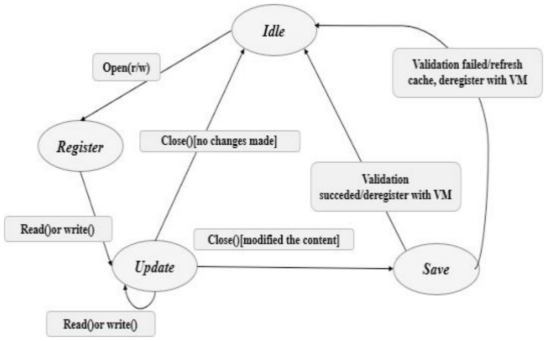
Idle
Open(r/w)
Validation failed/refresh cache, deregister with VM
Register
Close()[no changes made]
Validation succeded/deregister with VM
Read()or write()
Close()[modified the content]
Update
Save
Read()or write()

Fig. 3. The state diagram of a customer in the optimistic model

4. Biological storage: DNA genetic storage media

Computer data storage volume is one difficulty, but data persistence through time is another. All of the storage technologies we've built so far have finite life cycles and are fragile. As a result, there is an increasing demand for computer media that can store vast amounts of data in a long-term way. Because of its endurance and high density of information, the idea of biological storage of information via DNA comes into play to address these challenges and this continuing need, and it is seen as an appropriate storage medium of choice for the next generation [9][12].

4.1. Data storage process on synthesized DNA

The data in a file must first be translated from 1 to 0 into A, C, T, and G before it can be transcribed into the DNA. The encoding software module is responsible for this translation and for the addition of the error correction in the sequence of the load. Additional bases are added to the sequence to verify that its main and secondary structures are consistent with the reading process, and the DNA sequence is submitted to the synthesis module to be instantiated into Physical DNA molecules [4].

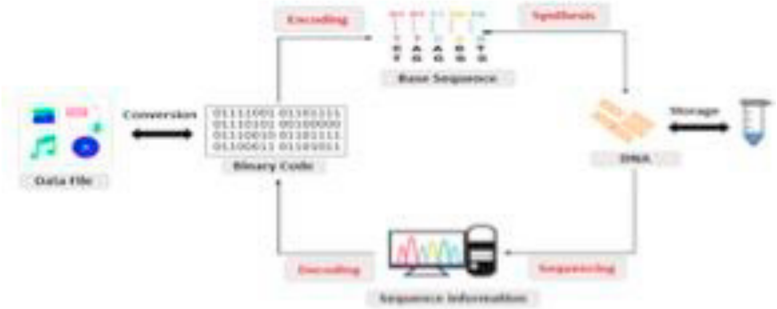

Fig. 4. Overview of DNA data storage system

4.2. Possible approach to the DNA computing for big data

The process of storing information in the DNA molecule goes through two stages: Process and Encoding / Decoding algorithm [5].

- **Encoding algorithm**

```
1) Read data stream: A
2) Check size of the data [r,c,n] = size (A) where r=rows, c=columns, n=number of matrix
3) Calculate DNA sequences size for image
4) Create a zero matrix of DNA sequences size
5) While DNA sequences size = max
        a. Convert even smallest piece of data to binary form.
        b. Insert binary DNA code of an individual data cell to DNA sequence.
        c. Continue till all of max size of DNA sequence is reached
```

- **Decoding algorithm**

```
1) Read DNA sequence
2) Calculate size: length, size of individual cells Convert DNA to real data decode
        a. Convert one cell to data
        b. Insert the converted value to template data matrix
        c. Stop when entire DNA is converted
```

4.3. Error correction code

Several types of errors may occur during the process of storing data in DNA, including insertion, deletion, and substitution errors in oligonucleotides, missing DNA strands, and synchronization errors on multiple oligonucleotides with the same address.

Several levels of error protection are required to overcome these issues and design a viable system. To identify each oligonucleotide, the system assigns it a unique address code. Digital data is stored on multiple oligonucleotides and is encrypted using modern two-dimensional matrix codes. While recovery accuracy is critical, these codes must also be effective in lowering the overhead costs related with DNA synthesis and sequencing [6].

5. Conclusion and Future Work

The current exponential rate of data production has created a demand for improved storage devices, and new options for keeping our data in a sustainable manner are becoming vital. As a result, more data can be stored faster, more sustainably, and with less energy. Researchers have shifted their focus to new methodologies with the purpose of adapting and improving technology and accessible tools or inventing new, more effective, and efficient methods. In this paper, we offered a brief summary of the various options available today, how they work, and, most importantly, their added values.

The first approach is to improve the HDFS storage system, resulting in a more powerful REHDFS. The system can now execute more jobs thanks to the algorithms. Random writing based on two optimistic and pessimistic models is the most significant task addressed in this study. The difference between the two random writing models is that in the pessimistic model (FLPM), a client must first acquire a lock on a file before being able to edit it. The order in which the required locks are requested can cause a deadlock when a client group wishes to alter a series of files. On the other hand, the optimistic model allows all clients to edit the blocks in their caches, with the client with the minimum timestamp being able to save their changes. When numerous clients are executing on the same file, the OMFC model outperforms the FLPM model. The REHDFS will be enhanced in the future to provide more comprehensive random write operations while also improving security.

The second method involves using DNA memory strands. We were able to store a high volume of data on a very tiny volume thanks to the solution for long-term information storage. If DNA is preserved in dry, dark, and cold conditions, information encoded in it can be recovered even thousands of years afterward. The data is kept in a lengthy virtual DNA molecule once it has been encoded and decoded in binary data to and from generated short DNA strands. Data storage in DNA medium is simple, convenient, and cost-free, and it can keep data fidelity for a long time. However, this technology is still highly expensive, and it requires a huge team and high-end hardware.

Acknowledgements

This work was supported by National Center for Scientific and Technical Research (CNRST).

References

[1] Rao Chandakanna, V. (2018). “REHDFS: A random read/write enhanced HDFS.” Journal of Network and Computer Applications **103**,85-100. Doi:10.1016/j.jnca.2017.11.017

[2] Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2018). “Big Data technologies: A survey”. Journal of King Saud University-Computer and Information Sciences, 30(4), 431-448.

[3] O' Driscoll, A., & Sleator, R. (2014). “Synthetic DNA The next generation of big data storage”, Department of Computing; Cork Institute of Technology, Ireland.

[4] Takahashi, C., Nguyen, B.H., Strauss, K., & Ceze, L. (2019). “Demonstration of End-to-End Automation of DNA Data Storage”. Scientific reports **9(1)**, 4998.

[5] Hanadi, A.H., Zenon, C., & Anup, K. (2015). “Review of Big Data Storage based on DNA Computing”. Asia-Pacific Conference on Computer Aided System Engineering. 113-117. Doi: 10.1109/APCASE.2015.27

[6] Blawat, M., Gaedke, K., Hütter, I., Chen, X.M., Turczyk, B., Inverso, S., Pruitt, B.W., & Church, G.M. (2016). “Forward Error Correction for DNA Data Storage”. Procedia Computer Science. The International Conference on Computational Science **80**, 1011–1022.

[7] Roy, C.,Pandey, M., & SwarupRautaray, S.(2018). “A Proposal for Optimization of Data Node by Horizontal Scaling of Name Node Using Big Data Tools”. 3rd International Conference for Convergence in Technology (I2CT), 1-6, doi:10.1109/I2CT.2018.8529795.

[8] Nakagami, M.,Kon, J., Lee, G.J.,Fortes, J.A.B., & Yamaguchi, S.(2018). “File Placing Location Optimization on Hadoop SWIM”. Sixth International Symposium on Computing and Networking Workshops (CANDARW),516-519, doi:10.1109/CANDARW.2018.00100.

[9] Sais,M., Rafalia, N., Abouchabaka, J.(2021). “Synthetic DNA as a solution to the Big Data storage problem”. Journal of Theoretical and Applied Information Technology **99(15)**, 3912-3922. Doi: 10.5281/zenodo.5353710.

[10] O’Driscoll, A., Daugelaite, J., & Sleator, R. D. (2013). “Big data, Hadoop and cloud computing in genomics”. Journal of biomedical informatics, 46(5), 774-781.

[11] Cox,J. P.(2001)."Long-term data storage in DNA". Trends in Biotechnology, 19(7),247-250, doi: 10.1016/s0167-7799(01)01671-7.

[12] Sun, L., He, J., Luo, J., Coy, D.V. (2019). “DNA and the Digital Data Storage”. Health Science Journal, **13(3)**,8.