EXPLOITATION OF HADOOP AND ELK STACK TO MANAGE BIG DATA

Andriavelonera Anselme A.[1], Rivosoaniaina Alain N.[1], Mahatody Thomas[2], Manantsoa Victor[2].

[1]Laboratory for Mathematical and Computer Applied to the Development Systems, University of Fianarantsoa, Madagascar.

[1]Laboratory for Mathematical and Computer Applied to the Development Systems, University of Fianarantsoa, Madagascar.

[2]Professor on the University of Fianarantsoa, Madagascar.

[2]Professor on the University of Fianarantsoa, Madagascar.

ABSTRACT

Today, hundreds of millions of computers and mobile devices are constantly generating a lot of data. On social networks, the sharing and publishing of information by users is constantly increasing. As a result, the amount of data almost exceeds the capacity of the storage server. The volume of data is growing faster, in the storage terms; it would be necessary to share this data through a network of several machines using a distributed file system like hadoop. This tool uses an HDFS or Hadoop Distributed File System which is a distributed, scalable and portable file system inspired by the Google File System or GFS. This paper presents an improvement on managing big data in Hadoop, and ELK for data storage services.

KEYWORDS

Server, Storage, Hadoop, Database, HDFS & ELK

1. INTRODUCTION

With the explosion of data generated by billions of users and hundreds of millions of computers connected to the internet, the big data exploitation has become a new opportunity for the enterprise. On the web, the proliferation of social networks is happening too fast, with users able to publish and share any type of content. From which, the big data management brings with it new challenges. To remedy this, enterprises are obliged to improve their IT architecture and datacenter to optimize in-depth information analysis. It has several tools to manage these big data such as hbase, Mongodb, cassandra ... The Hadoop Framework created by Doug Cutting in 2009 is used by major players such as Google, Amazon, Yahoo, Facebook, eBay and IBM, largely for applications involving search engines [2],[13],[14]. According to the article published by Amit Pathak, the active user of facebook reaches the number of 1.96 billion and more than 500 terabytes of data generated per day [15].

1.1. Objective and problematic of the research

Faced the potential flow of large volumes of data that also occupy Web 2.0 services, we want the deployment server to be always available to respond to user requests. To avoid overloading at a disk level, the storage capacity of the server must be monitored in real time.

The Hadoop database ensures that some of the issues related to data management and file system are resolved. Also the management of fault tolerance at the time of operation and data loading. Relative to the existence of different tools and technical on big data management. How can to improve big data storage by exploiting Hadoop and ELK stack?

1.2. Contribution

In this article, firstly, we propose the use of Hadoop database to provide big data management. Hadoop is a very powerful tool to manage the NoSQL database. It allows us to storage more than terabytes of data. This type of technology is also necessary when it comes to recover unstructured data. In view of the known difficulties during data recovery of such a volume on the web, the system administrator must systematically ensure data backup before and after processing. Indeed, in our approach, we have exploited Logstash to store in the log all the input data before passing them to Hadoop. And to avoid losing data in the face of various risks or technical incidents at the system or hardware level, we have implemented the policy on the exploitation of high availability in order to manage the server's fault tolerance.

2. STATE OF ART

In this section, we are going to address the Hadoop Distributed File System, the storage management and the data access control by ELK.

2.1. Hadoop Distributed File System - HDFS

HDFS file systems used by Hadoop provide storage for large data, despite its disadvantages needs to use other application to mount a hard drive. According to Jeffrey Dean, in this research work, the runtime system takes care of the details of partitioning the input data, scheduling the execution of the program on a set of machines, handling machine failures, and managing the required inter-machine communication [2]. The HDFS system is defined by the NameNode and the DataNode. The central point of HDFS is the NameNode. It keeps track of where the file data is kept on the cluster. The directory tree of all files in the system is also kept. When the client want to locate a file, the client applications send a request to the NameNode and it responds the DataNode address corresponding [7]. The NameNode loads the edit log and returns it to update the metadata loaded into memory in the previous step and then update the image file using with the current HDFS status information. The paper published by H. Dai reports the research on the metadata services in large-scale distributed file systems, which is conducted from three indicative perspectives that are still used to characterize SFNs: high scalability, high performance, and high availability. The focus is on their respective key challenges as well as their developed consumer technologies [8]. The NameNode starts running with a new empty edit file. The DataNode connect to the NameNode and sends it block reports listing all the data blocks stored by a DataNode. When the HDFS client want to read or write data to the root from the DataNode, the client contact the NameNode for block information and the empty blocks connect as a pipeline [3]. The client write the data into the first block, which is copied into a second block, and so on until the end of the pipeline. When a delete operation is performed, the physical link of the block is broken. Understanding this replication mechanism is very critical to improving big data storage services. Yeturu Jahnavi's article presents that weblog data analysis should be handled in a distributed environment due to its huge volume nature and also online streaming data generation [1].

2.1.1 HDFS System Components

NameNode Roles:

- HDFS information server for all client machines;
- Responsible for block distribution and replication;
- In case of read, the NameNode manages the list of blocks per file and it encompasses the list of DataNodes per block during read;
- Records metadata logs, transactions, and centralizes the location of data blocks distributed throughout the cluster;
- Provides metadata storage and management;
- Starts the system from an HDFS image;
- Reads and writes.

DataNode Roles:

- Block data server in HDFS format for all client machines;
- Stores the data blocks in the file system;
- Synchronizes the Heartbeat with the Namenode, a system that makes it possible to exploit the clustering of several servers in order to achieve fault tolerance between them. In this case, the Heartbeat process will pass information such as a message to the NameNode;
- Organizes the metadata about the owned blocks.

The figure 1 below shows the principle of storing two different files (file 1 and file 2) that are broken down into blocks where each block is replicated twice:

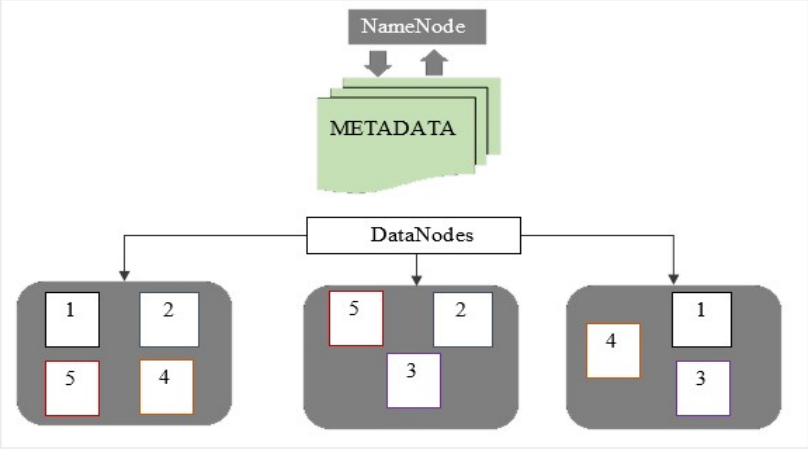

Figure 1. HDFS Architecture

In this architecture, a) The NameNode store the metadata only, b) The two files are distributed in the tree Datanodes, and, c) the repartition of two files will change with the block size configuration. Secondary NameNode:

The limitation of HDFS is that it can only scale the data nodes and not the name node where metadata is managed [9]. Despite the NameNode in the Hadoop architecture being a single point of failure to the system. If the NameNode is not functional, there is no way to be able to extract blocks from a file. Hence the existence of a Secondary NameNode service implemented in the Hadoop System to avoid this constraint.

Secondary NameNode Roles:

- Provides systematic loading of records into the log and provides update transfer of the latest log to the NameNode;
- Checks if the file system metadata check-in is completed;
- Storage the copy of the file and create a new image.

2.2. Data storage and access control

To manage the data of Enterprises, it is necessary to provide a real-time monitoring and protection system on the status of the system used. In addition, the data storage environment must be secure. For this reason, the data access management policy must be applied according to the user accounts. The ELK or Elasticsearch, Logstash and Kibana are the most used monitoring and storage tools by large companies. Without forgetting to observe the read and write time on the disk. In his article, Mohammad Nurul Islam present a solution by using many NameNodes. To reduce the time of reading and writing, the author present that it is more efficient to implement a load balancing system as a solution to the NameNode bottleneck problem [6]. All the same, the Datanode of the Hadoop System records the operations of the HDFS automatically in a log file. In a computer park or Datacenter, the instant arrival of logs to the server requires a tool to manage and monitor it, especially large enterprises. In any case, the System administrator must collect all the log information to facilitate the follow-up and monitoring of the system. To control access to data, Sanla present the advantages of the Kibana tool below in his article [11]:

- Elasticsearch Fully Integrated Visualization Tool;
- Provides real-time analysis, mapping, summarization, and debugging capabilities;
- An instinctive, user-friendly interface;
- Allows you to share snapshots of searched logs, save the dashboard, and manage multiple dashboards.

For this reason, the research realized by Aviecenna Yudhistira, seeks to design a Monitoring Log Server using the ELK (Elasticsearch, Logstash and Kibana), which can make it easier to read and analyse a service logs on the server [10]. Using the LogStash tool, we can analyze any type of log file by writing corresponding queries to sort out the relevant information. The content of the accessed log will be stored in the ElasticSearch database. However, the Hadoop framework allows for faster storage and processing of large volumes of data and is an easy tool to combine with other applications that process large amounts of data. According to Subhi R. M. Zeebaree and other many authors, using Hadoop, we can process, count and distribute each word in a big file and know the number of affects for each of them. In this paper, they explain what Hadoop is, its architectures, operation and performance in a distributed system also facilitates processing [4]. Recently, another work, realized by Neelesh Mungoli, we have presented the critical aspects of data storage and the management in cloud-based of AI systems, discussed the data pre-processing, feature engineering, privacy and security [12].

2.2.1 High availability and fault tolerance

A fault-tolerant configuration usually consists of a system capable of continuing to operate transparently in the event of the failure of one or more components. However, the NameNode is absolutely fundamental to the HDFS system, the Hadoop framework offers a High availability configuration in which there are two other NameNodes as backups, able to take over immediately, in case of failure of the initial NameNode. The rescue NameNodes function as a clone. They are in a standby state and are continuously updated using services called JournalNodes. The rescue NameNodes accomplish as much of the same task as the Secondary NameNode, to keep the system on the up states. According to Anurag Bhatnagar, the primary NameNode functions as a master for the DataNodes while the Secondary NameNode will keep track of the live status of the Primary NameNode. When the primary NameNode goes down, the secondary NameNode will change its IP to the IP of the primary NameNode, then it will become the master of the DataNodes. In this way, one can achieve high availability in older earlier versions of Hadoop [16].

The following figure is an illustration of how each component works, as explained in section 2.1.1

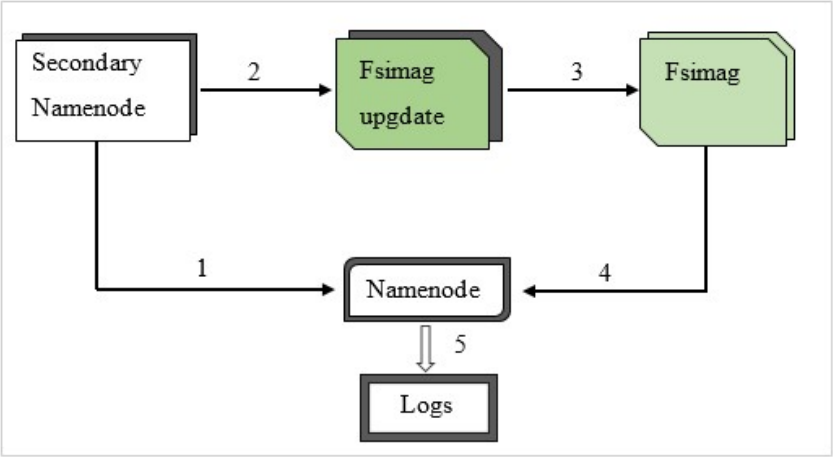

Figure 2. Secondary NameNode function

To prevent on the failed of DataNode, the Heartbeat service must be deployed to support distributed replication at the same time. In the event of a problem or failure of the NameNode, the backup of the logs on an external physical medium must be provided to make it easier to restore the system.

3. EXPERIMENTATION AND APPROACH

3.1. Fault tolerance management and high availability

To test the fault tolerance, we performed the tests on the data size of 59.13 MB by configuring a slave node. The table below summarize the processing execution time:

Table 1. Runtime on high availability.

File size (in MegaBytes)	Number of slave nodes	Execution time (in seconds)
59,13	1	122,14
	2	93,31
	3	100,11
	4	48,45

After having tested on different number of nodes, on the same file size, we observed that the execution time is disturbed.
The figure 4 below shows the results of data processing at the Hadoop database level:

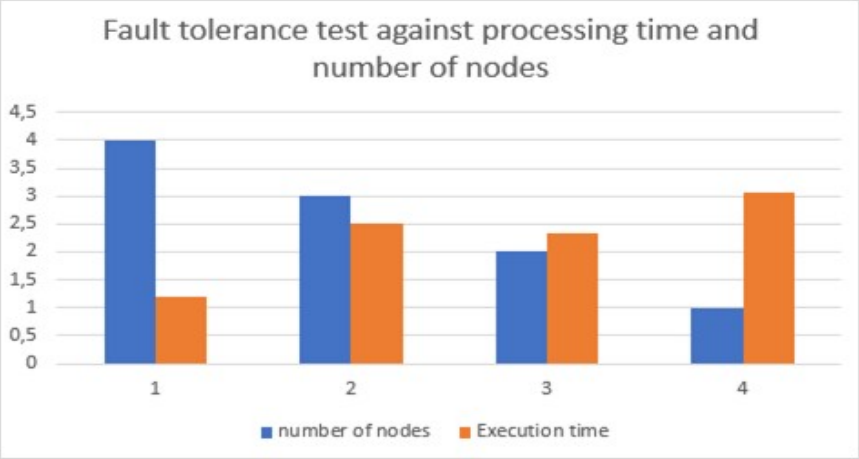

Figure 3. Fault tolerance testing.

During the test, the monitoring of the YARN administration tool shows the failure of the main Namenode, even if the processing at a node cluster is disrupted, it remains functional, which implies the high fault tolerance, hence the need to exploit the high availability policy of distributed systems such as Hadoop Distributed File System or HDFS. The proposition of the new approach is presented in Figure 3 below:

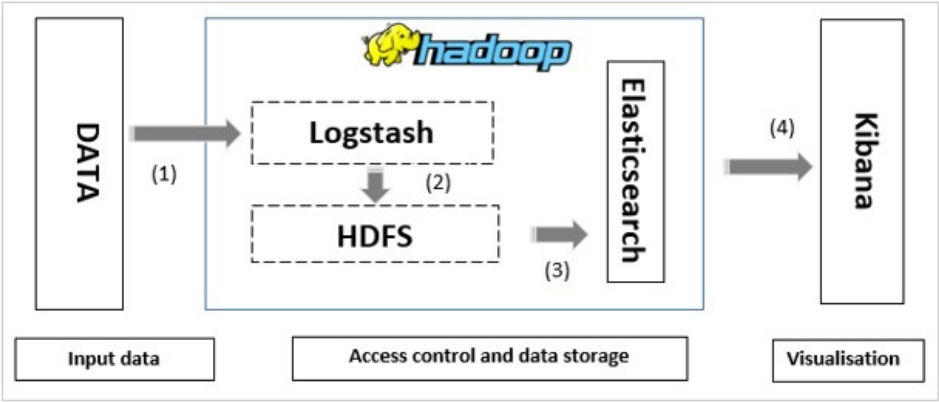

Figure 4. Approach of the data management

(1): User data entry.
(2): Data passage into Logstash as a data pipeline that migrates to HDFS and stores the data in the system.
(3): Data Transfer to the HDFS for indexing in Elasticsearch.
(4): Kibanna visualization interface used to monitor data stored in Elasticsearch.

4. RESULTS

After the experimentation, we obtained the following results.

4.1 Fast data storage:

- Terabytes of data can be accessed in minutes;
- Reading the data across multiple nodes is done faster by HDFS file systems;
- A large amount of data is divided into multiple machines in a cluster that is processed in parallel;
- Each DataNode processes a small amount of data, resulting in low traffic in a Hadoop cluster.

4.2 High availability and fault tolerance of a cluster:

- The fault tolerance is provided by the Heartbeat tool;
- In case of NameNode failure, log backup will be performed by the Logstash to reassure on restarting the last image of an HDFS system;
- Kibanna's interface simplifies the system administrator's tasks by visualizing and monitoring the storage server for onload massive data.

5. CONCLUSION

In conclusion, many tests have been performed to understand the operation of HDFS storage system in dealing with big data. From the experimental results, we have observed that the Hadoop database can store and retrieve the big data faster except in the case of processing multiple data or small files. Data access control is provided by implementing the ELK stack, which enhances security at the core of the HDFS system. In this paper, we have exploited Logstash to immediately synchronize the NameNode log file in the event of a system crash. However, deploying the hadoop framework requires experience of the necessary configurations, as well as the use of high-availability and fault-tolerant services. In fact, systematic backup of this system is recommended. In the future work, we intend to continue this research, by studying the load balancing of big data in a cluster of the Hadoop database nodes.

REFERENCES

[1] Yeturu Jahnavi, Y. Pavan Kumar Reddy, V. S. K. Sindhura, Vidisha Tiwari & Shaswat Srivastava, 2023, “A Novel Processing of Scalable Web Log Data Using Map Reduce Framework”, *Computer Vision and Robotics.*

[2] Jeffrey Dean & Ghemawat, S., 2004, “MapReduce : Simplified data processing on large clusters” *Google, Inc.*

[3] Shafer, J., Rixner, S., & Cox, A. L, 2010, "The hadoop distributed filesystem : Balancing portability and performance", *IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS), 122–133.*

[4] Subhi R. M. Zeebaree, Hanan M. Shukur, Lailan M. Haji, Rizgar R. Zebari, Karwan Jacksi & Shakir M.Abas, 2020, "Characteristics and Analysis of Hadoop Distributed Systems", *TRKU.*

[5] Rao, B. P., & Rao, 2019, "HDFS Logfile Analysis Using ElasticSearch, LogStash and Kibana", *In Integrated Intelligent Computing, Communication and Security.*

[6] Mohammad Nurul Islam & Md. Nasim Akhtar, 2019, "Improved Time Complexity and Load Balance for DFS in Multiple NameNode", *International Joint Conference on Computational Intelligence.*

[7] Tumpa Rani Shaha, Md. Nasim Akhtar, Fatema Tuj Johora , Md. Zakir Hossain , Mostafijur Rahman & R. B. Ahmad, 2019, "A noble approach to develop dynamically scalable namenode in hadoop distributed file system using secondary storage", *Indonesian Journal of Electrical Engineering and Computer Science.*

[8] H. Dai, Y. Wang, K. B. Kent, L. Zeng & C. Xu, 2022, "The State of the Art of Metadata Managements in Large-Scale Distributed File Systems Scalability, Performance and Availability*," in IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 12, pp. 3850-3869.*

[9] Praveen M Dhulavvagol, S G Totad, 2023, "Performance Enhancement of Distributed System Using HDFS Federation and Sharding", *Procedia Computer Science.*

[10] Aviecenna Yudhistira & Aviecenna Yudhistira, 2023, "Monitoring log server dengan elasticsearch, logstash dan kibana (ELK*) ", RABIT: Jurnal Teknologi dan Sistem Informasi Univrab*

[11] Sanla & Numnonda, 2019, "A Comparative Performance of Real-time Big Data Analytic Architectures'', *IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), 1–5.*

[12] Neelesh Mungoli, 2023, "Scalable, Distributed AI Frameworks: Leveraging Cloud Computing for Enhanced Deep Learning Performance and Efficiency'', *arXiv.*

[13] Doug Cutting, 2014, "The actual Hadoop elephant the project is named after, and a need for better security", *The Register.*

[14] Martin LaMonica, 2018, "Amazon Web Services adds resiliency to EC2 compute service", *CNet News.*

[15] Amit Pathak, 2022, "Relationship between Facebook and Big Data", *Analytics Vidhya.*

[16] Anurag Bhatnagar, Venkatesh Gauri Shankar, Bali Devi & Nikhar Bhatnagar, 2021, "An Efficient Model for High Availability Data in Hadoop 1.2.1", *Networks and Systems book series.*

Author

ANDRIAVELONERA Anselme Alexandre, Laboratory for Mathematical and Computer Applied to the Development Systems, University of Fianarantsoa, Madagascar.

Ten years of experiences in System and Network Administration

IT Instructor on the studies direction, INSCAE Madagascar