An Efficient Assessment on Fault Tolerance File System for Big Data Analytics

Padma Lochan Pradhan & , Shrihari Dillip Khatawkar, ADCET, ASHTA, Sangli, Maharashtra, India
Reena Sahu , Shri Rawatpura Sarkar University, Raipur, CG, India

ABSTRACT

This article focuses on fault tolerance computing system for Data Analytics. We address in depth the framework, architecture, activity, action, and response of FTFS (HDFS) to achieve the benchmarking, fault tolerance, scalability, reliability, high availability, and replication designed to deployed java programming on low-cost hardware for heterogeneous data sets that facilitate the collection, storage, processing, generation, and distribution of data from multiple sources for a better and faster analytical process. We are focusing on replication of data nodes in HDFS and implementation of Combinometry-based JAVA code. Our goal is to optimize, visualize, normalize and standardize the data and resources to review the large datasets without much more programming effort and evaluate the performance of the nodes. It is practically innovative and revolutionary to look after the replication task in a continuous process for Data Analytic.

Keywords

Distributed File System(DFS), Hadoop Distributed System (HDFS), Scalable Computing, Data Analytic, Human Computer Interaction (HCI), Fault Tolerance File System (FTFS).

1. INTRODUCTION

The goal of this research is to consider the Hadoop distributed file system as a distributed fault-tolerant file system for better data analysis. Distributed computing is the method of making multiple computers work together to solve a common problem. It makes a computer network appear as a single powerful single computer that provides large-scale resources to deal with complex challenges.

Distributed fault-tolerant replication of data between nodes (between servers or servers/clients) for high availability and offline (disconnected) operation. File systems provide structure to the address space of one or more physical, logical, or virtual devices. Starting with local file systems, other file systems were added over time to address specific needs such as data sharing, remote file access, distributed file access, parallel file access, HPC, archiving, security, etc. [2-5], [18]. Due to the dramatic growth of unstructured data, files as basic units for data containers are transforming into file objects that provide more semantics and rich content processing capabilities. In this talk, the basic principles of currently available file systems (e.g. local FS, shared FS, SAN FS, clustered FS, network FS, distributed FS, parallel FS, etc.) [6-8], [22] are categorized and explained. It also explains technologies such as scale-out NAS, NAS aggregation, NAS virtualization, NAS Clustering, Global Namespace, and Parallel NFS[5-8] (Ref to Table 1). All of these files system categories and technologies are complementary. They are being extended in parallel with additional value-added functionality. New file system architectures are being developed and some of them will be blended in the future.

Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of one or more faults within some of its components. If its operating quality of operation decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively designed system in which even a small failure can cause total breakdown [5-8], [18].

Distributed programming frameworks or distributed computing frameworks that play crucial role in big data processing are covered here. These frameworks provide pre-defined building blocks to developers to have faster application development [18]. Distributed programming(java) frameworks or distributed computing frameworks that play crucial role in big data processing are covered here. These frameworks provide pre- defined building blocks to developers to have faster application development. The distributed systems should have. objects to be accessed using identical operations. accessed without knowledge of their location. concurrently using shared information objects are used simultaneously without interference between them.

In Hadoop, the hdfs dfs -find or hadoop fs -find commands are used to get the size of a single file or the size for all files specified in an expression or in a directry[10-12]. By default, it points to the current directory when the path is not specified (Refer to Literature survey).

The benefits of distributed fault tolerance computing as follows:

- Performance of the Distributed computing can improve performance by allowing each computer in a cluster to process different parts of a task simultaneously.
- Scalability is always needed to allow stakeholders to scale up their tasks.
- Resilience and redundancy is one of the most important factors for continuous operations and services. Cost efficiency and ease of use for data analysis.
- Efficiency of operations and Services.
- Fault Tolerance Distributed Applications (HDFS) services.
- There are four major goals for building distributed systems, resource sharing, computation speedup, reliability, and communication. If a number different sites (with different capabilities) are onnected to each other, then a user at one site can use the resources available at another site.

The portability of Java-based software (which means that you can write your code in a system and run it anywhere) is one of the most important characteristics of Java. Java also provides dynamic capabilities like runtime, such as modifying code modification and garbage collection. This allows you to create modular programs and reusable code. Java is platform independent. One of the most significant advantages of Java is its ability to move easily from one computer system to another.

HDFS is highly fault tolerant. It uses replica process to handle faults. This means client data is repeated many times (default replica factor is 3 on different Data Node in the HDFS cluster. So that in case of any Data Node goes down, the data will be accessed from other Data Nodes [10-12], [18], [22]. Fault Tolerance is the ability of a system to continue to function properly even if some of its components fail (or have one or more faults within them).

Map Reduce is the most popular distributed paradigm thanks to its features such as fault tolerance for processing of large-scale data. The key purpose of creating fault tolerance is to avoid (or at least minimize as much as possible) a situation where the functionality of the system becomes unavailable due to a fault in one or more of its components [11-12], [18-22].

A fault-tolerant API invocation is a call from an API client to an API (Ref. to Figure 4) implementation where a failure of the invocation does not lead to a failure of the client. Because the client itself is typically an API implementation, the API provided by that API implementation does not cause its clients to experience failures.

2. Related Works

The fault tolerance refers to the ability of a system (computer, network, cloud cluster, etc.) to continue operations without interruption when one or more of its components fail. The resilience and redundancy is the one of the most importance factors for the continuous operation and services. The performance of the computing power can support improve speedup and scalability by having each computer in a cluster handle different parts of a task simultaneously. The efficiency, benchmarking, throughput, availability and reliability should be maintaining the balance for high-speed data pulling and pushing for modern data processing, work is called Data Analytics. There are a number of layer is satisfying the need of the public in various area like biomedical science, health care system, Climate analysis, and others business areas. We are going to be discussed the Hadoop technology from high performance computing, storage layer, processing layer and API layer for recent demand and supply through fault tolerance system. There are available the detail as per our day to day requirement.

The use of modern technology has increased vastly and today computing system are interconnected through various communication and sub system (WAN, Hotspot, Wireless LAN). The application of distributed systems in our daily life activities improved with data communication and distributions [10-15], [18]. This is because distributed systems enable nodes to organize and allow their resources to be apply among the connected systems or devices, so that make people to be integrated with geographically distributed computing facilities. The distributed systems may lead to a lack of service availability due to multiple system failures on multiple failure points. This article highlights the different fault tolerance mechanism in distributed systems that are used to prevent, detect, and correct the multiple system failures on multiple points of failure points by considering replication, high redundancy, and high availability of the distributed services [8-11].

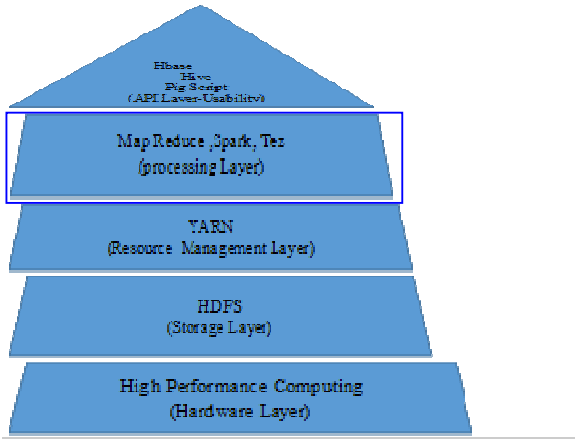

Fig. 1 Framework of HDFS for DA

2.1 Replication

We are going to focused and considering the most important components of HDFS for managing layers and nodes management, that is practically innovative and revolutionary

to look after the replication task in a continuous process for Data Analytic.

- Redundant equipment expresses the first feature of t he breach in three ways[10-12], [18], [22]: Replication: provide several identical instances of t he same system or subsystem, take a job or request all these events simultaneously and as a group
- Redundancy: Provide several instances of the same system and switch to one of the remaining ones in c ase of failure (failover);
- Diversity: Give many different uses with the same meaning and use them as mistakes.

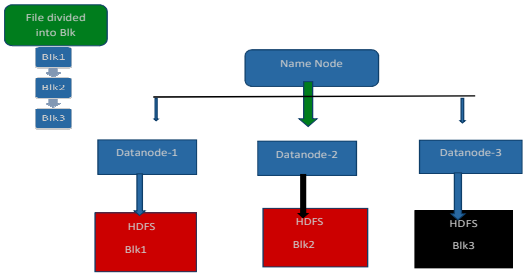

Fig. 2 Storage & Replication of Block in HDFS

3. Literature Survey and Data Collection

This proposed literature survey provides basic data regarding the first step of file identification for data collection, storage, and retrieval process on various operating system for the purpose of application operation and services. The file system operation and services are the two parts of the same coin. The necessity of study and analysis in any Fault Tolerance operating system has increased because of the changes in logic, structure, and the type of technology applied to services that generate quality of data. Finally, the business increases along with technology for business and society, which generate quality of information and share it over the customers. Therefore, we have to focus on performance, security, enhancement, integration, verification and validation the file system. This survey data proposes and resolves the heterogeneous file system by applying a hardening, re-configuration change management, version control, and access control mechanism up to the highest level of the operation and services for specific application requirement.

Research into the kinds of tolerances needed for critical systems involves a large amount of interdisciplinary work. The more complex the system, the more carefully all possible interactions have to be considered and prepared for considering the importance of high-value systems in biomedical science, mobile computing, transport, public utilities and the aerospace, the field of topics that touch on research is very wide: it can include such obvious subjects as software, hardware, modeling, and reliability, or hardware design, to arcane elements such as stochastic models, graph theory, formal or exclusionary logic, parallel processing, remote data transmission, and more.

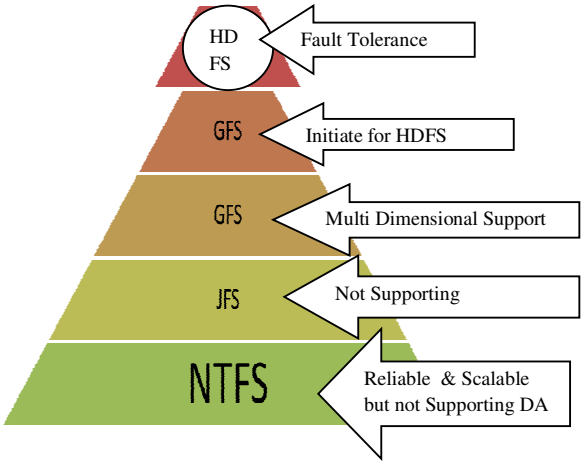

Fig.3 File System improvement for HDFS, Big Data & DA

We conclude that the above FTFS literature survey, how the layering technology utilized in past, present, and what we need for future. HDFS is the backbone of the Data Analytics.

3.1 PROBLEM STATEMENTS:

There is a big difference and un-balanced between traditional system & Big Data application, replication, propagation, and synchronization over the HDFS.
The hardware, software, operating system, and file systems are completely isolated with each other in the traditional system.

There are many more issue was with traditional system like scalabilities, replication, concurrency, *reliability, benchmarking, security, fault tolerance, and high availabilities.

There are many types of failures that can be happed Cause and Effect (AIML+HCI) success. If the processor loses the contents of its main memory during a crash, it will result in a successful but incomplete physical, logical operation, and resulting in inconsistent data (Replication & Load Balancing Issue). This can have an impact on the integrity of the stored data on data nodes/(Replication, Mirroring, and Load Balancing).

Table 1: Year wise Specification

Year	File System	Description	Remarks
1993	NTFS	File recovery and data protection and offers a number of improvements in terms of extendibility, security, and performance	Windows XP, Windows 7, and Windows 10 Reliable, scalable, and highly available. It support Novell Netware,, C, C++, Visual Basic, Dbase etc. The benchmarking, fault tolerance, replication are not available.
1987	CODA Distrib uted FS	Coda from Car negie Mellon University focu	Coda is a distributed file system developed as a research project at

		ses on bandwidth adaptive operation (including disconnected operation) using a client side cache for mobile computing. It is a descendant of AFS 2. It is available for Linux under the GPL.	Carnegie Mellon University since 1987 under the direction of Mahadev Satyanarayanan. It descended directly from an older version of Andrew File System and offers many similar features. The InterMezzo file system was inspired by Coda. The benchmarking, fault tolerance, replication are missing.
1990-99	JFS IBM-AIX	IBM Journal File System. A proprietary distributed file system developed by IBM to provide efficient, reliable access to data. Enhanced data integrity, security, performance, Fault Tolerance, reliability, high availability developed & improved by IBM	IBM-AIX Journaled File System (JFS) is a 64-bits journaling file system created by IBM There are versions for AIX, OS/2, eComStation, Arca OS and Linux operating systems. B+ tree concept became more reliable and applicable Dynamic inode allocation developed C, C++ supported by AIX OS Wireless technology enhanced. Internet, Intranet, Extranet concept was developed in world wide.
2000	GFS	In 2000's, Global file systems have found a use case in providing hybrid cloud storage, GFS2 Red Hat's Global File system Enhanced data integrity, security, performance, Fault Tolerance, reliability, high availability developed & improved. C++ and Java fully emphasized & implemented by global developers.	Global file system is a distributed file system that can be accessed from multiple locations, typically across a wide-area network, and provides concurrent access to a global namespace from all locations. Supported by: ✓ Clustered file system ✓ Distributed file system ✓ Shared memory ✓ Buffer Cache system developed Application interface & GUI utilities integrated with Distributed file system
		Web portal Implemented	
2010	GFS	GFS Google File System (GFS or Google FS, not to be confused with the GFS Linux file system) is a proprietary distributed file system developed by Google to provide efficient, reliable access to data using large clusters of commodity hardware. Google file system was replaced by Colossus in 2010	Google File System (GFS) is a scalable distributed file system (DFS) created by Google Inc. Focus on fault tolerance, high throughput and scalability. GFS is better than DFS Both the file systems are using the master slave architecture. The GFS works on the Linux platform on other hand the HDFS works on the cross platforms. GFS has two servers master node and chunk servers and the HDFS has name node and data node servers. The replication system was initiated.
2004-07	HDFS	Apache License Cross-platform Doug Cutting, Mike Cafarella	Distributed Fault Tolerance system Apache Hadoop is a collection of open-source software utilities that facilitates using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the Map Reduce programming model. Hadoop was originally designed. HDFS is far better than the GFS. The benchmarking, fault tolerance, replication are available[21], [27]., [32]

Table 2: FTFS Problem Analysis

Sn	Scripts commands Action plan	Descriptions	Result
01	iostat	Input /output statistics	CPU & Device Utilization, HA availability, Scalability, Reliability & Integrity of Processor. Primary Data Analysis
02	pmstat	Processors statistics	Global Statistics among all the processors & users:

			Primary Data Analysis
03	vmstat	Virtual memory statistics [MEMORY ACTIVITIES]	Statistics of all the processor runable, block, swap, free buffer, input/output block devices, CPU detail, system, user, idle, waiting stage. HA availability, Reliability, and Integrity of Memory. Primary Data Analysis
04	sar	system activities	Activities report on: paging and swapping of OS detail. Problem Analysis.
05	ps –ef | grep	ACTIVITIES OF PROCESSOR	The suspious processor or orphan/dead one. [space & time complexity issue]
06	lsof l more	FILE SYSTEM ACTIVITIES	list of open files system which is very high risk.
07	/etc/system	KERNEL SYSTEM ACTIVITIES	Can be update the kernel Primary Data Analysis for Performance.
08	/var/adm/message	System mesg. (event mgmt) DC	Date & time stamp.

4 Research Statements:

- Great demand and supply of AIML, Cognitive Science, Quality & Reliabilities Engineering.
- Rapid development and spread up the Data Science, Big Data, Data Analytics, and Visualization.
- Highly demands of health care system for the public and private sectors.
- Tremendous development of Cognitive Technology.
- Great popularities of high performance computing and mobile technologies.
- Health care system need the integration of all of the above points in a systematic way for better public service.
- Effective and efficient protection of patient privacy, data acquisition and retention in the area of Big Data.
- Rapidly demand of social engineering through DA & VA.
- Present Data and Application grown up and integrated. Now high speed data communication is more reliable, available, operative, and serviceable for all the time.
- In the current scenario, the unstructured data is more useful for visualization.

4.1 Research Methodology:

4.1.1 Define

We have to define, design, develop and deployment the large datasets pattern, mechanism, operation and services for big data to fix up dynamic fault tolerance system, integration, replication, scalability and availability over the HDFS for the real-time data analysis for top management. Meanwhile, we can maintain the organizational DSS by applying automated method, model, mechanism (M^3) & tools on FTFS(HDFS) system level to optimize, standardize, normalize and organize the data sets for the maximize the decision management to achieve the highest business objective.

This research paper contributes to the define, design, development of an optimization and demoralization model that aims and objective to determine the optimal cost, time and maximize the Quality of Service to be implemented into the dynamic fault tolerance model and mechanisms deciding on the measure components of HDFS as follows.

We have to consider in this ways to study of research project:
R = Replication, B=Bench Marking, F= Fault Tolerance, S = Scalability, A=Availability, T= Reliability
Let us take a relation R (R, B, F, S, A, T); we have following dependencies for a relation R, and we have checked each for being candidate key.
We have to do some research on functional dependency issue.
R (R, B, F, S, A, T);
The functional dependencies are:
R => B
FS => R
BT => F
How are these candidate keys derives from the above FDs?
Similarly, where R (R, B, F, S, A, T); The functional dependencies are:
R => F
FS => R
SA => T
RB => S

4.2Development Combinations

Sometimes, we want to count all of the possible ways that a single set of objects can be selected - without regard to the order in which they are selected.
In general, n objects can be arranged in $n(n - 1)(n - 2) ... (3)(2)(1)$ ways. This product is represented by the symbol $n!$, which is called **n factorial**. (By convention, 0! = 1.)(Bernard,2007;Mohapatra,2008;Seymour,2010).A **combination** is a selection of all or part of a set of objects, *without* regard to the order in which they were selected. This means that (Replication, Fault Tolerance, and Scalability)

4.3 The number of combination of *n* objects taken *r* at a time is denoted by ${}_nC_r$.

Rule 1. The number of combinations of n objects taken r at a time is
${}_nC_r = n(n - 1)(n - 2) ... (n - r + 1)/r! = n! / r!(n - r)!$
Therefore, this proposed Combination is the right things to evaluate the performance of data & information

4.3.1 Combination Rule: n!/r!(n-r)!

C= 4!/2!(4-2)!=6 Sets of metadata/datasets, Where n no of objects = 4 & r=2, r ways combination.
C= 4!/3!(4-3)!=4 Sets **of metadata, Where n no of objects = 5 & r=3, r ways combination.**

C= 6!/4!(6-4)!=30 Sets **of metadata, Where n number of objects = 5 & r=2ways combination.**

We have to define the various parameter for the beast robust system as follows:

Define:

Let us consider various parameters that: R= Replication; B=Benchmarking; F= Fault Tolerance; S=Scalabilities; A= Availability; T=Reliabilities.

Each and every parameter are the candidate key.

Node about the node is called meta node, block about the block is called meta block, key about the key is meta key.

We are going to be applying the Java Code for assessment as well as evaluation process the HDFS replication management on Data Nodes. We are very much concern with the Replication parameter in this JAVA Programming.

5. Implementation of JAVA PROGRAMMING (Experiment)

```
public class Combinations {
   private StringBuilder output = new StringBuilder();
   private final String inputstring;
   public Combinations( final String str ){
        inputstring = str;
      System.out.println("The input string  is  : " +
inputstring);
   }
         public static void main (String args[])
   {
      Combinations combobj= new
Combinations("RBFSAT");
      System.out.println("");
      System.out.println("");
      System.out.println("All possible combinations are :  ");
      System.out.println("");
      System.out.println("");
            System.out.println("");
      combobj.combine();
   }
      public void combine() { combine( 0 ); }
   private void combine(int start ){
      for( int i = start; i < inputstring.length(); ++i ){
         output.append( inputstring.charAt(i) );
         System.out.println( output );
         if ( i < inputstring.length() )
         combine( i + 1);
         output.setLength( output.length() - 1 );
      }
   }
}
```

D:\>java Combinations
The input string is : RBFSAT
All possible combinations are :

OUTPUT

R
RB
RBF
RBFS
RBFSA
RBFSAT
RBFST
RBFA
RBFAT
RBFT
RBS
RBSA
RBSAT
RBST
RBA
RBAT
RBT
RF
RFS
RFSA
RFSAT
RFST
RFA
RFAT
RFT
RS
RSA
RSAT

Fig. 4 Replication Classification and Pattern

5.1Replication Management on Nodes:

HDFS provides high availability, replication, scalability, reliability, fault tolerance, and benchmarking to store huge amount of data in a distributed environment as data blocks. Each blocks are also replicated to provide fault tolerance. The default replication factor is 3 which is again configurable. So, as you can see in the figure below where each block is replicated three times and stored on different Data Nodes (considering the default replication factor as follows) [15],[19].

5.2 COMBINATIAONAL TECHANIQUE:

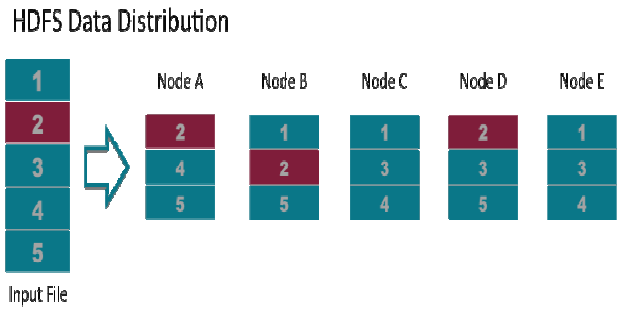

Fig.5 Dynamic Replication Management.(Ref to Fig 2)

Outcome Analysis:

There five types of data set available for Data Analysis are follows The combinations are: Frequent Pattern Data Analysis/ Classification of Pattern/Sub Sets/Group We are going to conclude that the Number of Replications (replication data sets) is Directly Proportional to the Quality of Data Analysis and but poor health of HDFS. We assume that if the replications data sets/ meta data are more the performance will be degrading. There is a great Cause and Effect happening for AIML+HCI. Now the Seeing & Think concept is increasing because of many more system parameters increasing. Therefore, AIML, HCI, and Cognitive science are getting integrated into health of systems as well as a better HDFS backbone for Data Analysis (Ref. Figure 3). This idea helps a lot with reliability and quality engineering

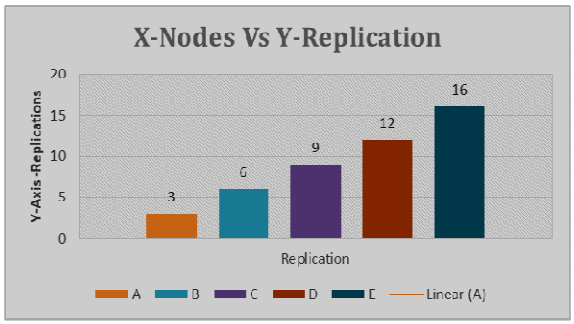

Fig. 6 Node Vs Replication Graph (Ref to Fig 3, 5 & Table 2)

5.3 Results Analysis

The number of replication data sets increase, and the set of metadata and frequent pattern data increases. Therefore, HDFS data nodes (meta candidate key) is directly proportional to the replication. The relation, association, dependency, collaboration, communication, integration, mapping, and synchronizations of subsystems became generated, established, and improve the technological DSS. Now this proposed research paper improves the performance of the HDFS, that can be normalizes, optimizes, and standardized the resources for the betterment of HDFS and Big Data management. The stronger generation of replication (candidate keys) and frequent patterns establish the anti-Fragile information system for top management. Therefore, Combination and Permutation are the right things to assessment and evaluate the relationship & performance of Fault tolerance system. The replication sets are more, the HDFS performance will be degraded.

- Improve the datasets patterns of various problem analysis.
- Improve the assessment and evaluation of FTFS.
- Improve the operation, services, and usability for end-user.
- Improve the composite referential integrity for the HDFS problem analysis.
- More accurate reporting and analysis of current and future forecasting.
- Better as well as good technical decisions.
- Improved operational and services are more efficient services.

Discussion

Apache Hadoop

Apache Hadoop is a Java-
based open source framework that deals with some big data.

- Apply an array/clusters of objects that controls the c ollection of values
  is the solution for data processors.
- Provides fast access to information and improve the actions & response.
- It applies Hadoop Distributed File System (HDFS) which allows heterogeneous
  types of data to be stored.
- It is where data **is** stored and processed without fault (due to Replication).
- Hadoop for cluster management, parallel process an d data storage.

Advantage:

- Big Data storage support massive data.
- Detect, Correct, and quickly respond to hardware failures (due to replication capability).
- Streaming data access.
- Simplified integrity and consistency model.
- High fault tolerance.
- Low cost commercial hardware.

Disadvantage:

- Can not achieve low-latency data access.
- Not suitable for a large number of small file storage.
- File modification is not supported (HDFS 2.x supports appending content to files).
- Parallel writing by users is not supported.
- Replication is degrading the performance.

5.4 Availability (Right Time & Right Data)

This is part of the time data will be available.This feature depe nds on the location and location of the customer.
Example:If the network deployment occurs due to a communi cation error, the connection will be available to customers of some nodes, but not to customers of other nodes. Replication i s the basic process for increasing usability and serviceability.

Robustness (Fault Tolerance) :

This refers to the ability to get rid of contradictions between stored data (data node) and data stored on the medium in which they are stored. High-end storage devices (HDFS)are often used to store reliable data. Unable to obtain reliable information before damaged items are shipped. Robustness is independent of the location of the data or the location of the client.

Recoverability

This means that once the performance of the data is pulled from the client, it can be rolled back
to a stable and consistent state. Use atomic update methods such as transfer methods.
The reliability engineering is covering high availability, fault tolerance, scalability, reliability, and benchmarking for better data management. The capability of replication is increasing the usability.

6. Conclusion

A distributed data center is the foundation for the continued operation of HDFS for large datasets. HDFS provides powerful data storage and support for end-user analytics based on a variety of tools, including today's open source tools. HDFS is data science research that engages the public through careful testing, analysis, analysis, lab work and experimentation (See & Think-AI/ML/HCI). Data scientists look for ways to prevent, analyze, and correct top management's practices. Therefore, HDFS technology is an electronic device designed to identify, process and extract data from non-critical data (big data) that cannot be done by the software that the data uses.
Many companies need big data processing technology to analyze big data in real time. HDFS provides economy, scalability, distributed integrity, expandability, reliability, availability, and redundancy designed for low-cost hardware, software, and communications. Finally, the authors concluded that HDFS is better than GFS. GFS is better than DFS (Ref to Table 1). In this scenario we conclude that the number of data

nodes is increases with the replication, that is called fault tolerance system, then the HDFS and Big Data alive lifelong usability.

The replication frequent pattern is very helpful to the present dependable technology to co-op with the Associations, Aggregation, Composition, Dependency & Collaboration satisfying to the unordered, uncertain environment as well as dynamic and self-autonomy system for Data Analytics services for all the time on every times. This model highly support to top management to improve the high BCP, DRP, Quality of Service, Scalability, Reliability and Availability for all the time.

This dynamic replication classification and frequent pattern approach helps to the any organization the compete more effectively in local, national and international markets at any time and any place around the clock as minimizing costs & maximize DSS, maximize profits, ROI & TCO. Meanwhile, improving functionality, quality & decision (TQM), Maximizing Production, Sales, Market, Share & generate the capital and mean while optimizing time & maximize the utilization of DFS, FTFS and HDFS, and solving multiple problems at the right time with optimal cost, and utilizing overall resources more effectively at the right time and right place.

REFERENCES

[1]. Abouelmehdi, K., Beni-Hssane, A., Khaloufi, H. and Saadi, M, Big Data Emerging Issues: Hadoop Security and Privacy. 2016 5th International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, 29 September-1 October 2016, 731-736. https://doi.org/10.1109/ICMCS.2016.7905621

[2]. Chen, W.H. and Tsai, J.C, Fault-Tolerance Implementation in Typical Distributed Stream Processing Systems, 2014.

[3]. Darcey Kobs, Data Science for Beginners: Comprehensive Guide to Most Important Basics in Data Science, Alex Published 2021.

[4]. Daoudhiri, K., Abouchabaka, J. and Rafalia, N, Attacks and Countermeasures in a Hadoop Cluster. International Journal of Scientific & Engineering Research, 9, 66-70, 2018.

[5]. Ekwonwune, E. and Ezeoha, B, Scalable Distributed File Sharing System: A Robust Strategy for a Reliable Networked Environment in Tertiary Institutions. International Journal of Communications, Network and System Sciences, 12, 49-58, 2019.https://doi.org/10.4236/ijcns.2019.124005.

[6]. Ebnenasir, A, Software Fault-Tolerance. Computer Science and Engineering Department, Michigan State University, East Lansing, 2005. http://www.cse.msu.edu/~cse870/Lectures/SS2005 /ft1.pdf.

[7]. Garg, V.K,Implementing Fault-Tolerant Services Using Fused State Machines. Technical Report ECE-PDS-2010-001, Parallel and Distributed Systems Laboratory, ECE Department, University of Texas, Austin, 2010.

[8]. Gifford, D.K, Weighted Voting for Replicated Data. Proceedings of the Seventh ACM Symposium on Operating Systems Principles, Pacific Grove, 10-12 December 1979, 150-162, 1979. http://dx.doi.org/10.1145/800215.806583

[9]. Howard, J.H., Kazar, M.L., Menees, S.G., Nichols, D.A., Satyanarayanan, M., Sidebotham, R.N., West, M.J, Scale and Performance in a Distributed File System. ACM Transactions on Computer Systems 6(1), February 1988.

[10]. Honnutagi, P.S. 2014. The Hadoop Distributed File System. International Journal of Computer Science and Information Technologies, 5, 6238-6243, 2014.

[11]. Jeremy L. Boerger, Rethinking Information Technology Asset Management Paperback – Import, 5 April 2021.

[12]. Kumar, V. and Chaturvedi, A, Challenges and Security Issues in Implementation of Hadoop Technology in Current Digital Era. International Journal of Scientific & Engineering Research, 8, 984-990, 2017.

[13]. Kumar, M.P. and Pattem, S, Security Issues in Hadoop Associated with Big Data. IOSR Journal of Computer Engineering (IOSR-JCE), 19, 80-85, 2017.

[14]. V.Suganya, The impact of online Brand trust on Sales Promotions and Online Buying Decision", The IUP journal of Marketing Management Vol. XVII No.3 Aug, 2018.

[15]. N. Meenakshi K. E. Rajakumari S. Hariharasitaraman. Data Science and Machine Learning Paperback – Notion Press, India, 2021.

[16]. Rajiv Chopra. 2022. Data Science with Artificial Intelligence, Machine Learning and Deep Learning - Simplified Q & A, Khanna Book Publishing, India.

[17]. Raj Sahil. Management Information System | Second Edition | By Pearson Paperback, 2017.

[18]. Sari, A. .2015. Two-Tier Hierarchical Cluster Based Topology in Wireless Sensor Networks for Contention Based Protocol Suite. International Journal of Communications, Network and System Sciences, 8, 29-42, 2015.http://dx.doi.org/10.4236/ijcns.2015.83004.

[20]. Sudarshan, Database System Concepts, Seventh Edition. McGraw-Hill, India, 2019

[21]. Sales Rodrigues, K. A, Book Review: An Introduction to Nonparametric Statistics. *Journal of Behavioral Data Science*, *2*(1), 124–127, 2022. https://doi.org/10.35566/jbds/v2n1/p8.

[22]. Sajwan, V., Yadav, V. and Haider, M, The Hadoop Distributed File System: Architecture and Internals. International Journal of Combined Research & Development (IJCRD), 4, 541-544, 2015.

[23]. Tian, D., Wu, K. and Li, X, A Novel Adaptive Failure Detector for Distributed Systems. Proceedings of the 2008 International Conference on Networking, Architecture, and Storage, Chongqing, 12-14 June 2008, 215-221. http://dx.doi.org/10.1109/NAS.2008.37.

[24]. Waggoner, P., & Kennedy, R, The Role of Personality in Trust in Public Policy Automation. *Journal of Behavioral Data Science*, *2*(1), 106–123, 2022. https://doi.org/10.35566/jbds/v2n1/p4/