A COMPARATIVE STUDY ON DISTRIBUTED FILE SYSTEMS

CSC802: Distributed Computing

Module 6: Distributed File Systems

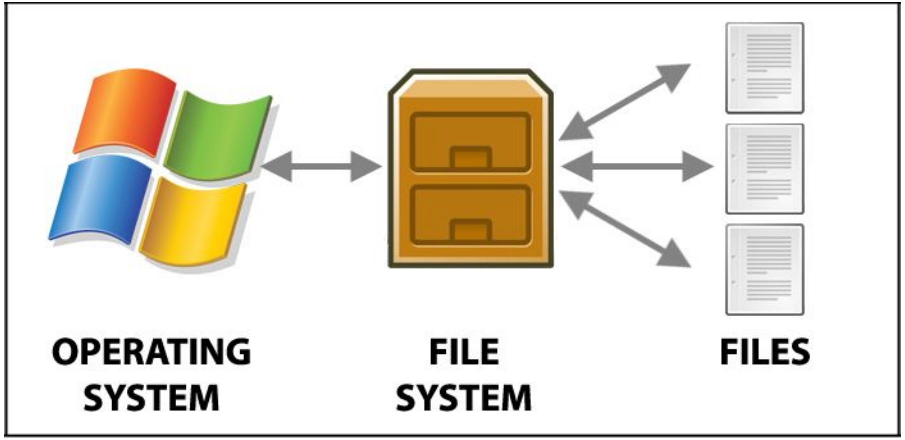
Presentation by:

Amey Thakur

Hasan Rizvi

Mega Satish

AGENDA



1. Abstract
2. Introduction
3. Literature Survey
4. Network File System (NFS)
5. Andrew File System (AFS)
6. Google File System (GFS)
7. Hadoop Distributed File System (HDFS)
8. Comparison
9. Conclusion
10. References

ABSTRACT

- Distributed File Systems are the backbone of how large volumes of data are stored.

- Hadoop File Systems, Google File Systems, and Network File Systems have all shifted the way data is maintained on servers.

- In terms of performance, fault tolerance, consistency, scalability, and availability, each file system has its own set of benefits and drawbacks.

- This presentation examines a file system comparison research and suggests a criterion for selecting a certain file system. The presentation also looks into the pros and drawbacks of using a file system.

INTRODUCTION

- A Distributed File System is a client/server programme that allows clients to access and process data stored on various servers while responding to them as if they were on a local system.

- This sort of file system centralises files from several servers into a single global directory.

- When a client requests the most recent version of the data, the Distributed Files System has a system in place to avoid conflicts and try to share the most recent version of the data.

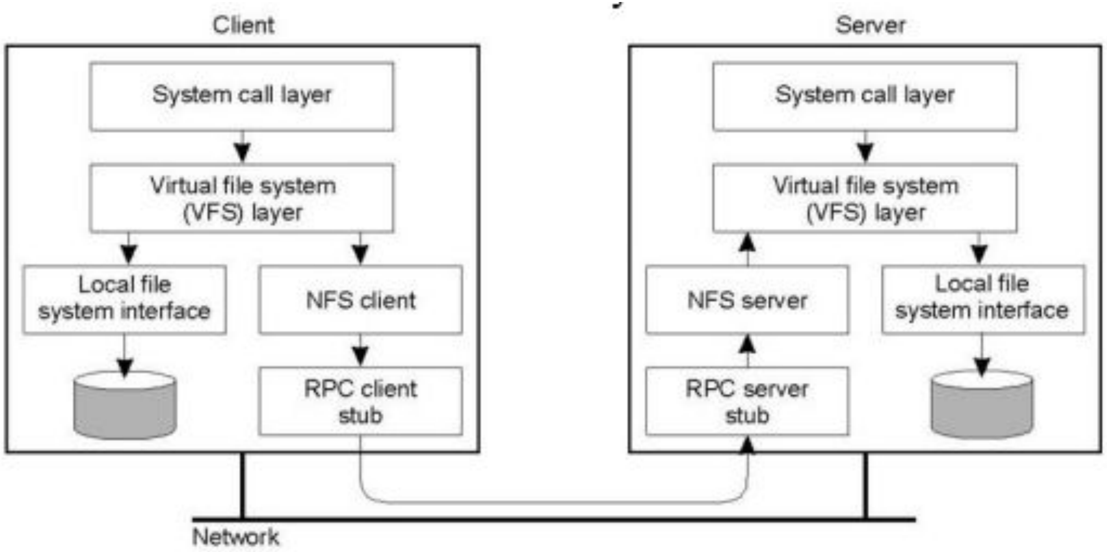
- Transparency, flexibility, dependability, performance, scalability, and security are all variables to consider while creating such systems, as are Architecture, Processes, Communication, Naming, Synchronization, Caching & Replication, and Fault Tolerance techniques.

LITERATURE SURVEY

Year	Paper	Summary
2017	“*An Efficient Cache Management Scheme for Accessing Small Files in Distributed File Systems*" by Kyuongsoo Bok, Hyunkyo Oh, Jongtae Lim and Jaesoo Yoo	They presented a distributed store the executives plot in Hadoop Distributed File Systems that takes into account reserve information for effective gets of little documents (HDFS).
2019	“*An Efficient Ring-Based Metadata Management Policy for Large-Scale Distributed File Systems*" by Yuanning Gao, Xiaochun Yang, Jiaxi Liu and Guihai Chen	They proposed AngleCut, a new hashing algorithm for segmenting metadata namespace trees and serving massive scope communicated capacity frameworks.

NETWORK FILE SYSTEM (NFS)

- Network File System is a mechanism for storing files on network. It is a distributed file system that allows users to access files and directories located on remote computers and treat those files and directories as if they were local. For eg., users can use os commands to create, remove, read, write and set file attributes for remote files and directories.


Client
System call layer
Virtual file system (VFS) layer
Local file system interface
NFS client
RPC client stub
Server
System call layer
Virtual file system (VFS) layer
NFS server
Local file system interface
RPC server stub
Network

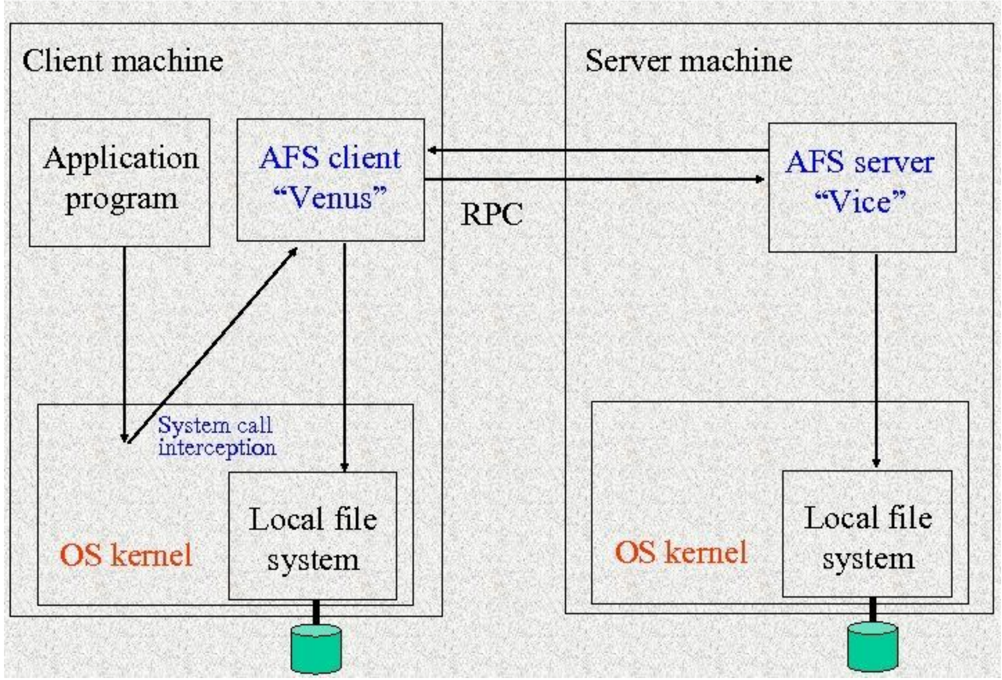
Architecture of NFS

NETWORK FILE SYSTEM (NFS)

Features of NFS:

- Enables multiple computers to use same files, so everyone on the network can access the same data.
- Reduces storage costs by having computers share applications instead of needing local disk space for each user application.
- Provides data consistency and reliability because all users can read the same set of files.
- Makes mounting of file systems transparent to users.
- Makes accessing of remote files transparent to users.
- Supports heterogeneous environment.
- Reduces system administration overhead.

ANDREW FILE SYSTEM (AFS)

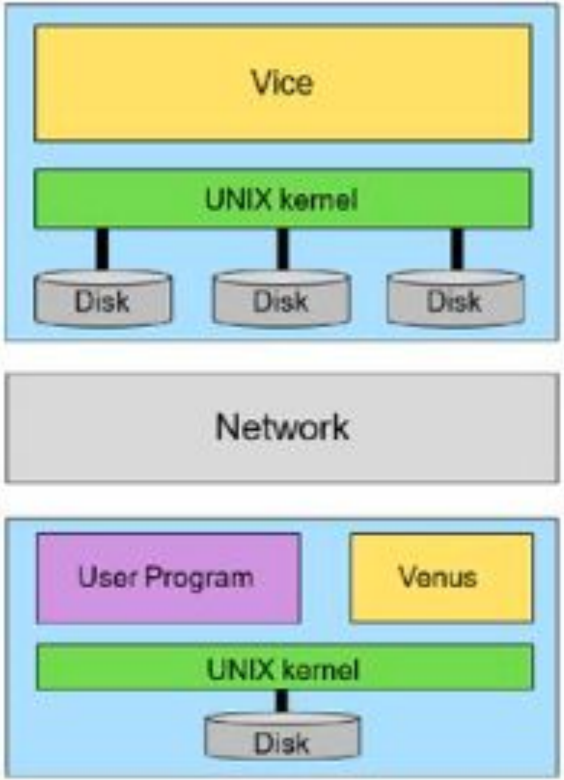


Structure of AFS

- Andrew File System began as part of a bigger Andrew project. It was originally known as "Vice" and was created by Carnegie Mellon University. This was primarily created for computers that run BSD, UNIX, or Mach operating systems.

- Andrew File System work is now being carried out as part of the OpenAFS project. This software is compatible with a variety of platforms, including Linux, Apple Mac OS X, Sun Solaris, and Microsoft Windows NT.

ANDREW FILE SYSTEM (AFS)



Vice (Server)

- Serve files to Venus
- A set of trusted servers, collectively called Vice
- A process running on server side
- Vice process dedicated to each Venus client

Venus (Client)

- Cache files from Vice
- Contacts Vice only when a file is opened or closed
- Reading and writing are performed directly on the cached copy (client side)

ANDREW FILE SYSTEM (AFS)

Features of AFS:

- File Backups:
  AFS data files are backed up nightly. Backups are kept on site for six months.

- File Security:
  AFS data files are protected by the Kerberos authentication system.

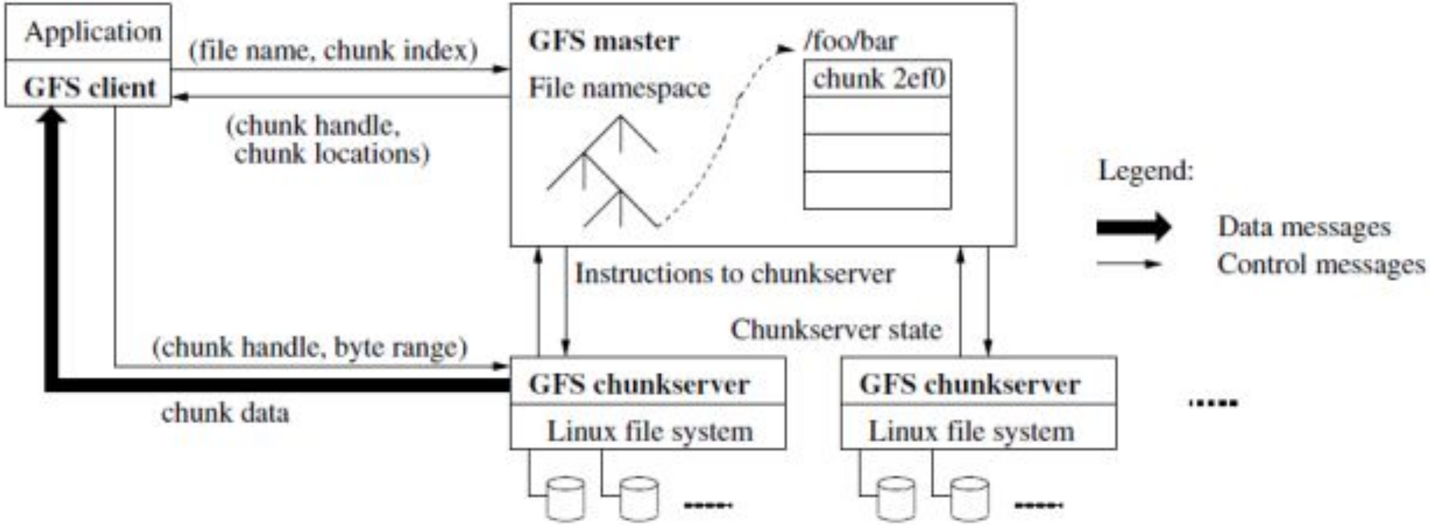
- Physical Security:
  AFS data files are stored on servers located in the UCSC data center.

- Reliability and Availability:
  AFS servers and storage are maintained on redundant hardware.

- Authentication:
  AFS uses Kerberos for authentication. Kerberos accounts are automatically provisioned for all UCSC students, faculty and staff. Kerberos uses the CruzID ‘blue’ password.

GOOGLE FILE SYSTEM (GFS)

- Google File System is made up of groups that contain a large number of storage servers that were built using less expensive tools and technologies and operate on a cluster-based approach. The files are dumped in tree-like structures with path names to differentiate them.



Architecture of GFS

GOOGLE FILE SYSTEM (GFS)

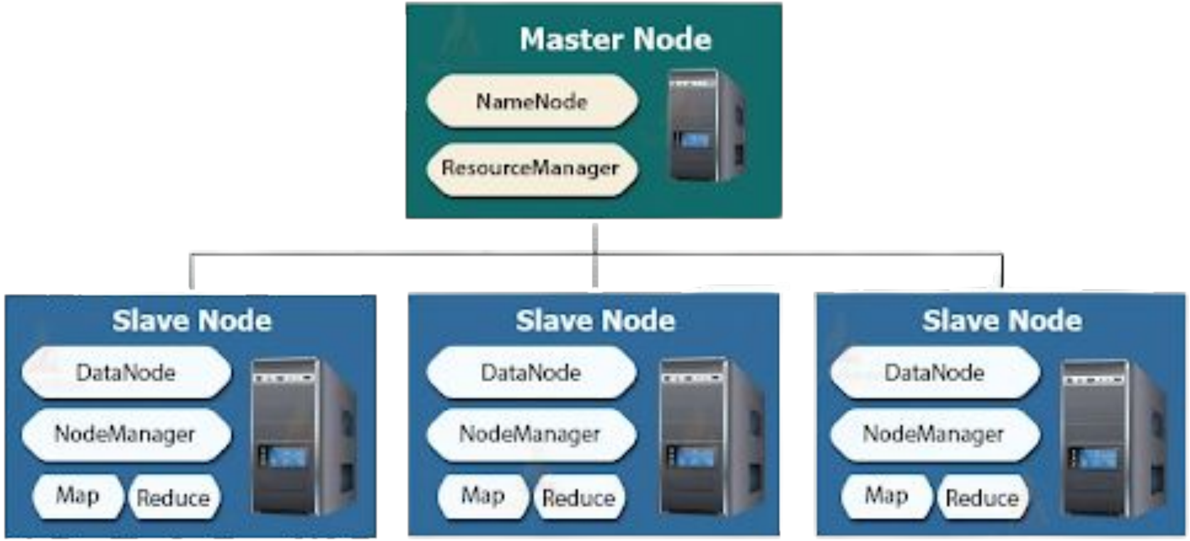
Google File System Characteristics:

- Error tolerance
- Copying mechanism of important data
- Self-reliant data backup and recovery
- Larger productivity
- Less communication of primary and sub-category servers because of block server
- Identification mechanism and authorization scenarios
- Significant presence and lesser downtime

GFS clusters with more than 1,000 nodes and 300 TB of disc storage capacity are the most powerful. This can be accessed by hundreds of clients on a continuous basis.

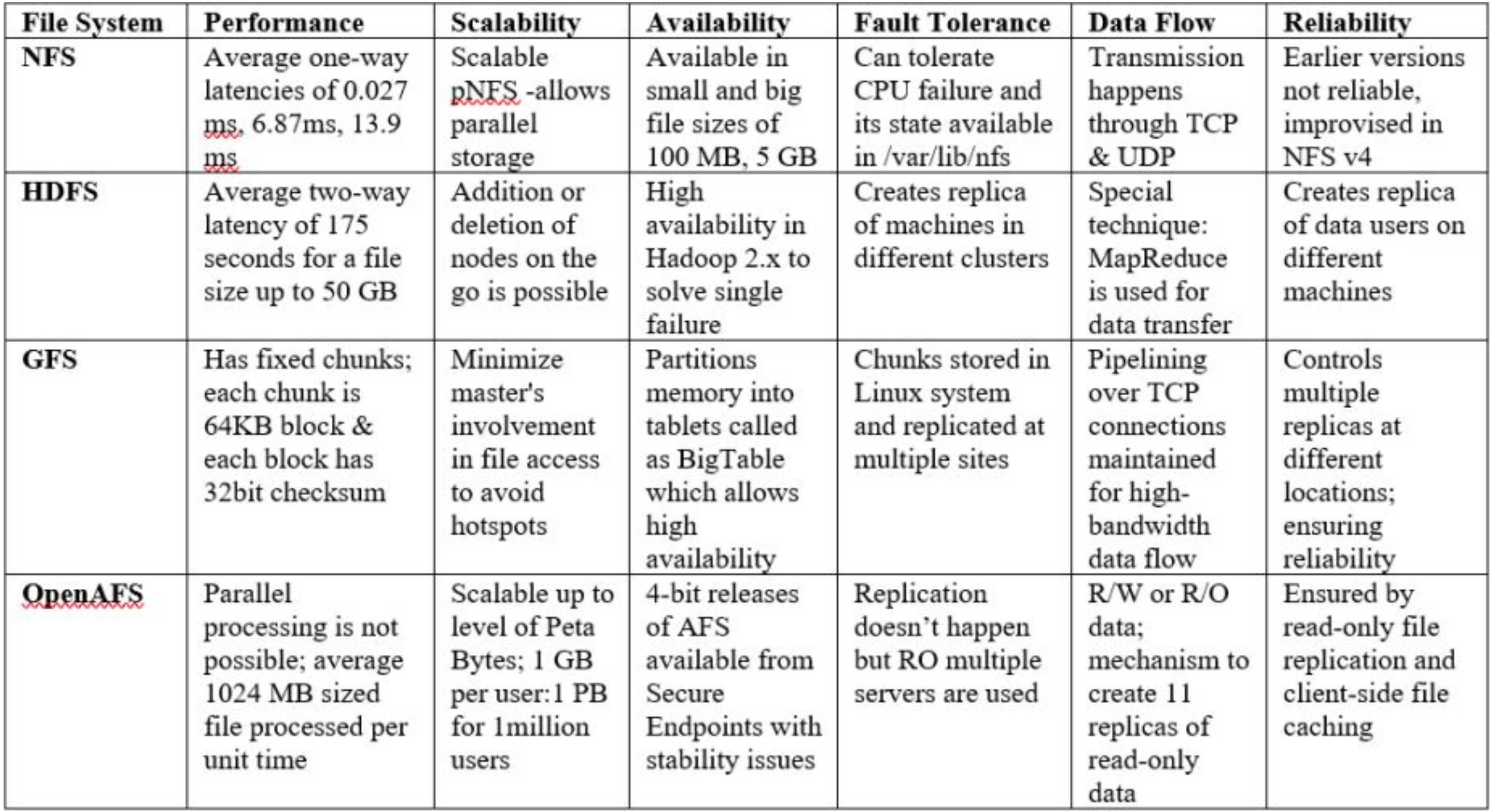
HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

- Hadoop Distributed File System is a free and open-source version of the Google File System. It grants greater information productive rights and is primarily designed for web services that contain large data sets. For eg, Facebook, eBay, LinkedIn, and Twitter are web companies that use Hadoop File System to manage big data volumes and project requirements for data analytics.



Architecture of HDFS

COMPARISON



File System	Performance	Scalability	Availability	Fault Tolerance	Data Flow	Reliability
NFS	Average one-way latencies of 0.027 ms, 6.87ms, 13.9 ms	Scalable pNFS -allows parallel storage	Available in small and big file sizes of 100 MB, 5 GB	Can tolerate CPU failure and its state available in /var/lib/nfs	Transmission happens through TCP & UDP	Earlier versions not reliable, improvised in NFS v4
HDFS	Average two-way latency of 175 seconds for a file size up to 50 GB	Addition or deletion of nodes on the go is possible	High availability in Hadoop 2.x to solve single failure	Creates replica of machines in different clusters	Special technique: MapReduce is used for data transfer	Creates replica of data users on different machines
GFS	Has fixed chunks; each chunk is 64KB block & each block has 32bit checksum	Minimize master's involvement in file access to avoid hotspots	Partitions memory into tablets called as BigTable which allows high availability	Chunks stored in Linux system and replicated at multiple sites	Pipelining over TCP connections maintained for high-bandwidth data flow	Controls multiple replicas at different locations; ensuring reliability
OpenAFS	Parallel processing is not possible; average 1024 MB sized file processed per unit time	Scalable up to level of Peta Bytes; 1 GB per user:1 PB for 1million users	4-bit releases of AFS available from Secure Endpoints with stability issues	Replication doesn't happen but RO multiple servers are used	R/W or R/O data; mechanism to create 11 replicas of read-only data	Ensured by read-only file replication and client-side file caching

WHAT WE LEARNT FROM THIS PRESENTATION?

- We studied different types of file systems.

GFS	NFS	AFS
CLuster based architecture	Client Server based architecture	CLuster based architecture
No caching	Client and server caching	Client caching
Not similar to Unix	Similar to Unix	Similar to Unix
File data is stored across different chunk servers thus reads come from different chunk servers	Reads come from same server	Reads come from same server
Server Replication	No replication	Server Replication
Location independent namespace	Not location independent namespace	Location independent namespace
Lease based locking	Lease based locking	Lease based locking

CONCLUSION

- There are many different file systems, and this presentation compares a few of them: NFS, AFS, GFS, and HDFS.

- With great performance, availability, and a powerful file replication method against fault tolerance, HDFS is the most preferred option.

- In terms of scalability and the use of chunks of data for pipelining transmission via TCP channels, GFS is the second preferred option.

- NFS is a bit more popular since it is an older file system that consumers believe to be more reliable, however OpenAFS also has several user-friendly features like scalability.

REFERENCES

[1] De, Suman, and Megha Panjwani. "A Comparative Study on Distributed File Systems." In Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough, pp. 43-51. Springer, Cham, 2021.

[2] Y. Gao, X. Gao, X. Yang, J. Liu and G. Chen, "An Efficient Ring-Based Metadata Management Policy for Large-Scale Distributed File Systems," in IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 9, pp. 1962-1974, 1 Sept. 2019, doi: 10.1109/TPDS.2019.2901883

[3] Kyoungsoo Bok, Jongtae Lim, Hyunkyo Oh and Jaesoo Yoo, "An efficient cache management scheme for accessing small files in Distributed File Systems," 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, 2017, pp. 151-155, doi: 10.1109/BIGCOMP.2017.7881731

[4] M. Nithya and N. U. Maheshwari, "Load rebalancing for Hadoop Distributed File System using distributed hash table," 2017 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, 2017, pp. 939-943, doi: 10.1109/ISS1.2017.8389317

[5] L.Sudha Rani, K.Sudhakar, S.Vinay Kumar, “Distributed File Systems: A Survey”, International Journal of Computer Science and Information Technologies, Vol. 5(3), 2014, 3716-3721

THANK YOU