*International Journal of Interdisciplinary Innovative Research & Development (IJIIRD)*

**ISSN: 2456-236X**

Vol. 03 Special Issue | 2019

# Private Distributed File System

Mr. Mihir Kulkarni[1], Mr. Pushkar Chinchanikar[2], Prof. Smita Pai[3], Mr. Shubham G Patil[4], Dr.Varsha Bodade[5]

[1,2,3,4,5] *Department of Information Technology, Terna Engineering College, Navi Mumbai, Mumbai University, Maharashtra, India*

**ABSTRACT**

***The Distributed file system (DFS) allows users of physically distributed computers to communicate or share data with each other by using a common file system. A typical configuration for a DFS is a collection of workstations and mainframes connected by a local area network (LAN). A DFS is installed in operating system meaning a part of OS and hence, no separate software is required. The paper claims that the principle of distributed file system is fundamental for a fault tolerant and scalable DFS design. The paper makes a point that DFS is suitable when using centralized storage architecture as this design is easy to use and has wide range of applications. Our distributed file system design is based on centralized storage architecture, which has properties like fault tolerance, concurrent access, multi-tenant environment etc.***

***Keyword— Distributed file system, FUSE, Linux OS, Network file system.***

## 1. INTRODUCTION

Before 21st century, people that wanted to share files used a method known as sneaker net. A sneaker net is basically the process of sharing files by copying them onto floppy disks, physically carrying it to another computer and copying it again. As computers evolved it became evident that the old file systems had many limitations that made them unsuitable for multiuser environments. Initially many users started to use FTP to share files. Although this method generally avoided the time consuming physical movement of removable media, files were still required to be copied twice: once from the source computer onto a server, and another second time from the server onto the destination computer. Additionally, users were required to know the physical addresses of every computer involved in the file sharing process for it carry out the operation.

As computer companies tried to solve the shortcomings above, entirely new systems such as Sun Microsystem's Network File System (NFS) were developed and new features such as file locking were added to existing file systems. The new systems which came forward such as NFS were not replacements for the old file systems, but an additional layer between the disk file system and the user processes. A file system is that piece of software that is in charge of storing, organizing and generally taking care of data represented as files and directories. A distributed file system (DFS) is a file system with data stored on a server. The data is accessed and processed by the user as if it was stored on the local client machine. The purpose of a distributed file system (DFS) is to allow different users having physically distributed computers to share data and store resources by using a common file system. A typical configuration for a DFS is a collection of workstations and mainframes connected by a local area network (LAN). With Distributed File System (DFS), system administrators can essentially make it easy for users to access, manage and process files that are physically distributed across a network. With DFS there are following advantages:

1. Remote information sharing
2. User mobility
3. Availability

Distributed file systems (DFS) resemble the API of local file systems. Even though the file system interface is general and fits a broad spectrum of applications, most distributed file system implementations are optimized, developed for a particular class of applications.

## 2. LITERATURE SURVEY

Distributed file systems provide persistent storage of data which is stored at remote location. Files are explicitly created and they can survive the lifetime of processes and nodes until deleted explicitly. Distributed file systems (DFS) resemble the API of local file systems. To applications, it should be transparent whether whatever data of applications is stored on a local file system or on a distributed file system.

This type of data model and the interface to applications distinguishes distributed file systems from other types of distributed storage such as databases etc. Now days, virtually all cloud service providers store their data in distributed file systems. They store data in a global federation of various cluster file systems rather than in a single, globally distributed file system.

## 3. PREREQUISITES

Following are the requirements that should be met for efficient working of the system:

**3.1 Software requirements:**

1) Linux
2) GCC or Clang
3) FUSE 2.6 or later
4) FUSE development files

**3.2 Hardware requirements:**

1) 700 MB free space
2) 2 GB RAM
3) Additional to the disk space required for the base installation, a running system will require more space for caching, persistent message stores, and other functions.

## 4. PROPOSED METHODOLOGY

For developing distributed file system there are three architecture:

1) Centralized Storage architecture
2) Peer to Peer architecture
3) Hybrid architecture

For our project we choose first architecture. Centralized architecture is widely used because it is easy to build, simple to maintain and secure.

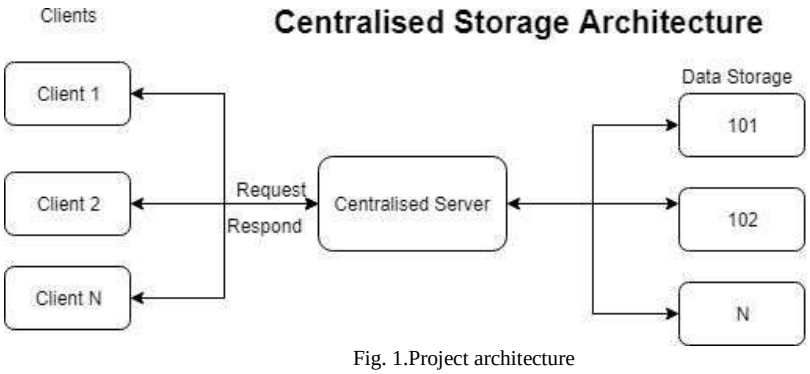


Fig. 1.Project architecture

There is one central server which can be connected to single or multiple data storage nodes. This server is called by clients which are basically users accessing the files. Every client has FUSE installed in it which enables users to create and use distributed environment. FUSE converts user requests and enables us to execute respective actions on it.

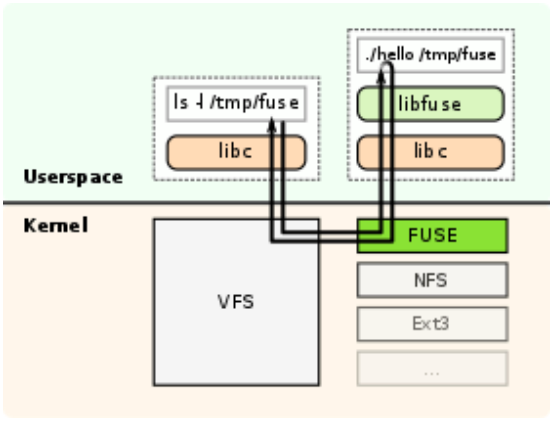


Fig.2. Architecture of FUSE

Every command that is entered by user is managed by FUSE. FUSE acts as a middleware. It is responsibility of the file system developer to execute appropriate functions and handle messages accordingly.

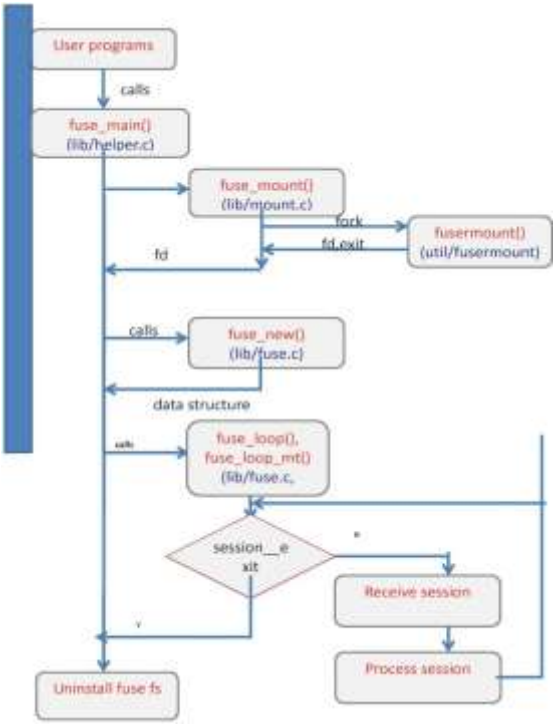


Fig. 3. Flow of operations in Fuse

For communication between client and server we have used socket concept. Server consist of socket server that listens to client requests and responds accordingly.

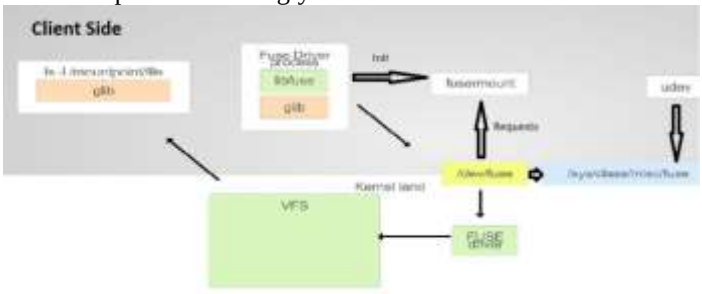


Fig. 4. Workflow of our project

## 5. RESULTS

Fig. 5, shows the operations client can perform on file system. User is unable to distinguish between local file system and distributed file system, that's what makes DFS very useful.

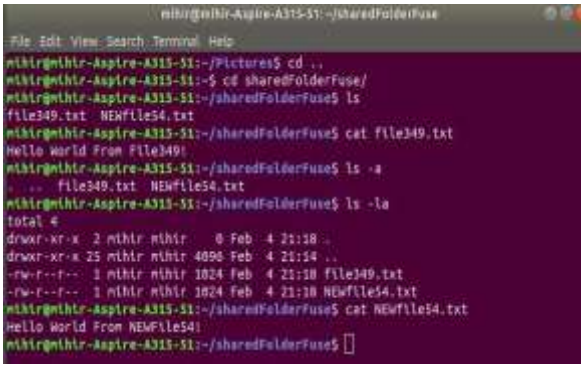


Fig. 5. Client side operation

Fig. 6, shows the log of operations client performed on the distributed file system. It can also help in checking history of operations performed by user.

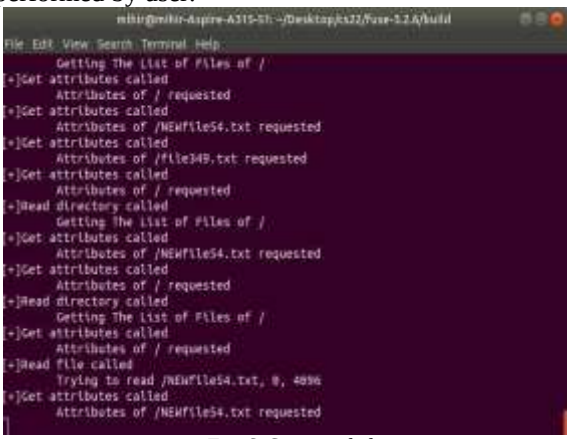


Fig. 6. Server side log

## 6. CONCLUSION AND FUTURE SCOPE

With the use of distributed file system we can achieve several of our objectives such as providing a distributed shared network for distributed file system. This distributed file system can be used in several large and small scales such as small laboratory or large company. Sharing, processing and optimization of resources can be achieved with the use of distributed file system.

By using our design, file sharing between servers to server can be implemented resulting in very easy sharing our data and convenience of remote sharing.

## 7. REFERENCES

1. http://en.wikipedia.org/wiki/Clustered_file_system
2. http://www.comp.leeds.ac.uk/mscproj/report s/1011/mista.pdf
3. ftp://ftp.irisa.fr/local/caps/DEPOTS/BIBLIO20 08/biblio_Lage-Freitas_Andre.pdf
4. www.cs.rice.edu/~gw4314/lectures/dfs.ppt
5. http://en.wikipedia.org/wiki/Andrew_File_System
6. http://en.wikipedia.org/wiki/Google_File_System
7. http://searchciomidmarket.techtarget.com/definition/distribu ted-file-system
8. http://technet.microsoft.com/enus/library/cc738688( v=ws.10).aspx
9. http://www.webopedia.com/TERM/D/distrib uted_file_system.html
10. www.cs.gsu.edu/~cscyqz/courses/aos/slides11/ch6. 5-Fall11.pptx - slide ch6.5