Terna Engineering College
Computer Engineering Department
Program: Sem VIII

Course: Distributed Computing Lab (CSL802)

Faculty: Rohini Patil

Experiment No. 9

A.1 Aim: To Discuss different types of a file system (NFS, AFS, Google Case Study).

PART B
(PART B: TO BE COMPLETED BY STUDENTS)

Roll No. 50	**Name:** AMEY MAHENDRA THAKUR
Class: BE COMPS B 50	**Batch:** B3
Date of Experiment: 18-03-2022	**Date of Submission:** 23-03-2022
Grade:	

B.1 Input and Output:

- **Network File System:**

- The Network File System is abbreviated as NFS. It's a distributed file system protocol. This protocol was designed by Sun Microsystems in the year of 1984.
- It is a client/server architecture that includes a client programme, a server programme, and a protocol that facilitates communication between the client and the server.
- It's the protocol that lets users access data and files via a network from anywhere. Because the NFS protocol is an open standard, any user may readily implement it.
- Any user can alter files in the same way that they could if they were using other protocols. This protocol is based on the ONC RPC system as well.
- The Network File System is an IP-based system that operates on all networks. It's used in a client/server application where the NFS server is in charge of authorization, authentication, and clients.
- This protocol is supported by Apple Mac OS X, Unix, and Unix-like operating systems including Solaris, Linux, FreeBSD, and AIX.

Advantages of a Network File System:

- An advantage of NFS is that it uses the existing IP infrastructure; NFS is a low-cost network file sharing solution that is easy to set up.
- A significant benefit of NFS is that it facilitates central management, eliminating the need for individual user systems for added software and disk space.
- NFS is user-friendly, enabling users to access files on remote hosts in the same way as local files are accessed. As fewer CDs, DVDs, Blu-Ray disks, diskettes, and USB drives are in circulation, this decreases the need for portable media storage devices and increases security.

Disadvantages of a Network File System:

- **Security:** The fact that NFS is built on RPCs, which are fundamentally unsafe and should only be utilised on a trusted network behind a firewall, is the first and greatest security risk. NFS will be exposed to online attacks if this does not happen.
- **Protocol chattiness:** In order to transport data, the NFS client-server protocol necessitates a lot of request activity. To read and write data, the NFS protocol takes a lot of little interactions or steps, which adds up to a lot of overhead for someone working with today's AI/ML/DL workloads, which consume a lot of small files.
- **File sharing is complicated:** At best, configuring and setting up effective shared file access via file locking and caching is a difficult undertaking. On the one hand, it adds a significant amount of protocol overhead, resulting in the above-mentioned chattiness. However, it still leaves a lot to be desired, in that any host's mount command for the same file system might easily go wrong.
- **Parallel file access:** NFS was designed to allow users to access a shared network file in a sequential manner, but today's applications deal with bigger files, necessitating non-sequential or parallel file access. Although this was added to NFSv4, few clients currently support it.
- **Limitations on block size:** The present NFS protocol standard permits only 1MB of data to be delivered per read or write request. 1MB was a large amount of data in 1984, however that is no longer the case. There are certain programmes that should be transmitting GBs of data rather than MBs.

Versions:

- **NFS version 1** was the initial two NFS protocol versions, followed by **NFS v2** or RFC-1094 (1989) and **NFS v3** or RFC-18133, which were released by the IETF after 1989. (1995).
- Both **NFS v2** and **NFS v3** are widely used in the IT sector, and in the UNIX environment, they have become the standard file-sharing protocol. NFS v4 was introduced in 2000 via RFC 3010.
- **NFS v4** is the current version of NFS, which was implemented using RFC 3530. It was released in 2003.
- Since then, **NFS v4** has vastly improved in terms of optional features such as security, caching, locking, and message communication speed as compared to early **NFS v4**.

NFS Architecture:

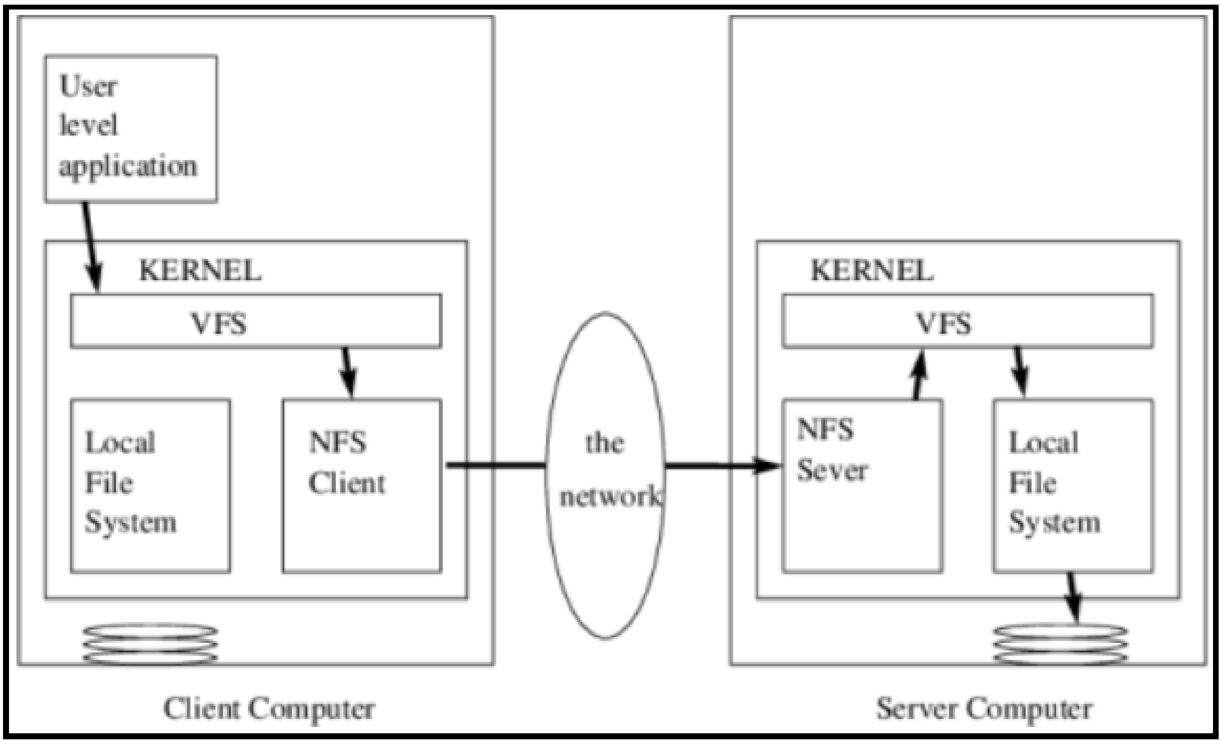

Figure: NFS Architecture

- **Andrew File System (AFS)**

- Businesses utilize an AFS to make it easier for AFS client workstations in different locations to access server files stored on the server. Through a network of trusted servers, it provides a homogenous, location-independent file namespace to all client workstations.
- Users share data and programmes after logging into workstations that connect inside the Distributed Computing Infrastructure (DCI).

- The idea is to reduce client-server communication in order to support large-scale information exchange. This is done by moving whole files between server and client computers and caching them until the servers receive a newer version.
- In scattered networks, an AFS employs a local cache to boost speed and reduce effort. When a server responds to a request from a workstation, it stores data in the workstation's local cache.

Andrew File System Architecture:

- **Vice:** The Andrew File System uses a collection of dependable servers known as Vice to provide a homogenous, location-transparent file namespace to all client workstations. On both clients and servers, the Berkeley Software Distribution of the Unix operating system is used. The operating system on each workstation intercepts file system calls and sends them to a user-level process on that workstation.
- **Venus**: Venus is a technique that caches files from Vice and sends updated copies of those files back to the servers where they came from. Venus communicates with Vice only when a file is opened or closed; individual bytes of a file are read and written directly on the cached copy, bypassing Venus.

The necessity for scalability was a major driving force for this file system architecture. Venus, rather than Vice, conducts as much labour as possible to increase the number of clients a server can serve. Vice only preserves the features that are required for the integrity, availability, and security of the file system. The servers are configured as an informal confederacy with limited connection.

The server and client components used in AFS networks are as follows:

- A client is any computer that makes requests for AFS server files that are shared over a network.
- Once a server responds and sends a requested file, the file is saved in the client machine's local cache and shown to the user.
- When a user views the AFS, the client uses a callback mechanism to send all changes to the server. The local cache on the client machine caches frequently used files for quick access.

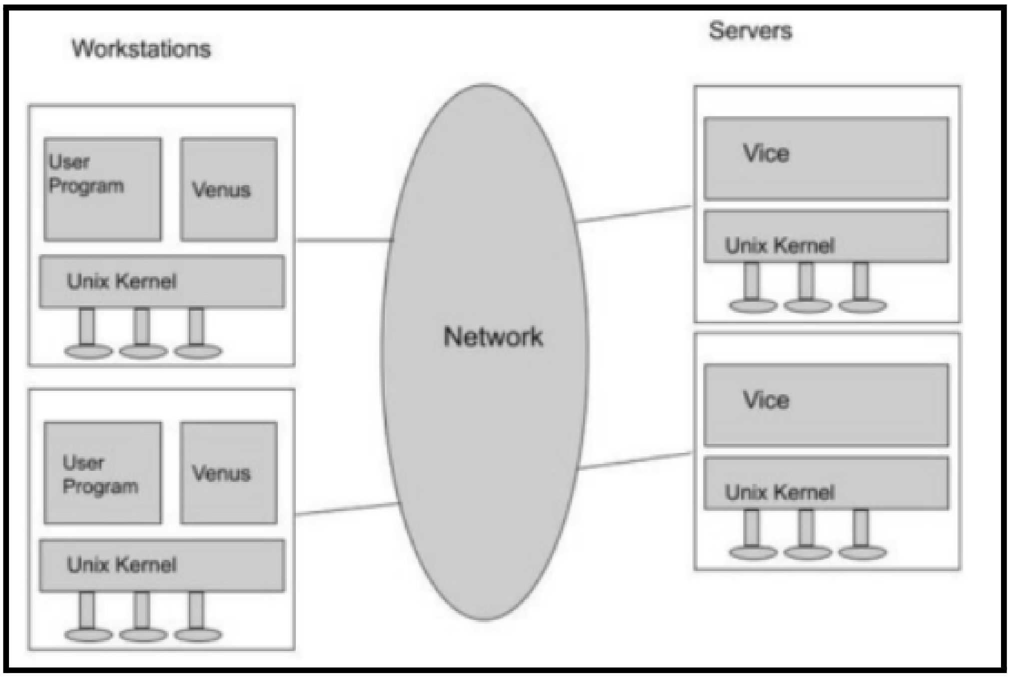

Figure: AFS Architecture

AFS Implementation:

- Standard system calls are used by client processes to connect with the UNIX kernel.
- The kernel has been modified considerably to detect references to Vice files in relevant activities and route requests to the Venus client process on the workstation.
- If a volume is missing from this cache, Venus contacts any server with which it already has a connection and requests the location information, which it then puts into the mapping cache. Unless the server already has a connection, Venus establishes a new one. This connection is then used to get the file or directory.
- The establishment of a connection is required for authentication and security. When the target file is found and cached, a copy is produced on the local disc.
- After that, Venus returns to the kernel, which opens the cached copy and passes the handle to the client process. The UNIX file system is used as a low-level storage system by both AFS servers and clients. The client cache is a local directory on the workstation's hard drive. Files with placeholder names for cache entries can be found in this directory. Venus and server processes both use the latter's modes to directly access UNIX files, bypassing the time-consuming path-name-to-inode translation.
- When a file is removed from the cache, Venus notifies the appropriate server that the file's callback has been removed.

- **Google File System(GFS)**

- The Google File System (GFS) is a scalable distributed file system (DFS) designed by Google Inc. to meet the company's growing data processing needs.
- Large networks and connected nodes benefit from GFS' fault tolerance, dependability, scalability, availability, and performance.
- GFS is made up of a number of storage systems made up of low-cost commodity hardware.
- It's designed to meet Google's various data storage and usage requirements, such as its search engine, which creates massive volumes of data that must be saved.
- The Google File System took advantage of off-the-shelf servers' strengths while limiting hardware flaws.
- GFS was created to meet Google's huge cluster requirements while minimizing the impact on applications.
- Path names are used to identify files in hierarchical folders.
- The master, which communicates with and monitors the status updates of each chunk server through timed heartbeat messages, controls metadata such as namespace, access control data, and mapping information.

Features of GFS:

- Tolerance to faults
- Replication of critical data
- Data recovery that is both automatic and efficient
- Aggregate throughput is high.
- Because of the huge chunk server size, there is less contact between the client and the master.
- Management and locking of namespaces
- Availability is high.

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.

B.2 Conclusion:
Hence, we were capable of discussing various sorts of file systems and design methodologies.

B.5 Question of Curiosity:

1] Chunk servers are implemented using

A. Hadoop distributed file system

B. Google file system

C. None of these

Ans: (B) Google file system

2] The information mapping data blocks with their corresponding files is stored in

A. Data node

B. Job Tracker

C. Task Tracker

D. Namenode

Ans: (D) Namenode

3] To keep the GFS (Google File System)highly available there are two strategies used namely _____________ and _____________.

A. Fast Recovery , Garbage Collection

B. Fast Recovery , Chunk Replication

C. Master Replication , Data Integrity

D. None of these

Ans: (B) Fast Recovery , Chunk Replication

4] GFS (Google File System) supports following operations:

A. read, write , make, delete

B. update, add , read, write

C. snapshot, remove, del

D. read, write , open , close

Ans: (D) read, write , open , close

5] In distributed file system is mapping between logical and physical address

A. Client Interfacing

B. Naming

C. Migration

D. Consistency

Ans: (B) Naming