

INTRODUCTION TO DISTRIBUTED FILE SYSTEMS.

- 1 - A DFS is a file system that is distributed on various file servers and locations.
- 2 - It allows programs and users to interact with files on remote servers in the same way they would with local files.
- 3 - Thus, applications can read, write and perform various operations on remote files as if they were working on files in their local machine.
- 4 - It provides a unified storage space that spans multiple physical devices.
- 5 - DFS's primary goal is to enable users of physically distributed systems to share resources and information through the common file system (CFS).

Two DFS have been widespread

① Sun Network File system (NFS)

② Andrew File System (AFS)

Filesystems and their characteristics

- ① - A file system is a process of managing how and where data on a storage disk.
- ② - It provides a structure and mechanism necessary for efficient file management
- ③ - They are responsible for
 - organization
 - storage
 - retrieval
 - naming
 - sharing
 - protection of files
- ④ - Files are stored on disk and other non-volatile storage media
- ⑤ - File contains both data and attributes
Data consist of sequence of data items
Attributes are held as a single record containing information such as length of file, time stamp and file type, etc.

File system characteristics

1 - Hierarchy and structure

- logical organisation and easy navigation

2 - File naming conventions

3 - File attributes

size, creation date, last modification,

permissions

4 - Access control.

Updated
by
System

File length
creation timestamp
Read timestamp
Write timestamp
Attribute timestamp
Reference count

Updated
by
owner

Owner
File type
Access control list

Eg: FAT, GFS, HFS, NTFS.

Requirements of DFS

① Transparency

Ⓐ Structure transparency

- clients should not be aware of the no. ~~of~~ locations of file servers or storage devices
- file server multiplicity is hidden

Ⓑ Access transparency

- users should be able to access remote files in the same way as they access local files.

Ⓒ Naming transparency

file's name should not reveal their physical location

Ⓓ Replication Transparency

Clients remain unaware of multiple file copies or their location.

② User Mobility: users can access their file from any node without being tied to a specific one

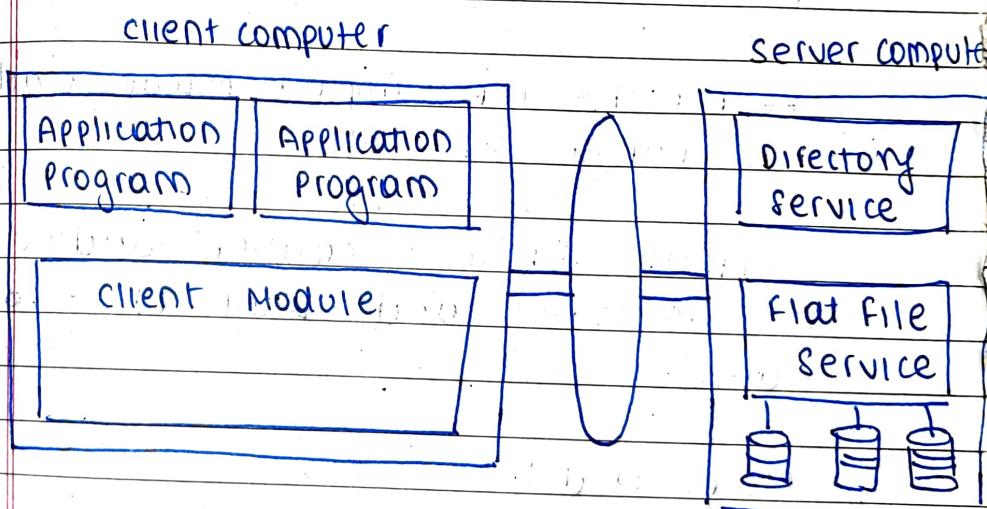
③ Performance: The system should deliver high responsiveness to handle client's request.

- ④ Scalability: the DFS must support growth in number of nodes and users without degrading performance
- ⑤ High Availability: The system should remain operational even during partial failures such as network or server issues.
- ⑥ Concurrent file updates: file system should handle simultaneous access by multiple users to same file while maintaining consistency and preventing conflicts.
- ⑦ Security: the system must safeguard data privacy and enforce secure access controls.

FILE SERVICE ARCHITECTURE

- (1) File service architecture in DS manages and provides access to files across multiple servers or location
- (2) An architecture that offers a clear separation of main concerns in providing access to files is obtained by structuring the file service in 3 components
 - Flat file service
 - Directory service
 - Client Module

(3) Diagram:



- (4) The flat file service and directory service each export an interface for use by client to provide a comprehensive set of operations for access to files.
- (5) The client module provides a single interface with operations on files similar to those in conventional file systems.

⑥ - The responsibilities and roles of each module are as follows -

(A) flat file service

① concerned with the implementation of operations on the contents of file.

(unique file identifiers).

② - It uses UFIDs to identify the file, which is a globally unique sequence ~~for~~ given to files

③ - It also generates new UFIDs when creating files.

(B) directory service.

① - Responsible for mapping human readable file names to UFIDs

② - supports directory creation, adding names to directories and obtain UFIDs from names.

(C) client module

① - runs on client computers, integrating flat file and directory service as a single API to application programs.

② - It provides a unified application interface for file operations, emulating UNIX file system system

- ③- cached recently used file blocks to improve performance.

D) Flat File service interface

- ①- offers an RPC interface for client modules with operations for reading and writing and managing files.
- ②- it includes access permission checks with exception handling for invalid UFIDs.

E) Access control

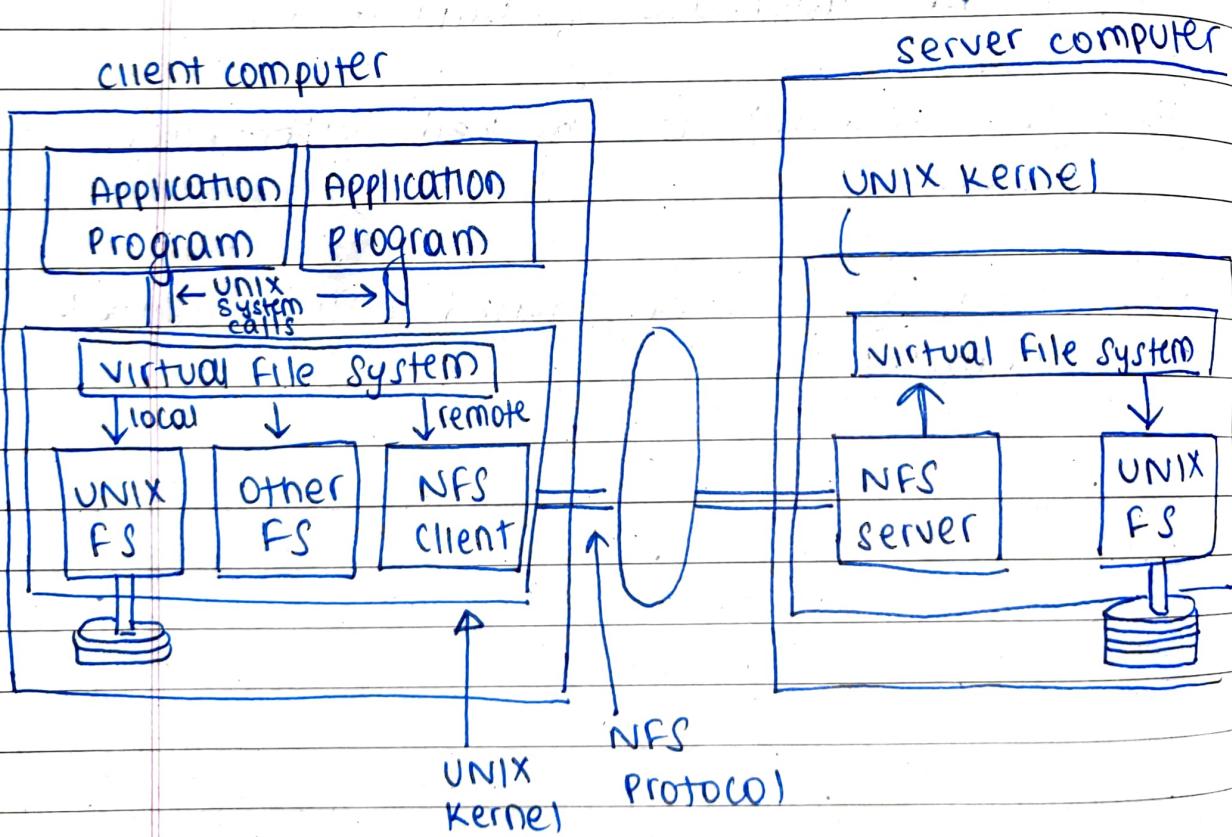
- ①- Access checks are performed on server as RPC interface is unprotected.
- ②- user identity is checked with each request.
- ③- security is strengthened using digital signatures



SUN Network File System

- ① - sun NFS is a distributed file system protocol - a set of remote procedure calls, that provide the means for clients to perform operations on remote file machines.
- ② - Developed by sun Microsystems
- ③ - NFS protocol is OS independent, but originally developed for UNIX systems.

④ diagram :



- ① NFS server and client module reside within the kernel
- ② It is a client server application that provides a shared file storage for clients across a network.
- ③ NFS is stateless; so server does not maintain "between call" information.

virtual file system (VFS)

- ① VFS is an abstraction layer that allows different file systems to be present in a unified way to applications.
- ② It abstracts the details of filesystems
- ③ It allows NFS to integrate with other file systems in UNIX environment, supporting multiple file systems concurrently.

NFS server interface

- ① NFS server interface defines a set of protocols, procedures and mechanism that allow all clients to communicate with the NFS server and access files and resources on the server.
- ② Because NFS servers are stateless, it must commit any NFS request to stable storage before returning results.

- ③- NFS server interface also includes mechanisms for authentication and security
- ④- It ensures that clients provide proper credentials to access resources on server and enforces ~~action~~ access control based on credentials.

NFS client interface

- ①- NFS client facilitates interaction b/w client machine and remote files located on all NFS servers
- ②- Since it is integrated with the kernel, it interacts with standard
- ③- It is responsible for translating higher level file system requests made by applications on the client machine to NFS protocol operation.
- ④- It manages the communication b/w the local FS and remote NFS server.
- ⑤- To improve performance, the NFS client caches data from server. This reduces the need for frequent server interactions and improves speed of file access for applications.