CASE STUDY
ON
SUN NETWORK FILE SYSTEM

Presented By:-

Rashmi Khadka

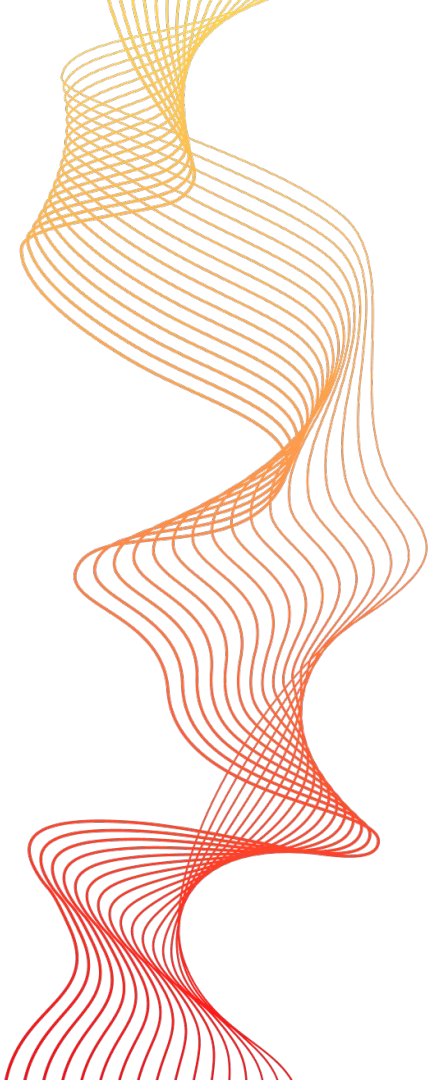
Sameer Shrestha

Shiwani Shah

Sujan Lamichhane

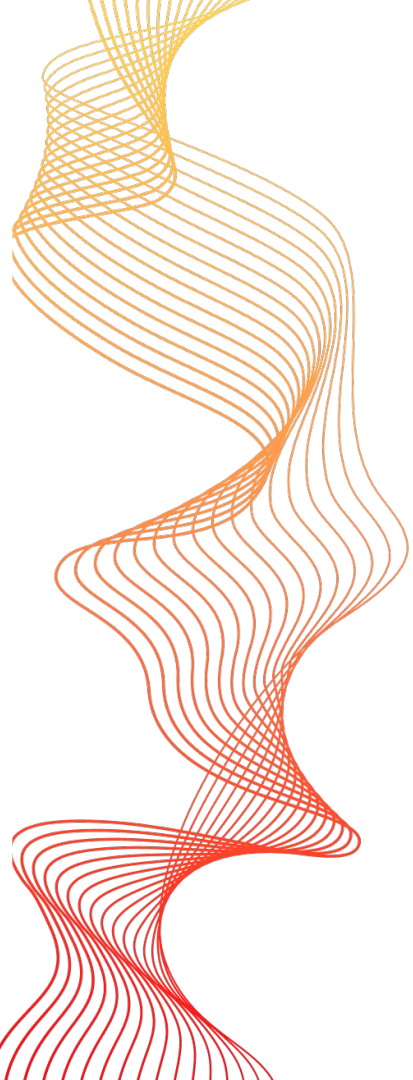
Distributed System
Er. Anku Jaiswal

TABLE OF CONTENTS



INTRODUCTION

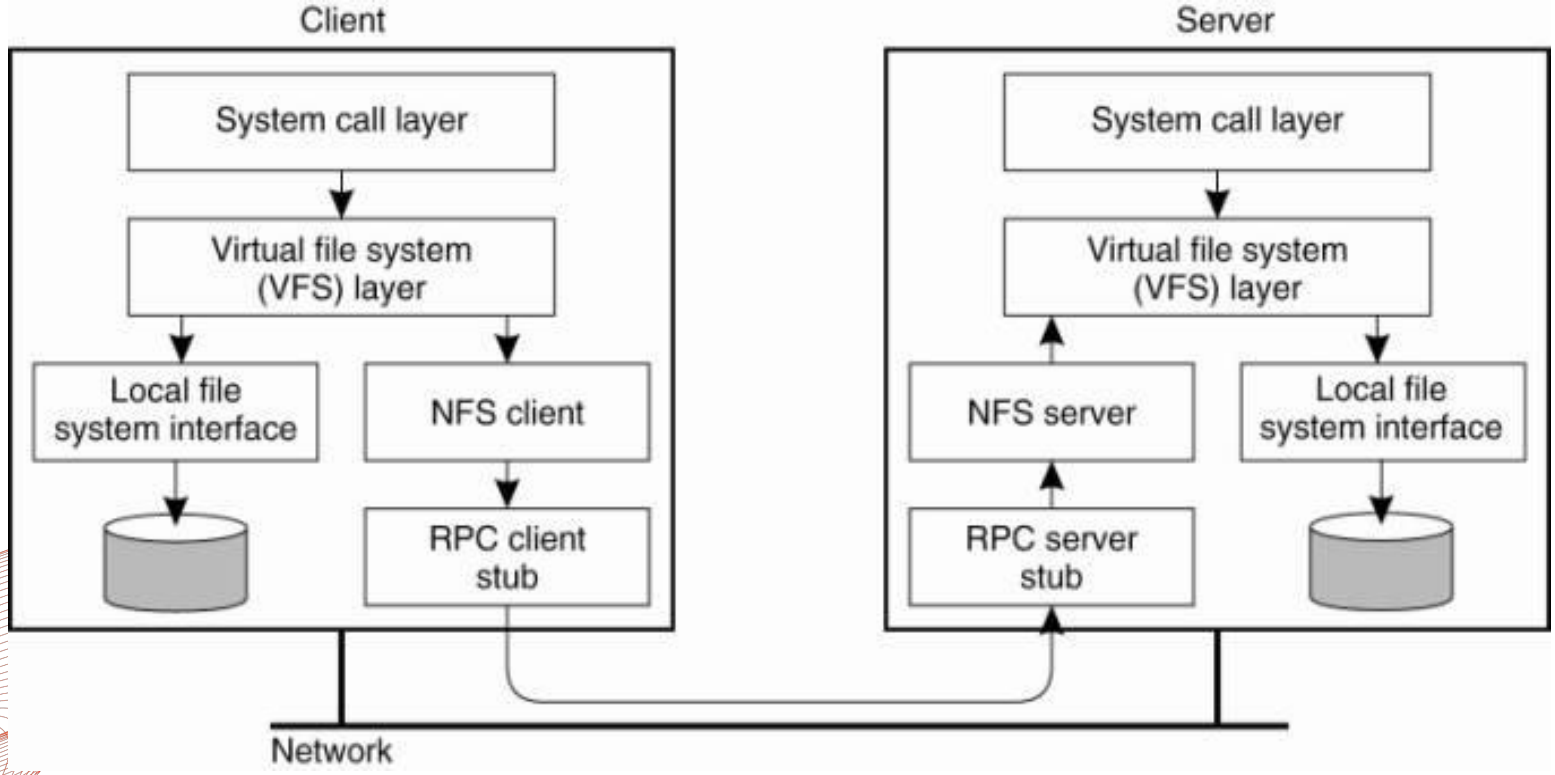
- A distributed file system that allows users to access files and directories on remote machines as if they were local.
- Developed by Sun Microsystems in 1984
- NFS is a stateless protocol i.e, the server does not keep track of any state of clients.
- Easier to implement and recover from crashes.
- Uses a client-server architecture
- Supports a variety of file operations (open, read, write, close, & readdir).
- Widely used as it is supported by most Unix and Linux distributions



ARCHITECTURE

- Built upon a distributed architecture, consisting of multiple components that work together to enable efficient file storage and retrieval
- The architecture consists of components client, server and protocol
  - Client: The computer that is accessing the remote files.
  - Server: The computer that is hosting the remote files.
  - Protocol: The set of rules that govern how clients and servers communicate with each other.

ARCHITECTURE DIAGRAM



Virtual File System (VFS)

- Acts as an intermediary layer between the NFS client code and the local file systems
- Maintains a table with an entry for each open file.
- Keeps track of file that is currently available both locally and remotely
- Contains v-nodes (virtual i-nodes) indicating whether the file is local or remote.

NFS Client/Server Structure

- Client programs have file descriptors, current directory
- Inside kernel, these refer to vnodes of type NFS.
- When client programs make system calls:
  - NFS vnode implementation sends RPC to server.
  - Kernel half of that program waits for reply.
  - So we can have one outstanding RPC per program.

NFS Client/Server Structure

- Server kernel has NFS threads, waiting for incoming RPCs:
  - NFS thread acts a lot like user program making system call.
  - Find *vnode* in server corresponding to client's vnode.
  - Call that vnode's relevant method.
  - Server vnodes are typically of type FFS.
  - This saves a lot of code in the server.
  - NFS will work with different local file systems.
  - Files are available on server in the ordinary way.
  - NFS server thread blocks when needed.

MOUNTING

- The client mounts the shared directories from the server to make them accessible within its own file system.
- It establishes logical connection between the client and the server's file system hierarchy.
- The shared directories appear as part of the client's local file system, enabling transparent file access.

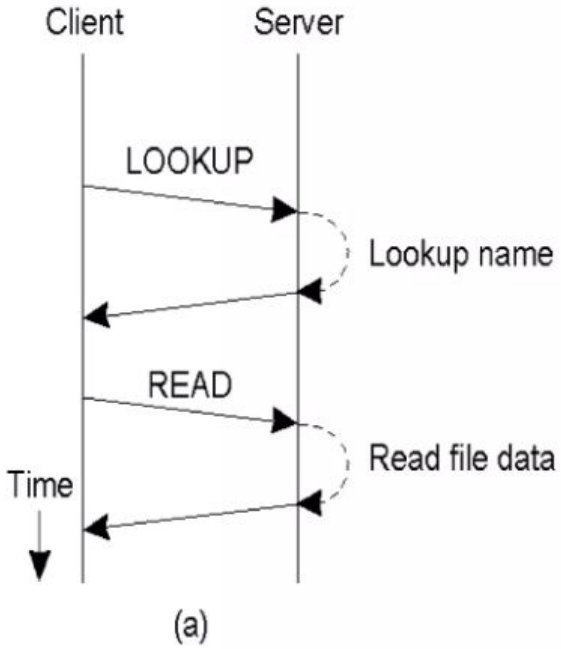
NFS Protocol

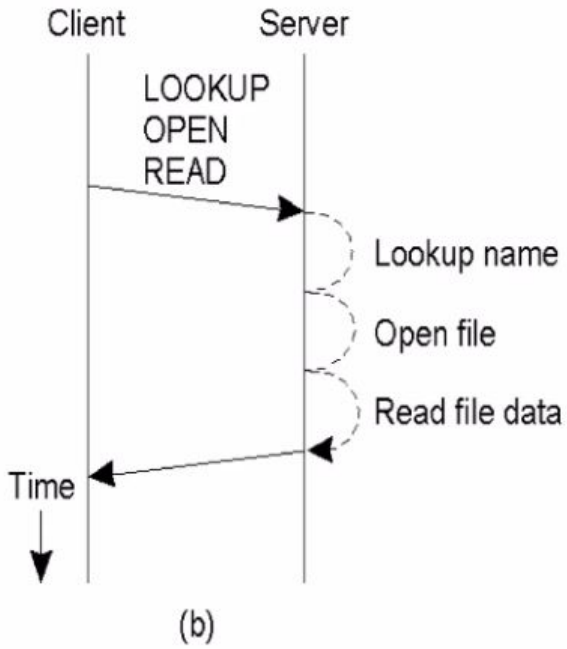
- Defines the standards and rules for communication between clients and servers.
- Specifies the message formats, procedures, and semantics for various file operations such as reading, writing, and file attribute retrieval.

NFS PROTOCOL

RPC request	Action
CREATE	Creates(or truncates) a file directory
LINK	Creates a hard link
LOOKUP	Looks up file in a directory
MKDIR	Makes a directory
READADDR	Reads the content of a directory
REMOVE	Removes a file in the directory
RENAME	Renames a file in the directory
RMDIR	Removes a directory
SYMLINK	Creates a symbolic link

NFS Communication





Client-Side Caching

- SNFS implements client-side caching to improve performance and reduce network overhead.
- Client-side caching involves storing frequently accessed files and their attributes locally on the client system.

Server-Side Caching

- SNFS implements server-side caching to optimize performance and reduce server load.
- Server-side caching involves storing frequently accessed files and their attributes on the NFS server.

Caching for Performance Improvement

01

Read-Ahead:

- Transfers between client and server occur in large chunks, typically 8 Kbytes.
- Even if fewer bytes are requested, a chunk of 8 Kbytes is transferred.

02

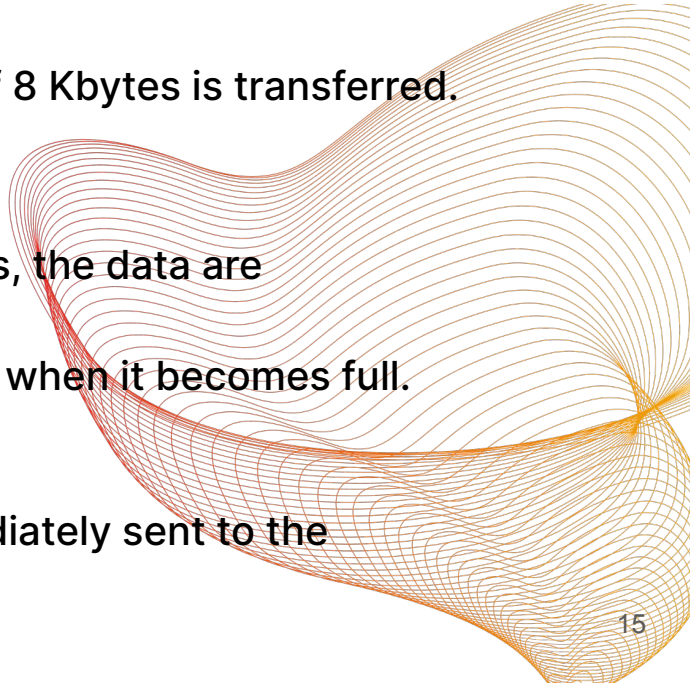
Write Accumulation:

- If a write system call writes less than 8 Kbytes, the data are accumulated locally.
- The entire 8K chunk is sent to the server only when it becomes full.

03

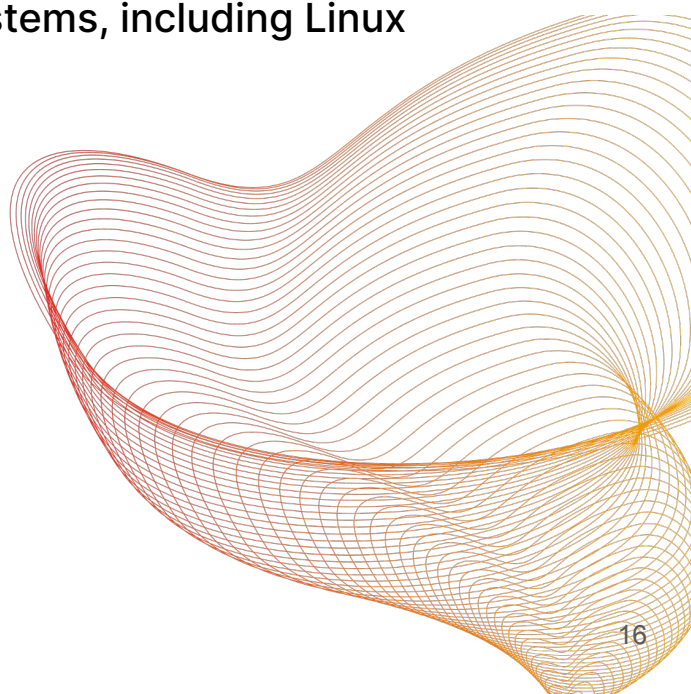
Immediate Data Sending:

- When a file is closed, all of its data are immediately sent to the server.



SNFS Deployment

- SNFS can be deployed on a variety of operating systems, including Linux and Windows
- It can be run on commodity hardware
- SNFS is easy to install and configure

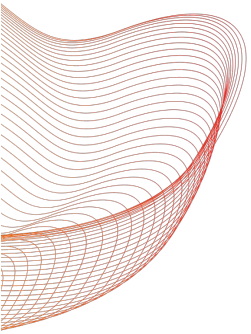


SNFS Security

SNFS has built-in security features, including authentication and access control

It supports secure file transfer protocols like SSL and SSH

SNFS can be integrated with existing security infrastructures



Application

- **Shared Home Directories:**

  NFS simplifies sharing user home directories across multiple machines in a network.
- **Diskless Workstations:**

  NFS enables diskless workstations to boot and run applications by accessing the OS and files over the network.
- **Centralized Data Storage:**

  Organizations use NFS to centralize data storage for easy management and backup.
- **Virtualization:**

  NFS is used to store virtual machine images and share them among different virtualization hosts.

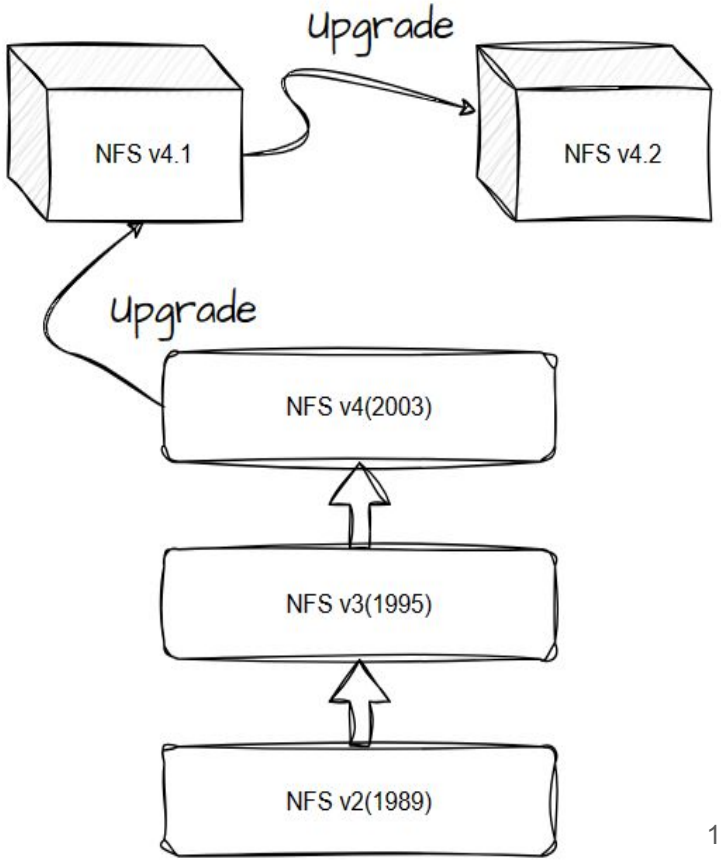
Different Versions of NFS

01 NFS Version 2(NFS v2)

02 NFS Version 3(NFS v3)

03 NFS Version 4(NFS v4)

- NFS v4.1
- NFS v4.2



NFS v2

- Developed in 1989.
- Uses UDP protocols.
- UDP runs over an IP network to ensure a hassle-free connection between the server and the client.
- UDP clients can continue sending request to another server even when server is not functional.
- If the UDC loses a data frame, an entire RPC request must be resent.
- Only allows the first 2 GB of a file to be read due to 32-bit limitations.

NFS v3

- Developed in 1995.
- Uses both TCP and UDP protocols
- Facilitates clients to access more than 2 GB of file data as this version can handle 64-bit files and offsets.
- Strong authentication, improved & advanced file caching, and efficient internalization capacity.
- Supports asynchronous writes and is more stable and reliable than NFS version 2, particularly in error handling.

NFS v4

- Developed in 2003.
- Uses only TCP protocols.
- If a data frame is lost by TPC, only the lost frame has to be resent.
- Includes performance improvements, mandates strong security, and introduces a stateful protocol
- NFS version 4 works well on the internet and can work through powerful firewalls
- Supports massive file sizes and offsets.
- Updated Versions of NFS v4
  - NFS v4.1
  - NFS v4.2

Advantages of NFS

- Shared Data Access
- Shared Applications
- Transparent File Mounting
- Heterogeneous Work Environment
- Centralized Data Management
- Enhanced Security

Disadvantages of NFS

- Uses RPC authentication which can be easily spoofed
- Filesystem data is transmitted in clear text hence data can be copied
- Network is slower than local disk
- Complexity ,Security issues

THANK YOU!