

Network File System (NFS)

Introduction

Clients

Clients represent the systems or users requesting access to files within the network file system. They serve as the primary interface to interact with the NFS, initiating various file-related operations such as reading, writing, deleting, streaming, and more.

Within the NFS ecosystem, clients enjoy a suite of essential file operations, enabling seamless interaction with the network file system.

Naming Server

The Naming Server (NM) stands as a pivotal component in the NFS architecture. It serves as a central hub, orchestrating communication between clients and the storage servers. Its primary function is to provide clients with crucial information about the specific storage server where a requested file or folder resides. Essentially, it acts as a directory service, ensuring efficient and accurate file access.

Storage Servers

Storage Servers (SS) form the foundation of the NFS. These servers are responsible for the physical storage and retrieval of files and folders. They manage data persistence and distribution across the network, ensuring that files are stored securely and efficiently.

Core File Operations

1. Writing a File/Folder

Clients can create and update files and folders within the NFS. This includes both synchronous and asynchronous writes, with asynchronous writing optimizing client response time for large files.

2. Reading a File

Clients can retrieve file contents stored within the NFS.

3. Deleting a File/Folder

Clients can remove files and folders, freeing up storage space.

4. Creating a File/Folder

Clients can generate new files and folders, allocating storage and initializing metadata.

5. Listing Files/Folders

Clients can list all files and subfolders within a directory.

6. Getting File Information

Clients can obtain metadata like size, access rights, and timestamps.

7. Streaming Audio Files

Clients can stream audio files directly from the NFS.

Specifications

1. Naming and Storage Servers

1.1 Initialization

- **Initialize the Naming Server (NM):**

The NM manages the directory structure and stores information about file locations. Its IP and port are publicly known to clients and storage servers during registration.

IP addresses must not be hardcoded (e.g., 127.0.0.1).

- **Initialize Storage Server 1 (SS₁):**

Each Storage Server sends the following to the NM upon initialization:

- IP Address
- Port for NM Connection
- Port for Client Connection
- List of Accessible Paths

- **Dynamic Registration:**

New storage servers can join dynamically and register with the NM at any time.

- **Commands from NM to SS:**

- Create empty file/directory
- Delete file/directory
- Copy files/directories from another SS

- **Client Interactions (via NM):**

- Reading and writing files
- Listing accessible paths
- Retrieving metadata
- Streaming audio

- **Initialize Additional Storage Servers:**

Repeat the process for SS₂, SS₃, etc.

- **Client Requests:**

Once all SSs are registered, the NM starts accepting client requests.

1.2 Storage Server (SS) Responsibilities

- Handle commands from NM:

- Create or delete files/directories
- Copy files/directories

- Handle client operations:

- Read and write files
- Return file information (size, permissions, etc.)
- Stream audio files
- Store and manage accessible paths

- Support dynamic addition of new storage servers and synchronization of accessible path data with NM.
-

1.3 Naming Server (NM) Responsibilities

- Maintain directory structure and accessible path listings across all registered SSs.
 - Manage IP and port mappings for each file path.
 - Handle client requests and forward them to the relevant SS.
 - Track task feedback and acknowledgments from SSs and relay to clients.
 - Maintain logs of requests, connections, and system status.
-

2. Clients

2.1 Reading, Writing, Retrieving Information, and Streaming

- Clients send requests such as READ <path> or STREAM <path> to the NM.
- NM returns the IP and port of the relevant SS.
- The client establishes a direct connection to the SS.
- The SS sends data or acknowledgments to the client.
- The connection terminates upon a STOP packet.

2.2 Creating and Deleting Files/Folders

- Clients send CREATE or DELETE requests to the NM.
- NM determines the responsible SS and forwards the request.
- SS performs the operation and sends acknowledgment back through NM to the client.
- NM updates accessible path records dynamically.

2.3 Copying Files/Directories

- Clients send COPY <source> <dest> requests to NM.
- NM validates paths and coordinates the copy between source and destination SSs.
- NM updates paths and notifies the client upon completion.

2.4 Listing Accessible Paths

- Clients can request a list of all accessible paths from NM.
 - NM aggregates data across all SSs and returns the combined list.
-

3. Other Features

3.1 Asynchronous and Synchronous Writing

- Large write operations can be handled asynchronously to improve response time.
 - SS acknowledges write acceptance immediately and flushes data to disk in chunks.
 - Upon completion, SS notifies NM, which updates the client.
 - Synchronous writes can be forced with a flag (e.g., --SYNC).
 - If an SS fails during async write, NM informs the client.
-

3.2 Multiple Clients

- NM supports multiple concurrent client connections.
 - Each request receives an initial ACK to prevent blocking.
 - Multiple clients can read a file simultaneously.
 - Only one client can write to a file at a time.
 - If a write is in progress, other clients cannot read that file.
-

3.3 Error Codes

Define a set of error codes to communicate issues:

- File not found
 - File locked for writing
 - Invalid path
 - Server unavailable
 - Permission denied
-

3.4 Search Optimization

- Use efficient data structures (e.g., Tries, Hashmaps) for path lookups in NM.
 - Implement **LRU caching** for recently accessed paths to improve search speed.
-

3.5 Backing Up Data

- Replicate each file/folder in two additional SSs (once there are more than two SSs).
 - NM retrieves data from backup SSs in case of failure.
 - Writes are asynchronously duplicated across backups.
 - When an SS recovers, its data is resynchronized before reactivation.
-

3.6 Redundancy (Bonus)

- Maintain redundant paths and failover logic for high availability.
 - NM seamlessly redirects requests if a server is down.
-

3.7 Bookkeeping and Logging

- NM logs all communication between NM, SS, and Clients.
 - Logs include timestamps, IP addresses, ports, and operation status.
 - Display relevant operation messages and acknowledgments in real-time.
-

Technical Notes

- Use **TCP sockets** for communication.
- Any **POSIX C library** may be used.
- Design should be modular and decomposed logically.
- Test components independently using tools like **Wireshark** or **netcat**.

- Ensure extensibility for adding more storage servers dynamically.
-