Chapter 48: The Andrew File System (AFS)

1. Introduction to AFS

- The **Andrew File System (AFS)** was designed at Carnegie Mellon University with the primary goal of **scalability**.
- It aimed to support a large number of clients with a minimal number of servers.

2. AFS Design Principles

- **Whole-File Caching:** When a file is opened, the entire file is downloaded to the client's local disk. All subsequent reads and writes are performed on this local copy, which is very fast.
- **Callbacks:** AFS uses a callback mechanism to ensure cache consistency. When a client caches a file, the server promises to notify the client if the file is modified by another user.
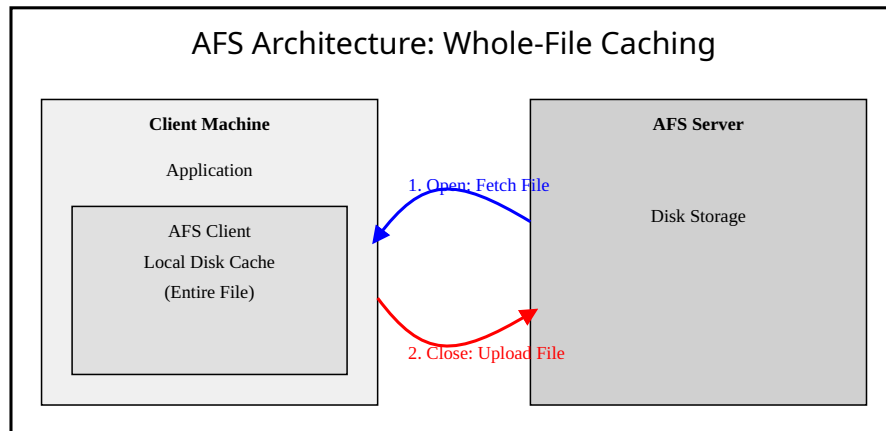

Figure 1: AFS Architecture

3. Cache Consistency with Callbacks

- When a client caches a file, it receives a **callback** from the server.
- If another client modifies the file and closes it, the server sends a **callback break** to all other clients that have the file cached.
- This tells the clients that their cached copy is now stale and they need to fetch the new version from the server.
- This approach significantly reduces server load compared to NFS's constant polling.
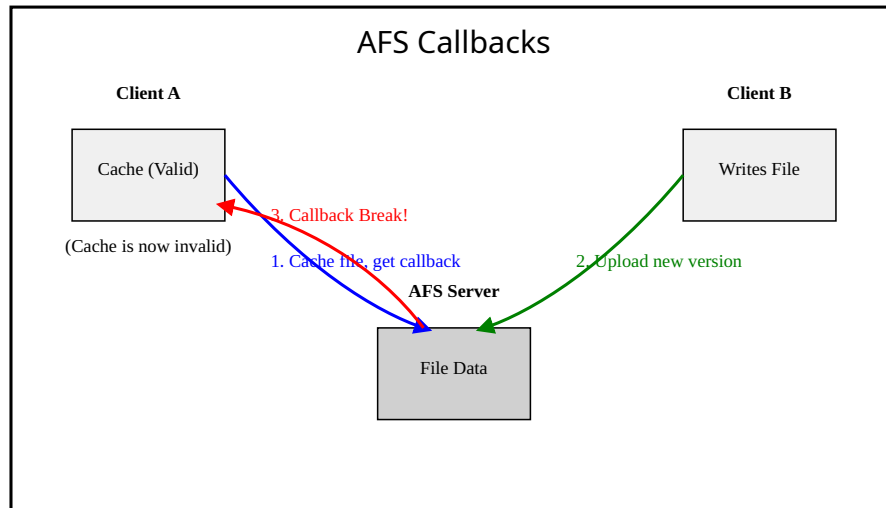
Figure 2: AFS Callbacks

4. AFS vs. NFS

Feature	AFS	NFS
Caching	Whole-file caching to local disk	Block-based caching in memory
Consistency	Callbacks (server-initiated)	Polling (client-initiated)
State	Stateful (server tracks callbacks)	Stateless
Performance	Better for whole-file reads	Better for small, random access

5. Other AFS Features

- **Global Namespace:** File paths are consistent across all client machines.
- **Security:** AFS has robust security features, including access control lists (ACLs).
- **File Identifiers (FIDs):** AFS uses FIDs to efficiently identify files after the initial path lookup.

6. Summary

- AFS was a pioneering distributed file system that introduced several important concepts, most notably callbacks.
- Its design choices were driven by the goal of scalability.

- While not as widely used as NFS today, the ideas from AFS have had a lasting impact on the design of distributed file systems.