Prof. Sheetal Jadhav

- **Case Study: Distributed File Systems (DFS)**

○ Network File System (NFS)

This NFS file system is a distributed file system that permits its client to access the file over the network. This file system is an open standard. That's the reason this file system can be implemented easily. Initially, this file system was created for experimental purposes, but later its second variety was released for public use after the first success.

All data is accumulated on one main system and all the remaining systems of the network can access the data stored on that as if it was stored in their local system. But here one problem arises. If the main system went down, then there is a high possibility of a loss of data and here the storage also relies on the space available on that system.

Here, a mount command is used for accessing the exported data. After the successful accessing of data, the client machine can interconnect with the file systems within the specified parameters.

○ Andrew File System (AFS)

In the 1980s, Andrew File System (AFS) was introduced by researchers at Carnegie Mellon University to mitigate the problem of scalability among distributed file systems. In the initial version of AFS, the whole requested file was cached on the local disk of the clients in order to increase the performance over multiple requests for the same file on the server.

Limitations of the AFS initial version are as follows:

High Path traversal time: In order to access the file on the client side the server has to traverse the complete path from the home directory to the location of the file. This traversal takes a considerable part of the server's processing time on the other hand which can be used by the server to process the other client requests.

High Traffic: In AFS, a huge amount of traffic is generated by the client side in the form of validation messages to check whether the file has been modified or not .

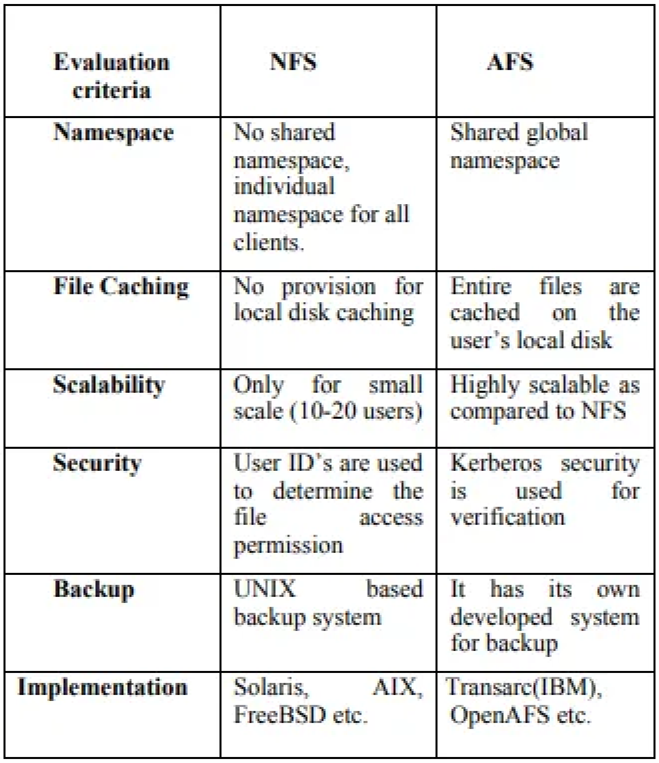
○ HDFS

It is a distributed file system that handles a large set of data that is running on commodity hardware and in which data is distributed among many data nodes or networked computers.

It is mainly used for enlarging a single Apache Hadoop cluster to hundreds and even sometimes thousands of nodes. It is considered one of the major components of Apache Hadoop. It is not similar to Apache HBase, which is a column-oriented non-relational DBMS that sits on top of HDFS, which can better support real-time data with its in-memory processing engine.

It is mainly used to store big data and also makes it responsible for faster data transactions.

This file system stores multiple replicas of files, that's why it is called fault-tolerant. Here the default replication level is 3.



Evaluation criteria	NFS	AFS
Namespace	No shared namespace, individual namespace for all clients.	Shared global namespace
File Caching	No provision for local disk caching	Entire files are cached on the user's local disk
Scalability	Only for small scale (10-20 users)	Highly scalable as compared to NFS
Security	User ID's are used to determine the file access permission	Kerberos security is used for verification
Backup	UNIX based backup system	It has its own developed system for backup
Implementation	Solaris, AIX, FreeBSD etc.	Transarc(IBM), OpenAFS etc.

Prof etal Jadhav

HDFS	NFS
It is a file system in which data is distributed among many data nodes or networked computers.	It is a file system or protocol which allows its client to access the file over the network.
It is mainly used to store and process big data.	It can store and process a small amount of data.
Its data blocks are dispersed on the local drives of hardware.	Data is stored on a single dedicated hardware.
Its data is stored reliably. Here, data is available even after machine failure.	No reliability, data is not available in case of machine failure.
It runs on a cluster of different machines, data redundancy may occur due to replication protocol.	It runs on a single machine, with no chance of data redundancy.
It is for multi-domain.	It is for a single domain.
Here, client identity is trusted by the OS.	Here, client identity is trusted by default.
It has different calls. It is mainly used for non-interactive programs.	It has the same system calls as O/S.

Prof. Sheetal Jadhav