**Distributed File Systems**

**Abstract-**  Distributed File Systems is a application based on client-server application that allows client to access data and in many cases it allow client to write data. For example, one can access a distant computer ‘s data from other location provided they have proper connection and setup. When a user access a file on the server, the server sends the user a copy of the file, which is cached on the user’s computer while the data is being processed and is then returned to the server.

1 Introduction

Distributed File Systems (DFS) is an application that organizes the file and directory services of individual servers into a global directory in such a way that remote data access is not location specific but is identical from any client. It is designed in such a way that it follows a hierarchical and directory-based application.

In order to make a multiple access permissible, the server must have a mechanism in place to organise updates so that the client always receives the most current version of data and that data conflicts do not arise.

Maintaining information during the time of access is very important. DFS uses a database replication or files to protect data loss against any data access failures.

**2. DFS Terms and Definitions**

To understand DFS more let us know some important terms and definition about Distributed file systems as below: -

1. DFS namespace- it is a virtual view of shared folders on different server as present on DFS. This DFS namespace consists of root and many links and targets and it starts with a root that maps to one or more targets.

2.DFS link- it is the component in DFS that lies below the root and maps to one or more links targets.

3. DFS path- it starts with a DFS root in any Universal Naming Convention.

4. DFS root – it is the starting point of DFS namespace. It resides on an NTFS volume A DFS root has one of the following formats: \\*ServerName*\*RootName* or \\*DomainName*\*RootName*

5. Domain-based DFs namespace- it contains multiple root targets, which offers fault tolerance and load sharing. It is stored in in active directory.

6. Link referral- it is list of link targets for a particular link.

7.Link target- it is the mapping destination of a link for example, a link target can be a shared folder or another DFS path.

8. Referral- A list of targets, transparent to the user, which a DFS client receives from DFS when the user is accessing a root or a link in the DFS namespace. The referral information is cached on the DFS client for a time period specified in the DFS configuration

9. Root target – it is also called root servers and in a physical server that hosts a DFS namespace, a domain-based DFS root can have multiple root targets.

**3. DFS Architecture**

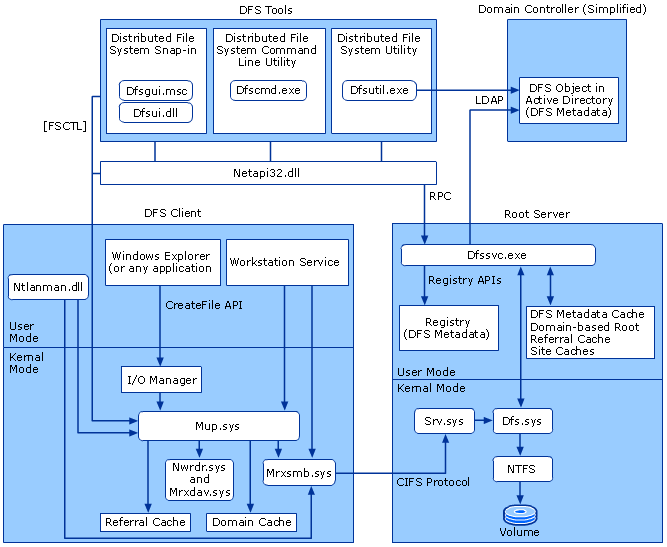
****

Fig 1. DFS Architecture

DFS architecture above DFS in three main blocks

1.DFS client

2.DFS server

3.DFS Tool with controller

DFS client and server are linked using a DFS tool and a controller. In this case, the domain controller also hosts the DFS metadata cache (regardless of namespace type) and the stand-alone DFS metadata in its registry (for stand-alone namespaces).

**4.DFS Implementation Using Client.py and Nameserver.py Python Codes**

Client can perform three actions with the server for example

A. read the file

B. write

C. delete

Server allows these action based on clients request. Server post, updates and deletes

.