University of Technology

قسم هندسة الحاسوب –الجامعة التكنولوجية

Department of Computer Engineering

Fundamentals of Operating Systems

Title

Distributed File Systems

Name:مهند محمد عزيز

Mohaned Mohammed Azez

Class:2nd | المرحلة الثانية

Branch: شبكات / مسائي | Network -Evening

سنة : ٢٠١٩ /٢٠٢٠

تاريخ التسليم : ٢٠٢٠/٨/٢٤

Abstract

The purpose of a distributed file system (DFS) is to allow users of physically distributed computers to share data and storage resources by using a common file system. A typical configuration for a DFS is a collection of workstations and mainframes connected by a local area network (LAN). A DFS is implemented as part of the operating system of each of the connected computers. This paper establishes a viewpoint that emphasizes the dispersed structure and decentralization of both data and control in the design of such systems. It defines the concepts of transparency, fault tolerance, and scalability and discusses them in the context of DFSs. The paper claims that the principle of distributed operation is fundamental for a fault tolerant and scalable DFS design. It also presents alternatives for the semantics of sharing and methods for providing access to remote files. A survey of contemporary UNIX-based systems, namely, UNIX United, Locus, Sprite, Sun's Network File System, and ITC's Andrew, illustrates the concepts and demonstrates various implementations and design alternatives. Based on the assessment of these systems, the paper makes the point that a departure from the extending centralized file systems over a communication network is necessary to accomplish sound distributed file system design.

What is DFS?

Distributed file system (DFS) is a method of storing and accessing files based in a client/server architecture. In a distributed file system, one or more central servers store files that can be accessed, with proper authorization rights, by any number of remote clients in the network.

Much like an operating system organizes files in a hierarchical file management system, the distributed system uses a uniform naming convention and a mapping scheme to keep track of where files are located. When the client device retrieves a file from the server, the file appears as a normal file on the client machine, and the user is able to work with the file in the same ways as if it were stored locally on the workstation. When the user finishes working with the file, it is returned over the network to the server, which stores the now-altered file for retrieval at a later time.

Distributed file systems can be advantageous because they make it easier to distribute documents to multiple clients and they provide a centralized storage system so that client machines are not using their resources to store files.

DFS Structure

- Service – software entity running on one or more machines and providing a particular type of function to a priori unknown clients.
- Server – service software running on a single machine.
- Client – process that can invoke a service using a set of operations that forms its client interface.
- A client interface for a file service is formed by a set of primitive file operations (create, delete, read, write).
- Client interface of a DFS should be transparent, i.e., not distinguish between local and remote files.

Why we need DFS?

As we discussed above the typical file sharing methods do have issues applying in to complex/ advanced network setups. Let's discuss about some of these common issues and why we should move in to advanced solution such as DFS.

Issue	Description
Availability	One of the biggest concerns for using traditional file sharing system for large network is availibity issues. When use simple file sharing methods, mostly data will be shared from one file server or few. When network getting larger demand of the data available in these file servers will increasing. May be the business critical information may share through these. If the server fails you will not be able to access it until it bring up again. May be will have to wait until backup is restored. For business it will be critical issues some time it may stop entire company operations. Of cause there can be disaster recovery plan in place but still there can be down time. Availability of the data is more important for any business as majority of business deals with digital data. Using traditional file sharing can cause single point of failure. As the demand of data in network share goes up we must ensure the availability of it.

Complexity of managing and also using	When network getting larger data will be shared from different servers for different purposes. So to access the share you need to type \\serverA for example in run command to access the files shared from serverA. But assume if there are 10 servers or more in network which is sharing files, you still need to remember all these paths with ip addresses or hostnames. If its hundreds of server this is not practical at all. If we look in to this in system administrator's end he/she will need to control all these shares with different permissions by log in to each server. But if there were method that users can use some kind of structure to access shares and also for administrator to manage these shares from central location it would have been more easy for everyone.
Load sharing (Load Balancing) Issues	When the network is getting bigger the number of users will be increase dramatically. So if you have file share server you may get hundreds or thousands requests to access files, update files etc. depending on the hardware, bandwidth link etc if the serving client numbers increased system will start to give performance issues. To address this we will need some sort of load balancing mechanism. But there are limitations on such process. To load balance the same data should be available in different servers and then need to distribute the access request among that group of servers. The issue is how we sync the data between all these servers in efficiency manner? Typically file sharing mechanism do not support for replication by default. If there is such proper replication method we can use it to configure the load balancing.

Branch office / different sites practical issues with file sharing	Some time as we discussed in previous some time the business may have Branch offices in different cities, countries etc. also some time it can be even from different active directory sites. But still those may require to share files among the braches, with headquarters etc. mostly these offices will be interconnect with using lease lines or vpn links and most of them will be slower links. So to copy a file from different branch file share can take a longer time. When the number of user request increase this will be slower and some time you may not even get the file as line bandwidth is maxed out. This is definitely effect on company operations. But if we can access same up to date file copy from a local office file share it is more fast. if files in HQ share folders can replicate with local office share folders it save the bandwidth also of the slow link. That way only branch office server will copy files through the link instead of hundreds of user requests.
Storage issues	When the volume of the data increase the storage requirement for file sharing will increase as well. Let's say you have a file server which support maximum of 1TB of storage. With data volume increase the server is reaching in to its maximum storage. There for no way you can continue this server and add more storage to the same share folder from different server. For ex- if you have share called \\serverA\Data you cannot add storage from different server to same share name if its reached the storage limit. So you will need to move in to new server and move the data. But if there is way we can add storage easily to the share from different server instead going n to new server will avoid downtimes and other access issues.

Distributed File System (DFS) Functions:

The Distributed File System (DFS) functions provide the ability to logically group shares on multiple servers and to transparently link shares into a single hierarchical namespace. DFS organizes shared resources on a network in a treelike structure. DFS supports *stand-alone* DFS namespaces, those with one host server, and *domain-based*namespaces that have multiple host servers and high availability. The DFS *topology data* for domain-based namespaces is stored in Active Directory. The data includes the DFS root, DFS links, and DFS targets.

Each DFS tree structure has one or more *root targets*. The root target is a host server that runs the DFS service. A DFS tree structure can contain one or more *DFS links*. Each DFS link points to one or more shared folders on the network. You can

add, modify and delete DFS links from a DFS namespace. When you remove the last target associated with a DFS link, DFS deletes the DFS link in the DFS namespace. (In earlier documentation, DFS links were called junction points.)
A DFS link can point to one or more shared folders; the folders are called *targets*. When users access a DFS link, the DFS server selects a set of targets based on a client's site information. The client accesses the first available target in the set. This helps to distribute client requests across the possible targets and can provide continued accessibility for users even when some servers fail.
An application can use the DFS functions to:

- Add a DFS link to a DFS root.
- Create or remove stand-alone and domain-based DFS namespaces.
- Add targets to an existing DFS link.
- Remove a DFS link from a DFS root.
- Remove a target from a DFS link.
- View and configure information about DFS roots and links.

Targets on computers that are running Microsoft Windows can be published in a DFS namespace. You can also publish any non-Microsoft shares for which client redirectors are available in a DFS namespace. However, unlike a share that is published on a server that is running Windows Server, they cannot host a DFS root or provide referrals to other DFS targets.
DFS uses the Windows Server file replication service to copy changes between replicated targets. Users can modify files stored on one target, and the file replication service propagates the changes to the other designated targets. The service preserves the most recent change to a document or files.

Conclusion :

Since the earliest days of distributed computing, file systems have been the most important and widely-used form of shared permanent storage. The continuing interest in distributed file systems bears testimony to the robustness of this model of data sharing. We now understand how to implement distributed file systems that span a few hundred to a few thousand nodes. But scaling beyond that will be a formidable challenge. As elaborated in the preceding section, availability, heterogeneity and support for databases will also be key issues. Security will continue to be a serious concern and may, in fact, turn out to be the bane of large distributed systems. Regardless of the specific technical direction taken by distributed file systems in the next decade, there is little doubt that it will be an area of considerable ferment in industry and academia.

References :

1. https://www.rebeladmin.com/2014/06/dfs-distributed-file-system/
2. https://docs.microsoft.com/en-us/windows/win32/dfs/distributed-file-system-dfs-functions
3. https://www.researchgate.net/profile/Abraham_Silberschatz/publication/220566486_Distributed_File_Systems_Concepts_and_Examples/links/0912f51373df96fe26000000/Distributed-File-Systems-Concepts-and-Examples.pdf