Impact Factor (SJIF): 4.542

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 4, Issue 4, April-2017

A Taxonomy and Survey on Different File Systems

Hemil Patel[#1], Shibin Sunny[#2], Ankit Gajera[#3], Prof. Gopi Bhatt[#4]

[#]*Computer Engineering Department, A. D. Patel Institute of Technology*
New Vallabh Vidhyanagar, Vitthal Udyognagar, Anand, Gujarat-388121, India

Abstract— *When taking computation into consideration, storing and retrieval of data is controlled by file system. If the computing system lacks proper file system, stored information would just be a large piece of data. Without it, there would not be any way to tell where one piece of information ends and where is the beginning of the next piece. The structure and logic rules used to manage the groups of information and their names is called a "file system" [1]. In this paper, we develop a comprehensive taxonomy for describing various file systems and use this taxonomy to survey several existing file systems. We use the taxonomy and the survey results to identify which file system is appropriate to use according to the requirements of an individual.*

Keywords— *File System, Disk File System, Network File System, Cluster File System, Distributed File System, Parallel File System*

.

I. INTRODUCTION

At present, computer market offers diverse opportunities for storing huge amount of information, either personal or corporate, in digital form. In a computer, different types of data like documents, music, videos, presentations, pictures, are stored in the form of file. For storage and retrieval of files from a computer, file system is used. Thus, the component of an operating system that is tasked with organizing, storing and retrieving data is called File System [2]. Naming conventions for file, which includes number of characters used in the name, type of characters, length of file name suffix etc. are specified by file system. It also determines the format specifying the path to a file. Some features of file system include:

⇒ Data access
Restricting and permitting access to data is one of the several features of filesystem. The main intention behind controlled access to data is to prevent an individual user or a group of users from reading or modifying files.

⇒ Space Management
It is the responsibility of file system to organize files and directories. File system allocates space to the data for a new file created. Some file system specifies the space needed initially and allocates it to the file. Subsequently, it increments the allocation as file grows. When a user deletes the file, the space allocated to that file is eventually used by other files. This in turn creates used and unused space of various sizes, which is known as free space fragmentation. If there are no contiguous space available for a file during its initial allocation, the space is assigned in fragments. Similarly, when a file requires more than its initially allocated space, file is fragmented and space is allocated elsewhere.

⇒ File Naming
A storage location in file system is identified by a file name. File naming differs for different file system. Some file system is case sensitive while some are not. Again, file names can contain a wide range of characters.

⇒ Metadata
A data describing other data is known metadata. For example, metadata of a file can include name of the file, length of the contents described inside the file, location of the file, time that file was created, time when it was last modified, time of last access etc.

File systems can be classified into three main categories:

a) Local Storage based
File systems used in local storage devices like hard disks, pen drives, memory cards etc. Examples include FAT, NTFS, ext2, ext3, ext4, UDF etc.

b) Shared Storage based
File system used in shared storage allows multiple users to access data at the same time. Examples include IBM General Parallel File System, Oracle Cluster File System etc.

c) Distributed Storage based

File systems used to manage data that is designed to scale to a very large size. This type of file system allows to access and process data as if it were stored on a local PC. Examples include Hadoop Distributed File System, Glustre File System, Google File System etc.

This paper includes a survey on the filesystems included in above three categories. Various parameters of the file systems are discussed in detail. Then a comparison between some file systems from each category is described in a tabular format. The findings for each category of file system, from each comparison is also described in detail.

II. TAXONOMY OF FILE SYSTEM

The motive behind developing this taxonomy is to study the features that develops different file systems and this helps in selecting appropriate file system that performs better under current circumstances, having more fault tolerance and provides security.

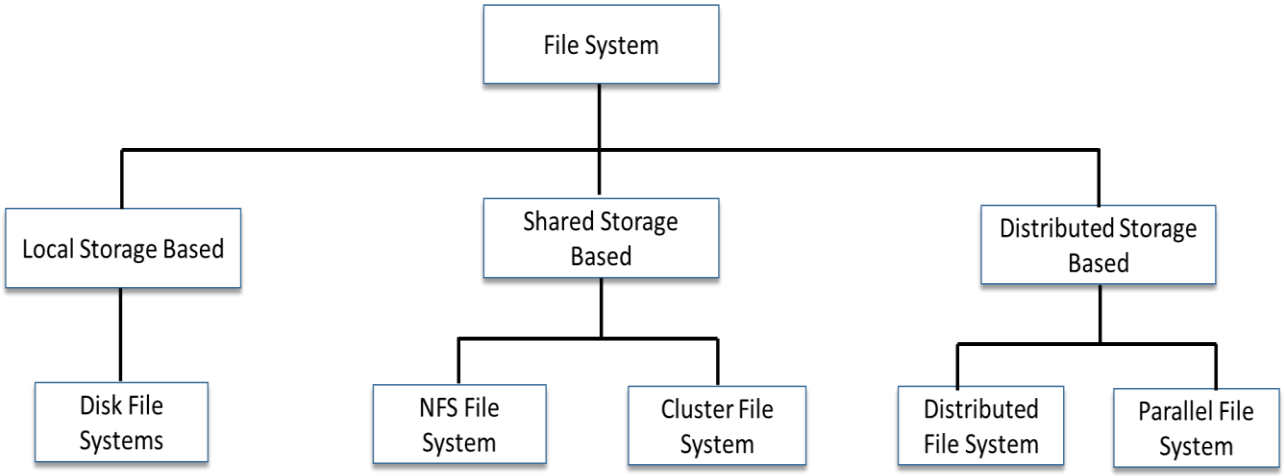

Figure 1: Taxonomy of File System

❖ LOCAL STORAGE BASED

A group of data or information is called a file and the structural and logical rules defined to manage these files is called "File System". Applications in a laptop or PC uses Local File system to store and retrieve files from storage. Files in Local File System are placed in a hierarchical structure. One of the main function of file system is to specify conventions used for naming files and specifying the position of the file in the structure tree. Each file system consists of many dynamic-link libraries and drivers. This defines the features and format of file system. Devices like Portable Hard Drive, Optical Disks, Memory Cards, and Jukebox contain file systems.

➢ Maximum File Size

File size is used to measure the capacity of data, which can be stored, by a computer file. This indicates the amount of storage it will consume when stored on any data storage. The unit of file size is byte but prefix such as megabyte and gigabyte is added to for indicating the file with larger size. The storage is divided into disk sectors. A sector is the smallest amount of space addressable by the file system [3]. Usually the sector size is several thousand bytes. If the sector size is reduced, denser use of disk space can be done but this will reduce the efficiency at which the file system is operating. When a file is provided to the file system, it might consume a little more disk space than the original size of the file because file system rounds up the size to include the unused space left in the disk sector.

➢ Maximum Volume Size

Using a volume manager, an administrator can allocate space for specific applications or data, move volumes as he/she wishes, typically with no downtime [4]. File systems having variable block/cluster sizes, a range of size are given, indicating maximum volume size for minimum and maximum possible allocation sizes of file system [5].

- File System Level Encryption

The process of converting data to an unrecognizable form or encrypted form is known as Encryption [6]. Several methods for encryption of user data are included in file system. With the help of file system level encryption users can encrypt individual files or directories. Unlike Full-disk encryption, where the whole disk of the file system is encrypted; in File system level encryption just individual files and directories are encrypted by file system itself. This gives the advantage of flexible file based key management where each file can be encrypted with separate encryption key [7].

- Sparse File

Sparse File is an empty computer file, which uses the file system space efficiently. The default value of sparse file is zero i.e. it contains zero as data. The metadata, which represents the empty block, is written to the disk instead of the actual empty space that takes up the block. This allows file to use less disk space. Sparse files are usually used for disk images, snapshots of database, log files and in scientific applications [8]. The main advantage of using sparse file is that even if there is storage insufficiency on the file system, large file can be created.

- Built-in Compression

The process of modifying, encoding, or converting the bit's structure of data in such a way that it consumes less space on disk is known as Data Compression [9]. The main advantage of compressing data files is that it allows you to store more files than available storage space. Some file system has built in compression feature available within them. With the help of a few clicks, one can compress files, thus taking less space on the hard drive. As it is a built in compression, files can be accessed normally without zipping or unzipping.

- Metadata-only Journaling

Some file system maintains a certain file called journal, which repairs irregularities that occurs because of crashing of PC or an improper shutdown. Such file system is called journaling file systems. This type of shutdown or crash occurs due to interruption in power supply or due to a software problem. Journaling file system writes metadata into a journal that is written to hard disk drive of the computer. If the system crashes or shuts down, if updates done to a file has not yet been fully committed, the file system will read the journal and the loads the most recent point of data consistency

Table-1: Comparison of Disk Based File System

Parameters	NTFS	FAT32	UDF	exFAT
Max File size	2^{64}-1 bytes	4 GB	2^{64}-1 bytes	2^{64}-1 bytes
Max Volume size	16 TB (4 KB Cluster Size) or 256TB (64 KB Cluster Size)	2^{32} blocks	2^{32} blocks	2^{32}–1 clusters (Maximum cluster size = 2^{25} –1)
File System Level Encryption	Yes	No	No	No
Sparse File	Yes	No	Yes	No
Built-in Compression	Yes	No	No	No
Metadata-only Journaling	Yes	No	No	No

- SHARED STORAGE BASED

As size of application increased, the storage required to store the data also increased. Affording a large local storage for the data was not possible by all of them. To overcome this limitation, a new concept came into existence called

external storage. External storage has much more feature in compared to local storage like sharing the storage space with other standalone systems.

- Supported Operating System

Operating System is a software which directly interacts with hardware. Without Operating system, any device would be a piece of useless components. Different operating system requires different environment to run. Not every operating system is supported by every hardware. An important part of operating system is file system. File system decides how the data is stored in a storage. File system has to be created for specific use only. So a particular file system might run on only few of the operating system or it might even run on only one of the many operating systems.

- Licensing

License is a legal body, which decides how the application is used or redistributed which grants the permission to the end user to use the software or generate the one or more copies, conserving the exclusive rights under copyright law. There are majorly two types of licensing under copyright law, namely proprietary software and free and open source software (FOSS). The main difference between both is granting the rights to modify and reuse software product [10]. FOSS allows user to access and modify the source code whereas in proprietary, user is not allowed to modify the source code as well as the source code is kept hidden and user cannot access it.

- File System Permission

Most of the file system gives permissions to the specific user or group of users which gives users the ability to view, navigate, change and execute the files and folders on the file systems [11]. There are three permissions given to any file. These are Read, Write and Executive. The symbol notation for read, write and execute is r, w and x respectively. There are three permission triads. The first triad shows what the owner can do. The second triad shows what the group members can do. The third triad shows what the other users can do. The format for file permission looks like -rwxrwxrwx, all the permissions are given to all the users.

- Transparent Encryption

Transparent Data Encryption is a technology used by IBM, Oracle and Microsoft to encrypt their database files. TDE encrypts the data at file level, which solves the major problem of protecting stored data. TDE does not protect the data which is being used or transferred.

Table-2: Comparison of Shared Storage based File System

Parameters	OCFS 2	GPFS	CXFS	Red Hat GFS 2
Max File size	4 PB	8 EB	8.5 EB	8 EB
Max Volume size	4 PB	8 YB	17 EB	8 EB
License	Open Source	Proprietary	Proprietary	Open Source
File System Permission	Unix permissions, ACLs and arbitrary security attributes (Linux 2.6 and later)	POSIX	Supported	Unix permissions, ACLs and arbitrary security attributes
Supported Operating System	Linux	AIX, Linux, Windows Server	Server: IRIX, Linux, Clients: IRIX, Solaris, Linux, Mac OS X, AIX, Windows	Linux
Transparent Encryption	No	Yes	No	No

❖ DISTRIBUTED STORAGE BASED

Storing of data has evolved during the years in order to accommodate rising needs of companies and individuals. With the combination of networking and storage, a platform has been created with a large number of possibilities, which allows distributed storage to adopt vast and varied roles, which fall well beyond data storage [12]. The main aim of distributed storage based file system is to allow users to share same data and store resources using a common file system when they are physically using different workstations. These file systems are implemented as part of the operating system and rests above local file system.

➢ Naming

Mapping between logical and physical objects is called naming. Naming plays an important role as each object in the file system has a logical path name and physical address associated with it. The distributed namespace includes a collection of all logical path names. In order to retrieve information from distributed system, access to objects is required for which physical address is used.

➢ Process

The main concern regarding distributed storage based file systems is whether the process should be stateless or not. The advantage of a stateless approach is simplicity. However, it will be difficult to follow during implementation because a stateless server cannot lock a file easily [13].

➢ Communication

Remote Procedure Call is a protocol that a program uses to request service from a program located in another computer without having to understand the network details. Most of the distributed storage based file system uses this protocol to communicate. This makes the system independent of the underlying operating system. In this type of approach, TCP and UDP are the communication protocols considered out of which TCP is used by most of the file systems.

➢ Synchronization

It is necessary to decide the semantics of reading and writing of same file shared by two users in order to avoid problems. Thus, synchronization becomes vital issue that is to be analysed for distributed storage based file system. Few approaches that are available include UNIX semantics, Sessions semantics etc. Second issue in synchronization include file locking system. Some systems choose to give locks on objects to clients, and some choose to perform all operations synchronously on the server [14].

➢ Consistency and Replication

File replication is a useful redundancy for improving the availability of file. The basic requirement from a replication scheme is that different copies of the same file reside on failure-independent machine [15]. It would be desirable if the details of replication were hidden from the user. Replication can be obtained in two ways i) Client-side caching ii) Server-side Replication. Most distributed storage based system uses checksum to validate data after sending through communication network. This provides consistency

➢ Fault Tolerance

A property that enables a system to continue operating properly in the event of failure. In distributed system, fault tolerance is very much related to the replication feature. There are two approaches used for fault tolerance: failure as exception and failure as norm. Failure as exception isolates the nod which has failed while failure as norm employ replication of all kind of data and executes re-replication whenever replication becomes unsafe. Thus, main advantage of fault tolerance design is that the system would be able to complete its intended operation without failing completely.

Table-3: Comparison of Distributed Storage based File System

Parameters	Lustre	Panasas	Hadoop	PVFS2
Naming	Central Metadata Server	Central Metadata Server	Central Metadata Server	Metadata distributed in all nodes
Synchronization	Hybrid locking mechanism, using leases	Give locks on objects to clients	Write-once-read many, give locks on objects to clients, using leases	No locking method, no leases
Process	Stateful	Stateful	Stateful	Stateless
Communication	Network Independence	RPC/TCP	RPC/TCP & UDP	RPC/TCP
Fault Tolerance	Failure as exception	Failure as exception	Failure as norm	Failure as exception
Consistency and Replication	Server side replication – Only metadata replication, Client side caching, checksum	Server side replication – Only metadata replication	Server side replication, Asynchronous replication, checksum	No replication, relaxed semantic for consistency

III. CONCLUSIONS

File system is a major component of operating system. Without it, proper storing and retrieval of data is not possible. Selection of proper file system depends on the requirement of the user and the environment in which the file system will be used. Taking into consideration such parameters like Maximum file size, communication, replication, encryption, compression, licensing, permissions, operating system, etc. an appropriate file system is selected.

REFERENCES

[1] A. S. Tanenbaum, "*Modern Operating Systems"*, 3rd ed., Prentice Hall, Pearson's Education, 1990.
[2] https://www.computerhope.com/jargon/f/filesyst.htm
[3] https://en.wikipedia.org/wiki/File_size
[4] http://searchstorage.techtarget.com/definition/volume-manager
[5] https://en.wikipedia.org/wiki/Comparison_of_file_systems#cite_note-note-4-9
[6] https://techterms.com/definition/encryption
[7] https://en.wikipedia.org/wiki/Filesystem-level_encryption
[8] https://wiki.archlinux.org/index.php/sparse_file
[9] https://www.techopedia.com/definition/884/data-compression
[10] https://en.wikipedia.org/wiki/Software_license
[11] https://en.wikipedia.org/wiki/Software_license
[12] http://www.cloudbus.org/reports/DistributedStorageTaxonomy.pdf
[13] https://www.computer.org/cms/ComputingNow/homepage/2010/0310/T_CONF_TaxonomyAndSurveyonDistributedFileSystems.pdf
[14] https://www.computer.org/cms/ComputingNow/homepage/2010/0310/T_CONF_TaxonomyAndSurveyonDistributedFileSystems.pdf
[15] http://www.cs.virginia.edu/~zaher/classes/CS656/levy.pdf