ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Window ReFS File System: A Study

Ms. Santwana Gudadhe[1], Mrs. Radha Deoghare[2], Ms. Pallavi Dhade[3]

Computer Engineering Department PCCOE, Pune[123]

Abstract: Today file system is the important concept in the design of computer operating system. Fie system is used to store and retrieve data. With the file system it is easy to identify any information stored in it. Various file systems are available in market. Some file system are design for specific application. Today as demand is increasing for large storage space, most researchers are paying attention to increase the storage capacity of file system. Recently window release the new file system ReFS called Resilient File System supports the large storage requirement.This paper presents the overall study of Window ReFS file system. In this paper, ReFS goals and features are discussed in section 1 and 2. Section 3 include the detail study of ReFS. And finally comparison and related study given in section 5 end with the conclusion.

Keywords: ReFS,NTFS, Bit-rot,salvage

I. Introduction

In operating system, file system plays a important role. File system is used to control how data is stored and retrieved. Without file system, information placed in storage area would be one large body of text with no way to tell where one piece of information stops and next information begins. If data is separated into individual piece and give the name then information is easily separated and identified.

There are many different types of file system are available. Some file system has been designed to be used for specific application. Each file system has its structure, speed, flexibility,security and size etc...

Windows has FAT, NTFS, exFAT and ReFS file systems. Recently, to fulfill the storage requirements of window users ReFS has been built. Resilient file system specifically built to increase data integrity, scalability and availability It offers the ability to handle large storage volumes (maximum volume size of 1 Yobibyte).[2] It allows shared storage pools across multiple machines and also provides more resiliency to corruption.

ReFS (Resilient File System) is the file system built for next-generation operating systems, such as Windows Server 8, Window server 12.

ReFS maintains a high level of compatibility with NTFS. Most of the features and semantics from NTFS like Bit Locker encryption, access-control lists for security, USN journal, change notifications, symbolic links, junction points, mount points, reparse points, volume snapshots, file IDs, and oplocks are inherited by ReFS.

II. Essential Features of REFS

A. Metadata Integrity.

The ReFS file system , Meta-data and file-level integrity checksums is used to provide advanced-level protection against data corruption. File system metadata is always protected. Optionally, user data can be protected on per-file,per-directory,per-volume basis. If data corruption occurs, ReFS can detect and, when configured with storage spaces, automatically correct the corruption [1].

B. Scalability

ReFS has an ability to support arge amounts of data storage keeping in mind with future data requirements . It supports up to 16 Exabyte of data storage, which can be scaled to 256 Zetabytes without performance impact.[3]

C. Scrubbing

The phenomenon called “bit-rot” means damaging the contents of disc and arrays little by little. Bit rot is rarely obtained by users and applications because it develops over time and usually able to see on disk locations as well as on directories [4] . Bit rot is difficult to identify as it is a form of data corruption.

Scrubbing is one of the major technique in ReFS file system , which identify and eliminates bit rots. Scrubbing performs some activities like reading, searching and validating. By performing such activities on each copy of data object using checksums , scrubbing manages and removes bit rot. If there is change in checksum and data mismatch then the corrupted copy is replaced by good copy. The FILE_ATTRIBUTE_NO_SCRUB_DATA attribute is used to specify the data whose integrity is maintained by third party applications[5].

D. Availability

ReFS is designed to prioritize the availability of data. With ReFS, if corruption occurs, the online salvage process is available. This online salvage process is localized to the area of corruption, it performs online. ReFS performs all repair operations online, it does not have an offline chkdsk command[8]. B+ tree is used for managing data and separating the corrupted part of file system.

E. Integrity Stream

ReFS has feature known as “Integrity stream” to fight against file corruption. If this feature is enabled , if at any point, a file has been modified then the copy of that file is available on different area of the disk. And because of any reason , if it is corrupted or lost, then original file is remain

DOI 10.17148/IJARCCE.2015.44138

at its own location. ReFS gives window a way to find the original file which has not been corrupted.

F. Storage pooling and virtualization

File system can be created and easily managed by Storage pooling and virtualization. Storage pooling and virtualization is an important and new concept for storage management in Windows Server 2012 [5]. In which, different types of disk group, are collected in a pool and they are “virtualized”. This disk of pool is acting as a unique resource for the creation of a Storage Space accessible by the system. After creating the storage pool it can be extended by adding additional disk, which gives the maximum storage space.

G. App Compatibility

The file system API, is the most critical and technically challenging area for compatibility. ReFS reuses the code for implementing window file system semantics to avoid the compatibility issue regarding semantics. This code implements the file system interface (read, write, open, close, change notification, etc.), maintains in-memory file and volume state, enforces security, and maintains memory caching and synchronization for file data[]. This reuse ensures a high degree of App compatibility.

III. Resilient file system (ReFS)

ReFS, stands for Resilient File System, built on the foundations of NTFS. For a new generation of storage technologies, ReFS file system has been architected and developed to maintain the crucial compatibility. But some features available in NTFS, it is not supported by ReFS such as named streams, sparse files, and quotas. ReFS works with Storage Spaces, a system that creates a pool of hardware and virtual storage devices and ensures the protection of data on that system.

ReFS has been designed from the base to meet a broad set of customer requirements. The Design goals of ReFS are listed below.

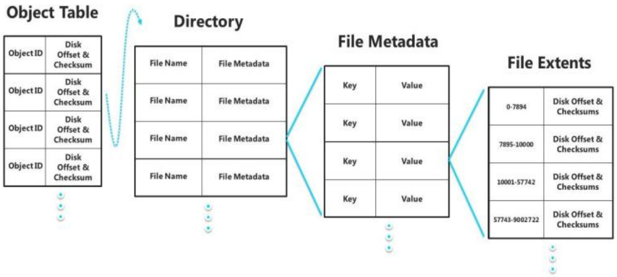
The key goals of ReFS are [5]:

1. Maintain a high degree of compatibility with a subset of widely adopted NTFS features.
2. Data should get automatically corrected and verified if corruption occurred because of any reasons.
3. For optimizing extreme scale, ReFS use the scalable structures for everything.
4. Never take the file system offline. This is helpful to identify the fault if any corruption while allowing access to the rest of the data storage.
5. Provide a resiliency architecture when used in conjunction with the Storage Spaces feature.

REFS FILE STRUCTURE

For Storage engine , ReFS uses the B+ trees. It is just like the single common on-disk structure to represent all information on the disk. Trees can be embedded within other trees (a child tree’s root is stored within the row of a parent tree). On the disk, trees can be very large or it can be compact with few keys embedded in another structure or multi-level[1,5]. This structure gives the scalability up and down for all aspects of the file system. Having a single structure significantly simplifies the system and reduces code. The new engine interface includes the “tables” that are countable sets of key-value pairs. Most tables have a unique ID by which they can be referenced. A special object table indexes all such tables in the system.

Fig. 1. File Structure



As shown in the figure above, directories are represented as tables. Because of implementation of table using B+ trees, directories can be scale efficiently, becoming very large. Files are implemented as tables embedded within a row of the parent directory, itself a table (represented as File Metadata in the diagram above).

The rows within the File Metadata table represent the various file attributes. The file data extent locations are represented by an embedded stream table, which is a table of offset mappings (and, optionally, checksums). This means that the files and directories can be very large without a performance impact, eclipsing the limitations found in NTFS[5].

CAPACITY LIMITS OF REFS

The table below shows the capacity limits of the on-disk format of ReFS[5].

TABLE 1 CAPACITY LIMIT OF REFS

Attribute	**Limit based on the on-disk format**
Maximum size of a single file	2^64-1 bytes
Maximum size of a single volume	Format supports 2^78 bytes with 16KB cluster size (2^64 * 16 * 2^10). Windows stack addressing allows 2^64 bytes
Maximum number of files in a directory	2^64

DOI 10.17148/IJARCCE.2015.44138

Maximum number of directories in a volume	2^64
Maximum file name length	32K 255 unicode characters (for compatibility this was made consistent with NTFS for the RTM product)
Maximum path length	32K
Maximum size of any storage pool	4 PB
Maximum number of storage pools in a system	No limit
Maximum number of spaces in a storage pool	No limit

IV. Comparison of REFS with NTFS

ReFS was designed to replace NTFS rather than FAT. The size of data storage devices and the number of stored files increase, NTFS is lacking in scalability. The following section discuss the comparison between ReFS and NTFS.

ReFS is faster than NTFS. It's main purpose is its resiliency.

ReFS handles Long File Names and File Path. The ReFS format has maximum volume size of 262,144 Exabytes while NTFS has volume size of 16 Exabyte. Both NTFS and ReFS stores the file information in metadata [7].

The biggest difference between ReFS and NTFS is the way it updates the metadata. In NTFS , the metadata updation is nothing but the “in-place”. When a new folder is created and if you assign a new name to the folder then its actual metadata itself is written over. In case of power failure while updating the disk, the metadata is overwritten causing data corruption.

ReFS creates a new copy of the metadata instead of metadata in-place. Once the new copy is created, all the write operations are taken place with the new copy. This important feature of creating a copy of the metadata permits us to recover from power failure without disk corruption.

ReFS performs better Detection and Repair Problems with the storage spaces. With NTFS, even a small amount of data corruption can cause big problems. But in ReFS , this problem of data corruption is overcome by salvage. Even a single file which is corrupted is removed by salvage.

Although ReFS is a different file system, there are similar features between ReFS and NTFS. Following table shows the features of NTFS and ReFS[6].

Table 2 Comparative study

NTFS Features	**ReFS Features**
Supports Case-sensitive filenames	Supports Case-sensitive filenames
Preserves Case of filenames	Preserves Case of filenames
Supports Unicode in filenames	Supports Unicode in filenames
Preserves & Enforces ACL's	Preserves & Enforces ACL's
Supports file-based Compression	Supports Sparse files
Supports Disk Quotas	Supports Reparse Points
Supports Sparse files	Supports USN Journal
Supports Reparse Points	Supports Open By FileID
Supports Object Identifiers	
Supports Encrypted File System	
Supports Named Streams	
Supports Transactions	
Supports Hard Links	
Supports Extended Attributes	
Supports USN Journal	
Supports Open By FileID	

V. Conclusion

ReFS based on previously existing NTFS code base. But some of the features such as named space,sparce file and quotas will not supported by ReFS. ReFS work with storage space. From the analysis it is observed that, ReFS is much faster than NTFS. Also as a file system, ReFS is not only good for resiliency, it is great for maintaining extremely large amounts of data. ReFS has lots of feature such as metadata integrity, Availability of data, Scaling can be possible at extended level. Scrubbing to identify and eliminate the bit-rot phenomenon, App compatibility. From the study, ReFS forms the foundation of storage on window for the next decades and more. ReFS proving to be the promising file system for massive data storage.
In future ReFS will be the massively deployed next decade file system.

References

[1] http://msdn.microsoft.com/en-us/library/windows/desktop/hh848060%28v=vs.85%29.aspx(last referred on 5/12/2014)

[2] www.easeus.com/resource/drive/refs.htm (last referred on 10/12/2014)

DOI 10.17148/IJARCCE.2015.44138

[3] http://www.awesomecloud.com/blog/cloud-insights/refs-file-system/#.VIfjwtY8g8o (last referred on 10/12/2014)

[4] www.refs-data-recovery.com/new-feature-in-refs.aspx (last referred on 11/12/2014)

[5] http://blogs.msdn.com/b/b8/archive/2012/01/16/building-the-next-generation-file-system-for-windows-refs.aspx (last referred on 10/12/2014))

[6] http://blogs.technet.com/b/askpfeplat/archive/2013/01/02/windows-server-2012-does-refs-replace-ntfs-when-should-i-use-it.aspx(last referred on 11/12/2014))

[7] http://www.petri.com/4-reasons-refs-is-better-than-ntfs.htm. (last referred on 12/12/2014))

[8] Www.theithollow.com/2014/01/microsofts-resilient-file-system-refs (last referred on 10/12/2014))

[9] http://en.wikipedia.org/wiki/ReFS (last referred on 11/12/2014))

DOI 10.17148/IJARCCE.2015.44138