Comparing NTFS File System with ETX4 File System

Valbona Dhjaku
Natural Science Faculty
Informatics Department
valbona.dhjaku@bankacredins.com

Nevila Xoxa
Natural Science Faculty
Informatics Department
nresulaj@yahoo.com

Albion Bame
Faculty of Information Technology
abame@msn.it

Igli Tafa
Polytechnic University of Tirana
itafaj@gmail.com

CEUR-WS.org/Vol-2280/paper-26.pdf

Abstract

This paper presents a study and analysis of ETX4 file system and data structures comparing them with the NTFS File System of Windows Operating System. It includes descriptions of ETX4 File System Features, ETX4 Disk Layout and ETX4 extent tree. All the components that we mentioned above will be in comparison with the adequate features of Windows Operating System that corresponds to the Features of ETX4 File System. Here we will talk about the volum size that support each of the file system and the speed of each file system including their performance.

Keywords: NTFS, ETX4, performance, cluster, iNode, hard drive sector, filesystem, features, volume size, kernel, defragment, timestamps, blocks.

1. Introduction

The purpose of this document is to attempt, in a straightforward technical manner, to describe the process of comparing NTFS file system with ETX4 file system with their advantages and disatvantages. Some of the concepts discussed in this document are rather difficult to understand without a proper background in programming. Since linux kernel 2.6.28, ETX4 has been included in the mainstream and has become the default filesystem with most distributions. In a very short span of time, it has grown in popularity as well as stability. This article aims to describe better this two file systems with their advantages and distavantages. This two file systems are widely use and we will give some point that we will use to compare this two file systems are as below:

- File System Volum Size
- Timestamps
- Disk Layout

2. Related Works

With NTFS and ETX4 file system is been done various work and some of them inspired me write this article. The information I had about this two file systems after working for almost four years with Windows and Ubuntu operating system helped me the most. I saw the need to expand my knowledge about this operating systems and the file system that both of them use seemed a good starting point for me to do this. For getting more knwoledges for this file systems i had to read a lot off articles and some of them are listed in references and others not. I have read o lot of information on different forums to and i do not regret it because it was very helpful to me.
The focus of this article is not the application of this two file systems but more the theoretical part of them to understand more how they work and about their performance.

3. Experimental Theory

3.1 File System Volum Size

ETX4 file system is the latest file system that Linux Operating System is now using and this file system supports the volume size up to 1 EB and file size up to 16 TB because is a 64-bit FS. This file system extends EXT2 and EXT3 file system and can map up to 128 MB of contionus space and a block with size up to 4 KB. ETX4 file system uses iNode with size of 256 Bytes. In ETX4 file system, are not allowed volumes larger than 16 TB. [MCB07]

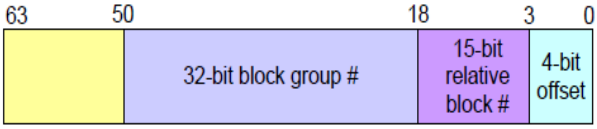

Figure 1: 64 bit i-node layout

NTFS, in theory reaches the maximum volume size $2^{64}-1$. However, in practice, the maximum NTFS volume size is $2^{32}-1$ clusters partly for the limitations of table partition. By using the cluster with the default size of 4 KB, NTFS file system can reach the volume size of (16 TB – 4 KB) because the Master Boot Record (MBR) supports file size up to 2 TB. The maximum file size in NTFS file system is 16 EB minus 1 KB. Since NTFS bases on clusters, we are giving the maximum file size of a cluster that is 64 KB.

Maximum size of files in Windows Server 2012 that runs NTFS file system si (16 TB - 64 KB). [KKPLSh02]

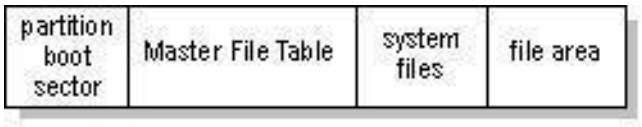

Figure 2: Formatted NTFS Volume Layout

3.2 Disk Layout

ETX4 allocates the space of a storage in units called "blocks". A block is a group of sectors in drive between 1 KB and 64 KB, and the number of sectors must be power of 2. Blocks are in turn, grouped into larger units called block groups. The size of the block is 4 KB and specified at mkfs time. By default, an ETX4 file system can contain up to 2^{32} blocks. If the 64 bit feature of ETX4 file system is enabled it can have up to 2^{64} blocks. [EXT418]

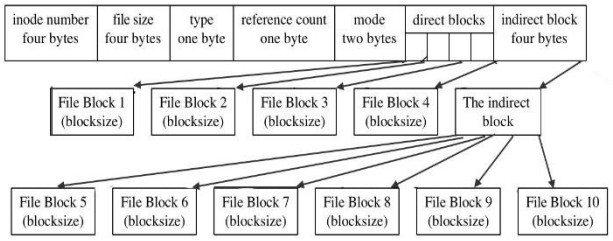

Figure 3: Inode Layout

Windows operating system with all the file systems that uses organize the hard drive with so called cluster size or as they are called sometime allocation unit size. The size of the cluster that is used represents that little amount of space in disk that holds a specific file. When file sizes are not the result of multiple cluster size additional, the disk space will be in use of holding those files. On a normal hard drive partition, the average space of the space that is lost in the disk calculated by the equation below:

*(Cluster size)/2 * (number of files)* [EXT418] [Ngu17]

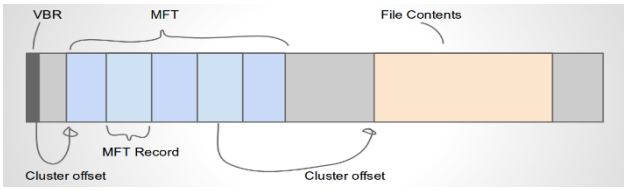

Figure 4: Cluster Layout

3.3 Timestamps

In ETX4, are four timestamps recorded in the lower 128 bytes of the inode structure. These four fields are 32-bit signed integers that represent seconds. For inodes that are not linked from any directory but are still open (orphan inodes), the dtime field is overloaded for use with the orphan list. Within this "extra" 32-bit field, the lower two bits used to extend the 32-bit seconds field to be 34 bit wide; the upper 30 bits used to provide nanosecond timestamp accuracy. ETX4 add the support for timestamps created by date as well as higher precision timestamps (64-bit values instead of 32-bit). [Ngu17]

NTFS keeps the track of many time stamps. Each file has its own time stamp to 'Create', 'Access', and 'Modify' the attributes, the lattest references to the time when the entry of MFT itself was modified. The values above abbreviated commonly as the 'MACE' values. We do not exclude the possibility that other attributes in each MFT record can also contain their own timestamps that are of forensic value. For NTFS a *file time* is a 64-bit long in value and represents the number of intervals by 100-nanosecond each that have elapsed using the UTC time. NTFS file system stores the values of time in the UTC format, so that the changes in daylight saving time or time zone does not affected in them. There are timestamps updated in different time and for different reasons. A file timestamp has the guarantee that the time of a file is correctly reflected only when the handle who makes the changes for a specific file is closed. Not all the file systems can record the creation and last access time of a file, and if they do not all of them record it in the same manner. The NTFS file system updates the last access time for a file by 1 hour after the last access. Timestamps for Files and directorys are one of the resources forensic analysis that are used in Windows for determining if something has happened, or in what particular order the sequence of this events took place. [CHLKL07]

3.4 Features

Below we are giving some of ETX4 features:

3.4.1 Large file system

ETX4 filesystem can support volume sizes maximum up to 1 exbibyte (EiB) and files with sizes up to 16 tebibytes (TiB).Volumes larger than 16 tebibytes (TiB) are not recommended.

3.4.2 Extents

Extent is a range of physical blocks that is contiguous. Its main duty is improving the performance of large files and reducing their fragmentation. A single extent can map up to 128 MiB of contiguous space in ETX4 filesystem with a 4 KiB block size.

3.4.3 Delayed Alocation

ETX4 filesystem use a technique for the performance called *"allocate on flush"* also known as *delayed allocation*. That makes ETX4 to delay blocks allocation until it writes data to disk. ETX4 filesystem use the delayed allocation that allows it to buffer the data and allocate the groups of filesystem blocks. Multiblock allocator consequently can make a better choice in allocating files contiguously on disk.

3.4.4 Persistent pre-allocation

ETX4 filesyste can previosly allocate on disk the space for a file. To do this ETX4 on most file systems adds zeros to the file on the creation of it. This way that the allocated space would be guaranteed and contiguous. [DT16] Here we are listing some of NTFS features:

3.4.5 Sparse File Support

There are sparse files distributed with segments that are empty for which and there will be no extra space on disk to use. The applications see this file like an ordinary one but with empty segments seen as regions filled with zero values. It uses a disk addresses of 64-bit and if can support up to 2^{64} bytes disk partitions. File names that are individual in NTFS are limited to 255 characters. The names are case sensitive.

3.4.6 Disk Usage Quotas

This allow the computer administrator to run different versions of Windows who support the NTFS file system for setting a verge of disk space to be used by users. It also allows the administrators to find out the quantity of the space in disk that each user is using.

3.4.7 Reparse Points

There are reparse points used for associating a reparse tag for the user space attribute of a folder or file. When the manager of the object parses a name lookup for a file system and he encounters a attribute that is reparse, it will reparse the lookup for that name by passing the controlled data reparse by users to all driver filters of file system that are loaded into Windows.

3.4.8 Distributed Link Tracking

Distributed link tracking allows the applications to track all the files like OLE links or shell shortcuts, This happens even if the files are renamed or migrated to another volume in the drive in the same machine. The tracking of the links implemented works as a service in system that uses the index identifier object that is stored in a metafile.

3.4.9 File Level Encryption (Encrypting File System)

EFS provides user transparent and strong encryption of all files or folders on an NTFS file system volume. File Level Encryption works by encrypting the file with a symmetric key that used because it takes a small amount of time for large amount of data encrypted and decrypted than an asymmetric key is used. The symmetric key used for encrypting the file, is encrypted itself using a public key before encrypting the file. Then this key linked with the user who made the file encryption. The data that is encrypted is then stored in a stream of data in the file that has been encrypted. [DT16] [Fuj10]

3.5 Speed

The speed of ETX4 file system is showing that it is excelling over other filesystems like NTFS by a considerable amount. Many manufactured flash drives, come with Microsoft FAT32 file system by default, which is fine because of its compatibility.

Most operating systems read and write FAT32 file system. FAT32 has a limit of 2GB for file sizes. Even if FAT32 is more compatible in different operating systems, the benefits of ETX4 make it less usable. As flash drives have been increasing in size, the 2GB file size limit of FAT32 is now a problem. Unfortunately, NTFS support is still popular, because is kept closed by Microsoft.

Even though, the tests made are showing minor differences in speed between different filesystems exist some advantages of the filesystem that is more efficient, like ETX4 besides speed alone. I think getting rid of

filesystems like FAT32 or NTFS, which are a mess and need defragmenting, and olso known for file corruption it will be better. The ETX4 filesystem is a complete journalling filesystem it does not need defragmenting utilities to run on it like FAT32 and NTFS.

With Linux distribution that is reaching a record boot speed with every new release, I have to believe that ETX4 file system need to take some of the credit. In addition, if someone is worried about ETX4 beeing compatible with Windows systems, there are drivers for ETX4 filesystem available for Windows. [RH13] [Ts18]

4. Experimental Phase

4.1 ETX4 Performance Against NTFS

I have tested Windows 7 x64 vs. Ubuntu 10.04 benchmark to see if Microsoft operating system is better than the Linux system distribution. Here I am giving tests on two disk looking at the performance of file system on Windows 7 x64 with NTFS against Ubuntu 10.04 with ETX4 file system. The test happened with the following hardware:

Processor: Intel Core i3 CPU 530 @ 3.32GHz, Motherboard: ECS H55H-M v1.0, Memory: 2024MB, Graphics: Intel Core 256MB, Disk: 320GB Seagate.

For this brief disk testing, i ran IOzone through the Phoronix Test Suite on clean installation of Ubuntu 10.04 linux system and Windows 7 Home Premium x64 system. The first test I made looks at the performance of IOzone on a 4GB write that has different record sizes.

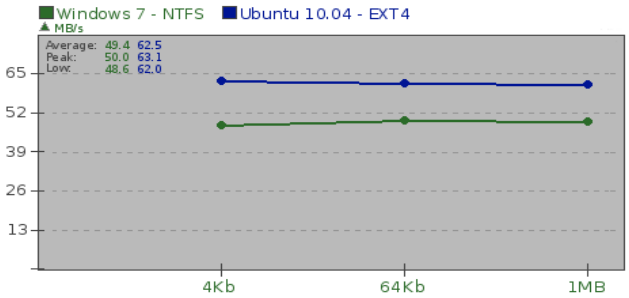

Figure 5: Write Record Size Analysis

On Ubuntu 10.04 with Linux kernel and ETX4 filesystem the test was 26.5% faster than Windows 7 with its NTFS filesystem.

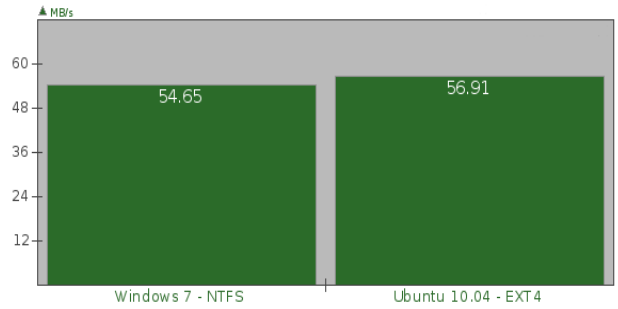

Figure 6: Record Size: 64 KB – File Size: 4 KB – Disk Test: Read Performance

The performance in reading in the two file systems with IOzone was very close.

While ETX4 has regressed a fair amount since it stable in the main Linux kernel, it seems like it is still able to be in advance against Windows 7 and NTFS at least with the synthetic disk benchmark. It will be interesting to see how Apache, SQLite and PostgreSQL performance will be between the two operating systems. Especially if that is where ETX4 had a challenging experience with recent releases of kernel that try to improve the safety of data at the cost of speed. [VW09] [XLLR17]

4.2 NTFS vs ETX4 on external 3.0 USB Hard Drive

Testing this two file systems in a external Hard Drive will help us to better deside which of them is more efficient and with better performance. Therefore, to do this i have tested the speed of data transfer on Windows 7 running the NTFS file system and Ubuntu 10.04 running the ETX4 file system by transferring data to a 1Tb Toshiba 3.0 external Hard Drive. By making this test, i received the following results:

Table 1: Results

Format	NTFS	ETX4	Units
Read Speed	27.8	28	Mb/s
Write Speed	21.7	21.3	Mb/s
Access Time	17.52	17.35	ms

As we see, the differences of this test are small and if I make the test, again the differences would change a little. To decide which of this file system we want to use in an external hard drive we have to take in count that from what system we want to use this hard drive. If we want to use it in a Windows operating system with NTFS file system, it is best to use the NTFS file system for the external hard drive. If we do not want to use it in a Windows operating system then we should consider using the ETX4 file system for the hard drive because it

is stable in these days. To give a summary of what we should take in count for which file system we should use let us consider the following reasons:

- If you don't need access from a Windows operating system then you should use ETX4 file system, because it is a native filesystem and is the only one that supports the linux structure properly
- If you do not have access to a Widows machine, you should use ETX4 becouse some filesystem errors that can occur can be fixed from a Windows operating system using the chdsk command
- ETX4 will give an improved performance over NTFS, particularly when trying to write multiple files in the same time or when creating a large file.
- NTFS filesystem needs to be defragmented from time to time and ETX4 filesystem does not need defragmenting.
- NTFS filesystem has permission issues on files and directories. From the other end, ETX4 does not have these issues.

5. Conclusion

Theoretically a drive with ETX4 file system performs better than a drive with NTFS file system.ETX4 file system allows in file names characters that NTFS file system doesn't allow.
ETX4 support individual files with size up to 16TB, and volume sizes up to 1EB. One of the reasons why ETX4 contributes to a best performance is that ETX4 can handle large extents. An extent is range of physical blocks with data in a drive that are contiguous. This feature allows ETX4 to work better with files that are large and reduce the need of fragmenting the drive.
Other factors include the technique called allocate-on-flush used by ETX4. By delaying, the allocation of blocks of data until they are ready for saving to disk. ETX4 improve the performance and reduces drastically the need for fragmentation compared to file systems that allocate the blocks of data earlier.

Using checksum for drive journaling improves the reliability and performance by avoiding the wait during the process of journaling in disk. When it comes to check the files, ETX4 is much faster then NTFS because of the ogranisation of the data in blocks that are unallocated and are marked as such. These data are skipped during the operations of disk check
ETX4 has some performance advantages over NTFS file system, but you must be careful for the file type ODT that LibreOffice uses. An ODT file type is 500% larger than the equivalent in DOCX file type.

6. Future Works

One of the main reason why I started this work was because of my interest in file systems. Their spreading in nowadays applications and their prespective of growth is very clear. I wanted to study them in a more specific way, expecialy this two that i havementioned in this document. In my future work, i would like to expand my knowledge in file systems to know them more and how they perform to make an operating system work.

References

[MCB07] A. Mathur, M. Cao, S. Bhattacharya. *The new ETX4 filesystem: status and future plans*, June 2007

[KKPLSh02] G. B. Kim, D. J. Kang, C. S. Park, Y. J. Lee, B. J. Shin. - *A Dynamic Bitmap for Huge File System in SANs*, January 2002

[EXT418] ETX4 Forum - *ETX4 Disk Layout*.

[Ngu17] N. Nguyen. - *The Structure of NTFS – Windows NT File System.* 2017

[CHLKL07] K. P. Chow, Frank Y.W. Law, Michael Y.K. Kwan, K.Y. Lai - *The Rules of Time on NTFS File System*, May 2007

[DT16] B. Djordjevic, V. Timcenko - *Features and Performance Analysis.* 2016

[RF06] R. Russon, Y. Fledel - *NTFS Documentation*. December 2006

[RH13] RED HAT Enterprise Linux. - *Comparing file system I/O performance.* April 2013

[VW09] M. Vanninen, J. Z. Wang. - *On Benchmarking Popular File Systems.* Clemson University Study, 2009

[XLLR17] Ch. Xie, X. Li, Z. Li, B. Rath - *Linux File System - ETX4.* 2017

[Ts18] Th. Ts'o - *Speeding up file system checks in ETX4.* 2018

[Fuj10] A. Fujita - *Outline of ETX4 File System & ETX4 Online Defragmentation Foresight*, September 2010