Directory Compaction Techniques for Space Optimizations in ExFAT and FAT File Systems for Embedded Storage Devices

Keshava Munegowda[1], Dr. G T Raju[2] and Veera Manikandan Raju[3]

[1] Advanced Storage Division, EMC Corporation
Bangalore, Karnataka, India

[2] Department of Computer Science and Engineering, R N S Institute of Technology
Bangalore, Karnataka, India

[3] Visual Technology Centre, Texas Instruments
Bangalore, Karnataka, India

Abstract
The Extend File Allocation Table (ExFAT) is the future file system for embedded storage devices. The File Allocation Table (FAT) file system de-facto file system for embedded storage devices such as Multimedia Cards (MMC) / Secure Digital (SD) / Micro SD cards, NOR, NAND flash memories. The MMC and SD card associations classify the ExFAT as the standard file system for storage flash cards of more than 32 Giga Bytes (GB) of size. This paper implements the directory compaction techniques for both FAT and ExFAT file system to improve the availability of the user space in the file systems.

Keywords: *Cluster, Contiguous, Compaction, Directory, ExFAT, FAT, File system, Flash memories, MMC, Micro SD, NOR, NAND, Storage.*

1. Introduction

The FAT [1] is widely used file system in tablet personal computers, mobile phones, digital cameras and other embedded devices for data storage and multi-media applications such as video imaging, audio/video playback and recording. The initial version of FAT file system was FAT12 by Microsoft Corporation, later it was extended as FAT16 and further as FAT32 to support higher storage capacity. The FAT file system was initially developed to use on floppy disks and hard-drives. Since most of the Personal Computer (PC) s implements the FAT file system, this file system has become a default and world-wide compatible storage format for embedded devices. Usually the device with the implementation of FAT is recognized as removable storage media in a PC. The Flash memories are default choice of any embedded device as they are low-priced, smaller size and higher storage capacity. Even though FAT file system does not define flash management techniques such as Wear-Levelling and Bad Block management, the embedded devices implements this file system along with the dedicated flash block management algorithms.

In FAT file system, the file or directory is the linked list of the clusters. A Cluster is a group of blocks or sectors of storage device. The File Allocation Table contains the linked list of clusters of files/directories. The FAT specification limits the maximum supported storage size to 32 GB. But, the maximum size supported by FAT32 implementation is 128 GB. But, today the flash storage cards of more than 32 GB are available in market.

The ExFAT [2] [3] file system is developed, by Microsoft Corporation, as successor of FAT32 file system. This file system is optimally designed to support large size flash storage cards with higher read and write performance. The maximum storage size supported by ExFAT file system in 128 peta bytes (PB).The ExFAT file system is not compatible with FAT File system. But, it uses the clusters concept to store file/directory data and File Allocation Table to store the cluster chain of file/directory. The ExFAT file system Contiguous clusters read algorithm [4] for file read performance improvements. The ExFAT file system optimizes the free cluster search by using “cluster heap”. The cluster heap is group of bits to indicate the status of all clusters of file system. Every bit in the cluster heap specifies the status of the data cluster, thus every byte indicates the status of 8 clusters. The binary value “0” indicates that cluster is free and value 1 indicates that cluster is allocated. The cluster heap is also referred as “Allocation Bitmap”. The cluster heap is similar to block bitmap and inode bit map structures used in Ext2 [5] and Ext3 [6] [7] file systems. Figure 3 shows the logical organization of ExFAT file system and the structure of cluster heap. Usually Cluster Heap exists in the 2nd cluster of ExFAT file system. The ExFAT file system is available in Windows 7 and higher operating systems. The FUSE [8] based ExFAT file system implementation [9] is available for Linux kernel based Ubuntu operating systems.

This paper defines and implements the directory compaction technique for both file systems without breaking the compatibility of the existing FAT and ExFAT

File system implementations in windows and Linux operating systems.

2. Directory compaction in ExFAT and FAT File systems

The continious usage of file system by creating, writing, deleting file and directories , the FAT file system can contain the directories having clusters with all deleted entries of files/directories. This paper implements the removal of such clusters from the clusters list of the directory and marks the clusters status in File allocation table as a free clusters.This process is referred as “Directory Compaction”. This improves the number of data clusters and hence optimizes the user space of the file system. The directory compaction improves the performance of the file/directory open operation by reducing the number of directory entries to be referred while searching for the file/directory name in compacted directory’s clusters list. The directory compaction technique is first identified and implemented for FAT file system [10] and in late 2012, it’s been identified that directory compaction is applicable to ExFAT file system also. The Directory compaction can be performed at the following levels of given directory.

2.1 First cluster in the clusters chain of the directory

If the first cluster of the given directory has the all the deleted entries, then Meta-data i.e., 32 byte directory entry should be updated to point to 2nd cluster of the clusters chain of the directory and the first cluster should be made as free cluster. The figure 1 illustrates the directory containing the deleted entries in the first cluster. The figure 2 shows that modified Directory entry and File allocation table in which first cluster is made as free cluster. Note that, According to FAT file system specification whenever file/directory is deleted the first byte of the 32-byte directory entry i.e., meta-data of the file/directory will be replaced by hexadecimal value 0xE5. Hence any 32 byte directory starting hexadecimal value 0xE5 is referred as a deleted file/directory entry. Note that, In case of ExFAT file system, the directory entry starting with hexadecimal value 0x05 indicates the deleted file/directory entry, the directory entry starting with hexadecimal value 0x40 indicates the deleted stream extension directory entry and the directory entry starting with hexadecimal value 0x41 indicates the deleted file name extension directory entry.
In ExFAT file system, if the contiguous clusters are allocated to directory, then File allocation table contains the status of first cluster of the directory and status of the rest of the clusters of directory are updated in the cluster heap. The “NoFatChain” [2] [3] bit field of generic secondary flags of the stream extension directory entry is set to value “1” to indicate contiguous cluster allocation to a directory. The figure 3 illustrates the directory containing the deleted entries in the first cluster of directory in ExFAT file system. Note that File allocation table does not indicate that clusters 4, 5 and 6 are allocated whereas the cluster heap shows that these clusters are allocated. The figure 4 shows that modified Directory entry, File allocation table and cluster heap in which first cluster is made as free cluster. If the clusters allocation to directory is not contigious, then the status of all allcoated clusters are updatd in the File allocation table. Upon deletion of first cluster of directory, the cluster chain of directory is modifed as show in figure 1.

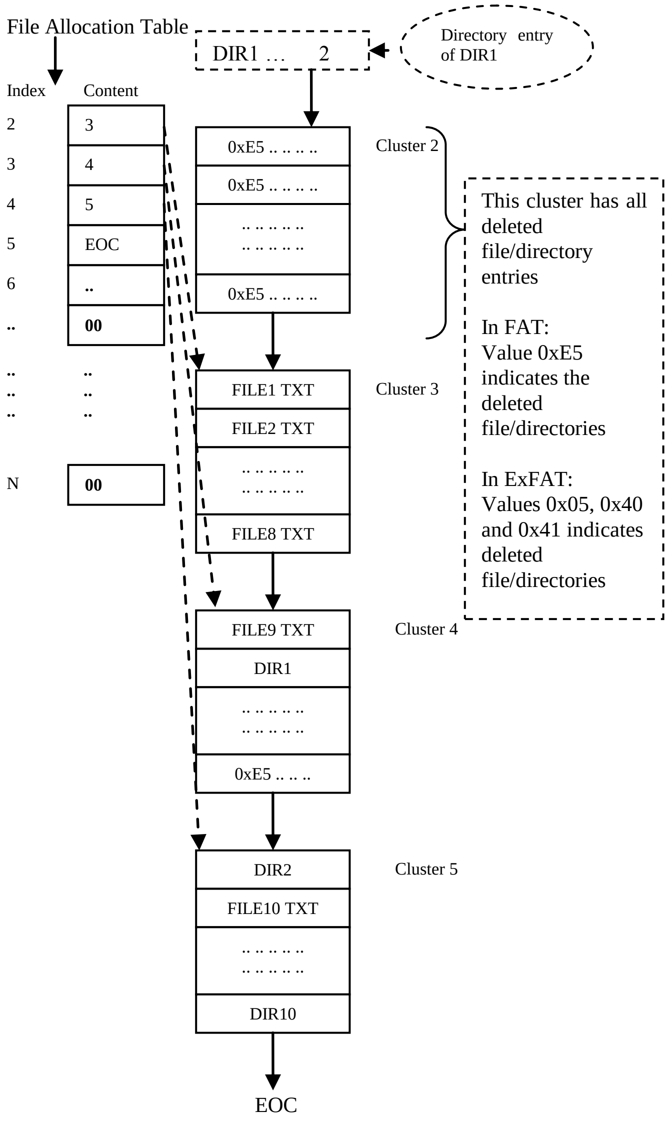

Fig 1: Directory containing deleted entries in the first cluster in FAT file svstem

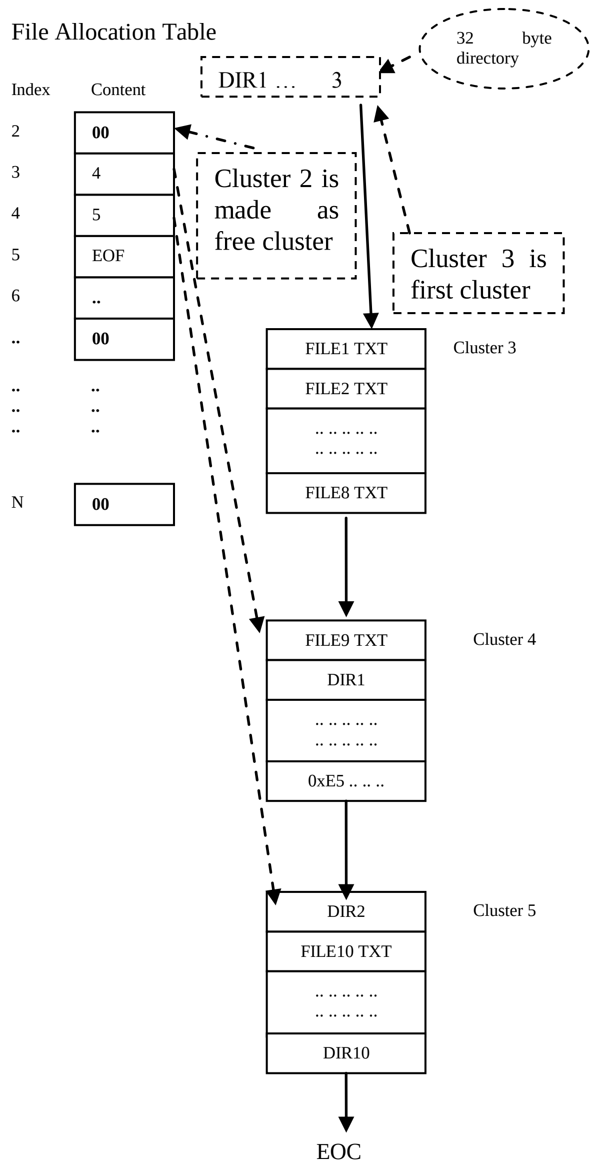

Fig 2: Compacted directory with new first cluster in FAT file system

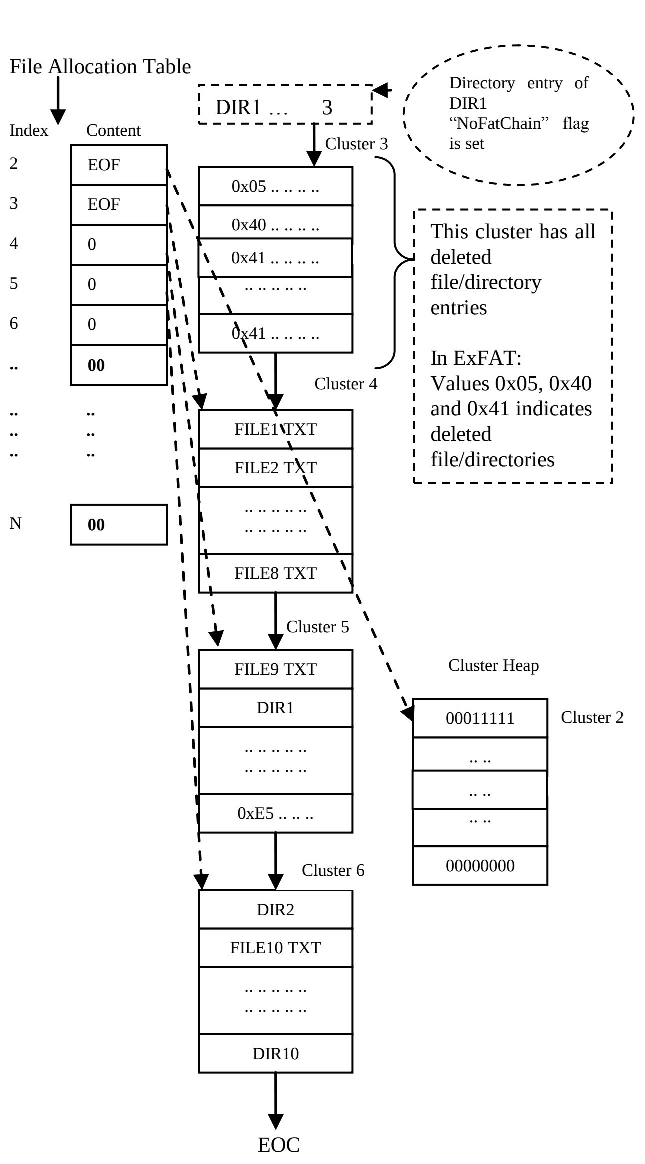

Fig 3: Directory containing deleted entries in the first cluster in ExFAT file system

2.1 An arbitrary cluster in between the first and last cluster of the clusters chain of the directory

If a cluster which is in between the first and last cluster of the cluster chain of the directory has all deleted file/directory entries, then that cluster is will be marked as a free. The index of the previous cluster in the FAT will contain the cluster number available at the index of deleted cluster. In figure 5 it is shown that cluster 3 has the deleted file/directory entries. Figure 6 shows that the directory is compacted by freeing the cluster 3 and cluster 2 is directory directly pointing to cluster 4 in the File Allocation Table.

In case of ExFAT file system, if there are contiguous cluster allocated as shown in figure 7, and if cluster 4 has deleted entries, then removing the cluster 4 from the directory makes the cluster chain as non-contiguous. Hence, After removing the cluster 4, the cluster chain need to updated in the File allocation table and The "NoFatChain" [2] [3] bit field of generic secondary flags of the stream extension directory entry is reset to value "0" to indicate non-contiguous cluster allocation to a directory as shown in figure 8. If the clusters allocation to directory is not contigious, then the status of all allcoated clusters are updatd in the File allocation table. Upon deletion of cluster containing the deleted entris , the cluster chain of directory is modifed as show in figure 8.

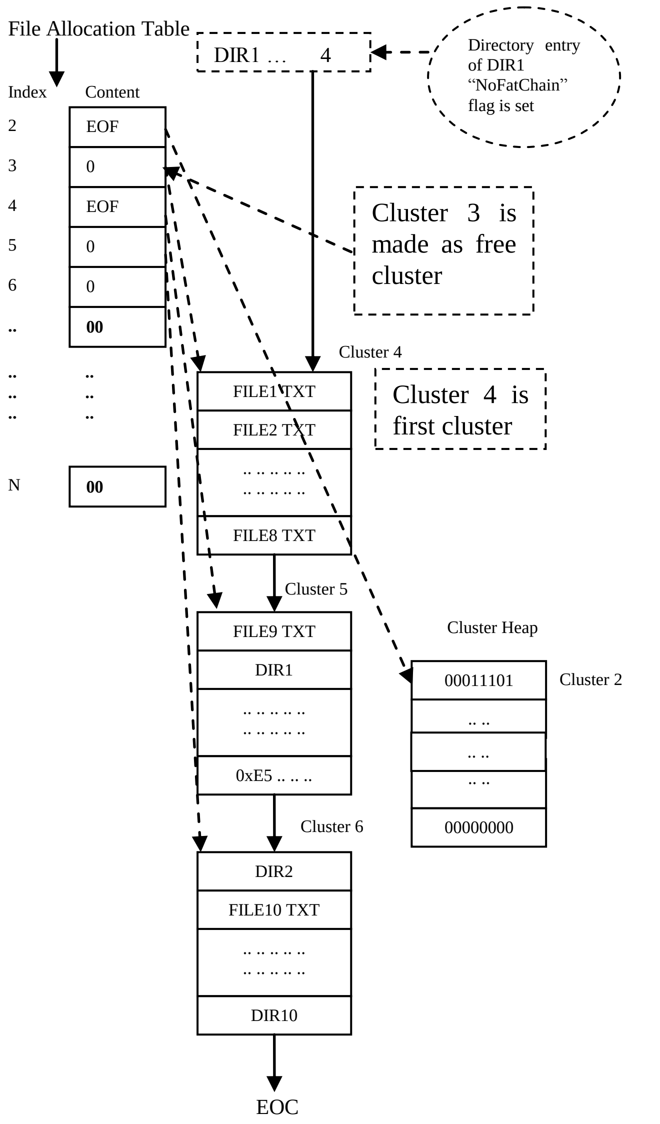

Fig 4: Compacted directory with new first cluster in ExFAT file system

2.3 Last cluster of the clusters chain of the directory

When the last cluster of the directory has deleted entries, then that cluster is will be marked free cluster and the previous cluster to the last cluster will be marked as last cluster. The figure 9 demonstrates that last cluster 5 has all deleted entries. Figure 10 shows that last cluster 5 is made as free cluster and previous cluster 4 is marked as last cluster in the File Allocation Table.

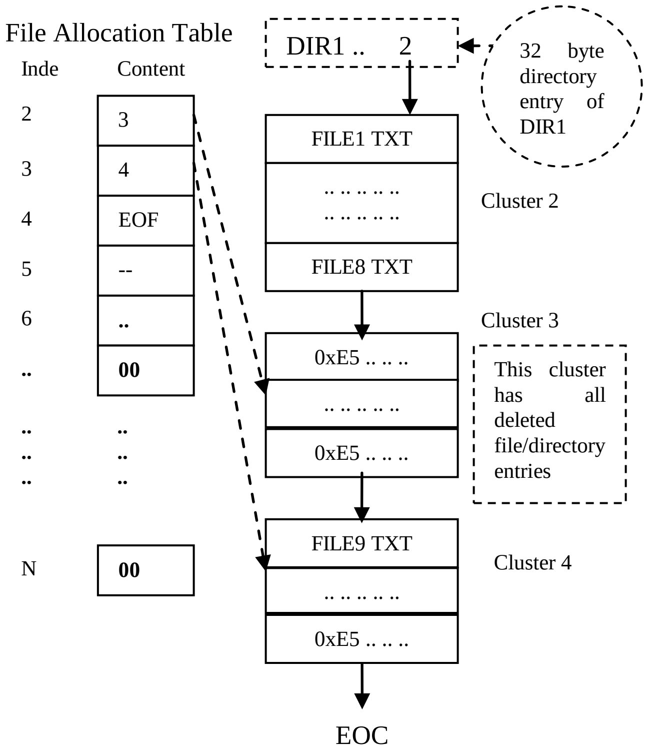

Fig 5: Directory containing deleted file/directory entries in the middle cluster chain in FAT file system

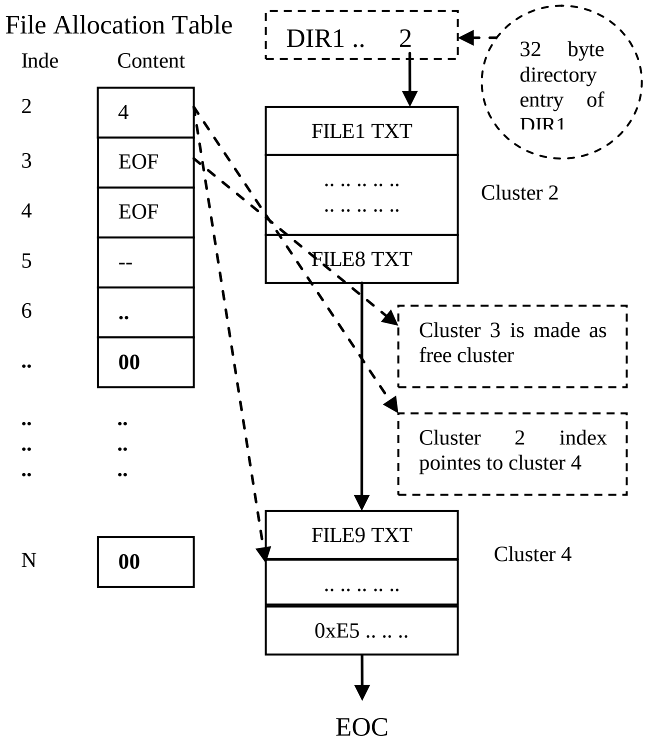

Fig 6: Compacted directory in which cluster 3 is removed from the cluster chain in FAT file system

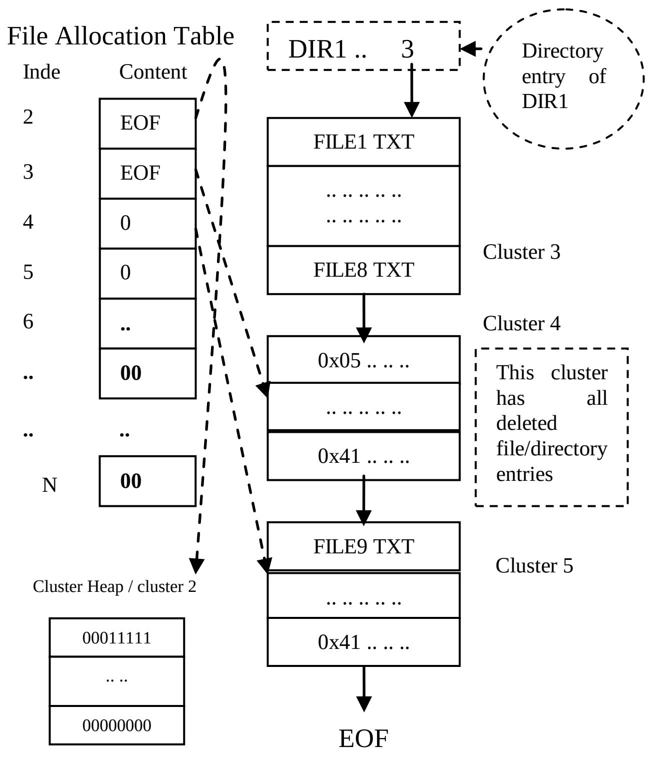

Fig 7: Directory containing deleted file/directory entries in the middle cluster of cluster chain of ExFAT file system

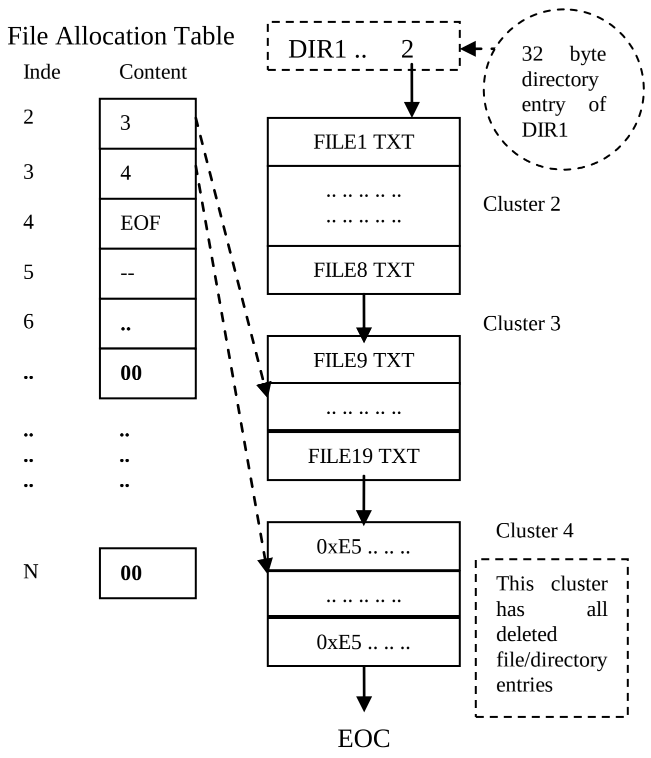

Fig 9: Directory containing deleted file/directory entries in the last cluster of cluster chain in FAT file system

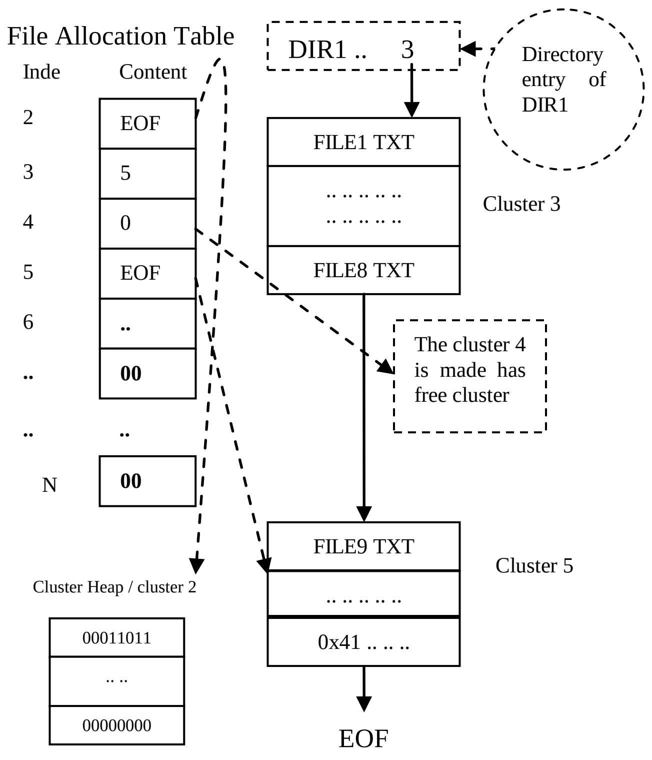

Fig 8: Compacted directory in which cluster 4 is removed from the cluster chain in ExFAT file system

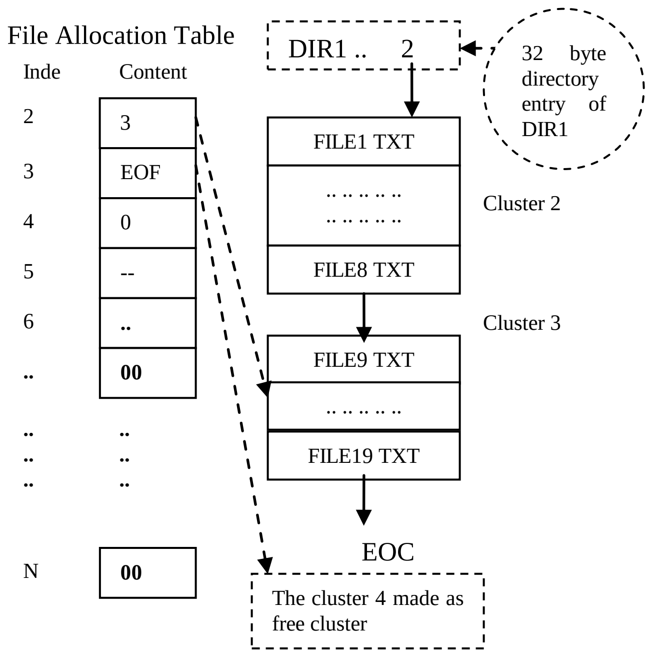

Fig 10: Compacted directory in which last cluster is made as free in FAT file system

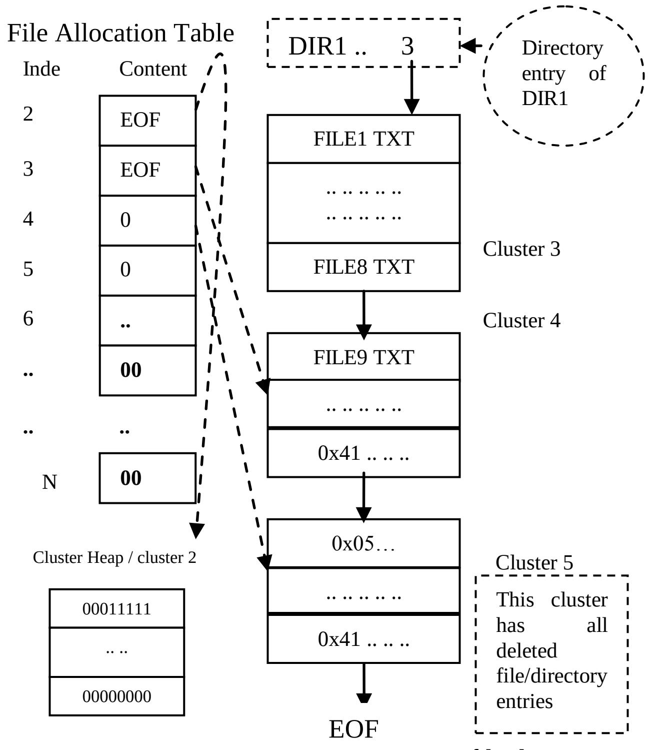

Fig 11: Directory containing deleted file/directory entries in the last cluster of cluster chain of ExFAT file system

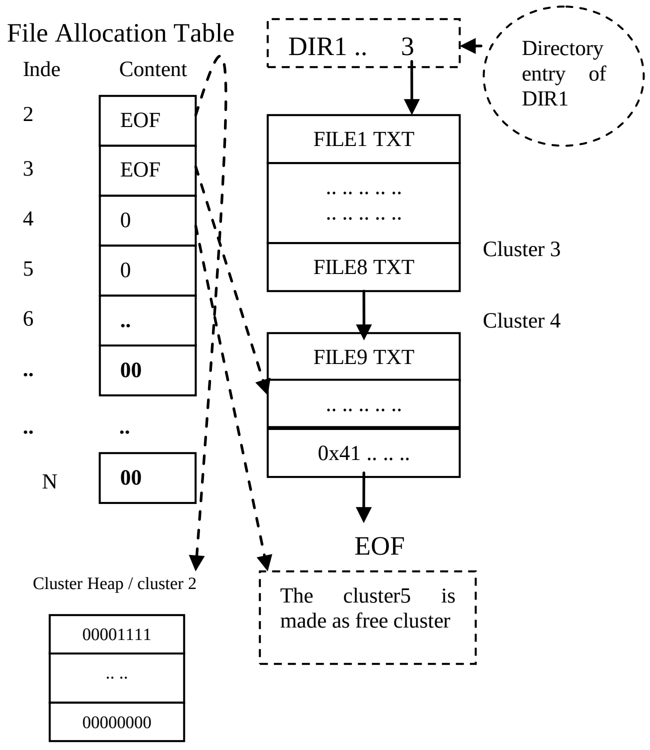

Fig 12: Compacted directory in which last cluster is made as free in ExFAT file system

In case of ExFAT file system, upon removing the last cluster of cluster chain of directory, only the cluster heap is updated the status as a free cluster. In figure 11, it shown that last cluster 5 has all deleted entries and upon removing this cluster from cluster chain the cluster heap is updated the status and File allocation table is not modified. If there are non-contiguous clusters allocation to directory and last cluster is removed then the cluster chain in updated in file allocation table.

4. Conclusions

The Directory Compaction truncates the directory by removing the clusters, having deleted entries, from the directory clusters chain. The directory compaction generates the free clusters and hence increases the availability of the storage space for user data. The directory compaction improves the performance of the file/directory open operation by avoiding the comparison of deleted directory entries with input file/directory name in compacted directory's clusters list.

Acknowledgments

It is a pleasure to acknowledge Mr. Robert Shullich, Enterprise Security Architect, Tower Group Companies, USA, for resolving technical queries to understand the ExFAT file system. Mr. Andrew Nayenko deserves special thanks for open souring the FUSE based ExFAT file system implementation to Ubuntu and other Linux operating system distributions.

References

[1] FAT32 File System Specification, FAT: General Overview of On-Disk Format, Microsoft, 2000.
[2] Ravishankar V.Puipeddi, Vishal V.Ghotge, Ravinder S.Thind, "Quick file name look up using name hash", USPTO Patent: 8321439, granted on November 27, 2012.
[3] Keshava Munegowda, Venkatraman S, Dr. G T Raju, "The Extend FAT file system: Differentiating with FAT32 file system", Linux Conference, Prague, Czech Republic, Europe, October 2011.
[4] Ravishankar V.Puipeddi, Vishal V.Ghotge, Ravinder S.Thind, "Contiguous File Allocation in Extensible File system", USPTO Patent: 8,606,830, granted on November 20, 2013.
[5] Remy Card, Theodore Ts'o, Stephen Tweedie, "Design and implementation of the Second Extended File system", First Dutch International Symposium on Linux
[6] Stephen C. Tweedie (May 1998). "Journaling the Linux ext2fs File System, Proceedings of the 4th Annual LinuxExpo, Durham, NC. Retrieved 2007-06-23

[7] Dr. Stephen C. Tweedie , "Ext3 , journaling file system", Ottawa Linux Symposium, Ottawa Congress Centre, Ottawa, Ontario, Canada on the 20th of July, 2000

[8] FUSE (File System in User Space) : http://fuse.sourceforge.net/

[9] Fuse based ExFAT implementation for Linux : http://code.google.com/p/exfat/

[10] Keshava Munegowda, Veera Manikandan Raju, Rohit Joshi, "Storage optimizations by directory compaction in a fat file system", USPTO patent: 8429331, granted on April 23, 2013.

Keshava Munegowda is working as Principal software engineer in EMC Corporation, Bangalore, India. He received Bachelor's Degree in Computer science and Engineering in 2001 from Bangalore University, India, and M-Tech degree in computer Science and Engineering in 2004 from Visvesvaraya Technological University (VTU), Belgaum, Karnataka, and currently pursing PhD in field of computer science in VTU. He is working in the field of storage drivers, file systems, USB and Network Protocols since 10 years. He has obtained 2 USPTO patents in the area of FAT file system. He has published and presented papers in journals, international level Conferences and published a book in field of WLAN secuirty. He is one of the open source contributors for USB host device driver of OMAP SOC in Linux kernel.

Dr. G T Raju is a professor and Head of Computer Science and Engineering Department of R N S Institute of Technolgy , Bangalore, India. He received his Bachelor's Degree in Computer Science and Engineering from Kalpataru Institute Of technology, Tiptur, Karnataka, in 1992 and M. E in Computer Science and Engineering from B.M.S College of Engineering, Bangalore, India in 1995 and Doctorate of Philosophy Ph.D.in 2008 in Computer Science and Engineering from Visveswaraya Technological University, Belgaum, Karnataka; He has 18 years of Experience in teaching and research in Data Mining, Data Warehousing, Image Processing, Databases, Artificial Intelligence and Computer Graphics. He has published and presented papers in journals, international and national level Conferences and published a text book**.**

Veera Manikandan Raju is a Senior Member Technical Staff in Texas Instruments, Bangalore, India. He received Bachelor's Degree in Electronics and Communication Engineering from National Institute of Technology, Trichy, Tamilnadu, India, in 1996**.** He as 17 years of experience in field of storage drivers, file systems, USB, Network protocols, Video Processing and Gesture Recognition Technologies. He has obtained 2 USPTO patents in the area of FAT file system and he has published several journal and conference papers.