

BLFS: Supporting Fast Editing/Writing for Large-Sized Multimedia Files

Seung Wan Jung¹, Seok Young Ko², Young Jin Nam³, Dae-Wha Seo¹,

¹ Kyungpook National University, Mobile Computing and Embedded System Lab.,
Daegu, South Korea
{tmdrod, dwseo}@ee.knu.ac.kr

² Samsung Electronics, memory business, Hwasung, South Korea
Seokyoung.ko@samsung.com

³ Daegu University, Computer Architecture Lab., Gyeongsan, Gyeongbuk, South Korea
yjinam@daegu.ac.kr

Abstract. We propose BLFS (Block Link File System) supporting fast editing/writing for large-sized multimedia files. In the BLFS, when large-scale multimedia files are stored and edited, there is no data I/O but only modification of the metadata. Additionally, by sharing data blocks between multimedia files, we can save disk space. The BLFS was implemented in a Linux environment by modifying Ext3, and validated by applying the Linux multimedia file-editing tool 'Avidemux'.

Keywords: Ext3 file systems, multimedia editing, data sharing

1 Introduction

Multimedia devices, such as smart TV, and digital camera record, store, and play multimedia files using a PVR (Personal Video Recorder) [1] function. The PVR supports the editing function, which is selected when the user wants to edit and store a section of video. However, current multimedia file editing methods have the serious problems for the required time and storage space. For multimedia file editing, the Ext3 [2] initially reads all the edited data into main memory sequentially and writes the data into the disk. This method is suitable for small files in terms of algorithm simplicity, but it requires too much time and disk bandwidth for large files. In established research FWAE [3], we cared and improved these problems by sharing data using metadata, not copying editing section into a new disk space. However, an outbreak of many system overheads during the creation of metadata, in order to manage additional sharing information, brought reduction of improvement rate.

Hence, in this paper, we propose a BLFS (Block Link File System) based on Ext3 to provide high speed multimedia editing and storing technique, which is improved technique of FWAE. The BLFS not only reduces system load completely on sharing information management but also is implemented on Linux Ext3 to solve the optimization problems. The BLFS was implemented in a Linux environment, and validated by applying the Linux multimedia file-editing tool 'Avidemux' [4].

2 Block Link File System

Fig. 1 shows the overall architecture of BLFS. The 'BL file Write Engine' is added to the existing Ext3 file system to create a BLFS. The 'BL file Write Engine' supports the high-speed multimedia editing function. It has a 'BL file Create' module, an 'SI manager' module, and a 'modified file deleting' module for deletion of files that share data blocks. The BL file Create module executes BL file creation while the SI manager module manages shared data block information.

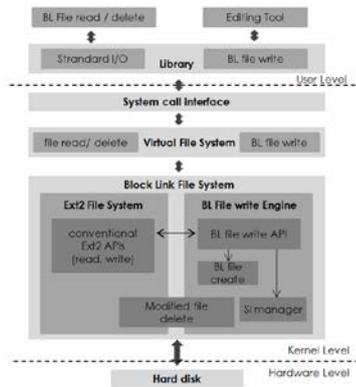


Fig. 1. The architecture of the BLFS

The new BL file format, which is used for high-speed editing of large-scale multimedia files in the BLFS, will be described. While large-scale multimedia files are being edited, a BL file is created. The BL file shares data blocks with the target file, and has only the address information of the data blocks.

The BL file, as shown in Fig. 2, is composed of metadata, such as inode, indirect block and shared information (SI). The BL file does not have actual data blocks, but only stores the address information of the target file and shares data blocks with inode and indirect blocks. The SI, a new metadata item, shows the state of the original file's data block and is created at the beginning when the original file shares data blocks with another file. The SI is composed of the inode number of the original file, the total number of edited sections, and information about the edited section.

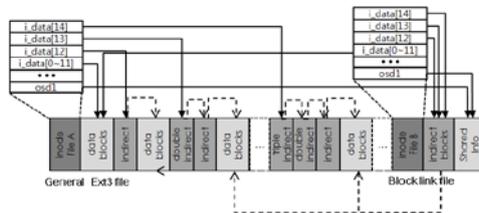


Fig. 2. The structure of a BL file

3 Performance Evaluation and Conclusion

Fig. 3 shows the results of comparison of the required times with various editing operations for the proposed file system, FWAE the Ext3. Note that Ext3 reads edited contents into memory, and then rewrites them in disk. The file editing technique of FWAE is similar to that of BLFS, but it has a different method of shared information from that. The BLFS removes the deleting segments of the original file by using modified file delete module. The bigger the edited file size, the longer the time is required to store edited files for the BLFS, FWAE and Ext3. The experiments for comparison of the required time are performed for various cases.

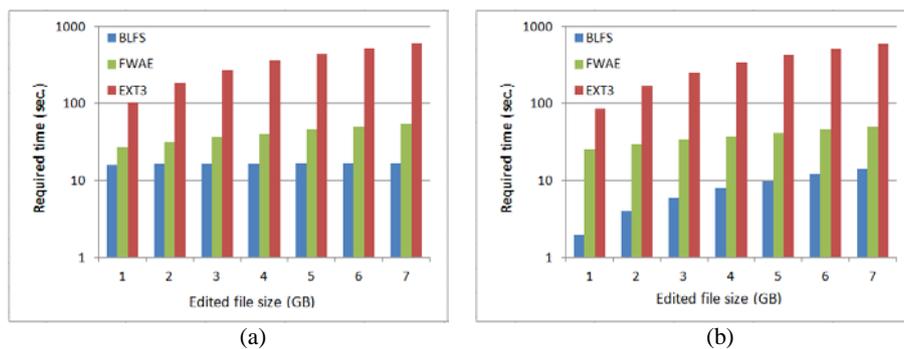


Fig. 3. Comparison of the time required for editing and storing a multimedia files: (a) Original file size is fixed 8GB, overwrite mode (*BLFW(over)*) (b) Original file size is fixed 8GB, new file mode (*BLFW(new)*)

References

1. Lim, S.Y., Choi, J.H., Seok, J.M., Lee, H.K.: Advanced PVR architecture with segment-based time-shift. In: International Conference on Consumer Electronics, 1--2, Las Vegas (2007)
2. Bovet, D., Cesati, M.: Understanding the LINUX KERNEL, 3rd Edition. pp. 738—774. O'REILLY (2007)
3. Jung, S.W., Nam, Y.J, Seo, D.W.: Amortized Linux Ext3 File System with Fast Writing after Editing for WinXP-Based Multimedia Application. In: IEICE transactions on Information and Systems, vol. E94-D, no. 11, pp. 2259--2270, IEICE (2011)
4. Avidemux, <http://avidemux.sourceforge.net/>

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0004478).