

A RELIABLE VIDEO STORAGE ARCHITECTURE IN HYBRID SLC/MLC NAND FLASH

Yimei Kang¹ Xingyu Zhang¹ Zili Shao² Renhai Chen^{3*} Yi Wang⁴

¹ Beihang University, 37 Xueyuan Road, Haidian District, Beijing, China

² The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

³ Tianjin Key Laboratory of Cognitive and Application, School of Computer Science and Technology, Tianjin University, 135 Yaguan Road, Jinan District, Tianjin, China

⁴ Shenzhen University, 3688 Nanhai Road, Nanshan District, Shenzhen, Guangdong, China

ABSTRACT

In this paper, we propose a reliable video storage architecture in hybrid SLC/MLC storage systems. In this architecture, the video stream is reconstructed as the key cluster and the non-key cluster according to the importance of video restoration. The key cluster is stored in SLC blocks to ensure the reliability, while the non-key cluster is stored in MLC blocks to utilize its capacity. The experimental results show that the proposed scheme can significantly improve the storage reliability of the video files and prolong the lifetime of the storage system 18x.

Index Terms— Reliable storage, storage architecture, hybrid NAND flash, key cluster, non-key cluster

1. INTRODUCTION

Video-based applications have become a vital component in many embedded systems. Examples include mobile phones, unmanned aerial vehicles and car driving recorder systems. In these systems, a large amount of video data is stored in the storage devices, which puts forward high requirements for the storage devices, such as large capacity, high reliability, fast data access, low power consumption, small package size, and high shock resistance. NAND flash memory storage device (e.g., SD card, CF card) can well meet the above requirements and thus has been widely used to serve video stores.

There are mainly two types of NAND flash memories: single-level-cell (SLC) and multi-level-cell (MLC). MLC is used more than SLC due to its higher capacity and lower price. However, the increasing density of MLC leads to the high bit error rate [1] and the short life time, which could not support high quality video stores and may cause severe reliability issues.

Hybrid SLC/MLC flash memory system [2, 3, 4, 5] is a way to overcome the limitations of MLC flash memory.

In this hybrid storage systems, the flash memory blocks can be divided into SLC and MLC regions by programming for a flexible way of utilizing the advantages of both SLC and MLC. Prior arts have demonstrated the significant reliability enhancement to MLC NAND flash memory while still leaving a huge design space to explore.

Both SLC and MLC have the characteristic of out-of-place update, i.e., a write operation can be implemented in page unit, but an erase operation can only be implemented in block unit which typically includes dozens of pages [6]. For this reason, the FTL (Flash Translation Layer) is introduced, which handle the translation of logical address to physical address and manage the bad blocks. Usually, FTL can be implemented in three different ways: page-level FTL [7, 8, 9, 10], block-level FTL [11, 12] and hybrid FTL [13, 14]. When a page is written for the first time, it is put in the data block where each page is in its own place, in other words, the offset of the physical location is identical to that of the logical address [4].

Generally, video files have large sizes which require high storage capacity. For example, the size of a high-definition movie perhaps over 1 GB. However, only reference data for most video coding standard require high reliable storage. I-VOP (Video Object Plane) is the key reference data for MPEG, H.263 and H.264 stream. The decoded image will be damaged seriously if only one bit of I-VOP is incorrect, while the image is weakly influenced when more bits of other VOPs are incorrect. Therefore, we present an effective video storage architecture which build a new layer in FTL for reliable video storage in hybrid SLC/MLC NAND flash storage system to jointly utilize the reliability advantage of SLC flash for I-VOPs and the capacity advantage of MLC flash for other VOPs.

The rest of this paper is organized as follows: Section 2 introduces the background and the motivation of this paper. Section 3 presents the proposed architecture and techniques. Section 4 presents the experimental results with discussion. Finally, the conclusion is given in Section 5.

This work is supported by the programs of the National Natural Science Foundation of China (61702357).

*The corresponding author: Renhai Chen

2. BACKGROUND AND MOTIVATION

2.1. Motivational Example

To test the error tolerance of key VOP and non-key VOP in MPEG, H.263 and H.264 video files, we present an example with analysis. According to the bit-stream structure, the data errors could happen in I-VOP, B-VOP or P-VOP. Fig. 1 (a) shows the 8 pictures decoded from a GOP (Group of Picture) in a bit-stream without any data error. The coding order of this GOP is IPBBPBPB". In order to analyze the roles of one particular VOP to a bit-stream, 5 cases with different errors of I-VOP, B-VOP and P-VOP are considered, respectively. All pictures in Fig. 1 are decoded from the same GOP in Fig. 1 (a).



(a) The 8 pictures decoded from a GOP without any data errors.



(b) The 8 pictures decoded from the same GOP as in (a) with 1-bit error in the I-VOP.



(c) The 8 pictures decoded from the same GOP as in (a) with 4 continuous bytes data error in the first P-VOP.



(d) The 8 pictures decoded from the same GOP as in (a) with 4 random bytes data error in the first P-VOP.



(e) The 8 pictures decoded from the same GOP as in (a) with 8 continuous bytes data error in the first B-VOP.



(f) The 8 pictures decoded from the same GOP as in (a) with 8 random bytes data error in the first B-VOP.

Fig. 1. Error tolerance comparison of I-VOP, B-VOP and P-VOP

Fig. 1 shows that only 1-bit error of I-VOP will make the decoded picture lose nearly all useful information, while several bytes of errors for B-VOP and P-VOP will not seriously influence the picture quality. Therefore, in order to maintain the high-quality of an MPEG file, I-VOP should be carefully stored.

2.2. Reliability of MLC and SLC Flash Memory

Error characterization of MLC and SLC flash is given on the block, page, and bit level. The uncorrectable bit error rate (UBER) is used to address NAND flash bit error rate after ECCs (error correction codes). Both SLC and MLC flash in practical use are protected with ECC.

Suppose $UBER_{SLC}(k)$ and $UBER_{MLC}(k)$ denote the storage error rates of a bit in SLC flash memory and MLC flash memory after its k -th program/erase (P/E) operation, respectively. Suppose an I-VOP of a video stream will occupy L bits when it is stored in flash memory. Then, the bit error rate $I-BER_{SLC}(k)$ of an I-VOP stored in SLC flash memory and the bit error rate $I-BER_{MLC}(k)$ of the I-VOP stored in MLC flash memory satisfy the following equations, respectively.

$$I-BER_{SLC}(k) = \sum_{i=1}^L \binom{L}{i} UBER_{SLC}(k)^i (1 - UBER_{SLC}(k))^{L-i} \quad (1)$$

$$I-BER_{MLC}(k) = \sum_{i=1}^L \binom{L}{i} UBER_{MLC}(k)^i (1 - UBER_{MLC}(k))^{L-i} \quad (2)$$

Suppose E_s is the endurance of SLC flash memory, and E_m is the endurance of MLC flash memory. The storage error rates are inversely proportional to the number of P/E cycles both for SLC and MLC flash memory. Hence, the storage error rates of a bit in SLC flash memory and MLC flash memory approximately satisfy the following equations, respectively.

$$UBER_{SLC}(k) = 1/(E_s - k) \quad (3)$$

$$UBER_{MLC}(k) = 1/(E_m - k) \quad (4)$$

Since $E_s \gg E_m$, $UBER_{SLC}(k)$ is very small while $UBER_{MLC}(k) \approx \infty$ when $k = E_m$. Therefore, for the same k , $I-BER_{SLC}(k) \ll I-BER_{MLC}(k)$. That means a much better picture quality will be guaranteed if I-VOPs are stored in SLC.

In hybrid SLC/MLC storage systems, SLC is more reliable, while MLC has the advantage of capacity. If I-VOP is stored in SLC NAND flash and other VOPs are stored in MLC NAND flash, it will take advantage of both SLC and MLC.

3. RELIABLE VIDEO STORAGE ARCHITECTURE

To keep the storage technological details of NAND flash transparent to applications, we design a layer in FTL to segment and merge the video bit-stream. The layer is called video bit-stream analyzer which includes the segmentation pipeline and merging pipeline as illustrated in Fig. 2.

When writing an MPEG, H.263 or H.264 video file to NAND flash, the video bit-stream analyzer reorganizes the video stream as a key cluster, including the headers and key VOPs, and a non-key cluster, including non-key VOPs. Then the proposed architecture applies cluster mapping algorithm to store key cluster to SLC blocks and to store non-key cluster to MLC blocks. When reading a video file from NAND flash, the video bit-stream analyzer will extract key frames and non-key frames from the key cluster and the non-key cluster, respectively. Then it merges the key VOPs and non-key VOPs to original video stream.

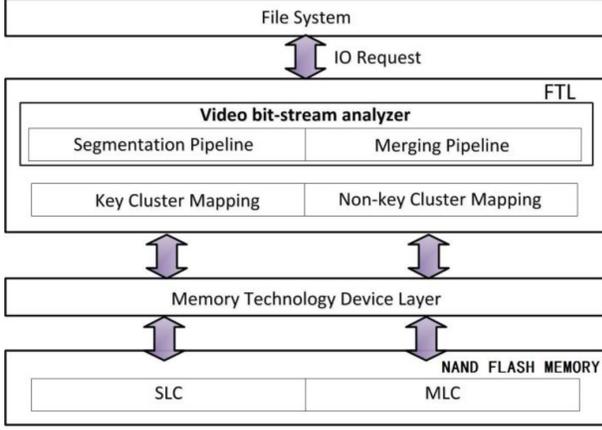


Fig. 2. The structure of proposed architecture

3.1. Video Data Clustering

The I/O requests for video stream are issued to flash sequentially. Therefore, the clusters have to be built sequentially with the stream input to improve the performance. Based on the properties of NAND flash memory, the data will be stored in pages. We build the data cluster with the granularity of a page. Generally, both a key frame and a non-key frame need to be spliced to fill the pages. If the stream cannot fill the last page, we fill '0' to the storage unit. The frame number will be recorded to compute the number of '0's to be added.

Compared to the size of VOPs in a GOP, the size of a header is very small and can be negligible. Let S_{key} be the size of a key frame, $S_{non-key}$ be the size of a non-key frame, and the number of key frames in the key cluster and the number of non-key frames in the non-key cluster both equal to n . Since a page is the basic unit for reading and writing, we will add some '0's to fill the last page. The number of '0's, Z_{key} , should be added to key cluster and the number of '0's, $Z_{non-key}$, should be added to non-key cluster to satisfy Eq. (5) and Eq. (6), respectively.

$$Z_{key} = \text{mod}(n \cdot S_{key}, 8) \quad (5)$$

$$Z_{non-key} = \text{mod}(n \cdot S_{non-key}, 8) \quad (6)$$

I-VOP includes the information of the whole picture, and the B-VOP and P-VOP are referred to as inter coded frames which are reconstructed by I-VOP. Therefore, the size of I-VOP would be larger than that of B-VOPs and P-VOPs.

3.2. Segmenting and Merging Algorithms

Since the video files are downloaded as data streams for mobile devices, we apply a pipeline architecture to segment and merge the data in a video stream. We design a segmentation pipeline and a merging pipeline for data writing to and reading from the flash memory, respectively. The segmentation pipeline gets the video stream from the file system. In the segmentation pipeline, data is parsed into GOPs by the parser.

Each GOP will be segmented into a header, a key frame and non-key frames by the segmenter. The merging pipeline gets the key cluster from SLC blocks and non-key cluster from MLC blocks. In merging pipeline, the analyzer will disaggregate the key cluster into headers and key frames, and disaggregate the non-key cluster into non-key frames. The VOPs in an MPEG video file begin with start codes. The start code of a VOP is "0x000001B6". Following the start code, there are two bits to distinguish if the following information is I-VOP, B-VOP, P-VOP or S-VOP, where "00" is for I-VOP, "01" is for P-VOP, "10" is for B-VOP and "11" is for S-VOP.

In the proposed architecture, the video bit-stream analyzer will segment or merge the video bit-stream according to the structure of MPEG bit-stream. There are two mapping operations and two mapping tables correspondingly. One is key cluster or SLC mapping, and the other is non-key cluster or MLC mapping. The parallel procedure of video stream segmenting and writing is presented in Fig. 3.

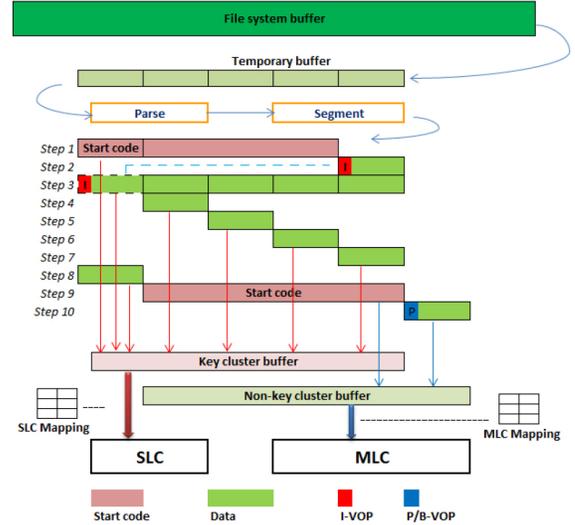


Fig. 3. The parallel procedure of segmenting and writing pipeline

4. EVALUATION

In the experiments, CDFTL [15] was used as the baseline FTL in the experiments. In real applications, the flash chips need to apply ECC to improve their reliability. SLC and MLC used the same ECC controller. The number of bits was 2112 bytes/page, including 2048 bytes video data and 64 bytes mega data (ECC code). The maximum number of error bits per code-word that the ECC code can correct was 4 bits.

4.1. The bit error rate of I-VOP

In this set of experiments, I-BER, the bit error rate of I-VOP, is used as a performance metric to evaluate the reliability of the video stream storage. We compared video storage reliability in hybrid SLC/MLC storage system by using the proposed

architecture and in the single MLC storage system by using CDFTL. We implemented the proposed architecture in an ARM/Linux embedded environment and the hardware platform was based on Tiny6410 platform, Samsung S3C6410 CPU, 256M DDR RAM. The flash chips included B-MLC32-1, D-MLC32-1, E-SLC8, B-SLC4 [16]. And the size of the I-VOP used in the simulation is 335,792 bits.

As shown in Fig. 4, I-BER is increasing with the growth of P/E cycles for both SLC and MLC blocks. $I-BER_{SLC}(k)$ is much less than $I-BER_{MLC}(k)$. $I-BER_{SLC}(k)$ is zero before 80,000 P/E cycles for B-SLC4, and for E-SLC8 it is zero before 100,000 P/E cycles. On the other hand, $I-BER_{MLC}(k)$ is zero before the block is programmed or erased 1000 cycles when the I-VOP is stored in D-MLC32.

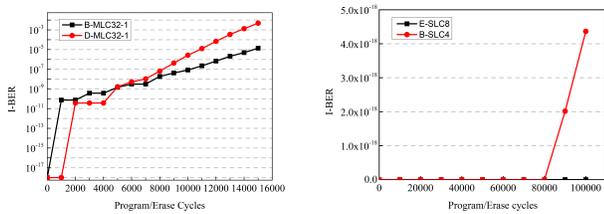


Fig. 4. The structure of proposed architecture

The decoded images are damaged seriously if there is only one bit error in I-VOP. Therefore, with the increasing of I-BER, video playing could not perform smoothly. By using the proposed architecture in hybrid SLC/MLC storage systems for MPEG files, the key cluster is stored in SLC blocks, and MLC blocks only need to store the non-key cluster. As a result, $I-BER_{SLC/MLC}(k)$ will be the same as that of $I-BER_{SLC}(k)$.

4.2. Lifetime of hybrid SLC/MLC flash for MPEG files

In the second set of experiments, we used movie video files to test the storage performance with an SLC and an MLC from vendor A in [17]. The size of the videos was about 2 GB. In the embedded system of the experiments, the size ranges of I-VOP, B-VOP and P-VOP were [500 MB, 1400 MB], [50 MB, 600 MB] and [100 MB, 800 MB], respectively.

Fig. 5 shows the lifetime comparison between hybrid SLC/MLC storage system and single MLC storage system for MPEG video files. The $I-BER(k)$ of high quality MPEG video files is expected to be zero. As shown in Fig. 5, by using the proposed architecture, the lifetime of hybrid SLC/MLC flash memory system for high quality video file storage approaches 3,600,000 P/E cycles, while the lifetime of single MLC storage system for high quality video files is about 200,000 P/E cycles. Therefore, the proposed architecture can significantly prolong the lifetime of video storage by 18x.

We played the same video in the hybrid SLC/MLC storage system with/without adopting our scheme in three different cases: (1) the video was the first file stored in the virgin flash;

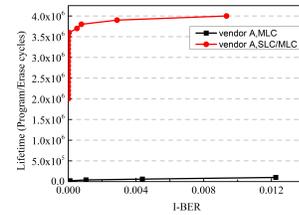


Fig. 5. The parallel procedure of segmenting and writing pipeline

(2) the video played in this flash after 200,000 P/E cycles (the lifetime of single MLC flash memory); (3) the video played in this flash after 3,600,000 P/E cycles (the lifetime of hybrid SLC/MLC storage systems). As shown in Fig. 6, the video information is lost without adopting our scheme, while the video still well played with the proposed architecture. This demonstrates that the proposed architecture greatly extends the MPEG video file lifetime in the hybrid SLC/MLC flash memory.



(a) the video was the first file stored in the virgin flash.



(b) Video played after 200,000 P/E cycles (the lifetime of MLC flash memory) with adopting our scheme



(c) Video played after 200,000 P/E cycles without adopting our scheme



(d) Video played after 3,600,000 P/E cycles (the lifetime of hybrid SLC/MLC storage systems) with adopting our scheme.



(e) Video played after 3,600,000 P/E cycles without adopting our scheme

Fig. 6. Video played in hybrid SLC/MLC storage system in 3 different cases

5. CONCLUSION

This paper presents a reliable video storage architecture which is designed based on a segmentation pipeline and a merging pipeline to capture the characteristics of MPEG, H.263 and H.264 video files. Experimental results show that the proposed architecture can significantly reduce the error rate and prolong the lifetime of video storage by 18x.

6. REFERENCES

- [1] C. Yang, D. Muckatira, A. Kulkarni, and C. Chakrabarti, "Data storage time sensitive ecc schemes for mlc nand flash memories," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 2513–2517.
- [2] H. G. Lee, S. Baek, J. Kim, and C. Nicopoulos, "A compression-based hybrid mlc/slc management technique for phase-change memory systems," in *2012 IEEE Computer Society Annual Symposium on VLSI*, 2012, pp. 386–391.
- [3] X. Jimenez, D. Novo, and P. Ienne, "Phoenix: Reviving MLC blocks as SLC to extend nand flash devices lifetime," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2013, pp. 226–229.
- [4] L. Yao, D. Liu, K. Zhong, L. Long, and Z. Shao, "TLC-FTL: Workload-aware flash translation layer for tlc/slc dual-mode flash memory in embedded systems," in *IEEE 17th International Conference on High Performance Computing and Communications, IEEE 7th International Symposium on Cyberspace Safety and Security, and IEEE 12th International Conference on Embedded Software and Systems (HPCC, CSS, ICESS)*, 2015, pp. 831–834.
- [5] Linbo Long, Edwin H.-M. Sha, Duo Liu, Liang Liang, Kan Zhong, and Xiao Zhu, "A compiler assisted wear leveling for morphable pcm in embedded systems," *Journal of Systems Architecture*, vol. 71, no. Supplement C, pp. 32 – 43, 2016.
- [6] Jeong-Uk Kang, Heeseung Jo, Jin-Soo Kim, and Joonwon Lee, "A superblock-based flash translation layer for nand flash memory," in *Proceedings of the 6th ACM & IEEE International Conference on Embedded Software*, New York, NY, USA, 2006, EMSOFT '06, pp. 161–170, ACM.
- [7] Xavier Jimenez, David Novo, and Paolo Ienne, "Wear unleveling: Improving nand flash lifetime by balancing page endurance," in *Proceedings of the 12th USENIX Conference on File and Storage Technologies*, Berkeley, CA, USA, 2014, FAST'14, pp. 47–59, USENIX Association.
- [8] J. Zhou, X. Chen, J. Wang, F. Wu, Y. Zhou, and C. Xie, "Leveraging semantic links for high efficiency page-level ftl design," in *2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, June 2016, pp. 84–89.
- [9] H. Lee and S. Ryu, "A high-performance NAND and PRAM hybrid storage design for consumer electronic devices," in *Digest of Technical Papers International Conference on Consumer Electronics (ICCE)*, 2010, pp. 247–248.
- [10] Sheng-Min Huang and Li-Pin Chang, "Exploiting page correlations for write buffering in page-mapping multi-channel ssds," *ACM Trans. Embed. Comput. Syst.*, vol. 15, no. 1, pp. 12:1–12:25, 2016.
- [11] Yi Wang, Duo Liu, Meng Wang, Zhiwei Qin, Zili Shao, and Yong Guan, "Rnftl: A reuse-aware nand flash translation layer for flash memory," *SIGPLAN Not.*, vol. 45, no. 4, pp. 163–172, 2010.
- [12] Chun Jiang Zhu, Kam-Yiu Lam, Yuan-Hao Chang, and Joseph Kee Yin Ng, "Linked block-based multiversion b-tree index for pcm-based embedded databases," *Journal of Systems Architecture*, vol. 61, no. 9, pp. 383 – 397, 2015.
- [13] J. Boukhobza, P. Olivier, and S. Rubini, "Cach-ftl: A cache-aware configurable hybrid flash translation layer," in *2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, 2013, pp. 94–101.
- [14] X. Li, Z. Shen, L. Ju, and Z. Jia, "SRFTL: An adaptive superblock-based real-time flash translation layer for NAND flash memory," in *IEEE International Conference on High Performance Computing and Communications, IEEE 6th International Symposia on Cyberspace Safety and Security, IEEE 11th International Conference on Embedded Software and System (HPCC, CSS, ICESS)*, 2014, pp. 332–339.
- [15] Z. Qin, Y. Wang, D. Liu, and Z. Shao, "A two-level caching mechanism for demand-based page-level address mapping in nand flash memory storage systems," in *2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium*, April 2011, pp. 157–166.
- [16] L. M. Grupp, A. M. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P. H. Siegel, and J. K. Wolf, "Characterizing flash memory: Anomalies, observations, and applications," in *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2009, pp. 24–33.
- [17] J. Thatcher, T. Coughlin, J. Handy, and N. Ekker, "Nand flash solid state storage for the enterprise," in *Storage Networking Industry Association (SNIA)*, 2009.