Probing Internal Architecture of NAND Flash Storage Device

Byeonggyu Park, Dongkun Shin
School of Information and Communication Engineering
Sungkyunkwan University
pbk85011@skku.edu, dongkun@skku.edu

Abstract

Recently card-type NAND flash memory devices such as MMC and SD card are widely used at mobile systems. Since these devices include a flash translation layer (FTL) to handle several special features of NAND flash memory, they are considered as block devices and host side software can be simplified. However, the I/O performance can be significantly different depending on storage access patterns. To provide a high performance, when the host side file systems generate its I/O requests, the internal architecture of flash device should be considered. Unfortunately, however, the internal architecture of card-type NAND flash device is not open to the public. In this paper, we propose a methodology of identifying the internal architecture of NAND flash device by analyzing various performance benchmark results. Two case studies on MMC and SD card devices are presented.

Keywords: NAND flash memory, FTL, MMC, SD card

1. Introduction

Recently card-type NAND flash memory devices such as MMC [1] and SD card [2] are widely used at mobile systems. These devices include a processor and an SRAM as well as pure NAND flash chips. A flash translation layer (FTL) software, which translates a logical address into a physical address, is executed at the processor to handle several special features of NAND flash memory such as erase-before-write characteristic. Since the FTL hides the complicacy of NAND flash memory by providing a block device interface to host system, the host side file system can be simplified.

The I/O performance of card-type NAND flash devices is determined by the internal hardware and software architectures. The hardware architecture includes the sizes of page and block, the number of concurrently accessible flash chips, and the SRAM buffer size. The software architecture is the address mapping algorithm of FTL. Different storage device access patterns of host system will show different I/O performances depending on whether the access pattern corresponds with the hardware and software architectures of the target NAND flash device. For example, if the size of update request is smaller than a physical flash page size, FTL should perform the read-modify-write operation thus a low I/O performance is provided.

Therefore, the host file system developer should consider the internal architecture of the target device to optimize I/O performance. Unfortunately, however, the internal architecture of card-type NAND flash device is not open to the public because it is a critical issue to device vendors. Therefore, in order to optimize the performance of card-type NAND flash devices, we need a methodology of identifying the internal architecture of flash device considering it as a black box.

In this paper, we propose a probing technique for card-type NAND flash devices to identify the internal architecture of the black box devices. By analyzing various performance benchmark results, our technique extracts several hardware and software architectures of the target flash storage devices.

2. Related works

FTLs can be divided into three categories depending on the granularity of address mapping: page-level mapping, block-level mapping and hybrid mapping. The hybrid mapping FTLs reserve several flash memory blocks for a log buffer. While the log blocks in the log buffer use the page-level mapping scheme, normal data blocks are handled using block-level mapping. A hybrid mapping FTL therefore requires a smaller mapping table than does a full page-level mapping technique. When a write request is sent to the FTL, the data is first written into a log block and the corresponding old data in the data block is invalidated.

When all log blocks are full and no empty space remains, one log block is selected as a victim, and all valid pages in that log block are moved into data blocks to make free space for subsequent write requests. This step is called a *log block merge*. There are two kinds of hybrid mapping FTLs, BAST and FAST, depending on the association policy between log block and data block [3].

3. Probing NAND flash storage

We use a performance benchmark tools called uFLIP [4] to generate various types of write requests. uFLIP is designed considering the special features of NAND flash memory. We analyze the performance results of different write patterns to identify internal architecture of card-type flash devices as follows.

The **page** size of flash memory can be identified from the alignment benchmark of uFLIP. When the write request is not aligned to the page size, FTL should access one more page than the aligned requests therefore it provides low performance. Therefore, we can identify the page size by analyzing the response times of write requests with different request size and alignment.

The **superpage** size can be identified from the granularity benchmark of uFLIP. The superpage means a set of pages which can be written simultaneously into parallel flash chips. The granularity benchmark measures the response times for different sizes of write requests. As the size increases, the response time of write requests also increases since it can exploit the parallel multiple flash chips. However, if the request size is larger than the superpage size, there is no change in performance even though the size increases.

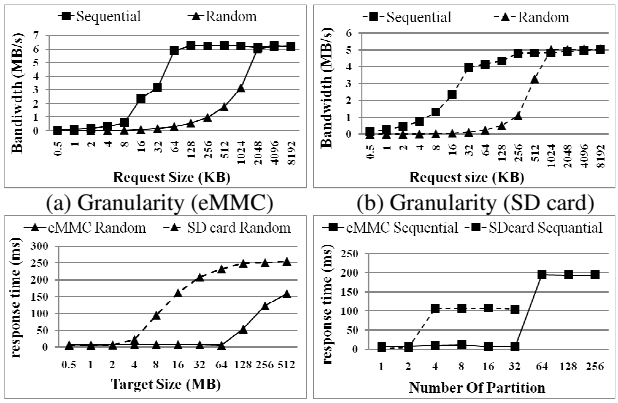
The **block** size also can be identified from the granularity benchmark. Generally, for small-sized requests, a sequential pattern shows a better performance than a random pattern does. However, if the write request size is larger than the block size, both performances of sequential and random write requests are equal.

The **address mapping** scheme of FTL can be identified from the locality and partition benchmarks of uFLIP. While the locality benchmark limits the target address range, the partition benchmark divides the total address space into several partitions and accesses the different partitions alternatively. If both performances of sequential and random write requests are identical, the device uses a page mapping. Otherwise, we can consider the hybrid mapping. When FTL uses the FAST hybrid mapping, there will be no change in performance when the target address range is increased. However, under BAST scheme, we can observe the performance degradation at a large target address space since the large address space invokes frequent log block merge operations.

From the results of locality and partition benchmarks, we can identify the **log buffer** size as well. Under BAST mapping, if the target address space is larger than the log buffer size or the number of partitions is larger than the number of log blocks, there is performance degradation.

4. Cases Studies

We identified the internal architecture of real card-type flash memory devices: 4 GB eMMC (KLM4G2DEJE) and 2GB SD card (MBSB2GMAGMAA). Figure 1 shows the performance results of granularity, locality and partition benchmarks. Each device shows unique results depending on its internal architecture. For example, since the response time of the target eMMC device is increased when the target address space is larger than 64 MB, we can infer that it uses 64 MB log buffer. The partition benchmark shows an identical result since 32 partitions correspond to 64 MB when the block size is 2 MB. From the performance results, we can summarize the internal structures of the target eMMC and SDcard as shown in Table 1.



(a) Granularity (eMMC) (b) Granularity (SD card)

(c) Locality (d) Partitioning

Figure 1. uFLIP benchmark results for the target devices.

Table 1. Internal architectures of the target devices.

	page size	super page size	block size	log buffer size	number of log blocks	mapping scheme
4 GB eMMC	16 KB	64 KB	2 MB	64 MB	32	BAST
2 GB SD card	8 KB	256 KB	1 MB	2 MB	2	BAST

5. Conclusion

Identifying the internal architecture of card-type NAND flash storage is important to design a high performance file system. We proposed a probing methodology which can identify the internal structure of a black box flash storage device. Our future work is to design a high performance file system or I/O scheduler exploiting the extracted parameters of NAD flash storage devices.

6. Acknowledgement

This research was supported by Future-based Technology Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2010-0020724).

7. References

[1] Multimedia Card Association, http://www.mmca.org/
[2] SD Association, http://www.sdcard.org/
[3] S.-W. Lee *et al*. "A log buffer-based flash translation layer using fully associative sector translation," ACM TECS, vol. 6, no. 3, 2007.
[4] uFLIP, http://uflip.inria.fr/~uFLIP/