Chapter 14
HARD: Host-Level Address Remapping Driver for Solid-State Disk

Young-Joon Jang and Dongkun Shin

Abstract Recent SSDs use parallel architectures with multi-channel and multi-way, and manages multiple pages in a group, called superpage, to reduce the size of address mapping information. Such a coavrse grained mapping provides a poor performance for small sized random write requests. To solve the problem, this paper proposes a novel host-level device driver, called HARD, which merges several small sized random write requests into a long sequential write requests. Experimental results showed the proposed HARD improved the random write performance by up to eight times.

Keywords SSD · Flash · Multi-channel · Multi-way · Superpage

14.1 Introduction

NAND Flash memories are widely used in embedded system such as MP3 player, mobile phone and digital camera because it has low power consumption, high random access performance and high shock resistance. Especially, solid state disk (SSD), which is made up of several NAND flash chips, is recently replacing hard disk drive [1]. Since SSD has no moving mechanical part, it shows faster random write performance than HDD. One drawback of flash memory is that it cannot be overwritten without erase operation. Therefore, SSD includes special address

Y.-J. Jang (✉) · D. Shin
School of Information and Communication Engineering,
Sungkyunkwan University, 300, Cheoncheon-dong, Jangan-gu,
Suwon 440-746, South Korea
e-mail: jparkjpark@skku.edu

K. J. Kim and S. J. Ahn (eds.), *Proceedings of the International Conference on IT Convergence and Security 2011*, Lecture Notes in Electrical Engineering 120,
DOI: 10.1007/978-94-007-2911-7_14,

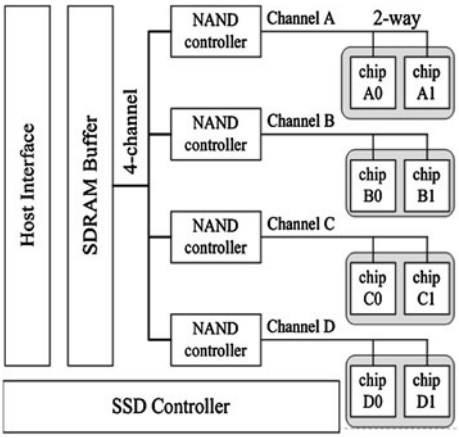

Fig. 14.1 Structure of SSD

translation software, called FTL, which translates a logical address into a physical address.

To increase the I/O bandwidth, multiple NAND flash chips in SSD are accessed simultaneously with multi-channel and multi-way architecture [2, 3]. Figure 14.1 shows an example of 4-channel and 2-way SSD. Two chips using different channels can send/receive data at the same time. Although the data transfer times of two chips using a same channel cannot be overlapped, they can be operated simultaneously.

To efficiently utilize the parallel architecture, SSD FTL manages the address translation information at the unit of a *superpage*. The superpage is a group of multiple pages from different chips which can be accessed in parallel. With the superpage-level mapping, we can reduce the memory space for mapping information. However, the coarse-grained mapping provides a poor performance for small sized random write requests since the read-and-modify operation is required.

In this paper, to solve this problem, we propose a host level address remapping driver (HARD) which merges small sized random write requests into a superpage to provide only sequential write requests to SSD.

14.2 HARD

Figure 14.2 shows the structure and operation of HARD. The Mapping table manages the mapping information between logical address and remapped virtual address. The VSP List manages the virtual superpages (VSP) of which size is equal to SSD superpage size. Each page has a sequential address at the VSP. The status field shows whether the VSPs are allocated or not allocated to any logical page. Therefore, if the status field has been set, the VSP is considered as a free VSP.

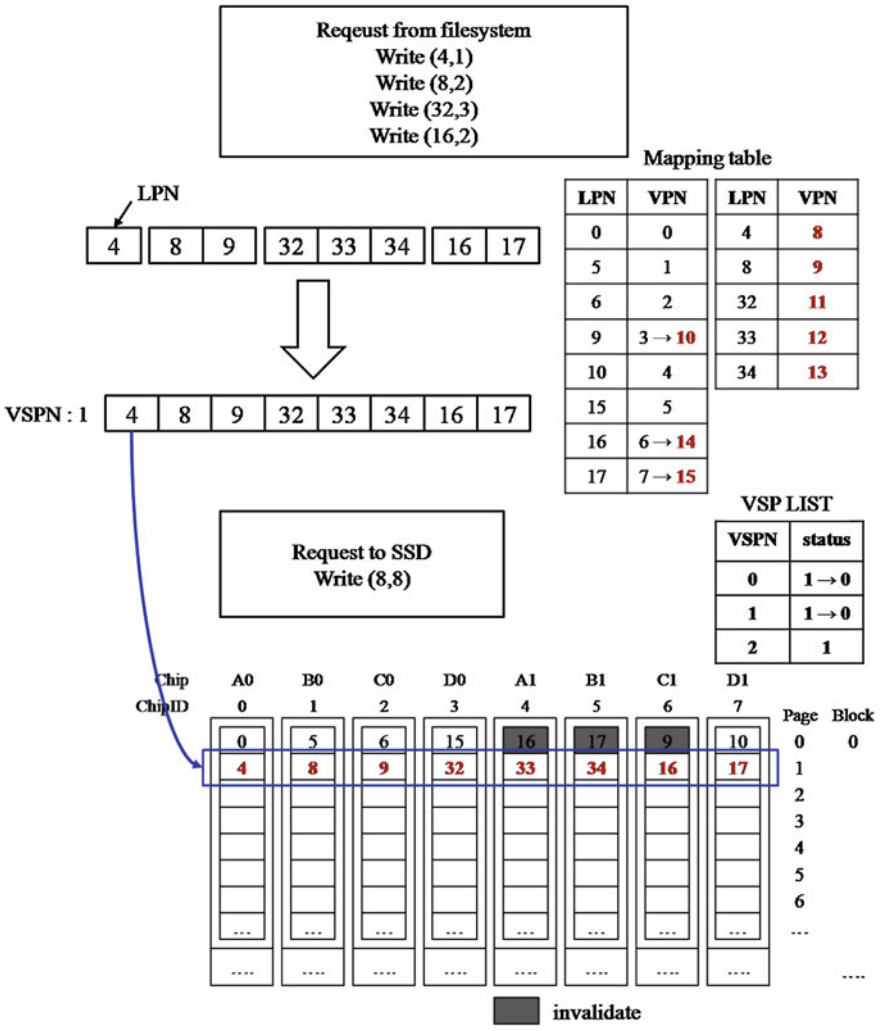

Fig. 14.2 HARD operation

HARD reduces the read-and-modify operation in SSD by allocating a free VSP and merging the small size requests into superpage size requests.

As shown in Fig. 14.2, when the file system sends write requests, HARD allocates a free VSP, merges I/O requests into a superpage and modifies the mapping table. To reclaim virtual address later, HARD invalidates virtual pages which previously allocated to the logical pages since these virtual pages will not be requested. Finally HARD writes data to SSD with virtual page address. Then SSD receives aligned requests of which size is equal to SSD superpage. Therefore, the read-and-modify operation does not occur in SSD.

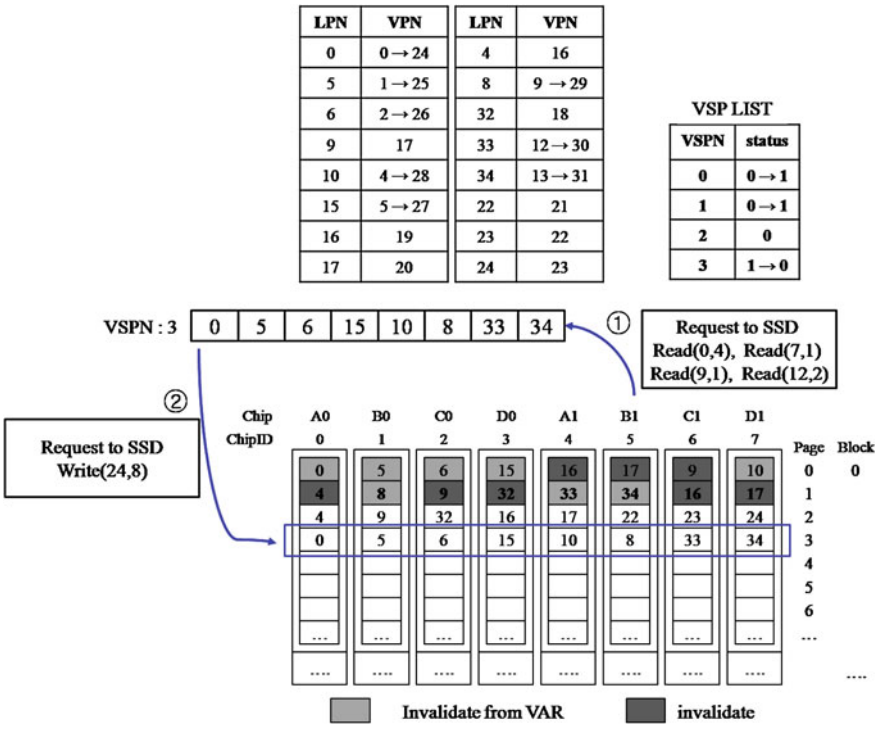

Fig. 14.3 VAR operation

The virtual superpage may be exhausted after many write requests. So HARD should reclaim the virtual superpages which have many invalid virtual pages. We call this operation as virtual address reclamation (VAR). Figure 14.3 shows operation which reclaims VSP 0 and VSP 1. HARD reads valid pages of VSP 0 and VSP1, allocates free VSP 3 and writes these pages to VSP3. Finally, VSP 0 and VSP 1 are considered as free VSP by setting status field in VSP list. Then HARD can exploit VSP 0 and VSP 1 for remapping other write requests.

14.3 Experiment

For the performance evaluation, we added the HARD layer in Linux 2.6.36. As a target workload, SSD used in experiment is Samsung 470 series.

Before the performance evaluation, we need to know the superpage size. To find out the superpage size, we made a tool which generates random write requests with I/O sizes from 4 to 1,024 KB. Figure 14.4 shows the results of random write performance. If I/O size is smaller than 16 KB, SSD cannot show the full performance because of the read-and-modify operation in the SSD. However, if

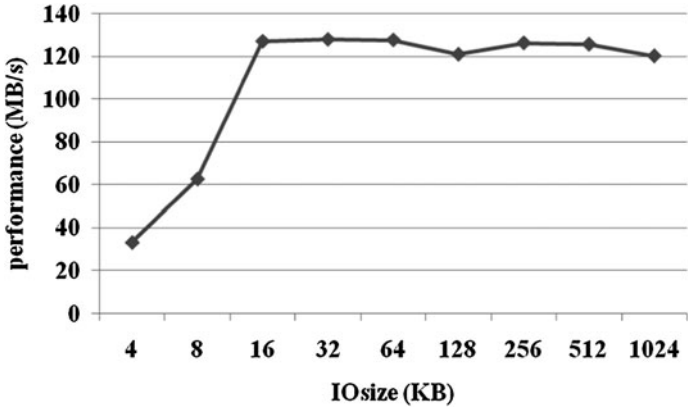

Fig. 14.4 Random write performance

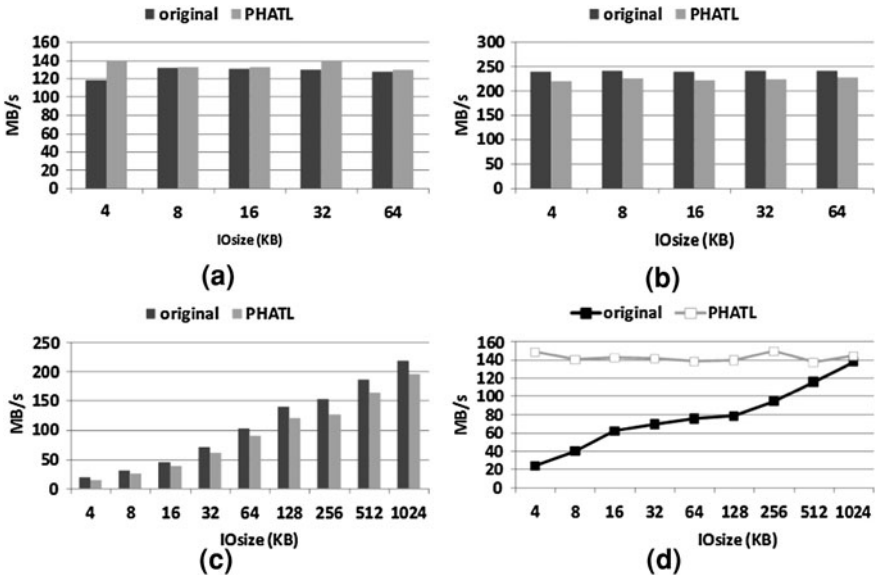

Fig. 14.5 Iozone result. **a** SW. **b** SR. **c** RR. **d** RW

I/O size is larger than 16 KB, SSD shows the full performance since there is no read-and-modify overhead. We set the superpage size as 16 KB based on these results.

We evaluated the performance using IOzone with several IO size. IOzone progresses sequential write, sequential read, random read and random write sequentially.

Figure 14.5 shows that sequential write performance of HARD is similar to that of original I/O scheduler. However, random write has read very large performance

improvement by up to 8 times since HARD arranges the alignment naturally using the VSP. Therefore, the read-and-modify operation can be reduced.

However, there is performance drop in sequential read and random about 5% since a logical address is splited into muti-virtual addresses by mixed mapping which suffers from metadata writes. However, its drop is negligible compared with random write performance improvement.

14.4 Conclusion

To make the best use of superpage mapping, we proposed HARD which converts small size random reqeusts into reqeusts of which size is superpage size using host resource. This scheme shows the performance drop about 5% in read operation, but it shows up to 8 times performance improvement in small size random write.

Acknowledgments 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).

References

1. Reinsel D, Janukowicz J (2008) Datacenter SSDs: Solid footing for growth. http://www.samsung.com/us/business/semiconductor/news/downloads/210290.pdf
2. Park C, Talawar P, Won D, Jung M, Im J, Kim S, Choi Y (2006) A high performance controller for NAND flash-based solid state disk (NSSD). In: Proceedings of IEEE non-volatile semiconductor memory workshop, pp 17–20
3. Kang J-U, Kim J-S, Park C, Park H, Lee J (2007) A multi-channel architecture for high-performance NAND flash-based storage system. J Syst Archit 53(9):644–658