Hybrid Buffer Management using NVM Write Buffer on SSDs

Se Jun Han Dong Hyun Kang Young Ik Eom
Sungkyunkwan University, Korea
{sejun.han, kkangsu, yieom}@skku.edu

Motivation: Over the last few decades, many researchers have studied on Solid State Drives(SSDs). As a result, the performance of the devices is dramatically increased and the cost of the devices is under one dollar per gigabyte. Nowadays, SSDs are widely used in not only the personal computing but also the enterprise environments, and rapidly replacing HDDs which have been traditional main storage media. But, The characteristics of SSDs are different from HDDs because main components of SSD are NAND flash memories that have properties such as asymmetric read/write speeds, limited block erase count, and impossible *in-place-update*. SSD employs its own Flash Translation Layer(FTL) and buffer management scheme in order to overcome aforementioned inherited drawbacks of NAND flash memory. In case of buffer management scheme, SSD mainly employs DRAM as internal buffer. However, the dirty data in the buffer may be lost when power failure happens because DRAM is volatile.

Non-Volatile Memory(NVM) is being considered as an emerging memory device during the last decade. It is persistent, provides similar read latency to DRAM, and is byte-addressable. However, It has insufficient capacity and is expensive until now. Recently, some researches have been studied on employing hybrid buffer, which is comprised of DRAM and NVM, in SSD. One of them is CBM. In order to improve performance of SSDs, CBM employs DRAM as a read cache and NVM as a write buffer. It improves overall performance by replacing several random page writes to a large sequential write. However, if the capacity of DRAM is small, CBM may have low chance to replace random page writes with sequential write, since merging chances with clean pages may be decreased. As a result, CBM may not show improved performance, when the capacity of DRAM is not sufficient.

Our Solution: In this paper, we propose a novel hybrid buffer management scheme that uses both DRAM and NVM as a buffer in SSD. Our scheme manages buffer layer of the SSD as page granularity group and block granularity group to exploit both temporal and spatial localities. In order to consider temporal locality, our scheme takes page granularity *CLOCK* algorithm in DRAM, because *CLOCK* algorithm is one of well-known algorithms for temporal locality. On the contrary, in NVM, in order to consider spatial locality, our scheme takes block granularity *CLOCK* algorithm for managing the pages in NVM. When read or write operations incur buffer misses, the pages are initially stored in DRAM. When DRAM is full, a victim page is selected by *modified CLOCK* algorithm. If the page is dirty, it will be migrated to NVM. If not, it will be deleted. Whenever a free page is needed on NVM, our scheme flushes whole pages of the selected victim block.

Buffer management in DRAM: In order to exploit temporal locality, our scheme employs *modified CLOCK* algorithm, which is different from the conventional algorithm in that is gives preference to clean pages. When buffer misses occur in the buffer layer of the SSD, those pages are initially stored in DRAM. In case of read misses, the pages are stored with "preference" mark. On the contrary, in case of write misses, the pages are stored without "preference" mark.

When DRAM is full, our scheme firstly selects a dirty page as a victim page. Therefore, clean pages stay in the DRAM as long as possible, and the selected dirty page is migrated to NVM ahead of clean pages, where it is called *early migration*. The *early migration* can fully take advantage of the NVM because of the following reasons: First, since NVM is non-volatile, the data consistency of the entire pages in NVM can be guaranteed when power failure happens. Second, unlike the NAND flash memory, NVM supports *in-place-update*, therefore dirty pages in NVM can be updated directly. Third, writing NVM does not require additional operations for reclaiming. Last, In OLTP trace used for our evaluation, there are many read operations than write operations, so keeping clean pages in DRAM is helpful for maximizing the hit ratio.

Buffer management in NVM: In SSDs, frequent random page writes can be disadvantageous. In order to mitigate the disadvantages, NVM gives chances for the random page writes to be transformed into sequential writes by block granularity *CLOCK* algorithm. Also, when the utilization of the victim block is not sufficiently high, our scheme performs flush operations together with clean pages which are related to the victim block in order to increase the possibility of sequential writes. With high ratio of sequential writes, lifespan of SSDs can be extended by improving the possibility for *switch merge*.

Evaluation: We developed a trace-driven simulator, and evaluated the buffer hit ratio and the number of evicted pages from a write buffer in SSD, with financial workload. Our experimental results show that our scheme maintains comparable buffer hit ratio to CBM, and significantly decreases the number of evicted pages by up to 21%.

This work was supported by ICT R&D program of MSIP/IITP. [10041244, SmartTV 2.0 Software Platform]