Yet Another Write-Optimized DBMS Layer for Flash-based Solid State Storage*

Hongchan Roh
Dept. of Computer Science
Yonsei University
Seoul 120-749, Korea
fallsmal@cs.yonsei.ac.kr

Daewook Lee
Dept. of Computer Science
Yonsei University
Seoul 120-749, Korea
daewook@gmail.com

Sanghyun Park
Dept. of Computer Science
Yonsei University
Seoul 120-749, Korea
sanghyun@cs.yonsei.ac.kr

ABSTRACT

Flash-based Solid State Storage (flashSSS) has write-oriented problems such as low write throughput, and limited lifetime. Especially, flashSSDs have a characteristic vulnerable to random-writes, due to its control logic utilizing parallelism between the flash memory chips. In this paper, we present a write-optimized layer of DBMSs to address the write-oriented problems of flashSSS in on-line transaction processing environments. The layer consists of a write-optimized buffer, a corresponding log space, and an in-memory mapping table, closely associated with a novel logging scheme called InCremental Logging (ICL). The ICL scheme enables DBMSs to reduce page-writes at the least expense of additional page-reads, while replacing random-writes into sequential-writes. Through experiments, our approach demonstrated up-to an order of magnitude performance enhancement in I/O processing time compared to the original DBMS, increasing the longevity of flashSSS by approximately a factor of two.

Categories and Subject Descriptors

H.2.2 [**DATABASE MANAGEMENT**]: Physical Design

General Terms

Design, Algorithms, Performance

1. INTRODUCTION

Flash-memory based Solid State Drives (flashSSDs) have been developed so as to resolve the limitations of the flash memory. FlashSSDs consist of a CPU, a RAM buffer, NAND flash memory chips, ECC (Error Correcting Code) modules, host/flash interfaces, and data/control buses that interconnect between each element of flashSSDs. In general, flashSSDs extend its capacity by embedding a plurality of the flash memory chips, and enhance write-throughput by utilizing parallelism between the flash memory chips. By the definition of Storage Networking Industry Assoication [1], flash-memory based Solid State Storage (flashSSS) includes all kinds of flash memory based storage devices such as flash memory chips, flashSSDs, and RAID solutions including a number of flashSSDs.

We define common problems of flashSSS caused by frequent write-operations as write-oriented problems. Raw flash-memory chips have asymmetric read/write throughput due to the much longer page-write latency regardless of the access pattern. It also has limited lifetime, which is determined by the maximum erase count of a flash-memory block. Consequently, frequent write-operations make the response time of flash memory worse, meanwhile shortening the lifetime of flash memory. For flashSSDs, access pattern to the devices has more significance than raw flash memory chips. FlashSSDs are vulnerable to the random-writes. Random-writes make the write-throughput of flashSSDs decrease, and it is questionable whether the sufficient longevity of a flashSSD device can be assured when a number of write-operations are requested in OLTP (Online Transaction Processing) environments.

Although IPL [4] and AppendPack [5] tried to address the write-oriented problems, each method has the following drawbacks. IPL proposed its own storage manager and buffer manager. In the buffer manager, additional memory space called in-memory log sector is allocated for every DBMS-page. Whenever a DBMS-page is updated, the dirty region is logged into the in-memory log sector. If the DBMS-page is evicted by the buffer manager, only the in-memory log sector is written to the log region, not the entire DBMS-page. Since the log-sector is smaller than the DBMS-page, the IPL storage manager is able to reduce page-writes considerably. However, if multiple log sectors for a DBMS-page are written to the log region, IPL induces additional cost to read the multiple log sectors when reading the DBMS-page. AppendPack replaces random writes into sequential ones through a new data layout. AppendPack adopts a mapping table analogous to the FTL mapping idea of flashSSDs, and contiguously writes the evicted DBMS-pages to the storage space. The written locations of the DBMS-pages are mapped to the DBMS-pages by the mapping table. However, this approach is not enough to

*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(2010-0010689). This work was also supported by Seoul Metropolitan Government 'Seoul R&BD Program(PA090903)'.

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.