



Journaling deduplication with invalidation scheme for flash storage-based smart systems[☆]



Seung-Ho Lim^a, Young-Sik Jeong^{b,*}

^a Department of Digital Information Engineering, Hankuk University of Foreign Studies, Republic of Korea

^b Department of Multimedia Engineering, Dongguk University, Republic of Korea

ARTICLE INFO

Article history:

Received 11 June 2013

Received in revised form 24 February 2014

Accepted 2 April 2014

Available online 26 April 2014

Keywords:

Filesystem

Journaling

FTL

Invalidation

Smart system

ABSTRACT

Transaction support for filesystems has become a common feature in modern operating systems where data atomicity is achieved by writing transactions to the log region in advance. The logging mechanism is appropriate for flash storage devices due to the inherent nature of flash memory. However, the logging schemes inherently create multiple copies of data, leading to a decrease in the bandwidth of storage systems. In this paper, we present a simple and efficient invalidation scheme for multiple copies of data in a common journaling module. We identify two types of duplications, one in which there is an explicit duplication of the journal region and original region with the same data, and the other in which there is an implicit duplication of transaction commit operations. The invalidation of duplicated data reduces internal write and erase operations and garbage collection overhead for flash devices, which would otherwise increases external I/O bandwidth. Experimental results show that the overall performance improves roughly from 5% to 35% with the invalidation scheme for journal transactions.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Recently, the capacity of NAND flash memory chips has become large enough for them to be used as a part of the main storage medium of electronic devices, and this size will continue to increase quickly. Flash-based storage devices, such as Solid State Drives (SSDs) and embedded Multi-Media Controller (eMMC) cards, have reached the mainstream market as storage solutions. SSDs are rapidly taking the place of traditional mechanical hard drives since they support the same interfaces at higher layers, and eMMC has also reached wide adoption in mobile systems.

However, there are several critical limits for flash memory. If data is written to flash memory, write operations should be preceded by erase operations. In other words, in-place update is not allowed in flash memory. Read and write commands are performed in pages, whereas erase operations are performed in blocks, whose size is much larger than that of pages. Thus, write operations should be implemented with efficient erase operations that perform proper garbage collection (GC) operation, which is the

process that makes regions available for writes. In most cases, the internal flash controller and the firmware of flash storage devices hide the limitations of the use of flash memory from the upper layers of the storage system, so that applications don't need to be changed. The main role of the flash controller and the firmware is the address translation between the logical address of the file system and the physical address of the flash memory itself. The controller and firmware perform out-of-place updates which in turn help to hide erase operations in the flash memory. Address translation is based on the internal mapping table.

Despite the advantages of flash-based storage, physical limitations necessitate the application of software-guided enhancement schemes to flash-based storage systems, including buffering, compression, exploiting parallelism, error correction, deduplication, etc. With the advent of larger and highly-accessible storage systems, ensuring data reliability and consistency has become one of the most important issues for modern storage systems. Supporting transactions in file systems has become a common feature in operating systems, where data atomicity is enabled by writing transactions to the log region in advance, within the start of the transaction and commit semantics. Logging is well-suited for flash devices because of the inherent copy-on-write nature of flash memory.

Several studies were performed attempting to reduce the overhead of transactional operations for flash devices. However, most

[☆] This work was supported by Hankuk University of Foreign Studies Research Fund of 2014.

* Corresponding author. Tel.: +82 10 6372 5339.

E-mail addresses: slim@hufs.ac.kr (S.-H. Lim), ysjeong@dongguk.edu (Y.-S. Jeong).

Download English Version:

<https://daneshyari.com/en/article/460444>

Download Persian Version:

<https://daneshyari.com/article/460444>

[Daneshyari.com](https://daneshyari.com)