Journal of Systems Architecture 88 (2018) 1–12

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Efficient flash-aware page-mapping cache management for on-board remote sensing image processing

Tong Zhang[a,*], Ping Luo[a,*], Ze Cheng[a], Jing Li[b]

[a] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China
[b] Department of Geography and the Environment, University of Denver, Denver, CO 80208-0710, USA

ARTICLE INFO

Keywords:
Page-mapping
Cache replacement
Remote sensing
Priority
Trace

ABSTRACT

The use of flash-based storage devices has become popular in both research and practice due to their intrinsic advantages over traditional magnetic counterparts. Many advanced flash-aware data management techniques have been developed to exploit the full potential of flash storage devices. It is then reasonable to adopt flash storage devices to deliver high I/O performance for on-board remote sensing image processing tasks, which pose particular challenges in data storage and I/O management. In this work, we focus on caching issues and propose a page-mapping cache management approach based on a page-level flash translation layer (FTL) to accommodate various types of workloads presented by different remote sensing image processing operations. The adaptability is achieved by three major techniques: 1) separating mapping cache into hot and cold regions according to access frequency; 2) a probability-based locality-aware cache replacement priority model to balance high hit ratio and low write/erase costs; and 3) adaptive mapping cache replacement algorithms that are seamlessly integrated with 1) and 2) to handle complicated I/O patterns efficiently. Combining these above three techniques, the proposed FTL-based cache management approach is superior to several existing methods, as demonstrated by extensive experiments running on realistic on-board remote sensing processing I/O traces.

1. Introduction

NAND flash memory-based storage devices are featured with attractive advantages: superior performance for random accesses, small form factor, low power consumption, and shock resistance. These merits enable flash memory-based devices to gradually replace traditional electromechanical magnetic disks as main storage media in personal computers, mobile devices, and even large-scale massive data centers over the last decade [17,21]. Researchers have been endeavored to develop advanced flash-aware data management and flash translation layer (FTL) techniques to exploit the full potential of flash memory-based storage devices [14,21]. Among these data management techniques, cache management plays an important role and has received wide attention. The built-in cache of flash memory can be effectively used to promote I/O performance, which has been demonstrated by man recent studies reported in the literature [6,11,12,18,22,28].

The last ten years witness a rapid development of remote sensing imagery acquisition techniques. Earth observation relies on satellites, high-altitude balloons, unmanned aerial vehicles, moving vehicles on the ground or other mobile devices to collect various information of the earth system. Massive amounts of remote sensing imagery data are being collected at an unprecedented rate and scale. This situation creates an increasing gap between data acquisition and capabilities to transfer and process these huge amounts of data. A plausible solution is to transfer image processing capabilities from ground stations to in-situ data collection platforms. Lately, this type of applications starts gaining momentum because of its utilities of reducing unnecessary data transfer to ground stations and promoting the efficiency of information extraction. Real-time on-board remote sensing image processing can exploit flash-based storage devices to promote I/O performance in light of many salient advantages of flash memory. Flash memory-based storage presents both opportunities and challenges for the management of massive geospatial data. While recent studies have explored the design of spatial indices for vector data [8,24,27], there have been much less attention dedicated to flash-aware remote sensing data storage and I/O management. However, on-board image processing faces many challenges due to constrained resources on earth observation platforms. Because only very limited storage resources are available, it is desirable to design efficient storage management methods for on-board image processing.

Cache management is a critical component for flash-based storage and has attracted research interests over the last decade [6,18,22]. On-

* Corresponding authors.
E-mail addresses: zhangt@whu.edu.cn (T. Zhang), pluto@whu.edu.cn (P. Luo), chengze@whu.edu.cn (Z. Cheng), jing.li145@du.edu (J. Li).

https://doi.org/10.1016/j.sysarc.2018.04.003
Received 21 November 2017; Received in revised form 9 March 2018; Accepted 30 April 2018

board remote sensing processing poses special challenges for flash-based cache management because of its particular data access patterns that are difficult to predict. Generic flash-based cache management schemes are thus unable to deliver satisfactory performance for on-board remote sensing processing tasks, which motivates us to re-consider the design of flash-aware cache management for this type of increasingly important applications.

Typical remote sensing processing tasks involve multiple inter-dependent sub-tasks, which may create complicated I/O patterns that the current flash software cannot handle efficiently. For example, an object detection task consists of image segmentation, feature extraction, and object detection, each of which has particular spatial-temporal I/O patterns. Their I/O traces consist of long strided sequential segments mixed with small-sized random patterns incurred by metadata accesses and communications. Compared with generic I/O traces, remote sensing processing traces generally have much larger request sizes and standard deviations. The number of sequential read dominates the overall read requests whereas sequential writes are relatively less dominant. The inter-arrival time varies much more dramatically larger. To summarize, remote sensing processing traces have significant variations in terms of request sizes and access frequency. In addition, the access frequencies on address mapping pages are not as concentrated as generic I/O traces. Most pages rarely have more than three consecutive accesses with the duration of test. Many traditional cache management methods are thus incapable of handling such access patterns because their cache replacement strategies may lead to too many cache misses with significant higher overheads.

The current methods are not delicate enough to accommodate these particular access patterns. To this end, we propose a comprehensive page-level cache management scheme that accounts for recent access frequency and locality within individual translation pages using fined-grained organization of access mapping data and responsive cache replacement algorithms for cached mapping tables. The reason we chose to investigate the cache management of access mapping data rather than realistic image data was because the proposed approach can be readily used in page-level address mapping in a FTL. It also complements other functionalities in a FTL and helps promote the overall I/O efficiency of remote sensing image processing operations. To the best of our knowledge, this study is the first to develop specialized cache management scheme for on-board remote sensing image processing tasks. The contributions of this paper are summarized as follows:

(1) An adaptive caching scheme to manage page-level mapping information. Being compatible with the current FTL architecture, the proposed caching scheme is responsive to varied heterogeneous I/O workloads generated by typical on-board remote sensing image processing operations;
(2) A locality-aware mapping cache replacement algorithm based on a probability priority model. The proposed replacement policy accounts for spatial and temporal locality of I/O requests;
(3) A comprehensive trace-driven evaluation to demonstrate the applicability and effectiveness of the proposed cache management approach using realistic remote sensing image processing workload traces.

2. Related work

2.1. Flash memory

NAND flash memory exhibits significant different particularities than traditional magnetic disks. In addition to read and write, flash memory needs an extra type of operation, i.e., erase, in order to perform formatting before writing data into the same location. Due to this erase-before-write feature, it is preferable to perform out-of-place update rather than in-place update concerning I/O efficiency and lifetime saving of flash devices. Erase operations are executed at a coarse granularity of block whereas read and write operations are performed at a fine-grained page level. The three types of operations present asymmetric performance: erase operations cost an order of magnitude more time than read and write operations. Compared with read and write operations, the number of erase operations that can be performed on each block is limited (usually between 10,000 and 100,000). Consequently, it is desirable to balance the number of erase operation over blocks to extend the lifetime of flash memory as long as possible (i.e. wear-leveling). All these above particularities of flash memory pose challenges to handle asymmetric read-write-erase performance, out-of-place update, and limited endurance when facing complicated I/O requests.

The most widely adopted approach to address these problematic issues is to integrate a dedicated software, coined “flash translation layer” (FTL) into flash memory controllers, aiming to implement specific modules to improve flash memory performance and endurance. Typically, a FTL implements the following functions: (1) address translation; (2) garbage collection; (3) wear leveling, and (4) cache management. Address translation is needed for out-of-place update to map the logical addresses of requested data to corresponding physical addresses in flash memory. These logical-to-physical mapping data are critical during the process of address translation and data access. Hence, we focus on the cache management of address mapping data in this study. Garbage collection selects and reclaims invalidated pages for future uses when the number of free blocks decreases to a certain threshold. Wear leveling is responsible for distributing erase operations evenly over blocks in order to prolong the lifespan of flash-based devices.

According to the mapping granularity, there are three primary categories of address translation in FTL: page-level mapping, block-level mapping, and hybrid mapping. A fine-grained page-level mapping [4,9] records logical-to-physical address translation information as individual entries at the page level, and is highly efficient in garbage collection yet demanding on memory space. A block-level mapping [15,23] requires little memory space but may incur extra unnecessary operations if only part of a block is accessed. Between these two extremes, hybrid mapping [13,16] uses block-level mapping for data blocks and manages a limited number of log blocks with page-level mapping. In this paper, we use the page-level mapping due to its flexibility in address translation and efficiency in garbage collection.

2.2. Cache management

Research on the cache management of flash memory-based devices has proliferated during the past decade. In addition to hit ratio, researchers have realized that it is also important to reduce replacement cost incurred when dirty pages are evicted from cache. Clean-first least recently used (CF-LRU) is an early work to account for both replacement cost and hit ratio [22]. The CFLRU scheme divides the LRU list into two separate adjustable regions, namely, a working region and a clean-first region. Eviction is performed in the clean-first region where high priority is given to clean pages over dirty pages. After CFLRU, more additional factors, such as access frequency, numbers of erase operation, and wear-leveling degree, have been taken into account in the literature. Historical cache replacement policies differ mostly in cache structures and replacement criteria. Many cache replacement methods use multiple LRU lists to manage hot/cold or clean/dirty pages separately, hoping to address particular performance improvement or life prolonging needs. Yoo et al. [28] proposes a set of new replacement algorithms that exploit the information of block erase count and dirty-page access frequency and demonstrate the advantages of the proposed algorithms in reducing erase operations and improving wear-leveling over CFLRU. However, it is required to pre-define the size of window for the region that stores eviction candidates. Jung et al. [12] proposes to trade hit ratio for reducing the numbers of write and erase operations by delaying the eviction of non-cold dirty pages as late as possible. This