ARTICLE IN PRESS
Pervasive and Mobile Computing ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

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

Exploiting write-only-once characteristics of file data in smartphone buffer cache management

Dohee Kim, Hyokyung Bahn *

Department of Computer Science and Engineering, Ewha University, Seoul, 120-750, South Korea

ARTICLE INFO

Article history:
Received 5 September 2016
Received in revised form 23 November 2016
Accepted 10 January 2017
Available online xxxx

Keywords:
Buffer cache
Smartphone
File data
Write-only-once
Power consumption

ABSTRACT

In this article, we analyze file access characteristics of smartphone applications and find out that a large portion of file data in smartphones are written only once. This specific phenomenon appears due to the behavior of SQLite, a lightweight database library used in most smartphone applications. Based on this observation, we present a new buffer cache management scheme for smartphone systems that considers non-reusability of write-only-once data that we observe. Buffer cache improves file access performances by maintaining hot data in memory thereby servicing subsequent requests without storage accesses. The proposed scheme classifies write-only-once data and aggressively evicts them from the buffer cache to improve cache space utilization. Experimental results with various real smartphone applications show that the proposed buffer cache management scheme improves the performance of smartphone buffer cache by 5%–33%. We also show that our scheme can reduce the buffer cache size to 1/4 of the original system without performance degradation, which allows the reduction of energy consumption in a smartphone memory system by 27%–92%.

1. Introduction

Buffer cache is widely used in mobile smart devices as well as traditional computing systems in order to alleviate the performance gap between memory and storage [1–5]. Buffer cache temporarily maintains recently used file data in memory, thereby servicing subsequent requests for the cached data directly without accessing slow storage. However, data inconsistency may arise if a sudden power failure occurs while cached data are being reflected to storage. To prevent such situations, SQLite, a lightweight database library, is usually used in smartphone applications [6]. To modify file data in SQLite, a copy is first made, which allows the recovery of corrupted file data to a previous consistent version upon a system crash [7]. However, it causes inefficiency as an additional data copy is created in the buffer cache. For example, if a file data A_{data} needs to be updated, a backup data $A_{journal}$ is first created and then flushed to storage. Once the data has been updated, as A_{data} is now the most recent version of A, all subsequent read/write requests are serviced by A_{data}. Nevertheless, $A_{journal}$ is still maintained in the buffer cache as file systems cannot recognize it as an old version of data that will never be used. Although $A_{journal}$ is necessary when a system crash occurs, we use the flushed copy in storage, not the cached copy. This useless copy in memory excessively wastes the buffer cache space of smartphones.

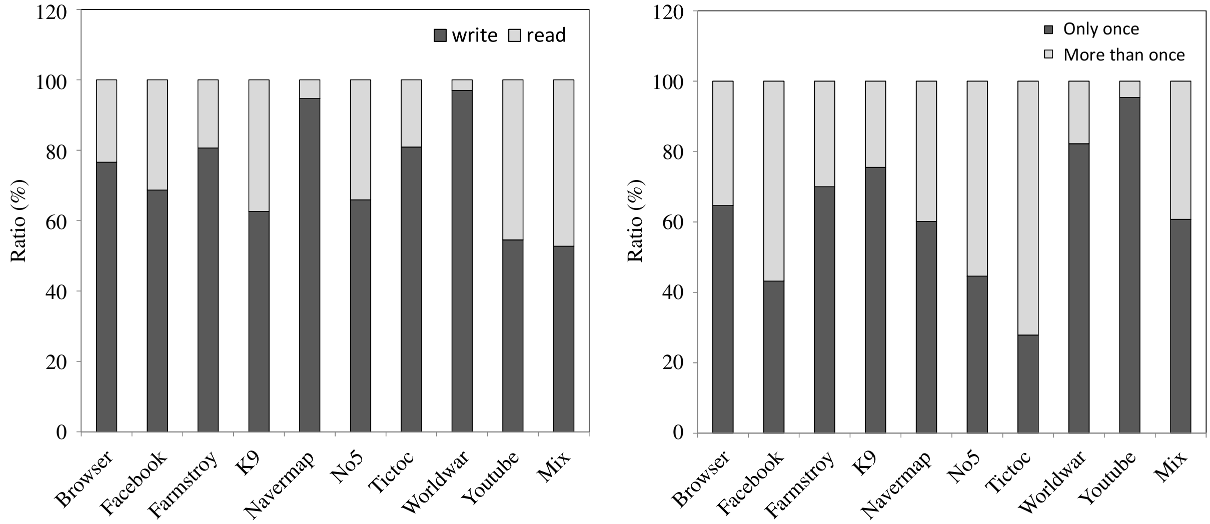
In this article, we analyze file access characteristics of smartphone applications and find out that a large portion of file data in smartphones are written only once. The main reason behind this phenomenon is the SQLite journaling, which creates an

* Correspondence to: Department of Computer Science and Engineering, Ewha University, 11-1 Daehyun-dong, Seodaemun-gu, Seoul, 120-750, South Korea. Fax: +82 2 3277 2306.

E-mail addresses: glf215@ewhain.net (D. Kim), bahn@ewha.ac.kr (H. Bahn).

http://dx.doi.org/10.1016/j.pmcj.2017.01.004
1574-1192/

Please cite this article in press as: D. Kim, H. Bahn, Exploiting write-only-once characteristics of file data in smartphone buffer cache management, Pervasive and Mobile Computing (2017), http://dx.doi.org/10.1016/j.pmcj.2017.01.004



(a) Write vs. read.

(b) Write-only-once vs. writes more than once.

Fig. 1. File write distributions for various smartphone applications.

old copy of file data before updating. As this old copy has a different file name from the updated data, it is classified as a write-only-once file data, which wastes the buffer cache space significantly. To alleviate this problem, we propose a new buffer cache management scheme for smartphone systems that considers non-reusability of write-only-once data. Specifically, the proposed scheme classifies write-only-once data and evicts them from the buffer cache when they are being flushed to storage. This allows more useful blocks to be cached, leading to an improved cache hit ratio. Trace driven simulations with a variety of smartphone applications show that the proposed buffer cache management scheme improves the cache hit ratio by 5%–33% compared to the systems that do not use it. Furthermore, we observe that we can reduce the buffer cache size to 1/4 of the original system without any performance degradation if our scheme is adopted to the smartphone buffer cache. This leads to the reduction of energy consumption in a smartphone memory system by 27%–92%, increasing the battery lifetime significantly.

The remainder of this article is organized as follows. Section 2 analyzes file access characteristics of smartphones and presents a new buffer cache management scheme considering the write-only-once data in file accesses. Section 3 presents the performance evaluation results to assess the effectiveness of the proposed scheme. In Section 4, we briefly summarize some related studies of the proposed buffer cache management scheme. Finally, Section 5 concludes this paper.

2. A new buffer cache management scheme

2.1. File access characteristics in a smartphone

We analyze file access characteristics of various real smartphone applications. Fig. 1(a) shows the file read/write ratio of each smartphone application running on ODROID-Q with Android version 5.0 and Linux kernel 3.4.0 installed. As shown in the figure, all smartphone applications exhibit write-intensive access patterns. In some applications such as Worldwar the ratio of writes is over 97%. Such high ratio of writes happens as smartphone applications use the SQLite library in file manipulations. Note that this is different from traditional desktop applications in which reads account for a large portion of file accesses [8].

Fig. 1(b) classifies write accesses of smartphone file data into write-only-once and writes more than once. As shown in the figure, the ratio of write-only-once data is 28%–95%. Note that these large portions of write-only-once data will not be used again although we cache them. This implies that such data are not helpful in improving system performances but pollute the precious cache space.

2.2. Early eviction of write-only-once data in cache management

Buffer cache manages file data by a unit of block, and cached blocks can be classified as clean and dirty. A clean block indicates that the cached data has not been modified since it was loaded into the cache, and thus it is identical to the original block in storage. If a clean block is evicted from the cache, it can simply be discarded. In contrast, if a write operation is performed on a cached block, it becomes dirty. Dirty blocks should be flushed to storage before evicted from the cache. To defense a system crash or a power failure situation, modern file systems such as Ext4 do not delay the flushing of dirty blocks until their eviction, but periodically flush modifications to storage [9]. For example, Ext4 flushes dirty blocks to storage every

Please cite this article in press as: D. Kim, H. Bahn, Exploiting write-only-once characteristics of file data in smartphone buffer cache management, Pervasive and Mobile Computing (2017), http://dx.doi.org/10.1016/j.pmcj.2017.01.004