The Journal of Systems and Software 86 (2013) 1408–1425

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

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

Lifetime and QoS-aware energy-saving buffering schemes

Liang Huai Yang[a,b,*], Jian Zhou[a,b], Weihua Gong[a,b], Jiakui Zhao[c], Lijun Chen[d]

[a] School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China
[b] Key Lab of Visual Media Intelligent Process Technology of Zhejiang Province, Hangzhou, China
[c] State Grid Electric Power Research Institute, Beijing, China
[d] School of Information Science and Technology, Peking University, Beijing, China

ARTICLE INFO

Article history:
Received 4 August 2012
Received in revised form 7 November 2012
Accepted 8 January 2013
Available online 18 January 2013

Keywords:
Energy conservation
Heterogeneous drive
Replacement policy
Buffering scheme
QoS

ABSTRACT

The heterogeneous drive (HDrive), which combines solid-state disk (SSD) and HDD, brings opportunity for energy-saving and has received extensive attention recently. This paper focuses on the file buffering schemes and adaptive disk power management (DPM) scheme for HDrive. As for the first issue, we propose a frequency–energy based replacement (FEBR) scheme based on an energy-cost model; as for the second issue, we present a sliding-window based adaptive DPM scheme by taking the HDD's lifetime into account. To make the trade-off among performance, HDD's lifetime and energy-saving, we contrive a QoS-aware DPM scheme. With extensive experiments on four real-world traces, we have evaluated the effectiveness of existing replacement schemes on energy-efficiency, performance, and HDD's lifetime and compare with our proposed schemes. The experimental results have demonstrated that energy-saving in HDrive is feasible and can reach as high as 60–80%, and that FBR and its variant FEBR, and GDS are the best ones among all those online schemes evaluated while FEBR has some advantage over FBR and GDS on the whole. The results have also revealed that our proposed adaptive sliding-window-based DPM scheme can effectively control the disk's lifetime and the QoS-aware DPM scheme works well in making tradeoffs among performance, HDD's lifetime and energy-saving.

1. Introduction

The energy consumption of storage subsystem takes up 20–30% within general-purpose computer systems (Douglis et al., 1994; Greenawalt, 1994); in data centers, it takes up 50% of the total energy consumption in systems. What is more, the energy consumption of storage subsystem is exacerbated by adding RAM to improve system performance (Kgil and Mudge, 2009). As the CPU architecture shifts from single core to multi-core, the energy consumption of storage subsystem accounts greater proportion in multi-core systems compared with single core systems (Geer, 2005). Many researchers have focused on reducing the energy consumption of the storage subsystem from different perspectives. The main idea is to predict the arrival time of next disk request, and if the idle time is long enough, then the disk is turned into low power state to save energy; when the access request arrives, the disk spins up into active state immediately to serve the request.

The advent of non-volatile, anti-vibration and low-power flash memory brings new opportunities for energy-conservation. NAND flash and SSD are now widely used, and how to use SSD to improve energy efficiency has been a popular topic in recent years. To fully utilize flash memory, smarter softwares are needed to drive systems and systems are also needed to be re-designed to make them faster, cheaper, greener and energy efficient. Narayanan et al. (2009) considered that the current cost/GB of SSD is 3–3000 times higher than HDDs, therefore it is hard to replace the role of HDD. However, comparable energy savings are achievable with low-power SATA disks if SSDs are used as the middle layer of the hybrid hard drive for buffering or write-ahead logs. Graefe and Alto (2009) pointed out that, within reasonable cost, the combination of SSD and HDD could create more idle time for energy saving. Over the past years there has been some research on this topic; we will discuss them in the related work section.

This paper focuses on file-grain buffering scheme for energy efficiency of heterogeneous drive. The basic idea is to buffer the frequently accessed files in SSD to extend the HDD idle time, and thus create the opportunity for the HDD to stay longer in the sleep/off state. Such scheme is especially useful in the daily office work scenario, where users usually focus on a certain piece of work within a period of time and access a cluster of related files, which exhibits strong access locality. Meanwhile, the performance is effectively improved due to the read speed advantage of SSD. The conceptual structure of the heterogeneous drive (HDrive) this paper investigates is shown in Fig. 1.

* Corresponding author at: School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. Tel.: +86 13003628652.
E-mail addresses: yang.lianghuai@gmail.com, yanglh@zjut.edu.cn (L.H. Yang).

0164-1212/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2013.01.014

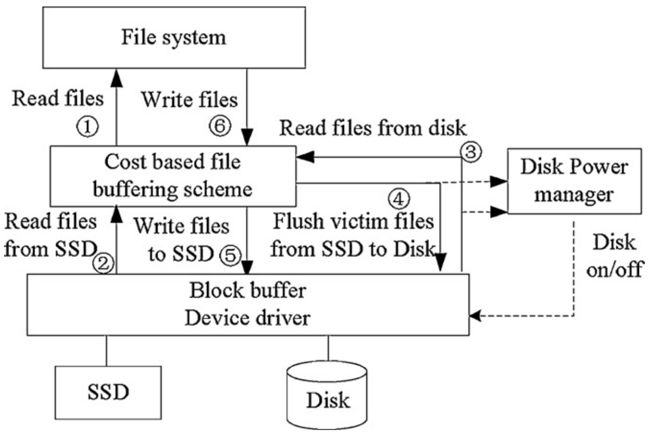

Fig. 1. The conceptual architecture of a HDrive.

In this architecture, SSD is used as file buffer and the file system is enhanced with a cost-based file buffer manager. The buffer manager monitors file operations (read, write, create, delete, change) in real-time, keeps track of file access statistics, admits/replaces files and decides file movement and synchronization between SSD and HDD. The power manager determines the state transition of HDD when HDD file access requests arrive. In our proposed HDrive, files can exist in both SSD and HDD. Newly created files only exist in SSD, and are moved to HDD when they are selected as victims (④); file copies are retained in HDD when they are migrated to SSD (③), and those unmodified SSD files are simply abandoned when replacement happens while those affected SSD files are flushed to HDD (④). This strategy trades space for performance.

For the file-grain energy saving in HDrive, one of the key issues is to design an effective file replacement algorithm. The offline optimal solution for caching files with different sizes and access costs is NP-hard (Chrobak et al., 2010; Albers et al., 1999). Recently, there are some online competitive analysis results for this problem (Epstein et al., 2011; Young, 2002). Are those in-memory web server caching algorithms (Podlipnig and Boszormenyi, 2003; Balamash and Krunz, 2004; Wong, 2006) applicable for the case of energy efficient HDrive? Or are those classical page replacement algorithms such as LRU (Mattson et al., 1970) applicable? Since LRU, a series of page caching algorithms are proposed, such as MRU, LFU (Mattson et al., 1970), FBR (Robinson and Devarakonda, 1990), LRFU (Lee et al., 2001), MQ (Zhou et al., 2001), 2Q (Johnson and Shasha, 1994), LRU-*K* (O'Neil et al., 1993), LIRS (Jiang and Zhang, 2002), ARC (Megiddo and Modha, 2003) and CAR (Bansal and Modha, 2004); each overcomes certain shortcomings of the other. One question is that, in the file-grain scenario, which algorithm(s) perform(s) better? In this respect, Liu et al. (2010) compared LRU with FBR, their preliminary experimental results showed that HDrive achieved good energy saving. And their work leaves us many questions to be answered. For example, will the adaptive replacement algorithm ARC, which outperforms LRU in page-grain scenario, still do so in file-grain scenario and have better energy-saving effect? What about other algorithms? There exist a multitude of replacement policies; almost every strategy claims to outperform the others, and sometimes even reaches conflicting conclusions. The reason is that applications have different scenarios. No strategy can do well in all scenarios. Considering the new scenario, is it necessary to reinvent the wheel? The energy saving of HDrive relies on the HDD on/off state transition, and the HDD on/off transitions affects the HDD's lifetime, so what kind of power strategy can ensure HDD's lifetime and still achieve good energy-efficiency? This paper aims to answer these questions.

Our contributions are as follows:

(1) Comprehensive comparisons are made on various energy-saving replacement schemes for HDrive. Their effectiveness on hit rate, energy-efficiency, performance, and HDD's lifetime is evaluated with extensive experiments on four real-world traces. And we draw some useful conclusions.
(2) Based on the energy-cost model, an improved scheme called frequency–energy based replacement (FEBR) is proposed by adapting the existing online replacement algorithms FBR. Web cache algorithm GDS (Cao and Irani, 1997) does well too; FBR, FEBR and GDS are better than the other on-line algorithms, while FEBR is the best. The optimal page replacement algorithm OPT (Belady, 1966) is not optimal anymore in file-grain buffer scheme.
(3) To ensure HDD's lifetime, an adaptive sliding-window based disk power management mechanism is proposed. It saves energy without compromising HDD's lifetime. We find that the traditional time-out duration of 5–10 s is not suitable for HDD's lifetime; it must be adjusted dynamically according to users' access patterns. Experiments show that the proposed strategy is effective.
(4) A QoS-aware disk power management scheme is proposed. There exists a trade-off among performance, energy-saving and disk's lifetime. It is highly desired that user can enforce an energy-saving HDrive without sacrificing too much performance drop, say, α%. To the best of our knowledge, this is the first attempt to tackle this trade-off.

This paper is organized as follows: Section 2 introduces the related work; Section 3 describes the file-grain buffer replacement algorithm framework for HDrive, the improved algorithm FEBR and the (QoS-aware) adaptive disk power management scheme; Section 4 presents experimental results; and Section 5 summarizes this paper.

2. Related work

2.1. Hybrid storage

Flash memory was first introduced in 1988. Due to its potential application value, various design combinations of flash memory and HDD have been proposed to achieve certain desired goals on the dimensions of price, capacity and performance. The ways of combining flash memory and HDD can roughly be divided into two categories: hybrid drive (Bisson and Brandt, 2007) and combo drive. For hybrid drive, flash memory is integrated inside HDD as part of it, and the granularity of allocating and scheduling is block. The goal is to extend idle time of HDD. Bisson and Brandt (2005) used flash memory as the read-only buffer of HDD to reduce the spin-ups.

A combo drive is the heterogeneous combination of SSD and HDD with different characteristics, the caching granularity is in block or file. Baker et al. (1992) proposed to use non-volatile random access memory (NVRAM) as the file system buffer to improve performance and reliability. Marsh et al. (1994) examined the impact of flash memory as a second-level file system buffer cache to reduce power consumption and file access latency on a mobile computer. They used LRU and FIFO with a fixed timeout strategy for HDD power control. Hsieh et al. (2007) used flash memory as the disk buffer by mapping the disk blocks to the flash memory with logical block addressing technology. Payer and Sanvido (2009) proposed the combo drive strategy which is the combination of a small capacity SSD and a large capacity HDD and used as a single disk. Many standard file systems work with the assumption that lower Logical Block Addresses (LBAs) are faster than higher LBAs,