Microprocessors and Microsystems 45 (2016) 283–296

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

On designing endurance aware erasure code for SSD-based storage systems

Saeideh Alinezhad Chamazcoti, Seyed Ghassem Miremadi*

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 4 February 2016
Revised 27 May 2016
Accepted 4 June 2016
Available online 14 June 2016

Keywords:
Solid state drive (SSD)
Erasure code
Endurance
Program/Erase (P/E) cycles
Storage system

ABSTRACT

Erasure codes are applied in both HDD and SSD storage systems to improve the reliability. The design of erasure codes for SSD-based systems should be performed with respect to a specific feature of SSDs, i.e., endurance. Endurance is defined as the number of Program/Erase (P/E)-cycles that one SSD can endure for reliable operation. The common metric for comparing the endurance of two systems is the number of P/E-cycles, which is yielded by time-consuming simulations. This paper proposes two new metrics called DPD-factor and GDP-pattern, for comparing the effect of erasure codes on the endurance of systems based on their encoding designs, without simulation. With respect to the endurance, EA-EO is designed as the modification of EVENODD with smaller DPD-factor. The endurance of EVENODD and EA-EO are compared regarding the system configurations: *the size of stripe unit, the number of disks*, and *the sizes of request*. The results of comparison show that the best configurations of system for enhanced endurance are: 1) a large number of disks are applied in systems, or 2) the size of request is equal to the stripe unit size. Furthermore, it concludes that a code with smaller DPD-factor and a sequential GDP-pattern can provide higher endurance for systems.

1. Introduction

Storage systems have been widely used for storing a large amount of data. *Solid State Disk* (SSD) and *Hard Disk Drive* (HDD) are two main technologies widely used in storage systems. In recent years, the use of SSD in several applications has been grown due to the advantages of SSD in comparison with HDD, such as higher performance and lower power consumption. As a drawback, SSDs suffer from limitation on the number of *Program/Erase* (P/E) cycles, called endurance. Endurance is the main concern of SSDs for deploying in future applications [1–5].

Erasure codes are the most common methods for protecting storage systems against failures. These codes protect an array of n data disks from m simultaneous disk failures, by using m parity disks ($m < n$). It is notable to mention that, erasure codes can be applied on the array of disks, no matter if it is HDD or SSD [5]. However, in HDDs, performance and reliability are two main concerns in the design of erasure codes. In SSDs, a third concern, i.e., the endurance plays an important role in the design of erasure codes. Several erasure codes have been proposed in literature to enhance the performance and reliability of storage systems [6–22]. To the best of our knowledge, endurance has not been the main concern in the design of erasure codes in the previous studies. The first study to consider the endurance in the design of erasure codes was presented in [23], which compared the effect of different erasure codes on the endurance of storage systems. The results of comparison in [23] showed that various erasure codes impose different endurance to the system due to their different encoding designs. The common metrics for evaluating the endurance in previous studies are the *Number of Writes* (NoW) and the *Number of Cleans* (NoC), which are yielded by running time consuming simulations. The first attempt on the design of erasure codes with the aim of improving endurance was presented in [24]. As two shortcomings of this attempt, only the traditional metrics i.e., NoW and NoC were used for analyzing the effect of proposed erasure code on the endurance of the storage system; and also the effects of system configurations on the endurance of systems was disregarded.

This paper extends the work previously presented in [24] to overcome the above mentioned shortcomings. The extension includes two contributions: 1) Two new metrics are proposed to compare the effects of erasure codes on the endurance of storage systems based on the code design without running the simulation. These new metrics are: a) *Data Parity Dependency* (DPD) factor, and *b) Grouping of Data and Parity* (GDP) pattern. The DPD factor indicates the dependency between data and parity units in the coding pattern of erasure codes. The GDP pattern indicates how a group of data units, which share the same parity unit, is placed in the array

---

* Corresponding author. Fax: +982166019246.
E-mail address: miremadi@sharif.edu (S.G. Miremadi).

http://dx.doi.org/10.1016/j.micpro.2016.06.003
0141-9331/

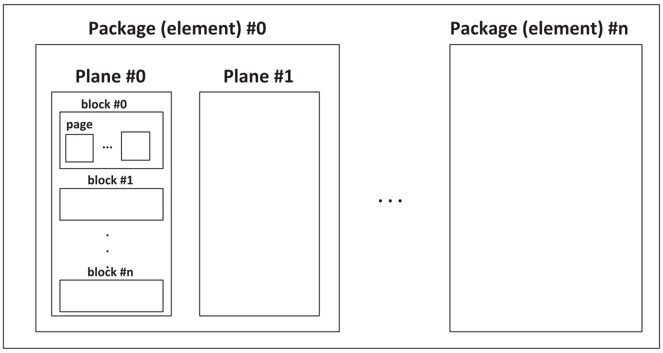

Fig. 1. A hierarchical structure of SSD [23].

of disks. 2) The effects of system configurations (i.e., *the number of disks in an array* and *the size of each stripe unit*), and also the effect of running applications (i.e., *the size of each request*) on the endurance of systems are investigated in this paper.

To evaluate the above mentioned contributions, a simulation environment is used to compare two erasure codes i.e., the EA-EO (Endurance Aware Erasure Code) and the EVENODD using four well-known real traces [25,26]. It is notable to mention that EA-EO is a modification of the EVENODD with smaller DPD factor. To show the improvement of the EA-EO as compared to the EVENODD, these two erasure codes are compared with respect to the system configurations as well as the running applications. The simulation results show that by selecting small DPD factor and sequential GDP pattern in the design of erasure codes, the endurance of storage systems would be improved. In addition, the results of simulation show that in two cases of simulation, the effect of DPD factor on the endurance of erasure codes is considerable: a) A large *Number of Disks* (NoD) is applied in an array, or b) the *Stripe Unit Size* (SUS) is equal to *the Size of Request* (SoR). With these cases, the EA-EO improves the endurance of system by 44% in comparison to EVENODD.

The rest of this paper is organized as follows: In Section 2, the background and related work of this study is stated. In Section 3, endurance aware erasure codes are introduced and the proposed EA-EO code is presented. The main factors in the design of endurance-aware erasure codes are discussed in Section 4. Section 5 reports the experimental results of evaluating the endurance of proposed erasure codes considering system configurations as well as the running applications. Finally, Section 6 presents the conclusions of this study and future work.

2. Backgrounds and related work

2.1. Solid state drive

Solid State Drives (SSDs) are kind of storing technologies which provide higher performance and lower power consumption compared to *Hard Disk Drives* (HDDs). As a main drawback of SSDs, they suffer from limitation on the number of P/E cycles. This limitation is an obstacle to employ SSDs for write-intensive applications, which imposes high number of writes to the system. The hierarchical structure of SSDs leads to specific characteristics for SSDs in I/O operations. As shown in Fig. 1, the hierarchical structure of SSDs is categorized as follows: each package of SSD consists of some planes, while each plane comprises of several blocks, and finally, each block contains of several pages. Read and write operations are performed in page unit, while erase operation is done in block unit. Before rewriting one page of SSD, the block of that page would be erased already, called “erase before write” property. Due to this property, one programmed page won’t be updated in the same physical place (out-of-place update). Instead of rewriting a page, any free pages of erased blocks could be selected for write operation with regard to the address mapping policy [1]. In order to hide the specific characteristics of SSDs from the host and file system, a translation layer called *File Translation Layer* (FTL) has been used in SSDs [5]. Three main tasks of FTL are: 1) address mapping, 2) wear-leveling, and 3) garbage collection.

The FTL is responsible for mapping the logical address to the physical address using proper mapping table. This layer also handles the write operations to distribute among the blocks of SSDs evenly (wear-leveling). In addition, the FTL provides free blocks in SSD by exerting cleaning policy to choose a victim block for erasing (garbage collection) [2,3].

Employing a group of SSDs in the array of disks (SSD-based RAID) is so common in recent decades to improve the performance, reliability, and capacity of system. The SSD-based RAIDs impose different challenges to the systems [5]. These challenges are classified in terms of performance and reliability, such as 1) input/output bottleneck, 2) alignment issue, 3) optimal stipe size, 4) RIAD implementation, and 5) asymmetric read and write, and so on. The more detail of each challenge was discussed in [5]. To address these challenges, some papers modified the SSD configuration of RAID structure with respect to the reliability issues [27–29].

SSDs are classified into two categories, i.e., *Single Level Cell* (SLC) and *Multi Level Cell* (MLC), according to the number of stored bits in each cell. The SLCs store one bit while MLCs store more than one bit per cell. MLCs provide higher capacity with lower cost, but endure smaller number of P/E cycles (smaller endurance) in comparison to the SLCs. The MLCs endure one order of magnitude lower number of P/E cycles in comparison to the SLCs. The MLCs are being increasingly employed in enterprise storage systems due to their advantages. Thus, the endurance limitation of MLCs is an important issue in recent researches [30,31].

2.2. Erasure code

Erasure codes are kind of *Error Correction Code* (ECC) which have been applied in storage systems to protect data against failures. This protection is done with minimum memory overhead as compared with other protection methods. In this coding, n data-units have been protected by m parity-units, to tolerate at most m failed units. There is a dependency between data and parity units in the coding pattern of erasure codes. Each parity unit is computed as a function of several data units, and is used for recovering failed units in decoding procedure. Different encoding procedures have been proposed for the erasure codes in literature with different coding patterns. It is obvious that, the effects of these erasure codes with different coding patterns on the characteristics of system (i.e., performance, endurance, reliability) are different.

Based on the operations used for encoding procedures of erasure codes, these codes are classified into two main classes: XOR-based erasure codes, and non-XOR based ones [6]. In XOR-based erasure codes, XOR is the base operation of computations (encoding and decoding procedure); while non-XOR based codes employ Galois field arithmetic in their computations which is complex and time-consuming. The XOR-based erasure codes provide faster recovery process (higher performance), and simpler implementation in comparison with non-XOR based codes. Several erasure codes have been proposed in literature for each of above mentioned classes. Such as Reed-Solomon [7], Liberation Code [9], and SD code [10] are examples of non-XOR based codes; and Cauchy Reed-Solomon [8], EVENODD [11], RDP [12], and P-Code [14] are examples of XOR-based ones.