Performance Evaluation 79 (2014) 235–257

Contents lists available at ScienceDirect

Performance Evaluation

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

Rectifying pitfalls in the performance evaluation of flash solid-state drives

Ilias Iliadis

IBM Research – Zurich, 8803 Rüschlikon, Switzerland

ARTICLE INFO

Article history:
Available online 11 July 2014

Keywords:
Write amplification
Stochastic modeling
Garbage collection
SSD
Log-structured systems

ABSTRACT

Today's data storage systems are increasingly adopting flash-based solid-state drives (SSD), in which new data is written out-of-place. The space occupied by invalidated data is reclaimed by means of a garbage-collection process. This involves additional write operations that result in write amplification, which negatively affects the performance, endurance, and lifetime of SSDs. Several garbage-collection schemes have been proposed in the literature and corresponding models have been developed for assessing their efficiency. We demonstrate that some of these publications arrive at conflicting results. We establish that the discrepancies identified are due to pitfalls in the modeling and analysis of some of the basic garbage-collection schemes. We effectively resolve these discrepancies by rectifying the pitfalls and developing proper analytical models that yield accurate results. We obtain new results for the circular scheme that are subsequently used to develop a new accurate model for the windowed greedy garbage-collection scheme. Results of theoretical and practical importance are analytically derived and experimentally confirmed. They demonstrate that, as the window size decreases, the write amplification increases, illustrating the tradeoff between computation time and write amplification. The results also show that, under certain conditions, the write amplification of the simple circular scheme can be similar to that of the optimum, albeit more complex greedy garbage-collection scheme. In this case, the write amplification for the windowed greedy scheme is essentially found to be independent of the window size.

1. Introduction

The data storage industry has increasingly been adopting non-volatile NAND-flash memories, such as solid-state drives (SSD), owing to their superior random I/O performance and access latency compared with those of hard-disk drives [1,2]. Also, these memories have been widely deployed in portable devices because of their good shock resistance and power consumption characteristics. Data is read in pages and written in *clean* pages, which then become *valid*. Written pages become clean again when *erased*, but they can only be erased in *blocks*, which contain a fixed number of pages. Typically, a block contains 64 or 128 pages, with the page size equal to 2 or 4 KB. For efficiency reasons, SSDs use out-of-place writes in a *log* fashion, in that data is updated by writing it in clean pages, without overwriting the pages in which data is currently stored. Thus, unlike disk drives, data is not updated in-place, and as a result the pages that store the old data become *invalid*. As the supply of clean pages is gradually exhausted, the creation of new clean pages becomes necessary for subsequent write operations. This task is performed by means of a *garbage-collection (GC)* process in a similar manner as in log-structured file

E-mail address: ili@zurich.ibm.com.

http://dx.doi.org/10.1016/j.peva.2014.07.015
0166-5316/

systems, disks, and arrays [3–5]. The GC process periodically performs block cleaning by first identifying a number of blocks for erasing and then relocating their valid pages to clean pages. Finally the blocks are erased so that their pages become clean and available for rewriting [6,7].

The GC mechanism identifies blocks for cleaning according to a given policy. Its performance depends on the computation time required by the policy for identifying such blocks and also on the amount of valid data contained in these blocks, which is subsequently copied (relocated) to other clean blocks. These additional writes are referred to as "write amplification" and quantified by the ratio of the total number of writes to the number of externally requested writes. The extent of the additional read and write operations required depends on the specific policy deployed as well as on the system parameters. However, the system performance is primarily affected by the write operations because the times required to write data are much longer than those to read data. Also, the number of erase/write operations that can be performed before an SSD wears out is limited. For single-level cell (SLC) NAND flash memories, this is typically 100,000 erasures per block, whereas for multi-level cell (MLC) it is reduced to 10,000 cycles [7]. Consequently, the extent of these additional writes, and therefore of the write amplification, should be kept low to prolong the endurance and lifetime of SSDs. Moreover, a low write amplification reduces the frequency of GC operations, which in turn results in reduced latency and thus improved performance. Therefore, a GC mechanism is efficient when its corresponding computation time and write amplification are as low as possible, and also if blocks are worn out as evenly as possible. To achieve these goals, various GC policies have been proposed in the literature, such as the "circular", "greedy", "cost-benefit" [3], "age-threshold" [5], "windowed greedy" [6] and "random" [8] policies. We have implemented two specific policies on a real SSD prototype built on an FPGA evaluation board: the circular (FIFO) policy, for which the computation time is minimal, and the windowed greedy policy, which attempts to minimize both the computation time and the write amplification. According to the theoretical prediction in [6], for a uniform random write workload the write amplification for the windowed greedy policy should have been much smaller than that for the circular one, but the experimental results obtained did not confirm this. For instance, in one of the cases discussed in Section 4.3.1, the experimental results yielded write amplifications for the circular and windowed greedy policies equal to 2.7 and 2.6, respectively, whereas according to the theoretical prediction the latter should have been equal to 2, which is off by 23%.

So we are left with a conundrum. Does this discrepancy arise owing to an oversight in the prototype implementation or in the theoretical analysis? In our effort to clarify this issue, we have studied other relevant publications and found that this discrepancy was also reported in [8,9], but the cause of the inaccuracy could not be determined. Thus, we continued studying previous works and we ended up reviewing all the models presented in the literature for assessing the write amplification associated with the various GC mechanisms. Our investigation further complicated matters, as we have found additional publications ranging from older ones dating decades back to very recent ones that arrive at conflicting results. As we will show, this is due to pitfalls encountered in the models developed in some of the publications considered, including publications examining the greedy, windowed greedy, and random policies. As new GC policies are proposed and compared with previous ones, (as e.g. in [6,8–10]), it is imperative to rectify these pitfalls so that research on SSD performance can advance on a solid ground.

The key contributions of this paper are the following. We review in great detail all the models presented in the literature for assessing the write amplification associated with the various GC mechanisms and identify several pitfalls. Addressing the weaknesses discovered in the models allows us to effectively resolve the preceding discrepancies and ensure that fallacies do not propagate in future works regarding the operation of the GC mechanisms and their performance. In this article, not only do we identify the pitfalls, but we also rectify them by obtaining novel results for the circular policy that are subsequently used to develop a new model that accurately captures the performance of the windowed greedy policy. The efficiency of SSDs is analytically assessed for the key system parameters, namely, the proportion of the memory space occupied by valid user data, the block size in terms of number of pages, and the window size. Results of theoretical and practical importance are analytically derived and confirmed using simulation. They illustrate the tradeoff between computation time and write amplification under a uniform random write workload. In this case, as the write updates are uniformly distributed over the entire address space, blocks get equal chances of being selected and erased by the GC policies considered, which in turn yields a good degree of wear leveling. For this reason, we focus our attention on the cleaning cost metric expressed by the write amplification. Furthermore, to assess the potential effect of lower-level SSD artifacts, we also obtain empirical results using a prototype implementation. These results confirm that the model developed is realistic because it properly captures the principal aspects of SSD operation and hence yields accurate results, although it makes some simplifying assumptions and does not take into account the hidden details of an SSD hardware implementation. The usefulness of this model follows from the fact that to assess the performance of flash-memory storage systems, which contain several thousands of blocks, neither Markov chain models nor simulators can effectively be used owing to the state and memory explosion. Moreover, and very importantly, it provides useful insights into the behavior of the GC mechanisms and yields results for the entire parameter space, which allows a better understanding of the design tradeoffs.

The remainder of the paper is organized as follows. Section 2 reviews the analytical models presented in the literature for assessing write amplification, points out several discrepancies, and discusses workload issues. Section 3 provides relevant literature on flash design and garbage-collection issues, and describes the operation of the circular, greedy, and windowed greedy GC schemes. Section 4 presents a detailed analytical evaluation of these schemes and rectifies the pitfalls encountered in the models considered by previous publications. Section 5 presents numerical results demonstrating the efficiency of the windowed greedy GC scheme for the key system parameters. Finally, we conclude in Section 6.