# Offline and Online Algorithms for SSD Management

TOMER LANGE, JOSEPH (SEFFI) NAOR, and GALA YADGAR, CS Department, Technion, Israel

The abundance of system-level optimizations for reducing SSD write amplification, which are usually based on experimental evaluation, stands in contrast to the lack of theoretical algorithmic results in this problem domain. To bridge this gap, we explore the problem of reducing write amplification from an algorithmic perspective, considering it in both offline and online settings. In the offline setting, we present a near-optimal algorithm. In the online setting, we first consider algorithms that have no prior knowledge about the input. We present a worst case lower bound and show that the greedy algorithm is optimal in this setting. Then we design an online algorithm that uses predictions about the input. We show that when predictions are pretty accurate, our algorithm circumvents the above lower bound. We complement our theoretical findings with an empirical evaluation of our algorithms, comparing them with the state-of-the-art scheme. The results confirm that our algorithms exhibit an improved performance for a wide range of input traces.

## 1 INTRODUCTION

The use of flash based solid state drives (SSDs) has increased in recent years, due to their high performance and low energy consumption. In an SSD, space is partitioned into *blocks*, where a block often contains hundreds of *pages*. Data to be stored is written into pages, and a page can be reused only when it holds invalid data and the block containing the page is *erased*. A *garbage collection* process is responsible for reclaiming space by choosing a *victim* block, rewriting any valid pages still written on this block to new locations, and erasing it. Erasure is a costly operation, and repeated erasures gradually wear out the flash media. Thus, a common measure for the efficiency of an SSD management algorithm is the *write amplification* (WA)—the ratio between the total number of page writes in the system, including page rewrites, and the number of writes issued in the input sequence.

Mechanisms to reduce write amplification have been studied extensively by the systems community. Most mechanisms are based on *the greedy algorithm*, which waits until the SSD is filled, and then chooses the block with the minimal number of valid pages for erasure. Several mathematical models were proposed for deriving the write amplification of the greedy algorithm under various assumptions about the workload distribution [1, 2, 6]. The optimality of the greedy algorithm under uniform page accesses was proven in [9], but it is known that its efficiency deteriorates when the workload distributions are skewed.

Garbage collection is typically conceived of as a process of selecting the optimal block for erasure; however, efficient garbage collection is also a matter of deciding **where** to place incoming data. Advanced techniques that separate frequently written (*hot*) from rarely written (*cold*) pages have been proposed. The idea behind these techniques is that placing pages with similar write frequencies in the same block increases the likelihood that all pages within that block will be overwritten by writes with temporal proximity. Several methods for predicting page temperatures have proven useful in practice [5, 8].

A recent study [4] observes that ideally, pages that are about to be accessed at nearby times in the future should be placed in the same block. The time of the next access to a page is called its *death time*, and the above rule is called *grouping by death time*. When all the pages in a block become invalid at approximately the same time, write amplification can be minimized by choosing the block whose pages have all been invalidated as victim. In a sense, this is precisely what hot/cold data separation strives to achieve; however, hot/cold separation can be viewed as grouping by *lifetime*, while two pieces of data can have the same lifetime (i.e., hotness) but distant death times. While hot/cold data separation has become standard practice, grouping by death time is only now coming into the attention of the systems research community.

The abundance of system-level optimizations for reducing write amplification, which are usually based on experimental evaluation, stands in stark contrast to the lack of theoretical algorithmic results in this problem domain. We present the first theoretical analysis and systematic evaluation of algorithms that observe the rule of grouping by death time, laying the ground for the development of a new class of algorithms that the systems community has not yet considered. Our results may also motivate the theory community to address the algorithmic aspects of SSD-related design challenges. Further details can be found in [7].

## 2 CONTRIBUTION

We study the *SSD management problem*, in which we are given a *request sequence* $\sigma = (\sigma_1, \sigma_2, ..., \sigma_T)$ as an input. At any time $t$, the page accessed in $\sigma_t$ must be written immediately. The goal is to serve $\sigma$ while minimizing the write amplification. A key parameter used in our bounds is the *spare factor*, denoted by $\alpha$. This factor expresses the relation between the physical and the logical capacities of the device. We now present our main technical results.

### 2.1 Online Setting

We first study algorithms that have no prior knowledge about the input, such as the well-known greedy algorithm. We analyze the performance of the greedy algorithm and show that its write amplification is never greater than $1/\alpha$.

Intuitively, erasing a block with more invalid pages is preferable, as it requires fewer rewrites and frees more space for reuse. Indeed, we prove that no deterministic algorithm provides a guarantee better than that of the greedy algorithm. We also provide a lower bound of $1/2\alpha$ on the write amplification guaranteed by any randomized algorithm.

## 2.2 Offline Setting

We consider an offline setting in which the request sequence is known in advance. We propose a novel placement technique that uses information about the death times of the pages to improve the performance of the SSD. We apply this technique and design a near-optimal algorithm whose write amplification given any input is not greater than $1 + \varepsilon$. This result highlights the fact that death times are very useful for minimizing write amplification. Moreover, this is the first time that an offline algorithm for SSD management is analyzed, and from a systems perspective, having a near-optimal offline algorithm is very useful for benchmarking.

We also discuss the hardness of the offline problem. While it is not known whether this problem is solvable in polynomial time, we take a step in resolving this issue and link it to the problem of *fragmented coloring* on interval graphs, which has been previously studied in [3]. Even though the complexity of the latter problem remains open, we extend previous results by providing an efficient algorithm for the special case in which the number of colors is fixed.

## 2.3 Online Setting with Predictions

We consider a middle ground setting in which we assume the availability of an oracle that predicts the death time of each accessed page. We develop an online algorithm with improved performance based on both those predictions. We note that in order to apply our placement technique, an additional small, fast memory is required.

Mechanisms to predict death times tend to exhibit errors when deployed. Hence, the performance of our algorithm is given as a function of the oracle's error $\eta$. Specifically, we prove that for any input $\sigma$, our algorithm satisfies:

$$WA_{online}(\sigma, \eta) \leq \min\left(1 + O\left(\frac{\eta}{T}\right),\ \frac{1}{\alpha}\right) + \varepsilon.$$

Note that our algorithm exhibits a graceful degradation in performance as a function of the average prediction error. Furthermore, our algorithm is *robust*, that is, its performance is never worse than that guaranteed by the greedy algorithm, even when the prediction error is large. When predictions are perfect ($\eta = 0$), the write amplification is bounded by $1+\varepsilon$, similarly to the offline setting. However, $\varepsilon$ here has a larger value, since the online model is less informative.

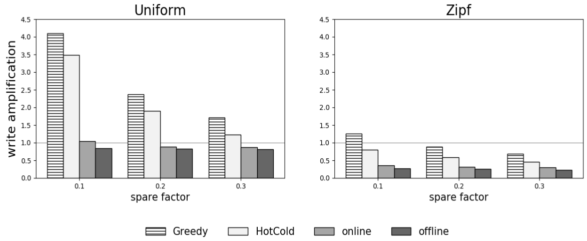

Fig. 1. The write amplification of Greedy, HotCold and our algorithms with varying spare factors.

## 2.4 Experimental Evaluation

We compared our algorithms to the greedy algorithm (*Greedy*) and to a state-of-the-art algorithm with hot/cold separation (*HotCold*). We used two trace types: synthetic and real. We pre-processed all traces in order to annotate each page request with its death time. To evaluate the efficiency of our algorithm, we also created erroneous versions of those annotations.

Figure 1 shows the write amplification of Greedy, HotCold and our algorithms on two synthetic traces with uniform and Zipf access distributions. All algorithms are equipped with an additional memory whose size is 1% of that of the entire SSD. The results confirm that our algorithms exhibit an improved performance for a wide range of input traces. The highest benefit is achieved for traces with modest or no skew, which are the worst-case inputs for Greedy and HotCold. Full descriptions of the algorithms, performance analyses and additional experimental results containing erroneous predictions can be found in [7].

## REFERENCES

[1] Werner Bux and Ilias Iliadis. 2010. Performance of Greedy Garbage Collection in Flash-Based Solid-State Drives. *Perform. Eval.* (2010).

[2] Peter Desnoyers. 2014. Analytic Models of SSD Write Performance. *ACM Trans. Storage* (2014).

[3] Ajit Diwan, Soumitra Pal, and Abhiram Ranade. 2015. Fragmented coloring of proper interval and split graphs. *Discrete Applied Mathematics* (2015).

[4] Jun He, Sudarsun Kannan, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau. 2017. The Unwritten Contract of Solid State Drives *(EuroSys)*.

[5] Jen-Wei Hsieh, Tei-Wei Kuo, and Li-Pin Chang. 2006. Efficient Identification of Hot Data for Flash Memory Storage Systems. *Transactions on Storage* (2006).

[6] Xiao-Yu Hu, Evangelos Eleftheriou, Robert Haas, Ilias Iliadis, and Roman Pletka. 2009. Write Amplification Analysis in Flash-Based Solid State Drives *(SYSTOR 2009)*.

[7] Tomer Lange, Joseph Naor, and Gala Yadgar. 2021. Offline and Online Algorithms for SSD Management. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* (2021). https://doi.org/10.1145/3491045

[8] Dongchul Park and David H.C. Du. 2011. Hot data identification for flash-based storage systems using multiple Bloom filters. In *MSST*.

[9] Yudong Yang, Vishal Misra, and Dan Rubenstein. 2015. On the Optimality of Greedy Garbage Collection for SSDs. *SIGMETRICS* (2015).