On the Endurance of the d-Choices Garbage Collection Algorithm for Flash-based SSDs

ROBIN VERSCHOREN and BENNY VAN HOUDT, University of Antwerp, Belgium

Garbage collection (GC) algorithms for flash-based solid state drives (SSDs) have a profound impact on its performance and many studies have focused on assessing the so-called write amplification of various GC algorithms. In this paper we consider the family of d-choices GC algorithms and study (a) the extent in which these algorithms induce unequal wear and (b) the manner in which they affect the life time of the drive. For this purpose we introduce two performance measures: the PE fairness and SSD endurance. We study the impact of the d-choices GC algorithm on both these measures under different workloads (uniform, synthetic and trace-based) when combined with two different write modes. Numerical results show that the more complex of the two write modes, which requires hot/cold data identification, may not necessarily give rise to a significantly better SSD endurance. Further the d-choices GC algorithm is often shown to strike a good balance between garbage collection and wear leveling for small d values (e.g., $d = 10$), yielding high endurance.

CCS Concepts: • **Computer systems organization → Secondary storage organization**;

Additional Key Words and Phrases: SSD, endurance, write amplification, wear leveling, garbage collection, hot/cold data identification, hot/cold data separation

ACM Reference Format:
Robin Verschoren and Benny Van Houdt. 2019. On the Endurance of the d-Choices Garbage Collection Algorithm for Flash-based SSDs. 1, 1 (April 2019), 24 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Data on flash-based solid state drives (SSDs) is organized in blocks that each consist of a fixed number of pages. While read and write operations are performed on a page level, erase operations can only be performed on a block level. To reduce the number of time consuming erase operations write operations are performed out-of-place, while the data in the old location is simply invalidated. The garbage collection (GC) algorithm is responsible for selecting the blocks that are used to write data. These *victim* blocks typically still contain some valid data that needs to be copied by the GC algorithm before the erase operation can take place, which gives rise to the so-called write amplification (WA).

In this paper, we consider the class of d-choices GC algorithms, which can be regarded as a combined GC and wear-leveling algorithm: it aims at selecting blocks with a low number of valid pages by selecting a block with the least number of valid pages out of a set of d randomly selected blocks, but therefore occasionally also ends up selecting a victim block with many cold valid pages (which becomes more likely for smaller d values). Hence, the d-choices algorithm aims at combining GC and wear-leveling without relying on any additional data structures such as erase counters commonly used by many wear-leveling algorithms [12]. It is possible to complement the system

Authors' address: Robin Verschoren; Benny Van Houdt, University of Antwerp, Dept. Math and Computer Science, Antwerp, Belgium.

XXXX-XXXX/2019/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

considered in this paper with an additional wear leveling mechanism such as Ban's algorithm [3, 12], which occasionally (say, every τ cleanings) picks a random victim block instead of using the GC algorithm. With Ban's algorithm blocks that contain many valid cold pages will also be cleaned eventually, which is key in achieving a more equal wear.

The main objective of this paper is to analyze to what extent the class of d-choices GC algorithms cause unequal wear and how they affect the life time of the drive. The contributions of this paper are as follows:

(1) We introduce two performance measures called the PE fairness and SSD endurance. The PE fairness indicates to what extent blocks undergo the same number of PE cycles during the life span of the drive. The SSD endurance measures the number of full drive writes that can be performed before any block reaches its maximum tolerable number of PE cycles. The SSD endurance is thus a combination of the WA and PE fairness.
(2) We study three types of workloads: uniform random writes, synthetic workloads with hot and cold data and trace-based workloads. The analysis for the uniform random writes case is based on an extension of the mean field model introduced in [29], while for the two other workload models we implemented the GC algorithms and write modes in the GC and FTL layers of the FlashSim simulator [17].
(3) We consider two modes of operation to support the out-of-place writes. The first simply separates the writes triggered by the host system and the GC algorithm, while the second relies on a hot/cold data identification technique and uses different blocks to write the hot and cold data.
(4) Some of the main insights provided include (a) that the mode based on the hot/cold data identification does not always provide a clear benefit in terms of the endurance over the other write mode and may even be inferior in some cases, (b) that lowering the WA may be more effective to prolong the life span of an SSD compared to further equalizing the wear and (c) the d-choices GC algorithm, for $d \leq 10$, seems to strike a good balance between garbage collection and wear leveling.

The remainder of the paper is structured as follows. Related works are discussed in Section 2. The properties of the SSD device considered in this paper are described in Section 3, while Section 4 introduces the PE fairness and SSD endurance performance measures. The mean field model and results for the uniform random writes setting are presented in Section 5. Section 6 focuses on synthetic workloads with hot and cold data. Trace-based workloads are the subject of Section 7. Finally conclusions are presented in Section 8 where we also discuss possible future work.

2 RELATED WORK

There has been a lot of work on assessing the WA caused by various GC algorithms, e.g., [5, 11, 15, 21, 24, 26, 29–31]. One popular approach to reduce the WA exists in implementing a hot and cold data identification technique (e.g.,[7, 8, 14, 25]). Such a technique attempts to identify the more frequently accessed logical pages, called the hot pages, and uses different blocks to store hot and cold pages (where any page that is not hot is termed cold). This reduces the WA as blocks packed with hot valid pages are invalidated more quickly and therefore tend to create blocks with a small number of valid pages.

Apart from achieving a low WA, flash-based SSD designers must also deal with the limited number of program-erase (PE) cycles that a block can tolerate [13, 20]. If a block is erased too often it can no longer retain its data for sufficiently long periods of time (as required by industry standards). One way to extend the life time of the drive exists in making sure that the number of

PE cycles performed on the drive is more or less evenly spread among its blocks, which is the main objective of a so-called wear leveling technique [6, 22].

A natural choice for the GC algorithm is the Greedy GC algorithm. This GC algorithm selects a victim block that currently holds the least number of valid pages among all blocks. In fact this algorithm is known to minimize the WA under uniform random writes [34] and its performance was shown to be close (within 5%) to an offline near-optimal GC algorithm for 10 MSR traces in case only a *single* block is used to write the new data [28]. However, in the presence of hot and cold data, the WA can be greatly reduced (by more than 50%) by relying on *multiple* blocks to store the new data (see [28, 30]). In this case the Greedy GC algorithm no longer minimizes the WA within the class of d-choices GC algorithms [30]. We therefore do not limit ourselves to the Greedy GC algorithm, but consider the more versatile class of d-choices GC algorithms. The WA of the class of d-choices GC algorithms considered in this paper was studied previously in [29–31]. Note that this class includes the Random and Greedy GC algorithm as special cases by setting $d = 1$ and $d = N$, respectively (where N is the number of blocks on the device).

3 SYSTEM DESCRIPTION

In this paper we focus on an SSD with a page-mapped FTL with a physical capacity of N blocks and user capacity of $U < N$ blocks, where each block holds exactly b pages. The spare factor S_f is defined as $(N - U)/N$ and the over-provisioning factor equals $N/U = 1/(1 - S_f)$.

We consider two write modes for the SSD device that both use two special blocks, called write frontiers (WFs), to support out-of-place writes. These two modes were previously considered in [31] and resemble other previously proposed write modes to reduce the WA, e.g., [8]. The main difference between the two modes exists in the fact that the second write mode requires that a hot/cold data identification technique is implemented.

Double write frontier (DWF). In this setup there is a WF for writes requested by the host, termed the external WF (WFE), and a WF for writes performed by GC, termed the internal WF (WFI). The objective of supporting these two WFs is to achieve a limited form of data separation without the need to implement a hot/cold data identification technique. Note that whenever we talk about a hot/cold data identification technique, we refer to a technique that explicitly marks a fraction of the pages as hot and this information is used when performing writes requested by the host.

The GC algorithm is invoked whenever the WFE becomes full. Assume that the last $b - j^*$ pages of the WFI are in the erase state when the GC algorithm is invoked, while the first j^* are in the valid/invalid state. Further assume the victim block selected by the GC algorithm contains j valid pages. Consider the following 2 cases:

(1) If $j \leq b - j^*$, the j valid pages of the victim block are simply copied to the WFI leaving the last $b - j^* - j$ pages in the erase state. After copying the j valid pages to the WFI, the victim block is erased and becomes the new WFE. Hence, the next b host writes make use of the WFE before the GC algorithm is invoked again.
(2) If $j > b - j^*$, $b - j^*$ of the j valid pages are copied to the WFI. The remaining $j - (b - j^*)$ valid pages are copied to RAM and back to the victim block after the victim block has been erased. In this case, the victim block becomes the new WFI and the GC algorithm is immediately invoked again.

Note that, in practice, copying any valid pages to RAM can be avoided by making use of a single free block [9]. This remark also applies to the HCWF write mode discussed next.

Hot/cold write frontier (HCWF). In this setup one block is labeled the hot write frontier (HWF) and another the cold write frontier (CWF) at all times. New hot (cold) data is sequentially written to

the HWF (CWF). All the remaining blocks are marked as either hot or cold at all times, depending on whether the block was last used as a HWF or CWF. The initial marking of the blocks (when the drive is empty) is irrelevant. If the HWF becomes full the GC algorithm is invoked and selects a victim block. Assume the victim block contains j valid pages, while the CWF has k pages in the erase state when the GC is invoked.

(1) If the victim block is marked hot, the j valid pages are copied back to the selected block after its pages have been erased and the selected block becomes the HWF.
(2) If the victim block was marked cold and $k \geq j$, the j valid pages are copied to the CWF and the selected block becomes the new HWF (which contains b pages in the erase state) and is labeled hot. Otherwise if $k < j$, k of the j valid pages are copied to the CWF, the remaining $j - k$ pages are written back to the victim block after its pages have been erased and the victim block becomes the new CWF.

In the latter case (i.e., the victim block is marked cold and $k < j$), the GC algorithm is immediately invoked again in search of a new HWF. Finally, when the CWF becomes full, instead of the HWF, the system operates as above if we exchange the terms hot and cold. A key feature of the HCWF setup is that it dynamically distributes the spare fraction S_f among the hot and cold data partition (formed by the hot/cold block markings) and blocks can move from one partition to the other.

Garbage collection algorithms. We focus on the set of d-choices GC algorithms [29–31], where $d \geq 1$ is an integer. Under d-choices GC the victim block is selected as follows: d blocks are chosen uniformly at random and the one containing the least number of valid pages among the d chosen blocks becomes the victim block (ties are broken arbitrarily). When $d = 1$ we obtain the Random GC algorithm, while setting d equal to the number of blocks on the SSD results in the Greedy GC algorithm [5, 11, 16]. Under uniform random writes the Greedy GC is known to minimize the write amplification [34], while in case of hot/cold data there exists an optimal finite d when minimizing the write amplification [30].

Note that the d-choices GC algorithm does not exploit any information that may be maintained by a potential wear leveling mechanism. Hence, the system under consideration does not rely on any explicit form of wear leveling. One of the questions we do intend to answer is how much room there is left for any wear leveling mechanism to further improve the endurance of the system. Note that as in Ban's wear leveling algorithm, the d-choices GC algorithm may select a block that contains only valid data (if all d randomly selected blocks happen to contain valid data only) and therefore clean it. Cleaning full blocks may appear to be useless, but it is necessary to allow such cleanings as otherwise blocks filled with read-only data are never cleaned (which causes very unequal wear).

Data retention. An important issue with respect to the life span of an SSD is the so-called data retention, which specifies the amount of time that a flash cell can correctly hold stored data. If a block is not rewritten after the retention time, its data can no longer be retrieved correctly. The retention time of a block also decreases as the number of PE cycles performed on it increases. For instance, in order to guarantee retention for as long as 3 years, the number of PE cycles on a block may be limited to as few as 3K [20]. The lifetime can be prolonged by relaxing the required retention time to several months or even a few days [19]. This however implies that (some of) the data must be internally refreshed in order to guarantee data integrity over long periods of time. The easiest way to achieve this is by triggering refreshes on all the blocks at a fixed refresh frequency to guarantee the retention time never falls below a predetermined threshold. These refreshes do cause additional PE cycles which reduces the life time improvement. More advanced retention

management schemes have been proposed that avoid some of these refresh operations by keeping track of the blocks that contain recently written data [20].

In this paper we assume that either the refresh overhead can be neglected or a very basic refresh management scheme is implemented that schedules periodic refresh operations on all the blocks. The refresh overhead can typically be neglected if either the number of writes per day is large (e.g., exceeds 10^8) or if the data retention time is large (e.g., a few months to several years) [20].

4 PERFORMANCE MEASURES

In this section we introduce the two main performance measures studied in this paper, for completeness we revisit the well-known write amplification (WA) first. The WA is equal to the ratio between the total number of writes performed on the drive divided by the number of writes requested by the host system. To be mathematically precise, let X_j be the random variable denoting the number of valid pages on the victim block selected during the j-th GC call, then the write amplification *up to the time of the n-th GC call* can be expressed as

$$WA(n) = \frac{bn}{\sum_{j=1}^{n}\sum_{i=0}^{b}(b-i)P[X_j = i]},$$

as selecting a victim block with i valid pages leaves room for $b - i$ writes by the host. Note that when talking about the WA one typically refers to $\lim_{n\to\infty} WA(n)$.

PE fairness. Let W_{max} represent the maximum number of PE cycles that a block can tolerate. The *PE fairness (PE_f)* is defined as the mean number of PE cycles performed on a block before any block reaches W_{max} PE cycles divided by W_{max}. More formally, let Y_k denote a random variable representing the number of times the GC algorithm is invoked before any block is erased for the k-th time, then the PE fairness is given by

$$PE_f(W_{max}) = \sum_{n\geq 1} P[Y_{W_{max}} = n]\frac{n/N}{W_{max}} = \frac{E[Y_{W_{max}}]}{W_{max}N},$$

as after n GC calls the mean number of PE cycles performed on a block part of a set of N blocks equals n/N. If the PE fairness is close to one, it is clear that there is little to no use in implementing a wear leveling technique. We believe this is an easier to interpret measure for the fairness than Jain's fairness index proposed in [18]. A somewhat more general fairness measure can also be defined by stating that Y_k is the number of invoked GC calls before a certain *fraction* of the blocks is erased at least W_{max} times, instead of just a single block. Experiments not shown here indicate that the insights presented in this paper carry over to such a setup as long as this fraction is small. Note that the PE fairness defined above remains meaningful in case periodic refresh operations are performed on the drive to guarantee data retention as these affect the number of PE cycles on all blocks in the same manner.

For simplicity we have assumed that the maximum number of PE cycles that a block can tolerate is a fixed number W_{max}. This assumption may not apply to very dense flash memory systems. For instance, recent measurements on the state-of-the-art Intel-Micron 32-tier 3D MLC NAND flash chips manufactured in 2016, indicate that the number of PE cycles that a block can tolerate fluctuates between 4000 and 6000 [33]. The authors point out that this is mostly due to the fact that the process control and the quality among pages for 3D NAND flash are imperfect and unstable compared with planar flash. As such this variability may reduce with the next generation of 3D NAND flash chips.

SSD endurance. The second measure of interest, termed the *SSD endurance (SSD_e)*, is a measure for the expected total number of host writes performed before any block reaches the predefined

maximum number W_{max} of PE cycles. Hence,

$$SSD_e(W_{max}) = \frac{E[\sum_{j=1}^{Y_{W_{max}}} \sum_{i=0}^{b}(b-i)P[X_j = i]]}{bN}.$$

Note that the unit used to express the SSD endurance is the total number of Full Drive Writes (FDWs). The SSD endurance is roughly equal to W_{max} times the PE fairness divided by the WA. While it is attractive to have a PE fairness close to one, the main reason for striving for high fairness exists in prolonging the life span of the drive, which is captured by the SSD endurance. Hence, having a PE fairness close to one is nice, but in the end the SSD endurance matters most.

We note that this definition of the endurance is the same as the one used in [4] for USB flash drives, which does not take NAND data refresh operations into account that may be needed to guarantee data retention [10, 20]. Relaxing the data retention time increases W_{max}, but also adds additional PE cycles due to the more frequent refresh operations. Although the above metric does not count these additional PE cycles, we expect that most of conclusions drawn in this paper remain valid if the refresh operations are performed periodically on *all* the blocks (which has the lowest implementation overhead).

5 UNIFORM RANDOM WRITES

In order to study the impact of a number of system parameters on the PE fairness and SSD endurance when subject to uniform random writes, we extend the mean field model of [29] that analyzed the write amplification only. The generalization exists in setting up drift equations for $m_{i,w}$, with $0 \leq i \leq b$ and $w \geq 0$, which represents the fraction of the total number of blocks holding i valid pages on which exactly w erase operations have been performed. Note that the model presented in [29] relies on a similar set of drift equations for m_i, which denotes the fraction of the total number of blocks holding i valid pages.

5.1 Mean field model

Let $p_{i,w}(\vec{m})$ be the probability that the GC algorithm selects a block with i valid pages that has been erased w times given that the occupancy vector equals $\vec{m}$. For instance, if we use the d-choices GC algorithm which does not take the number of PE cycles that occurred into account, we have

$$p_{i,w}(\vec{m}) = \frac{m_{i,w}}{m_i}\left[\left(\sum_{\ell=i}^{b}\sum_{w\geq 0} m_{\ell,w}\right)^d - \left(\sum_{\ell=i+1}^{b}\sum_{w\geq 0} m_{\ell,w}\right)^d\right]$$

for $m_i = \sum_{w\geq 0} m_{i,w} > 0$ and $p_{i,w}(\vec{m}) = 0$ for $m_i = 0$. Let $p_i(\vec{m}) = \sum_{w\geq 0} p_{i,w}(\vec{m})$.

Without going into detail, the set of ODEs for the generalized mean field model is given by $\frac{d}{dt}m_{i,w}(t) = f_{i,w}(\vec{m}(t))$, where

$$f_{i,w}(\vec{m}) = \frac{(i+1)m_{i+1,w} - i m_{i,w}}{b(1-S_f)}\left(\sum_{j=1}^{b} p_{b-j}(\vec{m})j\right) - p_{i,w}(\vec{m}). \tag{1}$$

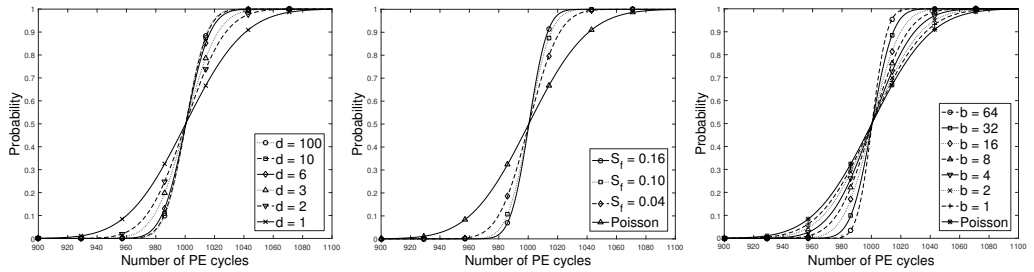
for $i < b$ and

$$f_{b,w}(\vec{m}) = \sum_{i=0}^{b} p_{i,w-1}(\vec{m}) - \frac{b m_{b,w}}{b(1-S_f)}\left(\sum_{j=1}^{b} p_{b-j}(\vec{m})j\right) - p_{b,w}(\vec{m}), \tag{2}$$

where $p_{i,-1}(\vec{m})$ is defined as 0. While the mean field model introduced in [29] considers a system using a single WF, it is not hard to see that the performance of such a system is identical to the DWF or HCWF setting of Section 3 in case of uniform random writes.

S_f	d	W_{max}	PE fairness simul	model	SSD endurance simul	model
0.10	10	500	.9351 ± .0012	.9387	98.6894 ± 0.1243	99.0881
0.10	10	1000	.9538 ± .0005	.9566	201.042 ± 0.3169	201.956
0.10	2	500	.8854 ± .0043	.8913	66.1325 ± 0.3240	66.5848
0.10	2	1000	.9210 ± .0027	.9225	137.577 ± 0.4124	137.833
0.06	100	500	.9244 ± .0023	.9283	67.1248 ± 0.1707	67.4176
0.06	100	1000	.9464 ± .0017	.9491	137.456 ± 0.2526	137.859

Table 1. Validation of the mean field model using simulation (averaged over 20 runs) for a system with N=10,000 blocks and $b = 32$ pages per block for various choices of S_f, d and W_{max}. Relative errors below 1% are observed in all cases.



(a) $b = 32$, $S_f = 0.1$. (b) $b = 32$, $d = 10$. (c) $d = 100$, $S_f = 0.1$.

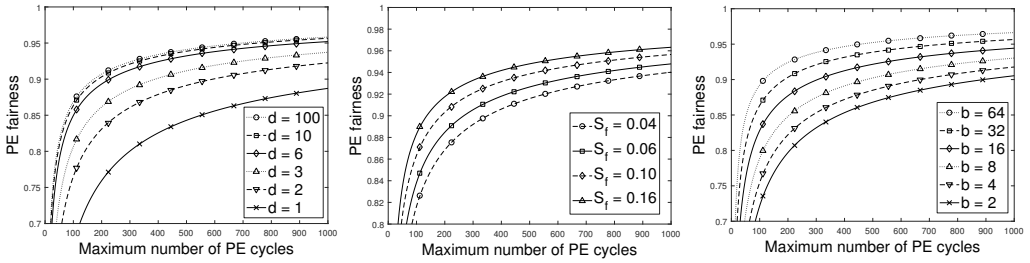
Fig. 1. The distribution of the number of PE cycles performed on a block under uniform random writes at time $t = 1000$.

Using simulation experiments we found that this model produces results with a relative error below 1% for both the PE fairness and SSD endurance for systems consisting of several thousand blocks (similar to [29] for the WA), a small arbitrary sample of the model validation results is presented in Table 1. Some details on the simulation program used are provided in Appendix A.

5.2 Numerical results

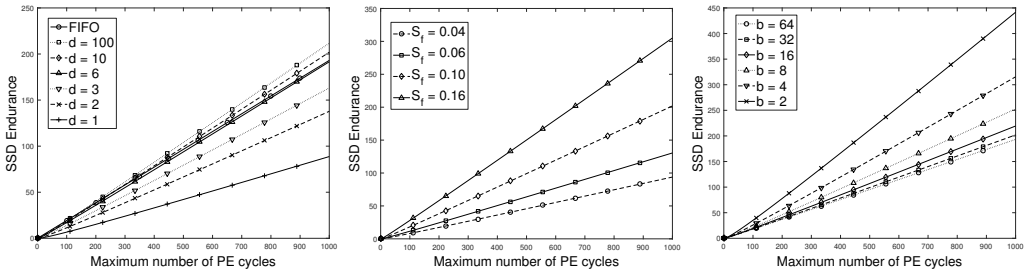
In this subsection we look at the impact of the different system parameters on the PE fairness and SSD endurance under uniform random writes. We start by looking at the distribution of the number of PE cycles.

Distribution of PE cycles: By relying on (1) and (2) we can determine the distribution of the number of PE cycles after Nt GC calls by numerically solving the ODE up to time t starting with $\sum_{i=0}^{b} m_{i,0}(0) = 1$. The results are depicted in Figure 1 and clearly indicate that the distribution of the number of PE cycles becomes less variable as the number of choices d increases, as the spare factor S_f increases and as the number of pages per block b increases. Thus, the Random GC algorithm (i.e., $d = 1$) performs the worst and the Greedy GC algorithm (i.e., d large) performs best both in terms of the write amplification and PE fairness. Note that when we state that the Greedy algorithm has the best PE fairness, we mean within the class of d-choices GC algorithms as the FIFO GC algorithm, which selects the victim block in a round robin manner, clearly has the best possible overall PE fairness (being $\approx (W_{max} - 1)/W_{max} = 1 - 1/W_{max} \approx 1$).



(a) $b = 32$, $S_f = 0.1$. (b) $b = 32$, $d = 10$. (c) $d = 100$, $S_f = 0.1$.

Fig. 2. PE fairness under uniform random writes ($N = 10^4$).



(a) $b = 32$, $S_f = 0.1$. (b) $b = 32$, $d = 10$. (c) $d = 10$, $S_f = 0.1$.

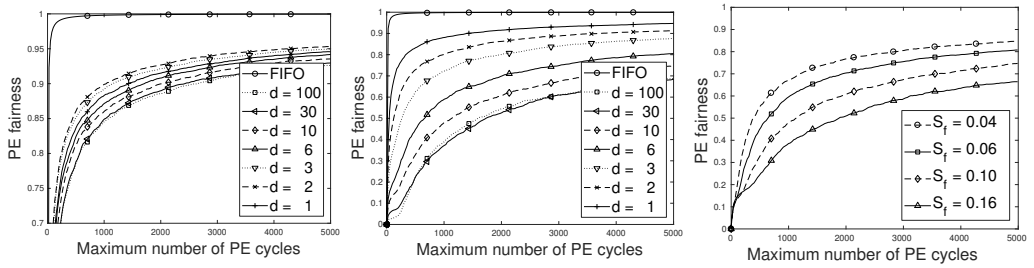
Fig. 3. SSD endurance under uniform random writes ($N = 10^4$).

When $d = 1$ it is easy to check that $m_w(t) = \sum_{i=0}^{b} m_{i,w}(t) = t^w e^{-t}/w!$. Hence, the distribution of the number of PE cycles on a block after Nt GC calls converges to a Poisson distribution with parameter t as N tends to infinity (as expected). This implies that the number of pages b per block as well as the spare factor S_f have no impact on the distribution of the number of PE cycles in case of the Random GC algorithm.

PE fairness: To determine the PE fairness via the set of ODEs specified by (1) and (2), we numerically solve the ODE starting with $\sum_{i=0}^{b} m_{i,0}(0) = 1$ up to time t_{max}, where t_{max} is the smallest t such that $\sum_{w \geq W_{max}} \sum_{i=0}^{b} m_{i,w}(t) > 1/N$. The PE fairness is found as t_{max}/W_{max}.

Figure 2 shows the PE fairness as a function of the maximum number of PE cycles W_{max} that a single block can tolerate. It indicates that increasing the number of choices d, number of pages per block b or the spare factor S_f results in an increase in the PE fairness. Also note that under uniform random writes one often observes a PE fairness above 0.95, even when the maximum number of PE cycles is as low as 1K. In other words, by the time that any block reaches 1K PE cycles the average number of PE cycles that an arbitrary block has endured is above 950. This implies that under uniform random writes there is hardly any room left to improve the PE fairness by implementing some form of wear leveling. This result is not unexpected, more surprisingly the PE fairness of the Random GC algorithm can still be improved substantially by the d-choices GC algorithm even by setting $d = 2$.

SSD endurance: Figure 3 depicts the SSD endurance in terms of the maximum number of PE cycles W_{max} that a single block can tolerate. The SSD endurance improves as the number of choices d or the spare factor S_f increases, which is in line with the previous results as larger d and S_f values



(a) $S_f = 0.1, r = 0.8, f = 0.2$. (b) $S_f = 0.1, r = 0.99, f = 0.01$. (c) $d = 10, r = 0.99, f = 0.01$.

Fig. 4. PE fairness under synthetic workloads with hot and cold data for $b = 32$ pages per block ($N = 10^4$) under the DWF mode of operation

improve the PE fairness and result in a lower write amplification. With respect to the impact of the number of pages b per block, we observe that lower b values result in a higher SSD endurance. The reason is that larger b values cause a higher write amplification, which outweighs the improvement in the PE fairness. Note that while the PE fairness is often above 0.9 the number of FDWs is well below W_{max} due to the (unavoidable) high WA under uniform random writes. Figure 3 also depicts the SSD endurance of the FIFO GC algorithm. While the FIFO GC has the best PE fairness, its SSD endurance is below that of the Greedy algorithm (i.e., d-choices with d large) due to its somewhat higher WA.

6 SYNTHETIC WORKLOADS WITH HOT AND COLD DATA

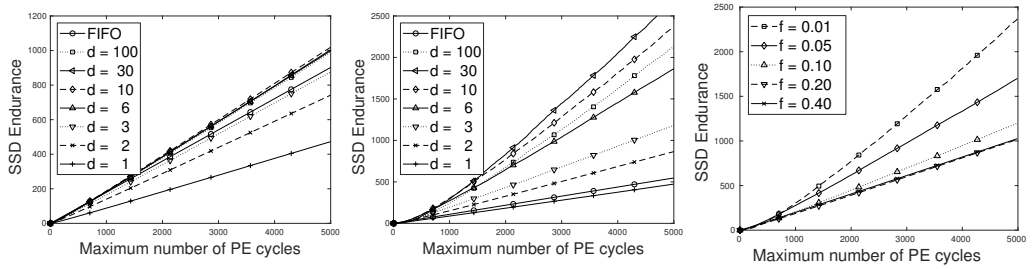
In order to be able to study the impact of data hotness in a structured manner, we focus for now on synthetic workloads of the Rosenblum type [27] and consider trace-based workloads in the next section. More specifically, in this section we assume we have two types of logical pages: hot and cold pages. A fraction f of the logical pages is hot and a write request updates a hot page with probability r and a cold page with probability $1 - r$. In other words a write request updates logical page x with probability $r/(Nf)$ if logical page x is hot and with probability $(1 - r)/(N(1 - f))$ if page x is cold.

Before discussing the results, note that (partially) separating hot and cold data has both a positive and negative impact on the SSD endurance. It is well known that data separation results in a lower WA (e.g., [11, 30, 31] for results under the Rosenblum model), but at the same time the PE fairness may worsen as the blocks holding (mostly) hot data may be subject to more PE cycles than blocks containing (mostly) cold data. Hence the main question is which of these two opposing forces dominates and to what extent does this depend on the GC algorithm and on the mode of operation (i.e., DWF versus HCWF).

In all of the experiments presented in this section, we have set the number of pages b per block equal to 32. In Appendix B.1 we show that very similar results are obtained when further increasing b to 64 or 128 as in some more recent flash architectures. A similar remark applies to the trace-based results presented in the next section.

6.1 Double write frontier mode

Although it is possible to extend the mean field model in [30] in a manner similar to the uniform random writes case, the computation times needed to numerically solve the set of ODEs becomes problematic and we therefore rely on simulation only (as stated before details on the simulation



(a) $r = 0.8$, $f = 0.2$. (b) $r = 0.99$, $f = 0.01$. (c) $d = 10, r = 1 - f$.

Fig. 5. SSD endurance under synthetic workloads with hot and cold data for $b = 32$ pages per block and spare factor $S_f = 0.1$ ($N = 10^4$) under the DWF mode of operation

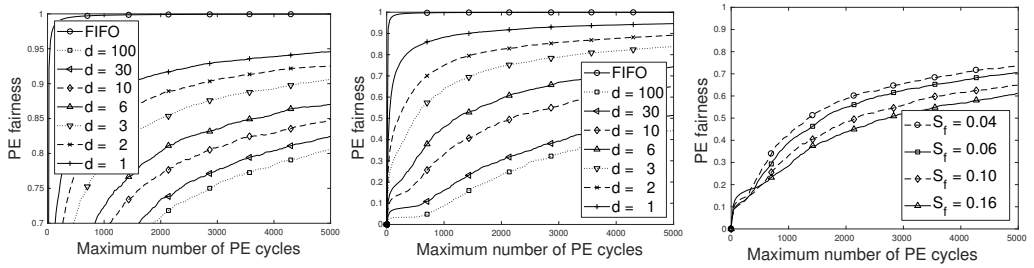
S_f	r	f	$d_{\text{MIN WA}}$	$d_{\text{MAX End}}$	$d_{\text{MAX Fair}}$
0.10	0.62	0.01	11	11	2
0.10	0.80	0.20	13	13	2
0.10	0.95	0.05	10	8	1
0.10	0.94	0.10	8	7	1
0.10	0.99	0.01	27	24	1
0.15	0.80	0.20	11	9	2
0.10	0.80	0.20	13	13	2
0.05	0.80	0.20	22	21	2

Table 2. Optimal d values to minimize the WA, maximize SSD endurance and maximize the PE fairness for various settings of S_f, r and f when $b = 32$. The optimal d value for the SSD endurance is somewhat lower than the optimal d to minimize the WA.

program are provided in Appendix A). All presented simulation results in this subsection are for a drive consisting of $N = 10$K blocks and are averaged over 50 runs.

PE fairness: Figure 4 depicts the impact of d and S_f on the PE fairness in the presence of hot and cold data. Figures 2a, 4a and 4b confirm that increasing data hotness leads to a lower PE fairness. The values for the PE fairness also indicate that in case of hot and cold data some form of wear leveling may help to prolong the SSD life span. These figures also show that while large d values gave rise to a better PE fairness under uniform random writes, the reverse happens in case of hot and cold data. This can be understood by noting that when the GC algorithm selects a new WFE, smaller d values increase the probability of selecting a block that previously stored (mostly) cold data. The WFE on the other hand mainly contains hot data when full (on average the WFE contains rb hot and $(1-r)b$ cold pages). Hence, for d small the hot data is less likely to circulate on the same set of blocks for a long period of time leading to a better PE fairness.

Figures 2b and 4c indicate that the impact of the spare factor S_f also changes when we introduce data hotness: with hot and cold data smaller spare factors result in a better PE fairness. This is probably due to the fact that the GC algorithm is less effective in selecting a victim block that previously stored mostly hot data when the spare factor is small.



(a) $S_f = 0.1, r = 0.8, f = 0.2$. (b) $S_f = 0.1, r = 0.99, f = 0.01$. (c) $d = 10, r = 0.99, f = 0.01$.

Fig. 6. PE fairness under synthetic workloads with hot and cold data for $b = 32$ pages per block ($N = 10^4$) under the HCWF mode of operation

SSD endurance: While it is desirable to have a PE fairness close to one, the main reason for striving for an equal wear lies in improving the SSD endurance. Figure 5 shows the SSD endurance for various d values and hotness values. We first note that while setting d small (e.g., $d \leq 3$) resulted in a higher PE fairness, this is typically not a good choice for the SSD endurance as the WA for very small d is much higher than for large d and this outweighs the better PE fairness. Further, as with the WA (see [30]) there is an optimal finite choice for d for the SSD endurance in case of hot and cold data. For instance, Table 2 shows the value of d that maximizes the SSD endurance is quite close to the d value that minimizes the WA (for various parameter settings).

More importantly, while Figure 4 showed that the PE fairness reduces significantly when the hot data becomes hotter, Figure 5c clearly indicates that making the hot data hotter is typically beneficial for the SSD endurance. This observation suggests that selecting a GC algorithm that minimizes the WA may lead to a much more profound improvement in the SSD endurance compared to selecting a GC algorithm that puts too much emphasis on the PE fairness, that is, on achieving a more or less equal wear on all blocks.

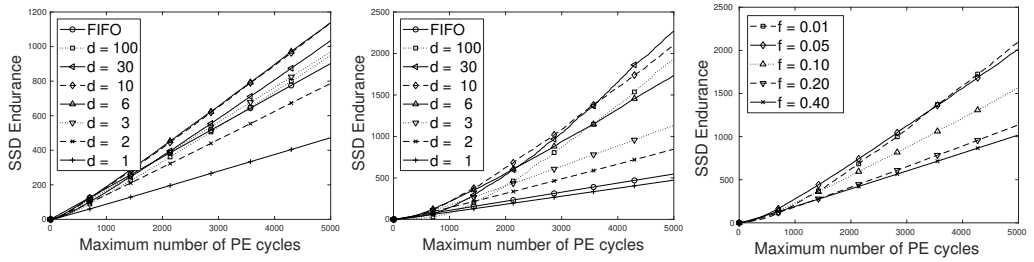
We do note that as the WA approaches one (it is 1.575 for $f = 0.01$ in Figure 5c) the importance of the PE fairness on the SSD endurance grows. Thus, GC algorithms that do take the wear into account may further increase the SSD endurance as long as the WA somehow kept equally low.

6.2 Hot/cold write frontier mode

Figures 6 and 7 depict the impact of various system parameters on the PE fairness and SSD endurance under a Rosenblum workload similar to Figures 4 and 5, but now for the HCWF mode of operation. The main trends with respect to the impact of the number of choices d, the hotness r and f, as well as the spare factor S_f are analogue to the DWF mode. A more intriguing question is how the performance of the DWF and HCWF modes compare.

Looking at Figures 4 and 6 it is clear that the DWF mode achieves a higher PE fairness than the HCWF mode, especially as d increases. Recall under the HCWF mode the hot and cold data is perfectly separated, while under the DWF mode the blocks still contain a mixture of hot and cold data. The more cold data that a block contains, the longer it takes to invalidate enough pages such that the GC algorithm selects this block as the victim block (unless d is small). Hence, a better separation of the hot and cold data tends to cause longer periods of unequal wear and is therefore expected to lower the PE fairness.

The better PE fairness of the DWF mode does not necessarily imply a better SSD endurance as the WA of the DWF mode is known to be worse compared to the HCWF mode under a Rosenblum workload [31, Figure 3]. By comparing Figures 5 and 7 we note that the HCWF has the highest SSD



(a) $r = 0.8$, $f = 0.2$. (b) $r = 0.99$, $f = 0.01$. (c) $d = 10, r = 1 - f$.

Fig. 7. SSD endurance under synthetic workloads with hot and cold data for $b = 32$ pages per block and spare factor $S_f = 0.1$ ($N = 10^4$) under the HCWF mode of operation

endurance when $r = 0.8$ and $f = 0.2$, while the opposite holds when $r = 0.99$ and $f = 0.01$. Thus when $r = 0.8$ and $f = 0.2$ the reduction in the WA obtained by replacing the DWF mode by the HCWF mode outweighs the decrease in the PE fairness, resulting in a better SSD endurance. The main reason for this is that while the DWF mode achieves a limited form of data separation, the reduction in the WA is less substantial compared to the case where there is a perfect hot/cold data separation. As the hot data becomes hotter (by increasing r and/or decreasing f) the difference between the WA achieved by the DWF and HCWF modes is expected to decrease as hotter data results in a better hot/cold data separation under the DWF mode (as confirmed by [31, Figure 3]). When $r = 0.99$ and $f = 0.01$ the reduction in the WA obtained by replacing the DWF mode by the HCWF mode is less substantial and no longer outweighs the decrease in the PE fairness, which causes the DWF mode to have a better SSD endurance.

This latter result is remarkable as it suggests that if the hot data is hot enough the DWF mode, which does not require any form of hot/cold data identification, achieves a better SSD endurance than the HCWF mode. The reason being that the somewhat worse WA of the DWF mode compared to the HCWF mode is compensated by a significantly better PE fairness. Further note that we are assuming here that the hot/cold data identification technique needed to implement the HCWF works perfectly, meaning there are no false positives/negatives. If we do incorporate such errors in the hot/cold data identification, the SSD endurance improvement achieved by the DWF mode would be even more pronounced as the endurance of the HCWF decreases in such case, as illustrated in Appendix B.2.

7 TRACE-BASED WORKLOADS

Under the Rosenblum model with parameters r and f, the fraction of hot pages is f by definition. For trace-based workloads it is far less obvious which fraction f of the pages should be labeled as hot. Most data identification techniques aim at identifying hot data [7, 14, 25], meaning similar to the Rosenblum model they partition the logical address space in two parts (although some schemes have been proposed that partition the space in more parts [8]). In this section we compare the PE fairness and SSD endurance of the DWF and HCWF mode using trace-based simulation experiments and consider various values for the fraction f of the logical pages that are marked hot.

I/O traces. We made use of the following I/O traces:

- **rsrch0** [1, 23]: an I/O trace collected at a server supporting research projects at Microsoft Research.

I/O trace	%Writes	#Requests	%ReadOnly
rsrch0 [23]	88.87	3,253,639	19.02
prxy0 [23]	96.36	22,136,692	19.53
webmail [32]	81.86	7,795,815	55.19
Booting [35]	23.31	245,792	79.05
Google Maps [35]	76.58	50,806	31.82
Installing [35]	99.58	416,556	00.48
Movie [35]	03.40	32,614	96.88
MusicFacebook [35]	63.32	113,553	53.23
MusicTwitter [35]	73.55	77,637	41.81

Table 3. Data set statistics. %Writes: percentage of writes, #Requests: number of requests and %ReadOnly: the fraction of accessed pages that is only read.

- **prxy0** [1, 23]: an I/O trace containing requests of a Firewall/web proxy server at Microsoft Research.
- **webmail** [2, 32]: an I/O trace of webmail traffic on a university department mail server.
- **Booting_176** [35]: an I/O trace of a smartphone launching.
- **Installing_181** [35]: an I/O trace collected during the installation of an application on a smartphone.
- **GoogleMaps_201** [35]: an I/O trace containing requests generated during road navigation while driving.
- **Movie_121** [35]: an I/O trace collected while watching a locally stored movie on a smartphone.
- **Music/FB_245** [35]: an I/O trace collected using the Facebook smartphone application, while listening to music.
- **Music/Twitter_225** [35]: an I/O trace generated by reading and posting tweets on Twitter, while listening to music.

We first preprocessed the traces by aligning the offset of each request to a multiple of 4 KB (all the offsets in the traces are multiples of 512 bytes). Requests with sizes above 4 KB (if present) were subsequently split into several (sequential) requests such that all requests have a size of at most 4 KB.

Some statistics on the trace files (after the preprocessing) are listed in Table 3. It lists the percentage of the requests that are write requests (%Writes), the number of requests in the trace (#Requests), and the percentage of the accessed logical address space that is only read (%ReadOnly). To make the simulation runs sufficiently long we adopted the *replay* method used in prior SSD work [18, 22]. By replaying a trace, we mean that the I/O pattern of the trace is repeated a number of times without change until one of the blocks has suffered W_{max} PE cycles. Table 4 provides information regarding the data locality of the *write* operations. More specifically, it lists the fraction of the write requests that access a logical page that is part of the fraction f of most frequently written logical page addresses, e.g., if 1% of the accessed pages is marked as hot, 61.92% of the writes are performed on hot data in the prxy0 trace.

SSD endurance. Figure 8 depicts the SSD endurance under trace-based workloads for $d = 10$ (the results for $d = 30$ and $d = 100$ choices are similar) and this for various hot data fractions f. We first note that among the hot fractions f considered, marking the 1% most popular logical pages as hot resulted in the best SSD endurance for the first three traces, while for the smartphone traces

	$f = 0.1$	$f = 0.05$	$f = 0.01$	$f = 0.005$	$f = 0.001$
prxy0	94.12%	77.09%	61.92%	60.02%	4.41%
webmail	69.89%	60.15%	48.96%	37.23%	6.77%
rsrch0	68.69%	61.07%	54.46%	51.83%	17.67%
Booting	19.50%	19.44%	15.88%	12.21%	8.77%
Google Maps	43.83%	39.70%	28.96%	25.34%	15.74%
Installing	26.88%	18.89%	9.23%	7.13%	4.63%
Movie	86.19%	85.47%	36.55%	22.02%	10.92%
MusicFacebook	57.83%	54.39%	47.21%	44.23%	38.24%
MusicTwitter	58.23%	54.27%	48.18%	45.68%	39.21%

Table 4. Data locality of the *writes*, e.g., when 1% of the accessed logical address space is marked as hot, 61.92% of the writes are performed on hot data in the prxy0 trace.

setting $f = 0.05$ is typically the best (among the choices considered). More importantly, we note that the DWF mode of operation achieves a better SSD endurance than the HCWF mode for some of the traces, e.g., the prxy0 and Movie trace. For other traces, such as the webmail and Booting trace, the DWF mode still has a better SSD endurance in case the fraction f is not appropriately set, while for the rsrch0 and Installing trace the HCWF mode is superior irrespective of the choice of f (among the ones considered). Thus, as with the synthetic workloads we see that the HCWF achieves a higher SSD endurance in some, but not all cases.

PE fairness. Recall that in case of the synthetic workloads considered before, the PE fairness of the DWF mode was superior to the HCWF mode, but its WA was inferior. This resulted in a better SSD endurance for the DWF mode in case the hot data was sufficiently hot such that the gain in PE fairness outweighs the increased WA. Figures 9 and 10 show that this observation does not extend to all the trace-based workloads: we see that the HCWF mode has a *better* PE fairness than the DWF mode in many cases, while the WA of the DWF mode is *lower* than the WA of the HCWF mode for the prxy0 and Movie workload.

There are a number of noteworthy observations with respect to Figure 9. First of all the fairness of the Installing workload is very high. Tables 3 and 4 indicate that this workload is very write intensive (more than 99% are writes and less than 0.5% is read-only) with rather limited data hotness, that is, this workload somewhat resembles the uniform writes workload considered by the mean field model of Section 5 which also yielded high fairness values (above 0.9). Second, for the Booting and Movie workloads we see very low fairness values when $d = 100$. For these two traces there is a large fraction of the data that is never updated (see Table 3). For such workloads these read-only pages are hard to move if d is set too large, causing a very unequal wear. By lowering d to 10 or below, the d-choices GC algorithm manages to get a much more equal wear (as the PE fairness exceeds 0.7 for $W_{max} = 5000$). This observation nicely illustrates that the d-choices GC algorithm (with moderate d values) acts as a hybrid garbage collection and wear leveling algorithm. Finally we also note that the PE fairness is influenced more heavily by the workload and less by whether we use the DWF or HCWF write mode.

There are probably several reasons why the comparison between the DWF and HCWF write modes is not as clear cut for the trace-based workloads as they were for the synthetic workloads. The intuition behind the synthetic workload results is that better separated hot and cold data yield a lower WA, but worse PE fairness as the hot data circulates longer on the same set of blocks. The difficulty with trace-based workloads is that there is no obvious definition of what better separated

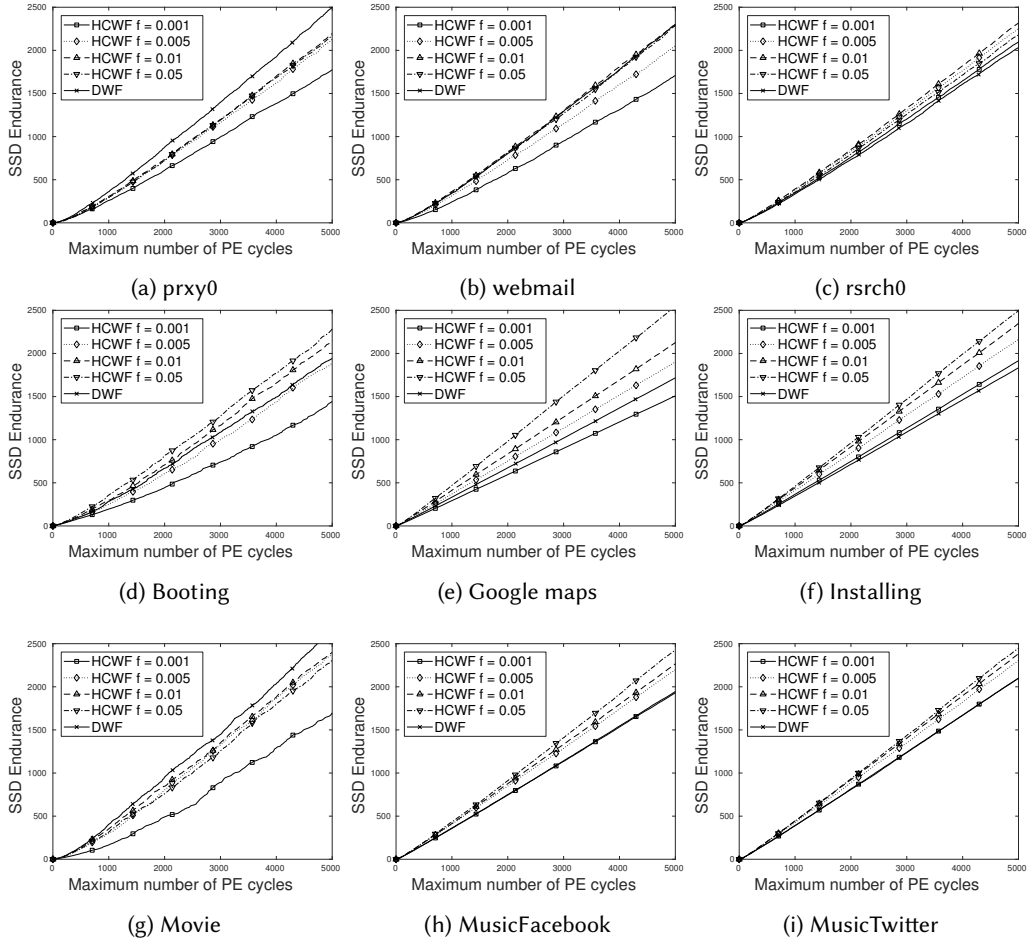
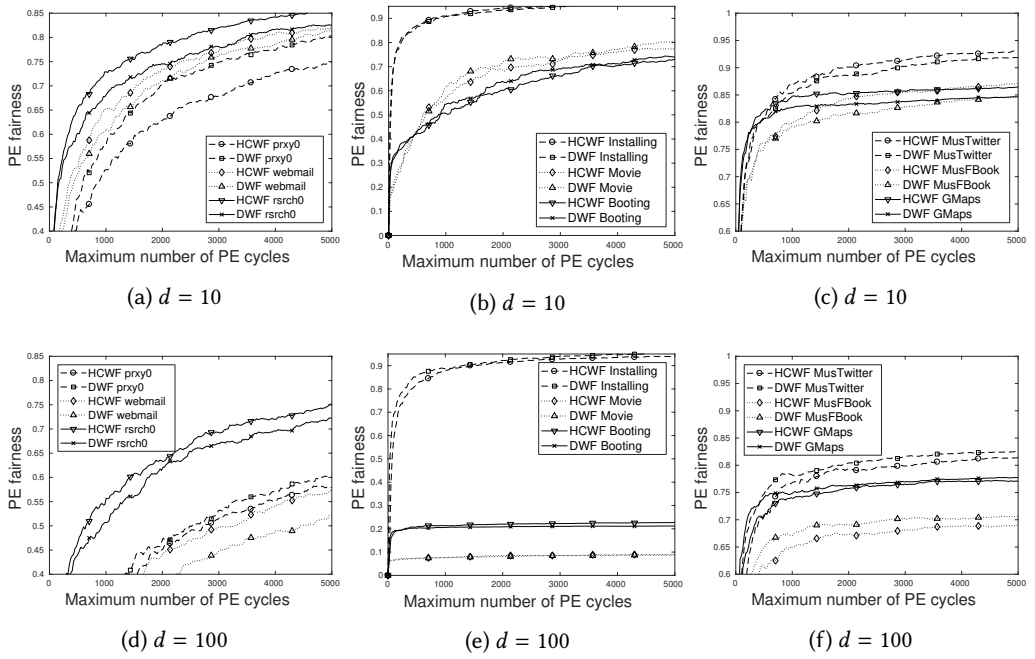


Fig. 8. SSD endurance for trace-based workloads with $b = 32$, $d = 10$ and $S_f = 0.1$ for various choices of the hot fraction f under the HCWF mode

hot/cold data means as there is no obvious unique way to select the hot and cold data. For instance, in our case we simply marked the most frequently accessed pages during the entire trace as hot, but even this is somewhat questionable as the popularity of some of the pages may fluctuate over time.

Despite the above, it is still clear that the more complex HCWF mode does not always offer a clear advantage over the DWF mode in terms of the SSD endurance and may even be inferior in some cases. We additionally note that the PE fairness values observed in Figure 9 are fairly high, especially for $d = 10$ where the PE fairness exceeds 0.7 in all cases. Keeping in mind that setting $W_{max} = 5K$ is probably somewhat conservative (especially if the data retention time is relaxed), this does not leave an awful lot of room for wear leveling techniques to further improve the SSD endurance. The higher PE fairness values for some of the trace-based workloads compared to the synthetic setting can be understood by noting that not all of the cold (hot) data is equally cold (hot), which implies that the time needed to invalidate a certain fraction of the pages on a cold or hot block differs less.



(a) $d = 10$ (b) $d = 10$ (c) $d = 10$

(d) $d = 100$ (e) $d = 100$ (f) $d = 100$

Fig. 9. PE fairness for trace-based workloads with $b = 32$ and $S_f = 0.1$ under the DWF and HCWF mode with $f = 0.01$.

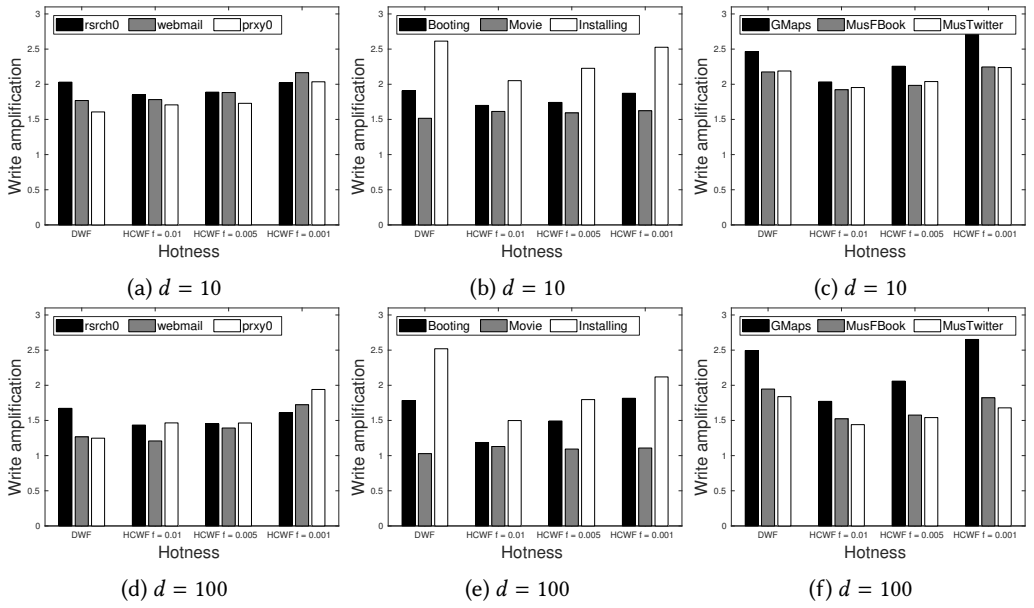
8 CONCLUSIONS AND FUTURE WORK

In this paper we introduced the PE fairness and SSD endurance performance measures and studied how these are affected by the d-choices GC algorithm under different workloads (uniform, synthetic and trace-based). We considered two different write modes: the DWF mode (which separates writes triggered by the host and GC algorithm) and the HCWF mode (which separates the hot and cold data). The following observations were made:

Under uniform random writes. The Greedy GC algorithm has a near optimal SSD endurance as it is known to be optimal with respect to the WA and has a PE fairness close to one. While the Random GC is often stated to have excellent wear leveling characteristics, its PE fairness can still be significantly improved by the d-choices GC algorithm with $d \geq 2$.

Under synthetic workloads with hot and cold data. The PE fairness may be well below one (especially for larger d), however a lower PE fairness often still yields a higher SSD endurance as the WA tends to have a more profound impact on the SSD endurance. The DWF mode gives rise to a better PE fairness than the HCWF mode, but suffers from a higher WA. If the hot data becomes sufficiently hot the DWF mode, which is much simpler to implement, achieves a higher SSD endurance.

Under trace-based workloads. The relative performance of the DWF and HCWF modes of operation is hard to predict as for some traces the DWF mode has a lower WA and/or higher PE fairness, but the same applies to the HCWF mode. The PE fairness is relatively high even with a conservative limit of 5K PE cycles (unless d is large and the workload contains a large fraction of read-only data),



(a) $d = 10$ (b) $d = 10$ (c) $d = 10$

(d) $d = 100$ (e) $d = 100$ (f) $d = 100$

Fig. 10. WA for trace-based workloads with $b = 32$ and $S_f = 0.1$ under the DWF and HCWF mode.

which bounds the margin to improve the SSD endurance by additional wear leveling techniques.

A general observation is that the d-choices GC algorithm often strikes a good balance between garbage collection and wear leveling if d is set sufficiently small (i.e., $d = 10$). Possible avenues for future work include developing GC algorithms that take the wear of blocks into account, incorporating wear leveling techniques, looking at the impact of the TRIM command and studying how the HCWF mode works when combined with specific hot/cold data identification techniques.

REFERENCES

[1] ftp://ftp.research.microsoft.com/pub/austind/msrc-io-traces/. MSRC-io-traces.

[2] http://sylab-srv.cs.fiu.edu/dokuwiki/doku.php?id=projects:srcmap:start. SyLab Energy Proportional Storage Systems Traces.

[3] Ban, A. Wear leveling of static areas in flash memory. US patent 6,732,221. Filed June 1, 2001; Issued May 4, 2004; Assigned to M-Systems., 2004.

[4] Boboila, S., and Desnoyers, P. Write endurance in flash drives: Measurements and analysis. In *Proceedings of the 8th USENIX Conference on File and Storage Technologies* (Berkeley, CA, USA, 2010), FAST'10, USENIX Association, pp. 9–9.

[5] Bux, W., and Iliadis, I. Performance of greedy garbage collection in flash-based solid-state drives. *Perform. Eval. 67*, 11 (Nov. 2010), 1172–1186.

[6] Chang, L.-P. On efficient wear leveling for large-scale flash-memory storage systems. In *Proceedings of the 2007 ACM Symposium on Applied Computing* (New York, NY, USA, 2007), SAC '07, ACM, pp. 1126–1130.

[7] Chang, L.-P., and Kuo, T.-W. An adaptive striping architecture for flash memory storage systems of embedded systems. In *Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'02)* (Washington, DC, USA, 2002), RTAS '02, IEEE Computer Society, pp. 187–.

[8] Chiang, M.-L., Lee, P. C. H., and Chang, R.-C. Using data clustering to improve cleaning performance for flash memory. *Softw. Pract. Exper. 29*, 3 (Mar. 1999), 267–290.

[9] Desnoyers, P. Analytic modeling of SSD write performance. In *Proceedings of International Systems and Storage Conference (SYSTOR 2012)* (2012).

[10] Desnoyers, P. What systems researchers need to know about NAND flash. In *Presented as part of the 5th USENIX Workshop on Hot Topics in Storage and File Systems* (Berkeley, CA, 2013), USENIX.

[11] Desnoyers, P. Analytic models of SSD write performance. *ACM Trans. Storage 10*, 2 (Mar. 2014), 8:1–8:25.

[12] Gal, E., and Toledo, S. Algorithms and data structures for flash memories. *ACM Computing Surveys 37* (2005), 138–163.

[13] Grupp, L. M., Davis, J. D., and Swanson, S. The bleak future of NAND flash memory. In *Proc. of USENIX Conference on File and Storage Technologies* (2012).

[14] Hsieh, J., Kuo, T., and Chang, L. Efficient identification of hot data for flash memory storage systems. *ACM Trans. on Storage 2* (2006), 22–40.

[15] Hu, X., Eleftheriou, E., Haas, R., Iliadis, I., and Pletka, R. Write amplification analysis in flash-based solid state drives. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference* (New York, NY, USA, 2009), SYSTOR '09, pp. 10:1–10:9.

[16] Iliadis, I. Rectifying pitfalls in the performance evaluation of flash solid-state drives. *Perform. Eval. 79* (2014), 235 – 257. Special Issue: Performance 2014.

[17] Kim, Y., Tauras, B., Gupta, A., and Urgaonkar, B. FlashSim: A simulator for NAND flash-based solid-state drives. In *Proceedings of the 2009 First International Conference on Advances in System Simulation* (Washington, DC, USA, 2009), SIMUL '09, IEEE Computer Society, pp. 125–131.

[18] Li, Y., Lee, P., and Lui, J. Stochastic modeling of large-scale solid-state storage systems: Analysis, design tradeoffs and optimization. *ACM SIGMETRICS Perform. Eval. Rev. 41*, 1 (2013), 179–190.

[19] Liu, R., Yang, C., and Wu, W. Optimizing NAND flash-based ssds via retention relaxation. In *Proceedings of the 10th USENIX conference on File and Storage Technologies, FAST 2012, San Jose, CA, USA, February 14-17, 2012* (2012), p. 11.

[20] Luo, Y., Cai, Y., Ghose, S., Choi, J., and Mutlu, O. Warm: Improving nand flash memory lifetime with write-hotness aware retention management. In *MSST* (2015), IEEE Computer Society, pp. 1–14.

[21] Menon, J. A performance comparison of RAID-5 and log-structured arrays. In *Proceedings of the 4th IEEE International Symposium on High Performance Distributed Computing* (Washington, DC, USA, 1995), HPDC '95, pp. 167–178.

[22] Murugan, M., and Du., D. Rejuvenator: A static wear leveling algorithm for NAND flash memory with minimized overhead. In *Proc. of IEEE MSST* (2011).

[23] Narayanan, D., Donnelly, A., and Rowstron, A. Write off-loading: Practical power management for enterprise storage. *Trans. Storage 4*, 3 (Nov. 2008), 10:1–10:23.

[24] Odeh, S., and Cassuto, Y. NAND flash architectures reducing write amplification through multi-write codes. In *IEEE 30th Symposium on Mass Storage Systems and Technologies, MSST 2014, Santa Clara, CA, USA, June 2-6, 2014* (2014), pp. 1–10.

[25] Park, D., and Du, D. Poster: Hot data identification for flash memory using multiple bloom filters. In *Proc. of USENIX Conference on File and Storage Technologies* (2011).

[26] Robinson, J. Analysis of steady-state segment storage utilizations in a log-structured file system with least-utilized segment cleaning. *SIGOPS Oper. Syst. Rev. 30*, 4 (Oct. 1996), 29–32.

[27] Rosenblum, M., and Ousterhout, J. K. The design and implementation of a log-structured file system. *ACM Trans. Comput. Syst. 10*, 1 (Feb. 1992), 26–52.

[28] Shafaei, M., and Desnoyers, P. Near-optimal offline cleaning for flash-based SSDs, 2017. http://storageconference.us/2017/Papers/CleaningFlashBasedSSDs.pdf.

[29] Van Houdt, B. A mean field model for a class of garbage collection algorithms in flash-based solid state drives. *ACM SIGMETRICS Perform. Eval. Rev. 41*, 1 (2013), 191–202.

[30] Van Houdt, B. Performance of garbage collection algorithms for flash-based solid state drives with hot/cold data. *Perform. Eval. 70*, 10 (2013), 692–703.

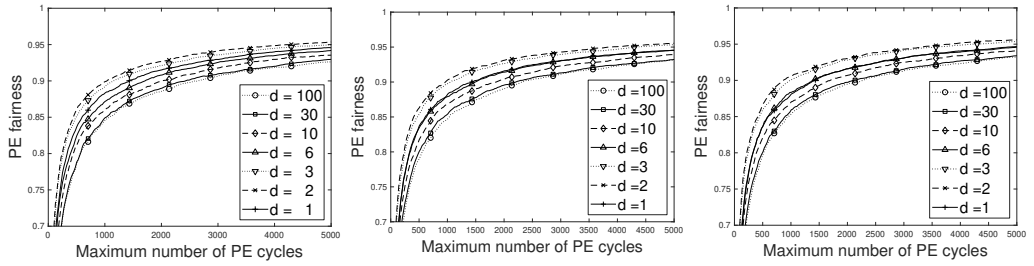
[31] Van Houdt, B. On the necessity of hot and cold data identification to reduce the write amplification in flash-based SSDs. *Perform. Eval. 82* (2014), 1 – 14.

[32] Verma, A., Koller, R., Useche, L., and Rangaswami, R. SRCMap: energy proportional storage using dynamic consolidation. In *Proceedings of the 8th USENIX conference on File and storage technologies* (Berkeley, CA, USA, 2010), FAST'10, pp. 267–280.

[33] Xiong, Q., Wu, F., Lu, Z., Zhu, Y., Zhou, Y., Chu, Y., Xie, C., and Huang, P. Characterizing 3D floating gate NAND flash. *SIGMETRICS Perform. Eval. Rev. 44*, 1 (June 2017), 31–32.

[34] Yang, Y., Misra, V., and Rubenstein, D. On the optimality of greedy garbage collection for SSDs. *ACM SIGMETRICS Perform. Eval. Rev. 43*, 2 (Sept. 2015), 63–65.

[35] Zhou, D., Pan, W., Wang, W., and Xie, T. I/O characteristics of smartphone applications and their implications for eMMC design. In *Workload Characterization (IISWC), 2015 IEEE International Symposium on* (2015), IEEE, pp. 12–21.



(a) $b = 32$ (b) $b = 64$ (c) $b = 128$

Fig. 11. PE fairness under synthetic workloads with b pages per block ($N = 10^4$), $S_f = 0.1$ and $r = 0.8$, $f = 0.2$ under the DWF mode.

A SIMULATION SETUP

The simulation experiments were performed using an extension of the FlashSim simulator [17]. More specifically, we relied on the extended FlashSim implementation of Matias Bjørling[1]. The DWF and HCWF FTL write modes were implemented in the FTL layer of the simulator[2]. The FTL layer also keeps track of the write frontiers and hotness of each block. Furthermore, we implemented a new layer within the simulator for hot/cold identification techniques, which communicates with the selected FTL.

The drive was initialized as follows. For each of the available logical page numbers (1 to $\lceil (1 - S_f)bN \rceil$ when using the synthetic workloads and 1 to the number of unique requested logical page numbers when using trace-based workloads), a corresponding physical page number was selected in a uniform random fashion. Care was taken that data corresponding to hot logical page numbers (i.e. logical page numbers 1 to $\lceil f(1 - S_f)bN \rceil$) were written to one of the $\lceil fN \rceil$ blocks designated as hot when using the HCWF setup. Experiments not presented in this paper have shown that this approach to initialize the drive generated results similar to the approach of starting with an empty drive that is filled during a warm-up period.

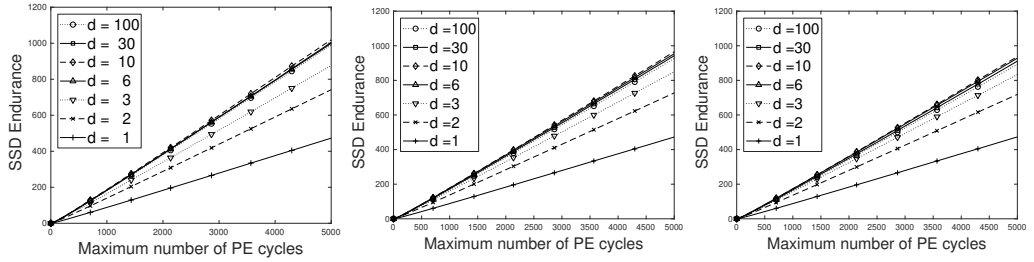
Each write request updates the data in the FTL and the simulated blocks and pages. Whenever one of the WF blocks is full (according to the chosen setup), the GC algorithm is invoked and a victim block is selected and erased. The contents of the victim block are copied according to the setup, and the drive and FTL is updated. The number of (internal) writes performed is maintained and used to compute the write amplification factor. A simulation run ends whenever the first block on the device reached the maximum number of PE cycles W_{max}. We did not implement a specific hot/cold identification technique, but assume one is in place that marks a page as hot if a page belongs to the fraction f of the most frequently accessed logical pages.

B SENSITIVITY ANALYSIS

B.1 Impact of block size

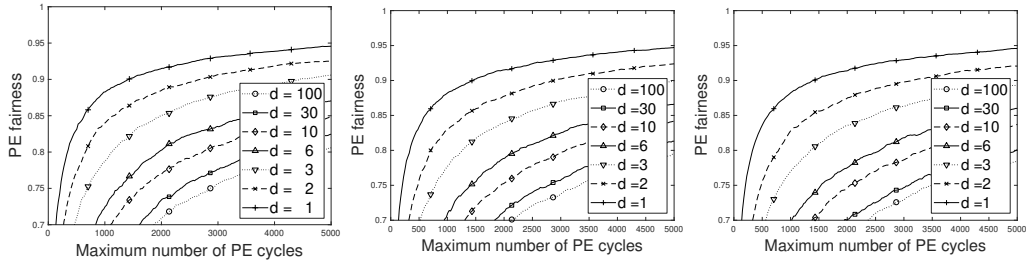
Each block on the flash memory device in Sections 6 and 7 was assumed to contain $b = 32$ pages. More recently, the number of pages on each block fluctuates and larger block sizes are used [10]. In this subsection we look at the impact of the block size on the PE fairness and SSD endurance for both synthetic and trace-based workloads.

[1] Available online at https://github.com/MatiasBjorling/flashsim
[2] Available online at https://github.com/RVerschoren/flashsim-dchoices



(a) $b = 32$ (b) $b = 64$ (c) $b = 128$

Fig. 12. SSD endurance under synthetic workloads with b pages per block ($N = 10^4$), $S_f = 0.1$ and $r = 0.8$, $f = 0.2$ under the DWF mode.



(a) $b = 32$ (b) $b = 64$ (c) $b = 128$

Fig. 13. PE fairness under synthetic workloads with b pages per block ($N = 10^4$), $S_f = 0.1$ and $r = 0.8$, $f = 0.2$ under the HCWF mode.

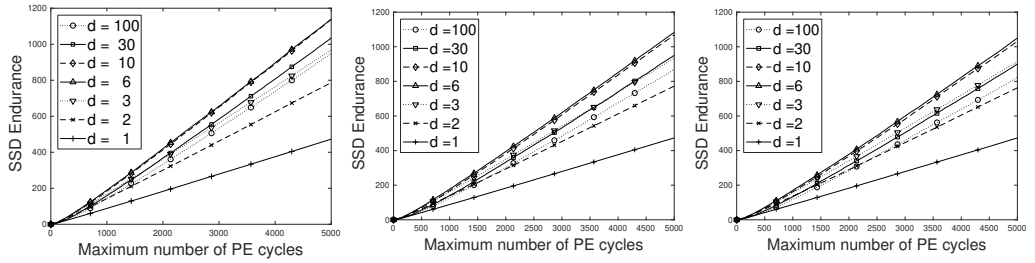
Synthetic workloads

Figures 11 and 12 illustrate the impact of increasing b on the PE fairness and endurance for a Rosenblum workload with $f = 0.2$ and $r = 0.8$ in case of the DWF write mode. These plots indicate that while increasing b has some minor impact on the fairness and endurance values, the curves and their relative order look very much alike for different b values. The minor increase in the PE fairness observed doubling b is intuitively clear as the number of blocks N in our experiment is fixed and therefore more writes are needed before one block reaches its maximum number of PE cycles. Nevertheless, the endurance decreases for larger b as increasing b is known to result in a higher write amplification which outweighs the minor improvement in the PE fairness. The impact of increasing b in case of the HCWF write mode has a similar effect as shown in Figures 13 and 14.

Very similar results (not shown here) were also obtained using other combinations of r and f (e.g., $f = 0.01$ and $r = 0.62$ or $f = 0.1$ and $r = 0.94$).

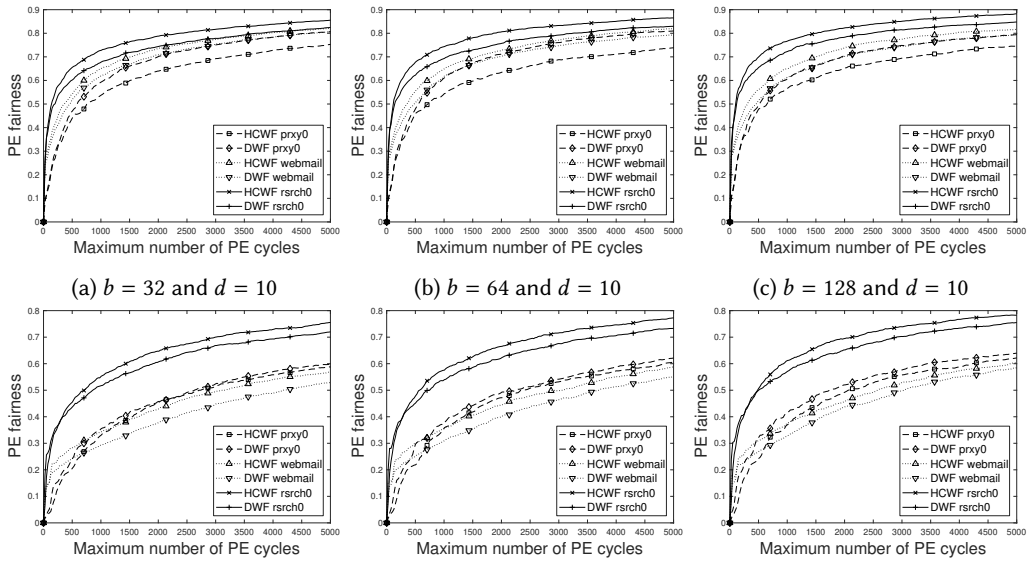
Trace-based workloads

Figures 15 and 16 show the PE fairness and endurance for the prxy0, rsrch0 and webmail workloads for $d = 10$ and $d = 100$. These results are very much in line with our observations in case of synthetic workloads. More specifically, increasing b beyond 32 slightly improves the PE fairness, but decreases the endurance due to the increase in write amplification. We do note that the decrease in the endurance appears to be the most pronounced in case of the HCWF mode and d large.



(a) $b = 32$ (b) $b = 64$ (c) $b = 128$

Fig. 14. SSD endurance under synthetic workloads with b pages per block ($N = 10^4$), $S_f = 0.1$ and $r = 0.8$, $f = 0.2$ under the HCWF mode.



(a) $b = 32$ and $d = 10$ (b) $b = 64$ and $d = 10$ (c) $b = 128$ and $d = 10$

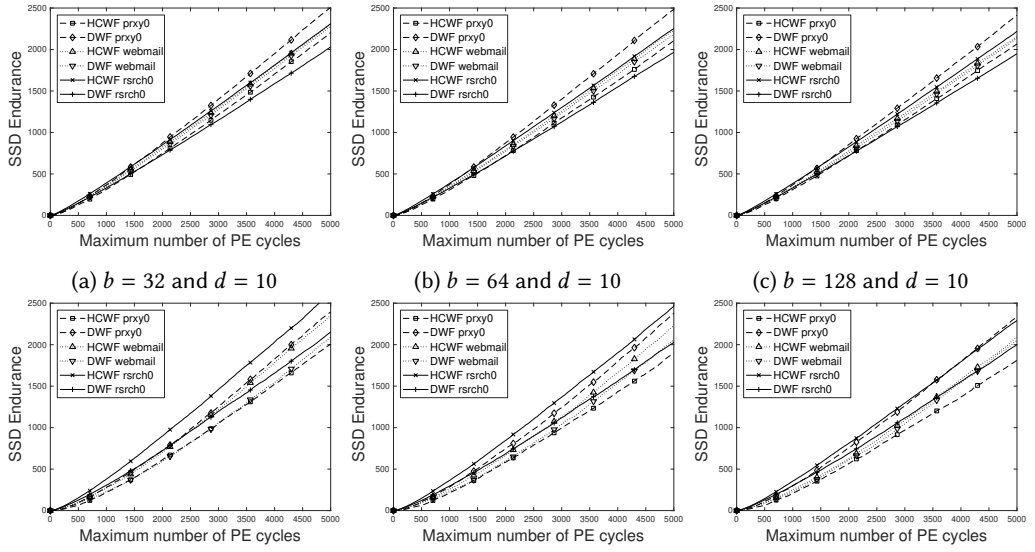
(d) $b = 32$ and $d = 100$ (e) $b = 64$ and $d = 100$ (f) $b = 128$ and $d = 100$

Fig. 15. PE fairness for trace-based workloads with $S_f = 0.1$ under the DWF and HCWF mode with $f = 0.01$.

B.2 Hot/cold data identification

In all of our experiments with the HCWF mode we assumed that the hot/cold data identification mechanism is perfect. In this section we relax this assumption. We focus on the setting with a synthetic Rosenblum workload and expect similar results for trace-based workloads. Instead of studying the impact of specific data identification mechanisms, like those presented in [14] and [25], we assume that some hot/cold identification technique is in place that misidentifies a certain fraction of the hot and cold pages. More specifically, we assume that a fraction f_p (f_n) of the cold (hot) pages is incorrectly labeled as hot (cold), resulting in a false positive (negative). We believe this makes our insights more generally applicable compared to focusing on a specific identification mechanism.

Figure 17 illustrates the impact of having false positives and negatives on the PE fairness for $r = 0.8$ and $f = 0.2$ (similar insights were obtained using different r and f values). This figure



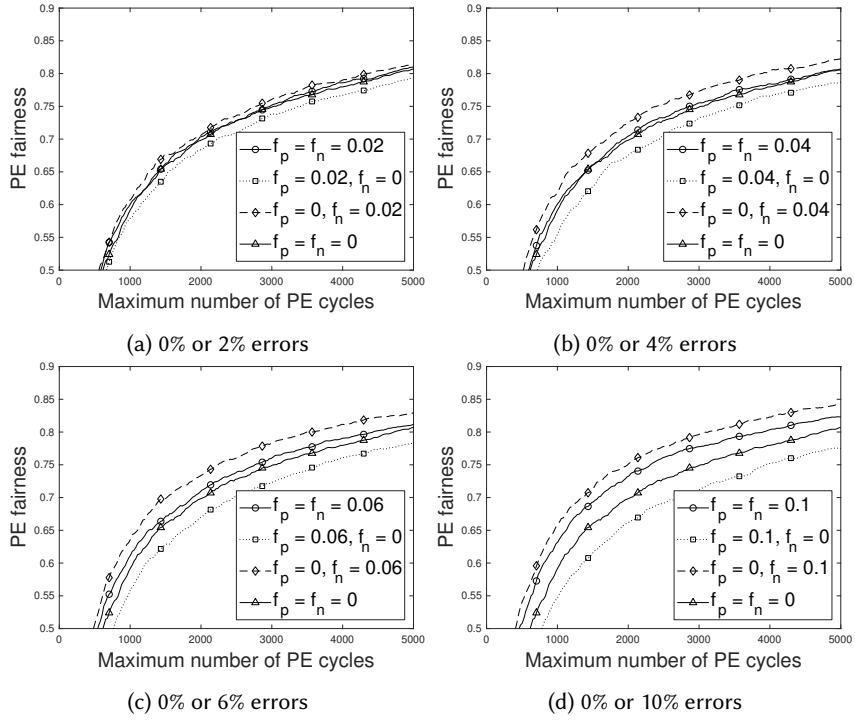
(a) $b = 32$ and $d = 10$ (b) $b = 64$ and $d = 10$ (c) $b = 128$ and $d = 10$

(d) $b = 32$ and $d = 100$ (e) $b = 64$ and $d = 100$ (f) $b = 128$ and $d = 100$

Fig. 16. SSD endurance for trace-based workloads with $S_f = 0.1$ under the DWF and HCWF mode with $f = 0.01$.

indicates that false negatives tend to improve the PE fairness, while false positives decrease the PE fairness. This can be understood by noting that the presence of false negatives decreases the number of blocks that are marked as hot since fewer pages are labeled hot. Due to this, it becomes less likely that a hot block is chosen as a victim during garbage collection (when d remains fixed), resulting in more cold victim blocks and hence, an improved PE fairness. The presence of false positives causes an increase in the number of hot blocks, which following the same reasoning decreases the PE fairness. We further observe that the negative impact of false positives on the PE fairness is outweighed by the positive effect of false negatives when both are present. Additional experiments (not shown here) indicate that the impact of having false positives and negatives becomes more pronounced as the data hotness increases.

As illustrated in Figure 18, the SSD endurance is best in the absence of false positives or negatives, i.e., when the hot and cold pages are separated perfectly. This can be understood as follows. In the case of false positives, the mislabeled cold valid pages are placed on hot blocks and these pages take a long time to be invalidated. Therefore, false positives increase the number of valid pages in hot victim blocks which significantly increases the write amplification (see [30]). This increase outweighs the improved fairness observed before and thus results in a poorer endurance. In the case of false negatives, the mislabeled hot pages are placed on cold blocks. While this also worsens the write amplification, the damage is less severe as the hot blocks still only contain hot pages and therefore hot victim blocks still contain many invalid pages. Hence, when selecting a hot/cold data identification technique w.r.t. the SSD endurance, having very few false positives seems more desirable to having very few false negatives. As was the case with the PE fairness, the results in Figure 18 become even more pronounced when further increasing the data hotness (as expected).



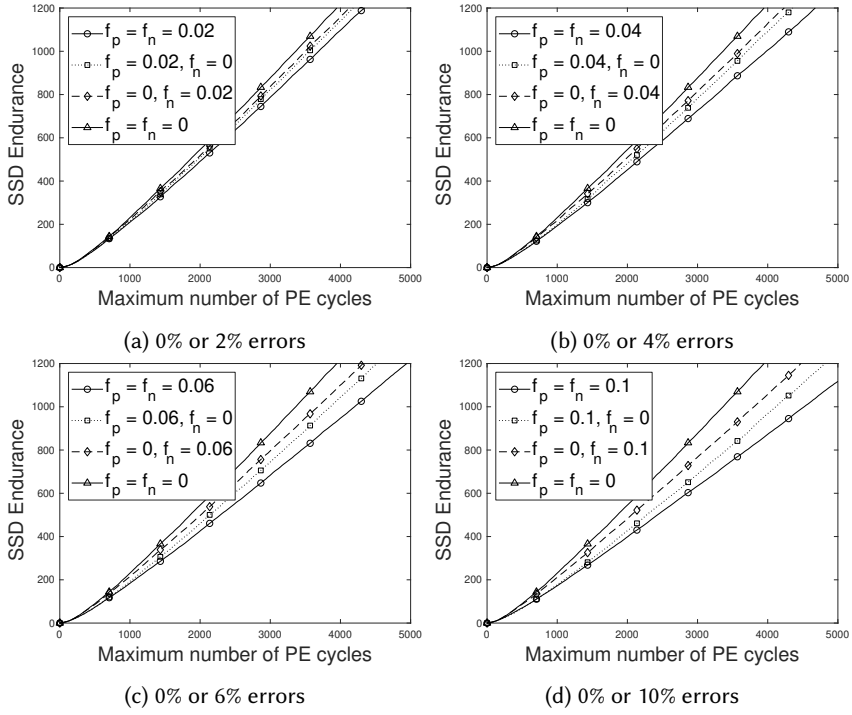
(a) 0% or 2% errors

(b) 0% or 4% errors

(c) 0% or 6% errors

(d) 0% or 10% errors

Fig. 17. Effect of false positives/negatives on PE fairness for synthetic workloads with $b = 32$ ($N = 10^4$), $d = 10$ and $S_f = 0.1$ under the HCWF mode with $r = 0.9, f = 0.1$.



(a) 0% or 2% errors

(b) 0% or 4% errors

(c) 0% or 6% errors

(d) 0% or 10% errors

Fig. 18. Effect of false positives/negatives on SSD endurance for synthetic workloads with $b = 32$ ($N = 10^4$), $d = 10$ and $S_f = 0.1$ under the HCWF mode with $r = 0.9$, $f = 0.1$.