Analytic models for flash-based SSD performance when subject to trimming

R. Verschoren and B. Van Houdt

Department of Mathematics and Computer Science
University of Antwerp - iMinds
Middelheimlaan 1, B-2020 Antwerp, Belgium
Email:{robin.verschoren,benny.vanhoudt}@uantwerpen.be

***Abstract*—Garbage collection is known to have a profound impact on SSD performance as it strongly influences the write amplification. Another key value that impacts the write amplification is the amount of over-provisioning, which lowers the write amplification at the cost of reducing the user-visible storage capacity. Write amplification occurs as the valid pages that remain on a block selected by garbage collection need to be copied to a free block before an erasure can take place. Some of these valid pages may however belong to a deleted file and therefore copying them is redundant. To avoid copying deleted data, the operating system can issue a Trim command to invalidate pages whenever a file is deleted.**

Prior analytical studies on the write amplification in SSDs assumed that no trimming takes place. In this paper we generalize a number of mean field models to assess the impact of trimming on the write amplification. Using these models we argue that the write amplification in a (large) system with trimming can be determined by analyzing a system without trimming that uses a larger over-provisioning factor (and modified hot fraction, in case of hot and cold data). Using numerical results we further show that trimming cold data results in a more significant reduction in the write amplification compared to trimming hot data.

I. INTRODUCTION

Modern SSDs make use of several channels each connected to a small number of flash chips. Each flash chip in turn may accommodate multiple dies, each consisting of a few planes. Finally, a single data plane contains thousands of blocks each holding a fixed number of pages, e.g., 64 pages per block. Data reads and writes are performed at the granularity of flash pages. Logical page addresses are often mapped to planes in a deterministic manner, that is, the channel, chip, die and plane numbers are fully determined by the logical address [13]. However, data within a plane is written in an out-of-place manner. Therefore the flash translation layer (FTL) maintains a table that maps logical page numbers to physical page number for each plane.

The reason for supporting out-of-place writes is that in order to write data to a page, the page must be in the erase state, which is one of the three possible states, the other two being valid or invalid. However, erasing data can only occur on a block level. Thus, in order to erase a single page, the entire block containing the page must be erased. Erasing a complete block and writing back all the valid data for each write would detriment the SSD performance. Therefore all data is written to a special block on the plane, called the write frontier (WF), and the physical page corresponding to the former location of the logical page is marked as invalid.

When the WF of a plane becomes full the garbage collection (GC) algorithm is triggered. The GC algorithm creates a new WF by selecting a victim block within the plane and erasing all of its pages. The victim block typically contains several valid pages and these are copied to a free block before the victim block is erased. These internal copy operations contribute to the so-called *write amplification*. The write amplification is equal to the ratio of the total number of writes divided by the number of writes requested by the operating system. If we denote p_j as the probability that the victim block contains j valid pages and denote the number of pages per block as b, the write amplification (WA) is given by $WA = b/(b - \sum_j jp_j)$.

When the write amplification increases, SSD performance degrades and its life span reduces as write operations are much slower than read operations and each block can only be erased a limited number of times, e.g., $10K$ times [6], before becoming useless. As such it is important that the GC algorithm, in the long run, selects blocks with as few valid pages as possible. To help reduce the write amplification SSDs are over-provisioned. More specifically, the user-visible address space, consisting of U pages, is smaller than the total number of pages T on the SSD. The fraction of over-provisioning $1-U/T$ is defined as the spare factor S_f and the larger S_f the more likely the GC algorithm is to find a block with a limited number of valid pages, which reduces the write amplification.

As with traditional hard disk drives, file delete operations are invisible to the SSD. Thus, some of the pages may be marked as valid and may be copied by the GC algorithm, while its data might belong to a deleted file. To prevent such unnecessary copy operations the Trim command was introduced (in the ATA command set). This command allows the operating system to inform the SSD about data that can be invalidated on the SSD (when a file is deleted).

Prior analytical models to assess the write amplification in SSDs have analyzed various GC algorithms: Random, Greedy [2], [3], Windowed [8], [9], FIFO [3], d-choices [14], [15], [16]. These models considered uniform random write workloads, hot and cold data models [12], as well as systems with hot and cold data identification [7], [11]. In all of these studies the FTL layer was assumed to be page-mapped, meaning a logical page can we written on any physical page of the plane, which provides better read and write performance than block-mapped FTLs at the expense of requiring more memory to store the FTL map. Further, these studies also assumed the Trim command was either not supported by the operating system/SSD device or was disabled.

In this paper we generalize (some of) the mean field models developed in [14], [15], [16] such that the impact of the Trim command can be studied while still considering a page-mapped FTL. Using these models we argue that the write amplification of a system that supports trimming can be computed from the models without trimming if the over-provisioning factor (and hot fraction) in the latter models is properly set. For instance, consider a system with spare factor S_f, i.e., utilization $\rho = 1 - S_f$, subject to uniform random writes. Assume the trim usage is such that logical pages are written at rate λ and if valid trimmed at rate μ. In this case the system with trimming is shown to have the same write amplification as a system without trimming with spare factor $1 - \rho\lambda/(\lambda + \mu)$. Note that $\lambda/(\lambda + \mu)$ is the probability that a logical page is stored on the drive and thus $\rho\lambda/(\lambda + \mu)$ is the *mean* effective utilization. If we assume a moderate trim rate $\mu/\lambda = 0.07$ (as found in some SSD workloads), this implies that an SSD with 5% over-provisioning and trimming behaves as an SSD with 11.2% over-provisioning without trimming as far as the write amplification is concerned. To the best of our knowledge, the only analytical models to study the effect of trimming were presented in [5], [4], which were mostly limited to uniform random writes and trim operations, where an incoming request is a trim request with a fixed probability $q < 1/2$.

The remainder of the paper is structured as follows. Section II discusses the GC algorithms, workload models and system setups considered in this paper. The mean field models, their validation and the equivalence with the models without trimming is presented in Section III. Numerical examples are presented in Section IV, while conclusions and possible generalizations are discussed in Section V.

II. System description

Although the GC algorithm typically runs on a plane level, the mean field models presented below can be used for SSDs that perform GC either globally or on a plane level. In the latter case the workload specified below corresponds to the workload offered to a single plane and the blocks considered are the set of blocks belonging to this plane.

A. GC algorithms

We consider the same class $\mathcal{C}$ of GC algorithms as in [14]. A GC algorithm belongs to $\mathcal{C}$ if and only if the following two conditions hold:

C1: Let b be the number of pages per block, m_i the fraction of blocks containing exactly i valid pages and denote $\vec{m} = (m_0, \ldots, m_b)$, then there should exist a set of probabilities $p_j(\vec{m})$ where $p_j(\vec{m})$ reflects the probability that a block containing exactly j valid pages is selected by the GC algorithm. In other words, whether block n, for any n, is selected by the GC algorithm should only depend on the number of valid pages j on block n and the fraction of blocks m_i containing exactly i valid pages, for $i = 0, \ldots, b$.

C2: For $j = 0, \ldots, b$, the probabilities $p_j(\vec{m})$ should be smooth in $\vec{m}$ with $\vec{m} \in \Delta = \{\vec{m} \in \mathbb{R}^{b+1} | m_i \geq 0, \sum_{i=0}^{b} m_i = 1\}$.

An important subset of $\mathcal{C}$ is the class of d-choices GC algorithms which select $d \geq 1$ blocks uniformly at random and erase a block containing the least number of valid pages among the d selected blocks. Hence,

$$p_j(\vec{m}) = \left(\sum_{\ell=j}^{b} m_\ell \right)^d - \left(\sum_{\ell=j+1}^{b} m_\ell \right)^d . \tag{1}$$

Note, when $d = 1$ this algorithm corresponds to the Random GC algorithm, while letting d tend to infinity coincides with the Greedy GC algorithm, which selects the block containing the least number of valid pages.

The mean field models introduced in this paper can be directly used to study a more general class of GC algorithms, where $p_j(\vec{m})$ is replaced by $p_{i,j}(\vec{m})$, which equals the probability that a block is selected containing exactly j valid pages, i of which are *hot*. In this case $\vec{m} \in \Delta^{h/c} = \{(m_{0,0}, m_{0,1}, m_{1,1}, m_{0,2}, \ldots, m_{b,b}) | m_{i,j} \geq 0, \sum_{0 \leq i \leq j \leq b} m_{i,j} = 1\}$ and $m_{i,j}$ is the fraction of blocks containing exactly j valid pages, i of which are hot.

B. Workload model

As read requests do not impact the write amplification, it suffices to model the write and trim requests. Let m_i be the fraction of the number of blocks containing exactly i valid pages. We consider two setups.

1) Uniform random writes and trims: We assume that there exists a set of probabilities $w(\vec{m})$, such that $w(\vec{m})$ is smooth in $\vec{m} \in \Delta$. An incoming request is a write request with probability $w(\vec{m})$ and a trim request with probability $x(\vec{m}) = 1 - w(\vec{m})$, where $\vec{m}$ is the current occupancy vector. Thus, the probability that an incoming request is a write or trim request depends only on the fraction of blocks containing exactly i valid pages, for $i = 0, \ldots, b$.

f	r (HW/(HW+CW))	HT/(HT+CT)	HT/HW	CT/CW
0.05	0.6619	0.5841	0.0623	0.0868
0.10	0.7291	0.6976	0.0676	0.0788
0.15	0.7768	0.7641	0.0695	0.0746
0.20	0.8126	0.8109	0.0705	0.0712
0.25	0.8417	0.8395	0.0704	0.0716
0.30	0.8663	0.8627	0.0703	0.0725
0.35	0.8872	0.8712	0.0693	0.0806
0.40	0.9040	0.8985	0.0702	0.0747
0.45	0.9197	0.9009	0.0692	0.0871
0.50	0.9327	0.9127	0.0691	0.0916

TABLE I: Fractions of write and trim requests to hot (HW,HT) and cold data (CW,CT) in the JEDEC Test Trace [10] for various choices of f.

An example that fits within the above framework is the case where each logical page is requested (i.e., written) at rate λ and any valid page on the SSD is invalidated by a trim request at rate μ, which yields

$$w(\vec{m}) = \frac{\lambda\rho}{\lambda\rho + \mu\sum_{i=0}^{b}\frac{i}{b}m_i},$$

where $\rho = U/T = 1 - S_f$.

2) Rosenblum workload [12]: A fraction f of the logical pages are called hot and the remaining pages are called cold. An incoming write updates a hot page with probability $r \geq f$ (and a cold page with probability $1-r$) [12]. When combined with trim requests, we assume there exists a set of probabilities $w_h(\vec{m})$, $w_c(\vec{m})$, $x_h(\vec{m})$ and $x_c(\vec{m})$ that are a smooth function of $\vec{m} \in \Delta^{h/c}$, such that $w_h(\vec{m})$ $(w_c(\vec{m}))$ is the probability that an incoming request writes a hot (cold) page and $x_h(\vec{m})$ $(x_c(\vec{m}))$ is the probability that an incoming request trims a hot (cold) page. Further, these probabilities are such that $r = w_h(\vec{m})/(w_h(\vec{m}) + w_c(\vec{m}))$.

For example, assume hot (cold) logical pages are written at rate λ_h (λ_c) and hot (cold) valid pages on the SSD are trimmed at rate μ_h (μ_c), respectively, then $(w_h(\vec{m}), w_c(\vec{m}), x_h(\vec{m}), x_c(\vec{m}))$ is proportional to

$$(\lambda_h\rho f, \lambda_c\rho(1-f), \mu_h\sum_{0\leq i\leq j\leq b}\frac{i}{b}m_{i,j}, \mu_c\sum_{0\leq i\leq j\leq b}\frac{j-i}{b}m_{i,j}),$$

where $m_{i,j}$ is the fraction of blocks containing j valid pages, i of which are hot. In this case $r = \lambda_h f/(\lambda_h f + \lambda_c(1-f))$.

To get an estimate of the fraction of requests that correspond to Trim commands in real workloads, we analyzed the JEDEC Test Trace [10]. Table I shows the relation between the number of write and trim requests, where the f most frequently requested addresses (w.r.t. both trim and write requests) were designated as hot. Table I indicates that the fraction of trim requests is approximately 7% for both hot and cold data, regardless of the fraction of pages that are labeled as hot.

C. Write frontier(s)

We consider two modes of operation, where the second mode relies on a hot/cold data identification technique.

1) Single write frontier: In this mode we use a single special block called the *write frontier* (WF). Pages are written sequentially to the WF, until it is full. Assume that the first j pages of the WF are in the (in)valid state, while the last $b-j$ are in the erase state at some point. Next, assume a write operation takes place on a logical page that is physically stored on page k of block number x. This operation first writes the new content to page $j+1$ of the WF (changing its state from erase to valid) and afterwards invalidates page k on block number x. When the WF is full, the GC algorithm creates a new WF as follows: it first selects a new block, say block number y, copies all the valid pages of block y to the random-access memory (RAM), erases block number y and copies the valid pages back from RAM to block y. Note, in practice one avoids the need to copy the valid pages to RAM and back by making use of a single *free* block [3]. Whether or not such a free block is used does not affect the write amplification as we only count the writes to the SSD.

2) HCWF: In this mode, one block is labeled as the hot write frontier (HWF) and another as the cold write frontier (CWF) at all times. New data is written to the corresponding WF, based on its hotness. We assume that a perfect hot/cold data identification technique is used, so a page is always written to the correct WF. Every block is also marked as hot or cold, based on whether it was last used as a hot or cold WF. The initial marking of the blocks is irrelevant. If the HWF becomes full, the GC algorithm is triggered to select a new HWF. Assume the GC algorithm selects a block containing j valid pages, called the victim block. Assume the CWF contains $b-k$ valid pages and thus, has k pages in the erase state when the GC algorithm is called. There are now 2 cases:

- If the victim block was marked hot, the j valid pages are simply copied back unto the victim after erasing it. The current HWF is released back into the data pool and the victim becomes the new HWF.
- If the victim block was marked cold and $j \leq k$, the j valid pages are copied to the CWF and the victim becomes the new HWF (with b pages in the erase state) after erasure. Otherwise if $j > k$, k of the j pages are (randomly) selected and copied to the CWF. The remaining $j-k$ pages are written back to the victim after erasure and the victim block becomes the new CWF. In this case the GC algorithm is triggered again to find a new HWF.

A same approach is used when the CWF becomes full, but with the terms hot and cold reversed.

III. Mean field models

In this section, we introduce three mean field models that extend some of the models presented in [14], [15], [16] to

Symbol	Explanation
N	number of physical blocks
b	number of pages per block
S_f	spare factor
ρ	utilization/load $(1 - S_f)$
m_i	fraction of blocks with i valid pages
$\vec{m}$	occupancy vector $(m_0, \ldots, m_b)$
$\rho_{\text{eff}}(\vec{m})$	effective utilization/load given $\vec{m}$
$p_j(\vec{m})$	prob. that GC selects block with j valid pages given $\vec{m}$
$w(\vec{m})$	prob. that a request is a write request given $\vec{m}$
$x(\vec{m})$	prob. that a request is a trim request given $\vec{m}$
$\pi_j(\vec{m})$	prob. that the WF holds i valid pages given $\vec{m}$

TABLE II: Table of notations for uniform random writes

incorporate the impact of the Trim command. When setting the trim probability equal to zero, the drift equations of the first two models do not coincide with to the ones presented in [14], [15] as the system is observed at different time epochs. For the third model, the drift equations do reduce to the ones presented in [16] if the trim probability is set to zero.

A. Uniform random writes

We first consider a system subject to uniform random writes that relies on a single write frontier. We show that under uniform random writes the new model reduces to the model in [14] with a modified parameter ρ, where $\rho = 1 - S_f$. In other words, under uniform random writes a system with spare factor S_f that has the Trim command enabled performs identical to a system with a larger spare factor with the Trim command disabled. Thus, the result presented next provides theoretical support for the approach taken in [5] under uniform random writes and trim operations.

1) Model description: We define a discrete-time model by observing the system state at the time epochs just prior to a write, a trim request or an invocation of the GC algorithm. Let $X_n^N(t) \in S = \{0, \ldots, b\}$ for $n = 1, \ldots, N$, denote the number of valid pages on block number n at the t-th point of observation (i.e. the t-h time the GC algorithm is activated or a write or trim request is received). Let $M^N(t)$ be the occupancy measure of $X_n^N(t)$, that is, $M^N(t) = \left(M_0^N(t), M_1^N(t), \ldots, M_b^N(t)\right)$ and

$$M_i^N(t) = \frac{1}{N}\sum_{n=1}^{N} 1[X_n^N(t) = i]$$

for $i \in S$.

Let $w(\vec{m})$ and $x(\vec{m}) = 1 - w(\vec{m})$ represent the probability that an incoming request is a write and trim request, respectively, given that the current occupancy vector equals $\vec{m}$. Define the effective load $\rho_{\text{eff}}(M^N(t))$ at time t as

$$\rho_{\text{eff}}(M^N(t)) = \frac{1}{b}\sum_{i=0}^{b} i M_i^N(t).$$

Let $J^N(t) \in S$ represent the number of valid pages in the write frontier at time t. The process $J^N(t), t \in \mathbb{N}$ forms a time inhomogeneous Markov chain with transition probability matrix $K(M^N(t))$ with entries

$$\left(K(M^N(t))\right)_{j,j'} = \begin{cases} x(M^N(t)) & \text{if } j' = j \\ w(M^N(t)) & \text{if } j' = j+1 \\ p_{j'}(M^N(t)) & \text{if } j = b \\ 0 & \text{otherwise,} \end{cases}$$

where $p_j(\vec{m})$ is the probability that the GC algorithm selects a block with j valid pages provided that the current occupancy vector equals $\vec{m}$. These transitions are as expected as a Trim command does not alter the content of the WF, a write adds a single page to the WF and a GC call selects a new WF.

For *fixed* $\vec{m}$, one readily verifies that the steady state probability vector $\pi(\vec{m}) = (\pi_0(\vec{m}), \ldots, \pi_b(\vec{m}))$ of $K(\vec{m})$ is given by

$$\pi_j(\vec{m}) = \frac{p_0(\vec{m}) + \ldots + p_j(\vec{m})}{w(\vec{m})}\pi_b(\vec{m}), \tag{2}$$

for $j < b$ and

$$\pi_b(\vec{m}) = w(\vec{m})/(w(\vec{m}) + \sum_{k=0}^{b} k p_{b-k}(\vec{m})).$$

While the process $\left\{\left(M^N(t), J^N(t)\right), t \in \mathbb{N}\right\}$ clearly forms a Markov chain, solving this Markov chain for practical values for N, e.g. $N = 10,000$, is unfeasible. For this reason, we define $\bar{M}^N(\tau)$ as a rescaled process such that $\bar{M}^N(t/N) = M^N(t)$ for $t \in \mathbb{N}$ and $\bar{M}^N(t)$ affine in $[t/N, (t+1)/N]$. Similarly, we define the rescaled process $\bar{J}^N(\tau)$ for the process $J^N(t)$ modeling the write frontier behavior. We will argue that the limit process of $\left(\bar{M}^N(t), \bar{J}^N(t)\right)$ as N tends to infinity is a deterministic process $\vec{\nu}(t) = \{\nu_i(t) | i \in S\}$, the evolution of which is given by the following set of ODEs:

$$\frac{d\nu_i(t)}{dt} = \sum_{j=0}^{b} \pi_j(\vec{\nu}(t)) f_i(\vec{\nu}(t), j) \tag{3}$$

The drift $f_i(\vec{m}, j)$ represents the expected change in the number of blocks containing i valid pages in between two points of observation, given that the occupancy measure is equal to $\vec{m}$ and the write frontier contains j valid pages. The drift can be defined as

$$f_i(\vec{m}, j) = \begin{cases} w(\vec{m})\frac{(i+1)\, m_{i+1} - i\, m_i}{b\rho} \\ \quad + x(\vec{m})\frac{(i+1)\, m_{i+1} - i\, m_i}{b\rho_{\text{eff}}(\vec{m})} & \text{if } j < b \\ & \\ - p_i(\vec{m}) + 1[i = b] & \text{if } j = b \end{cases} \tag{4}$$

for $i, j \in S$, where we assume $m_{b+1} = 0$. Indeed, if the WF is not full ($j < b$) a write occurs that invalidates a page with probability $w(\vec{m})\rho_{\text{eff}}(\vec{m})/\rho$ and this page belongs to a block with i valid pages with probability $im_i/\rho_{\text{eff}}(\vec{m})b$, while a trim occurs with probability $x(\vec{m})$ and invalidates a page on a block with i valid pages with probability $im_i/\rho_{\text{eff}}(\vec{m})b$.

2) Model equivalence: We now show that any fixed point of (3) coincides with a fixed point of the mean field model without trimming presented in [14] if ρ is replaced by the effective load.

Theorem 1. *Let $\vec{m}^\star$ be a fixed point of* (3)*, then $\vec{m}^\star$ is a fixed point of the set of ODEs given by (8) and (9) in [14] if ρ is replaced by $\rho_{\mathit{eff}}(\vec{m}^\star)$. Further, $\rho_{\mathit{eff}}(m^\star)$ satisfies*

$$\rho_{\mathit{eff}}(\vec{m}^\star) = \rho \left(1 - \frac{x(\vec{m}^\star)}{w(\vec{m}^\star)}\right). \tag{5}$$

Proof. As $\vec{m}^\star$ is a fixed point of (3) we have

$$\begin{aligned} 0 &= \sum_{j=0}^{b} \frac{\pi_j(\vec{m}^\star)}{\pi_b(\vec{m}^\star)} f_i(\vec{m}^\star, j) \\ &= \left(\frac{\sum_{j=0}^{b-1} \pi_j(\vec{m}^\star)}{\pi_b(\vec{m}^\star)}\right) f_i(\vec{m}^\star, 0) + f_i(\vec{m}^\star, b), \end{aligned}$$

as the drift $f_i(\vec{m}^\star, j)$ is identical for all $j < b$. Further, we have

$$\frac{\sum_{j=0}^{b-1} \pi_j(\vec{m}^\star)}{\pi_b(\vec{m}^\star)} = \frac{\sum_{k=0}^{b} k p_{b-k}(\vec{m}^\star)}{w(\vec{m}^\star)},$$

due to (2). Hence, by (4)

$$\begin{aligned} 0 = \frac{W(\vec{m}^\star)}{w(\vec{m}^\star)} f_i(\vec{m}^\star, 0) + f_i(\vec{m}^\star, b) = 1[i = b] - p_i(\vec{m}^\star) \\ + \frac{W(\vec{m}^\star)}{w(\vec{m}^\star)} \left(\frac{w(\vec{m}^\star)}{\rho} + \frac{x(\vec{m}^\star)}{\rho_{\text{eff}}(\vec{m}^\star)}\right) \left(\frac{(i+1)\, m^\star_{i+1} - i\, m^\star_i}{b}\right) \end{aligned} \tag{6}$$

where $W(\vec{m}^\star) = \sum_{k=1}^{b} k p_{b-k}(\vec{m}^\star)$ for brevity. Multiplying this expression by i and summing over i yields

$$\begin{aligned} &\frac{W(\vec{m}^\star)}{w(\vec{m}^\star)} \sum_{i=0}^{b} i f_i(\vec{m}^\star, 0) + \sum_{i=0}^{b} i f_i(\vec{m}^\star, b) = b - \sum_{i=0}^{b} i p_i(\vec{m}^\star) \\ &+ \frac{W(\vec{m}^\star)}{w(\vec{m}^\star)} \left(\frac{w(\vec{m}^\star)}{\rho} + \frac{x(\vec{m}^\star)}{\rho_{\text{eff}}(\vec{m}^\star)}\right) \left(-\sum_{i=0}^{b} \frac{(i+1) m^\star_{i+1}}{b}\right) \\ &= W(\vec{m}^\star) - \frac{W(\vec{m}^\star)}{w(\vec{m}^\star)} \left(\frac{w(\vec{m}^\star)}{\rho} + \frac{x(\vec{m}^\star)}{\rho_{\text{eff}}(\vec{m}^\star)}\right) \rho_{\text{eff}}(\vec{m}^\star) = 0, \end{aligned}$$

which proves (5). Further, combining (5) with (6) shows that

$$0 = 1[i = b] - p_i(\vec{m}^\star) + W(\vec{m}^\star) \left(\frac{(i+1)\, m^\star_{i+1} - i\, m^\star_i}{\rho_{\text{eff}}(\vec{m}^\star) b}\right), \tag{7}$$

which is the fixed point equation the mean field model presented in [14] if we replace $\rho_{\text{eff}}(\vec{m}^\star)$ by ρ. □

In other words, by setting the load ρ to be equal to $\rho_{\text{eff}}(\vec{m}^\star)$, one can investigate the impact of the Trim command on a system using uniform random writes without explicitly modeling the Trim command itself. The write amplification WA proposed by the model is expressed in terms of the fixed point using

$$WA = \frac{b}{b - \sum_{j=0}^{b} j p_j(\vec{m}^\star)},$$

where the fixed point is obtained numerically in a matter of seconds (see further).

In the special case where a logical page is written at rate λ and trimmed at rate μ if present on the SSD, the probability of a logical page to be stored on the SSD equals $\lambda/(\mu + \lambda)$. Therefore the *mean* load/utilization equals $\rho\lambda/(\mu + \lambda)$ and the next corollary shows that this corresponds to the effective load of the fixed point.

Corollary 1. *Let $w(\vec{m}) = \frac{\lambda\rho}{\lambda\rho + \mu\rho_{\mathit{eff}}(\vec{m})}$ and let $\vec{m}^\star$ be a fixed point of* (3)*, then $\rho_{\mathit{eff}}(\vec{m}^\star) = \frac{\lambda}{\lambda+\mu}\rho$.*

Proof. The result is immediate by plugging the expression for $w(\vec{m}^\star)$ into (5). □

The next corollary corresponds to the setting considered in [5] where the trim probability is assumed to be fixed:

Corollary 2. *Let $w(\vec{m}) = q < 1/2$ (as in [5]) and let $\vec{m}^\star$ be a fixed point of* (3)*, then $\rho_{\mathit{eff}}(\vec{m}^\star) = \frac{1-2q}{1-q}\rho$.*

3) Convergence and model validation: In order to show that the rescaled process $\bar{M}^N(t)$ converges to the unique solution of the ODE (3) we can rely on the mean field framework introduced in [1] by verifying that conditions H1 to H5 hold. We should note that Corollary 1 in [1] only guarantees convergence over finite time scales. In order to show that the convergence extends to the stationary regime, one needs to prove that the set of ODEs has a global attractor. As in [14] such a proof appears hard in general and is outside the scope of the paper.

When generating numerical results we restrict ourselves to the d-choices GC algorithm, but similar results can be generated for other GC algorithms belonging to the class introduced in [14]. We further assume that $w(\vec{m}) = \frac{\lambda\rho}{\lambda\rho+\mu\rho_{\mathit{eff}}(\vec{m})}$. We determine a fixed point of the ODE given by (3) using Euler's method with $\nu_i(0) = \binom{b}{i}\left(\frac{\lambda}{\lambda+\mu}\rho\right)^i\left(1 - \frac{\lambda}{\lambda+\mu}\rho\right)^{(b-i)}$ and a maximum step size $h = 0.001$ until $||\vec{\nu}(t+h) - \vec{\nu}(t)||_1 < 10^{-13}$.

Tables III and IV show a perfect agreement for the write amplification and effective load between the simulation results and the ODE-based prediction for various parameter settings. The simulations in tables III and IV were averaged over 10 runs each with a length of $10bN$. The length of the warm-up period was $\frac{10}{3}bN$, initialized with $b\rho N$ valid pages distributed uniformly at random over the bN available pages. The width of the 95% confidence intervals was approximately 0.01%, as indicated in tables III and IV.

B. Hot/cold data model with single write frontier

We now consider a system using non-uniform random writes, i.e. not all pages are updated or trimmed with equal probability, while still assuming the usage of a single write frontier. More specifically, we rely on the Rosenblum workload model discussed in Section II-B. Typical case studies with hot/cold

b	d	ρ	μ/λ	ODE (3)	sim. (95% conf.)
32	10	0.90	0.07	3.1761	3.1762 ± 0.0001
32	10	0.86	0.07	2.6455	2.6457 ± 0.0001
32	16	0.86	0.07	2.5999	2.5997 ± 0.0001
32	2	0.79	0.20	2.1260	2.1261 ± 0.0001
32	10	0.79	0.20	1.6611	1.6611 ± 0.0001
64	10	0.86	0.10	2.4768	2.4768 ± 0.0001
64	2	0.79	0.20	2.1405	2.1406 ± 0.0001

TABLE III: Comparison of ODE-based results and simulation experiments w.r.t. write amplification for a system with $N = 10,000$ blocks for various parameter settings (10 runs).

b	d	ρ	μ/λ	ODE (3)	$\frac{\lambda}{\lambda+\mu}\rho$	sim. (95% conf.)
32	10	0.90	0.07	0.8411	0.8411	0.8410 ± 0.0001
32	10	0.86	0.07	0.8037	0.8037	0.8037 ± 0.0001
32	16	0.86	0.07	0.8037	0.8037	0.8038 ± 0.0001
32	2	0.79	0.20	0.6583	0.6583	0.6583 ± 0.0001
32	10	0.79	0.20	0.6583	0.6583	0.6583 ± 0.0001
64	10	0.86	0.10	0.7818	0.7818	0.7819 ± 0.0001
64	2	0.79	0.20	0.6583	0.6583	0.6583 ± 0.0001

TABLE IV: Comparison of ODE-based results, analytic solution and simulation experiments w.r.t. effective load ρ_{eff} for a system using uniform random writes with $N = 10,000$ blocks for various parameter settings (10 runs).

data utilize values of $f \leq 0.2$ and $r \geq 0.8$, meaning more than 80% of the write requests are to less than 20% of the data [3].

1) Model description: Let $H_n^N(t) \in S = \{0, \dots, b\}$ for $n = 1, \dots, N$, denote the number of hot valid pages on block number n at time t (i.e. the t-h time the GC algorithm is activated or a write or trim request is received) and $C_n^N(t) \in S$ for $n = 1, \dots, N$ denote the number of cold valid pages on block number n at time t. Denote $X_n^N(t) = \left(H_n^N(t), H_n^N(t) + C_n^N(t)\right) \in \Omega = \{\{i,j\} | 0 \leq i \leq j \leq b\}$.

Let $M^N(t)$ be the occupancy measure of $X_n^N(t)$, i.e. $M^N(t) = \left\{M_{i,j}^N(t) | (i,j) \in \Omega\right\}$ and

$$M_{i,j}^N(t) = \frac{1}{N} \sum_{n=1}^{N} 1[X_n^N(t) = (i,j)]$$

for $(i,j) \in \Omega$.

Let $w_h(\vec{m})$ ($w_c(\vec{m})$) be the probability that an incoming request writes hot (cold) data and denote $x_h(\vec{m})$ ($x_c(\vec{m})$) as the probability that an incoming request trims hot (cold) data, given that the current occupancy measure equals $\vec{m} \in \Delta^{h/c}$. Further, let $w(\vec{m}) = w_h(\vec{m}) + w_c(\vec{m})$ and $x(\vec{m}) = x_h(\vec{m}) + x_c(\vec{m})$.

Let the hot load $\rho_h = \rho f$ and the cold load $\rho_c = \rho(1 - f)$ represent the maximum fraction of pages that contain hot and cold data respectively. Similarly, define the effective hot load $\rho_{\text{eff},h}(\vec{m}) = \sum_{i=0}^{b} \frac{i}{b} m_{i,j} \leq \rho_h$, the effective cold load $\rho_{\text{eff},c}(\vec{m}) = \sum_{i=0}^{b} \frac{j-i}{b} m_{i,j} \leq \rho_c$ and the total effective load $\rho_{\text{eff}}(\vec{m}) = \rho_{\text{eff},h}(\vec{m}) + \rho_{\text{eff},c}(\vec{m}) \leq \rho$.

Symbol	Explanation
$m_{i,j}$	fraction of blocks with j valid pages, i of which are hot
$\vec{m}$	occupancy vector $(m_{0,0}, \dots, m_{b,b})$
$\rho_{\text{eff},h}(\vec{m})$	effective hot load given $\vec{m}$
$\rho_{\text{eff},c}(\vec{m})$	effective cold load given $\vec{m}$
$p_{i,j}(\vec{m})$	prob. that GC selects block with j valid pages, i of which are hot, given $\vec{m}$
$w_h(\vec{m})$	prob. that a request writes a hot page given $\vec{m}$
$w_c(\vec{m})$	prob. that a request writes a cold page given $\vec{m}$
$x_h(\vec{m})$	prob. that a request trims a hot page given $\vec{m}$
$x_c(\vec{m})$	prob. that a request trims a cold page given $\vec{m}$
$\pi_{i,j}(\vec{m})$	prob. that WF holds a block with j valid pages, i of which are hot, given $\vec{m}$

TABLE V: Table of notations for SWF with hot/cold data

Let $J^N(t) \in \Omega$ represent the number of valid pages in the write frontier at time t. The process $J^N(t), t \in \mathbb{N}$ forms a time inhomogeneous Markov chain with transition probability matrix $K(M^N(t))$ with the following entries:

$$K_{(k,l),(k',l')}\left(M^N(t)\right) = \begin{cases} x(M^N(t)) & k = k' < b, l = l' < b, \\ w_h(M^N(t)) & k < b, l < b, k' = k+1, l' = l+1, \\ w_c(M^N(t)) & k < b, l < b, k' = k, l' = l+1, \\ p_{k',l'}(M^N(t)) & l = b, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $p_{k,l}(\vec{m})$ is the probability that the GC algorithm selects a block with l valid pages, k of which are hot, given that the occupancy vector equals $\vec{m}$.

For *fixed* $\vec{m}$ the steady state probability vector $\pi(\vec{m})$ is a solution of $\pi(\vec{m})K(\vec{m}) = \pi(\vec{m})$. Its entries can be computed by first computing the entries $\hat{\pi}_{k,l}(\vec{m})$ for $(k,l) \in \Omega$ recursively as

$$\hat{\pi}_{k,l}(\vec{m}) = \frac{p_{k,l}(\vec{m}) + w_h(\vec{m})\hat{\pi}_{k-1,l-1}(\vec{m}) + w_c(\vec{m})\hat{\pi}_{k,l-1}(\vec{m})}{w(\vec{m})}$$

$$\hat{\pi}_{k,b}(\vec{m}) = p_{k,b}(\vec{m}) + w_h(\vec{m})\hat{\pi}_{k-1,b-1}(\vec{m}) + w_c(\vec{m})\hat{\pi}_{k,b-1}(\vec{m})$$

where $\hat{\pi}_{k,l}(\vec{m})$ is defined as 0 for $(i,j) \notin \Omega$. We subsequently normalize $\hat{\pi}(\vec{m})$ to determine $\pi(\vec{m})$:

$$\pi_{i,j}(\vec{m}) = \frac{\hat{\pi}_{i,j}(\vec{m})}{\sum_{l=0}^{b} \sum_{k=0}^{l} \hat{\pi}_{k,l}(\vec{m})}.$$

We can now define the drift $\vec{f}(\vec{m}, k, l) = \{f_{i,j}(\vec{m}, k, l) | (i,j) \in \Omega\}$, which represents the expected difference in between two points of observation in the fraction of blocks containing j valid pages, i of which are hot, given that the occupancy measure is equal to $\vec{m}$ and the write frontier contains l

valid pages, k of which are hot. We define this drift as follows:

$$f_{i,j}(\vec{m},k,l) = \begin{cases} w_h(\vec{m})\dfrac{(i+1)m_{i+1,j+1} - im_{i,j}}{b\rho_h} \\ \quad +x_h(\vec{m})\dfrac{(i+1)m_{i+1,j+1} - im_{i,j}}{b\rho_{\text{eff},h}(\vec{m})} \\ \quad +w_c(\vec{m})\dfrac{(j+1-i)m_{i,j+1} - (j-i)m_{i,j}}{b\rho_c} \\ \quad +x_c(\vec{m})\dfrac{(j+1-i)m_{i,j+1} - (j-i)m_{i,j}}{b\rho_{\text{eff},c}(\vec{m})} & l < b, \\ -p_{i,j}(\vec{m}) + 1[i=k, j=b] & l = b, \end{cases} \tag{9}$$

which can be understood in a similar fashion as (4).

Using this drift we define the deterministic process $\vec{\nu}(t) = \{\nu_{i,j}(t) | \{i,j\} \in \Omega\}$, the evolution of which is given by the following set of ODEs:

$$\frac{d\nu_{i,j}(t)}{dt} = F_{i,j}(\vec{\nu}(t)) \tag{10}$$

where $F_{i,j}(\vec{m}) = \sum_{(k,l)\in\Omega} \pi_{k,l}(\vec{m}) f_{i,j}(\vec{m},k,l)$. As for the uniform random writes case, the results in [1] can be used to see that a rescaled version of the stochastic process $\{(M^N(t), J^N(t)), t \in \mathbb{N}\}$ converges to the unique solution of the above set of ODEs over any finite time scale.

2) Model equivalence: We now show that any fixed point of the set of ODEs for the model with trimming is also a fixed point of the set of ODEs for the model without trimming presented in [15] provided that the parameters ρ and f are properly adjusted.

To prove this, we start with the following lemma:

Lemma 1. *For $\vec{m} \in \Delta^{h/c}$ we have*

$$\sum_{i=0}^{b} i\pi_{i,b}(\vec{m}) \Big/ \sum_{k=0}^{b} \pi_{k,b}(\vec{m}) = \frac{w_h(\vec{m})}{w(\vec{m})} \sum_{j=0}^{b} jp_{b-j}(\vec{m}) + \sum_{j=0}^{b}\sum_{i=0}^{j} ip_{i,j}(\vec{m}), \tag{11}$$

and

$$\sum_{i=0}^{b} (b-i)\pi_{i,b}(\vec{m}) \Big/ \sum_{k=0}^{b} \pi_{k,b}(\vec{m}) = \frac{w_c(\vec{m})}{w(\vec{m})} \sum_{j=0}^{b} jp_{b-j}(\vec{m}) + \sum_{j=0}^{b}\sum_{i=0}^{j} (j-i)p_{i,j}(\vec{m}), \tag{12}$$

Proof. From (8) it follows that the probabilities $K_{(k,b),(k',l')}(\vec{m})$ do not depend on k. This allows us to state

$$\frac{\pi_{i,b}(\vec{m})}{\sum_{k=0}^{b} \pi_{k,b}(\vec{m})} = \sum_{k=0}^{i} \sum_{j=i-k}^{b-k} p_{i-k,j}(\vec{m}) B_k^{b-j,r}, \tag{13}$$

where $B_k^{b-j,r} = \binom{b-j}{k} r^k (1-r)^{b-j-k}$ and $r = w_h(\vec{m})/w(\vec{m})$. Multiplying by i, summing over i and changing the order of the sums implies

$$\sum_{i=0}^{b} i\pi_{i,b}(\vec{m}) \Big/ \sum_{k=0}^{b} \pi_{k,b}(\vec{m}) = \sum_{k=0}^{b}\sum_{j=0}^{b-k}\sum_{i=0}^{j} (i+k) p_{i,j}(\vec{m}) B_k^{b-j,r}.$$

The equality in (11) now follows by noting that

$$\sum_{k=0}^{b}\sum_{j=0}^{b-k}\sum_{i=0}^{j} ip_{i,j}(\vec{m}) B_k^{b-j,r} = \sum_{j=0}^{b}\sum_{i=0}^{j} ip_{i,j}(\vec{m}).$$

and

$$\sum_{k=0}^{b}\sum_{j=0}^{b-k}\sum_{i=0}^{j} kp_{i,j}(\vec{m}) B_k^{b-j,r} = \sum_{j=0}^{b} p_j(\vec{m})(b-j)r.$$

The equality in (12) can be derived similarly by noting that

$$\sum_{i=0}^{b} (b-i)\pi_{i,b}(\vec{m}) \Big/ \sum_{k=0}^{b} \pi_{k,b}(\vec{m}) = \sum_{k=0}^{b}\sum_{j=0}^{b-k}\sum_{i=0}^{j} [(b-j-k)+(j-i)] p_{i,j}(\vec{m}) B_k^{b-j,r}.$$

□

Theorem 2. *Let $\vec{m}^\star$ be a fixed point of* (10). *Define $\tilde{\rho} = \rho_{\mathit{eff},h}(\vec{m}^\star) + \rho_{\mathit{eff},c}(\vec{m}^\star)$ and $\tilde{f} = \rho_{\mathit{eff},h}(\vec{m}^\star)/\tilde{\rho}$, then $\vec{m}^\star$ is a fixed point of the set of ODEs given by (7) and (8) in [15] if (r,f,ρ) is replaced by $(r,\tilde{f},\tilde{\rho})$. Further, $\rho_{\mathit{eff},z}(m^\star)$ satisfies*

$$\rho_{\mathit{eff},z}(\vec{m}^\star) = \rho_z \left(1 - \frac{x_z(\vec{m}^\star)}{w_z(\vec{m}^\star)}\right). \tag{14}$$

for $z = h, c$.

Proof. We start with the observation that the fixed point equation for (10) can be written as

$$\left(\sum_{l=0}^{b-1}\sum_{k=0}^{l} \pi_{k,l}(\vec{m}^\star)\right) f_{i,j}(\vec{m}^\star, 0, 0) + \sum_{k=0}^{b} \pi_{k,b}(\vec{m}^\star) f_{i,j}(\vec{m}^\star, k, b) = 0, \tag{15}$$

Due to (9), we have

$$\sum_{k=0}^{b} \pi_{k,b}(\vec{m}^\star) f_{i,j}(\vec{m}^\star, k, b) = -p_{i,j}(\vec{m}^\star)\left(\sum_{k=0}^{b} \pi_{k,b}(\vec{m}^\star)\right) + \pi_{i,b}(\vec{m}^\star) 1[j=b]. \tag{16}$$

and

$$f_{i,j}(\vec{m}^\star,0,0) = \left(\frac{w_h(\vec{m}^\star)}{\rho_h}+\frac{x_h(\vec{m}^\star)}{\rho_{\text{eff},h}(\vec{m}^\star)}\right)\frac{(i+1)m^\star_{i+1,j+1}+im^\star_{i,j}}{b}+ \left(\frac{w_c(\vec{m}^\star)}{\rho_c}+\frac{x_c(\vec{m}^\star)}{\rho_{\text{eff},c}(\vec{m}^\star)}\right)\frac{(j+1-i)m^\star_{i,j+1}+(j-i)m^\star_{i,j}}{b} \tag{17}$$

Next, we note that

$$\sum_{l=0}^{b-1}\sum_{k=0}^{l}\pi_{k,l}(\vec{m}^\star)\Big/\sum_{k=0}^{b}\pi_{k,b}(\vec{m}^\star) = \sum_{l=1}^{b}\frac{l}{1-x(\vec{m}^\star)}\sum_{k=0}^{b-l}p_{k,b-l}(\vec{m}^\star)=\frac{1}{w(\vec{m}^\star)}\sum_{l=1}^{b}lp_{b-l}(\vec{m}^\star), \tag{18}$$

as this ratio is equal to the mean number of incoming trim or write requests in between two GC calls.

Combining (15), (16) (17) and (18) with (13) in Lemma 1 results in the following fixed point equation

$$0=\left(\frac{w_h(\vec{m}^\star)}{w(\vec{m}^\star)}\frac{(i+1)m^\star_{i+1,j+1}+im^\star_{i,j}}{\rho_{\text{eff},h}(\vec{m}^\star)b} +\frac{w_c(\vec{m})}{w(\vec{m}^\star)}\frac{(j+1-i)m^\star_{i,j+1}+(j-i)m^\star_{i,j}}{\rho_{\text{eff},c}(\vec{m}^\star)b}\right)\sum_{l=1}^{b}lp_{b-l}(\vec{m}^\star) - p_{i,j}(\vec{m}^\star)+\sum_{k=0}^{i}\sum_{j=i-k}^{b-k}p_{i-k,j}(\vec{m}^\star)B_k^{b-j,r}1[j=b],$$

provided that (14) holds. As this equation is identical to the fixed point equation of (7) and (8) in [15] if (f,ρ) are replaced by $(\tilde{f},\tilde{\rho})$, it suffices to prove (14) to complete the proof.

To establish (14) for $z=h$ we multiply (15) by i and sum over i and j. By (16) and (11) we find

$$\sum_{j=0}^{b}\sum_{i=0}^{j}\sum_{k=0}^{b}\pi_{k,b}(\vec{m}^\star)if_{i,j}(\vec{m}^\star,k,b)\Big/\left(\sum_{l=0}^{b}\pi_{l,b}(\vec{m}^\star)\right) =\frac{w_h(\vec{m}^\star)}{w(\vec{m}^\star)}\sum_{j=0}^{b}jp_{b-j}(\vec{m}^\star), \tag{19}$$

Due to (17)

$$\sum_{j=0}^{b}\sum_{i=0}^{j}if_{i,j}(\vec{m}^\star,0,0) = -\left(\frac{w_h(\vec{m}^\star)}{\rho_h}+\frac{x_h(\vec{m}^\star)}{\rho_{\text{eff},h}(\vec{m}^\star)}\right)\underbrace{\left(\frac{1}{b}\sum_{j=0}^{b}\sum_{i=0}^{j}im^\star_{i,j}\right)}_{\rho_{\text{eff},h}(\vec{m}^\star)}. \tag{20}$$

Combining (15), (18), (19) and (20) yields (14) for $z=h$. The argument for $z=c$ proceeds in a similar fashion, but we multiply (15) by $j-i$ before summing over i and j and rely on (12) instead of (11). $\square$

It is worth remaking that the effective hot load $\rho_{\text{eff},h}(\vec{m}^\star)$ only depends on ρ_h and the ratio $x_h(\vec{m}^\star)/w_h(\vec{m}^\star)$. However, the latter may in general also depend on the effective cold load $\rho_{\text{eff},c}(\vec{m}^\star)$, meaning the fixed point equations for $\rho_{\text{eff},h}(\vec{m}^\star)$ and $\rho_{\text{eff},c}(\vec{m}^\star)$ are not necessarily decoupled as in the next two corollaries.

Corollary 3. *Let* $(w_h(\vec{m}),w_c(\vec{m}),x_h(\vec{m}),x_c(\vec{m}))$ *be proportional to* $(\lambda_h\rho_h,\lambda_c\rho_c,\mu_h\rho_{\mathit{eff},h}(\vec{m}),\mu_c\rho_{\mathit{eff},c}(\vec{m}))$*, then* $\rho_{\mathit{eff},h}(\vec{m}^\star)=\frac{\lambda_h}{\lambda_h+\mu_h}\rho f$ *and* $\rho_{\mathit{eff},c}(\vec{m}^\star)=\frac{\lambda_c}{\lambda_c+\mu_c}\rho(1-f)$*.*

The above special case corresponds to the setting where logical hot/cold pages are written at rate λ_h/λ_c and if present on the SSD trimmed at rate μ_h/μ_c, respectively. It shows that the effective hot and cold loads match the mean number of hot and cold pages on the device, as anticipated.

Corollary 4. *Let* $x_h(\vec{m})=q_h, x_c(\vec{m})=q_c, w_h=r(1-q_h-q_c)$ *and* $w_h=(1-r)(1-q_h-q_c)$ *and let* $\vec{m}^\star$ *be a fixed point of* (3)*, then* $\rho_{\mathit{eff},h}(\vec{m}^\star)=\rho f(1-q_h/[r(1-q_h-q_c)])$ *and* $\rho_{\mathit{eff},c}(\vec{m}^\star)=\rho(1-f)(1-q_c/[(1-r)(1-q_h-q_c)])$*.*

The fixed write and trim probabilities of the latter corollary were considered in [4, Section 6.1], where the write amplification for the system with mixed hot and cold data was estimated by replacing the load of the uniform model with the overall effective load. As shown in Figure 8 in [4, Section 6.1], this approach results in a very poor agreement with simulation as opposed to the results presented below.

3) Model validation: As for the uniform random writes case we focus on the d-choices algorithm and set the trim and write probabilities as specified in Corollary 3. The numerical results for the write amplification and valid page distribution were generated by numerically determining a fixed point of the ODE (10) with $\nu_{i,j}(0)=\binom{i}{j}f^i(1-f)^{j-i}\eta_j$ using Euler's method with a variable step size h until $||\nu(t+h)-\nu(t)||_1<10^{-10}$, where η_j is the fixed point of (3) when $\lambda=f\lambda_h+(1-f)\lambda_c$, $\mu=f\mu_h+(1-f)\mu_c$ and $\rho=\frac{\lambda_h\rho_h}{\lambda_h+\mu_h}+\frac{\lambda_c\rho_c}{\lambda_c+\mu_c}$.

The ODE-based results were compared with the mean of 10 simulation runs of the system with the same parameters and $N=10,000$. Each run consisted of $500bN$ external (write or trim) requests and the length of the warm-up period was $\frac{500}{3}bN$. The results are shown in tables VI and VII and indicate that there is an excellent agreement for the write amplification and hot effective load between the simulation results and the ODE solution (similar results were obtained for the cold effective load).

C. Hot/cold data model with HCWF

In this subsection, we consider the HCWF mode of operations outlined in Section II-C when subject to a Rosenblum workload as discussed in Section II-B.

1) Model description: Let $X_n^N(t)\in S=\{0,\dots,b\}$ denote the number of valid pages in block n and $Y_n^N(t)\in\{h,c\}$ reflect whether block n is marked hot (h) or cold (c) at the t-th point of observation, i.e. the t-the time the GC algorithm

d	ρ	λ_h	μ_h/λ_h	μ_c/λ_c	ODE (10)	sim. (95% conf.)
2	0.82	16	0.20	0.20	2.4316	2.4317 ± 0.0001
2	0.87	16	0.20	0.20	2.7536	2.7536 ± 0.0001
10	0.90	16	0.07	0.07	3.5069	3.5069 ± 0.0001
10	0.90	16	0.07	0.14	2.9056	2.9057 ± 0.0001
16	0.90	24	0.07	0.07	3.5275	3.5277 ± 0.0001
10	0.87	16	0.20	0.20	2.2933	2.2935 ± 0.0001
10	0.87	12	0.20	0.03	3.1853	3.1854 ± 0.0001

TABLE VI: Comparison of ODE-based results and simulation experiments w.r.t. write amplification for a system using hot/cold writes and single WF with $\lambda_c = 1$, $N = 10,000$ blocks of size $b = 32$ and a fraction $f = 0.2$ of hot data for various parameter settings (10 runs).

d	ρ	λ_h	μ_h/λ_h	μ_c/λ_c	ODE (10)	$\frac{\lambda_h \rho_h}{\lambda_h+\mu_h}$	sim. (95% conf.)
2	0.82	16	0.20	0.20	0.1367	0.1367	0.1366 ± 0.0001
2	0.87	16	0.20	0.20	0.1450	0.1450	0.1450 ± 0.0001
10	0.90	16	0.07	0.07	0.1682	0.1682	0.1683 ± 0.0001
10	0.90	16	0.07	0.14	0.1682	0.1682	0.1683 ± 0.0001
16	0.90	24	0.07	0.07	0.1682	0.1682	0.1683 ± 0.0001
10	0.87	16	0.20	0.20	0.1450	0.1450	0.1451 ± 0.0001
10	0.87	12	0.20	0.03	0.1450	0.1450	0.1450 ± 0.0001

TABLE VII: Comparison of ODE-based results and simulation experiments w.r.t. hot effective load for a system using hot/cold writes and single WF with $\lambda_c = 1$, $N = 10,000$ blocks of size $b = 32$ and a fraction $f = 0.2$ of hot data for various parameter settings (10 runs).

is activated or an external write or trim request is issued. Let $M^N_{z,i}(t)$ be the occupancy measure of $X^N_n(t)$ and $Y^N_n(t)$, i.e., $M^N(t) = \{M^N_{z,i}(t) | z \in \{h, c\}, i \in S\}$, where

$$M^N_{z,i}(t) = \frac{1}{N} \sum_{n=1}^{N} 1\left[X^N_n(t) = i, Y^N_n(t) = z\right]$$

for $i \in S$ and $z \in \{h, c\}$.

Furthermore, let $J^N(t) \in \Omega = \{(k,l) | 0 \le k, l \le b\} \setminus \{(b,b)\}$ represent the number of valid pages in the HWF and the CWF at time t.

The mean field model is defined by means of the deterministic process $\vec{\nu}(t) = \{\nu_{z,i}(t) | z \in \{h,c\}, i \in S\}$, the evolution of which is given by the following set of ODEs:

$$\frac{d\nu_{z,i}(t)}{dt} = \sum_{(k,l)\in\Omega} \pi_{k,l}(\vec{m}(t)) f_{z,i}(\vec{m}(t), k, l) \tag{21}$$

where the drift $f_{z,i}(\vec{m}, k, l)$ is defined below and $\vec{\pi}(\vec{m})$ is the invariant probability vector of $K(\vec{m})$, where $(K(\vec{m}))_{i,j} = P[J(t+1) = i | J(t) = j, M(t) = \vec{m}]$ with $(i,j) \in \Omega$.

The expression for the drift is identical to [16, Section 3.1], except that the case $k, l < b$ needs to be adapted, similar to

Symbol	Explanation
$m_{h,i}$	fraction of blocks marked hot holding i valid pages
$m_{c,i}$	fraction of blocks marked cold holding i valid pages
$\vec{m}$	occupancy vector $(m_{h,0}, \ldots, m_{c,b})$
$p_{h,j}(\vec{m})$	prob. that GC selects a hot block with j valid pages given $\vec{m}$
$p_{c,j}(\vec{m})$	prob. that GC selects a cold block with j valid pages given $\vec{m}$
$\pi_{i,j}(\vec{m})$	prob. that HWF contains i and the CWF j valid pages given $\vec{m}$

TABLE VIII: Table of notations for HCWF with hot/cold data

the single WF model, to incorporate the changes caused by the trim requests

$$f_{h,i}(\vec{m}, k, l) = \tag{22}$$
$$\begin{cases} w_h(\vec{m}) \dfrac{(i+1)m_{h,i+1} - i m_{h,i}}{b\rho f} \\ \quad + x_h(\vec{m}) \dfrac{(i+1)m_{h,i+1} - i m_{h,i}}{b\rho_{\text{eff},h}(\vec{m})} & k < b, l < b, \\ -p_{h,i}(\vec{m}) & i < b, k = b \text{ or } l = b, \\ 1 - p_{h,b}(\vec{m}) - p_{c,(b-l)+}(\vec{m}) & i = b, k = b, \\ p_{h,(b-k)+}(\vec{m}) - p_{h,b}(\vec{m}) & i = b, l = b, \end{cases}$$

where $p_{z,i+}(\vec{m}) = \sum_{j=i}^{b} p_{z,j}(\vec{m})$ for $z = h, c$ and analogously for blocks containing cold data,

$$f_{c,i}(\vec{m}, k, l) = \tag{23}$$
$$\begin{cases} w_c(\vec{m}) \dfrac{(i+1)\, m_{c,i+1} - i\, m_{c,i}}{b\rho(1-f)} \\ \quad + x_c(\vec{m}) \dfrac{(i+1)m_{c,i+1} - i m_{c,i}}{b\rho_{\text{eff},c}(\vec{m})} & k < b, l < b, \\ -p_{c,i}(\vec{m}) & i < b, k = b \text{ or } l = b, \\ 1 - p_{c,b}(\vec{m}) - p_{h,(b-k)+}(\vec{m}) & i = b, l = b, \\ p_{c,(b-l)+}(\vec{m}) - p_{c,b}(\vec{m}) & i = b, k = b. \end{cases}$$

The entries of the probability matrix $K(\vec{m})$, are also nearly identical to [16, Section 3.1], except that when $k, l < b$, there is a probability $x(\vec{m})$ that a trim request arrives. As trim requests do not affect the WFs, the new state $(k', l') = (k, l)$ with probability $x(\vec{m})$, hence

$$K_{(k,l),(k',l')}(\vec{m}) = \tag{24}$$
$$\begin{cases} x(\vec{m}) & k = k' < b, l = l' < b, \\ w_h(\vec{m}) & k < b, k' = k+1, l = l' < b, \\ w_c(\vec{m}) & l < b, l' = l+1, k = k' < b, \\ p_{h,k'}(\vec{m}) & k = b, k' < b, l = l' < b, \\ p_{c,l'}(\vec{m}) & l = b, l' < b, k = k' < b, \\ p_{c,l'-l}(\vec{m}) & k = b, k' = 0, b \ge l' \ge l, \\ p_{h,k'-k}(\vec{m}) & l = b, l' = 0, b \ge k' \ge k, \\ p_{c,b-l+l'}(\vec{m}) & k = k' = b, 0 < l' < l, \\ p_{h,b-k+k'}(\vec{m}) & l = l' = b, 0 < k' < k, \\ p_{h,b}(\vec{m}) + p_{c,b}(\vec{m}) & k = k', l = l', k \text{ or } l = b, \\ 0 & \text{otherwise}. \end{cases}$$

2) Model equivalence: As for the single write frontier we now show that any fixed point of the set of ODEs for the model with trimming is also a fixed point of the set of ODEs for the model without trimming presented in [16, Section 3.1] provided that the parameters ρ and f are modified.

Theorem 3. *Let $\vec{m}^\star$ be a fixed point of (21). Define $\tilde{\rho} = \tilde{\rho}_h(\vec{m}^\star) + \tilde{\rho}_c(\vec{m}^\star)$ and $\tilde{f} = \tilde{\rho}_h(\vec{m}^\star)/\tilde{\rho}$, then $\vec{m}^\star$ is a fixed point of the set of ODEs given by (1) in [16] if (r, f, ρ) is replaced by $(r, \tilde{f}, \tilde{\rho})$. Further, $\tilde{\rho}_z(\vec{m}^\star)$ is a solution of*

$$\frac{1}{\tilde{\rho}_z(\vec{m}^\star)} = \frac{1}{w_z(\vec{m}^\star)}\left(\frac{w_z(\vec{m}^\star)}{\rho_z} + \frac{x_z(\vec{m}^\star)}{\rho_{\mathit{eff},z}(\vec{m}^\star)}\right), \qquad (25)$$

for $z = h, c$.

Proof. The fixed point equation can be written as

$$\begin{aligned}
&-\sum_{k=0}^{b-1} \pi_{k,b}(\vec{m}^\star) f_{z,i}(\vec{m}^\star, k, b) - \sum_{l=0}^{b-1} \pi_{b,l}(\vec{m}^\star) f_{z,i}(\vec{m}^\star, b, l) \\
&\quad = \sum_{(k,l)\in\Omega_{<b,<b}} \pi_{k,l}(\vec{m}^\star) f_{z,i}(\vec{m}^\star, k, l) \\
&\quad = \sum_{(k,l)\in\Omega_{<b,<b}} \pi_{k,l}(\vec{m}^\star)(1 - x(\vec{m}^\star)) \times \\
&\quad \frac{1}{w(\vec{m}^\star)}\left(\frac{w_z(\vec{m}^\star)}{\rho_z} + \frac{x_z(\vec{m}^\star)}{\rho_{\text{eff},z}(\vec{m}^\star)}\right)\frac{(i+1)m^\star_{z,i+1} + i m^\star_{z,i}}{b},
\end{aligned}$$

where $\Omega_{<b,<b} = \{(k,l) | 0 \le k, l < b\}$. This yields

$$\begin{aligned}
&-\sum_{k=0}^{b-1} \tilde{\pi}_{k,b}(\vec{m}^\star) f_{z,i}(\vec{m}^\star, k, b) - \sum_{l=0}^{b-1} \tilde{\pi}_{b,l}(\vec{m}^\star) f_{z,i}(\vec{m}^\star, b, l) \\
&\quad = \sum_{(k,l)\in\Omega_{<b,<b}} \tilde{\pi}_{k,l}(\vec{m}^\star)\frac{w_z(\vec{m}^\star)}{w(\vec{m}^\star)}\frac{(i+1)m^\star_{z,i+1} + i m^\star_{z,i}}{(\tilde{\rho}_z(\vec{m}^\star)/\tilde{\rho})b\tilde{\rho}},
\end{aligned}$$

where $\tilde{\pi}_{k,l}(\vec{m}^\star) = \pi_{k,l}(\vec{m}^\star)$ if k or l equals b and $\tilde{\pi}_{k,l}(\vec{m}^\star)/(1 - x(\vec{m}^\star)) = \pi_{k,l}(\vec{m}^\star)$ otherwise. The probabilities $\tilde{\pi}_{k,l}(\vec{m}^\star)$ are the steady state probabilities of the $K(\vec{m}^\star)$ matrix defined in [16, Section 3.1] as this matrix is identical to (24) with $x(\vec{m}^\star) = 0, w_h(\vec{m}^\star) = r$ and $w_c(\vec{m}^\star) = 1 - r$. Hence, the above fixed point equation is identical to the one in [16, Section 3.1] if we replace $(\tilde{f}, \tilde{\rho})$ by (f, ρ) as $w_h(\vec{m}^\star)/w(\vec{m}^\star) = r$ and $w_c(\vec{m}^\star)/w(\vec{m}^\star) = 1 - r$. ☐

We suspect that $\tilde{\rho}_z(\vec{m}^\star)$ equals $\rho_{\text{eff},z}(\vec{m}^\star)$, but proving this in general for the HCWF appears more challenging than in the single WF case. The next corollary indicates that simple explicit expressions for $\tilde{\rho}_z(\vec{m}^\star)$ exist in some particular cases:

Corollary 5. *Let $(w_h(\vec{m}), w_c(\vec{m}), x_h(\vec{m}), x_c(\vec{m}))$ be proportional to $(\lambda_h\rho_h, \lambda_c\rho_c, \mu_h\rho_{\mathit{eff},h}(\vec{m}), \mu_c\rho_{\mathit{eff},c}(\vec{m}))$, then $\tilde{\rho}_h(\vec{m}^\star) = \frac{\lambda_h}{\lambda_h+\mu_h}\rho f$ and $\tilde{\rho}_c(\vec{m}^\star) = \frac{\lambda_c}{\lambda_c+\mu_c}\rho(1 - f)$.*

In fact, in this case we find that $\tilde{\rho}_z(\vec{m}^\star)$ is equal to $\rho_{\text{eff},z}(\vec{m}^\star)$ of the SWF setting (see Corollary 3).

d	ρ	λ_h	μ_h/λ_h	μ_c/λ_c	ODE (eq. (10))	sim. (95% conf.)
2	0.82	16	0.20	0.20	2.0770	2.0772 ± 0.0001
2	0.87	16	0.20	0.20	2.3446	2.3451 ± 0.0001
10	0.90	16	0.07	0.07	2.5730	2.5735 ± 0.0001
10	0.90	16	0.07	0.14	2.1687	2.1691 ± 0.0001
16	0.90	24	0.07	0.07	2.4920	2.4925 ± 0.0001
10	0.87	16	0.20	0.20	1.6938	1.6940 ± 0.0001
10	0.87	12	0.20	0.03	2.3815	2.3820 ± 0.0001

TABLE IX: Comparison of ODE-based results and simulation experiments w.r.t. write amplification for a system using hot/cold writes and HCWF with $\lambda_c = 1$, $N = 10,000$ blocks of size $b = 32$ and a fraction $f = 0.2$ of hot data for various parameter settings (10 runs).

3) Convergence and numerical solution: The convergence of the rescaled stochastic process towards the mean field model over finite time scales is once more guaranteed by [1, Corollary 1].

To generate numerical results we determine a fixed point $\vec{\nu} = \{\nu_{z,i} | z \in \{h, c\}, i \in S\}$ by solving the ODE using Euler's method. Euler's method is iterative and may require as many as a few thousand iterations. During iteration $i + 1$, we need to compute the drifts $\vec{f}(\vec{m}^{(i)}, k, l)$ for all $(k, l) \in \Omega$, where $\vec{m}^{(i)}$ represents the occupancy vector after iteration i. Following this, it is also necessary to compute the steady state probabilities $\pi_{k,l}(\vec{m})$ during each iteration. This becomes time consuming for realistic values of b, e.g. $b = 32$ or 64, since $K(\vec{m})$ is a $b(b + 2)$ state Markov chain.

To reduce computation times per iteration in Euler's method from $O(b^6)$ to $O(b^4)$ we can follow the approach taken in [16, Section 3.2]. It may appear that this approach cannot be applied directly as the matrices $K_{<b,<b}(\vec{m})$ and $K_{<b,=b}(\vec{m})$ now depend on $\vec{m}$. However, it suffices that we compute $(I - K_{<b,<b}(\vec{m}))^{-1}K_{<b,=b}(\vec{m})$ during each iteration and it is not hard to see that this product is still independent of $\vec{m}$.

4) Model validation and numerical results: As for the single WF case, we focus on the d-choices GC algorithm and set the write and trim rates as specified in Corollary 5. The numerical results for the write amplification and valid page distribution were generated by numerically solving the ODE given in eq. (21) using Euler's method with a variable step size h until $||\nu(t + h) - \nu(t)||_1 < 10^{-10}$. This procedure was initialized with $\nu_{z,0}(0) = \rho_z/\rho - \nu_{z,b}(0)$, $\nu_{z,b}(0) = \rho_z\lambda_c/(\lambda_c + \mu_c)$ and $\nu_{z,j}(0) = 0$ for $j \in S \setminus \{0, b\}$ and $z \in \{h, c\}$.

Tables IX and X compare the ODE-based write amplification and effective hot load and the mean of 10 simulation runs of a system with $N = 10,000$ blocks. Each simulation run consisted of $500bN$ external (write or trim) requests and the length of the warm-up period was $\frac{500}{3}bN$. The results indicate that there is a strong agreement between the simulation results and the numerical solutions of the set of ODEs in eq. (21) for various parameter settings.

d	ρ	λ_h	μ_h/λ_h	μ_c/λ_c	ODE (eq. (10))	$\frac{\lambda_h \rho_h}{\lambda_h+\mu_h}$	sim. (95% conf.)
2	0.82	16	0.20	0.20	0.1367	0.1367	0.1365 $\pm$ 0.0001
2	0.87	16	0.20	0.20	0.1450	0.1450	0.1450 $\pm$ 0.0001
10	0.90	16	0.07	0.07	0.1682	0.1682	0.1683 $\pm$ 0.0001
10	0.90	16	0.07	0.14	0.1682	0.1682	0.1682 $\pm$ 0.0001
16	0.90	24	0.07	0.07	0.1682	0.1682	0.1682 $\pm$ 0.0001
10	0.87	16	0.20	0.20	0.1450	0.1450	0.1451 $\pm$ 0.0001
10	0.87	12	0.20	0.03	0.1450	0.1450	0.1450 $\pm$ 0.0001

TABLE X: Comparison of ODE-based results and simulation experiments w.r.t. hot effective load for a system using hot/cold writes and HCWF with $\lambda_c = 1$, $N = 10,000$ blocks of size $b = 32$ and a fraction $f = 0.2$ of hot data for various parameter settings (10 runs).

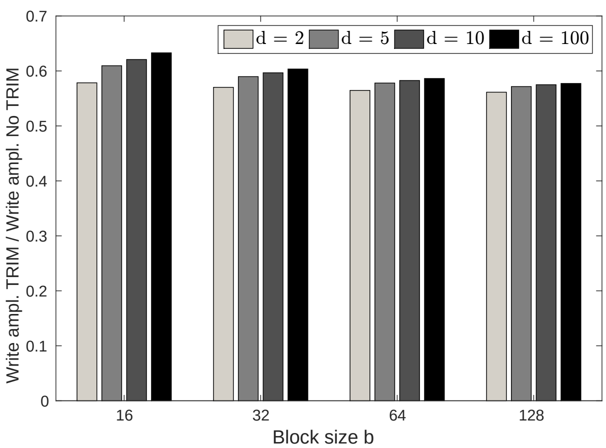

Fig. 1: Write amplification as a function of the block size b for various choices of d in a system using uniform random writes with $\rho = 0.9$, $\lambda = 1$ and $\mu = 0.1$, relative to that of a system without trimming.

IV. NUMERICAL RESULTS

In this section, we present some numerical results for the systems discussed in this paper that complement the theorems presented in the previous section. The trim probabilities are set according to Corollary 1, 3 or 5, depending on whether or not we consider hot/cold data and a SWF or HCWF setup.

A. Uniform random writes

Figure 1 demonstrates that under uniform random writes and trim operations the impact of the block size b on the reduction of the write amplification achieved by trimming is rather limited (for various d), where somewhat larger gains are observed as the number of pages b on a block increases. For the considered block sizes, the observed write amplification when the trim rate is 10 times as small as the write rate is approximately 60% of the write amplification without trimming. This shows that the lifespan of an SSD can be greatly extended by supporting the Trim command as far fewer erasures are needed to support the same number of external writes. More

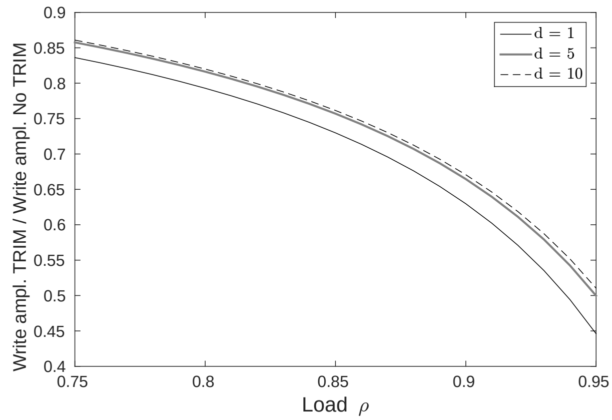

Fig. 2: Write amplification for varying loads ρ and various choices of d in a system using uniform random writes with $b = 32$, $\lambda = 1$ and $\mu = 0.07$.

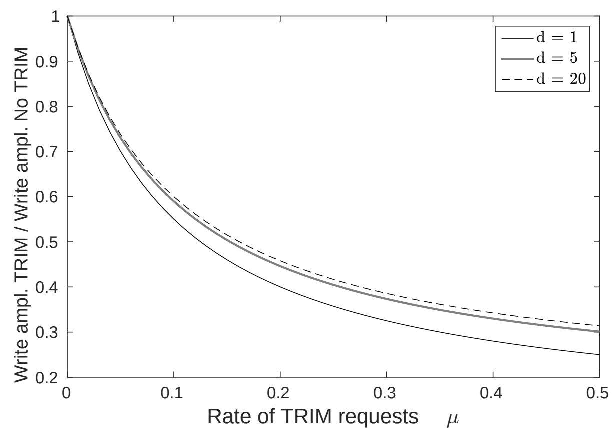

Fig. 3: Write amplification as a function of varying trim rates μ for a system using uniform random writes with $b = 32$, assuming $\lambda = 1$, proportional to the write amplification for a system without trimming.

specifically, as the life span is inversely proportional to the write amplification a 40% reduction in the write amplification corresponds to an increased life span of about 66%.

The write amplification as a function of the load ρ is depicted in fig. 2. Larger gains are observed for systems with a smaller space factor $S_f = 1-\rho$. In other words, if the system is heavily over-provisioned, the Trim command offers limited additional gain, while for small spare factors (e.g., 5%) it may double the lifespan. Another way to view these results is that with trimming one can reduce the spare factor, allowing a larger user-visible storage space, without jeopardizing the lifespan of the device. Figure 3 shows the impact of the trim rate and indicates that the write amplification decreases significantly even when trimming at a low rate. The observed decrease is larger for small choices of d, meaning the Random algorithm ($d = 1$) benefits somewhat more from trimming than the Greedy algorithm ($d = \infty$).

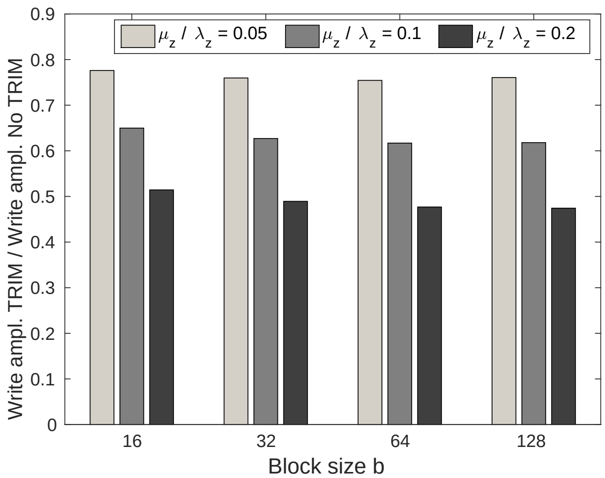

Fig. 4: Write amplification as a function of block size b for various choices of μ_z/λ_z in a HC (SWF) system with $d = 10$, $f = 0.2$, $\rho = 0.9$, $\lambda_h = 16$ and $\lambda_c = 1$, relative to that of a system without trimming.

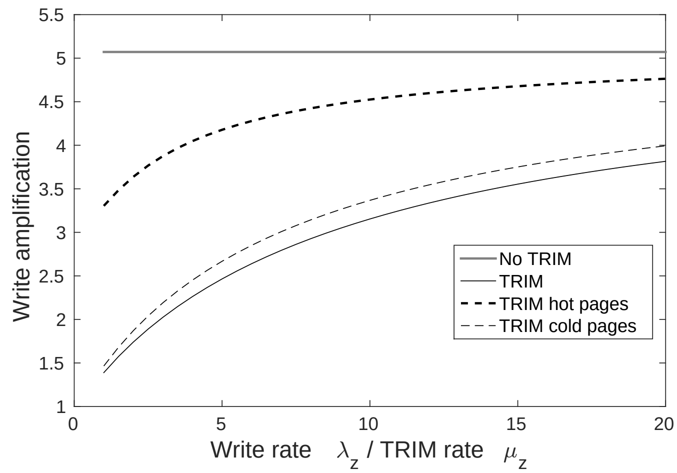

Fig. 5: Write amplification for several values of λ_z/μ_z for HC (SWF) systems where no data, all data, only hot data and only cold data is trimmed with $d = 10$, $b = 32$, $f = 0.2$, $\rho = 0.9$ and $\lambda_h = 16\lambda_c$.

B. Hot and cold data

In this section we consider hot/cold data combined with a single WF, meaning the hot and cold data is mixed and there is no hot/cold data identification technique in place. We assume that hot (cold) logical pages are requested at rate λ_h (λ_c) and that the hot (cold) pages stored on the SSD get trimmed at rate μ_h (μ_c), respectively. The impact of trimming on the write amplification for various choices of λ_z/μ_z, with $z \in \{h, c\}$, is not heavily influenced by the block size b as indicated in fig. 4, where $d = 10$ and $S_f = 0.1$.

In fig. 6, we look at the write amplification as a function of λ_z/μ_z without trimming, with hot or cold data trimming only and in case all data is trimmed. We see that the decrease in the write amplification is much more pronounced when only trimming the cold pages, compared to only trimming hot data. This can be understood by noting that cold data remains on the device for longer periods of time without being invalidated, while hot data is invalidated more often due to incoming external writes or updates. Hence, trimming hot pages does not contribute as much as to lowering the effective load.

In the previous plots we considered the GC algorithm with $d = 10$ choices, in fig. 6 we see that larger values of d tend to further decrease the write amplification, but setting $d = 10$ results in a write amplification that is already fairly close to the greedy algorithm, which corresponds to setting $d = \infty$. Further, most of the gain is achieved by trimming the cold data irrespective of the number of choices d used.

Figure 7 looks at the same setup as Figure 5 except that the SWF is replaced by the HCWF. In other words, the hot and cold data is written to separate WFs which implies that a hot/cold data identification technique is needed. As in the SWF case, trimming cold data result in more significant gains (as expected). Further, when the hot and cold data is separated trimming hot data only seems to result in a more significant relative gain compared to the case with mixed hot and cold data.

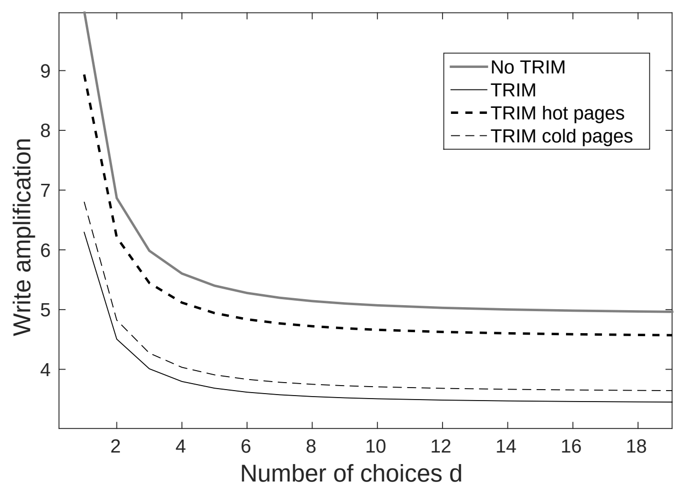

Fig. 6: Write amplification as a function of the number of choices d for HC (SWF) systems where no data, all data, only hot data and only cold data is trimmed with $b = 32$, $f = 0.2$, $\rho = 0.9$, $\lambda_h = 16\lambda_c$ and $\mu_z/\lambda_z = 0.07$.

Figure 8 looks at the impact of the number of choices d in the HCWF case. We observe that the Greedy GC algorithm is no longer optimal. This is expected given Theorem 3 and the fact that the Greedy GC algorithm is known to be suboptimal in the HCWF setting without trimming (see [16]).

Finally, fig. 9 depicts the gain achieved by implementing a hot/cold data separation technique and using the HCWF compared to a system that simply relies on the SWF. It

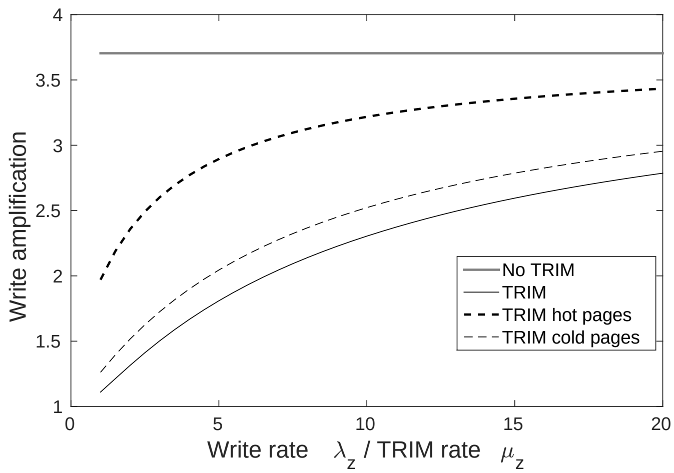

Fig. 7: Comparison of evolution of the write amplification for several values of λ_z/μ_z for HCWF systems where no data, all data, only hot data and only cold data is trimmed with $b = 32$, $f = 0.2$, $\rho = 0.9$ and $\lambda_h = 16\lambda_c$.

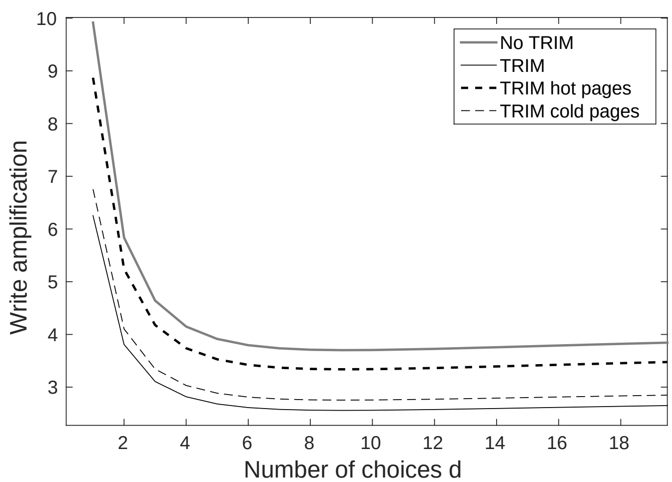

Fig. 8: Comparison of evolution of the write amplification in function of number of choices d for HCWF systems where no data, all data, only hot data and only cold data is trimmed with $d = 10$, $b = 32$, $f = 0.2$, $\rho = 0.9$, $\lambda_h = 16\lambda_c$ and $\mu_z/\lambda_z = 0.07$.

indicates that the overall gain decreases somewhat as the trim rate increases, but remains significant even at high trim rates. When only trimming the hot data the results suggest that the relative gain of the HCWF setup grows as the trim rate increases. This is in line with the previous results and can be understood by noting that in the SWF case the hot data is mixed with the cold data and a block contains mostly cold pages and a few hot ones, thus trimming the hot pages provides only limited gain. When only trimming the cold pages, the opposite occurs: the SWF setting appears to benefit more from trimming.

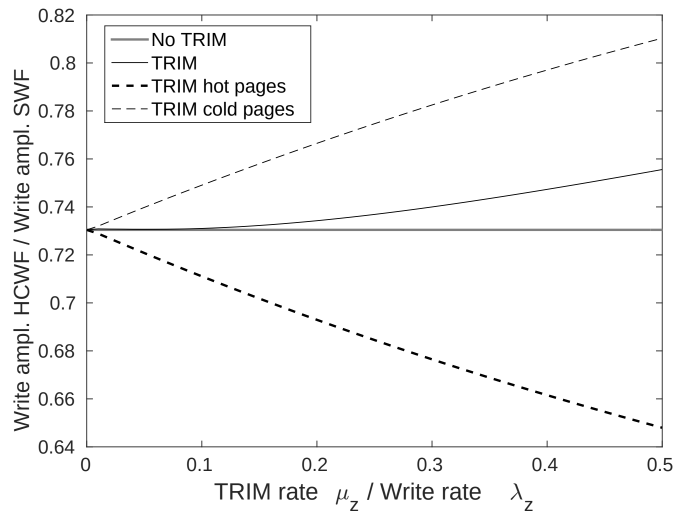

Fig. 9: Write amplification of SWF versus HCWF setup as a function of varying trim rates μ for a system using uniform random writes with $b = 32$, $d = 10$, $\rho = 0.9$, $f = 0.2$, $\lambda_h = 16$ and $\lambda_c = 1$.

V. Conclusion

In this paper we introduced a number of mean field models to assess the write amplification in the presence of trimming. The main take-away message is that the write amplification of these models corresponds to the write amplification of the corresponding model without trimming provided that the spare factor and hot data ratio is properly adjusted. Numerical results also show that the Trim command considerably extends the life span of an SSD especially when the spare factor is small and trimming cold data has a more profound impact on the write amplification than trimming hot data.

The equivalence between the mean field models with and without trimming should be fairly easy to generalize to the setting where there are $n > 2$ levels of data hotness, e.g., a fraction f_i of the data is requested at rate r_i, for $1 \leq i \leq n$. Another possible extension is to consider the setting with the double write frontier (DWF) presented in [15]. The idea behind the DWF approach is to achieve a partial separation between the hot and cold data without the need to implement a hot/cold data identification technique by using a different write frontier for the internal and externally requested writes.

References

[1] M. Benaïm and J. Le Boudec. A class of mean field interaction models for computer and communication systems. *Performance Evaluation*, 65(11-12):823–838, 2008.

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

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

[4] T. Frankie, G. Hughes, and K. Kreutz-Delgado. Analysis of trim commands on overprovisioning and write amplification in solid state drives, 2012. http://arxiv.org/abs/1208.1794.

[5] T. Frankie, G. Hughes, and K. Kreutz-Delgado. A mathematical model of the trim command in NAND-flash SSDs. In *Proceedings of the 50th Annual Southeast Regional Conference*, ACM-SE '12, pages 59–64, New York, NY, USA, 2012. ACM.

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

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

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

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

[10] JEDEC Solid State Technology Association and JC-64.8 and JC-64. Solid State Drive (SSD) Endurance Workloads. JESD219A, JESD219A_MT, JESD219A_TT.

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

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

[13] A. Tavakkol, M. Arjomand, and H. Sarbazi-Azad. Unleashing the potentials of dynamism for page allocation strategies in SSDs. *SIGMETRICS Perform. Eval. Rev.*, 42(1):551–552, June 2014.

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

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

[16] B. Van Houdt. On the necessity of hot and cold data identification to reduce the write amplification in flash-based SSDs. *Performance Evaluation*, 82:1 – 14, 2014.