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

Robin Verschoren and Benny Van Houdt

*Dept. Mathematics and Computer Science
University of Antwerp
Antwerp, Belgium

MSST 2016

Universiteit Antwerpen

Outline

- SSD basics
- Prior work
- Trimming
- Model description
  - GC algorithms
  - Workloads
  - Framework
- Model validation
- Main findings
- Future work

Universiteit Antwerpen

Flash-based SSD

SSD Structure (plane level)

- Data is organized in N blocks
- Fixed number of b pages per block (e.g., $b = 32$)
- Unit of data exchange is a page
- Page has 3 possible states: erase, valid or invalid.

Operations

- Data can only be written on pages in erase state
- Erase operations can be performed on entire blocks only
- Out-of-place writes are supported (old data becomes invalid)

Universiteit Antwerpen

Flash-based SSD

Internal operation (internal log structure)

- New data is sequentially written to one or more special blocks called write frontiers (WFs)
- When a WF is full, a new WF is selected by the garbage collection (GC) algorithm

Write Amplification

- Valid pages in the victim block are temporarily copied to perform erase
- Assume j valid pages on a victim block with probability p_j, write amplification A equals

$$A = \frac{b}{b - \sum_{j=0}^{b} j p_j}$$

Write Amplification

Importance

- Affects IOPS and life span of the drive

Over-provisioning

- Physical storage capacity exceeds the user-visible (logical) capacity
- Measure is spare factor $S_f = 1 - \rho$:

$$\rho = \frac{\text{the user-visible capacity}}{\text{total storage capacity}}$$

$\Rightarrow$ fraction S_f of the pages is guaranteed to be in erase/invalid state

Prior work

Analytical models

- Mostly under uniform random writes and Rosenblum (hot/cold) workloads
- Exact (closed-form) results as N tends to infinity
  - Random GC
  - FIFO/LRU GC (Menon, Robinson, Desnoyers)
  - Greedy GC (Bux, Illiadis, Desnoyers)
  - d-choices GC (Van Houdt, Li et al.)
  - Approximation for Windowed GC (Hu et al.)
  - etc.

Prior work

Main observations w.r.t. Write Amplification (WA)

- Greedy is optimal under uniform random writes, *d*-choices close to optimal (for *d* as small as 10)
- Increasing hotness worsens WA in case of single WF (as no hot/cold data separation takes place)
- Double WF (separates writes triggered by host and GC): WA decreases with hotness (as partial hot/cold data separation takes place)
- Hot/cold WF (separates hot and cold pages): WA decreases even further (not much) with hotness
- Greedy is no longer optimal with hot/cold data: there exists optimal *d* for *d*-choices

Universiteit Antwerpen

Trimming

Trim command

- When a file is deleted by the host, the Trim command can be used to invalidate the associated pages on the SSD
- This clearly lowers the WA
- All prior models (except for one) assume no trimming

Main questions

- How do we model trim behavior and develop accurate analytical models?
- How does trimming impact the WA and do the main observations remain valid?

Class $\mathcal{C}$ of GC algorithms modeled

Definition

- Let $\vec{m}(t) = (m_0(t), \ldots, m_b(t))$, where $m_i(t)$ is the fraction of blocks containing i valid pages at time t
- A GC algorithm belongs to $\mathcal{C}$ if
  1. A block containing j valid pages is selected by the GC algorithm with probability $p_j(\vec{m})$
  2. The probabilities $p_j(\vec{m})$ are smooth in $\vec{m}$ (can be slightly relaxed)
- It is possible to further extend this class when hot/cold data identification techniques are in place

Universiteit Antwerpen

Class $\mathcal{C}$ of GC algorithms modeled

Examples

1. Random GC algorithm: $p_j(\vec{m}) = m_j$
2. d-choices GC algorithm selects $d \geq 2$ blocks uniformly at random and erases a block containing the smallest number of valid pages among the d selected blocks:

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

3. Greedy GC algorithm: d-choices with $d = N$.

Universiteit Antwerpen

Workload model

Rosenblum model (proofs can be extended to more than 2 classes)

- A fraction f of the data is termed hot
- Hot pages are updated at rate $r \geq f$, cold pages at rate $1 - r$
- Reducing f or increasing r makes hot data hotter
- When $r = f$: uniform random writes

Trim model (special case, see paper general setting)

- Uniform random writes: each logical page is written at rate λ and any valid page on the SSD is invalidated by a trim request at rate μ
- Hot/cold data: write and trim rates also depend on hotness, we have λ_h, λ_c, μ_h and μ_c

Model framework

Background on mean field models

- Stochastic system of N interacting blocks (N-dimensional Markov chain)
- Problem: impractical to compute steady state for large N
- Solution: consider the limit of N tending to infinity
- Limit is a deterministic system, its evolution captured by the trajectories of a set of ODEs (called drift equations)
- Drift corresponds to studying the behavior of one (type of) block, averaging the effects of other blocks

Universiteit Antwerpen

Model framework

Drift equations and fixed point (for uniform random writes)

- Let $f_i(\vec{m}, j)$ represent the expected change in the fraction of blocks containing i valid pages, given WF contains j valid pages (happens with probability $\pi_j(\vec{m})$, which depends on $\vec{m}$)
- Determine fixed point $\vec{m}^\star$ where

$$\sum_{i=0}^{b} \sum_{j=0}^{b} \pi_j(\vec{m}^\star) f_i(\vec{m}^\star, j)) = 0$$

- Write amplification and effective load based on fixed point
$A(\vec{m}^\star) = \frac{b}{b - \sum_{j=0}^{b} j p_j(\vec{m}^\star)}, \qquad \rho_{\text{eff}}(\vec{m}^\star) = \sum_{j=0}^{b} j \, {m^\star}_j$
- Gives exact results for N tending to infinity (provided that limits are exchangeable)

Validation: Uniform random writes

b	d	$1 - S_f$	μ/λ	model	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 : 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).

Universiteit Antwerpen

Validation: Hot/cold WF and Rosenblum workload

d	ρ	λ_h	$\frac{\mu_h}{\lambda_h}$	$\frac{\mu_c}{\lambda_c}$	model	sim. (95% conf.)
2	0.82	16	0.20	0.20	2.0770	2.0772 $\pm$ 0.0001
2	0.87	16	0.20	0.20	2.3446	2.3451 $\pm$ 0.0001
10	0.90	16	0.07	0.07	2.5730	2.5735 $\pm$ 0.0001
10	0.90	16	0.07	0.14	2.1687	2.1691 $\pm$ 0.0001
16	0.90	24	0.07	0.07	2.4920	2.4925 $\pm$ 0.0001
10	0.87	16	0.20	0.20	1.6938	1.6940 $\pm$ 0.0001
10	0.87	12	0.20	0.03	2.3815	2.3820 $\pm$ 0.0001

Table : 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).

Main findings

Main takeaway

- Trimming results in effective load (utilization) $\rho_{\text{eff}} \leq \rho$
- Proof that fixed points of models with and without trimming coincide if parameters are properly set:
  - Uniform random writes: $\rho \leftarrow \rho_{\text{eff}}$
  - Hot/cold data (SWF/HCWF):

$$\rho \leftarrow \rho_{\text{eff}} = \rho_{\text{eff},h} + \rho_{\text{eff},c}, \qquad f \leftarrow \frac{\rho_{\text{eff},h}}{\rho_{\text{eff}}}$$

- Special case
  - Uniform random writes: $\rho_{\text{eff}} = \dfrac{\lambda}{\lambda + \mu}\rho$
  - Hot/cold data: $\rho_{\text{eff},h} = \dfrac{\lambda_h}{\lambda_h + \mu_h}\rho f, \quad \rho_{\text{eff},c} = \dfrac{\lambda_c}{\lambda_c + \mu_c}\rho(1-f)$
- Write amplification reduces up to 40% even with limited trimming

Other findings

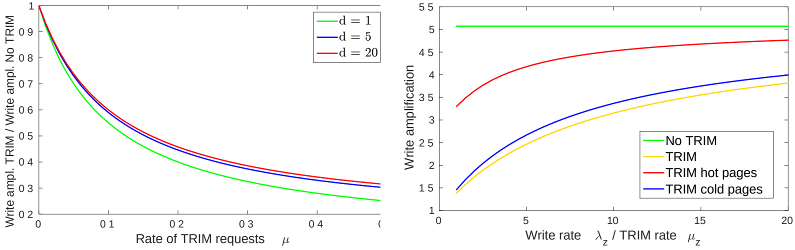

Figure : Left: Reduction in WA under uniform random writes for $b = 32$, $S_f = 0.1$, $\lambda = 1$ and $d = 1$, 5 and 20. Right: WA with hot/cold data (SWF) as a function of λ_z/μ_z with $b = 32$, $S_f = 0.1$, $r = 0.8$ and $f = 0.2$.

Possible extensions and ongoing work

Possible extensions

- Arbitrary number $n > 2$ of data hotness levels
- Other GC algorithms
- Other WF mechanisms (e.g., DWF)

Ongoing and future work

- Effect of WF mechanism on device lifespan
- Impact of several wear leveling schemes on device lifespan