Zurich Research Laboratory

Write Amplification Analysis in Flash-Based Solid State Drives

X.-Y. Hu, E. Eleftheriou, R. Haas, I. Iliadis, R. Pletka

SYSTOR'09, Haifa, 2009

Outline

- Introduction
  - Context of the Problem
  - Related Work
- Write Amplification Analysis
  - Write Amplification Definition
  - Garbage Collection Framework
    - Windowed Greedy Reclaiming Policy
  - Analytical Model for Write Amplification Computation
- Results and Extensions
  - Simulation vs. Analysis
  - Impact of Separating Static Data from Dynamic Data
- Conclusions

Context of Problem

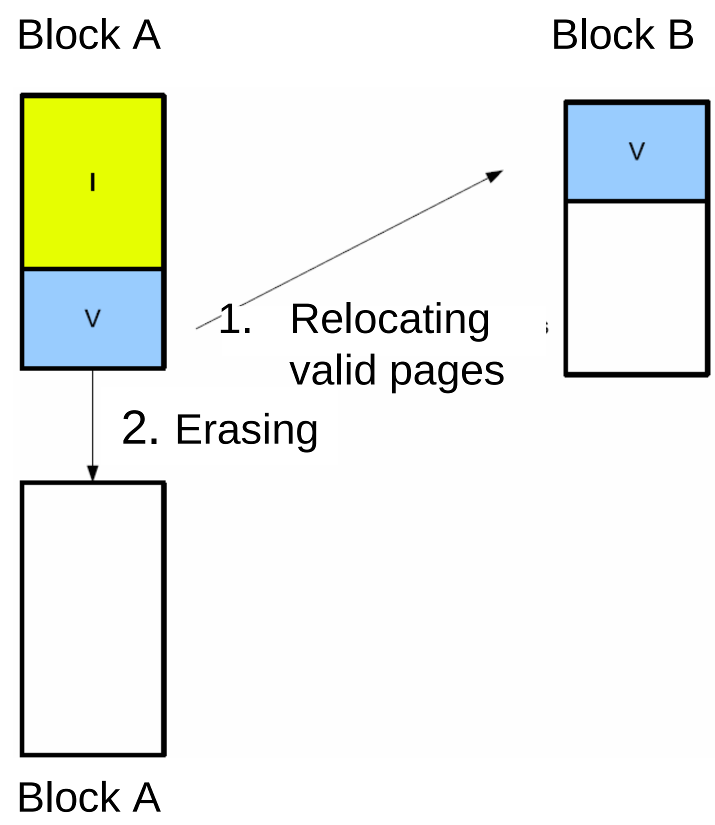
- NAND flash memory characteristics
  - Organized in terms of blocks, each block consisting of typically 64 pages, 4 KiB each
  - Block must be erased before data can be written
  - A block is the elementary unit for erase operations that are slow
  - Reads and writes are processed in terms of pages
  - Each block has a limited program/erase cycle count. Typically, SLC sustains 10^5, MLC ~ 10^4 cycles
- Relocate-on-write is needed for high performance
  - Necessitates garbage collection, causing write amplification.
  - Write amplification is a critical factor affecting
    - Short random write performance
    - Endurance lifetime
- The impact of garbage collection on write amplification is influenced by
  - Level of over-provisioning
  - Choice of reclaiming policy
  - Type of workloads (we only consider uniformly-distributed random workload with 4KiB in this paper)

Related work

- Log-Structured Filesytem (LSF)
  - M. Rosenblum and J. K. Outsterhout (1992)
- Age-threshold algorithm for garbage collection in LSF
  - J. Menon and L. Stockmeyer (1998)
- Competitive analysis of wear-leveling algorithms
  - A. Ben-Aroya and S. Toledo (2006)
- Two comprehensive surveys:
  - Algorithms and data structures for flash memories by E. Gal and S. Toledo (2005)
  - Design tradeoffs for SSD performance by N. Agrawal, et al. (2008)
- Other works:
  - Real-time garbage collection, L.-P. Chang, et al. (2004)
  - Efficient static wear-leveling, Y.-H. Chang, et al. (2007)

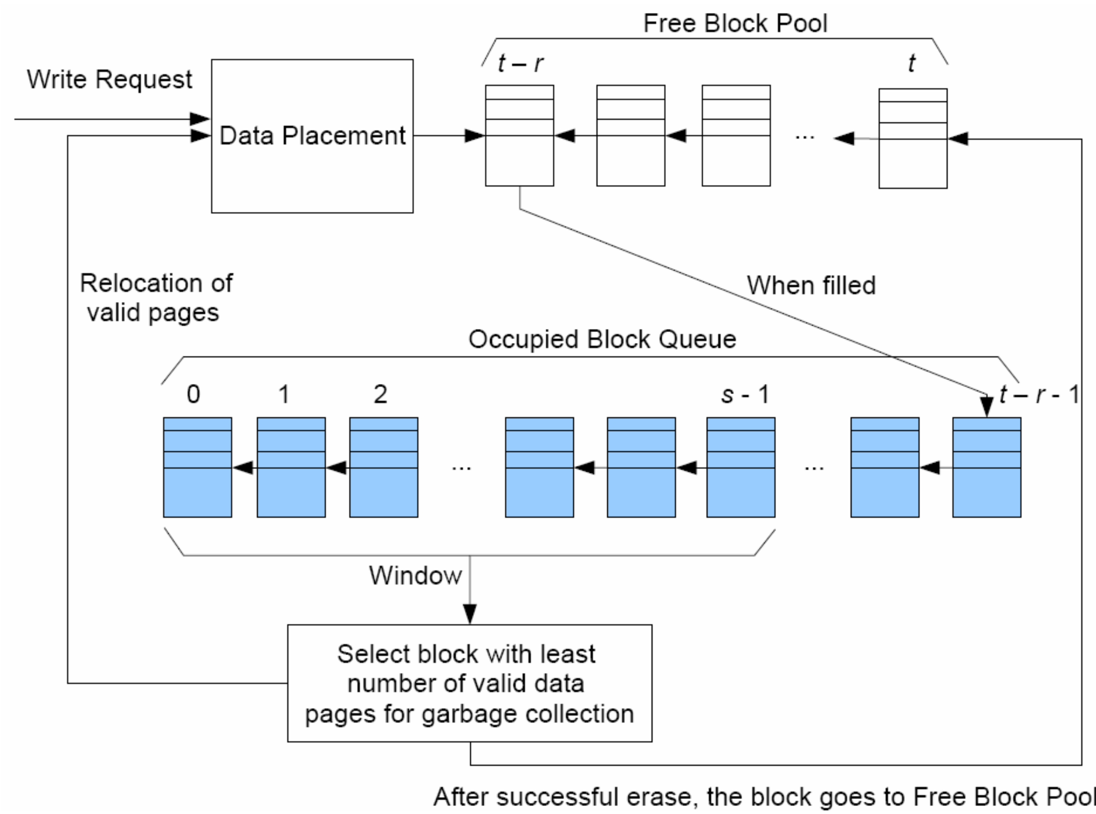
Definitions and Notations

- Write Amplification
  - In a relocate-on-write system, write amplification, *A*, due to garbage collection, is defined as the average of actual number of page writes per user page write
  - *A* is always greater than 1
- Write Amplification Factor
  - Define A_f as A-1. Namely $A_f=V/I$
- Over-Provisioning
  - Assume a raw storage capacity of *t* blocks of which the user can only use *u* blocks, the over-provisioning factor, O_f , is defined as $O_f = t/u$
  - the spare factor, S_f , is defined as $S_f =(t - u)/t$.



Garbage Collection Framework

- Garbage Collection Framework
  - Data placement
  - Free block pool (size *r*)
  - Occupied block pool
  - Reclaiming policy
- Windowed Greedy Reclaiming Policy
  - Configurable window size *s* to reduce complexity
  - Delaying garbage collection as much as possible, choosing *r* small
  - Restricting selection to the oldest *s* ($s < t - r$) blocks only



Garbage collection example

Analytical Model for Write Amplification Computation (I)

1. Denote by $p_0^*, p_1^*, \ldots, p_{np}^*$ the probability that the selected block has $0, 1, \ldots, n_p$ valid pages,

$$A_f = \frac{\sum_{k=0}^{n_p} k p_k^*}{n_p - \sum_{k=0}^{n_p} k p_k^*}.$$

2. Denote $p(V^{(j)} > k)$ the probability that the number of valid pages on block *j*, *(0<=j<=s-1)*, is greater than *k*

$$\begin{aligned} p(\forall_j V^{(j)} > k) &= p(V^{(0)} > k, V^{(1)} > k, \cdots, \\ &\quad V^{(s-1)} > k) \\ &\cong p(V^{(0)} > k) p(V^{(1)} > k) \cdots \\ &\quad p(V^{(s-1)} > k) \\ &= \prod_{j=0}^{s-1} p(V^{(j)} > k). \end{aligned}$$

3. Notice that

$$p(\forall_j V^{(j)} > k-1) = p_k^* + p_{k+1}^* + \cdots + p_{n_p}^*,$$

so that one can compute p_k^* by

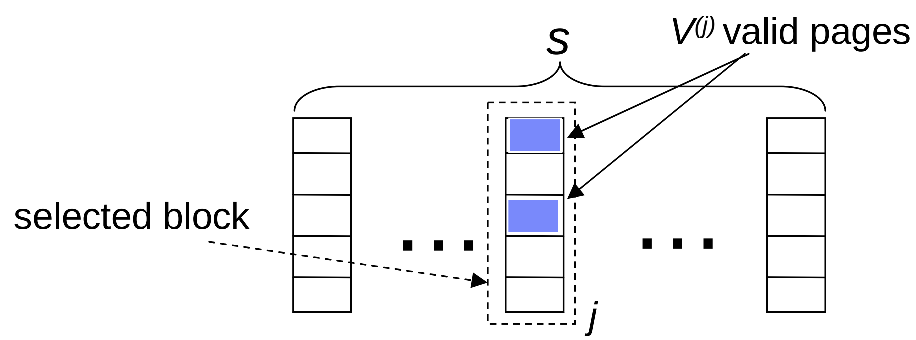
$$p_k^* = p(\forall_j V^{(j)} > k-1) - p(\forall_j V^{(j)} > k),$$

for k=1, …, $n_p - 1$. For $k = 0$,

$$p_0^* = 1 - p(\forall_j V^{(j)} > 0),$$

and for $k = n_p$,

$$p_{n_p}^* = p(\forall_j V^{(j)} > n_p - 1),$$



Analytical Model for Write Amplification Computation (II)

- Denote by $p_j(m)$ the probability that the j-th block has m valid pages, then

$$p(V^{(j)} > k) = 1 - \sum_{m=0}^{k} p_j(m).$$

- Consider $p_{i,j}$, the probability of i-th page on j-th block being valid,

$$\begin{aligned} p_{i,j} &= (1 - \frac{1}{un_p})^{[h(j)+(n_p-i-1)]} \\ &\approx (1 - \frac{1}{un_p})^{h(j)}, \end{aligned}$$

where $h(j)$ is the number of pages being written after the j-th block up to the $(t-r-1)$-th block that could invalidate this page. Then $p_j(m)$ can be approximated by a binomial function

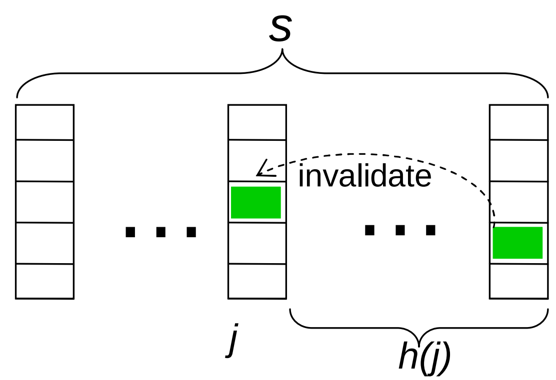
$$p_j(m) = \binom{n_p}{m} p_j^m (1-p_j)^{n_p-m}$$

- Two models to evaluate $h(j)$
  - “fixed” model

$$h(j) = \begin{cases} (t-r-u)n_p & \text{if } j \le u-1 \\ (t-r-j)n_p & \text{otherwise.} \end{cases}$$

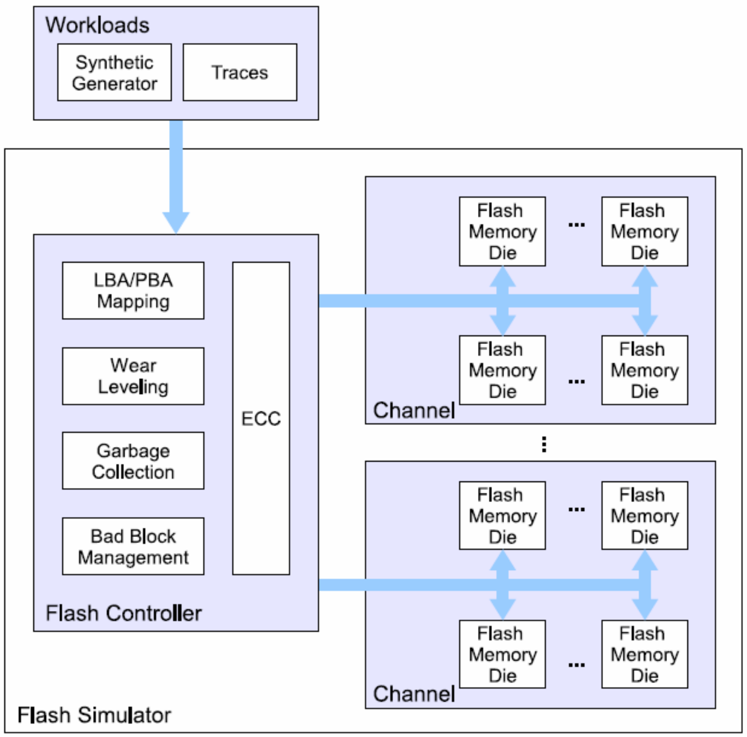
  - “cou

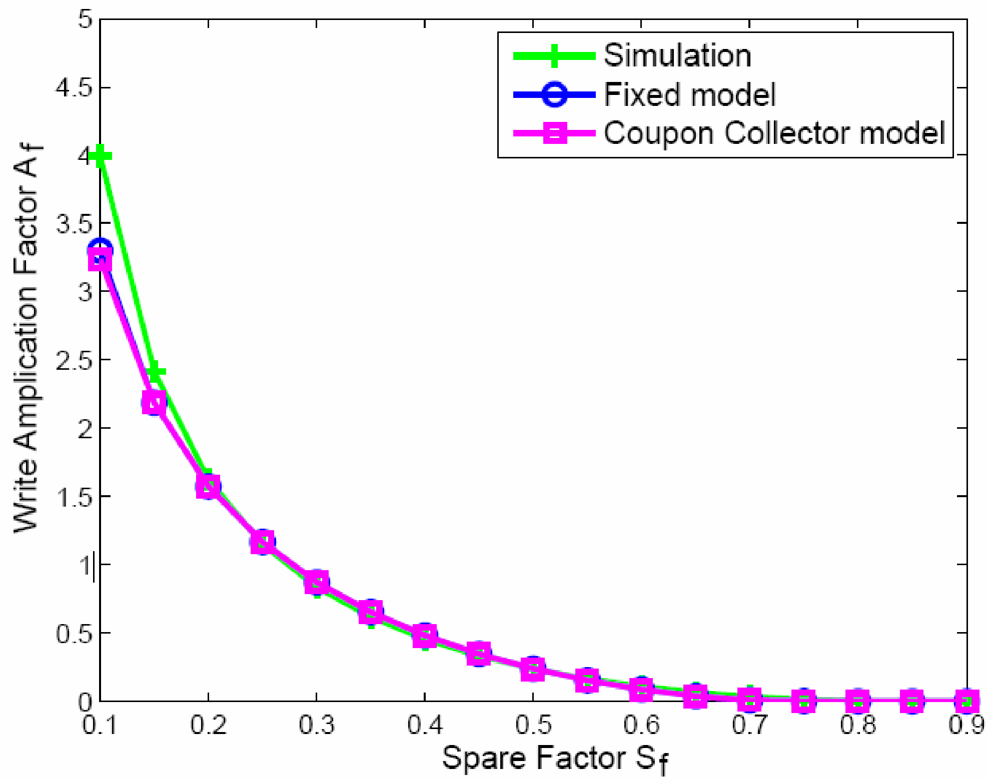
$$h(j) = \max\Big(0, n_p(t-r-j-1) - un_p\Big(\big(1 - \frac{1}{un_p}\big)^{[(j+1)n_p]}\Big)\Big)$$



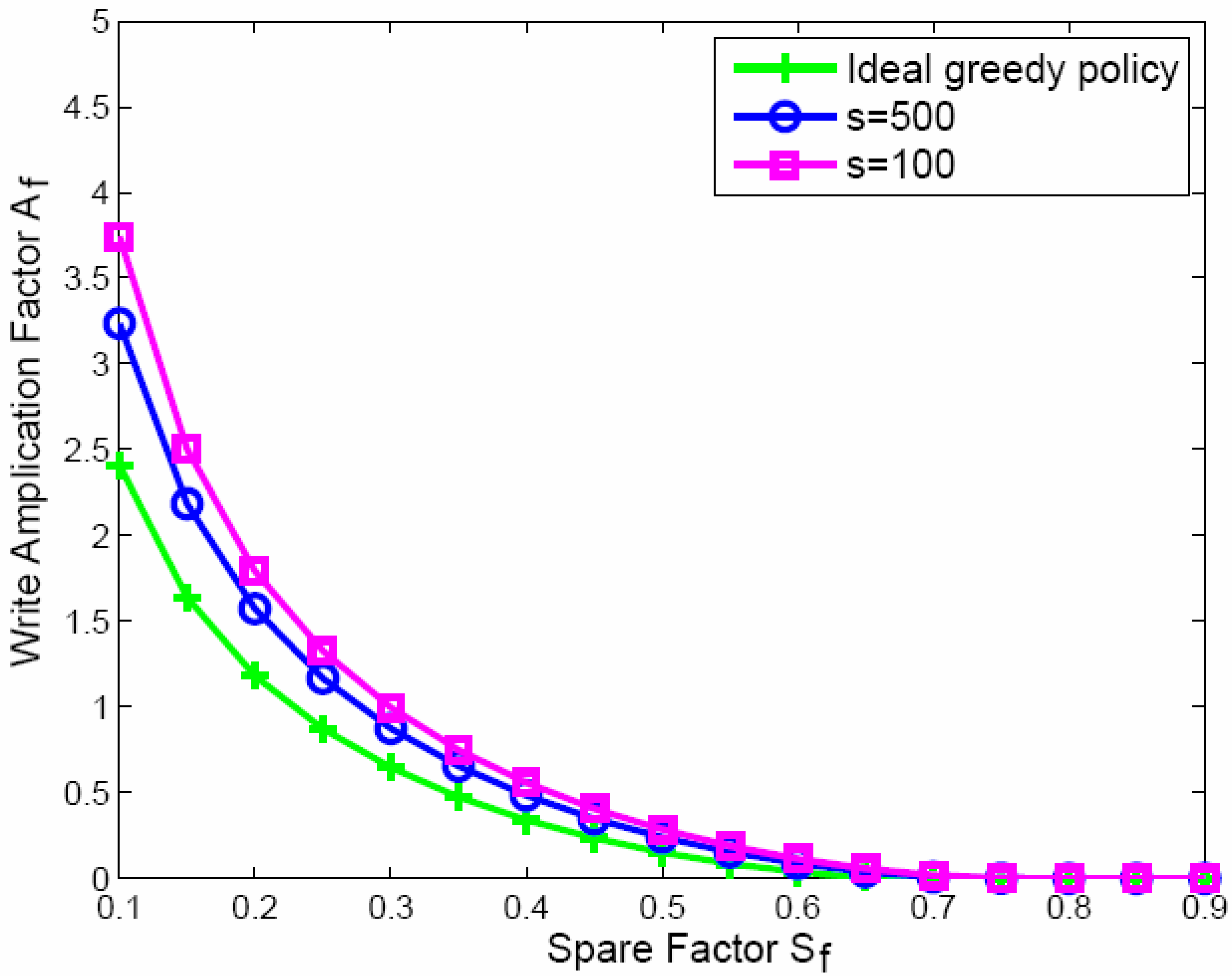
Simulation vs. Analysis

Parameter	Notation	Value
Total number of blocks	t	400000
Reserved number of blocks	r	10
Number of pages per block	n_p	64
Window size for applying reclaiming policy	s	500



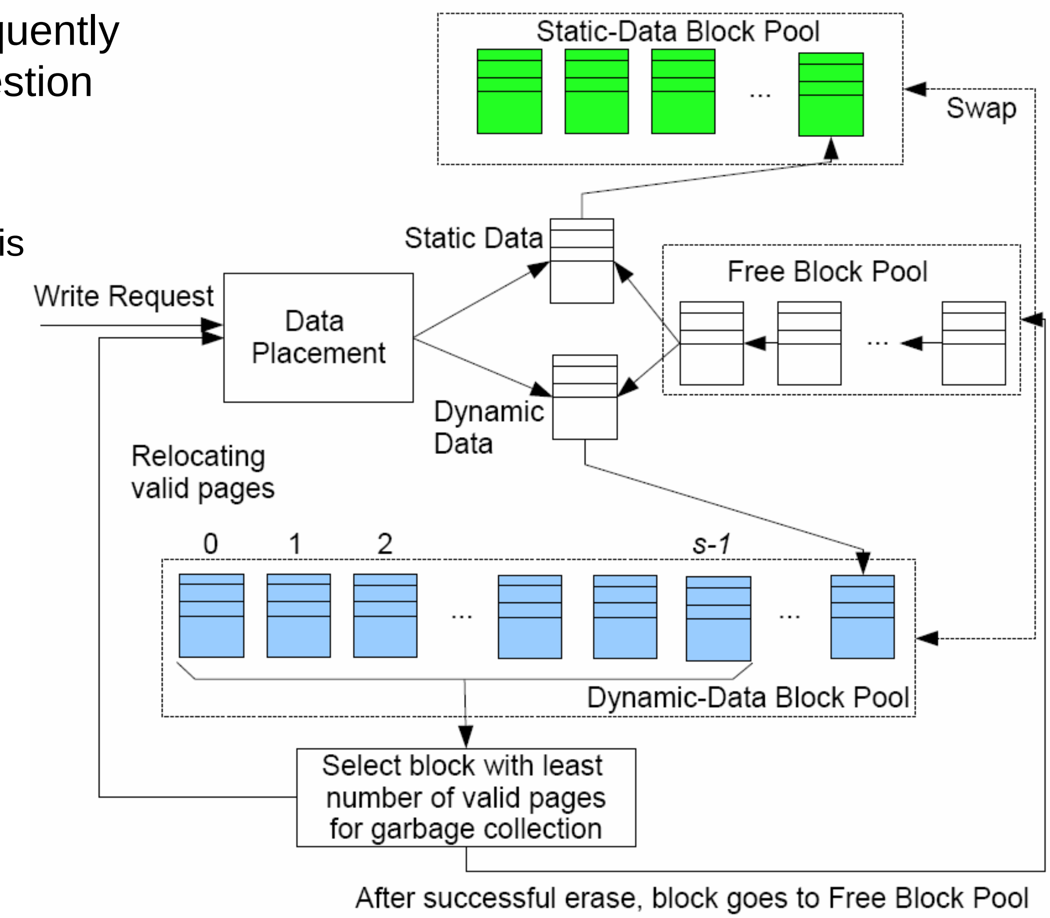


Analytical Results for Various Window Sizes

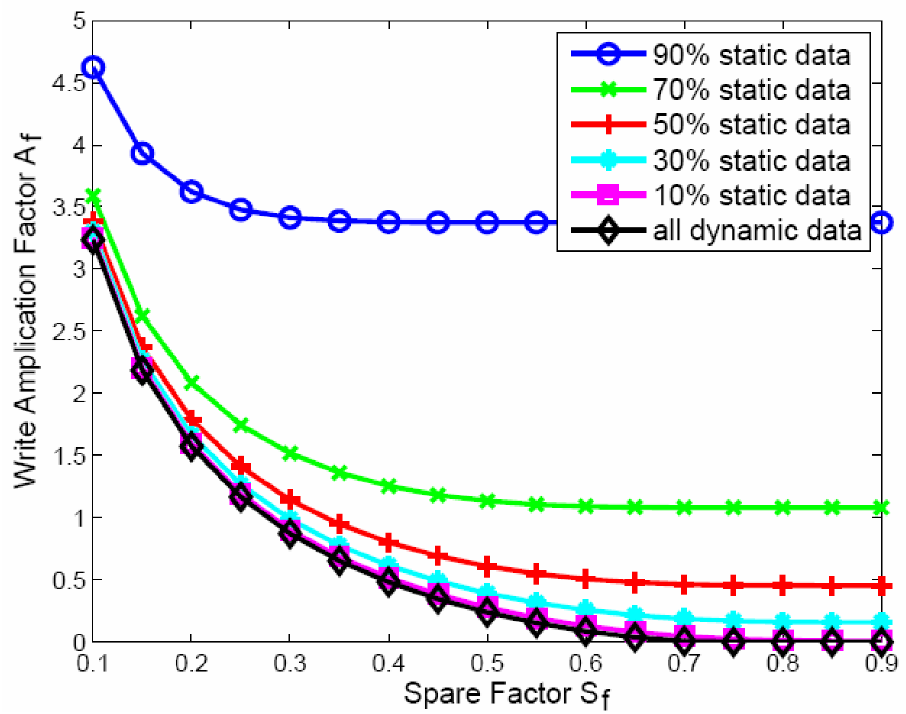


Impact of Separating Static Data from Dynamic Data

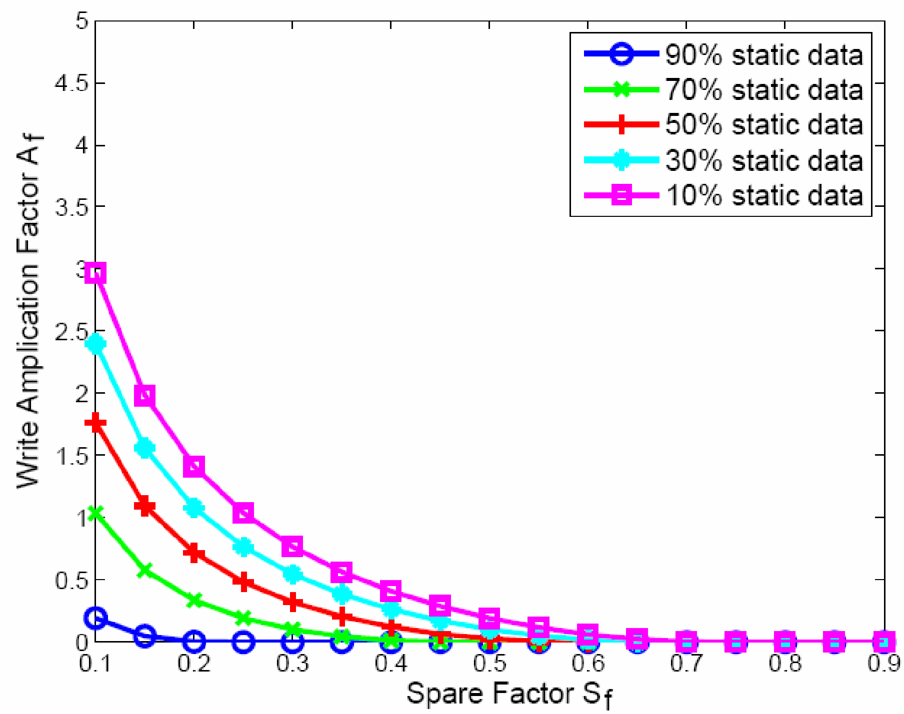
- Suppose we have a perfect information that part of the data is getting updated frequently and the rest is never updated, the question is how to place these data: mixed or separate?
  - From wear-leveling point of view, mixed is desirable
  - however, write amplification suffers.



Comparison



Mixed data placement



Separate data placement

Conclusions

- We have considered write amplification of SSD
  - Based on windowed greedy garbage collection policy
- We have assessed the magnitude of write amplification by simulation and analysis
  - Demonstrated that write amplification decreases as over-provisioning increases
  - Separating static and dynamic data reduces write amplification
  - Analytic results match with simulation for sufficiently large window sizes and typical spare factors in SSDs
- Future work
  - Garbage collection and wear-leveling co-existence