On the Optimality of Greedy Garbage Collection for SSDs

Yudong Yang, Vishal Misra, Dan Rubenstein
Columbia University
SSD Introduction

- Block
- Page
- Page has three states: valid, invalid, and empty

- write 16 pages for 8 write requests
  → write-amplification = 2
System Model

- $N$ blocks, $B$ pages in each block
- write-amplification: \( \frac{\text{overhead} + \text{data}}{\text{data}} \), or \( \lim_{M \to \infty} \frac{MB}{\sum_{j=1}^{M} i_j} \)
  
  $M$ : number of cleans
  
  $i_j$ : number of invalid pages on the $j$-th clean

smaller write-amplification \( \rightarrow \) better performance

Goal: Minimize write-amplification
Related works

• Greedy algorithm  Hu 2006, Bux 2010, Desnoyers 2012
  – The performance of greedy algorithm is analyzed
• Dual-greedy  Lin 2012
  – A heuristic algorithm optimized for traced data
• D-choice  Van 2013, Li 2013
  – D-choice algorithm is proposed and analyzed
• Optimality of greedy  Hass 2010
  – A sketch proof of optimality of greedy on random write workload.

Contribution

First formal proof of the optimality of greedy algorithm on memoryless workloads.
Memoryless workloads

All valid pages have equal probability of becoming the next invalid page.

– Page lifetimes: independent, exponential with identical rate $\mu$
Greedy GC Algorithm

The Greedy GC algorithm:

- waits until the SSD is filled.
- cleans a block with maximal number of invalids.
- ties are broken arbitrarily.
Clean-and-move system

• allows the valid pages to write on other blocks.
Optimality of Greedy

- Lemma 1. A block should only be cleaned when it is full (of valid and invalid pages).
- Proof:
  Cleaning can always be delayed while the block containing empties.

![Diagram](image-url)
Optimality of Greedy

Theorem 1. In clean-and-move systems with memoryless workloads, Greedy is optimal.

Proof: X: the block choose by another algorithm
Y: the block choose by greedy
Optimal of Greedy

Theorem 2. For memoryless workloads, there is no advantage to moving active pages between blocks.

Proof: Suppose we are moving n pages from X to Y upon an write request p

Y must has exact n empty pages, otherwise we can delay move. If p is placed on Y: move → clean Y → write p

The move operation can be delayed, so the sequence become clean Y → write p → move(upon next request)
Optimal of Greedy

X must have exact no empty pages, otherwise we can delay move.
If p is placed on X: move n pages from X to Y → clean X → write p in X
delay the move by a stochastic equivalence:
  write p in Y → move n-1 pages from X to Y (upon later arrival)
Discussion & Future work

- **D-choice variants**: choosing a block from a randomly selected set of size D
  Greedy still be the optimal on D-choice variants.

- **more general workloads**
  - not optimal for Rosenblum workloads and long-tailed workloads.  
    Lin 2012, Van 2013
  - conjecture: still optimal for short-tailed workloads.

- **Heuristics for general workload**