# Wear Leveling in SSDs Considered Harmful

#### **Ziyang Jiao**, Janki Bhimani, Bryan S. Kim Syracuse University, Florida International University





# Overview

- SSD background
- Wear-leveling in SSDs
  - Write amplification
  - WL effectiveness
- A capacity-variant SSD
- Evaluation
- Conclusion

# SSD background

- Limited lifetime of SSDs
  - Fixed program-erase limit
  - Blocks become bad when erase counts exceed the endurance limit
  - Unusable when physical space <= logical space</li>



A typical system architecture of flash devices.

# The role of wear leveling in SSDs

- Limited lifetime of SSDs
- WL in SSDs:
  - Equalize the amount of wear within the SSD
  - Static / Dynamic



The role of WL in SSDs.

# WL: boon or bane?

- Write amplification
  - Static: data relocation
  - Dynamic: lower cleaning efficiency
- Downward trend in endurance
  - Challenging to design an effective yet efficient wear leveler
  - Suboptimal WL decisions do more harm than good



The downward trend in SSD endurance.

## WL: boon or bane?

- Write amplification
- Downward trend in endurance
- Effective under limited scenarios
  - "Wear leveling is not perfect"



# The performance of WL under modern SSDs

- WL algorithms
  - *DP*
  - PWL
  - DAGC
- Synthetic workloads

   r/h: r fraction of writes occur on
   the h fraction of the workload
   footprint.

Name	Туре	Parameters	Principle	Comparisons
DP	Static	Fixed, a predefined threshold ( <i>TH</i> )	Hot-cold swapping	HC, 2L, EP, OBP
PWL	Static	Adaptive, an initial threshold $(THR_{int})$	Cold-data migration	BET and Rejuvenator
DAGC	Dynamic	Adaptive, no external parameters	Adjust GC victim	DTGC

Representative WL algorithms.

- L. Chang, "On efficient wear leveling for large-scale flash-memory storage systems," in ACM Symposium on Applied Computing (SAC), 2007.
- F. Chen, M. Yang, Y. Chang, and T. Kuo, "PWL: a progressive wear leveling to minimize data migration overheads for NAND flash devices," in Design, Automation & Test in Europe Conference & Exhibition, (DATE), 2015.
- Z. Chen and Y. Zhao, "DA-GC: A dynamic adjustment garbage collection method considering wear-leveling for SSD," in Great Lakes Symposium on VLSI (GLSVLSI), 2020.
- P. Desnoyers, "Analytic modeling of SSD write performance," in International Systems and Storage Conference (SYSTOR), 2012.

# The performance of WL under modern SSDs

- Investigate the following three aspects:
  - Write amplification
  - Effectiveness in equalizing the erase count
  - Behaviors under different access footprints



The write amplification under the synthetic workload of r/h = 0.9/0.1 on full logical space.



The write amplification under the synthetic workload of r/h = 0.9/0.1 on full logical space. The WL threshold parameter dictates the WA for DP.



The write amplification under the synthetic workload of r/h = 0.9/0.1 on full logical space. The overall write amplification can be as high as 11.49 for PWL(50).



The write amplification under the synthetic workload of r/h = 0.9/0.1 on full logical space.



The distribution of erase count when full logical address space is used after writing 25TiB, under r/h = 0.9/0.1 workload.



The distribution of erase count when full logical address space is used after writing 25TiB, under r/h = 0.9/0.1 workload with DP algorithm.



The distribution of erase count when full logical address space is used after writing 25TiB, under r/h = 0.9/0.1 workload with PWL algorithm.



The distribution of erase count when full logical address space is used after writing 25TiB, under r/h = 0.9/0.1 workload with DAGC algorithm.

![](_page_16_Figure_1.jpeg)

WL shows the performance anomaly under r/h = 0.9/0.1 workload with large footprint.

![](_page_17_Figure_1.jpeg)

The benefit from wear leveling is negligible compared to not running at all under r/h = 0.5/0.5 with large footprint.

# Small access footprint

![](_page_18_Figure_1.jpeg)

The distribution of erase count under a skew workload when only 5% of the logical address space is used. The red dot indicates the average erase count for NoWL.

#### Small access footprint

![](_page_19_Figure_1.jpeg)

The distribution of erase count under a skew workload when only 5% of the logical address space is used.

# Summary of findings

![](_page_20_Figure_1.jpeg)

# A capacity-variant SSD

- Forgoing wear leveling and instead adopting capacity variance (CV).
- The CV-SSD would gracefully reduce its exported capacity:

![](_page_21_Figure_3.jpeg)

A fixed capacity block device.

• Bryan S. Kim et al, "CPR for SSDs", HotOS 2019

![](_page_21_Figure_6.jpeg)

A capacity-variant device.

# A capacity-variant SSD

- The CV-SSD would gracefully reduce its exported capacity:
  - Wear leveling becomes unnecessary and thus incurs no write amplification overhead.
  - The lifetime of an SSD would be extended significantly.
  - It is easier to determine when to replace an SSD by checking the exported capacity.

- Evaluate the eight designs: – NoWL, DP, PWL, and DAGC on a fixed capacity SSD vs. on a capacity-variant SSD
- SSD is pre-conditioned into a steady-state prior to each experiment

Parameter	Value	Parameter	Value
Page size	4 KiB	Physical capacity	284GiB
Pages per block	256	Endurance limit	500
Block size	1 MiB	Over-provisioning	11%
Block allocation	FIFO	Garbage collection	Greedy

The summary of SSD configuration and policies.

• Nine I/O traces

![](_page_24_Figure_1.jpeg)

The lifetime comparison with eight different designs. Capacity variance without WL outperforms other designs.

![](_page_25_Figure_1.jpeg)

Workloads with a relatively small footprint.

![](_page_26_Figure_1.jpeg)

Workloads with a relatively small footprint.

![](_page_26_Figure_3.jpeg)

# Conclusion

- We present the write amplification caused by wear leveling (WL), instead of relying on a back-of-the-envelope calculation.
- We uncover that existing wear leveling techniques can exhibit anomalous behaviors under modern settings.
- We explore and quantify the benefits of capacity variance in an SSD that gracefully reduces its capacity as flash memories become bad.
- Codebase will be available soon https://github.com/ZiyangJiao/FTLSim-WL

Thank you!

Ziyang Jiao zjiao04@syr.edu