Generating Realistic Wear Distributions for SSDs

Ziyang Jiao
Syracuse University

Bryan S. Kim
Syracuse University

Abstract

NAND flash-based solid-state drives (SSDs) have already replaced hard disk drives (HDDs) as the mainstream storage device. Highly accurate and complex SSD development platforms, such as Amber [5], FEMU [12], and MQSim [18], that model various flash technologies are becoming increasingly used to conduct large-scale simulation-based research.

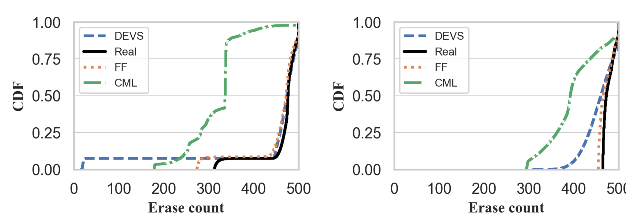
Due to the intrinsic NAND idiosyncrasies, flash memories wear out as they are programmed and erased, and progressively exhibit more errors as the wear accumulates [2, 13, 14]. Previous work has shown that SSDs present the *fail-slow* symptoms [6] and the internal wear state can have a significant impact on the performance results [2, 9, 13]. which in turn is affected by several factors, important amongst them are garbage collection, wear leveling, and external workloads. Unfortunately, it is not rare that SSD evaluations are performed under a fresh unworn or an unrealistic state [1, 4, 7, 10, 17].

However, generating realistic wear distribution for the purposes of evaluating SSDs is challenging. *Preconditioning* is a process to create representative internal states by applying workload to an SSD over a period of time prior to the actual evaluation phase. However, it is prohibitively expensive to rely on preconditioning to reach a meaningful wear state. For example, aging an SSD with three years worth of workloads would also take three years of time or even longer on existing SSD simulators and emulators.

In this work, we propose *Fast-Forwardable SSD*, a machine learning-based SSD aging framework that generates representative future wear-out states. However, the challenges of using a machine learning approach for making online, fine-grained inferences on SSD internal states are two-fold. First, the inference must be accurate. Modern SSDs are a complex embedded system, managing all of their internal resources with background operations such as garbage collection, wear leveling, error handling, and data scrubbing. We need to learn these internal complexities to make the inference highly accurate. Second, the inference must be fast and efficient relative to the simulation time; otherwise, it either brings negligible benefits or even prolongs the overall process, which indicates deep learning models like CNN or RNN that would introduce more complexities and computation overhead are not applicable anymore.

To address these, *FF-SSD* incrementally builds a lightweight regression model for each block to capture the changes in SSD-internal states and predicts their trajectory, using the information from past executions. This model would approximate the future wear state of an SSD device if the same workload were to be repeated, and is much faster than running the repeated workload. We evaluate our design using real-world workloads [8, 11] and Figure 1 shows our preliminary results on FTLSim [3]: with one-third workload saving compared to full simulation, *FF-SSD* continuously learns the behavior within the SSD using the information from the past two iterations of the workload, and then predicts the wear state after one additional iteration. *FF-SSD* generates the final states of SSD, and achieves the highest accuracy compared to two prior works, DEVS [16] and C-ML [15].

The following directions will be investigated for future work: (1) deeper analysis of *FF-SSD* over various workloads (2) validating *FF-SSD* with different platforms and SSD models (i.e., OC-SSD, ZNS-SSD) (3) different methods to increase both prediction accuracy and aging efficiency.



(a) Developer Tools Release [8]. (b) Virtual Desktop Infra [11].

Figure 1: Evaluation under real-world workloads. *FF-SSD* achieves the highest accuracy (91% for DTRS and 97% for VDI) compared to DEVS (87% for DTRS and 93% for VDI) and C-ML (68% for DTRS and 83% for VDI). The accuracy is computed using the mean difference in erase counts across all blocks relative to their real values from full simulation.

References

[1] Rajiv Agarwal and Marcus Marrow. A closed-form expression for write amplification in NAND flash. In *2010 IEEE Globecom Workshops*, pages 1846–1850, 2010.

[2] Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu. Error characterization, mitigation, and recovery in flash-memory-based solid-state drives. *Proceedings of the IEEE*, 105(9):1666–1704, 2017.

[3] Peter Desnoyers. Analytic modeling of SSD write performance. In *International Systems and Storage Conference (SYSTOR)*, 2012.

[4] Peter Desnoyers. Analytic models of SSD write performance. *ACM Trans. Storage*, 10(2), Mar. 2014.

[5] Donghyun Gouk, Miryeong Kwon, Jie Zhang, Sungjoon Koh, Wonil Choi, Nam Sung Kim, Mahmut Kandemir, and Myoungsoo Jung. Amber*: Enabling precise full-system simulation with detailed modeling of all SSD resources. In *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 469–481, 2018.

[6] Haryadi S. Gunawi, Riza O. Suminto, Russell Sears, Casey Golliher, Swaminathan Sundararaman, Xing Lin, Tim Emami, Weiguang Sheng, Nematollah Bidokhti, Caitie McCaffrey, Deepthi Srinivasan, Biswaranjan Panda, Andrew Baptist, Gary Grider, Parks M. Fields, Kevin Harms, Robert B. Ross, Andree Jacobson, Robert Ricci, Kirk Webb, Peter Alvaro, H. Birali Runesha, Mingzhe Hao, and Huaicheng Li. Fail-slow at scale: Evidence of hardware performance faults in large production systems. *ACM Trans. Storage*, 14(3), Oct. 2018.

[7] Xiao-Yu Hu, Evangelos Eleftheriou, Robert Haas, Ilias Iliadis, and Roman Pletka. Write amplification analysis in flash-based solid state drives. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, SYSTOR '09. Association for Computing Machinery, 2009.

[8] Swaroop Kavalanekar, Bruce L. Worthington, Qi Zhang, and Vishal Sharda. Characterization of storage workload traces from production windows servers. In *International Symposium on Workload Characterization (IISWC)*, 2008.

[9] Bryan S. Kim, Jongmoo Choi, and Sang Lyul Min. Design tradeoffs for SSD reliability. USENIX Association, 2019.

[10] Jihun Kim, Joonsung Kim, Pyeongsu Park, Jong Kim, and Jangwoo Kim. SSD performance modeling using bottleneck analysis. *IEEE Computer Architecture Letters*, 17(1):80–83, 2018.

[11] Chunghan Lee, Tatsuo Kumano, Tatsuma Matsuki, Hiroshi Endo, Naoto Fukumoto, and Mariko Sugawara. Understanding storage traffic characteristics on enterprise virtual desktop infrastructure. In *ACM International Systems and Storage Conference (SYSTOR)*, 2017.

[12] Huaicheng Li, Mingzhe Hao, Michael Hao Tong, Swaminatahan Sundararaman, Matias Bjørling, and Haryadi S. Gunawi. The case of FEMU: Cheap, accurate, scalable and extensible flash emulator. USENIX Association, 2018.

[13] Neal R. Mielke, Robert E. Frickey, Ivan Kalastirsky, Minyan Quan, Dmitry Ustinov, and Venkatesh J. Vasudevan. Reliability of solid-state drives based on NAND flash memory. *Proceedings of the IEEE*, 105(9):1725–1750, Sep. 2017.

[14] Open NAND Flash Interface. ONFI 5.0 spec. http://www.onfi.org/specifications/, 2021.

[15] Daniel Christopher Powell and Björn Franke. Using continuous statistical machine learning to enable high-speed performance prediction in hybrid instruction-/cycle-accurate instruction set simulators. Association for Computing Machinery, 2009.

[16] Hesham Saadawi, Gabriel Wainer, and German Pliego. DEVS execution acceleration with machine learning. In *2016 Symposium on Theory of Modeling and Simulation (TMS-DEVS)*, pages 1–6, 2016.

[17] Hui Sun, Xiao Qin, Fei Wu, and Changsheng Xie. Measuring and analyzing write amplification characteristics of solid state disks. In *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 212–221, 2013.

[18] Arash Tavakkol, Juan Gómez-Luna, Mohammad Sadrosadati, Saugata Ghose, and Onur Mutlu. MQSim: A framework for enabling realistic studies of modern multi-queue SSD devices. USENIX Association, 2018.