Far-and-Near: Co-Designed Storage Reliability Between Database and SSDs

Jinwoo Jeong
Kibin Park
Hanyang University
South Korea

Sangjin Lee
Philippe Bonnet
IT University of Copenhagen
Denmark

Alberto Lerner
Philippe Cudre-Mauroux
University of Fribourg
Switzerland

ABSTRACT

Storage media is inherently unreliable. This is a crucial issue for database management systems (DBMS) that promise reliable storage to applications. Today, storage reliability is tackled independently at two different layers. First, Solid-State Drives (SSDs) must comply with strict reliability requirements. Second, database systems must deal with rare SSD errors or potential SSD failures. In this abstract, we outline the case for co-designed storage reliability across DBMS and SSDs.

Our argument is motivated by two recent observations. First, SSDs have experienced tremendous diversification to support eclectic workloads driven by Artificial intelligence, Cloud Computing and Big Data. The simplistic model in which SSDs deliver the same reliability characteristics independently of the application is no longer practical. Second, due to the tremendous repercussions of the end of Moore's Law and Dennard's scaling, the hardware community has coalesced around specialization as a way to move forward [3], and co-designing is a flavor of specialization that has been gaining increased traction (e.g., [2]). It is in this context that we revisit the nature of the storage reliability contract between database systems and SSDs.

Within an SSD, the storage medium is typically NAND flash, packaged into flash chips. An SSD is composed of tens of flash chips connected in parallel to a controller via multiple shared channels. NAND flash is notorious for being an unreliable medium. We denote by Bit Error Rate (BER) the fraction of bits that are read incorrectly [4]. Within an SSD, these errors result from the mapping between the digital representation within a computer and the analog representation on NAND flash, and from interferences, leaks or faults within the storage medium. The BER of NAND flash varies in time and depending on the number of writes. The average BER for NAND flash is in the order of 10^{-4}, i.e., one bit is flipped at roughly every 10K bits read [1]. Such error rate is alarmingly high.

SSD customers, however, are shielded from its implications because SSDs must comply with a fixed reliability target. This target is defined by the JEDEC's JC-64.8 Subcommittee[1]. Their SSD Requirement standard (JESD219) distinguishes enterprise-grade and customer-grade SSDs, and defines for each class, a target BER over the lifetime of the SSD: 10^{-15} for customer-grade SSDs and 10^{-16} for enterprise-grade SSDs. Put differently, SSDs achieve over ten orders of magnitude improvement in terms of reliability compared to the underlying NAND flash.

To bridge that gap, SSDs rely on Error Correction Codes (ECC) [5, 6] that detect and correct bit errors. Yet, database systems must account for the non-zero probability of bit errors and for SSD failures. As a result, they implement data recovery techniques (e.g., replication, RAID or Reed-Solomon codes [5]). Both ECC and data recovery techniques introduce redundancy in the form of parity bits that are associated with the original data to enable error detection, error correction, or tolerance to failure [5].

The strict layering of ECC (at the SSD level) and data recovery techniques (at the database level) provides a separation of concerns. However, it also introduces inefficiencies. The ECC component within each SSD does not recognize RS parity blocks, so it computes parity over parity. Conversely, the parity from the ECC component is inaccessible to the database system. Intuitively, this redundancy of parity computations wastes resources, increases I/O latency, and increases data movement.

The issue is more profound than simple inefficiencies. While NAND Flash is structurally designed to write data at page-size granularities, e.g., 4 or 16KB, data can be read at finer granularity. Since a significant performance component is associated with the amount of data to transfer, reading smaller chunks of data could deliver lower latency and better goodput. However, the current *one-size-fits-all* way to implement ECC on SSDs makes it impossible for SSDs to offer this option to applications.

We postulate that a new reliability contract between DBMS and SSDs based on co-design rather than strict layering can reduce I/O latency, data movement, space amplification and resource usage while providing stronger reliability than current solutions. Such co-design can support *far ECC*, where ECC is partly performed above SSDs, i.e., inside the database. It can also support *near ECC*, where ECC is done in an earlier stage than usual to accomodate moving database computations deep into the device's architecture. This requires DBMS system designers to treat ECC as a first-class citizen.

REFERENCES

[1] Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu. Errors in Flash-Memory-Based Solid-State Drives: Analysis, Mitigation, and Recovery. *arXiv:1711.11427 [cs]*, January 2018. arXiv: 1711.11427. URL: http://arxiv.org/abs/1711.11427.

[2] Cliff Grossner. Announcing a New Hardware-Software Co-design Strategy. URL: https://www.opencompute.org/blog/open-compute-project-foundation-announces-a-new-hardware-software-co-design-strategy.

[3] John L Hennessy and David A Patterson. A new golden age for computer architecture. *Communications of the ACM*, 62(2):48–60, 2019. doi:10.1145/3282307.

[4] Neal Mielke, Todd Marquart, Ning Wu, Jeff Kessenich, Hanmant Belgal, Eric Schares, Falgun Trivedi, Evan Goodness, and Leland R. Nevill. Bit error rate in NAND Flash memories. In *2008 IEEE International Reliability Physics Symposium*, pages 9–19, April 2008. ISSN: 1938-1891. doi:10.1109/RELPHY.2008.4558857.

[5] Todd K. Moon. *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley, Hoboken, NJ, 2nd edition edition, December 2020.

[6] Frederic Sala, Kees A. Schouhamer Immink, and Lara Dolecek. Error control schemes for modern flash memories: Solutions for flash deficiencies. *IEEE Consumer Electron. Mag.*, 4(1):66–73, 2015. doi:10.1109/MCE.2014.2360965.

[1] https://www.jedec.org/committees/jc-648