Replicate, Reduce, Recycle: Extending the Lifetime of Flash Memory

Kuei Sun* (kuei.sun@utoronto.ca), Jamie Liu* (jamie.liu@utoronto.ca), Ryan Yang* (ryan.yang@utoronto.ca)
Ashvin Goel (ashvin@eecg.toronto.edu, Angela Demke Brown (demke@cs.toronto.edu)
University of Toronto

1 Abstract

Flash memory has seen a dramatic increase in popularity in recent years, fueled by increasing capacity and decreasing cost per bit. Multiple-level cell (MLC) NAND flash, in which a single memory cell stores multiple bits, delivers the highest capacity at the lowest cost. However, MLC flash memory wears out an order of magnitude faster than single-level cell (SLC) technology. Typical manufacturer specifications report 100K erase cycles for SLC flash, but only 10K cycles for MLC flash. Although some experimental results indicate endurance may be much higher in practice, the limited lifetime of MLC flash remains a concern for write-intensive deployments, such as storage servers, limiting the use of flash memory in data centers. Moreover, as process technology shrinks, the write endurance is expected to continue to degrade, and this effect is expected to be more severe for MLC chips.

While enterprise storage solutions have typically employed high-endurance SLC flash technology, recently some vendors have begun to offer enterprise-class products based on MLC flash memory. Techniques such as wear-leveling, write coalescing and over-provisioning are used to mitigate the poor endurance of NAND flash. The first two reduce the number of program/erase cycles seen by any block, while the last is used to replace a failed block with a spare backup block. Once the device runs out of backup blocks, the drive is failed and must be replaced.

Our work explores whether these "failed" drives can actually still be used, by designing *replication* schemes that are tailored to the ways that flash memory fails. Although the increased data redundancy *reduces* the available storage capacity, the drives can continue to be used for many times longer than their original lifetimes, allowing them to be effectively *recycled*.

We present experimental results that explore the failure properties of MLC flash memory *after* the first error is detected in a block. We use a microcontroller to read and write flash memory directly, so that the underlying behavior is not masked by a Flash Translation Layer [**?**]. Based on our findings and the properties of flash memory, we design and evaluate two redundancy schemes to increase the lifetime of these devices, with some loss of capacity. With these schemes, we are able to increase endurance by roughly an order of magnitude.

The first scheme takes advantage of the way that data pages are mapped to MLC cells. Specifically, a pair of pages share the same set of cells, with the first page stored in the most significant bit (MSB) of each cell, and the second page stored in the least significant bit (LSB) of each cell. By writing a single page of data to both pages in the pair, we only use the '00' and '11' voltage levels (on reads, the two pages are combined by OR'ing them together, since we observe the dominant error mode to be 1's inadvertently flipped to 0's). Intuitively, the result should be similar to an SLC cell, since only a single bit needs to be distinguished in each cell. Experimentally, we observed that this strategy allows increasing the usable lifetime of the device by over 6X.

The second redundancy scheme is designed to mitigate the problem of program disturbs, wherein a cell is "accidentally" programmed from a "1" to a "0" due to charge leakage when programming a nearby cell. We verified experimentally that some bit patterns were more vulnerable to program disturbs than others (for example, the pattern "010" subjects the "1" bit to stresses from both sides). To alleviate this problem, we use a "forbidden pattern" encoding to ensure that 1's are never surrounded by 0's. Our encoding simply replicates each bit in the original data, storing a single page of data across two pages. For example, "010" becomes "001100". We find that this scheme has similar benefits to the first, although for different reasons.

We are currently expanding our test coverage to examine the failure characteristics of additional devices, and under more conditions. We are also exploring other encodings that may yield similar lifetime improvements with smaller capacity overheads. The current replication strategies demonstrate enormous potential for "recycling" failed MLC SSDs.

This poster does not include a demo.

*Student author