Verifying the Flash Device Drive

Yichi Zhang
yz321@cam.ac.uk
University of Cambridge, Computer Laboratory

UNIVERSITY OF CAMBRIDGE

Flash Memory

Flash memory is a type of electrically erasable programmable read-only memory. It is nonvolatile (retains its content without power), so it is widely used in portable devices.

However, compare to other programmable memory, flash devices suffer from **two limitations**:

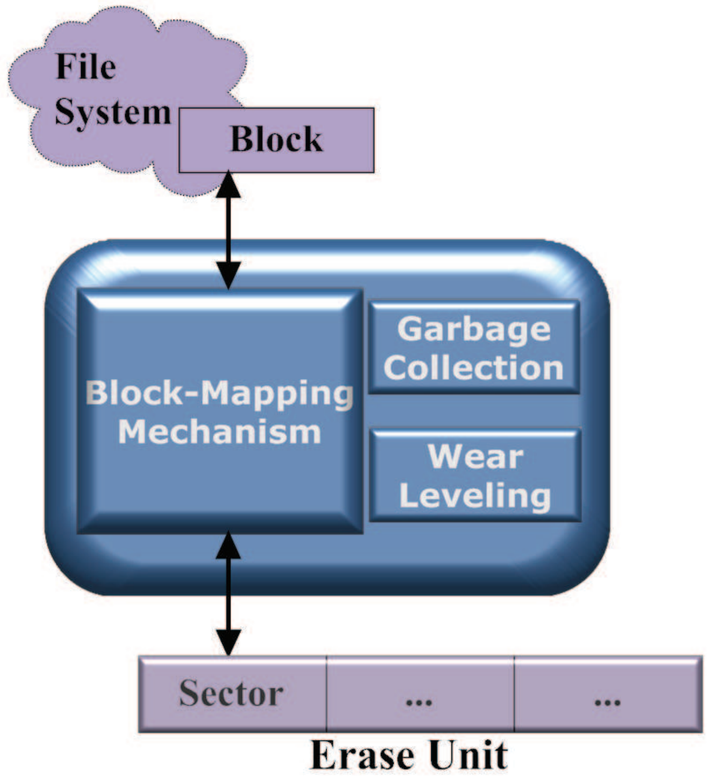
Memory Wear: the memory cells in flash can be written to only a limited number of times, between 10,000 and 1,000,000, after which they wear out and become unreliable.

Regional Erasure: write operation in flash can only clear bits (change their value from 1 to 0). The only way to set bits (change their value from 0 to 1) is to erase the entire region memory. These regions, typically ranging from several kilobytes to hundreds of kilobytes, are called erase units.

Flash Device Driver

- Presenting the flash to higher-level software (file system) as a rewritable block device.
- Has more general impact than flash-specific system since it is platform independent.
- The block mapping mechanisms of flash drivers have been reused in some flash systems with tiny modification.
- The standardization of the flash device driver provides a unified open interface for garbage collection and wear leveling, which is crucial for the analysis of those algorithms.

Flash Translation Layer



The mapping technique maps virtual blocks onto the flash sectors. Instead of overwriting the sector where the block is currently stored, simply writing new data to another sector and then updating the block-to-sector map. Wear leveling method could be applied when the system needs a new sector. Over time, garbage collection will be used to reclaim obsolete sectors.

Verification Plan

My plan is to verify the FTL with a highly abstracted garbage collection model first. This highly abstracted model only contains 3 major steps and will not specifying the exact unit for reclamation:

1. Reclaim a unit
2. Relocate valid data
3. Update related map*(may require different models)

The second part is trying to examine the correctness and effectiveness of some wear leveling algorithms.

Evaluation Challenge

It is extremely hard to the define correctness of wear leveling algorithms. But in general, we can prove some properties:

- ***there is no blind spot:*** with enough write operations, in any status, any unit should be reclaimed eventually.
- ***there is no risk of losing data :*** in case of power failure, the system always has a valid copy of target piece of data.

Rather than correctness, effectiveness is more interesting:

- ***Clearly we have to deal with probability:*** a wear-leveling algorithm may itself be deterministic, but we want to be able to establish that the probability of the algorithm failing as a whole is below some threshold.