Using the Appropriate Wear Leveling to Extend Product Lifespan

Presented by: Bill Roman
Datalight Software Architect

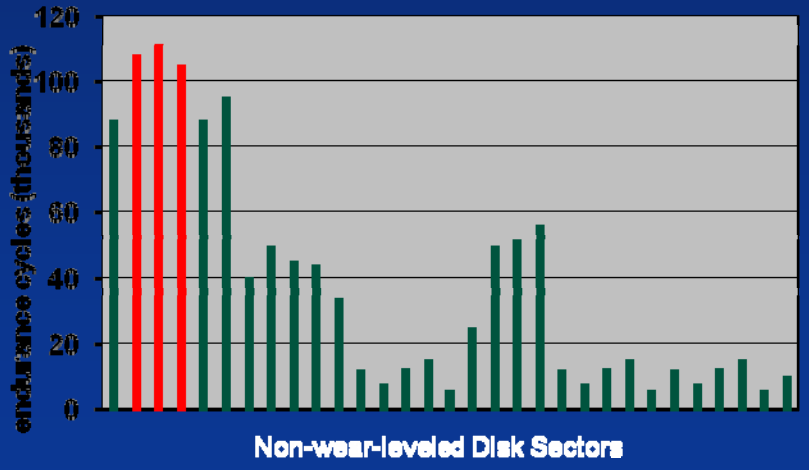
Santa Clara, CA USA
August 2009

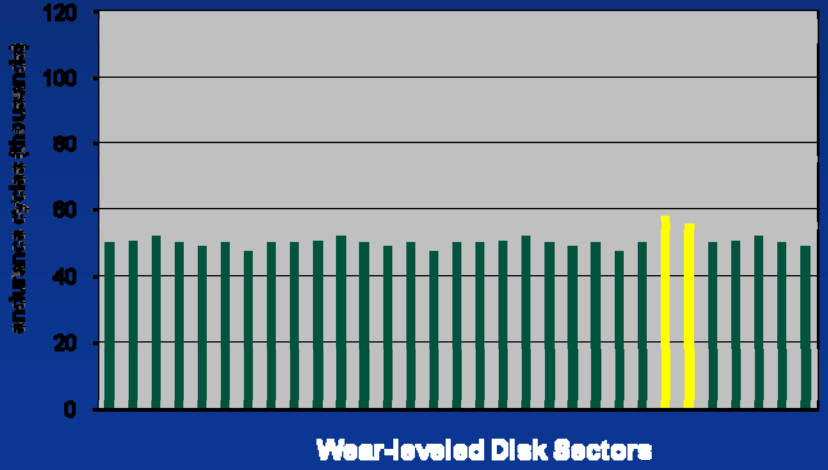
Overview

- Basics of wear leveling
- Dynamic vs. Static Wear Leveling
- Other Considerations
- Conclusions

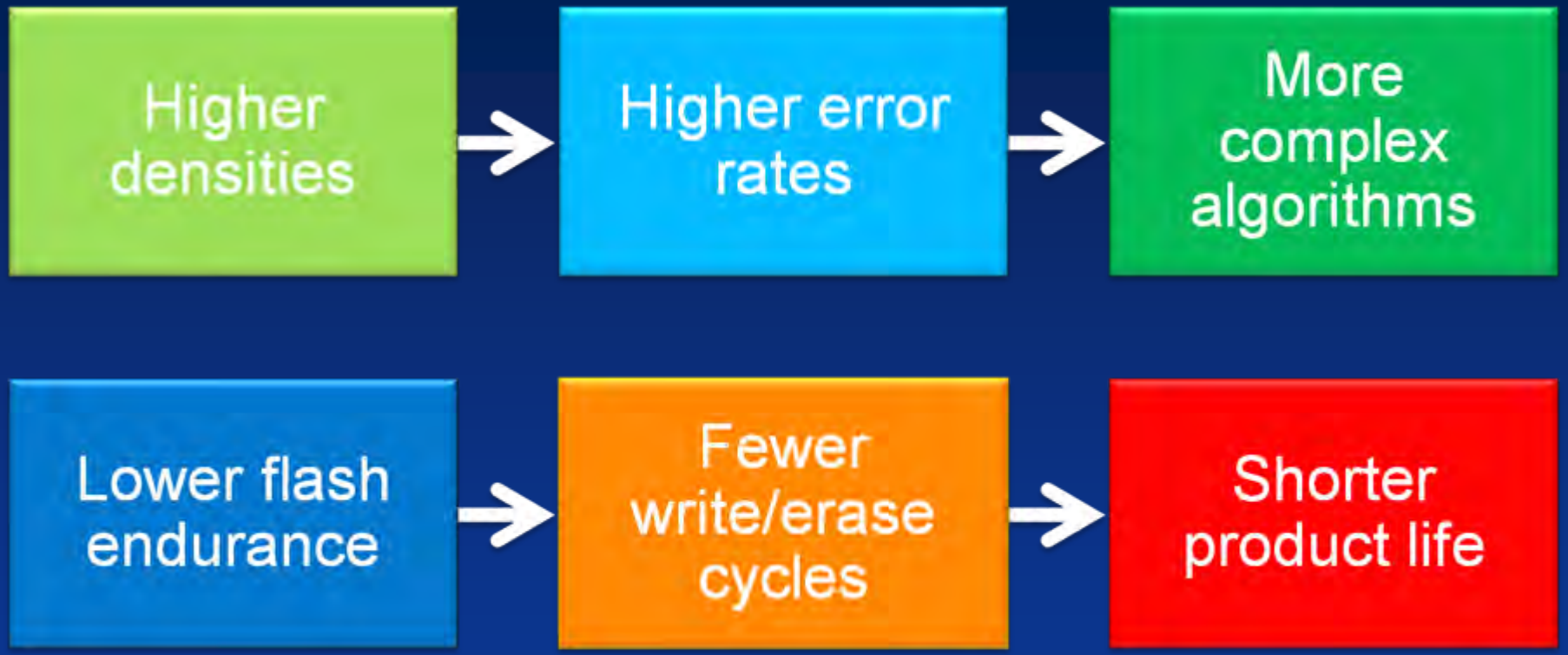
What is Wear Leveling?

Wear leveling: a set of algorithms that attempt to maximize the lifetime of flash memory by evening out the use of individual cells.





Growing Impact on Product Life

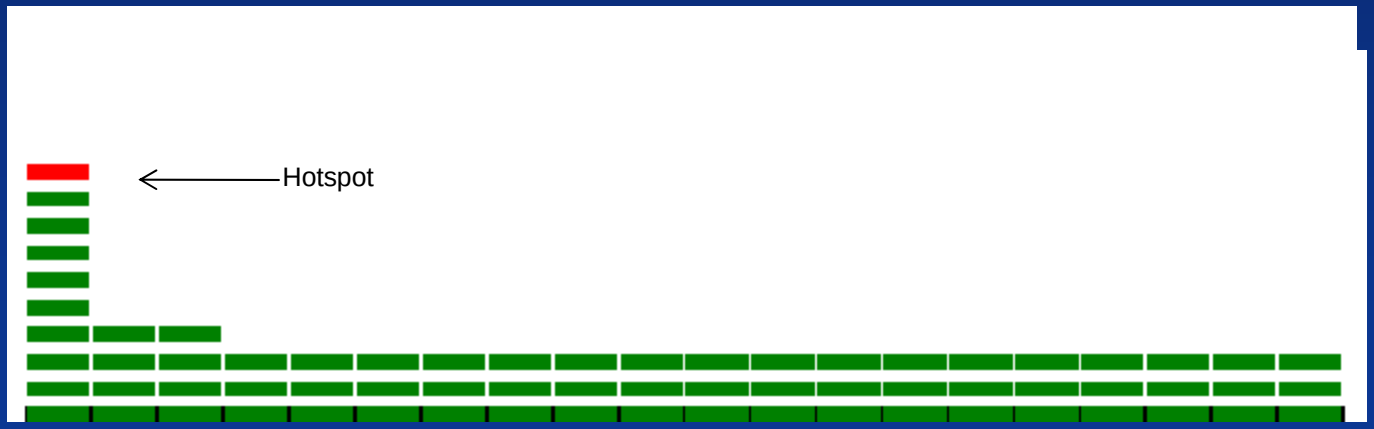


Ignorance is Costly

- Controller using read-modify-write operations
- If there is no wear leveling, an area of the flash can become unusable in just a few months
- Wear Leveling cannot be ignored, but does not need to be a ‘big deal’

Conventional File System on Flash

- Inefficient
- Hot spots
- Not interruption-safe
- Usable for read-only

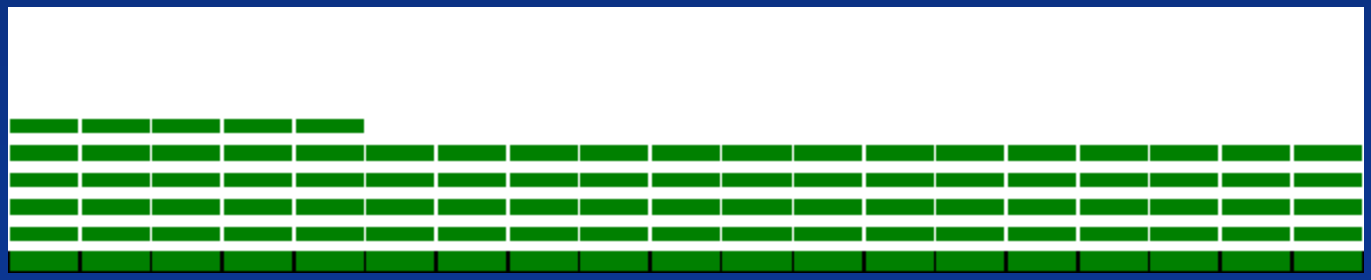


Dynamic is Usually Standard

- Largely inherent in any flash management system
  - Overwrites in flash are generally not allowed
  - Writes to NAND must be sequential
  - Requires data to be moved to efficiently use NAND

FIFO

- Original JFFS used this strategy
- Strictly linear
- “Perfect” wear-leveling



FIFO

Immediately after erasing

Erase Block

FIFO

A file is written

Erase Block

FIFO

Data in the file is modified

Erase Block

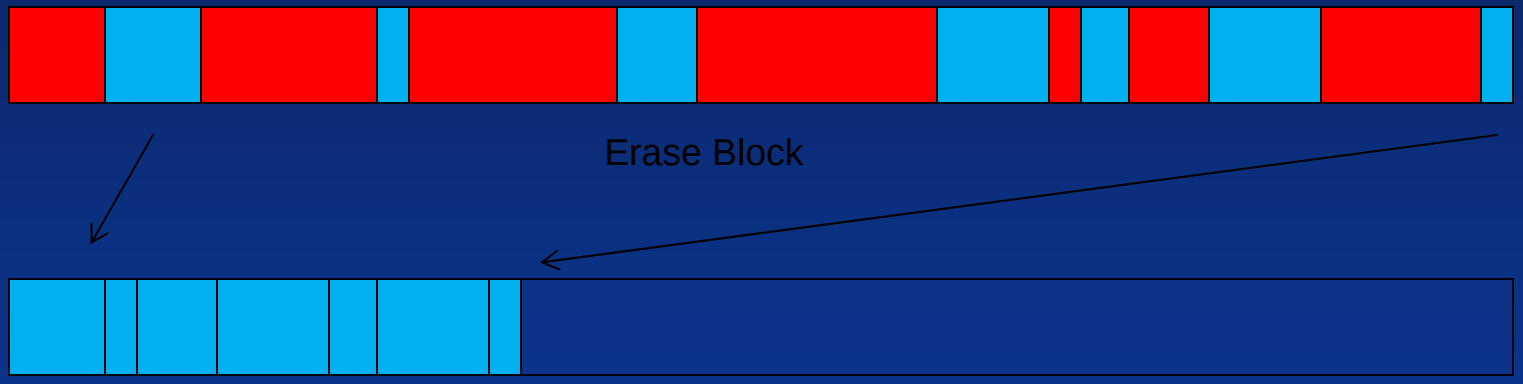
FIFO

More files are created, modified, deleted

Erase Block

FIFO

So it’s necessary to compact valid data into a free erase block



FIFO

Becomes

Simple Cuts Both Ways

- Effectiveness is based on application use case
  - How much data is unchanged?
  - Only works on areas of flash which are frequently updated
  - Static data therefore reduces life of the flash

Effect of FIFO on Flash Life

- Actual Life = Specified Life * (1- static data size/total flash size)
- Balance “perfect” wear leveling against efficiency by occasionally moving static data

Dynamic Considerations

- Ignorance is bliss
  - The system can ignore what it does not know
  - If data is never written, it is not considered (remains unmoved)
  - Decisions to move data can be based purely on performance needs

Static Does the Heavy Lifting

- Static WL is most effective
  - Largely independent of the systems' use case
  - Large (or all) portions of the media are considered
  - Static data does not significantly degrade life of the flash

Static Complicates Things

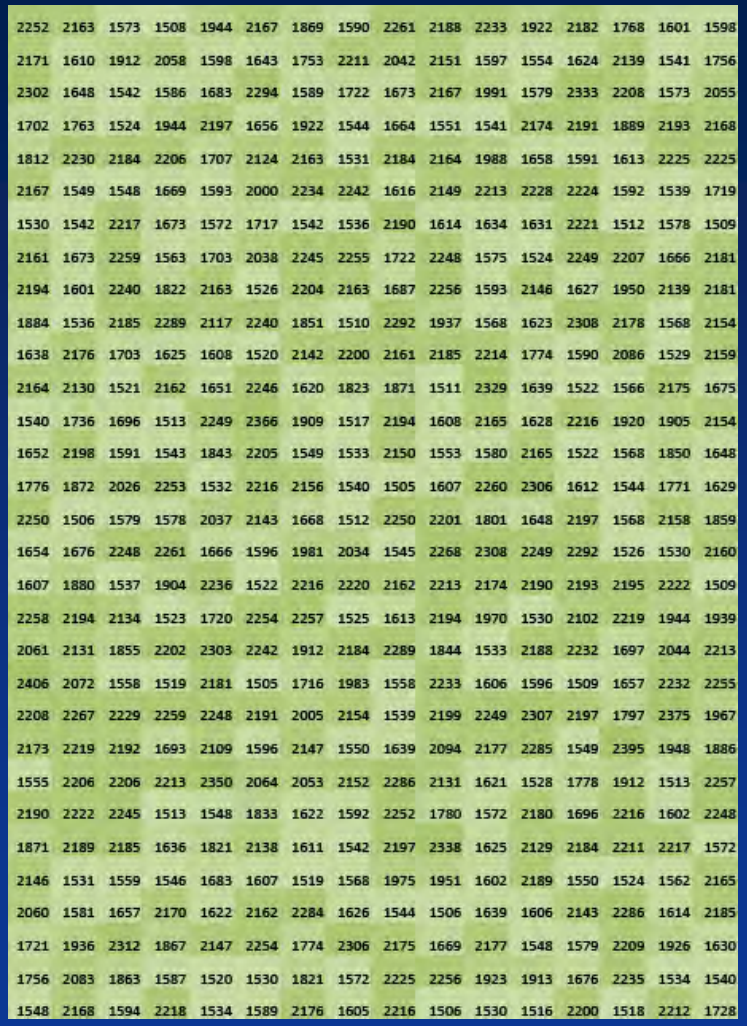
- Implementations are more complex
  - The system must track media usage and evaluate otherwise unused media
  - Requires additional erase and copy operations with zero return to performance
  - Impacts to performance can be substantial

Static and Dynamic Together

- Both approaches should be evaluated for an effective wear leveling solution.
- Dynamic is low effort and high performance
- Static ensures that media life is maximized, but at a cost to performance
- Statistical implementations may not handle certain use cases or break down entirely

Dynamic vs. Static

Dynamic Only



Static + Dynamic

Do I really need to care about Wear Leveling?

- Use case description
  - System is a 40GB SSD in a notebook running Windows XP
  - OS and applications account for 18GB and 12GB respectively
  - Use is primarily communication, presentations, etc.
  - The flash used consists of MLC parts with 2.5K cycle rating

A Use Case

- Average daily writes by the OS… about 1.8GB… Surprised?
- User application writes are dwarfed by comparison
- A single cycle of the entire media requires nearly a month (22 days)

Use Case Static Wear Leveling

- And assuming reasonably effective static wear leveling (60%), entire media is available for wear leveling
- 2,500 cycles x 22 days = 1,855 months x .60 = 211 years

Use Case Dynamic Wear Leveling

- Assuming we have a lackluster dynamic algorithm (90%) reduce available size by 30GB (application and OS)
- 2,500 cycles x 6 days = 750 months x 0.6 = 37 years

What Wear Leveling Does for Reads

- Dynamic wear leveling will not move areas that are only read
- High differences in erase counts result in higher BER
- Uncorrectable error rates are increased by 2-3 orders of magnitude

Wear Leveling for Reads

- Reads ~ Writes, but startup and hibernate cost additional 7GB of reads
- NAND manufacturers recommend cycling after 100,000 reads
- In use case, upwards of 2000 reads from the same areas daily
- With no caching and no wear leveling, cycle limits reached in 2.5 months

Balancing Wear Leveling with Performance

- Wear level operations should be bounded
  - Impacts to performance must be low
  - Take advantage of idle time if possible
- Features such as trim, pre-erase, discards will mitigate negative performance impacts of wear leveling

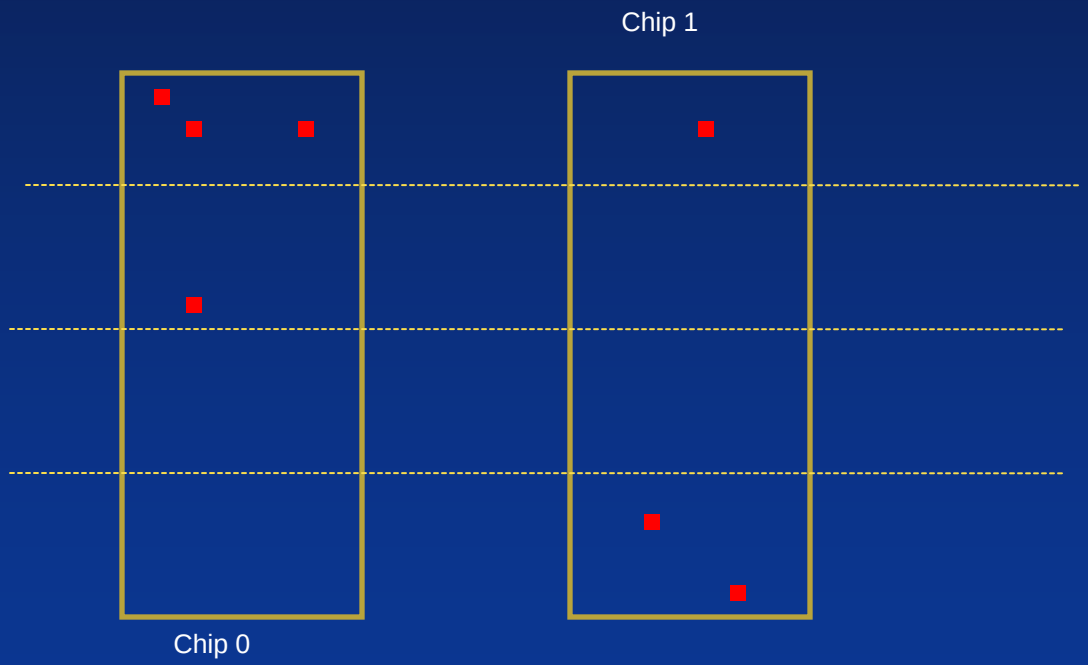
Balancing Wear Leveling with Performance

- Interleaving = speed
  - We interleave multiple plans on single devices for concurrency
  - Then two or more devices for a wider data path (more concurrency)
  - And then we do it again with multiple NAND channels or banks
  - And then once again in the field (RAID)

Definition of Interleaving

- To arrange data in a non-contiguous way to improve performance – Webopedia.com

Interleaving



Interleaving can cause unacceptable error counts

Interleaving = More Errors

- Interleaving two NAND devices can double the number of bad blocks
- Requiring more complex systems, more overhead, and/or stronger EDC

Conclusions

- Wear leveling requirements are *substantially* dependent upon use case
- Impacts to performance will increase with more static data in the disk
- Embedded systems will have more stringent requirements
- Understand your target customers' use case

References

- http://ieeexplore.ieee.org/Xplore/login.jsp?url=http://ieeexplore.ieee.org/iel5/4550747/4558854/04558857.pdf&authDecision=-203
- http://www.stanford.edu/class/ee380/Abstracts/081112-Fazio-slides.pdf
- http://download.micron.com/pdf/presentations/events/WinHEC_Cooke.pdf