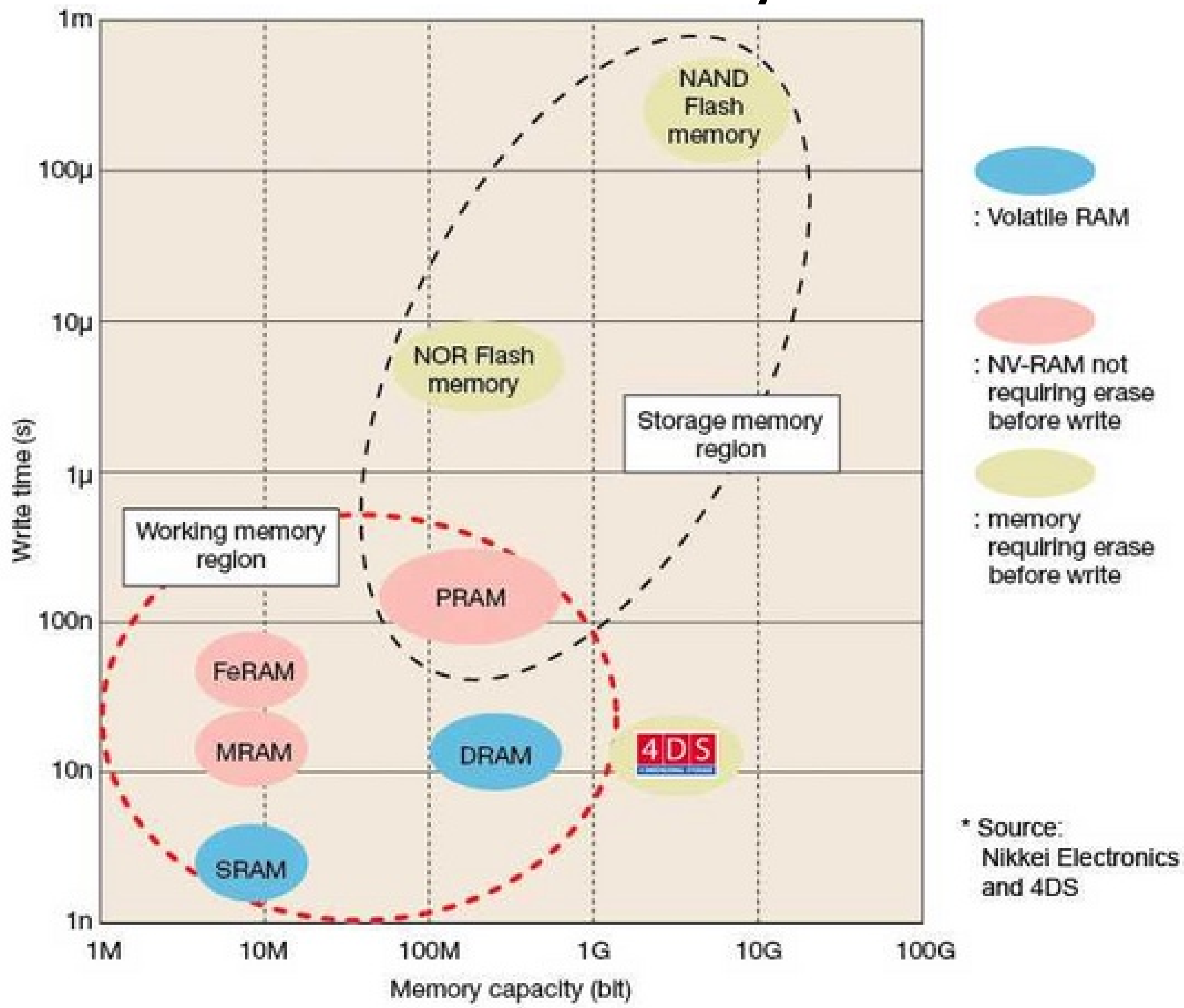
Not Your Father's Data Storage: Hardware and software aspects in mobile storage

Prof. Li-Pin Chang

National Chiao Tung University

2018 Dec

Non-Volatile Memory



Faster write

Smaller capacity

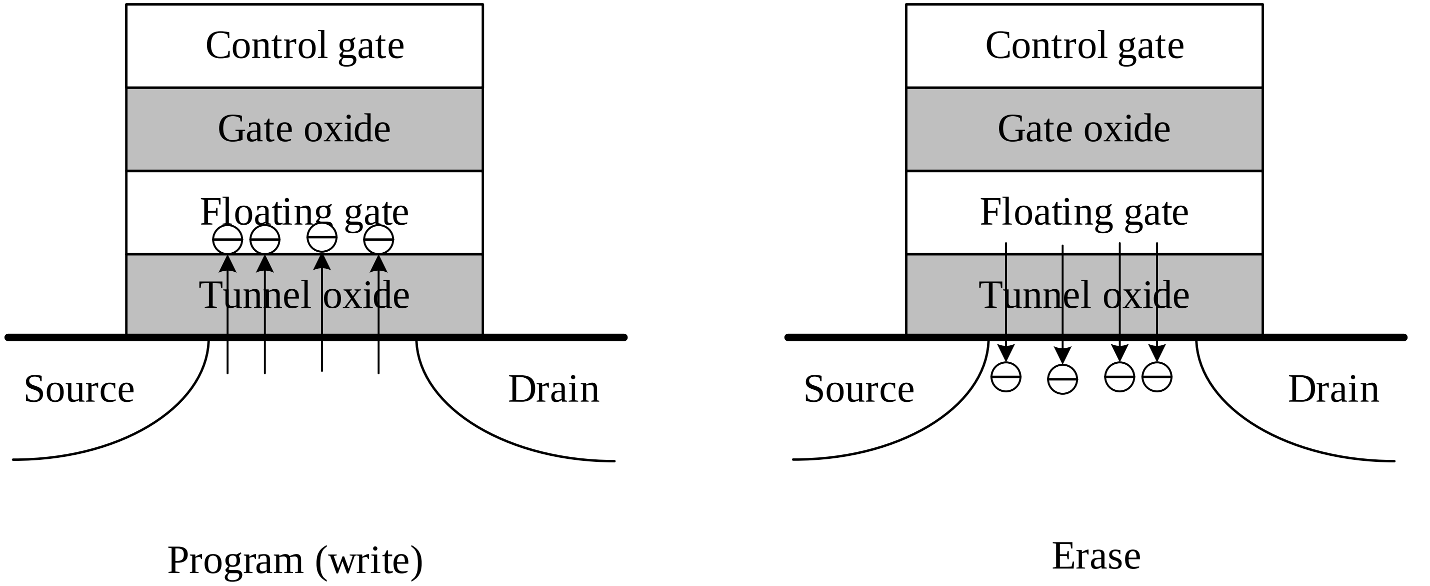
http://www.storagenewsletter.com/news/flash/4ds-start-up-profile?snc=7835

Common Issues of NVRAM

- Writes are slow and consume more energy
  - Inherently need more time and energy to change the physical characteristics of the medium
  - Charging (flash memory); heating (PRAM)
- Write endurance
  - Every write consumes a bit of flash lifetime
  - Just like how batteries wear out
- Asymmetric read and write latencies
  - Reads are much faster than writes

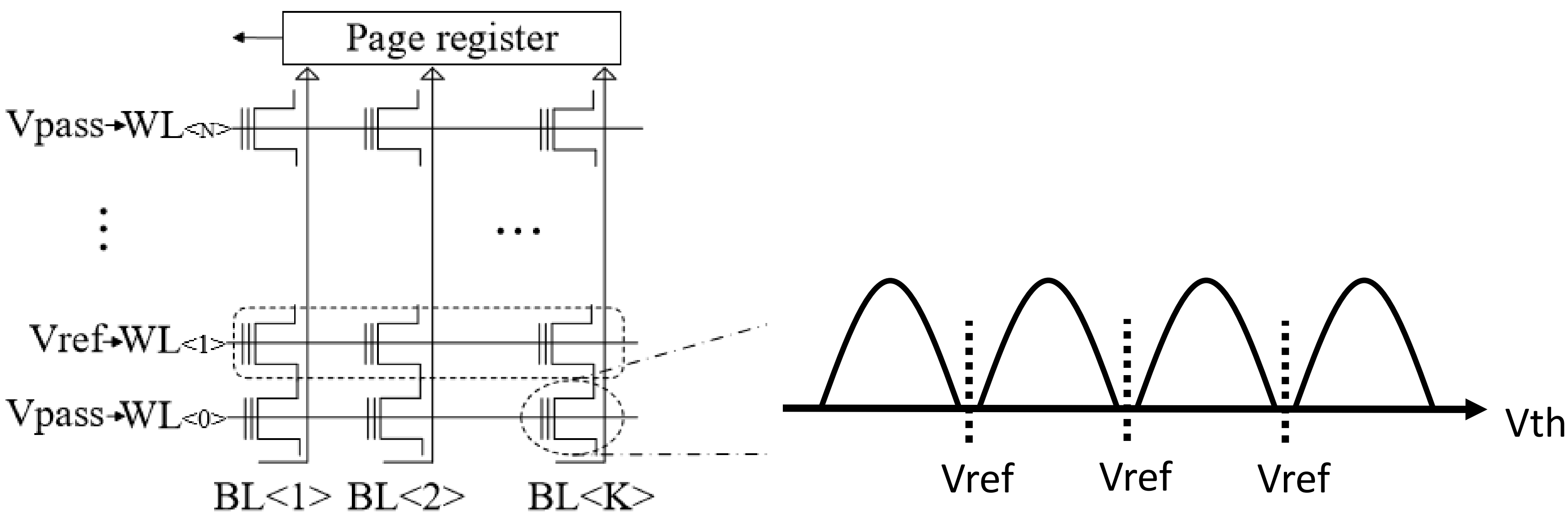
NAND Flash Memory

- Cell structure, flash program and erase



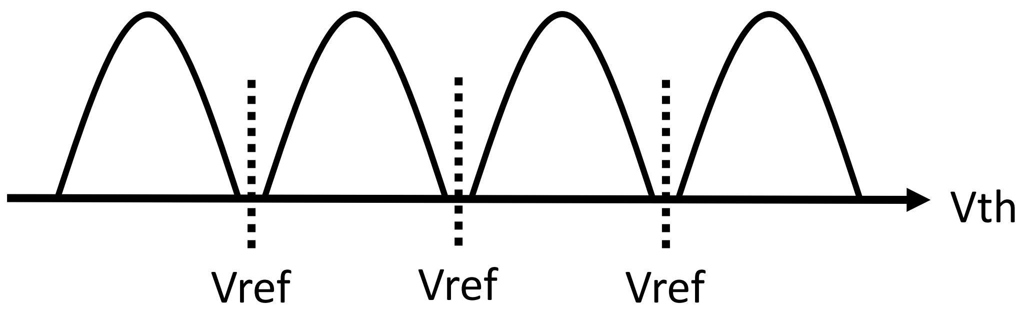
NAND Flash Memory

- A piece of NAND flash is formed by an array of cells
  - Horizontal: word lines
  - Vertical: bit lines
- The logical value of a memory cell is determined by the threshold voltage of the cell

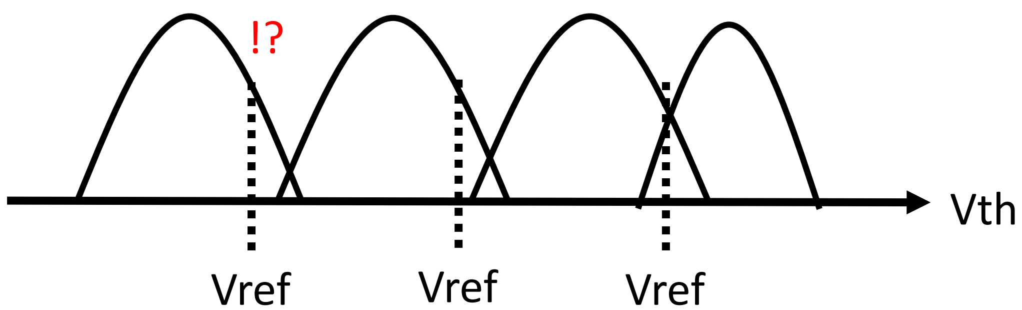


Reliability

- Threshold voltage distribution right after programming

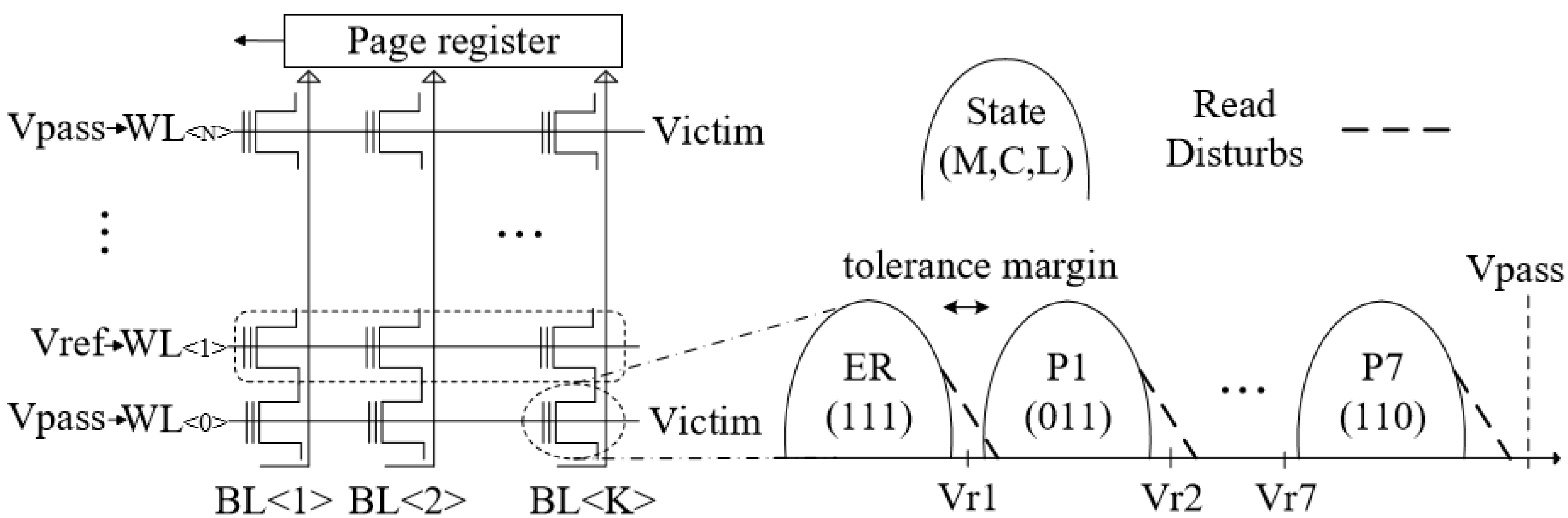


- Vth distribution disturbed by various types of noises



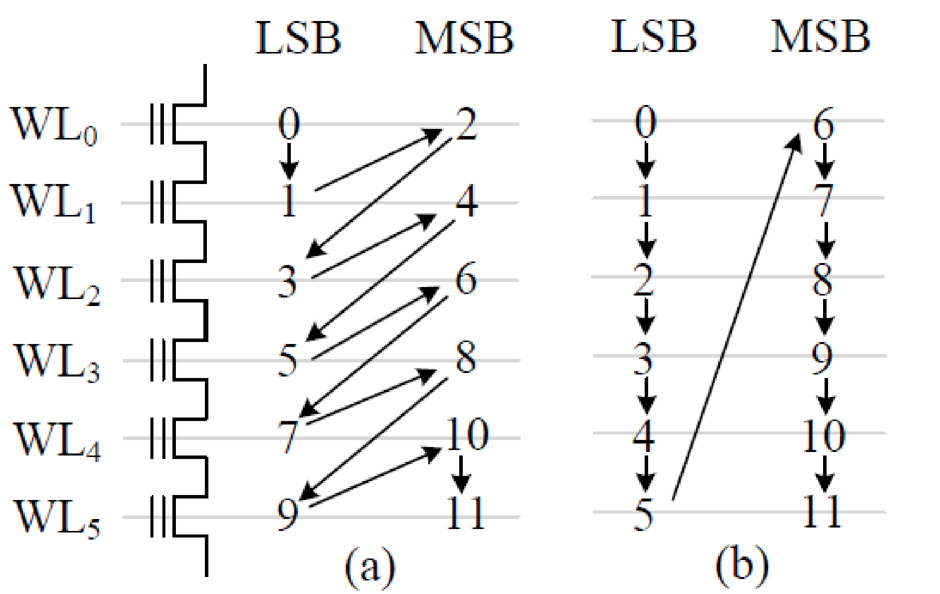
Read Disturbance

- Reading data from a requires to "isolate" neighbor word lines by applying a high Vpass
- Unexpected electron charge injection -> read error



Write Disturbance

- Writing a word line may unexpected inject electron charge to neighbor word lines
- Program (write) word lines with a fixed order to minimize such interference

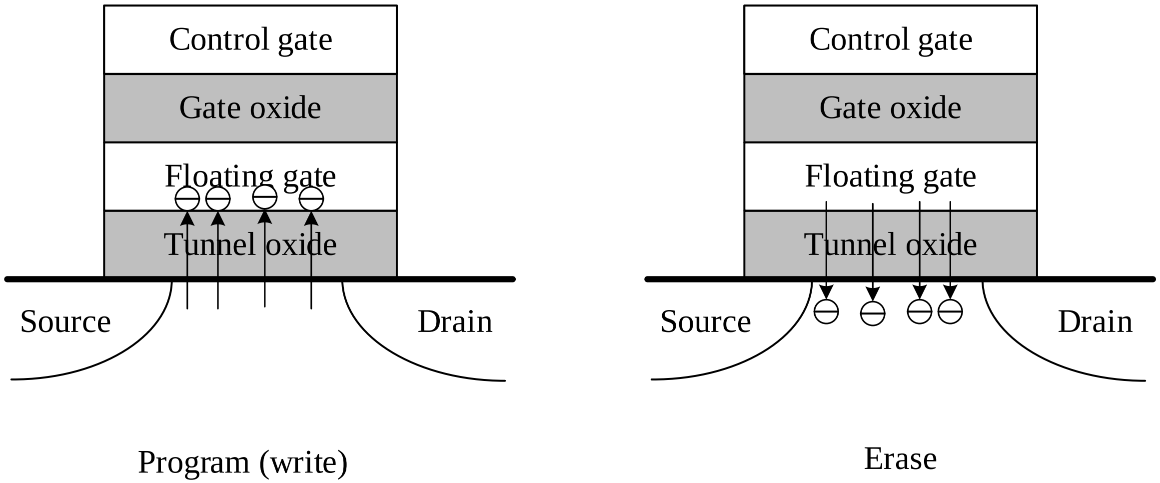


Retention Error

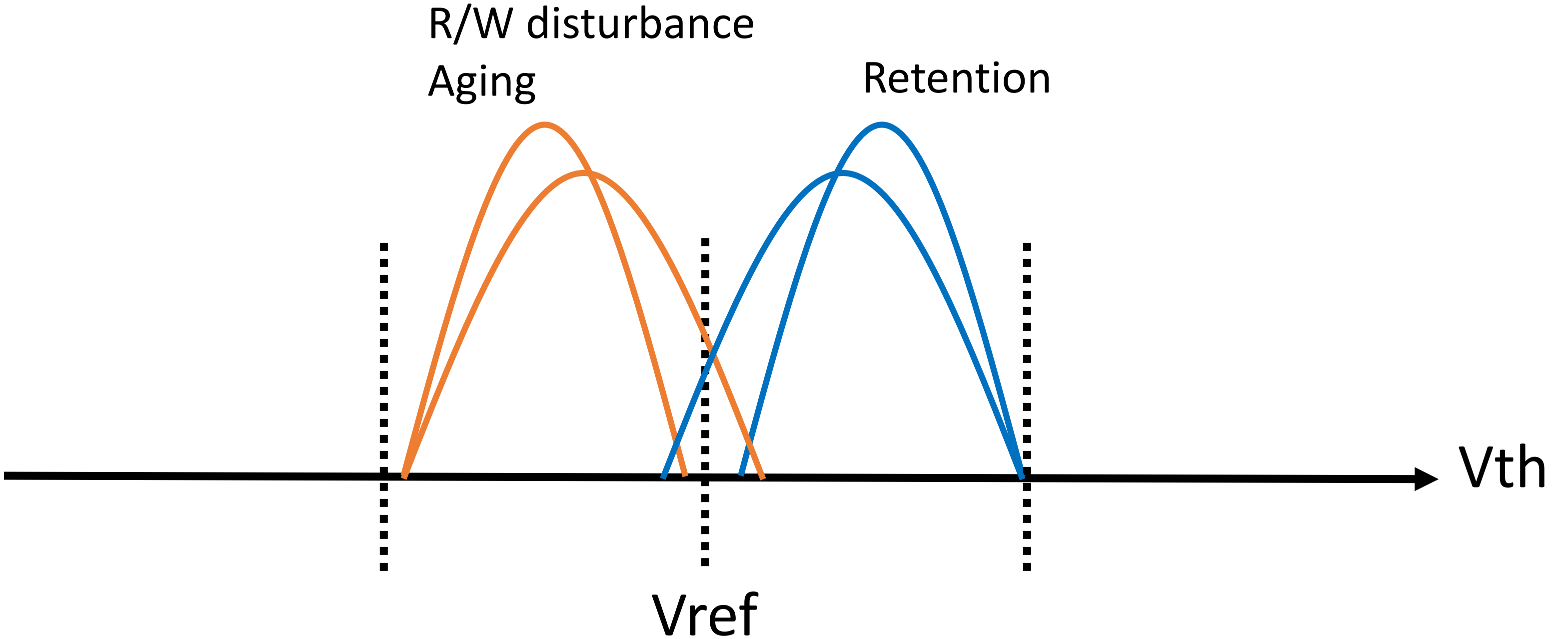
- Electronic charge escape from the floating gate over time; eventually Vth levels become undecidable
- Just like batter level decreases over time
- If you do not re-write data in 3 years (typical retention requirement) then you lose your data

P/E Cycling

- Electronic charge injection and discharge accumulate damage to the tunnel oxide
  - Charge becomes easier to escape from the floating gate (retention)
  - Charge trapped in tunnel oxide due to structural damage



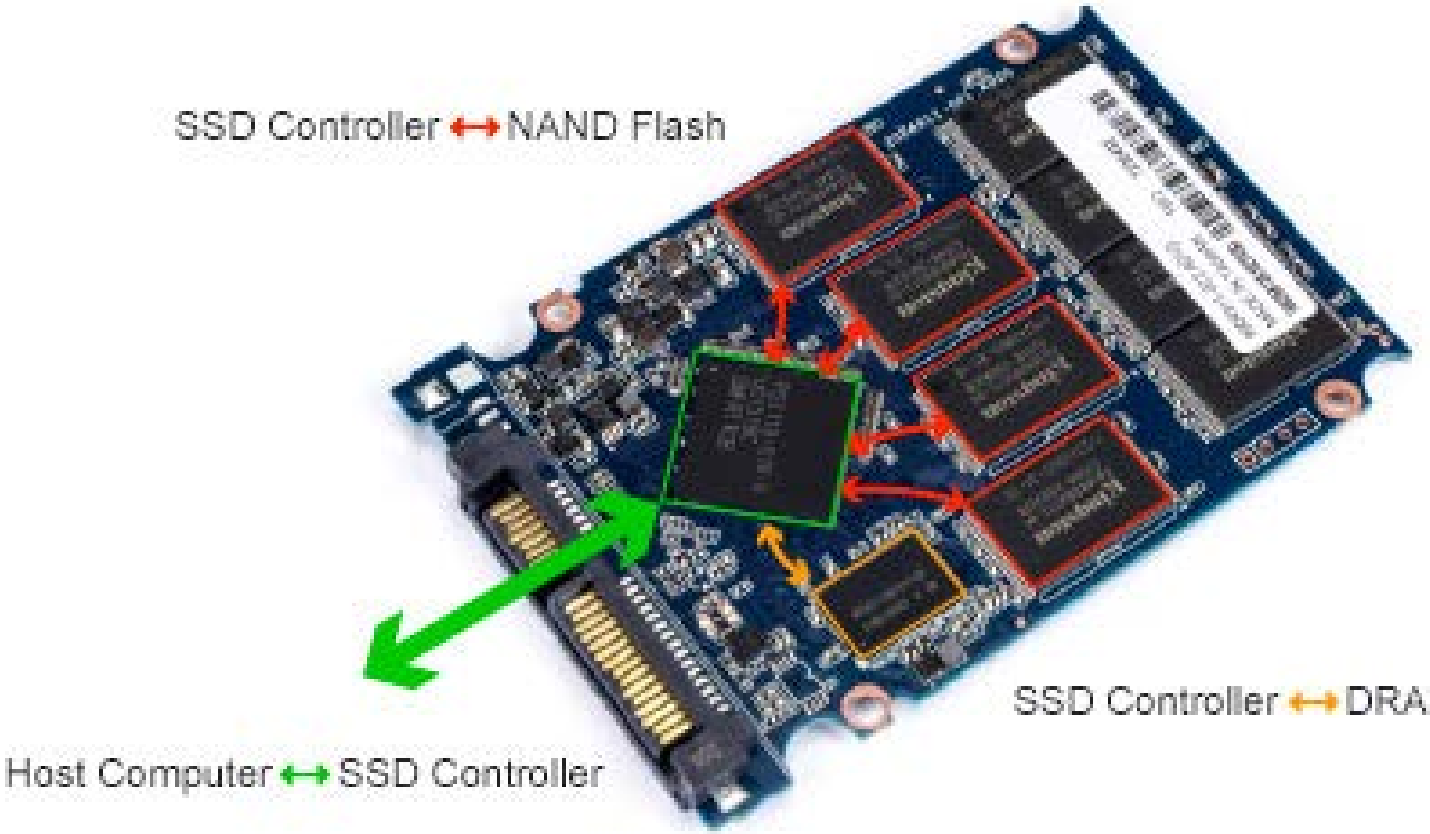
Reliability



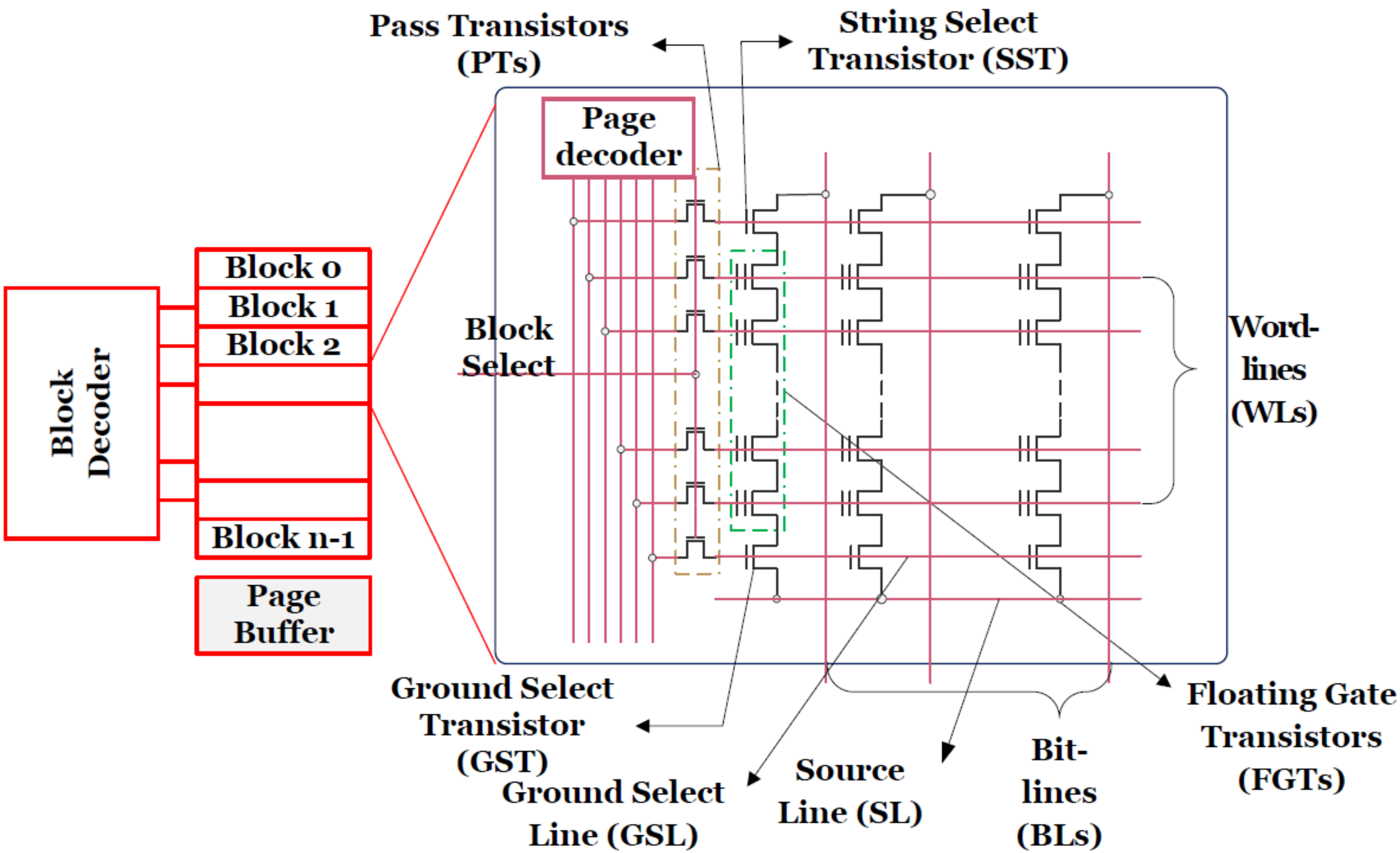
Solid-State Disks (SSDs)

- Storage devices that emulate standard block devices using non-volatile memory
  - Flash memory or battery-backed RAM
  - The OS use the legacy I/O stack on top of SSDs
- Products
  - Embedded flash cards, SD cards, USB thumb drives, SSDs, PCI-e flash cards
- Performance
  - RAM disk > SSD >> HDD
- Applications
  - Cloud storage: tier storage, cache SSDs
  - Personal computer: HDD replacement, system drive
  - Embedded storage: Smartphones, tablets, laptops, wearables

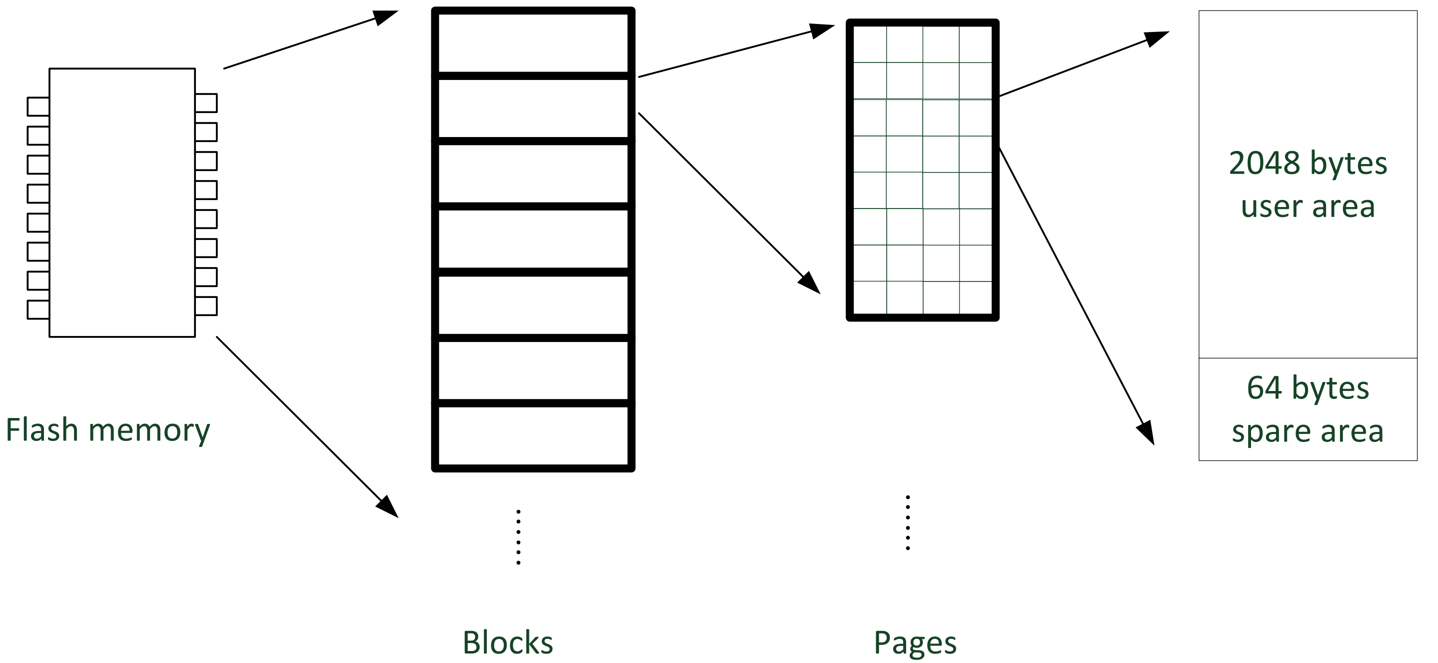
SSD Internal Organization



https://www.kingston.com/tw/ssd/data-protection


Pass Transistors
(PTs)
String Select
Transistor (SST)
Page
decoder
Block 0
Block 1
Block 2
Block n-1
Block
Decoder
Page
Buffer
Block
Select
Word-
lines
(WLs)
Ground Select
Transistor
(GST)
Ground Select
Line (GSL)
Source
Line (SL)
Bit-
lines
(BLs)
Floating Gate
Transistors
(FGTs)

NAND Flash Geometry



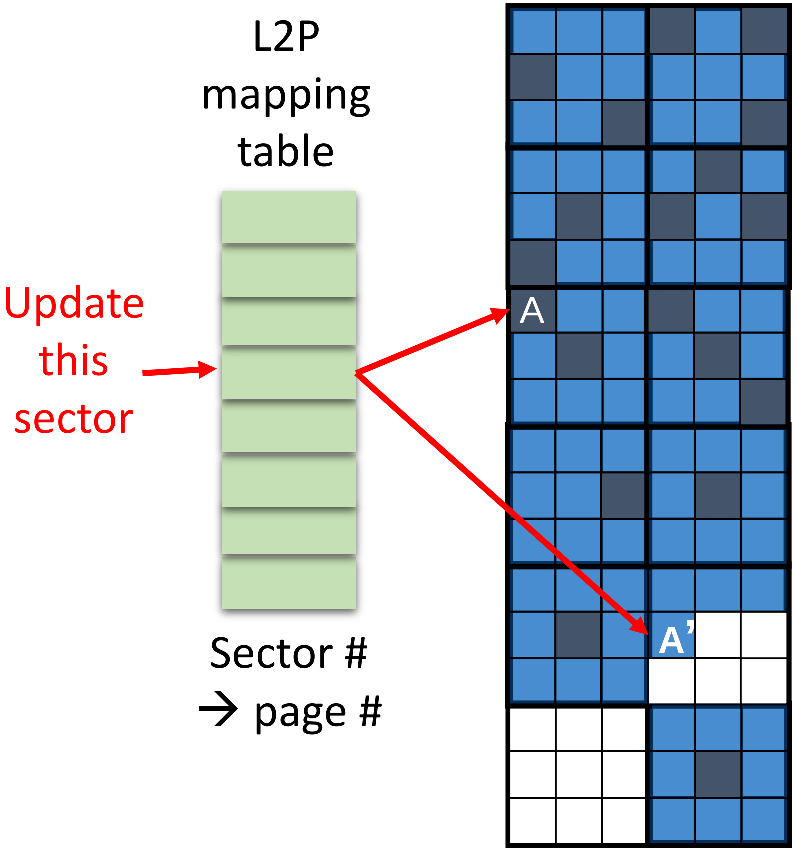
- Unit size
  - Read/write: page
  - Erase: block

Flash Translation Layer (FTL)

- A firmware layer inside of SSDs
  - Hiding flash memory physics from the host
- Provide block device emulation to the host
- Manage flash memory inside of SSDs
  - Logical-to-physical address translation
  - Garbage collection
  - Wear leveling

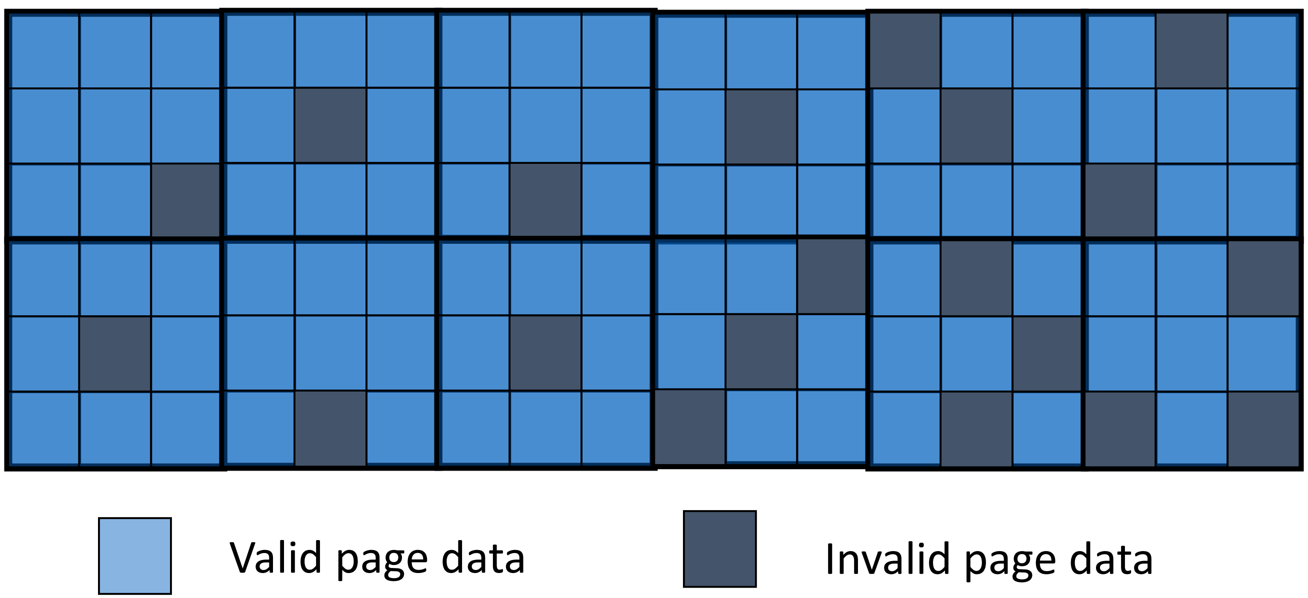
Logical-to-Physical Address Translation

- Pages cannot be overwritten unless being erased
- Erase a block every time a page is overwritten
  - Too inefficient
- Out-of-place update; mark old data invalid
- Need logical-to-physical address translation
  - From logical sector # to physical page #



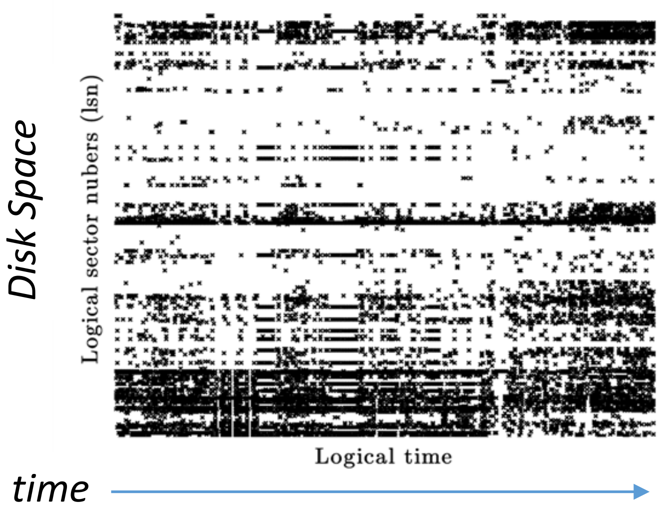
Garbage Collection

- Recycle memory space occupied by invalid data through block erasure
- Victim selection
  - Minimize the page-copy overhead

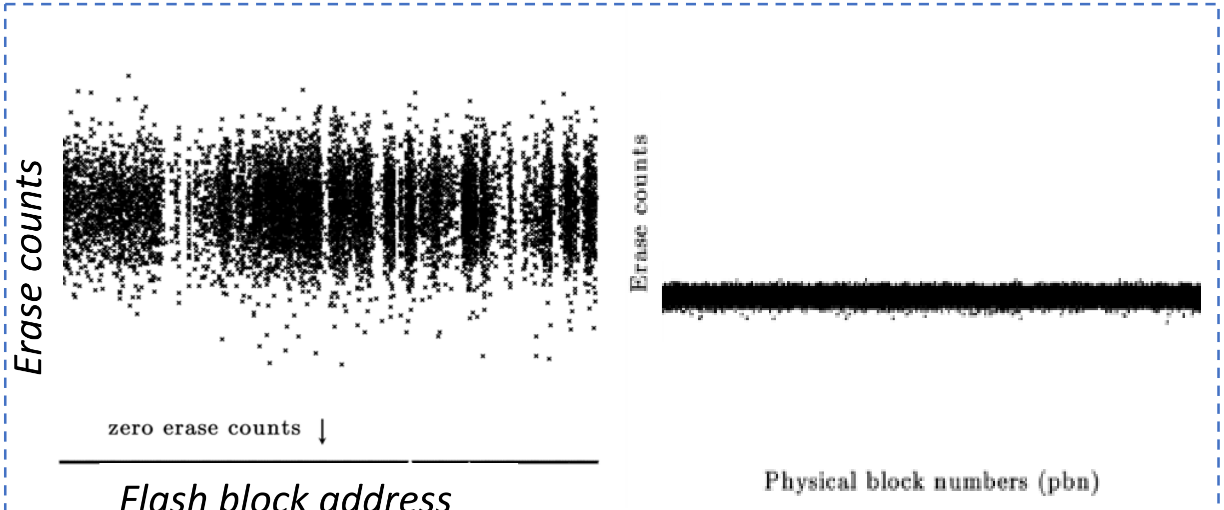


Wear Leveling

- Typically a (MLC) block endures 3000 cycles of program-erase operations (P/E cycles)
- Locality of write creates frequently written blocks
- Delay the first block retirement by migrating cold data



Disk write pattern

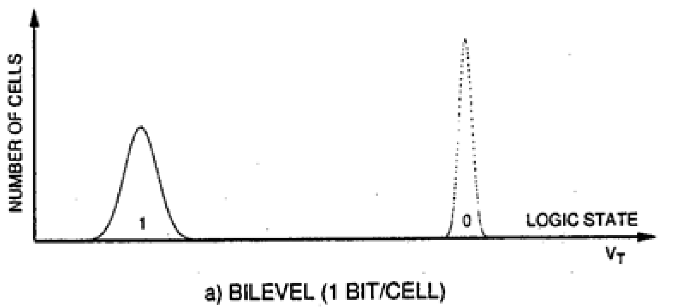


Block ECs without WL

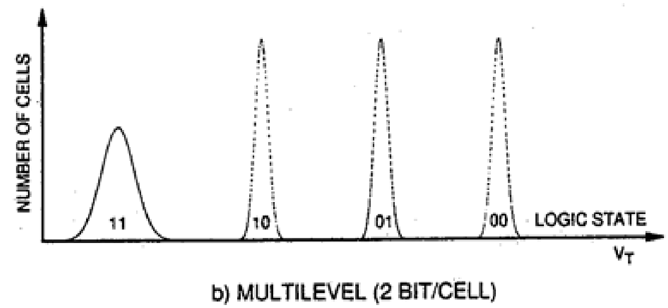
Block ECs with WL

Li-Pin Chang, Tung-Yang Chou, and Li-Chun Huang, "An Adaptive, Low-Cost Wear-Leveling Algorithm for Multichannel Solid-State Disks," ACM Transactions on Embedded Computing Systems, Volume 13, Issue 3, 2013.

Multilevel Cells



Single-level cell



Multi-level cell

- SLC vs. MLC flash
  - Comparable read speed
  - SLC writes about 2x or 3x faster than MLC
  - P/E endurance: 5K cycles (MLC), 100K (SLC)
  - SLC is 2x or 3x more expensive than MLC (and increasing)
  - Hybrid SSDs, dynamic density SSDs
- Now TLC, QLC are in mass production

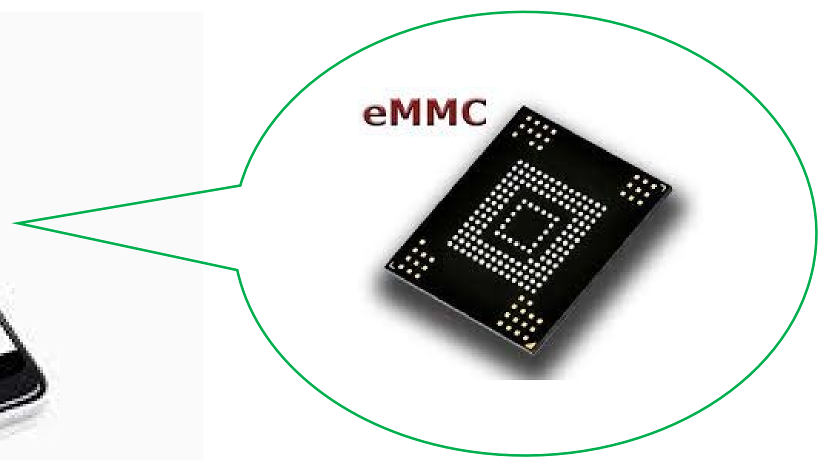
Recap

- Noises
  - Read disturbance
  - Write disturbance
  - Retention error
- Flash management
  - Address mapping
  - Garbage collection
  - Wear leveling

Now...

What's the Problem with Mobile Storage?

- Our goal: to reliably operate NAND-flash-based mobile storage throughout the entire smartphone replacement cycle
- Our approach: write stress reduction using data compression and data deduplication

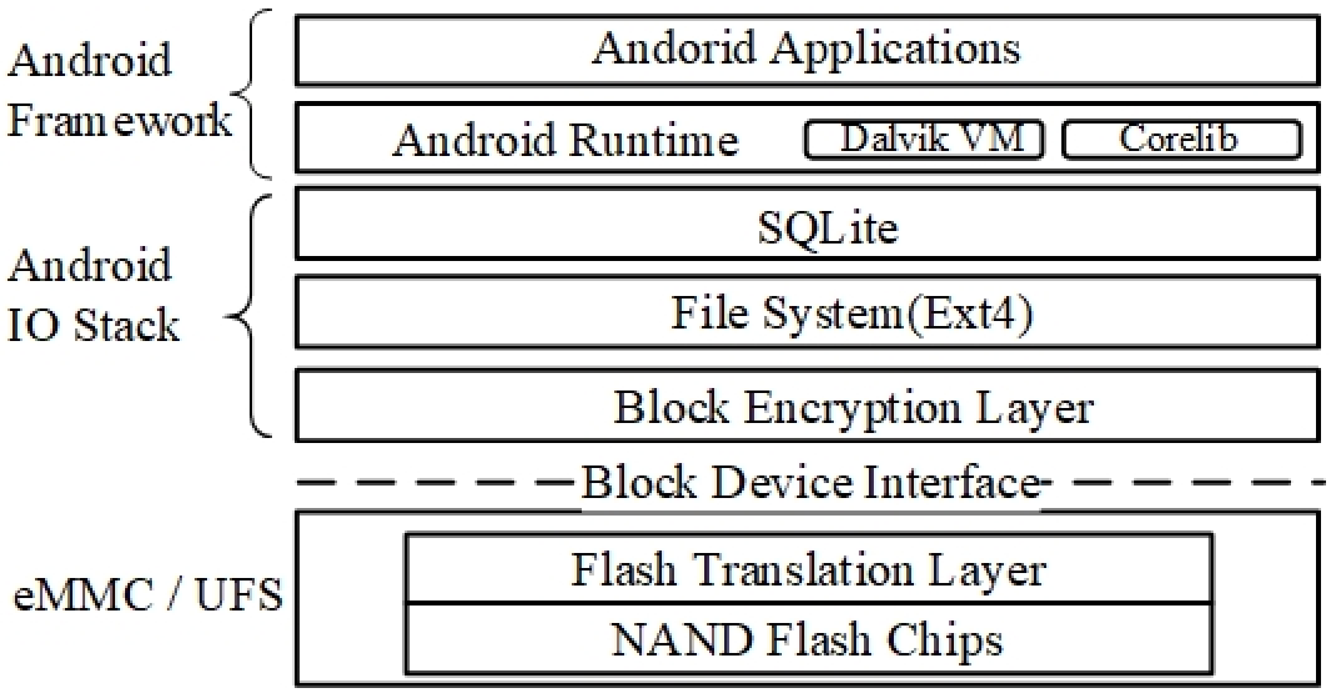


Mobile Storage Lifespan

- Is mobile storage lifespan a real problem?
  - End users are becoming reluctant to replace their smartphones (replacement cycle -> 3 years)
  - High-level-cell flash memory offers improved storage density at a cost of reduced endurance
  - I/O patterns of mobile storage is write intensive, nearly 90% of I/Os are write [FAST'12][EMSOFT'12]
- Consumers: How safe are my data?
- Vendors: How much the cost cut for flash memory is possible?

Android I/O Stack

- It operates multiple journaling layers
- It eagerly flushes dirty data to mobile storage



Write Stress Reduction

- Possible approaches for write stress reduction
  - Data compression
  - Data deduplication
- But first we need to investigate data compressibility and duplication in mobile storage
  - *A journey to debunk common beliefs begins…*

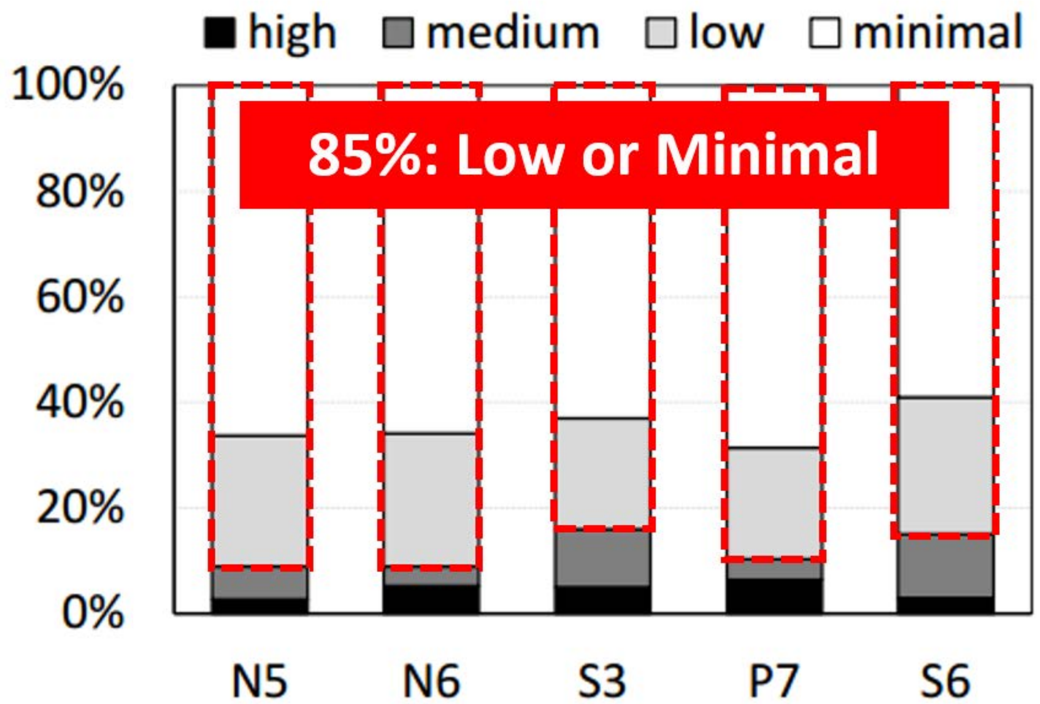
Data Compressibility Study

- Disk volume snapshots
  - Investigate compressibility on a set of aged smartphones

	Samsung Galaxy S3	Google Nexus 5	Google Nexus 6	Huawei Ascend P7	Samsung Galaxy S6
Usage Age	24 months	12 months	6 months	12 months	16 months
Release Date	2012	2013	2014	2014	2015
Linux Ver.	3.0.8	3.4.0	3.10.40	3.0.8	3.10.61
Android Ver.	4.3	5.0.1	5.1.1	4.4.2	5.0.2
Storage	eMMC	eMMC	eMMC	eMMC	UFS
Partition Size	11.6 GB	26.8 GB	26 GB	11.8 GB	25.6 GB
Utilization	63%	93%	57%	51%	92%

A summary of Android smartphones in empirical study

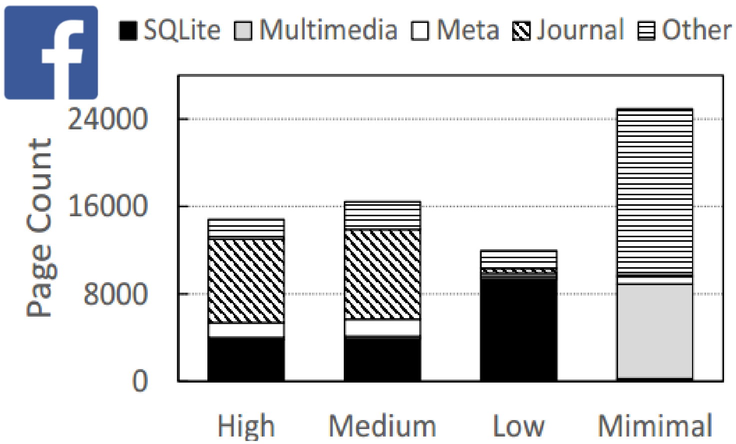
Data Compressibility in Smartphones

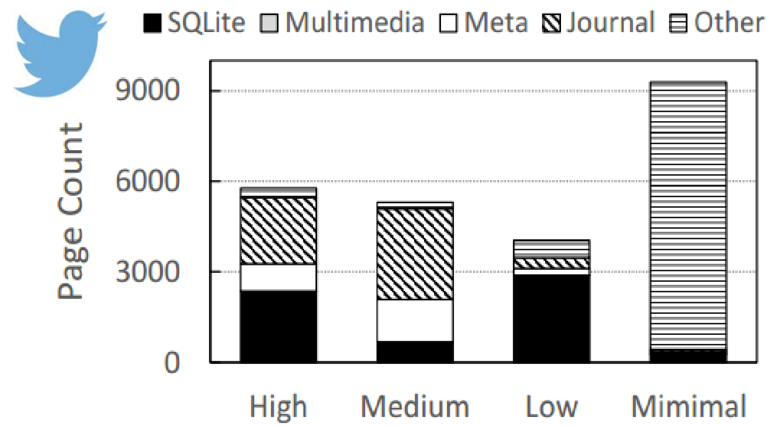


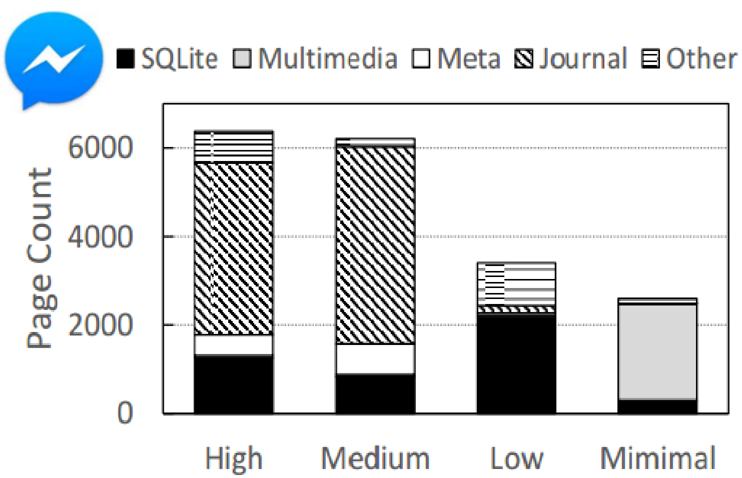
Compressibility of a file block is classified to 4 levels:

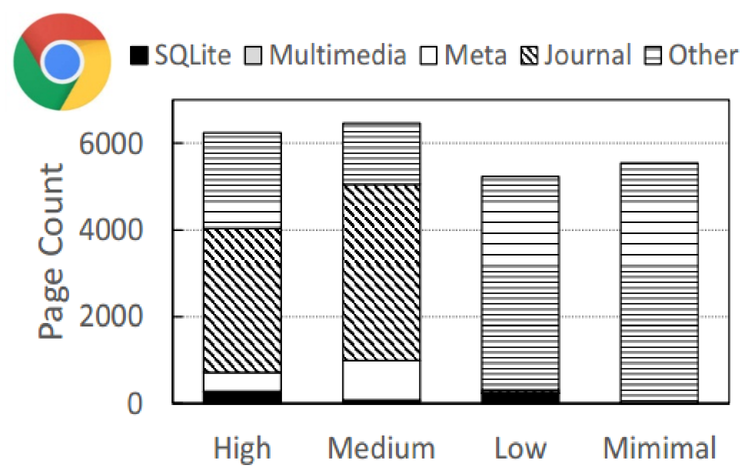
***High* (Cr<0.25), *Medium* (0.25≤Cr<0.65),**
***Low* (0.65≤Cr<0.95), *Minimal* (0.95≤Cr ≤1)**

- The compressibility of write traffic on mobile devices

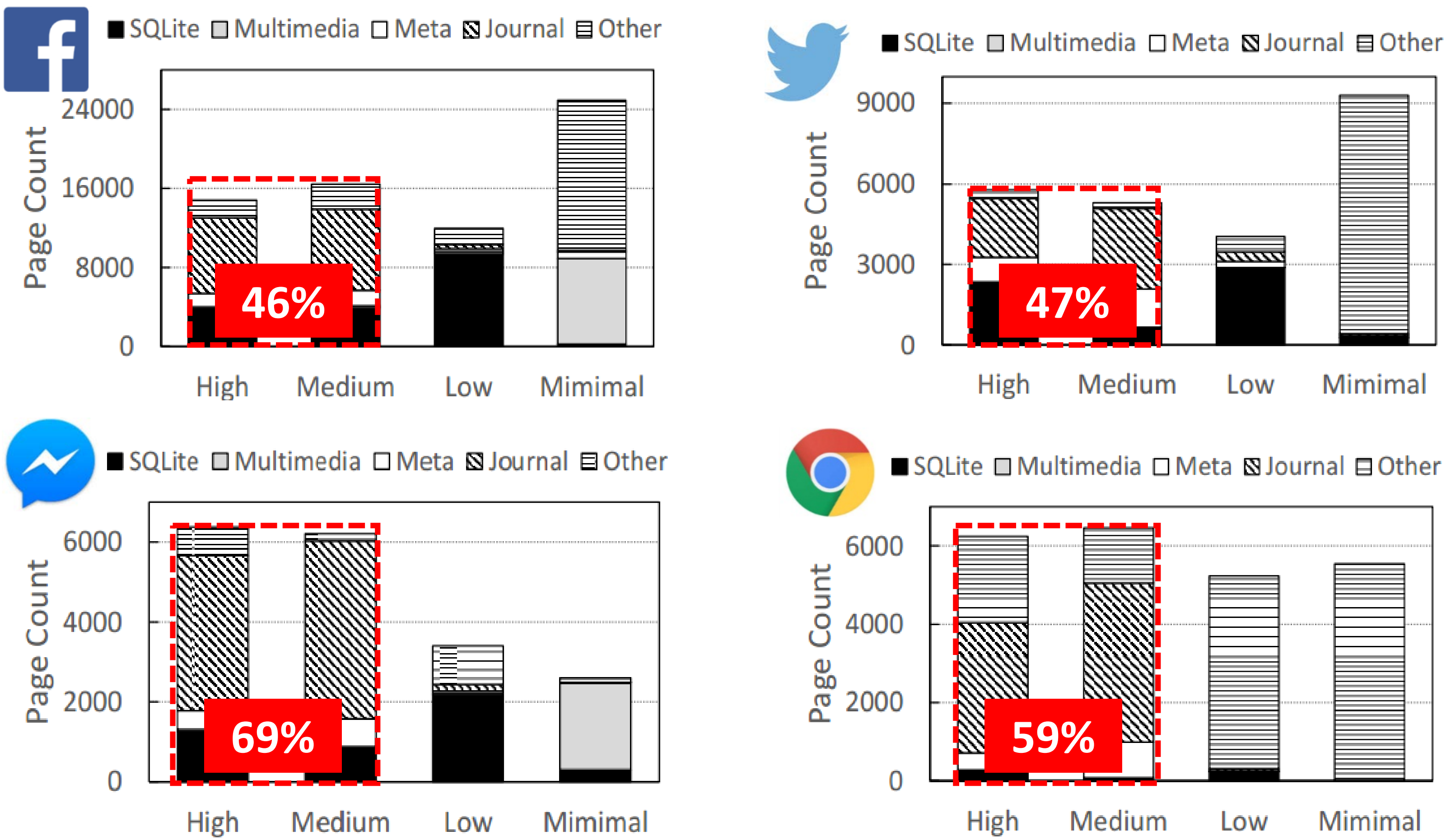








- The compressibility of write traffic on mobile devices



- Application write traffic is highly compressible！
  - Related to the mobile application behaviors and the file system design

Frequent SQLite operations and Ext4 journaling

Q: Are files compressible on mobile devices?

Yes, highly promising!!!

- Compression timing overhead
  - Compression time is proportional to CPU frequency [Wu, EuroSys’12]
- Controller running frequency is slow
  - Compression timing overhead is noticeable
  - Unconditional compression is not efficient!!

However, controllers of typical eMMCs are not faster than 200MHz because of

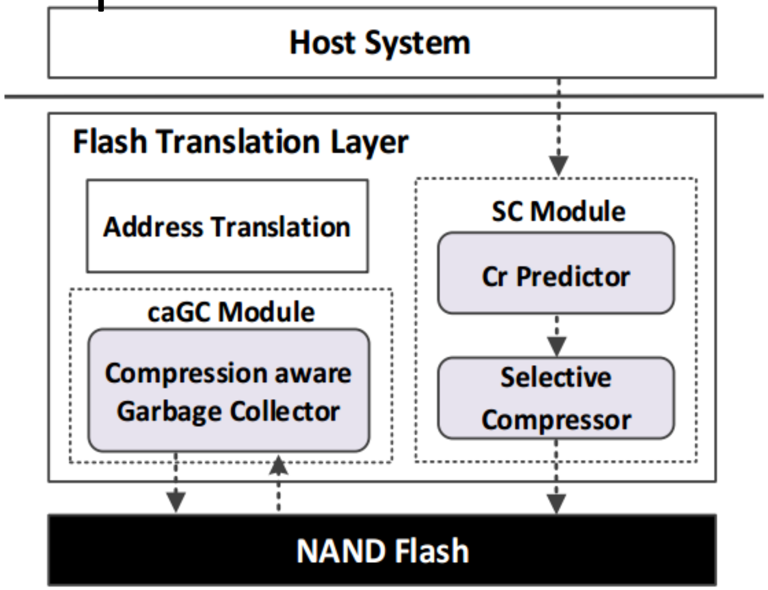
1. Cost control;
2. Power consumption issue.

Slow controller → **High compression overhead**

Wu G, et al. Delta-FTL: improving SSD lifetime via exploiting content locality[C]//Proc. of EuroSys’12.

Transparent Data Compression

- A firmware-level approach to transparent data compression
- Main issue
  - Slow processor
  - Limited energy
- Solution
  - LZO compression
  - Fast compressibility prediction

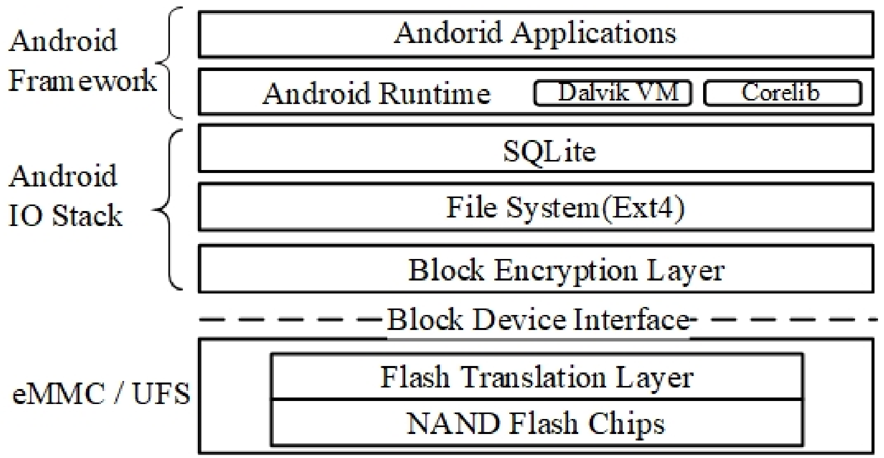


Architecture of lightweight data compression

- Result
  - Reducing the total # of erases by about 50%!

But Wait…

- Compression
  - LZO-based block compression [EMSOFT'17][TCE'11]
  - Ineffective under encryption!
- Deduplication
  - Duplicate data remapping [MSST'17][FAST'11]
  - Duplicate data are still duplicate after encryption

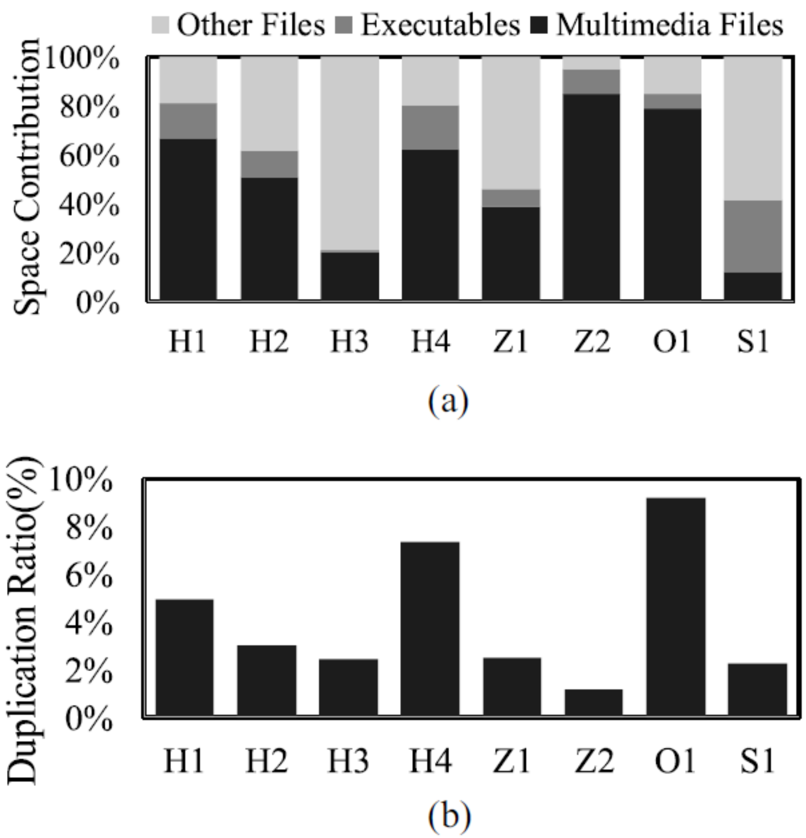


- Compression
  - XXXXYYYYYZZZZZZZXXXXXXX → 00011001
  - How about ZYXXYZYYZXZYXZYXZYXZYY??

- Deduplication
  - Write ZYXXYZYYZXZYXZYXZYXZYY and ZYXXYZYYZXZYXZYXZYXZYY
  - The 2nd write is not necessary

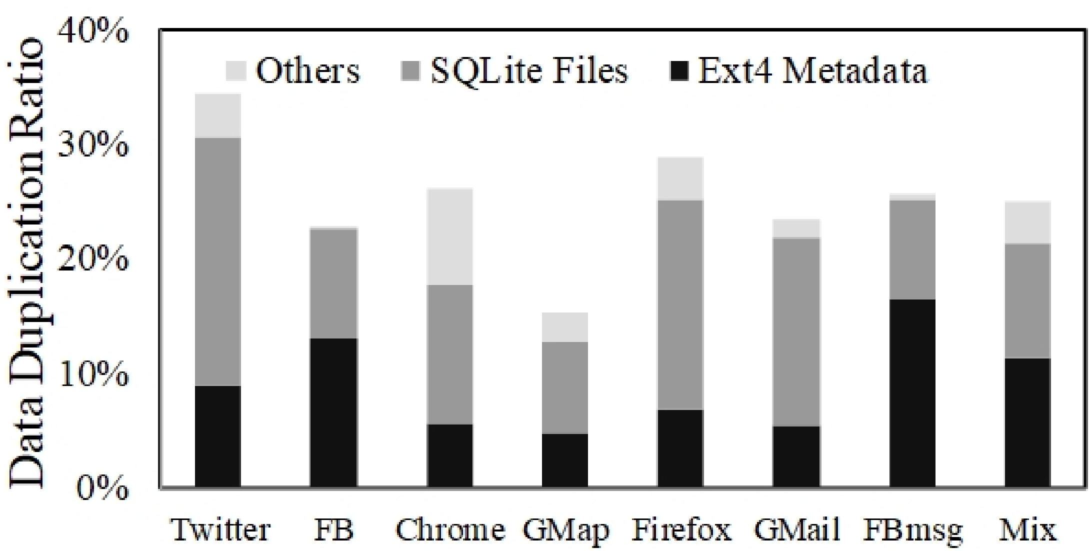
Snapshot Analysis

- Common belief: there are few duplicate data in smartphones

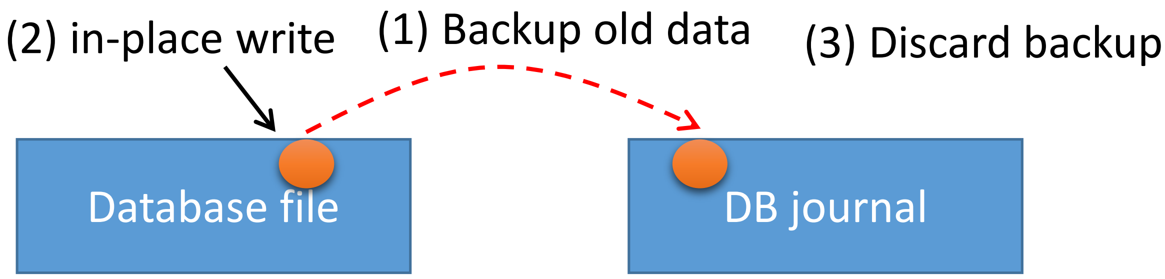


Write Traffic Analysis

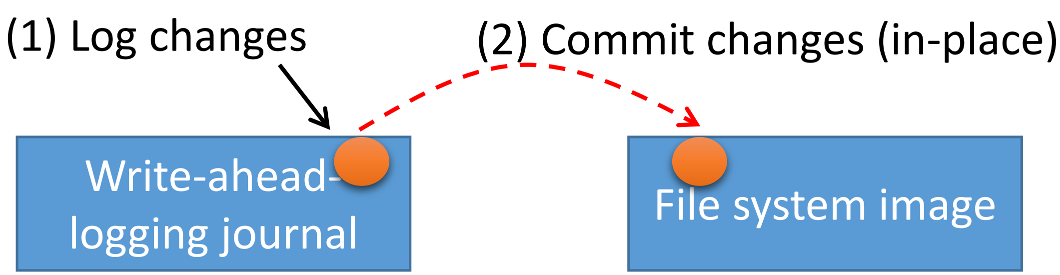
- On average, write traffic bound for mobile storage carries ~=30% duplicate data!
- Contributed by SQLite files and Ext4 journal
  - But why?



Copy, Copy, and Copy…



SQLite roll-back journaling

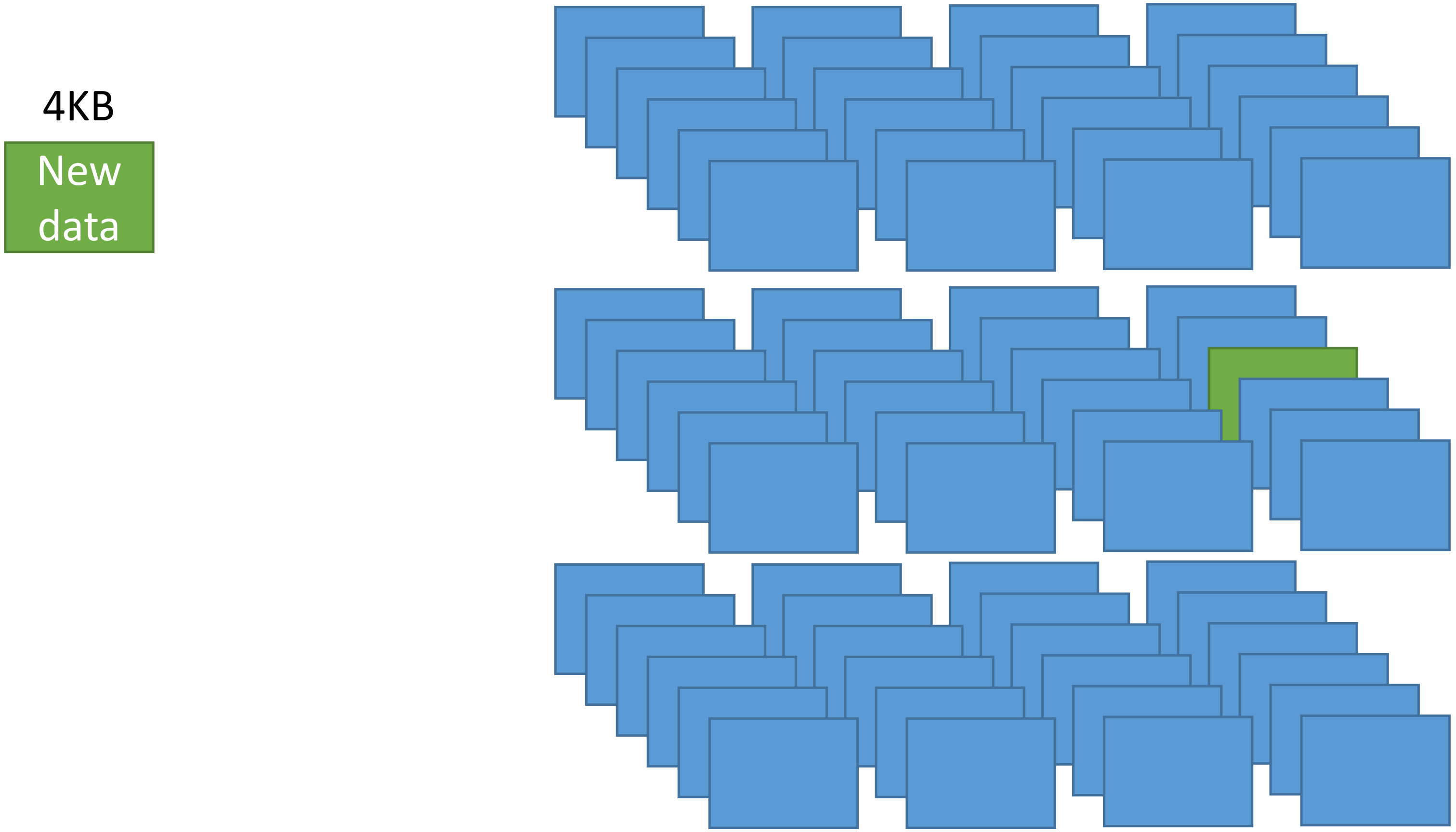


Ext4 ordered-mode journaling

A lot of copy-induced data duplication in write traffic

Basics of Data Deduplication

- How to identify whether the incoming data is a duplicate of existing data? By data comparison?

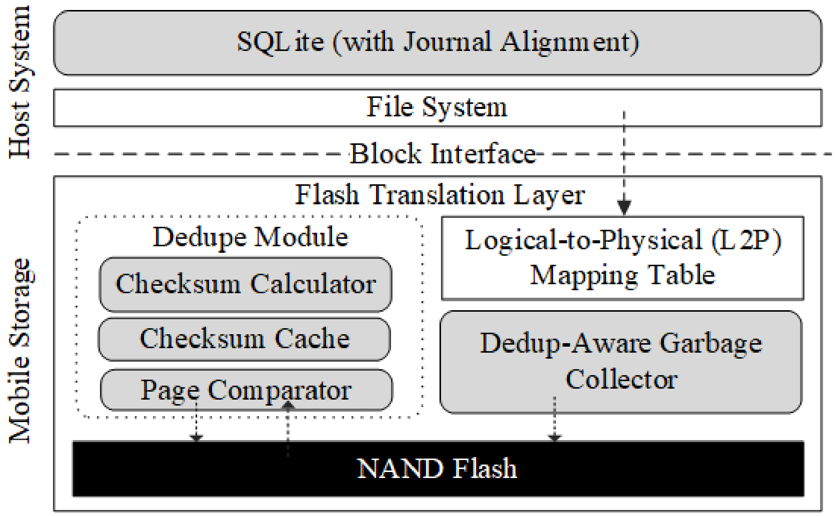


Firmware-Based Deduplication

- Efficient deduplication is based on collision-resistent hash functions, such as SHA-1 [FAST'02]
  - Hash collision rate < hardware malfunction rate
- Existing approaches
  - Take 2.36 ms to compute an SHA-1 value for a 4KB page on a 200 MHz embedded processor (too slow!)
  - Store hash values of pages in embedded RAM (too large!)
- Ideas
  - Use fast (but weaker) checksum algorithms
  - Cache only checksums of recently written data

Transparent Data Deduplication

- A firmware-level approach to transparent data deduplication
- Main issue
  - Slow processor
  - Limited energy
- Solution
  - Fast checksum algorithm
  - SQLite alignment
  - Improved garbage collection
- Result
  - Reducing the total # of erases by about 47.8%!



Recap

- While mobile storage snapshots are not compressible, mobile write traffic is
- While mobile storage snapshots contain little duplication, mobile write traffic contains a lot

- Compression and/or deduplication reduce write traffic volume → lengthen mobile storage lifespan
- Any proposed designs are subject to severe resource constraints (time and space)

Conclusion

- Flash memory performance and reliability will continue to degrade due to cost consideration
  - Poor storage performance will eventually create perceived latency
  - You may have to ditch your smartphone in 1 years because storage stops working
  - Need more research effort here!

- Even though most of flash management is carried out by system software, it involves deep understanding of physical (hardware) aspect