DBMS & Flash Storage

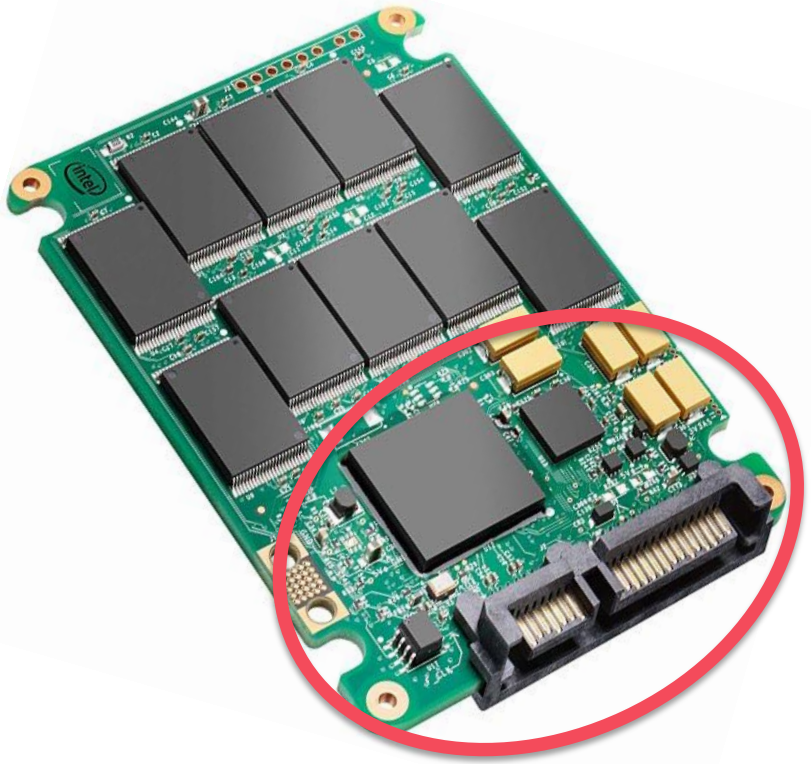
HDD != Flash

	HDD	Flash
Overwrite	in-place	after ERASE
Endurance	“infinite”	3-100K erases
Latency	$T_{read} = T_{write}$	$T_{read} << T_{write}$ (~10x)
Pattern	seq. != rand	seq. == rand
Parallelism	no	yes

HDD ~ SSD

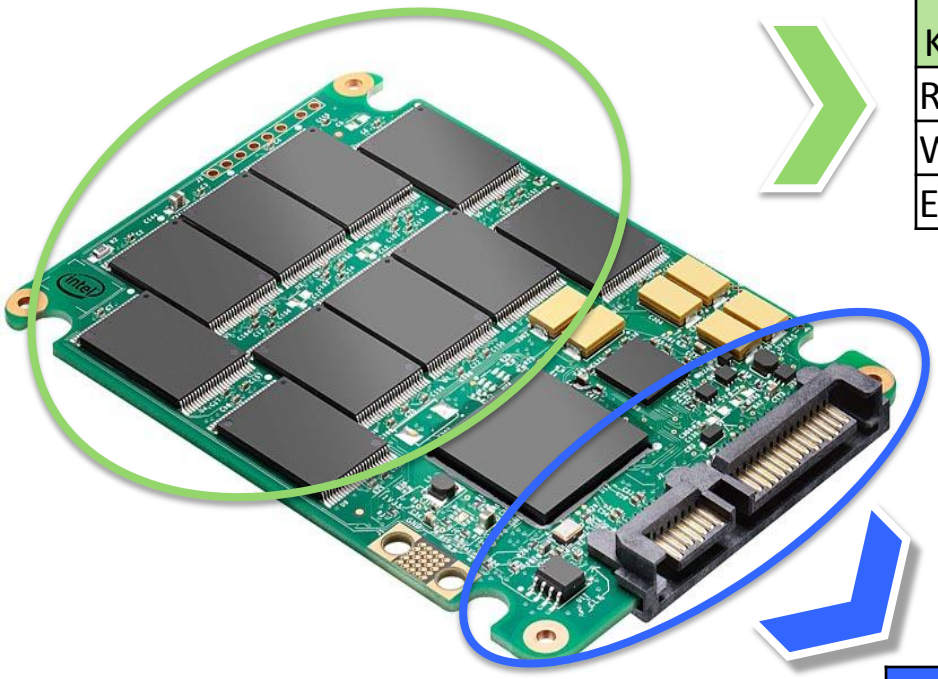
- SSD = FTL + Flash

- FTL = Flash Translation Layer
  - On-device layer that ensures low-level block interface compatibility
  - Erase-before-rewrite principle → out-of-place update → address translation (mapping)
  - Other FTL processes:
    - Garbage Collection
    - Wear-leveling
    - Bad Block Management
    - Error Correction



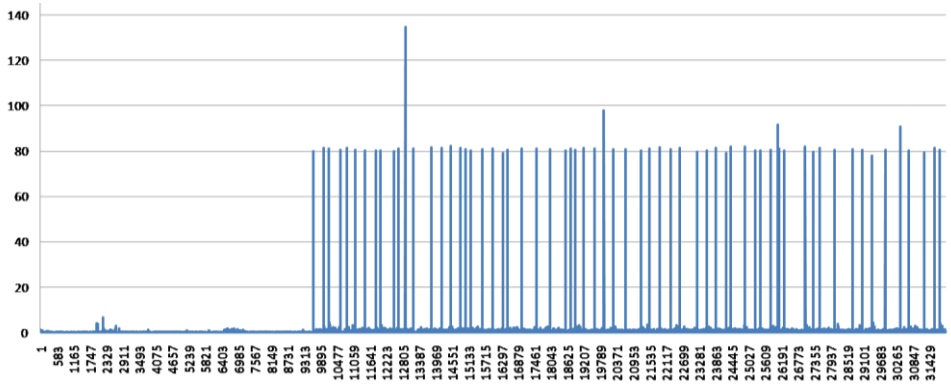
SSD != Flash

- Performance
- Predictability



Samsung K9K4G08U1M	
Read	25 us
Write	200 us
Erase	2 ms

Intel X25E SLC SSD	
Read	167 us
Write	455 us



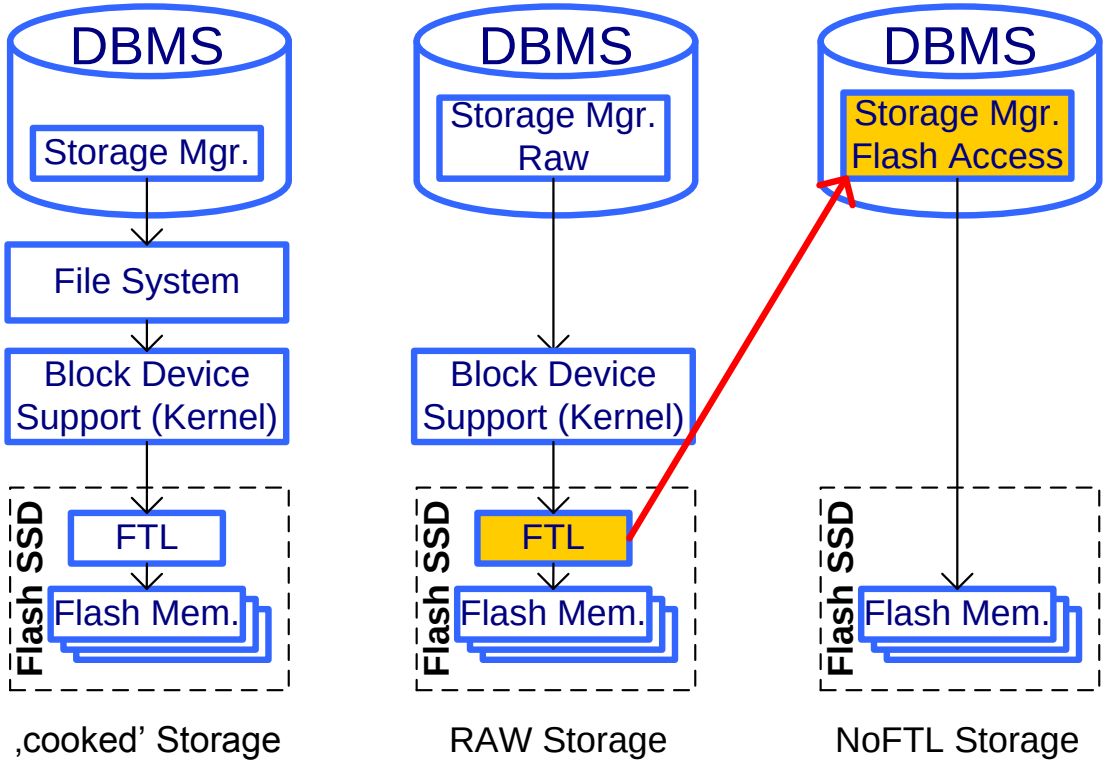
NoFTL: FTL-less Flash storage for DB

- Continuation of the long history of simplifying the I/O stack: DBMS on RAW storage

- DBMS operates directly on RAW NAND
- DBMS has full control over the NAND storage
- Integrate all NAND maintenance into DBMS

  (WL, GC, BBM, ECC, mapping)
- Remove intermediate layers

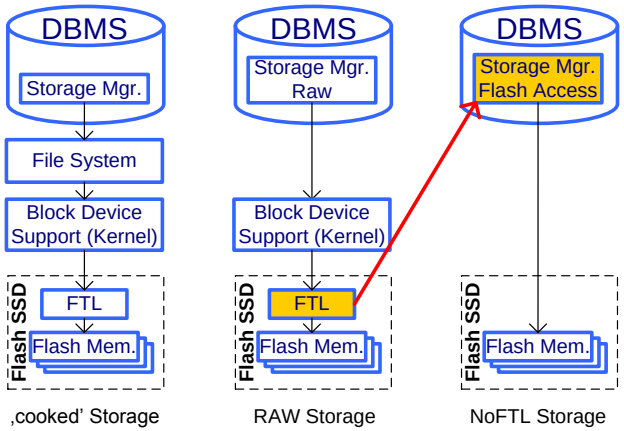
  FS, block device layer, FTL



S. Hardock, I. Petrov, R. Gottstein, A. Buchmann. *NoFTL: database systems on FTL-less flash storage. VLDB'13*
http://dblab.reutlingen-university.de/tl_files/downloads/NoFTL_vldb13.pdf

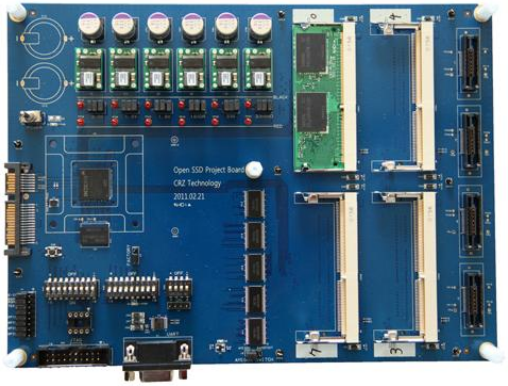
Advantages of NoFTL

- Predictable performance
- Usage of host resources for FTL-like functionalities
- Reduced functional redundancy on the I/O path
- Better utilization of Flash parallelism
  - Data placement
  - Buffer manager
- “Build-in” atomicity of write IOs
- “Multiple-FTLs”
- etc.



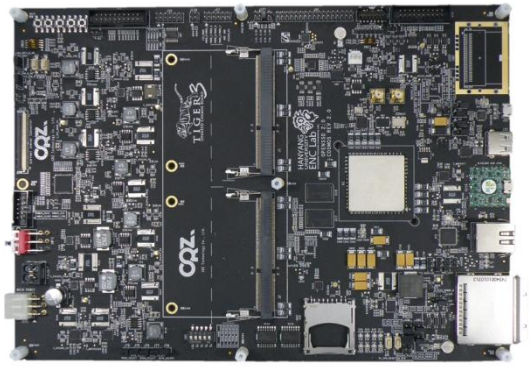
Shore-MT, OpenSSD, Linux driver

- DBMS
  - Shore-MT (C++)[1]
  - Modification of:
    - Buffer Manager
    - Storage Manager
    - Log Manager
    - TRX Manager



Jasmine OpenSSD Platform[2]

- SSD / Flash
  - Jasmine / Cosmos board (C, programming controller)
  - Flash emulator (C, Linux device driver)



Cosmos OpenSSD Platform[3]

[1] https://sites.google.com/site/shoremt/

[2] http://www.openssd-project.org/wiki/Jasmine_OpenSSD_Platform

[3] http://www.openssd-project.org/wiki/Cosmos_OpenSSD_Platform