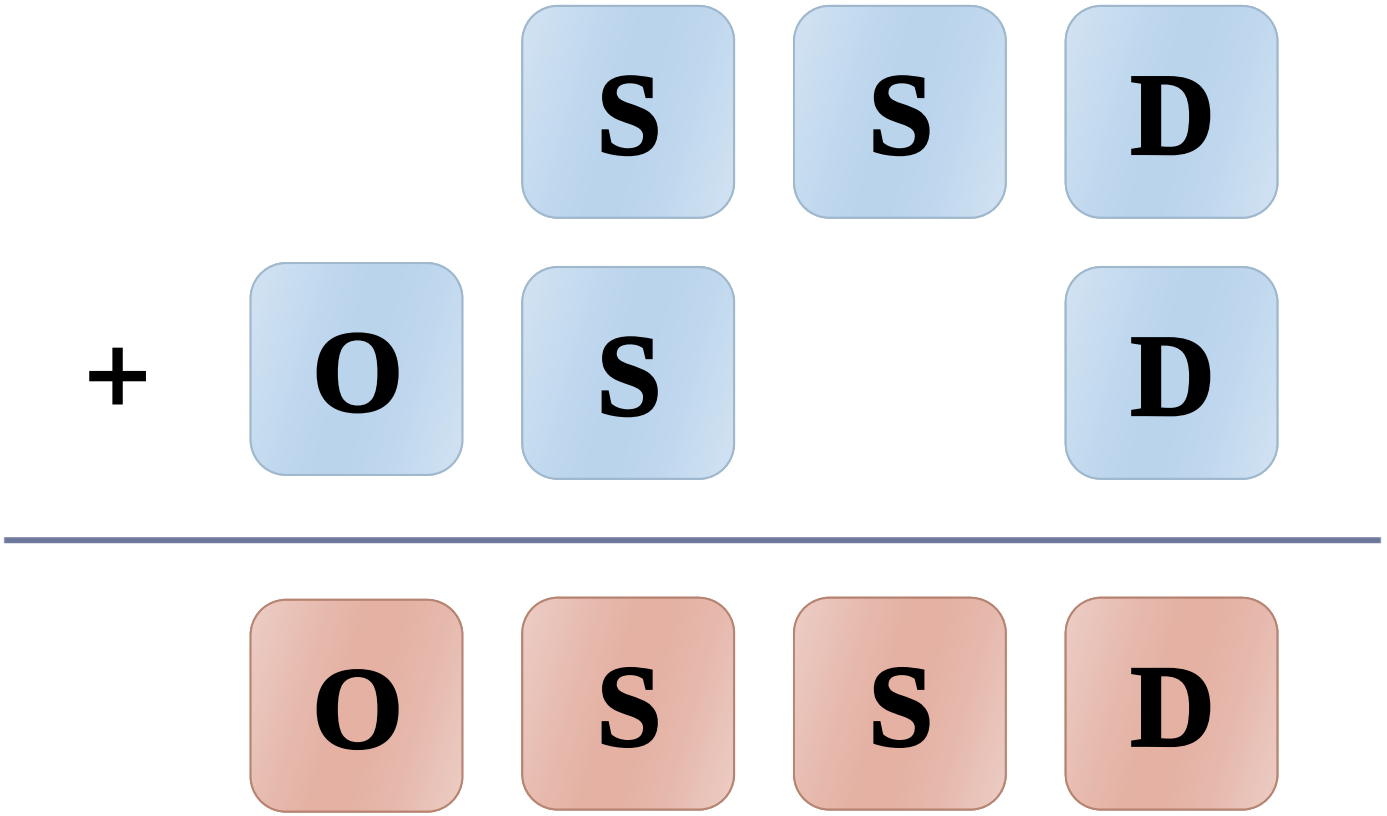
OSSD: Object-based Solid State Drive

Jin-Soo Kim (*jinsookim@skku.edu*)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu

OSSD



Motivation

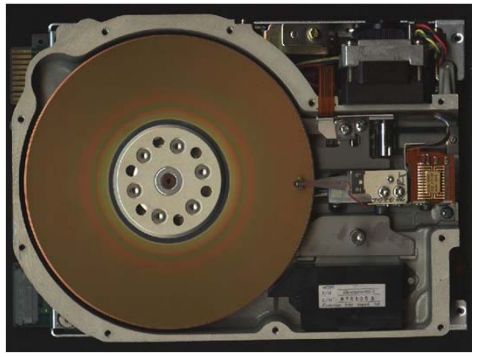
Storage Interface Evolution

1980, Seagate

1986, SCSI
1994, ATA

ST-506

Logical block addressing



"Dumb"

"Integrated controller"

More intelligence?

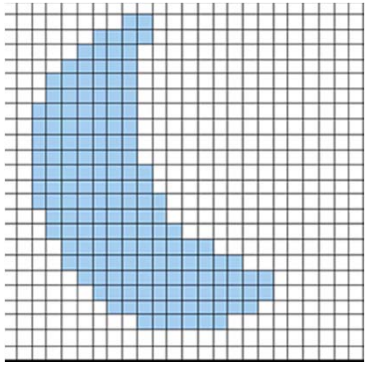
Graphics Interface Analogy

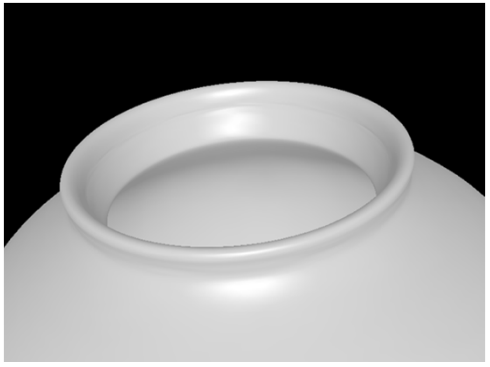
Setpoint (x, y);

Drawline (x1, x2, y1, y2);
Drawtext (x, y, “Hello”);
…

Polygons,
Lighting,
Z-buffering,
Texture mapping,
Alpha blending, …

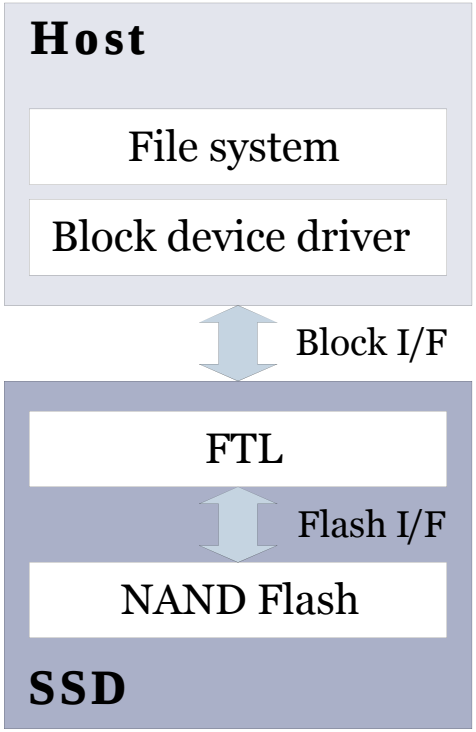
Raster graphics → Vector graphics → OpenGL Direct3D





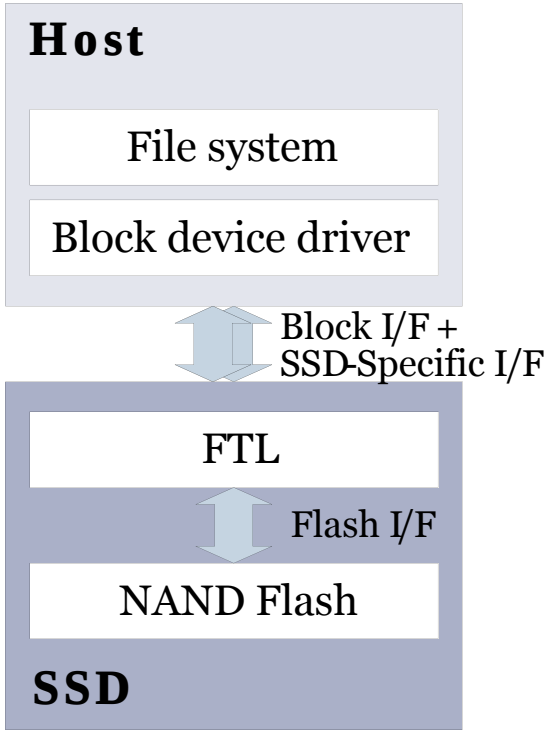
Interfaces for SSDs (1)

- Traditional block interface
  - ATA/SCSI
  - No block-level liveness information
  - No high-level semantics on data
  - Several “unwritten contracts” do not hold for SSDs



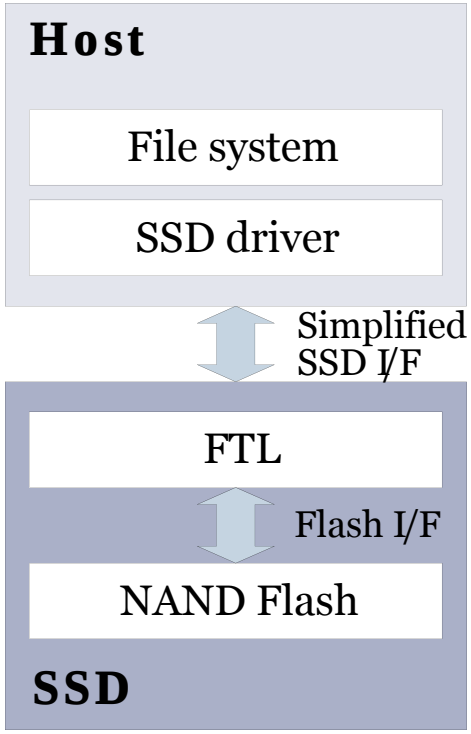
Interfaces for SSDs (2)

- Block interface with SSD extensions
  - TRIM command
  - What else?
  - Not practical, sometimes impossible



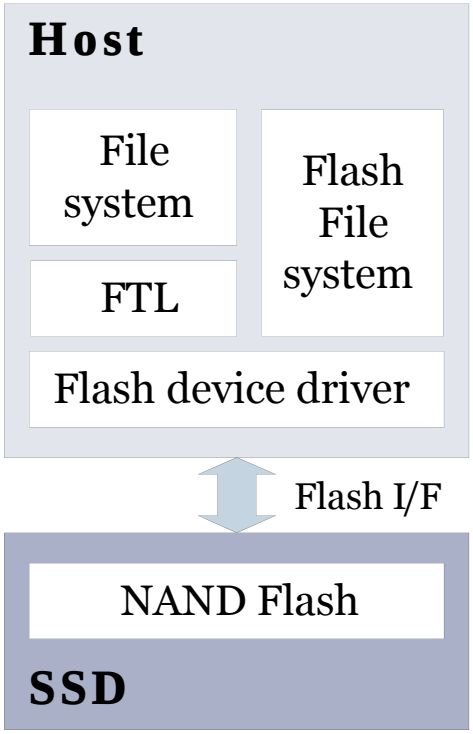
Interfaces for SSDs (3)

- Simplified SSD interface
  - Interfaces specific to SSDs
  - NVMHCI (Non-Volatile Memory Host Controller Interface)
  - Mainly for PCIe-based SSDs
  - Still works at the block level



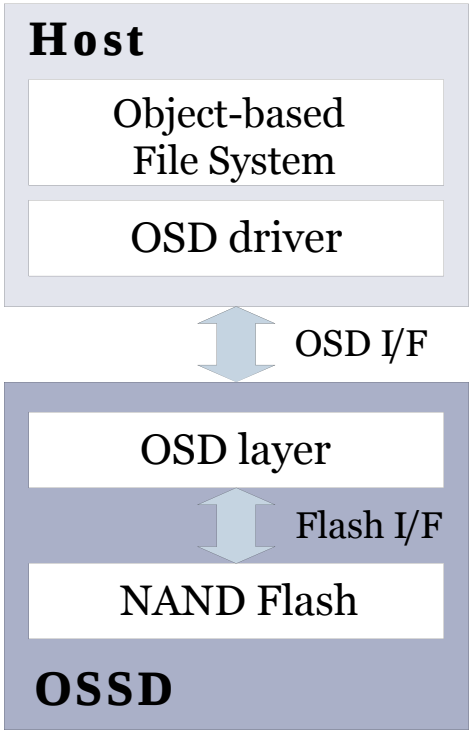
Interfaces for SSDs (4)

- Native flash interface
  - READ PAGE, PROGRAM PAGE, ERASE BLOCK
  - Used in many embedded systems
  - Host handles everything, storage remains dumb and simple



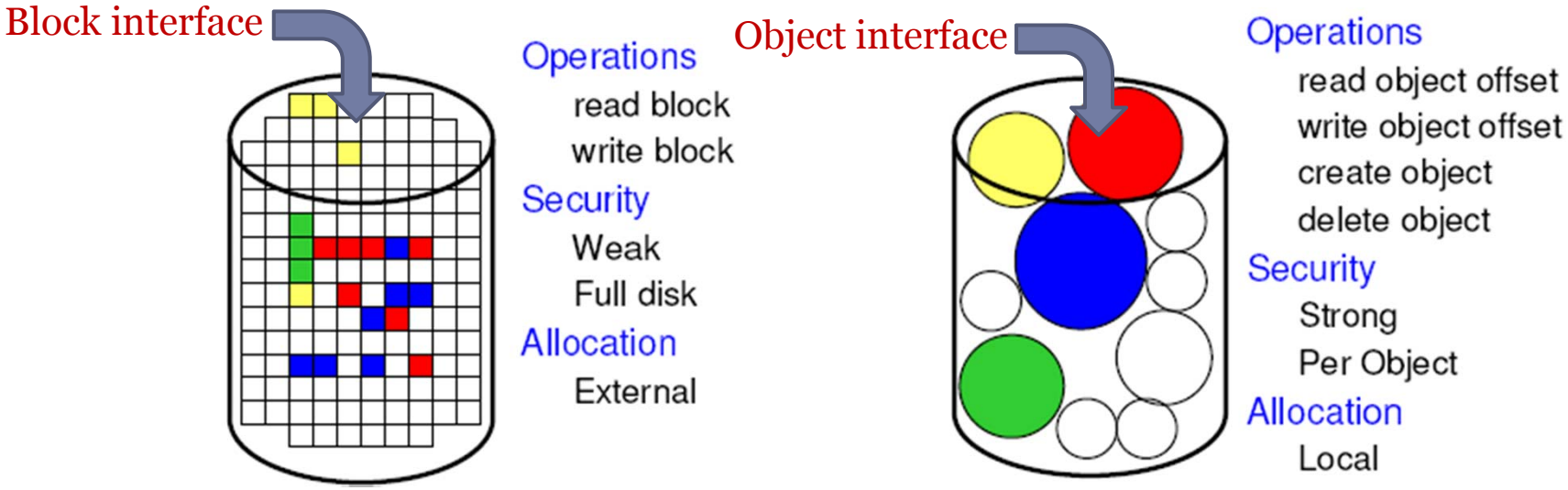
Interfaces for SSDs (5)

- Object interface
  - More abstract interface
  - Improved drive intelligence
  - OSSD:
    Object-based Solid State Drive

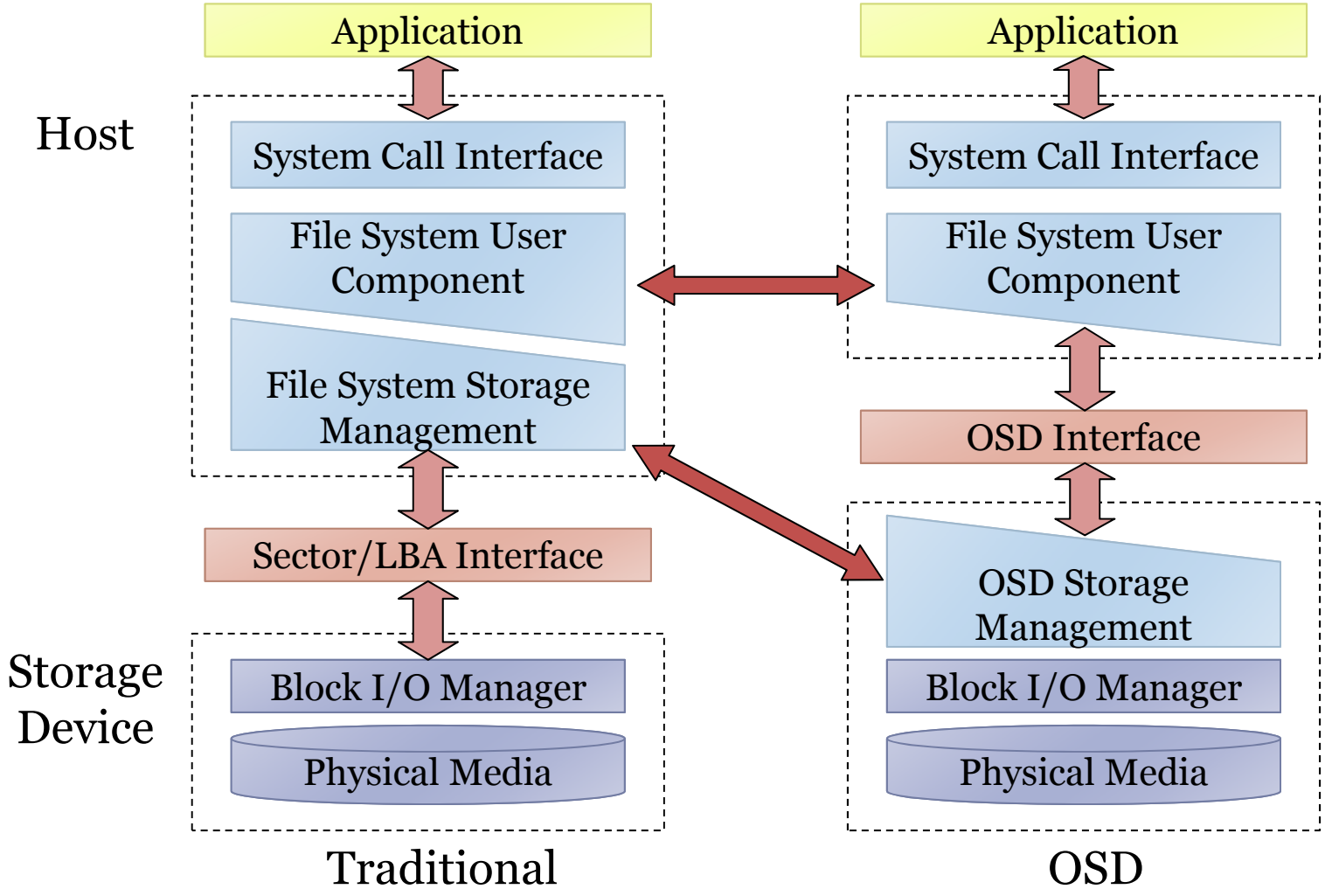


OSD

- Object-based Storage Device
  - Virtualizes physical storage as a pool of objects
  - Offloads space management to storage devices
  - Standardized as a subset of SCSI command set

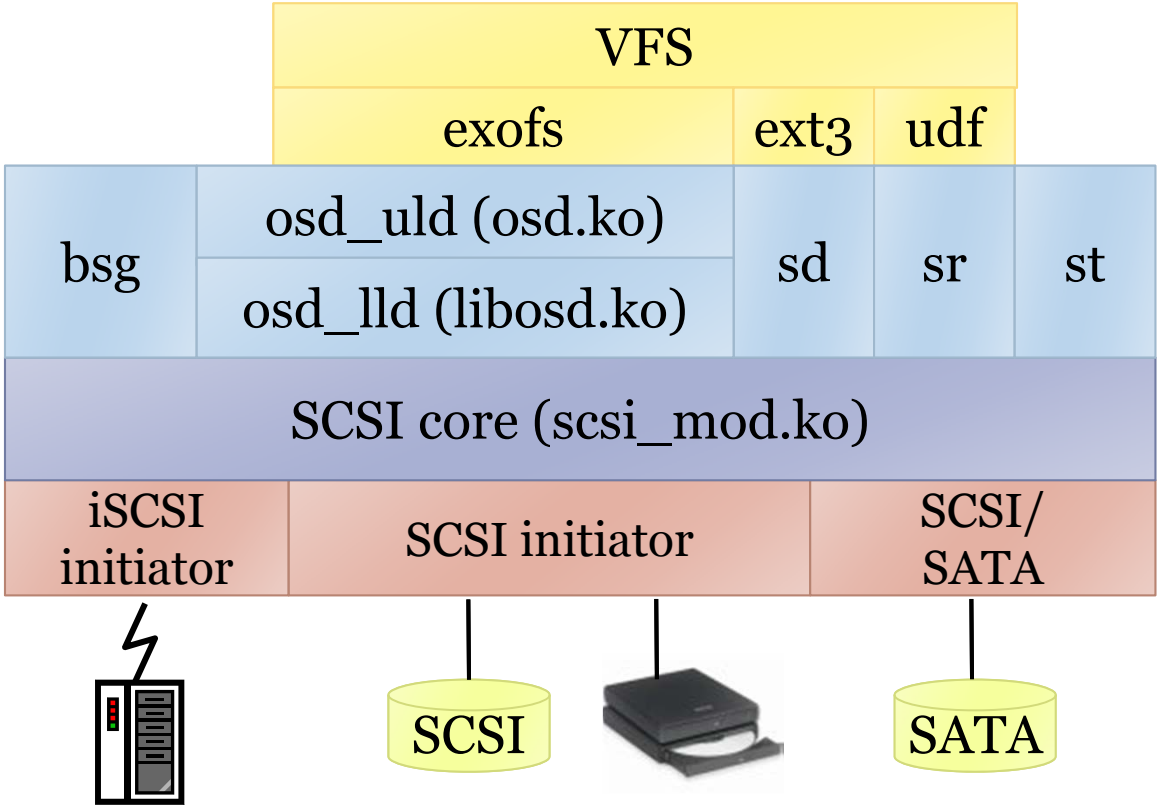


Storage Model



Linux Support for OSD

- Since Linux kernel 2.6.30
  - Exofs + OSD initiator + iSCSI transport



OSSD Benefits (1)

- Simplified host file system
  - No need for SSD-specific parameter tuning
  - File-to-object mapping & access control

- More efficient management of flash storage
  - Block-level liveness
  - Metadata separation
  - Object-aware storage management

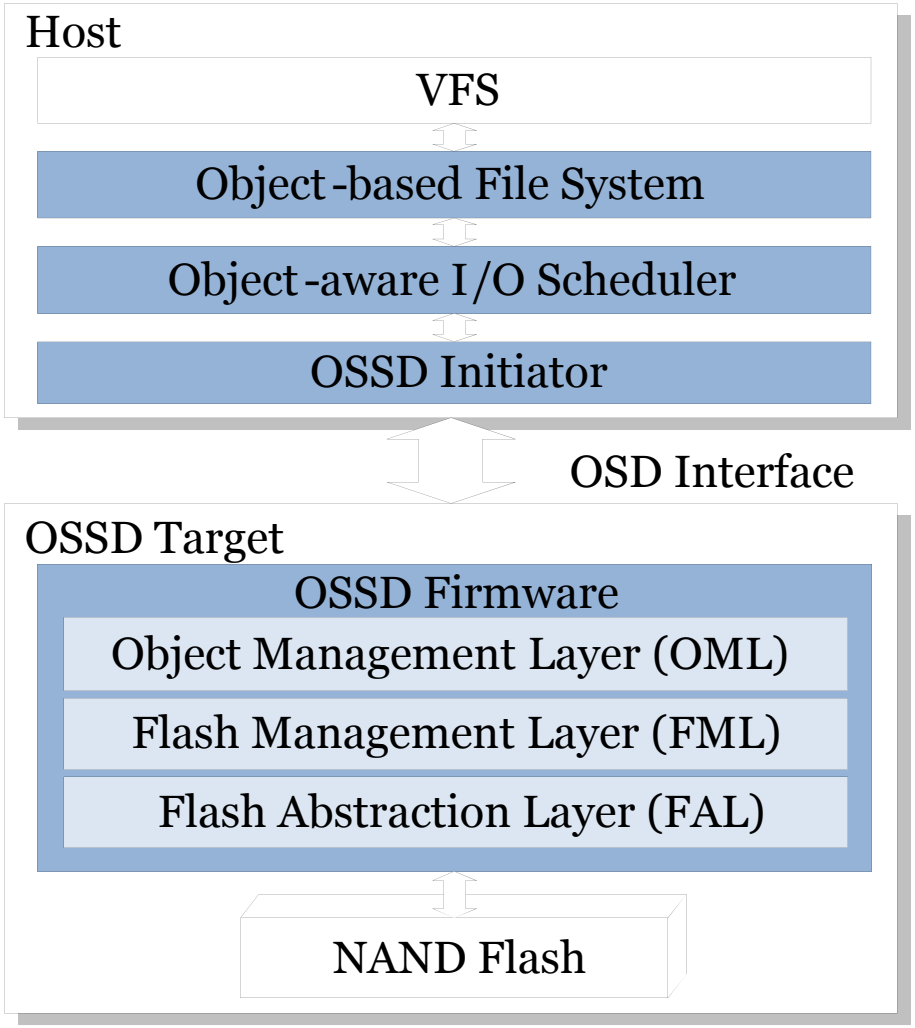
OSSD Benefits (2)

- Application-aware storage management
  - QoS requirements via attributes
  - Application hints

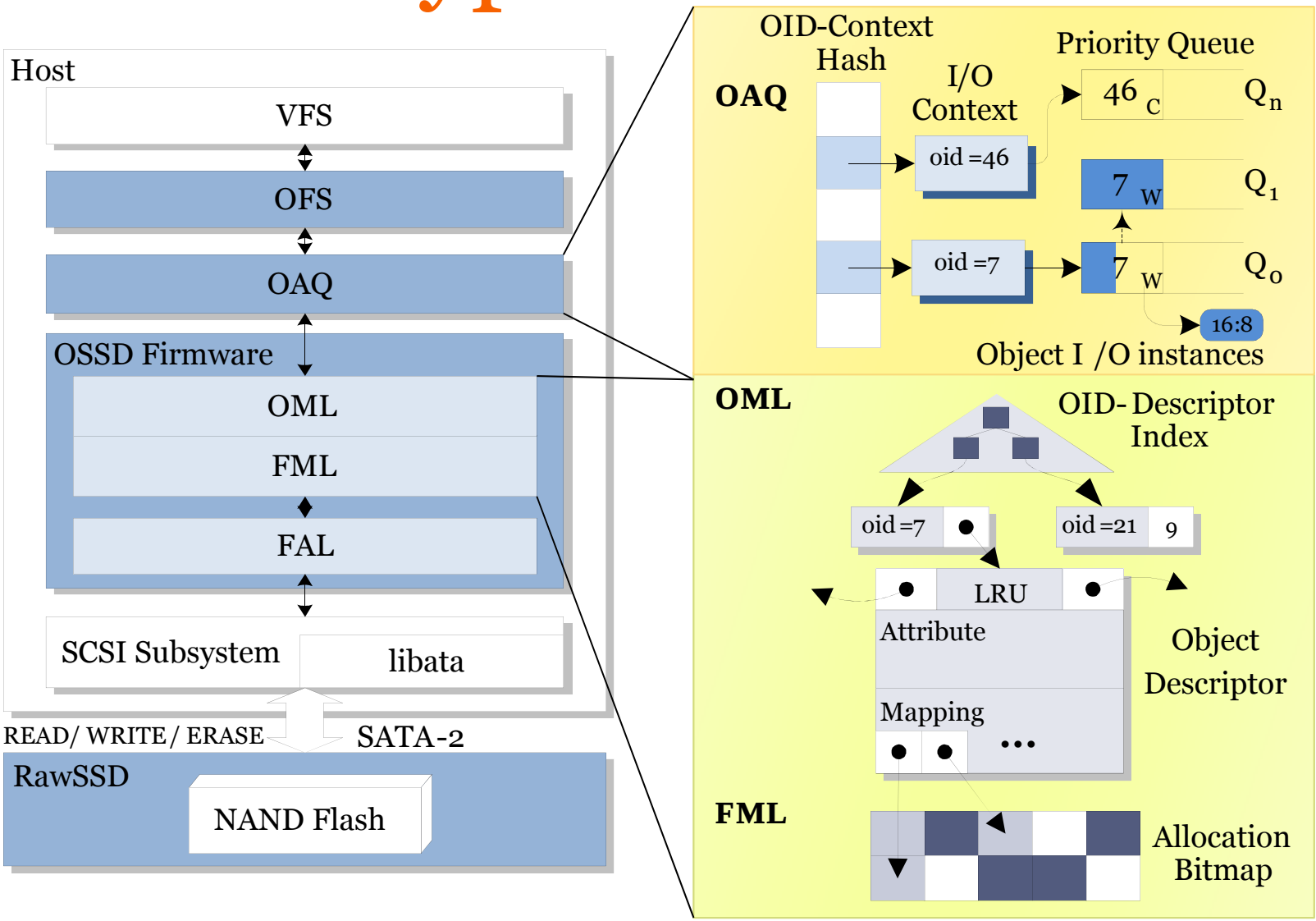
- Storage virtualization
  - Dynamic storage pool
  - Multi-tier storage architecture with DRAM, PCM, SLC NAND, MLC NAND, and HDD

Prototype Implementation

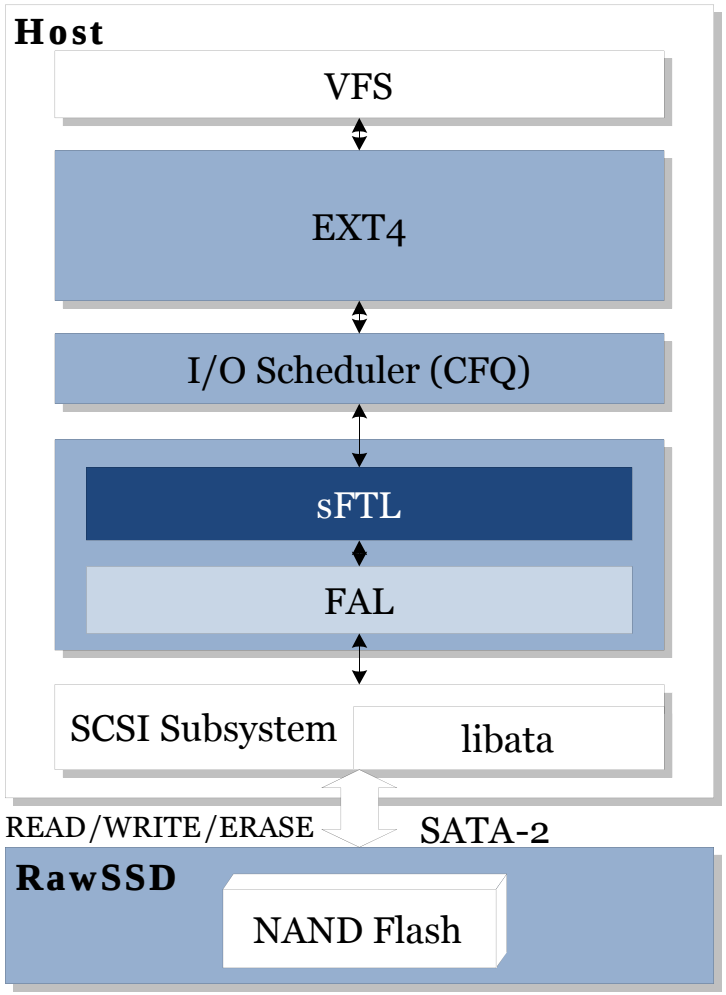
Overall Architecture



Prototype Architecture



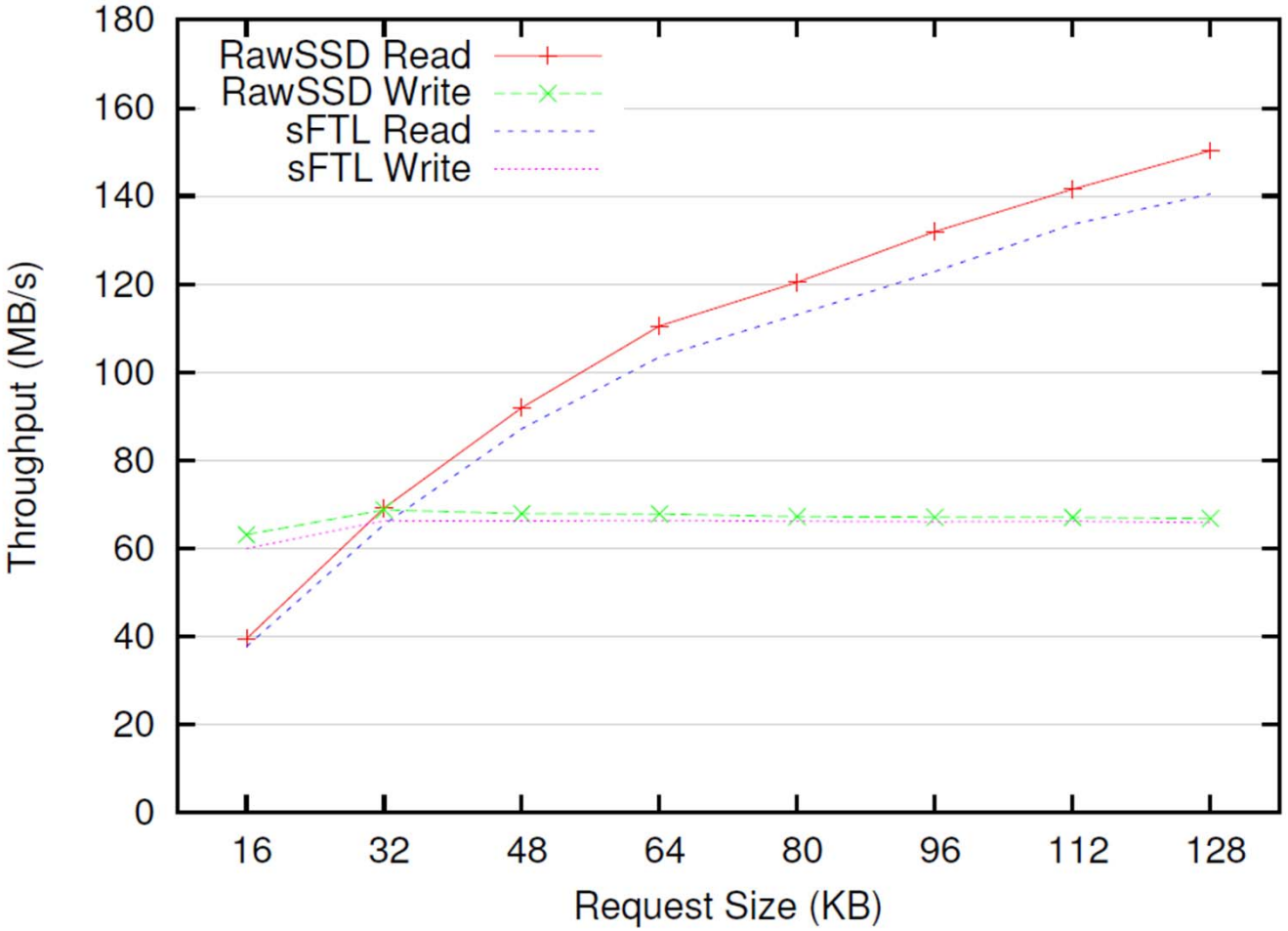
Legacy Stack



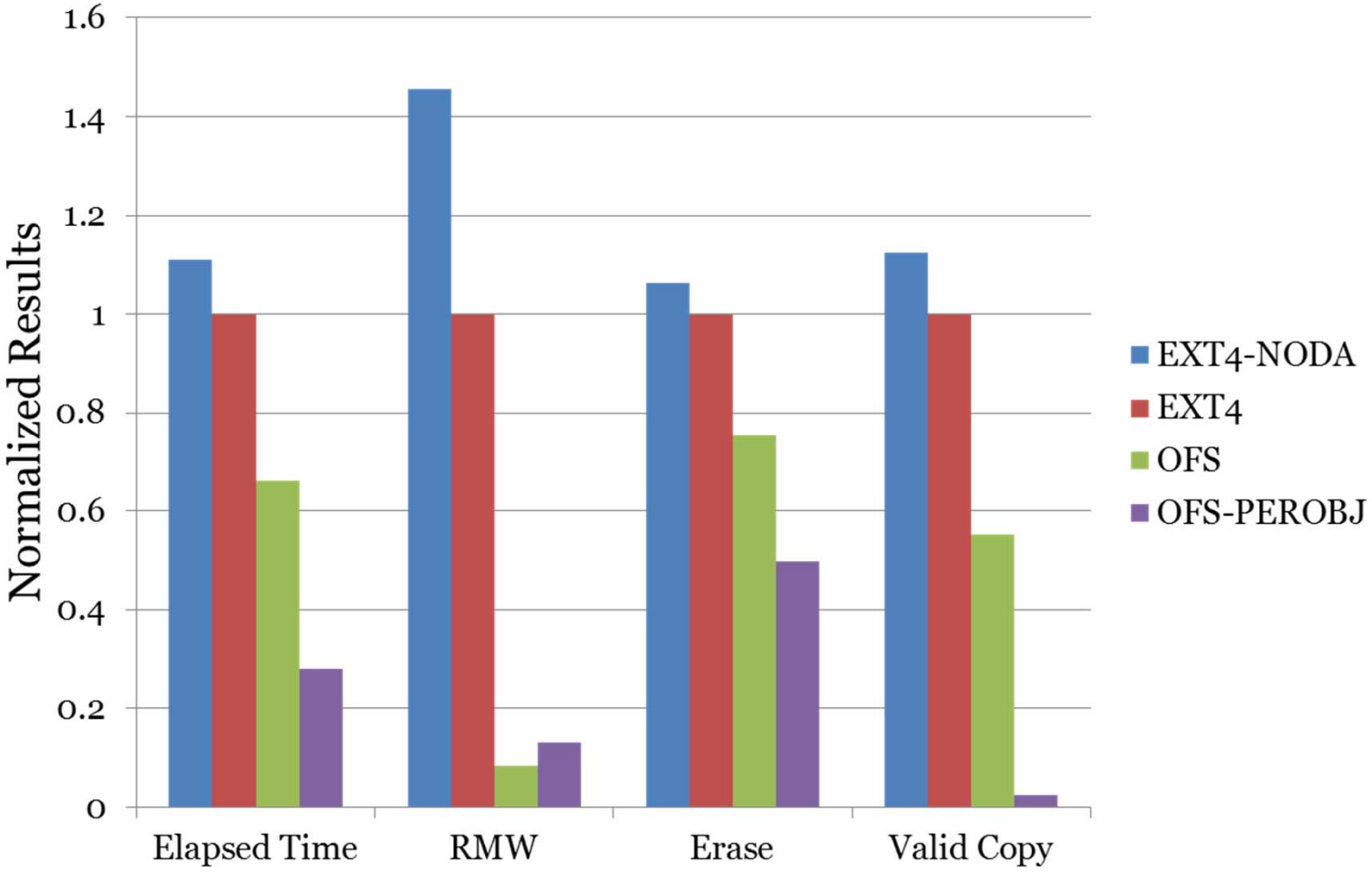
RawSSD

- Native NAND I/O interface over SATA-2
- 16 chips * 8 dies/chip * 2GB/die = 256GB
- 8 channels (2 channels logically combined)
- 16KB logical page, 8MB logical block
  - Two dies with two-plane operation
- Page-wise channel-level striping
- Asynchronous writes with DRAM buffer

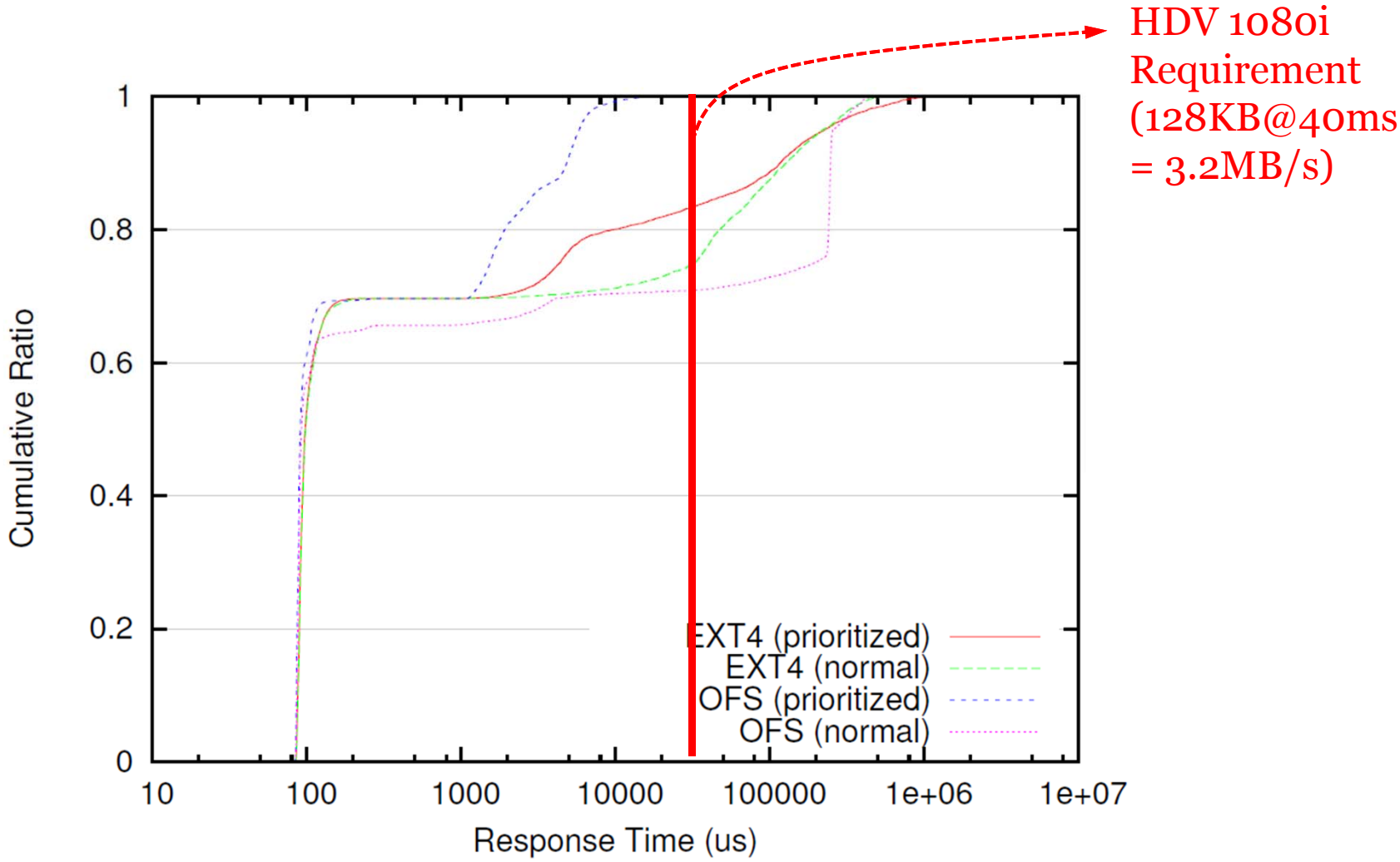
Base Performance



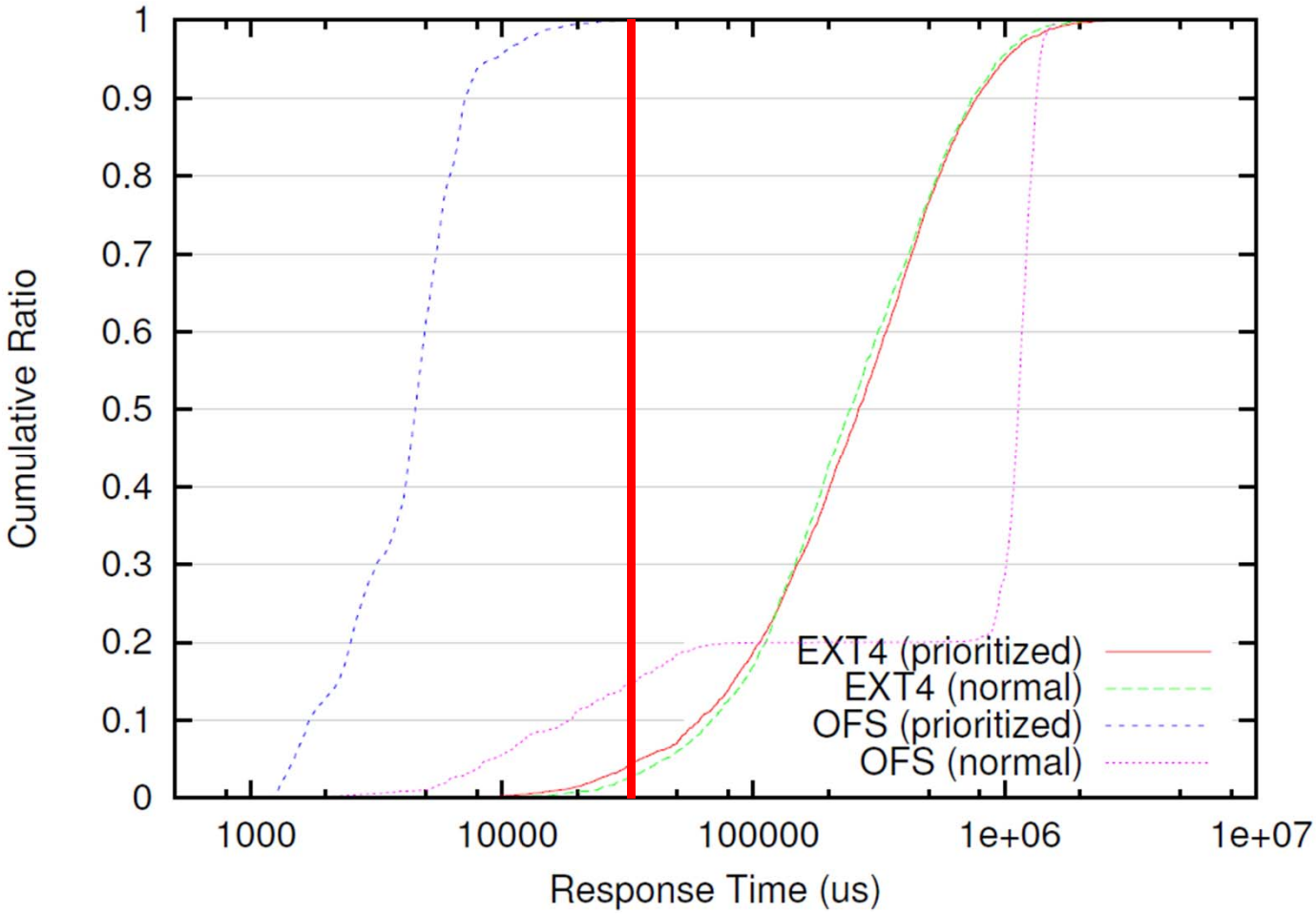
Multi-threaded writes



Prioritized Reads

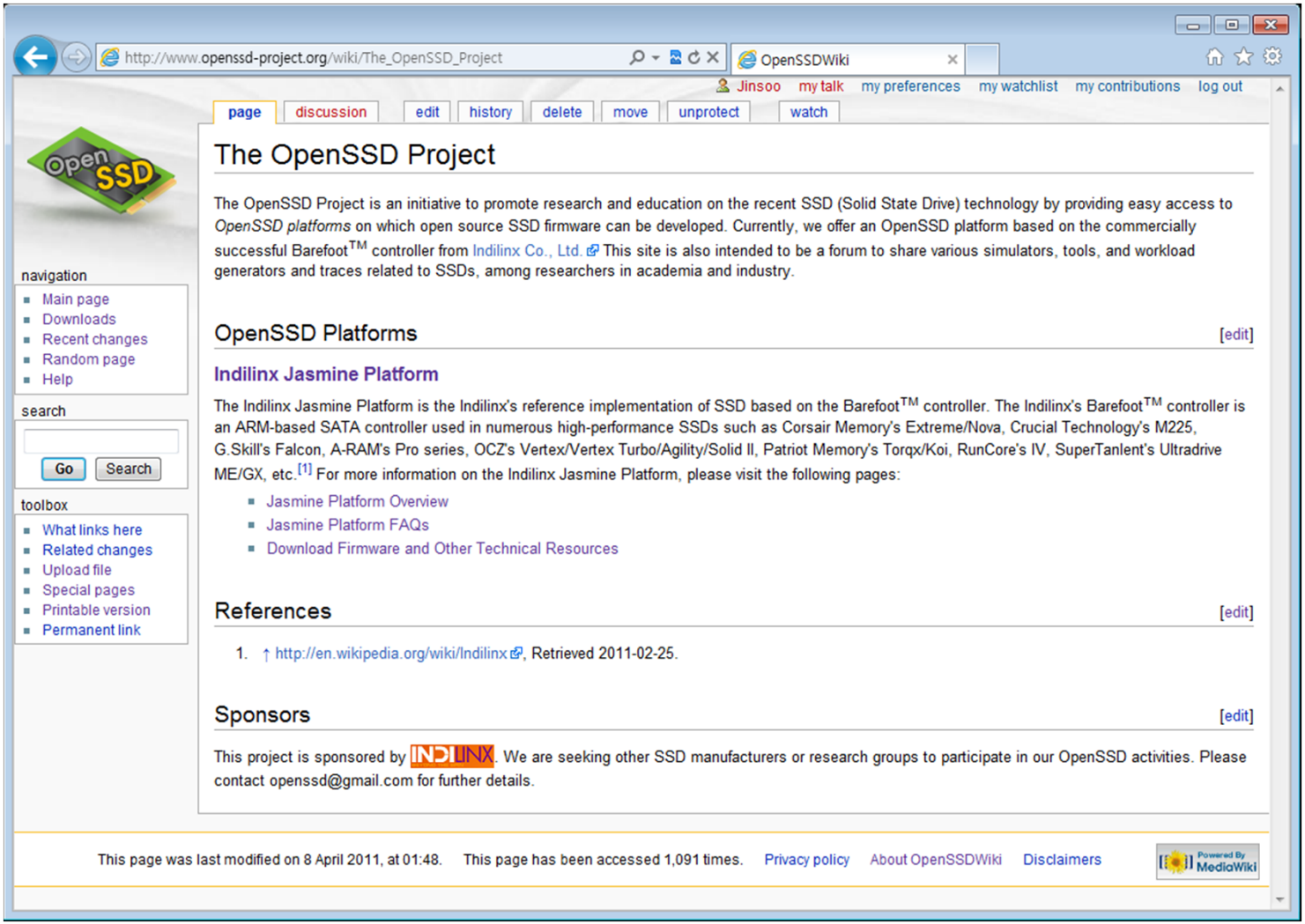


Prioritized Writes



The OpenSSD Project

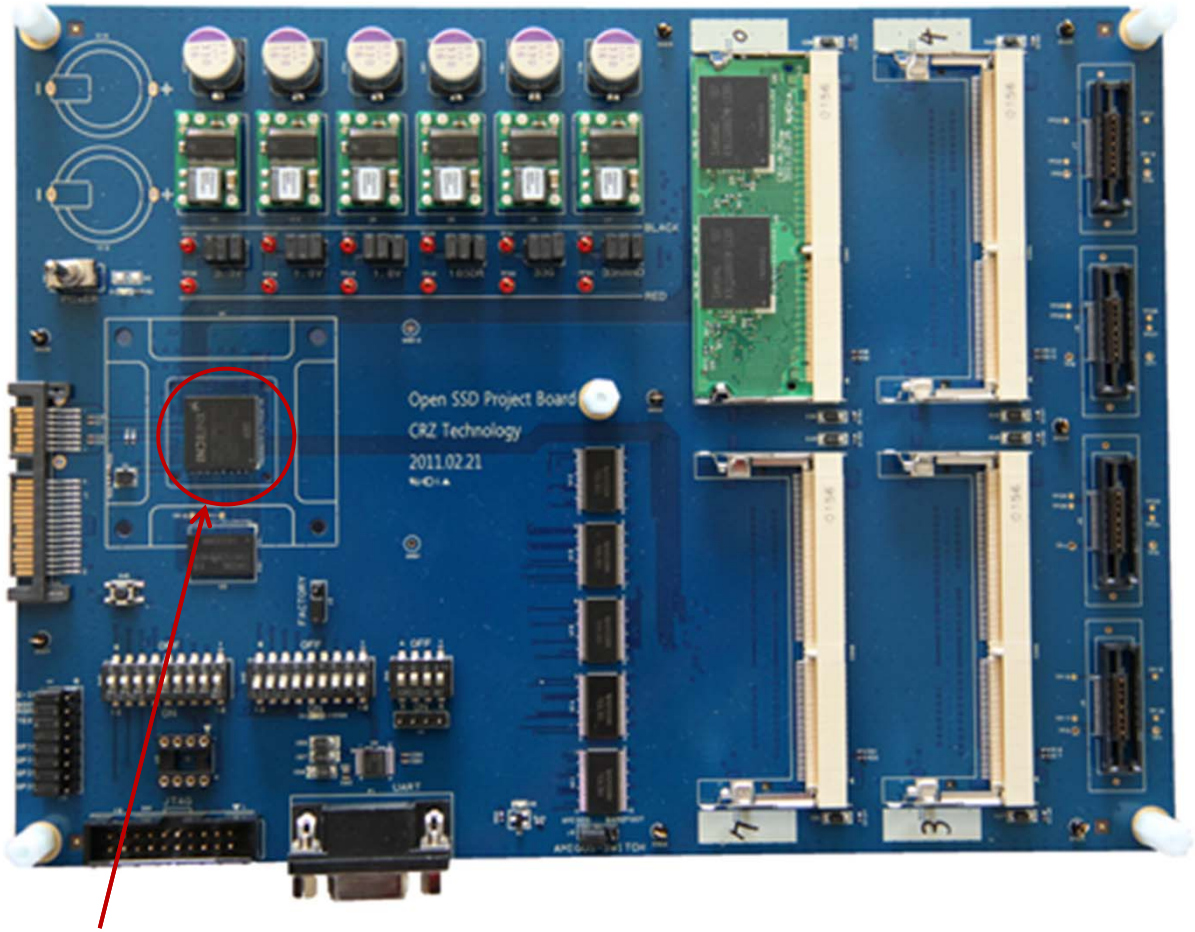
http://www.openssd-project.org



Why OpenSSD?

- No more simulations
- Broaden research horizon
- Train people
- Share expertise
- …
- Initiated with Prof. Sang-Won Lee (SKKU) and Indilinx

Jasmine OpenSSD Platform



Jasmine Firmware

- Firmware 1.0.0 released on April 7, 2011
- The latest version: 1.0.2 (April 15, 2011)
- Three sample FTLs
  - Tutorial FTL (By Indilinx)
  - Greedy FTL (By VLDB Lab. @ SKKU)
  - Dummy FTL (By Indilinx)
- Toolchains
  - ARM RVDS
  - Codesourcery G++ lite edition

Participants

- 30 sets shipped to 9 institutions (15 labs)
  - Sungkyunkwan U., Hanyang U., Ajou U., Hongik U., Korea U., Kwangwoon U., POSTECH, Soongsil U., U of Seoul
- 1 set shipped abroad

- We plan to have a workshop in May.

Jasmine Resources

- FAQs
- Board schematics
- Technical Reference Manual
- FTL Developer's Guide
- Barefoot Controller Technical Reference
- Contributions from community
  - Developer Information (by Jeremy Brock)
  - …

Thank You!