Why are state-of-the-art flash-based multi-tiered storage systems performing poorly for HTTP video streaming?

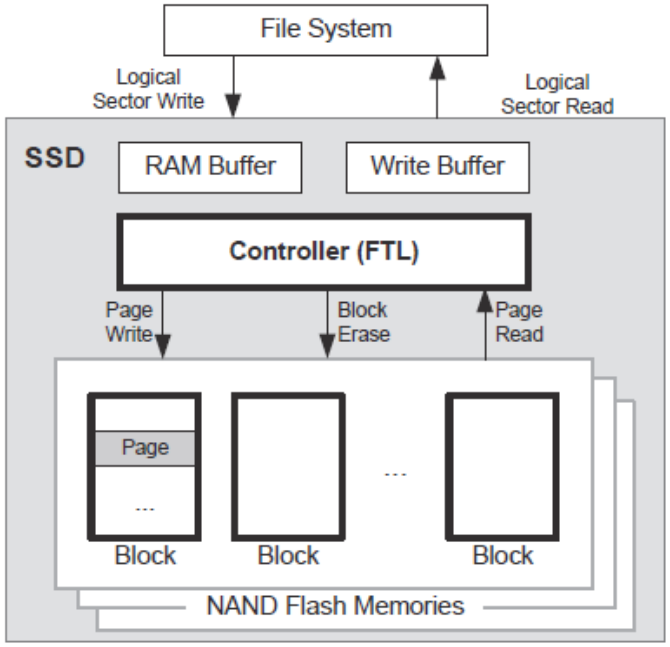
Moonkyung Ryu
Hyojun Kim
Umakishore Ramachandran

Georgia Institute of Technology

Contents

- Background
  - Flash Memory SSD
  - Adaptive HTTP Video Streaming
- Why Multi-tiered Storage System?
- State-of-the-art Multi-tiered Storage Systems
- Evaluation
- Analysis
- Recommendations
- Conclusion

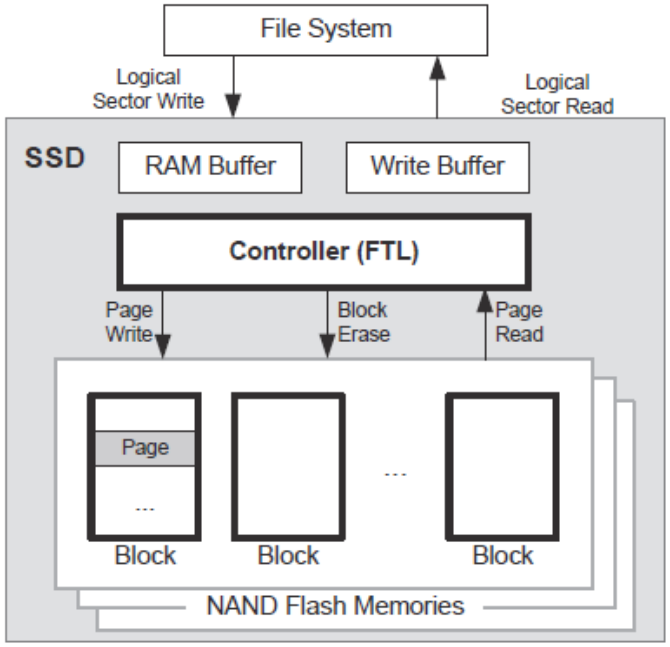
Flash Memory



- Page size
  - 4 ~ 8 KB
- Block size
  - 256 ~ 512 pages
- Constraints
  - Read/Write in a PAGE unit
  - Erase in a BLOCK unit
  - No in-place update
    - No write on a page unless it's clean
  - Limited erase count
    - SLC (100,000), MLC (10,000)
    - Wear-leveling

Access Time			
Type	Page Read (us)	Page Write (us)	Block Erase (ms)
SLC	25	250	1.5
MLC	60	900	3.5

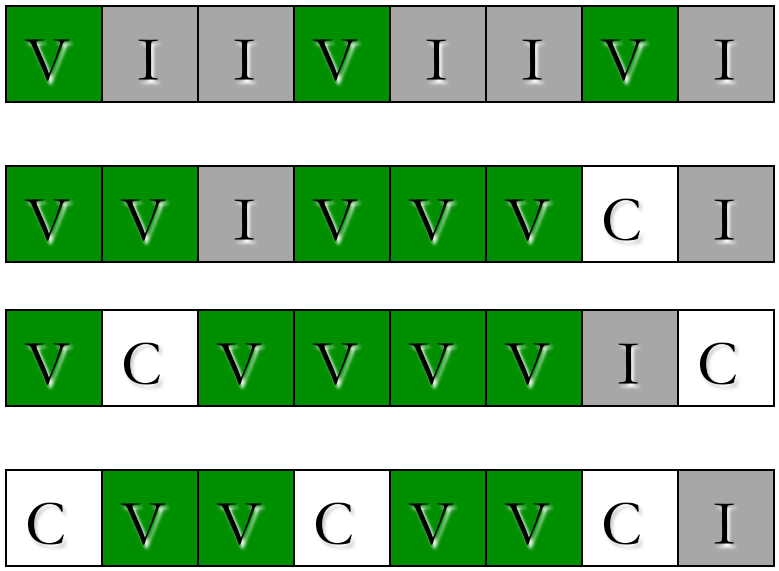
Flash Memory



- Flash Translation Layer
  - Mapping table on RAM
    - Virtual address (exposed to upper level) → physical address (on flash chips)
  - Page/Block/Hybrid mapping
  - Hides erase operation to upper level
- Logging technique
  - Over-provisioning
    - 5 ~ 10 % for low-end SSD
    - 30 ~ 50 % for high-end SSD
  - Garbage Collection
    - Critical to performance

Flash Memory

- Garbage Collection

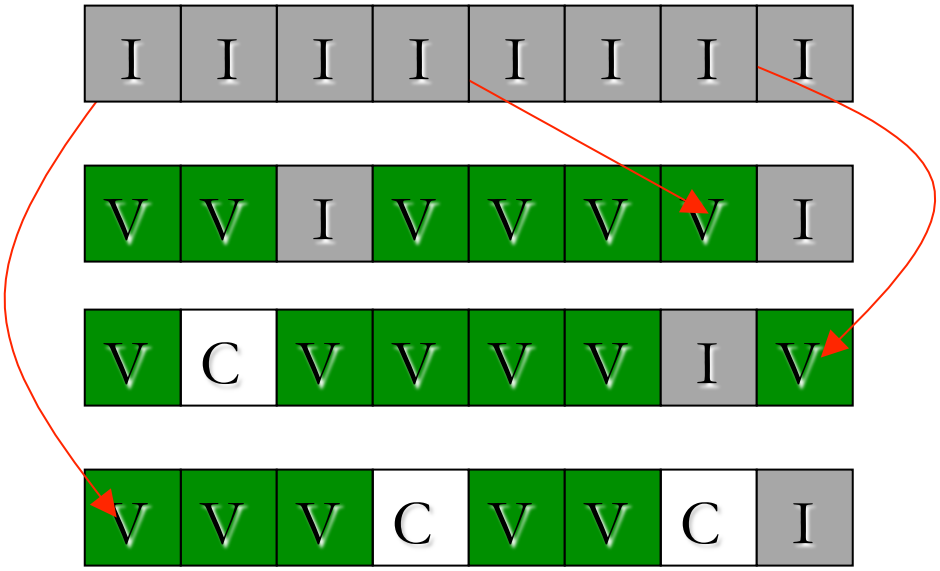


This block is to be recycled.
(3 valid pages and 5 invalid pages)

Valid page
Invalid page
Clean page

Flash Memory

- Garbage Collection

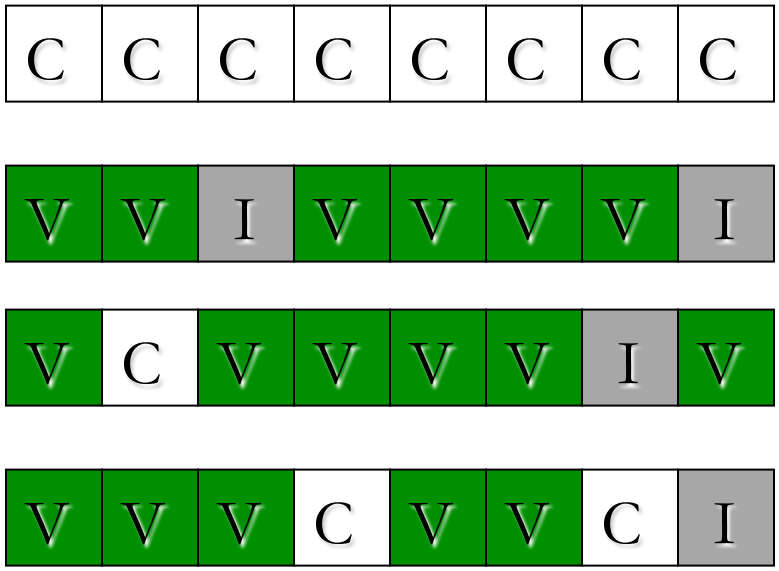


Valid pages are copied to clean pages.

Valid page
Invalid page
Clean page

Flash Memory

- Garbage Collection



The block is then erased.

Overheads:

- Valid page copying
- Block erasing
- Mapping table update

Valid page

Invalid page

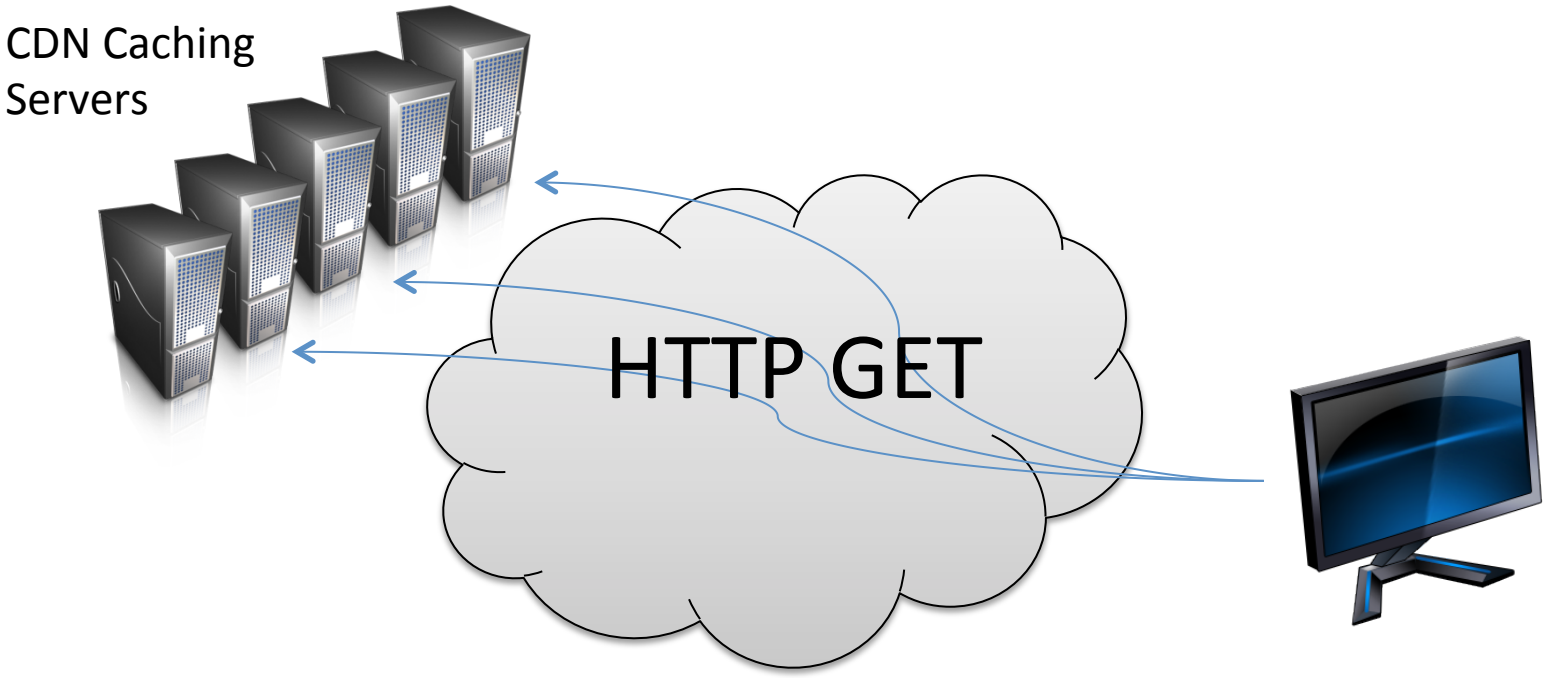
Clean page

Flash Memory

- Pros
  - Fast random read
  - Fast sequential read/write
  - Low power/heat
  - No vibration or noise
- Cons
  - Poor small random write
  - Limited life time
  - Unpredictable/Uncontrollable GC
  - Higher cost/GB than HDDs

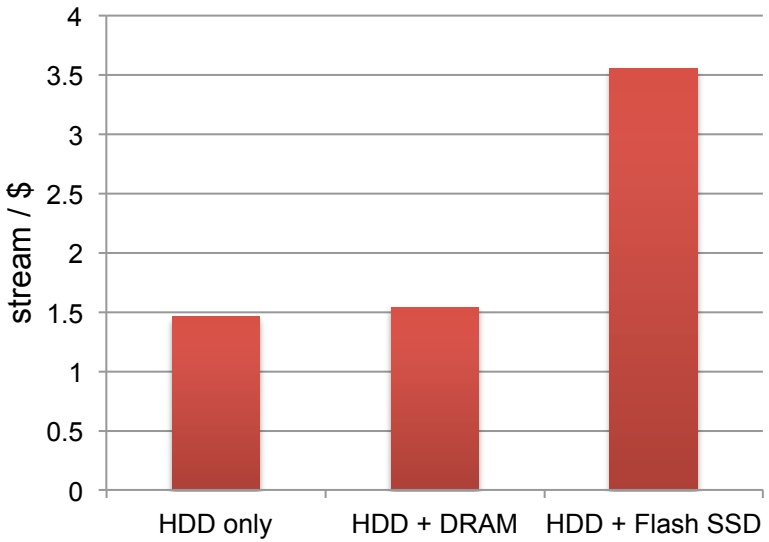
Adaptive HTTP Streaming

- Streaming by WEB servers
- Client adaptively selects different bitrate segments
- NETFLIX



Why Multi-tiered Storage?

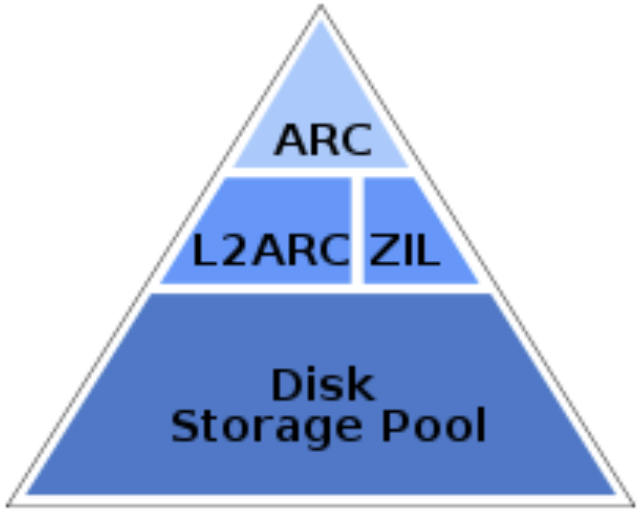
- Highly skewed video access pattern
- Cheap MLC Flash Memory
- Low power and heat



Reference: Ryu et al., "Impact of Flash Memory on Video-on-Demand Storage", In Proc. of MMSys, 2011

Flash Caching Systems

- Flashcache
  - Developed by Facebook
  - Block-device solution
- ZFS
  - Developed by SUN for SOLARIS
  - File-system solution
  - L2ARC is read cache
  - ZIL is write cache

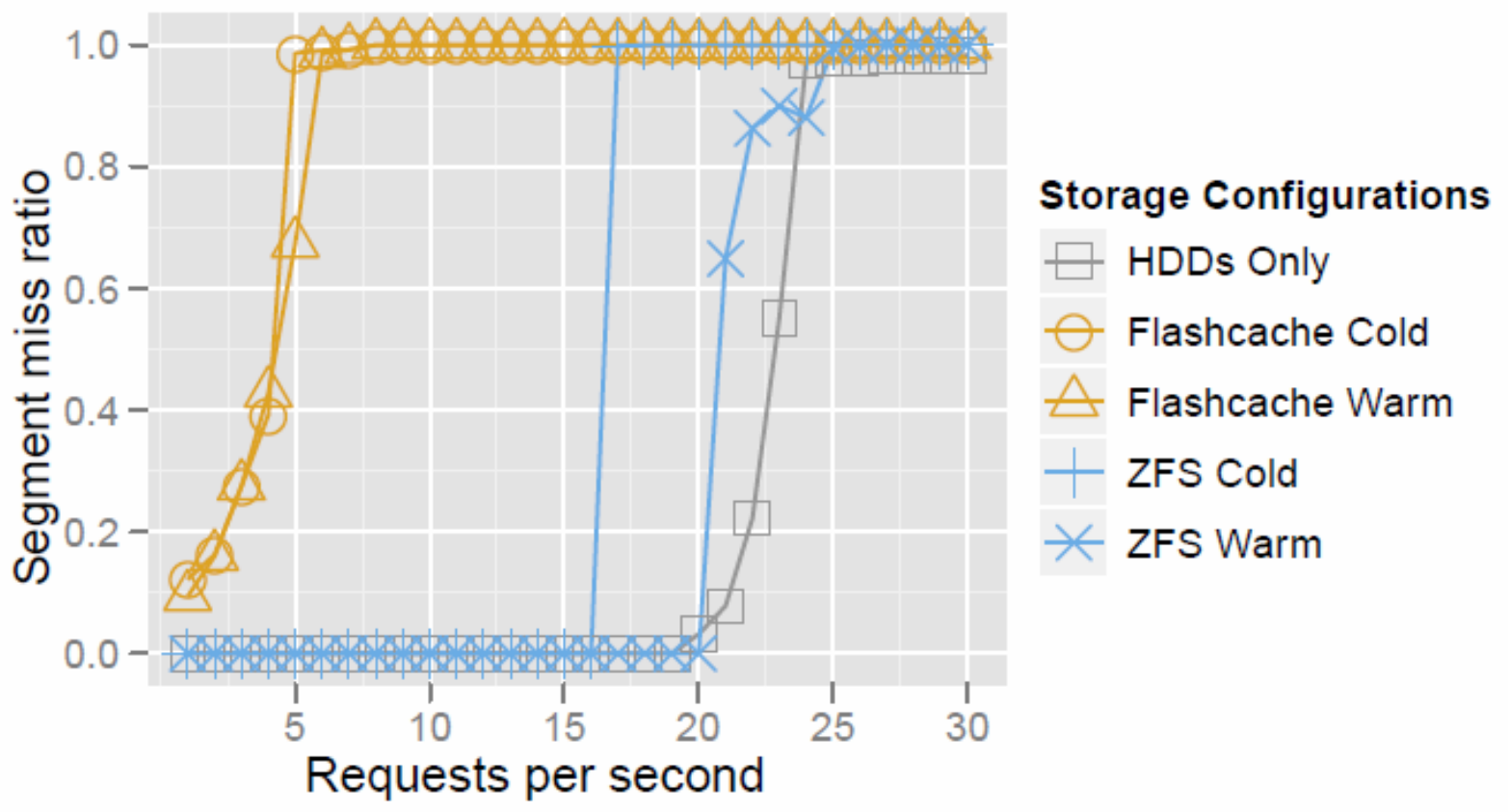


Evaluation

- Apache web server
- Linux kernel 2.6.32
- Workload
  - Video sequence
    - Valkaama (78 mins, 1.06 GB)
    - 2 Mbps on average
    - 10 sec segments (466 total segments)
    - 300 video objects
  - Zipf 0.271
- Hardware
  - Xeon 2.26 GHz Quad core
  - 4 GB RAM
  - Two 7200 RPM HDDs
    - Linux Software RAID-0
  - SSDs
    - INTEL X25-M G1 (60 / 80 GB)
    - OCZ Core V2 (60 / 124 GB)

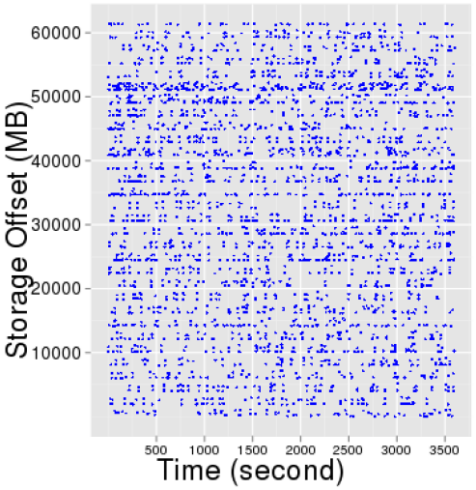
Performance

- Flashcache and ZFS are even worse than HDDs Only!

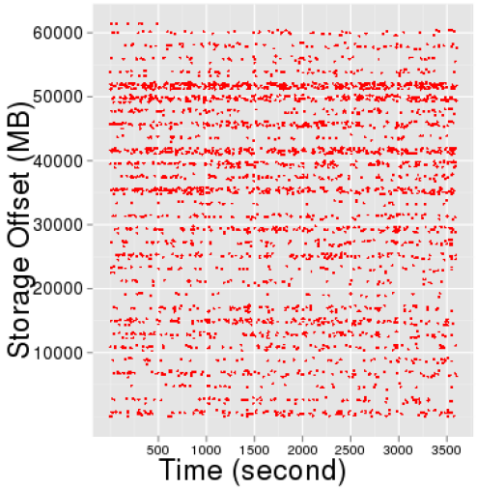


Flashcache Analysis

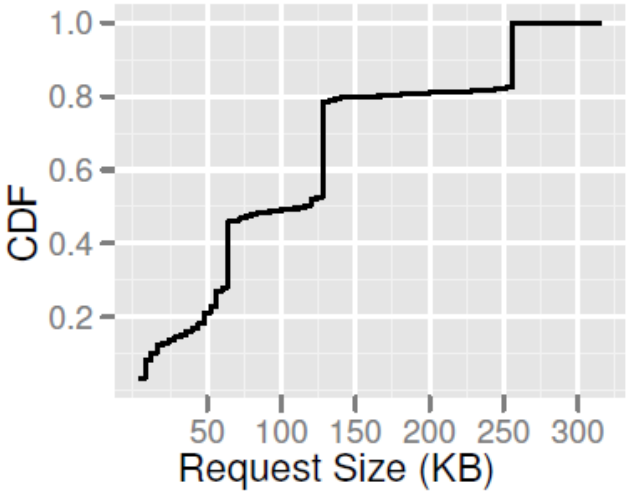
- Severe small random reads/writes



Read

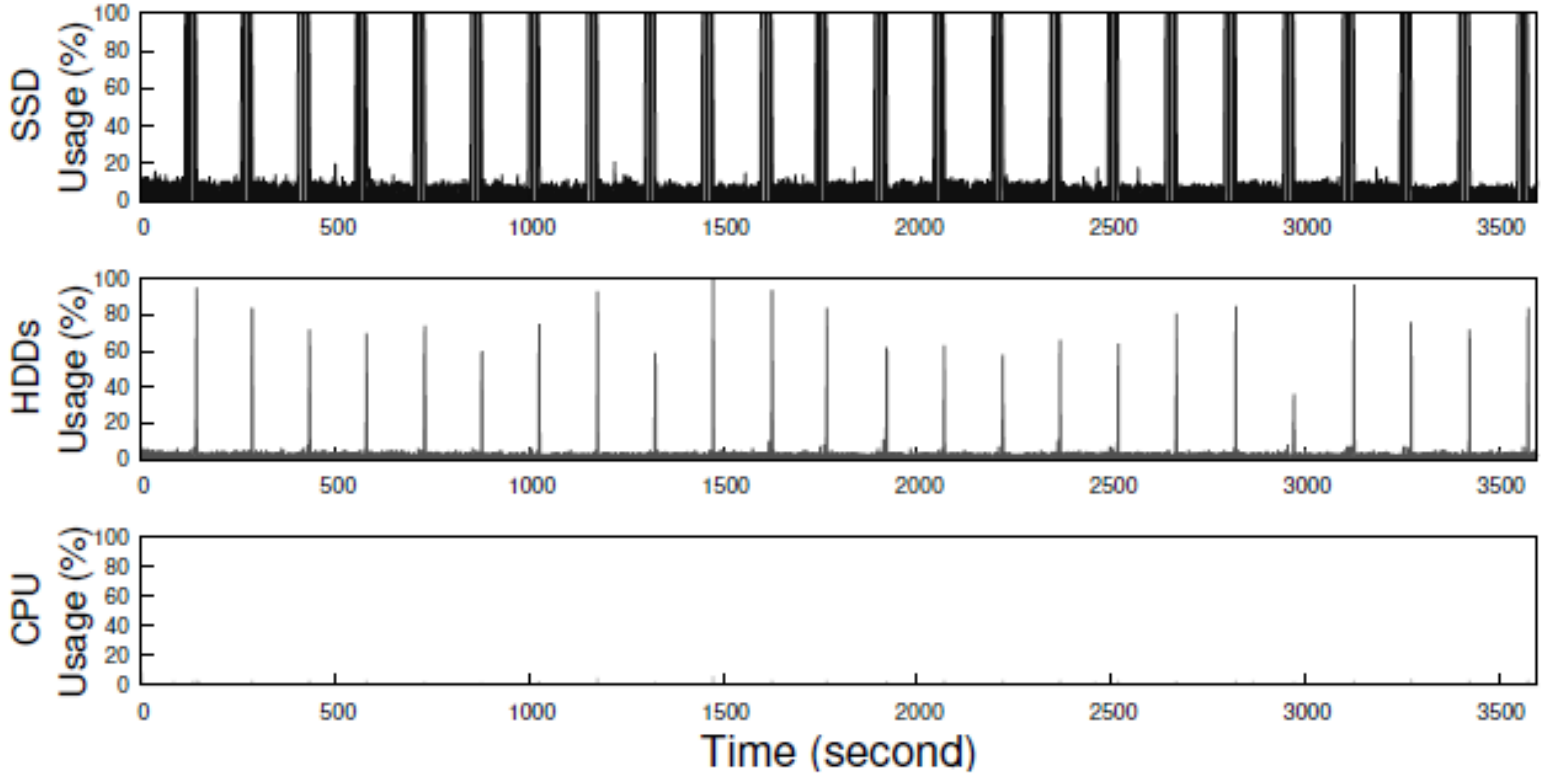


Write



Flashcache Analysis

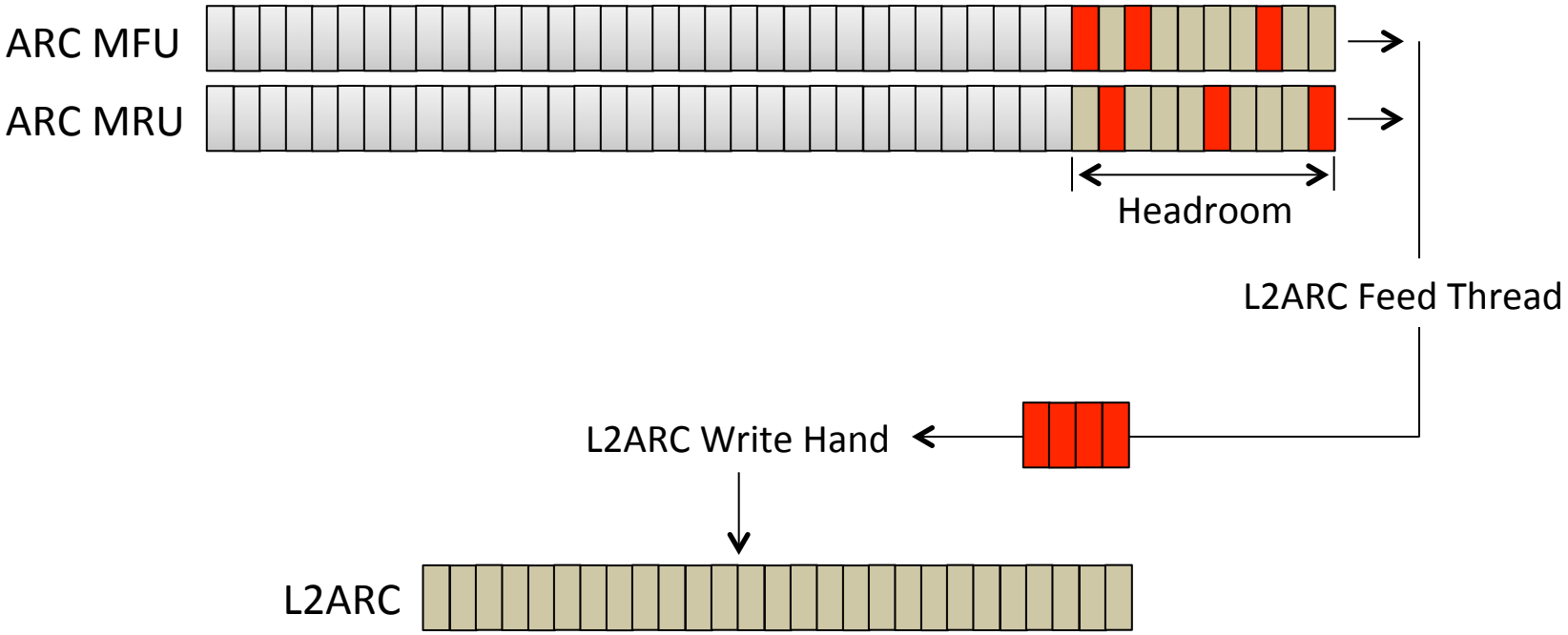
- Severe small random reads/writes
  - Frequent GC in SSD



Flashcache Analysis

- Cache hit ratio
  - Cold cache: 0.7%
  - Warm cache: 25.7%
- Reasons for poor performance
  - Writes to flash memory is on the critical path
  - Small random writes fire frequent GC
  - No priority for reads over writes

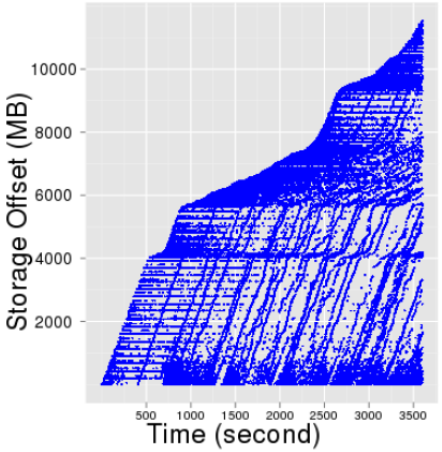
ZFS L2ARC



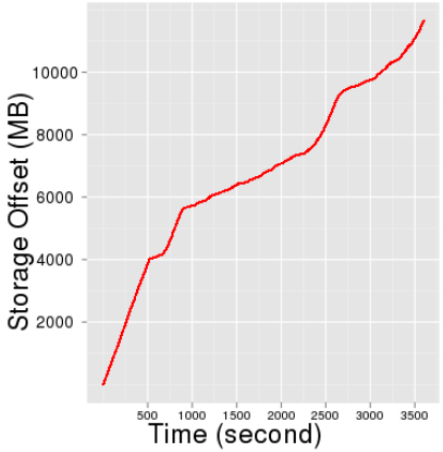
- Evicts ahead
- Flash writes are separated from data serving
- FIFO replacement policy

ZFS Analysis

- Good design decisions
  - Write access pattern to flash is sequential



Read



Write

  - Flash writes are not on the critical path

ZFS Analysis

- Reasons for poor performance
  - FIFO replacement policy for L2ARC
    - Ideal for flash memory write throughput
    - Avoids frequent GC
    - However, does not capture video popularity
    - Low hit ratio (11.1 %)

Recommendations

- No small random writes
- No flash writes on the critical path
- Higher priority for reads
  - Flash reads (serving video) should have higher pri ority than flash writes (caching to flash)
- Object-level caching
  - MLC flash memory is sufficiently cost-effective to hold hot videos as a whole
  - Very good hit ratio (55.4 %) with skewed video acc ess pattern (zipf 0.271)

Conclusion

- Flashcache and ZFS are not properly designed for HTTP video streaming
- New design for multi-tiered video storage system is needed

Thank You!