FlashTier: A Lightweight, Consistent and Durable Storage Cache

Mohit Saxena
PhD Candidate
University of Wisconsin-Madison
msaxena@cs.wisc.edu

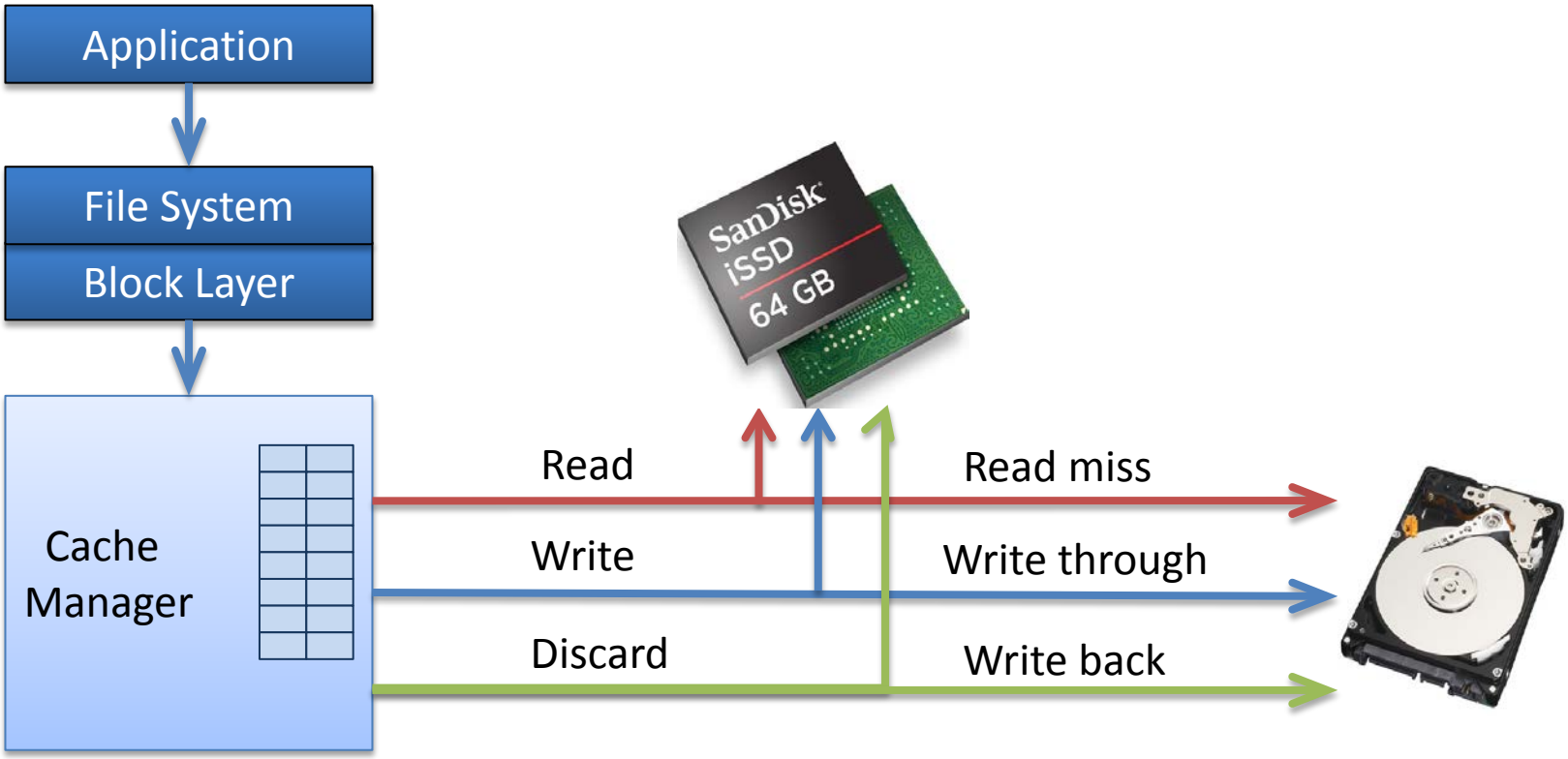
Flash Memory Summit 2012
Santa Clara, CA

Flash is a Good Cache

- Faster than disk: latency and IOPS
- More expensive than disk

- Flash caching is widespread
  - OS vendors: Oracle, Microsoft, Linux
  - Storage vendors: Intel, OCZ, NetApp, EMC, FusionIO, Samsung
  - Applications: Client and Enterprise - Facebook, Google

Block Caching with an SSD



Problems with SSD block cache

P1. Address Space Indirections
- Memory overhead

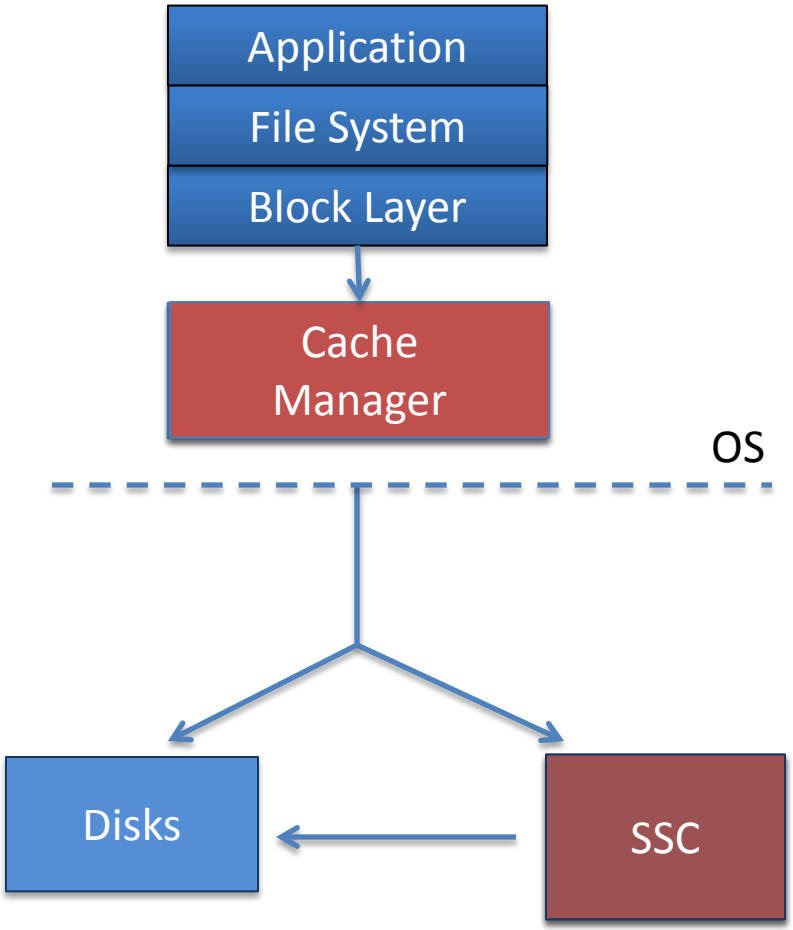
P2. Free Space Management
- Cache performance

P3. Cache Consistency & Durability
- Consistency cost

➢ Observation: **Caching is different from Storage**
- Exploit caching behavior and requirements

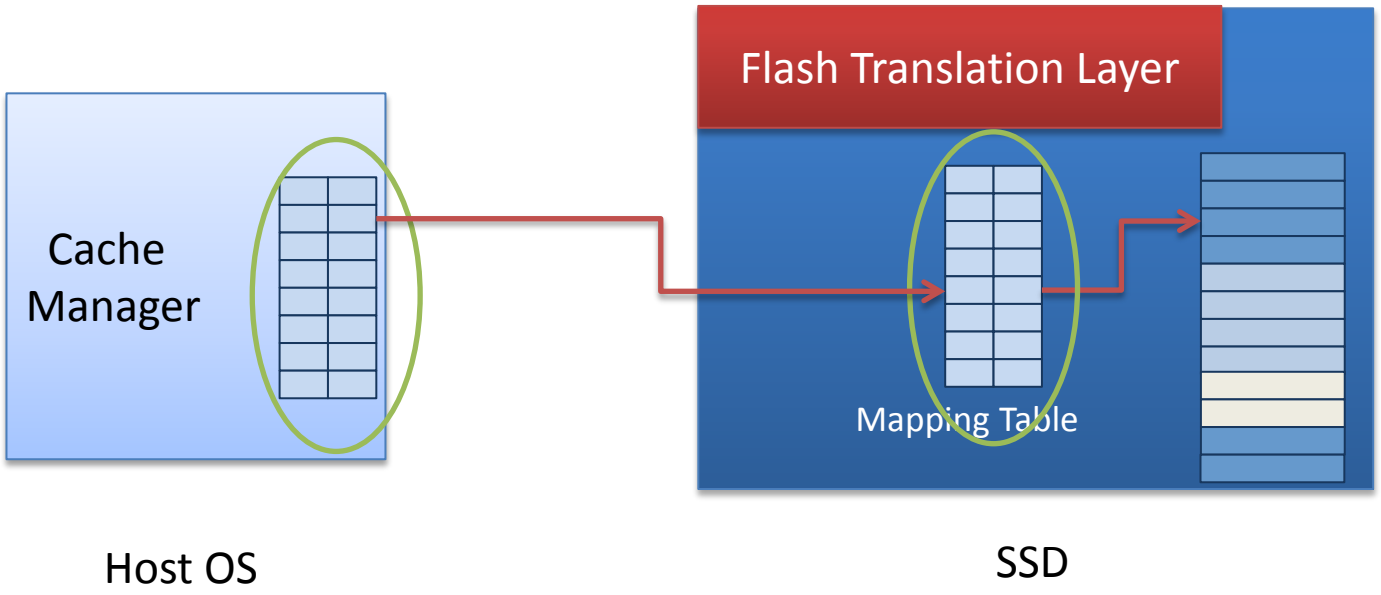
FlashTier Design Solutions

- Solid State Caches (SSCs)
  - **S1.** Unified Address Space
  - **S2.** Cache-aware free space management
  - **S3.** Caching interface and consistency
- Cache Manager
  - Write-back/write-through policies



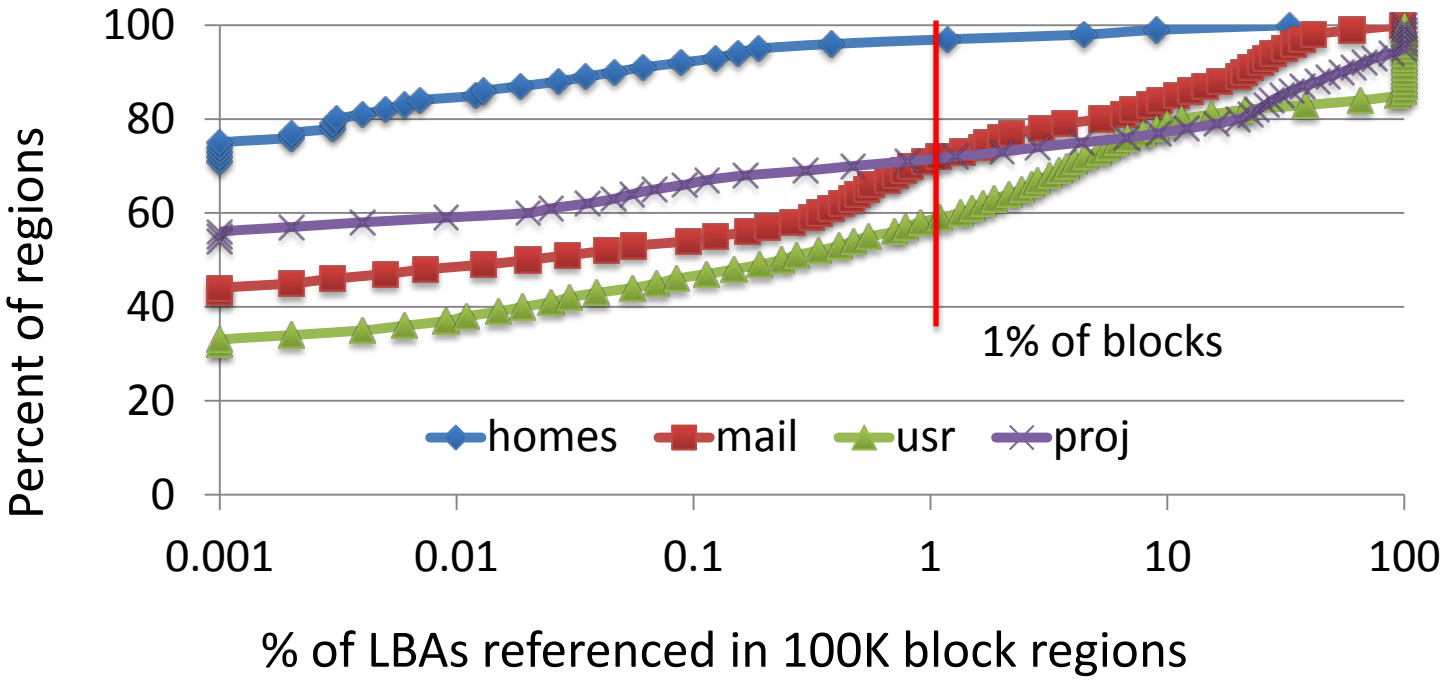
P1. Address Space Indirections

- Cache manager: Disk LBA→SSD LBA
- SSD FTL: SSD LBA→SSD PBA



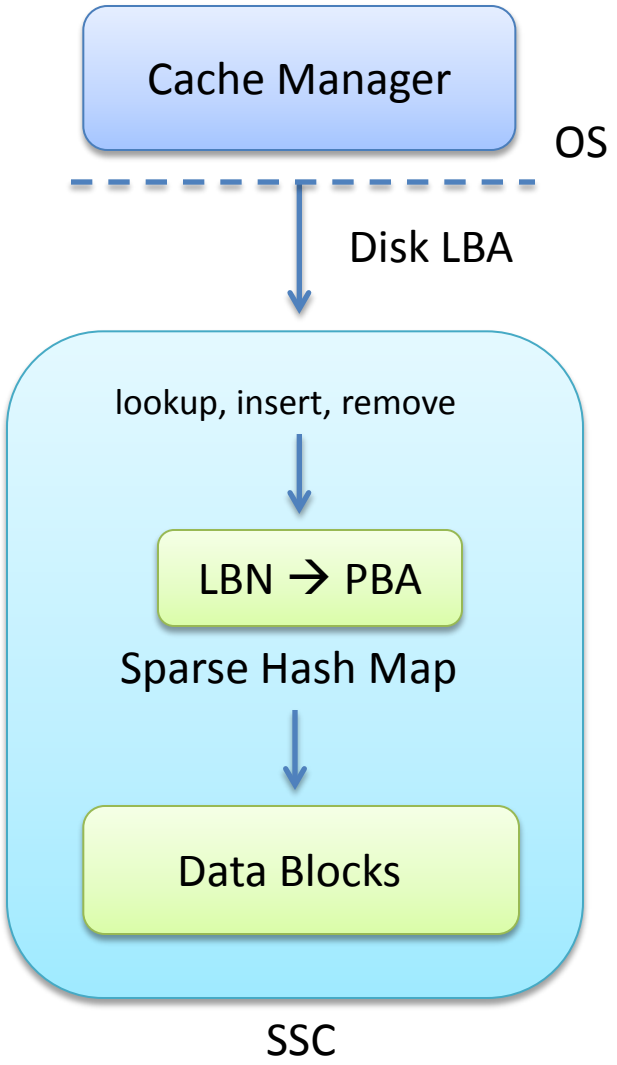
S1. SSC Unified Address Space

- Single Indirection: Disk LBA → SSD PBA
  - Cached addresses are sparse
  - ➢ Linear structures wasteful



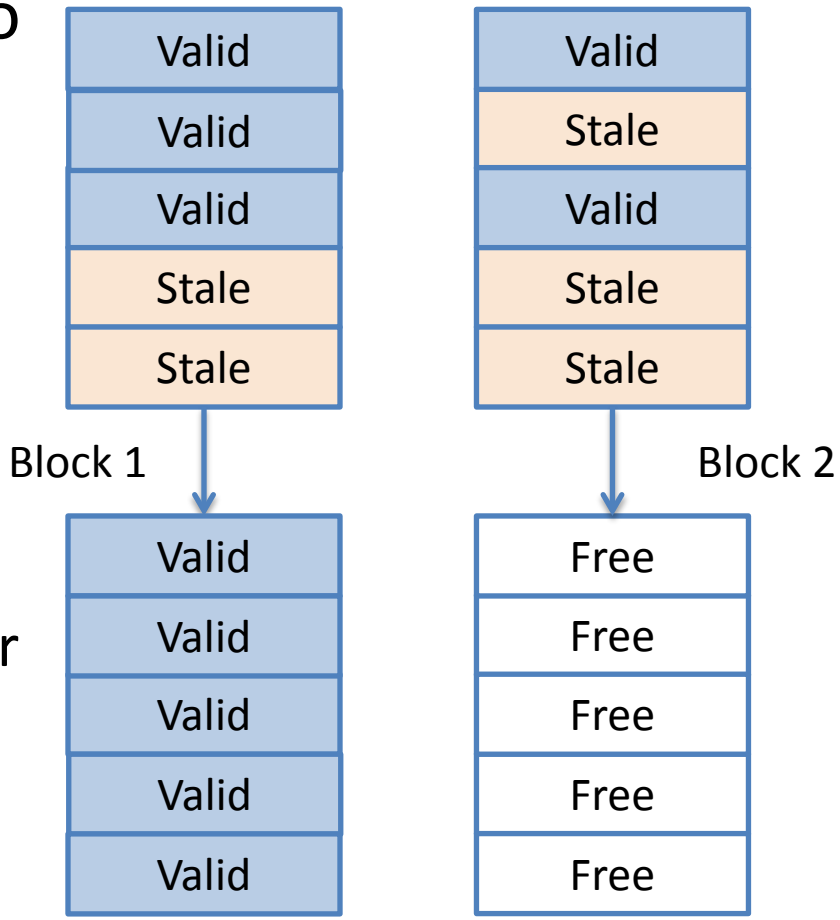
S1. SSC Address Mapping

- Sparse hash map
  - Developed at Google
  - 8.4 bytes/key
  - Hybrid Address Mapping
    - Data in 256KB erase blocks
    - Log in 4KB pages
- Status data
  - Clean/dirty bit per *page*



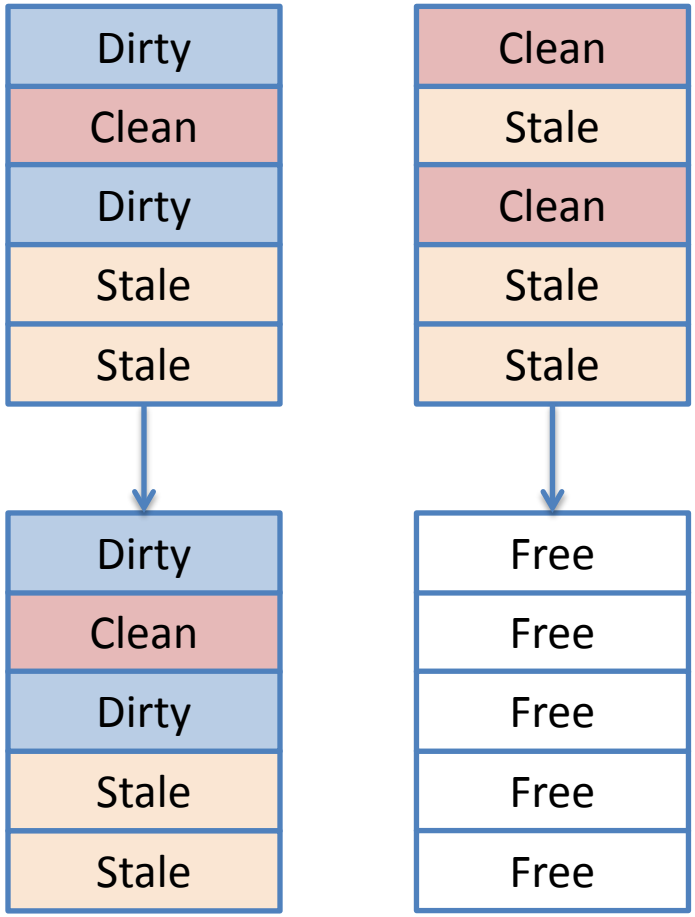
P2. Free Space Management

- Garbage collection leads to additional writes for data copy
  - Low write performance and endurance
- Full devices behave worse
  - Up to **83%** lower write performance and **80%** lower endurance [Intel IDF '10]
- **Caches are often full**



S2. Cache-Aware Space Management

- **Silent eviction** drops clean data rather than copying
- **Division of Responsibility**
  - Cache manager in OS identifies cold and clean data
  - Device silently evicts least-utilized cold clean data

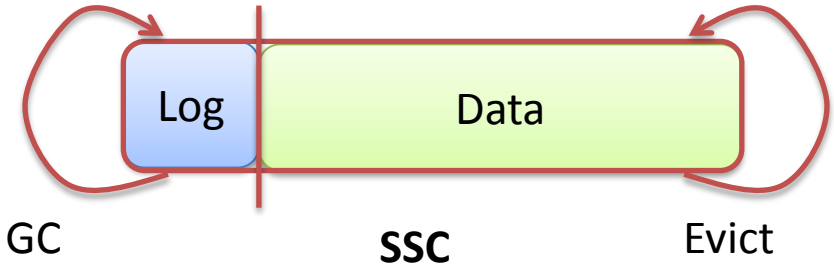


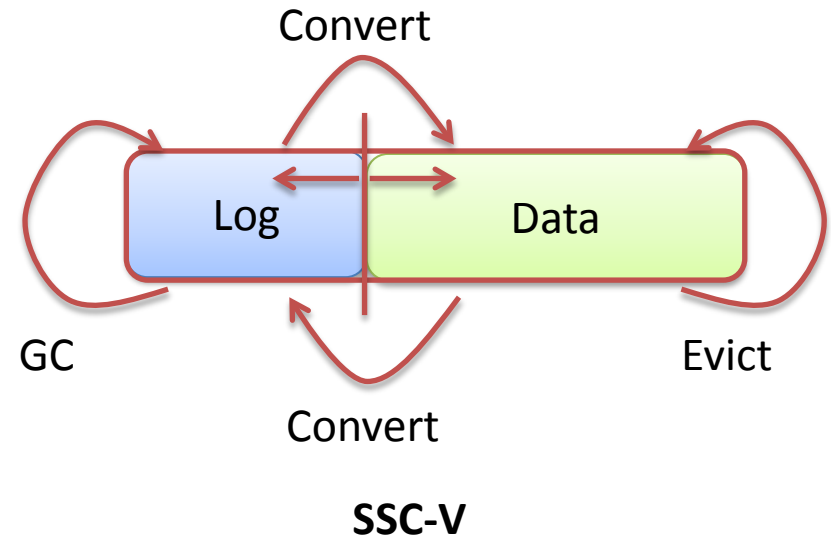
No data copy

S2. Silent Eviction Policies

- Evict and Convert

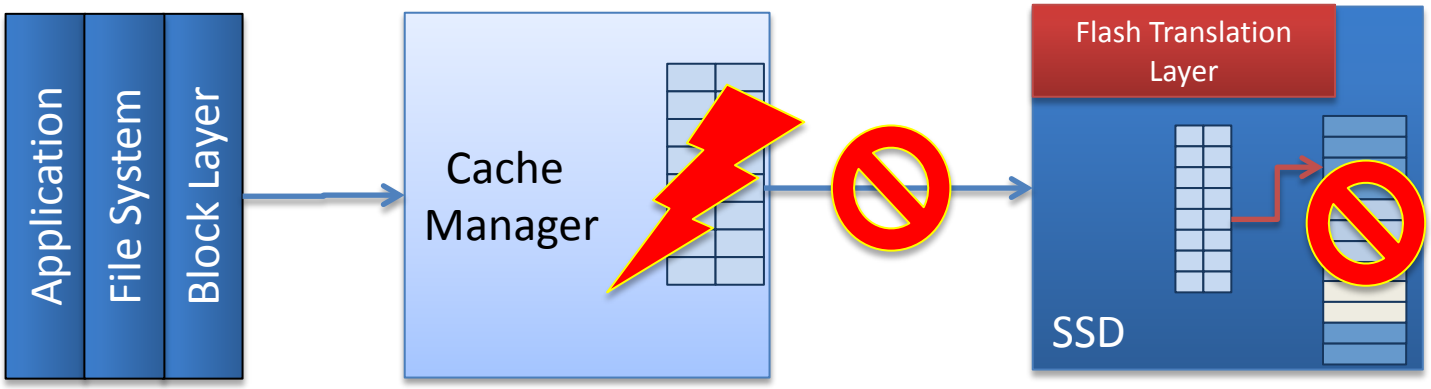
- **SSC**: fixed log space
  - Evict data blocks
  - Convert into data block
- **SSC-V**: variable log space
  - Evict data blocks
  - Convert into log block
  - Convert full log blocks to data blocks
- Performance vs. device memory for log blocks





P3. (In)consistency

- Without durability, warming a large SSD cache can take a long time
  - Cache manager must save **mapping** to survive crashes
  - Without consistency, data can be stale



S3. SSC Caching Interface

- Cache Management: clean/dirty pages separately
- Mapping Consistency: evict/clean operations

Command	Purpose
write-dirty	Insert new block or update existing block with **dirty data**.
write-clean	Insert new block or update existing block with **clean data**.
read	Read block if present **or return zeros**
evict	Invalidate block immediately
clean	Allow **future eviction** of block
exists	Test for presence of dirty blocks

S3. Crash Consistency Guarantees

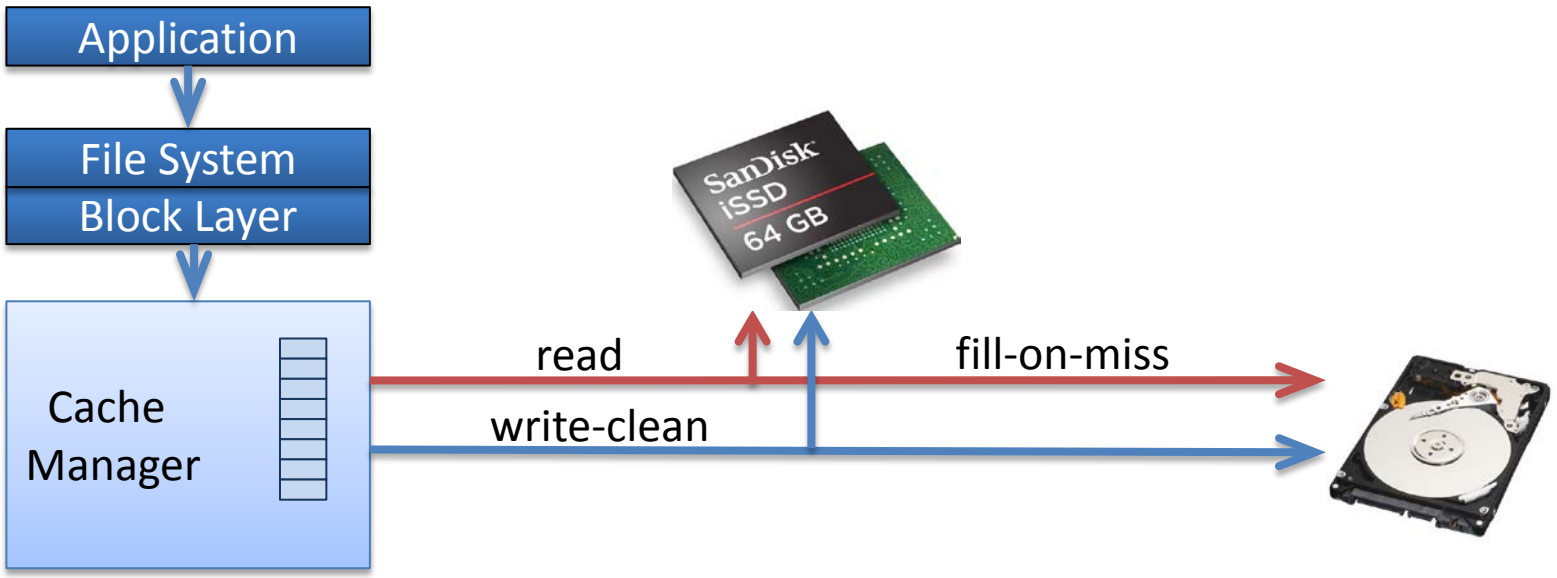
- Always safe to consult cache after crash
  - Never lose dirty data

1	A **read** following a **write** of **dirty** data will return that data.

  - Never return stale data

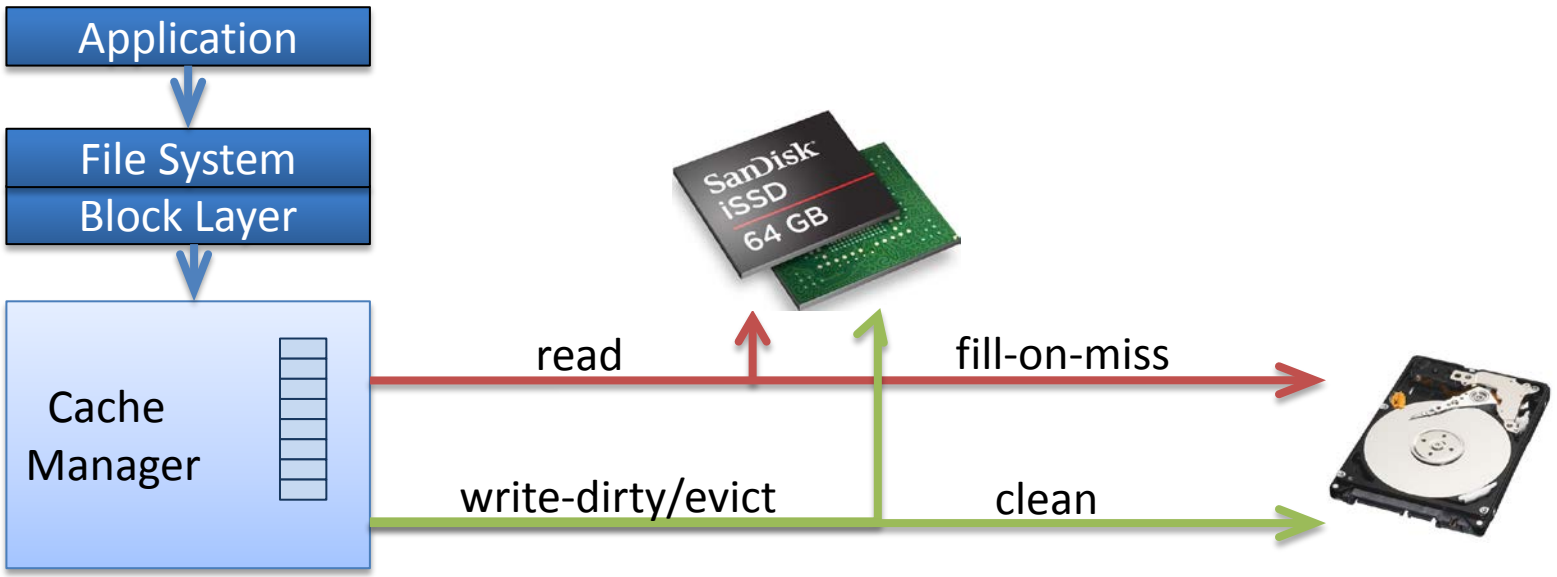
2	A **read** following a **write** of **clean** data will return *either* that data or zeros.
3	A **read** following an **eviction** will return zeros.

4. Cache Manager



- Write through: no per-block state
  - Access with **read**
  - Write with **write-clean**
  - Fill on miss with **write-clean**

4. Cache Manager



- Write back: dirty block state only
  - Access/fill same as write-through
  - Write with **write-dirty**
  - Above fixed % dirty mark, with **clean** on write-back
  - Recover dirty blocks with **exists**

Outline

- Introduction
- Motivation
- FlashTier Design
- Evaluation
  - Does it perform well?
  - Does it improve reliability?
  - Does it save memory?
- Conclusion

Implementation

- Modified Facebook FlashCache cache manager
- Modified FlashSim flash timing simulator [Kim et. al]
  - Trace-based simulation
  - SSD, SSC and SSC-V device models
  - Silent eviction: for clean data only

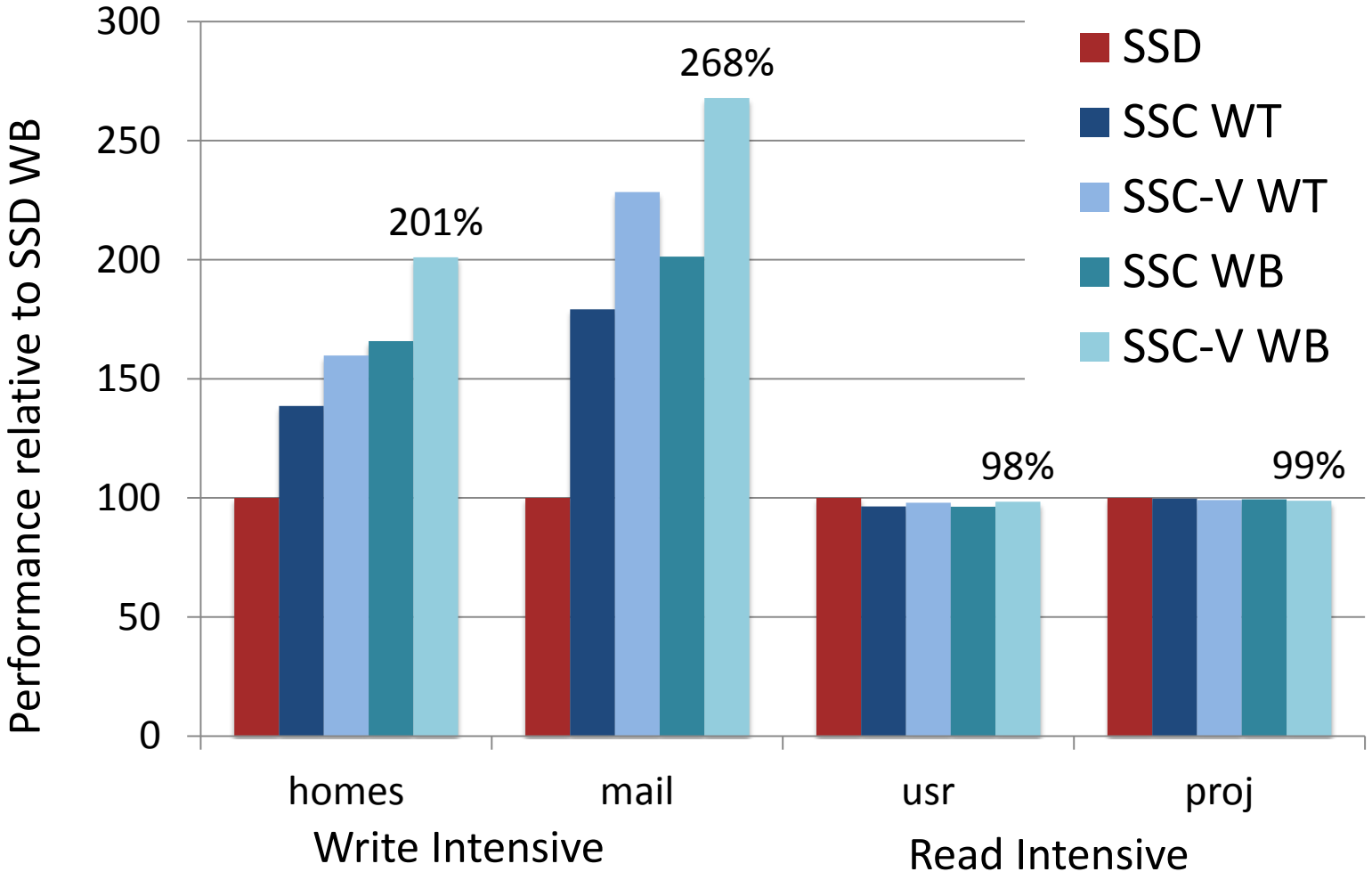
Model	Configuration
SSD/SSC	7% fixed log space
SSC-V	0-20% variable log space, more page-level mappings
Device Parameters	Intel 300 series SSD
Cache Manager	20% dirty mark for write-back operation

Methodology

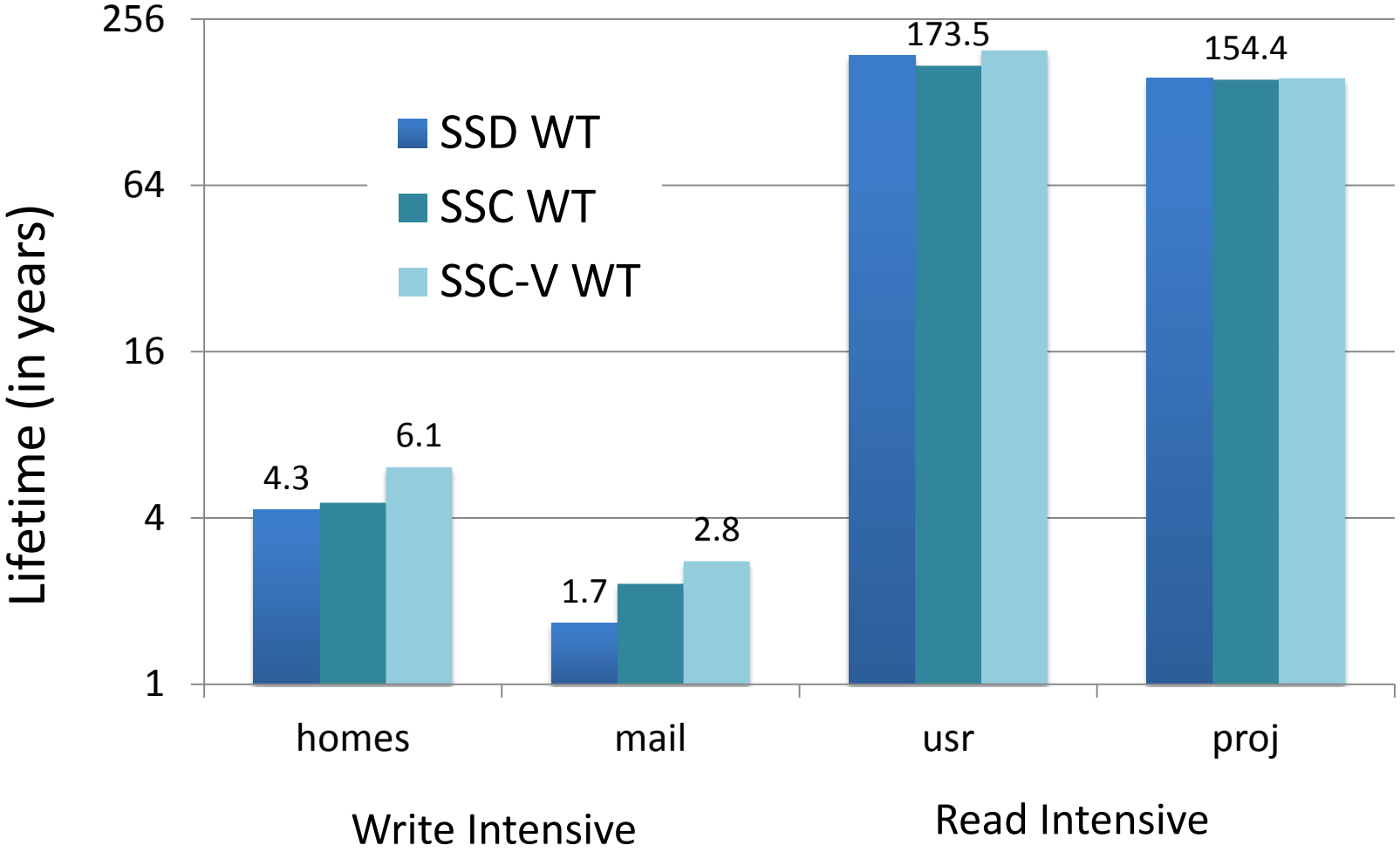
- Systems for comparison
  - Facebook FlashCache using SSD with GC
  - FlashTier using SSC and SSC-V with silent eviction
- Workload: production server traces [FAST '08, FAST '10]

Trace Name	Unique blocks	Percent writes
homes	1,684,407	96%
mail	15,136,141	88%
usr	99,450,142	6%
proj	107,509,907	14%

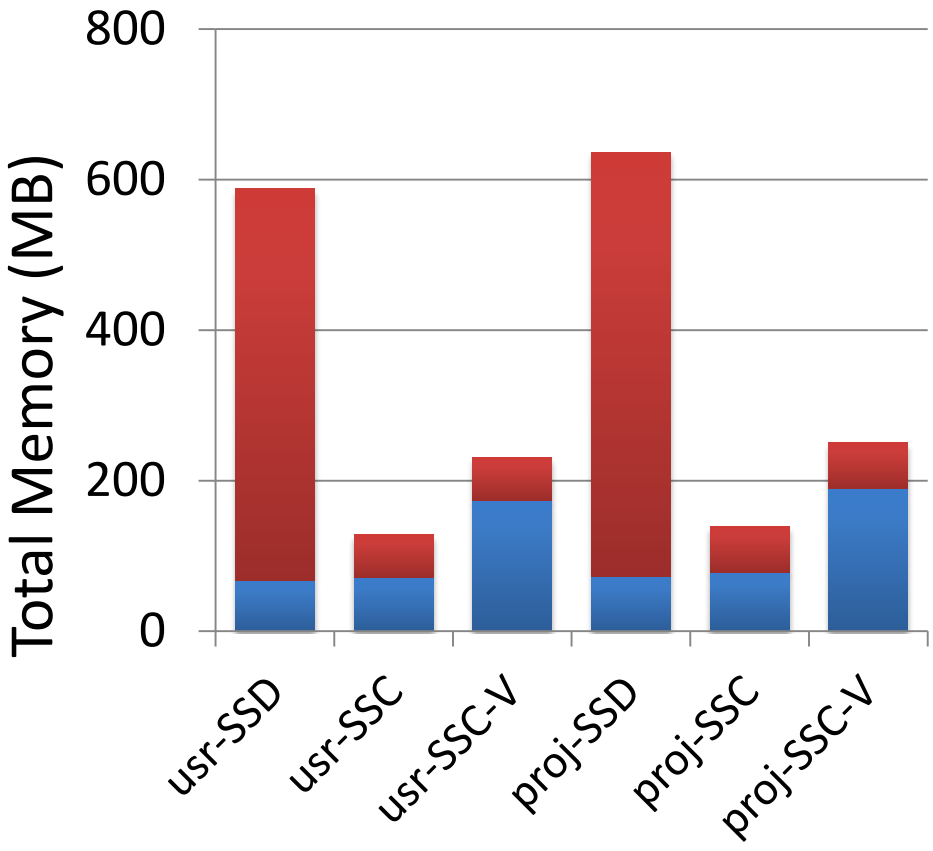
Performance



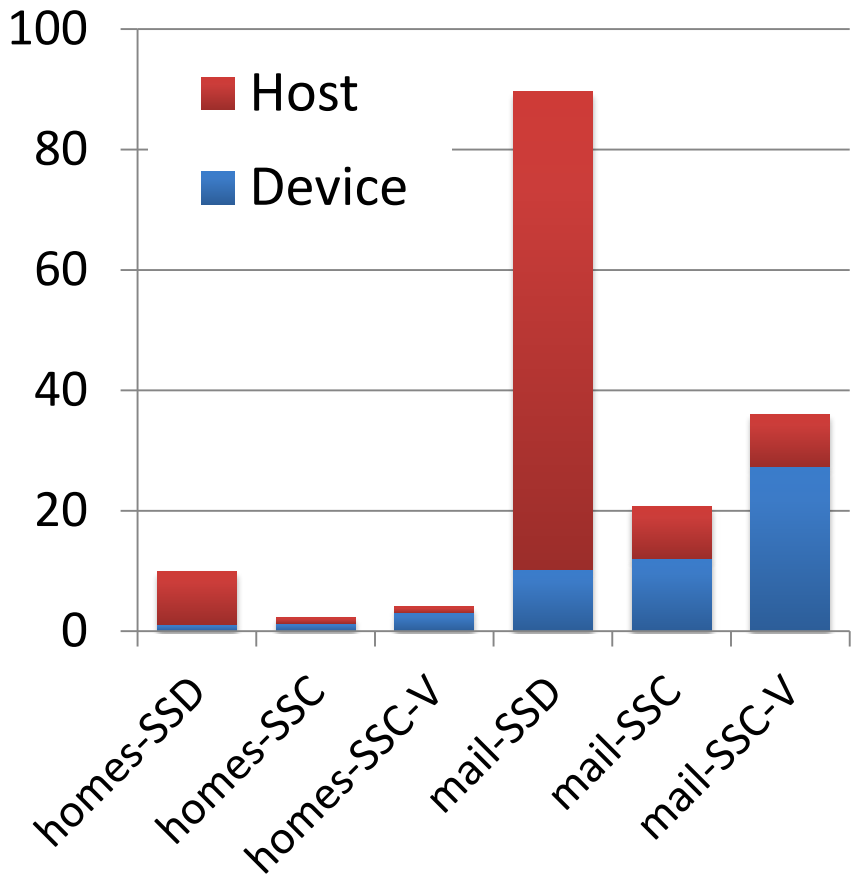
Endurance


256
64
16
4
1
Lifetime (in years)
SSD WT
SSC WT
SSC-V WT
4.3
6.1
1.7
2.8
173.5
154.4
homes
mail
usr
proj
Write Intensive
Read Intensive

Memory Usage (Write Back)



Write Intensive



Read Intensive

Summary

- FlashTier:
  - Simplifies cache management
  - Reduces memory consumption on all workloads
  - Improves performance and cache lifetime on write-heavy workloads
  - Decreases cost of crash consistency

Thanks!

FlashTier: A Lightweight, Consistent and Durable Storage Cache

Mohit Saxena
PhD Candidate
University of Wisconsin-Madison
msaxena@cs.wisc.edu

http://research.cs.wisc.edu/sonar/projects/FlashTier