

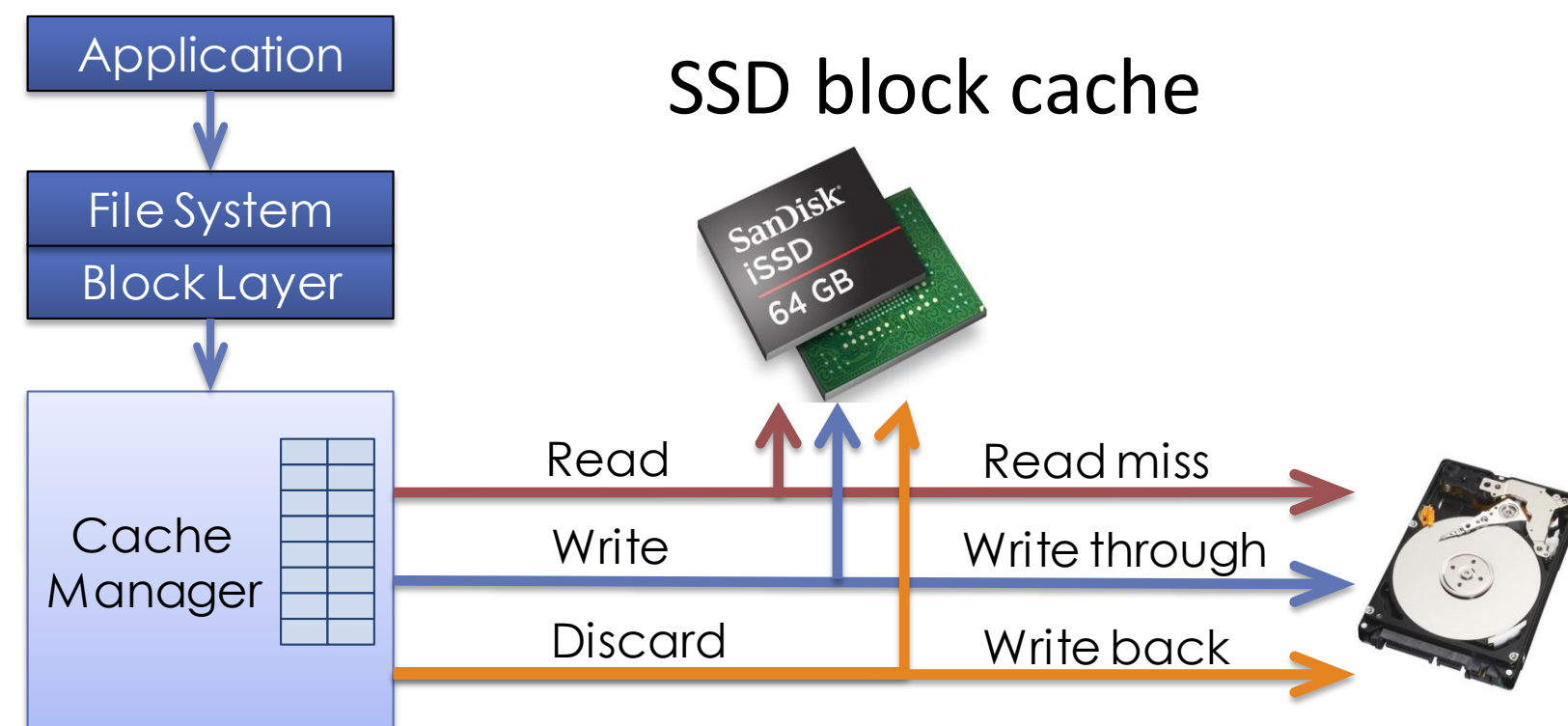
# FlashTier: A Lightweight, Consistent and Durable Storage Cache

Mohit Saxena, Michael M. Swift and Yiying Zhang

## Introduction

High-speed solid-state drives (SSDs) composed of NAND flash are often deployed as a cache in front of high-capacity disk storage. Several vendors adopt SSDs as a block cache transparent to filesystems:

- OS Vendors: Oracle, Microsoft, Linux
- Storage Vendors: Intel, OCZ, NetApp, EMC, FusionIO
- Applications: Facebook, Google



## Problem Statement

### Caching is different from Storage

An SSD block cache is hindered by the narrow block interface and internal block management of SSDs designed to serve as a disk replacement for persistent storage.

Inefficiencies with SSD block cache	
<b>Address Space Management</b>	Two levels of indirections for address translation
<b>Consistency and Durability</b>	Several hours to days for warm-up after reboot/crash
<b>Free Space Management</b>	Low write performance and endurance due to garbage collection for caching

## Design Overview

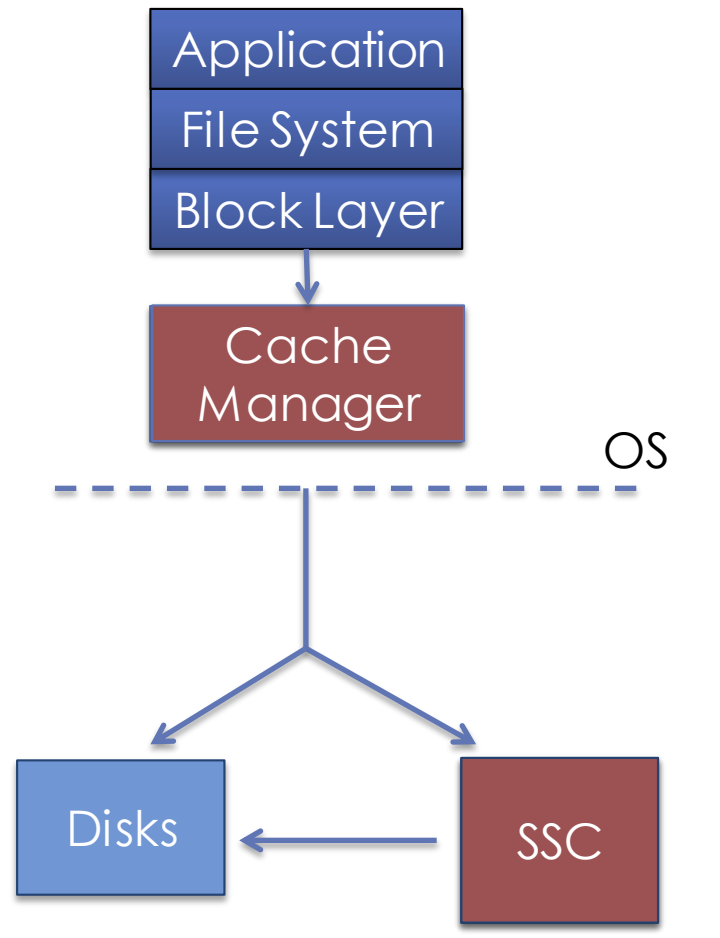
The FlashTier system consists of two components: a *solid-state cache (SSC)*, a flash device designed specifically for caching, and a *cache manager* within the OS for migrating data between the flash caching tier and disk storage.

**SSC:** The SSC matches the requirements of caching behavior. It provides:

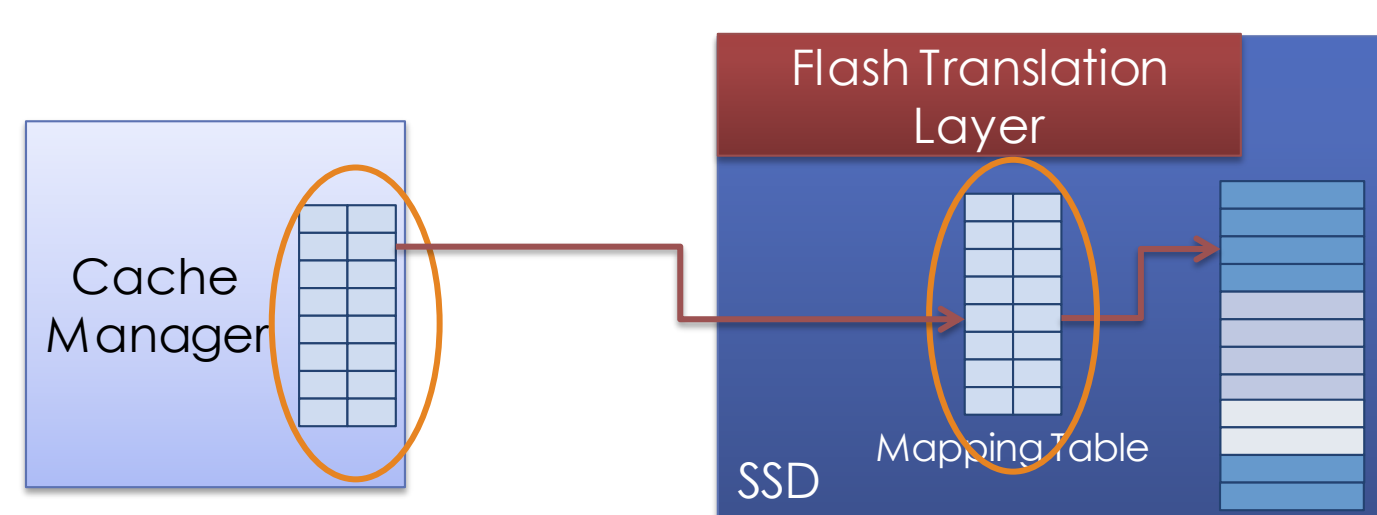
- Unified cache address space
- Consistent cache interface
- Cache-aware free space management

### Cache Manager in OS:

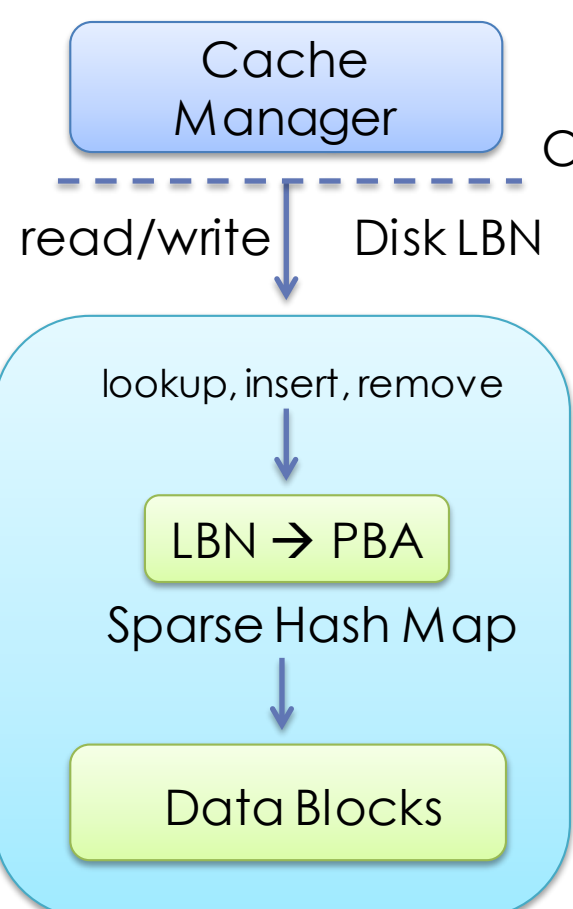
- Adapts to the new interface of the SSC
- Operation Modes: write-back for higher performance or write-through for higher safety



## Unified Address Space



**Two levels of indirection:** On an SSD cache, the cache manager maintains a mapping table within the host DRAM translating disk addresses to SSD addresses. The SSD block cache keeps another mapping (FTL) within the device memory translating SSD logical addresses to flash addresses to avoid in-place writes. These two mappings pose memory overhead for host DRAM and device memory.



### FlashTier Unified Address Space

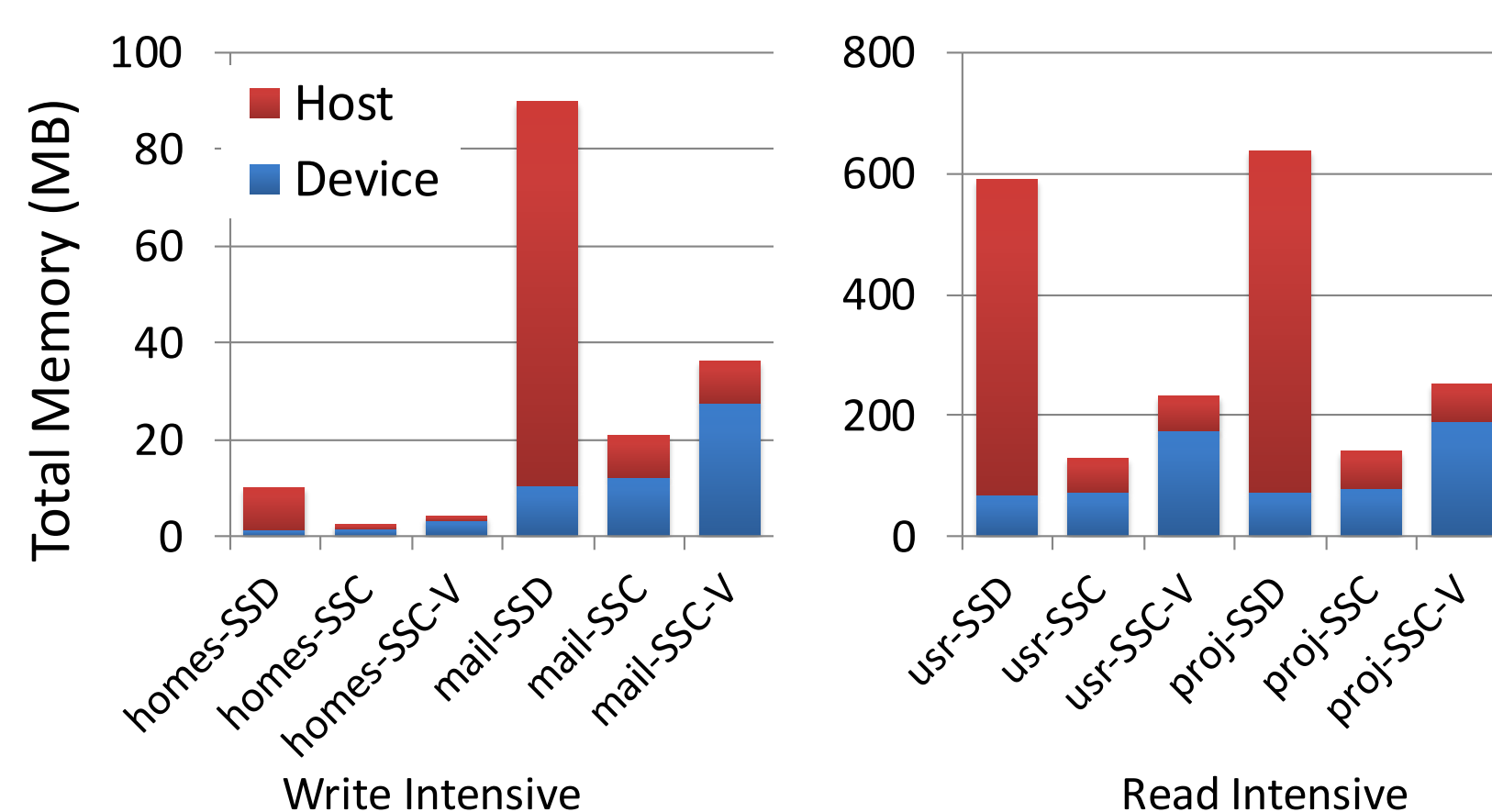
- The cache manager directly addresses flash blocks in the SSC by disk LBN.
- The SSC only stores the hot data out of the terabytes of disk storage. As a result, the SSC optimizes for sparse address space using a Sparse Hash Map [3] requiring only 8.4 bytes/key to keep low device memory footprint.

### Host and Device Memory Usage

The graph shows the host and device memory usage for SSD cache and FlashTier with SSC and SSC-V.

- The SSD cache manager stores state for all cached blocks.
- The FlashTier cache manager only maintains a dirty block table to clean/flush dirty LRU blocks to disk on exceeding the dirty mark.
- SSC-V stores more page-level mappings for variable log space and improved performance.

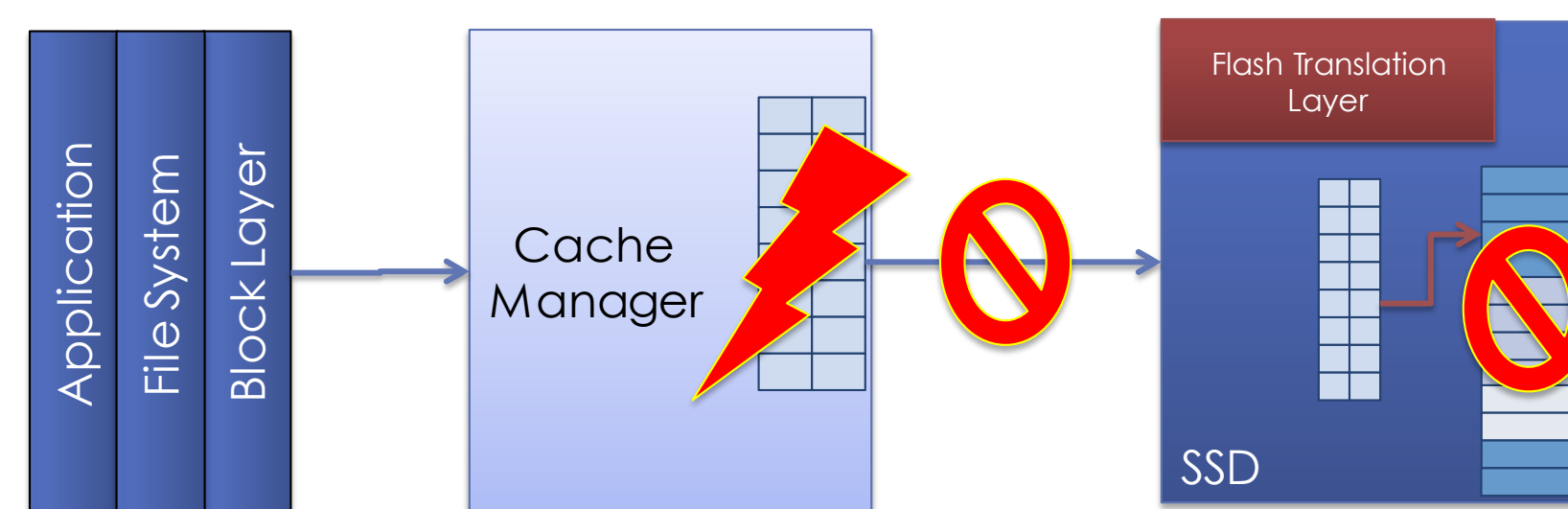
FlashTier reduces memory usage for host and device combined by 78% for SSC and 60% for SSC-V relative to SSD block cache.



## References

- [1] N. Agrawal, V. Prabhakaran, T. Wobber, J. Davis, M. Manasse, and R. Panigrahy. Design Tradeoffs for SSD performance, In *Usenix ATC* 2008.
- [2] Facebook FlashCache, <https://github.com/facebook/flashcache>.
- [3] Google Inc, Sparse Hash Map, <http://goog-sparsehash.sourceforge.net>
- [4] M. Saxena and M. Swift, FlashVM: Virtual Memory Management on Flash, In *Usenix ATC* 2010.

## Consistent Cache Interface



**Inconsistency:** On an SSD cache, maintaining cached data consistent and durable requires the cache manager to persist the cache block state and mapping within the host DRAM. To avoid the runtime cost of persisting the mapping, most cache managers do not provide any consistency or durability.

- Without **durability**, filling a 100GB cache from a 500 IOPS disk system can take over 14 hours after a reboot.
  - Without **consistency**, cache update and invalidate operations can result in a stale copy of cached data to be read later.
- FlashTier provides an SSC interface to keep the mapping consistent and durable.

### Consistent SSC Interface

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

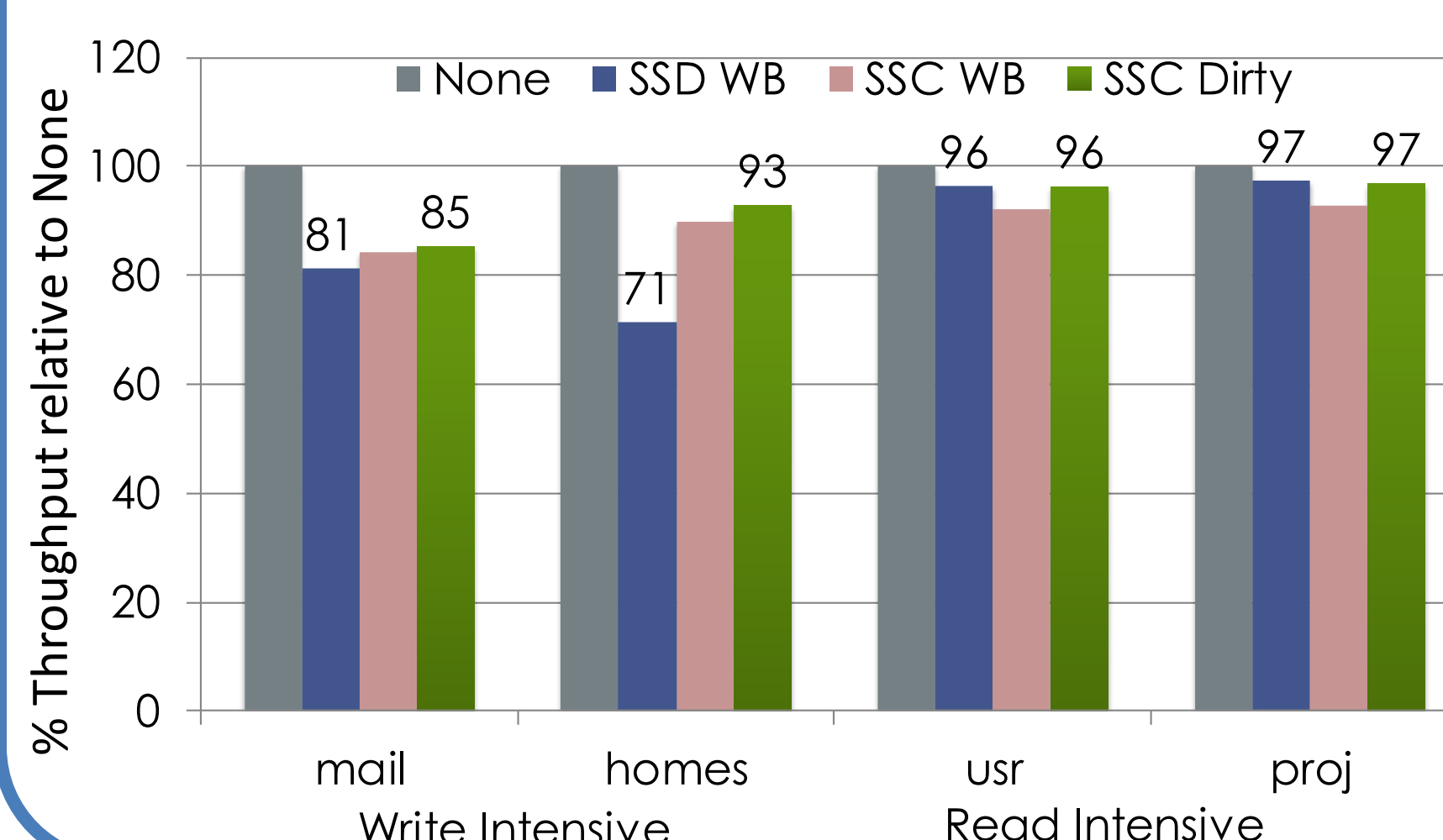
**Crash Guarantees:** An SSC never returns stale data or loses dirty data. This is guaranteed by consistent reads following a cache write to dirty/clean data and cache eviction. It is always safe for the cache manager to consult the SSC after a crash.

### Cost of Crash Consistency

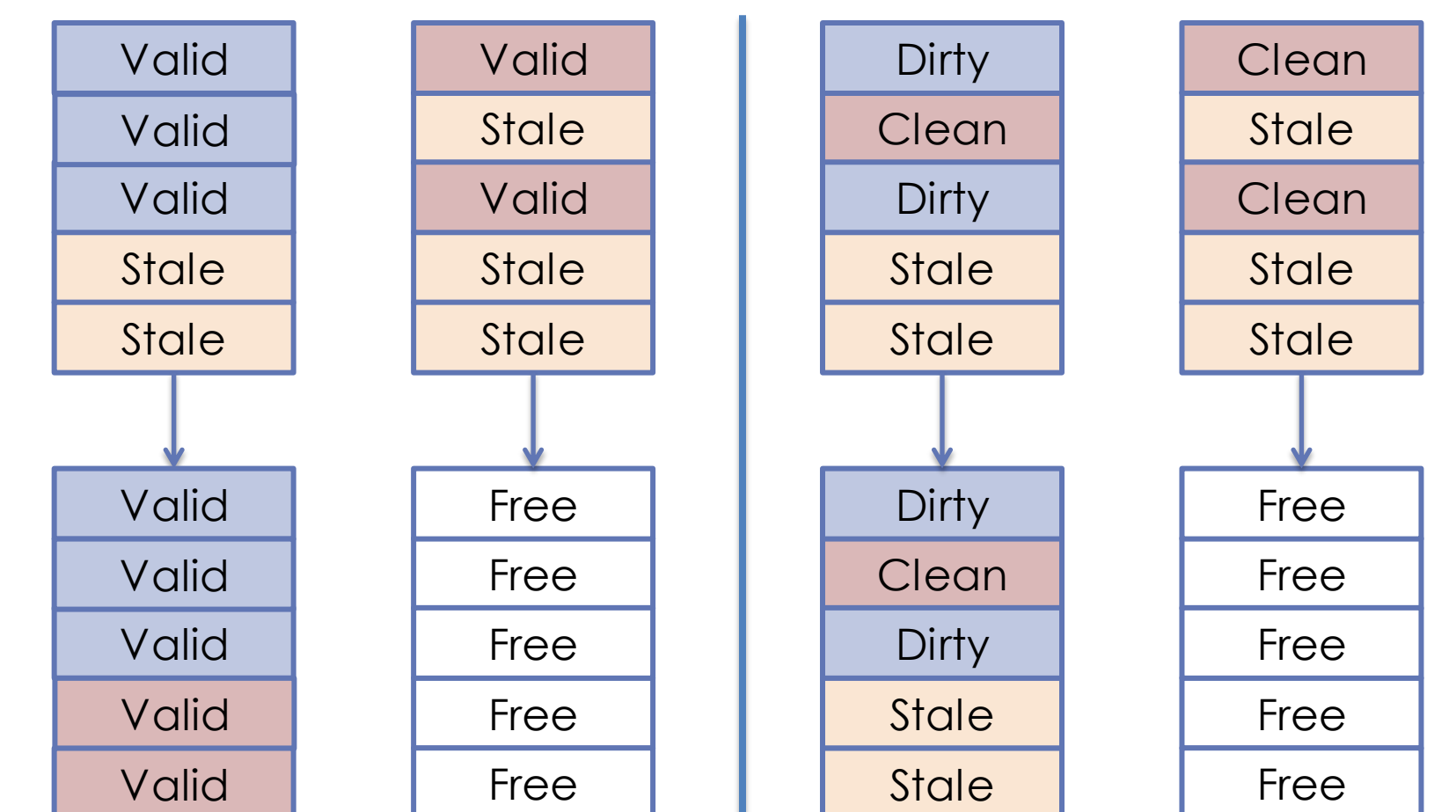
The graph shows the relative cost of consistency for SSD block cache persisting dirty pages, SSC persisting both clean and dirty pages, and SSC Dirty persisting only dirty pages.

The SSD block cache manager uses synchronous metadata updates, and the SSC uses internal logging and checkpointing mechanisms for persistent mapping.

FlashTier decreases the cost of crash consistency relative to the SSD block cache. It has less than 16% overhead for all workloads.



## Free Space Management



SSD: 5 reads/writes

SSC: No reads/writes

**Garbage collection:** On an SSD cache, garbage collection result in additional copies for valid pages to create free erased blocks for new writes.

- Full devices lacking free blocks behave worse with up to 83% lower performance and 80% lower write endurance
- **Caches are often full.**

**Silent Eviction:** An SSC employs silent eviction to lose *clean* data rather than copying it. It uses a cost/benefit mechanism to select blocks for eviction:

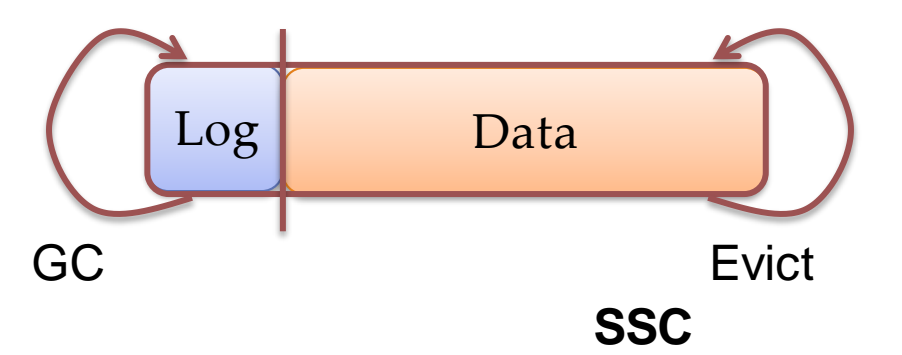
- Usage benefit: The cache manager uses evict/clean to identify cold clean data.
- Eviction cost: The SSC silently evicts least-utilized cold clean data.

### Policies for Silent Eviction

The SSC uses two different policies for silently evicting a data block, which trade between cache performance and device memory for storing page-level mappings of log blocks:

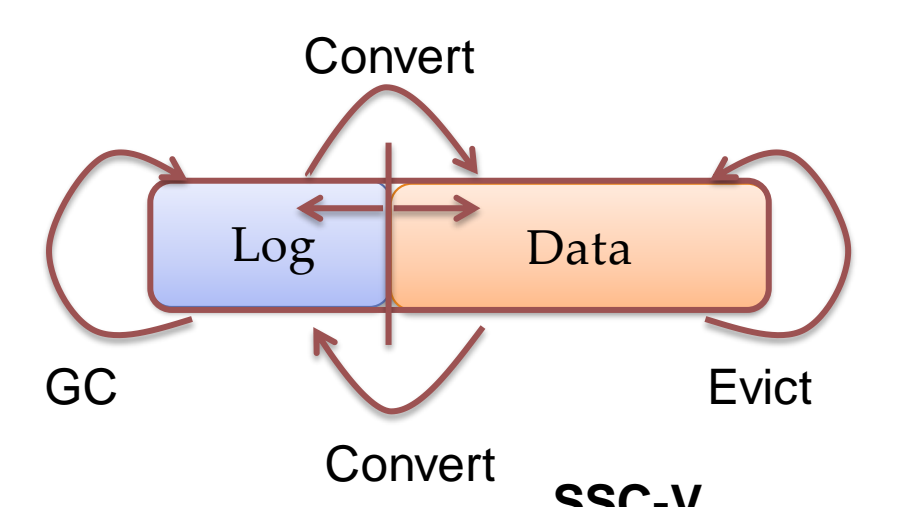
- **SSC with fixed log space:**

Evicted data blocks are recycled for use as data GC blocks.



- **SSC-V with variable log space:**

Evicted data blocks can be recycled for use as data blocks or increase the fraction of log blocks.



### System Performance

The graph shows the relative performance of SSC and SSC-V in write-back and write-through modes relative to SSD cache in write-back mode.

FlashTier improves cache performance for write-intensive workloads by 168% with SSC-V and 101% with SSC. For read-intensive workloads, FlashTier performs equally well as SSD block cache despite silent evictions.

