Extending SSD Lifetimes with Disk-Based Write Caches

Gokul Soundararajan
University of Toronto

Vijayan Prabhakaran
Mahesh Balakrishnan
Ted Wobber
Microsoft Research

Solid-State Devices (SSDs)

Solid-State Devices (SSDs)

- **Replacing Hard Disk Drives (HDDs)**
  - Fast I/O reads
  - Consume low power, no moving parts, and more reliable

Solid-State Devices (SSDs)

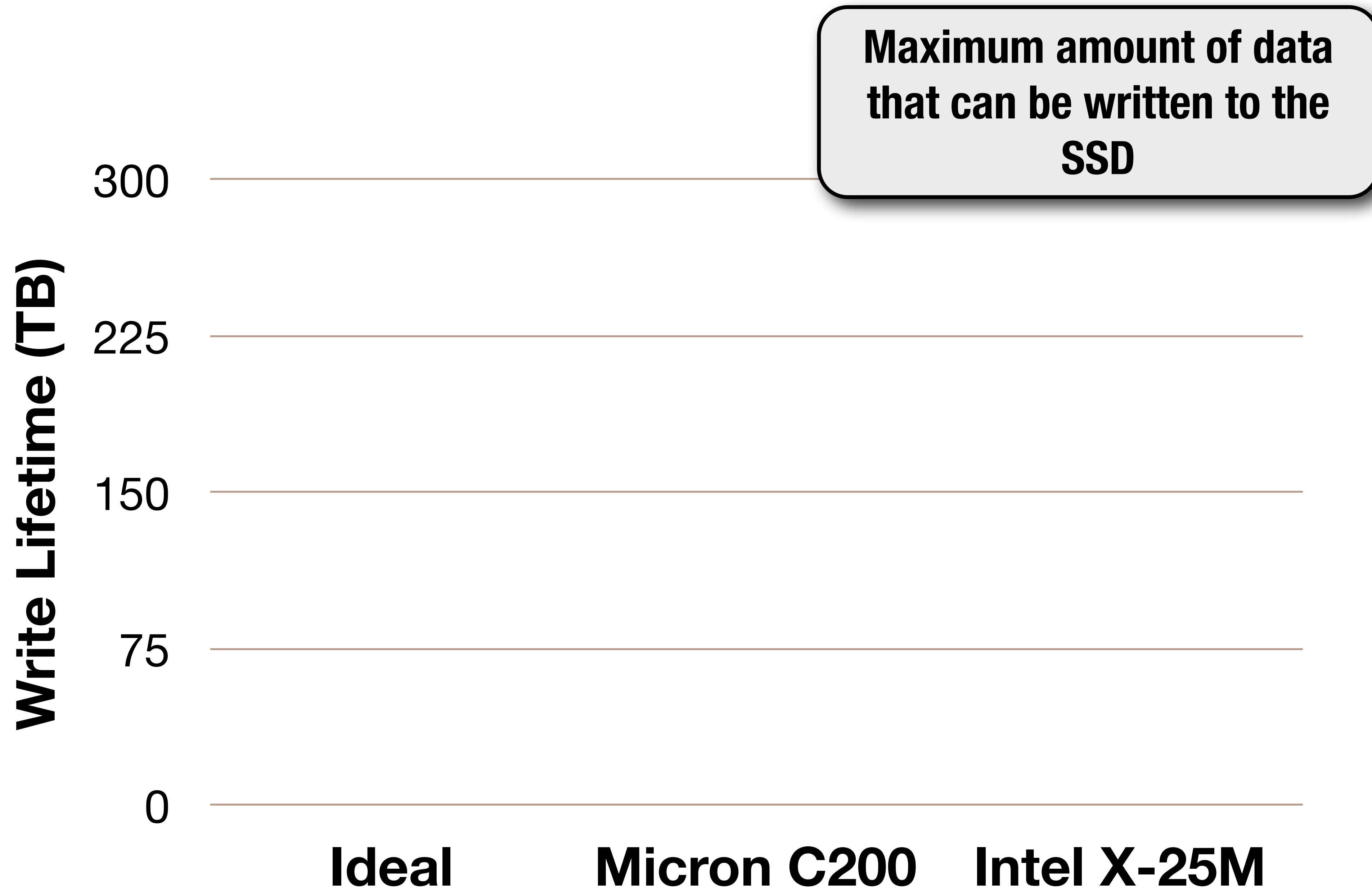
- **Replacing Hard Disk Drives (HDDs)**
  - Fast I/O reads
  - Consume low power, no moving parts, and more reliable
- **Limited write cycles**
  - Need to erase block before re-write
  - High end SLC provide 100,000 erase cycles
  - Mainstream MLC provides 5,000-10,000 erase cycles

Solid-State Devices (SSDs)

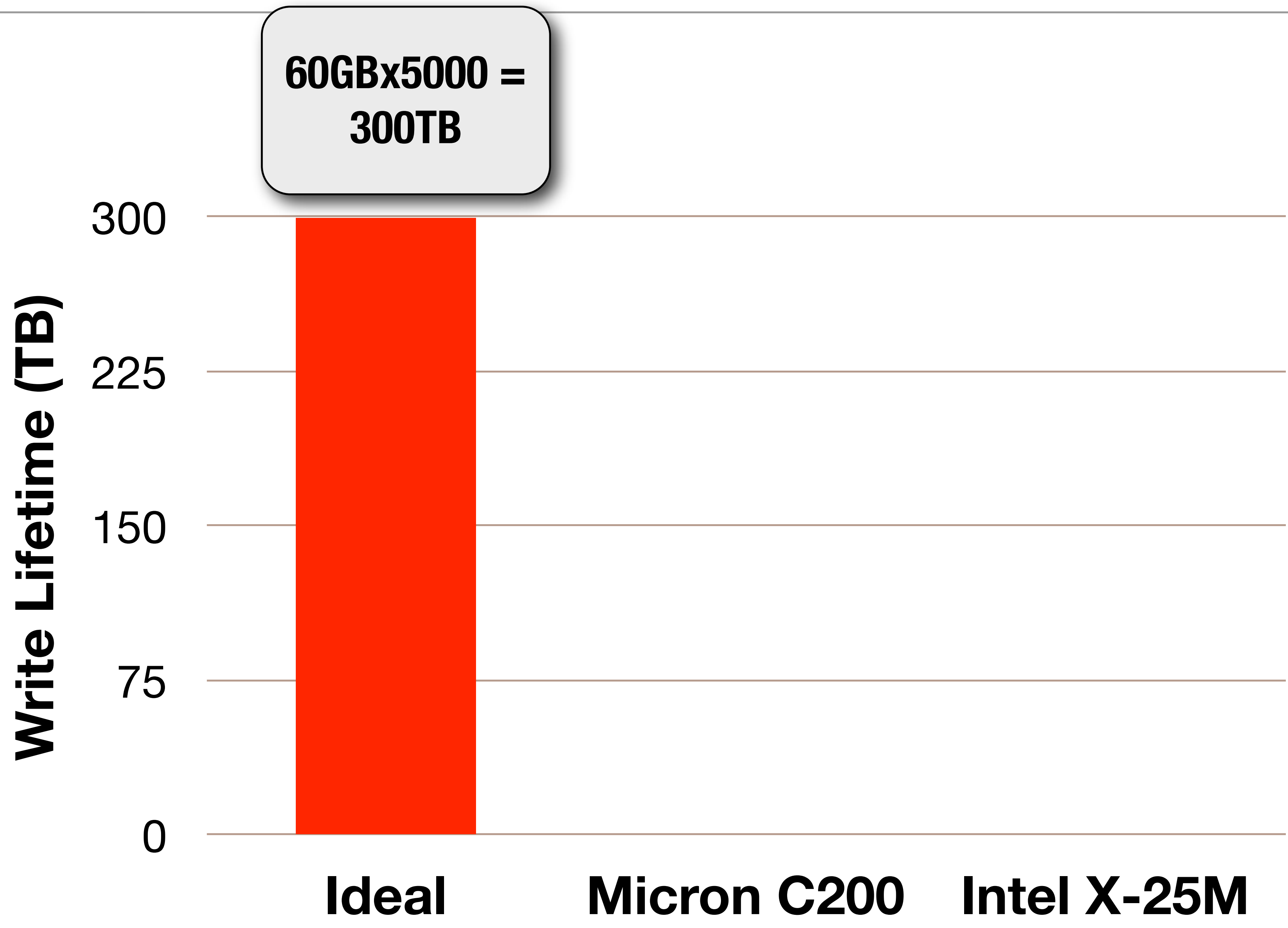
- **Replacing Hard Disk Drives (HDDs)**
  - Fast I/O reads
  - Consume low power, no moving parts, and more reliable
- **Limited write cycles**
  - Need to erase block before re-write
  - High end SLC provide 100,000 erase cycles
  - Mainstream MLC provides 5,000-10,000 erase cycles
- **Starting to be used in laptops/desktops**
  - Contain write intensive workloads

SSD Write-lifetime

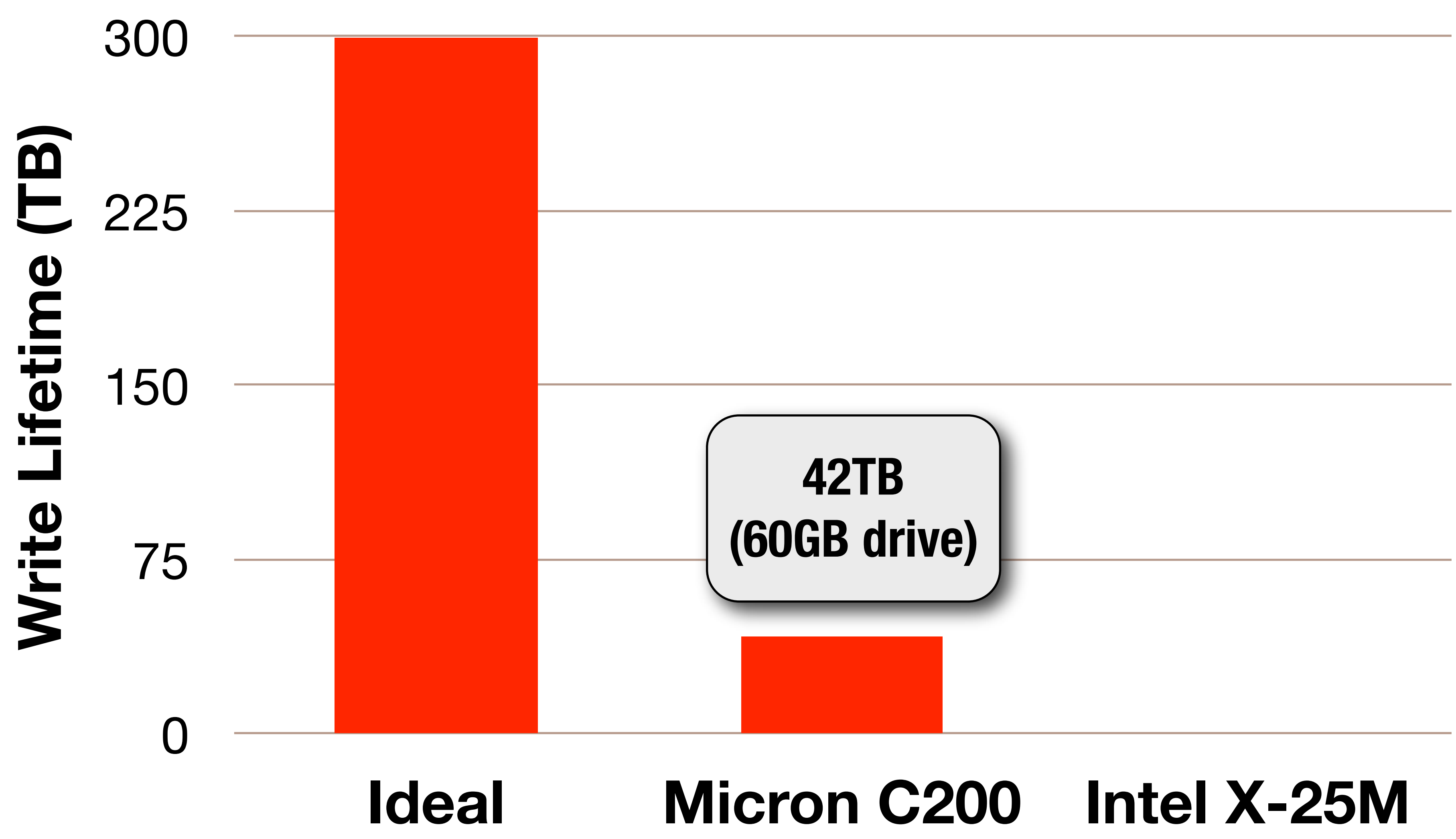
SSD Write-lifetime



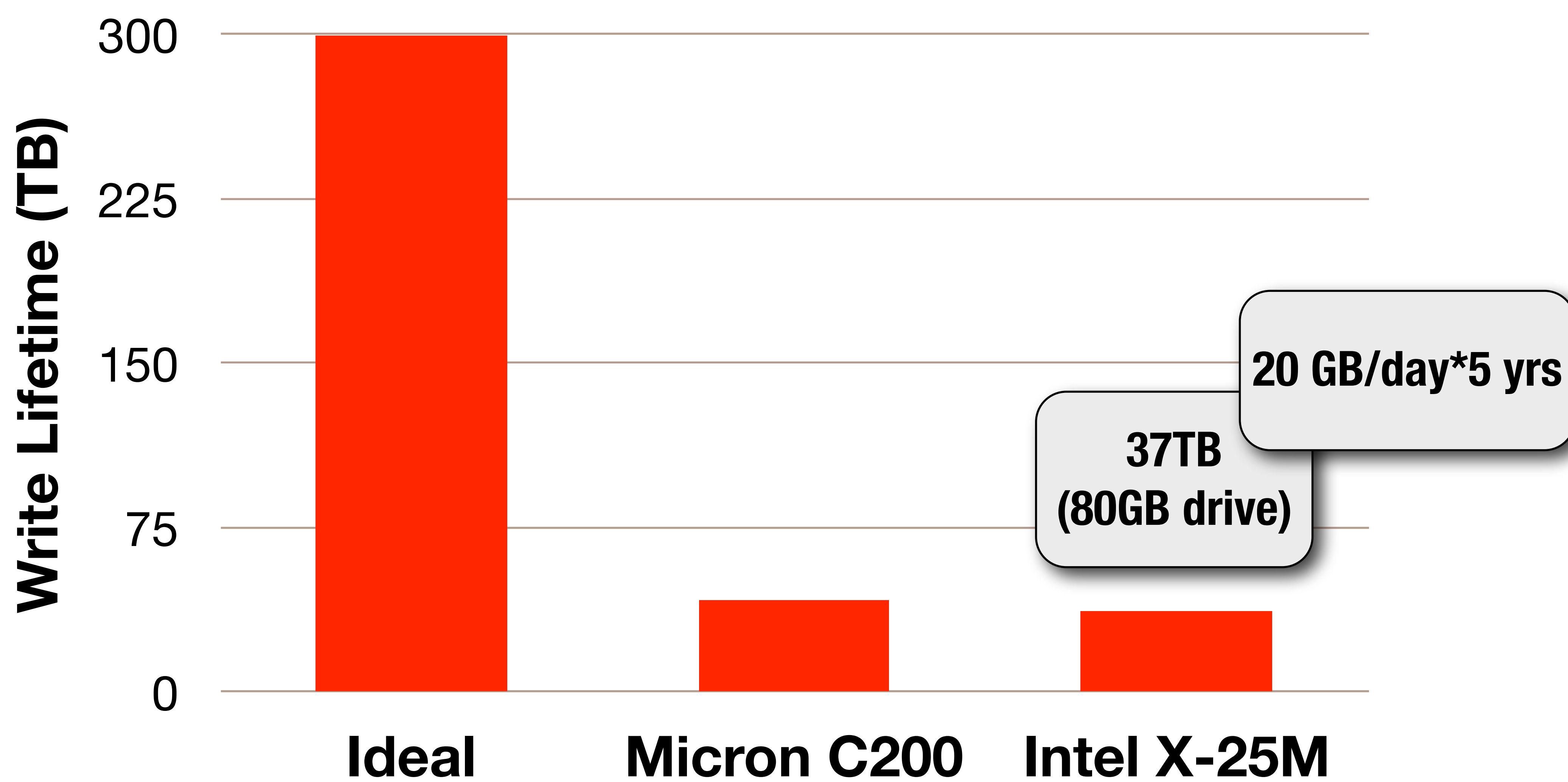
SSD Write-lifetime



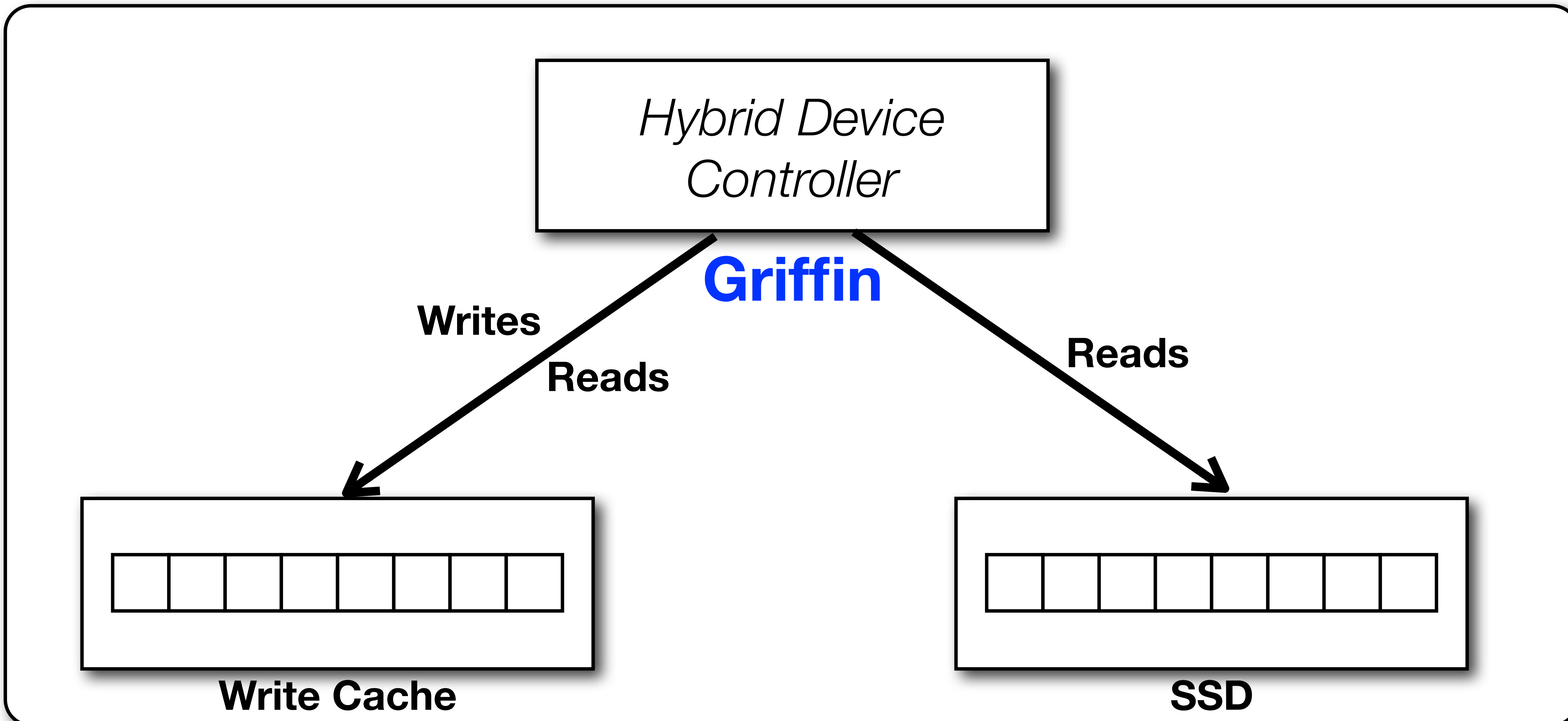
SSD Write-lifetime



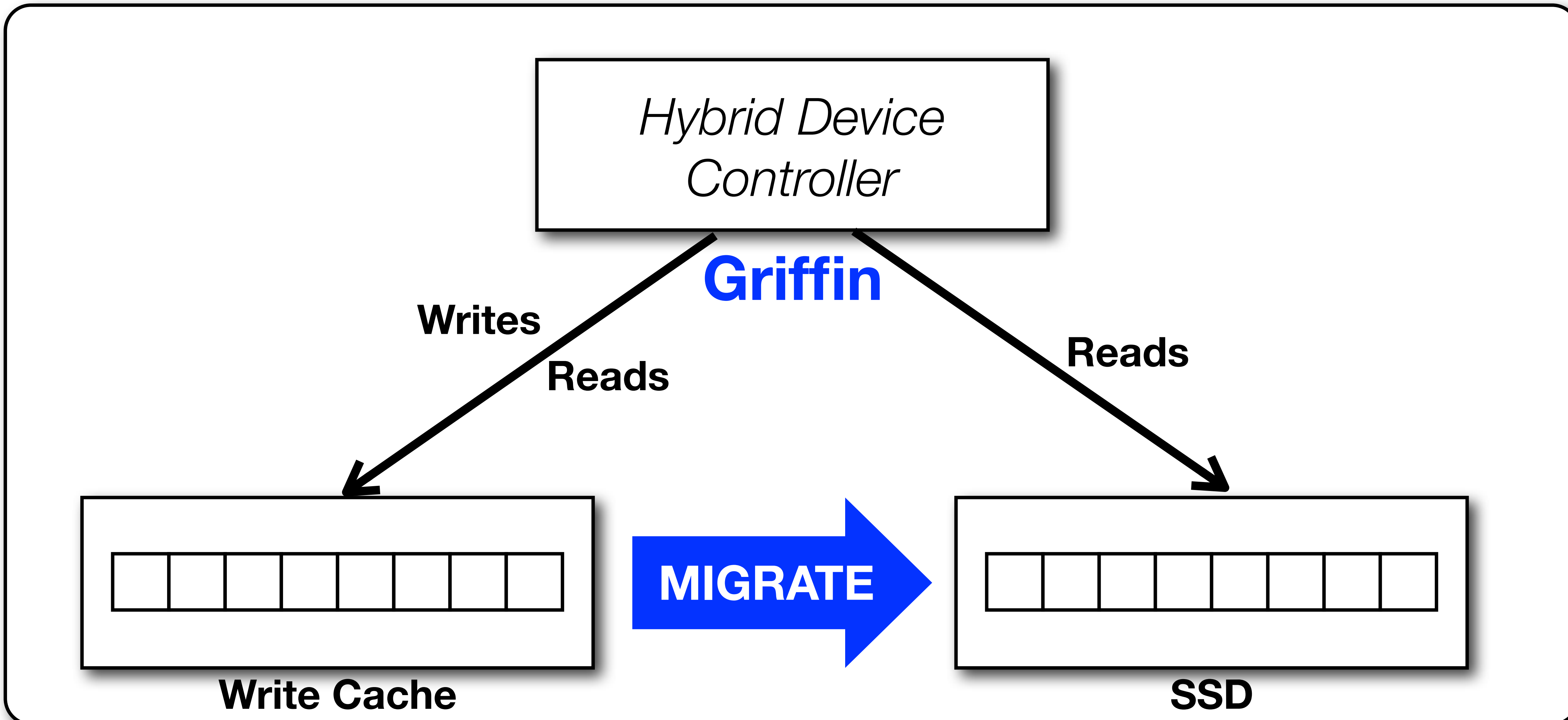
SSD Write-lifetime



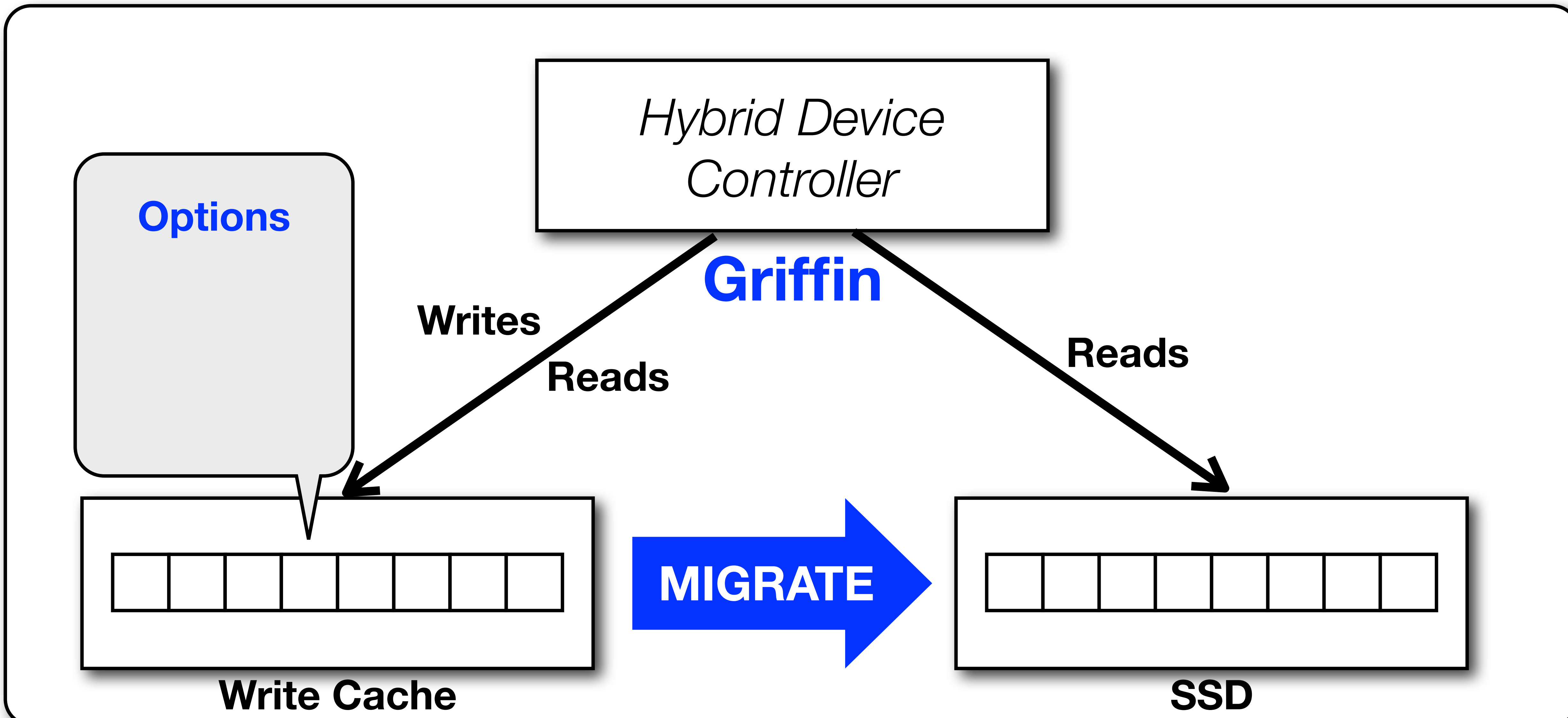
Griffin Hybrid Device



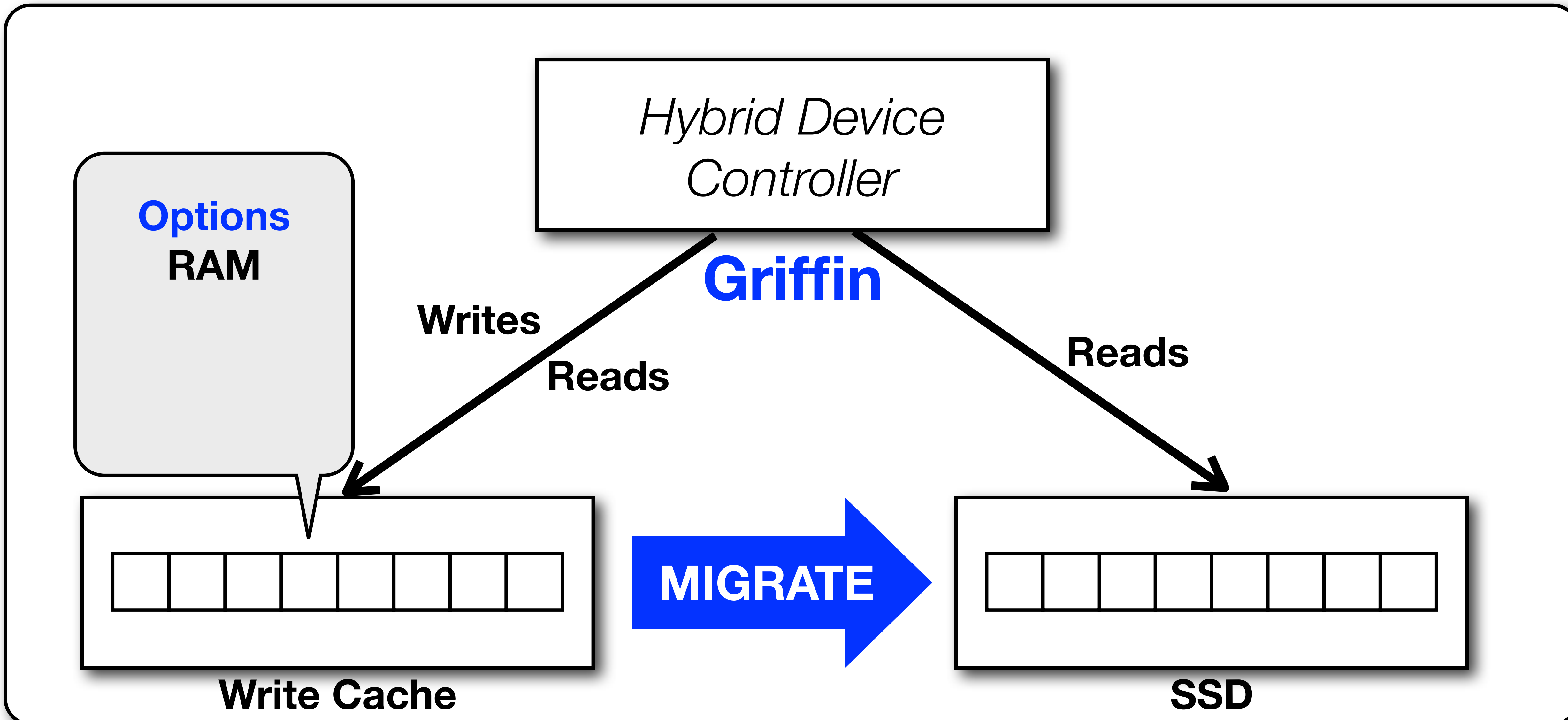
Griffin Hybrid Device



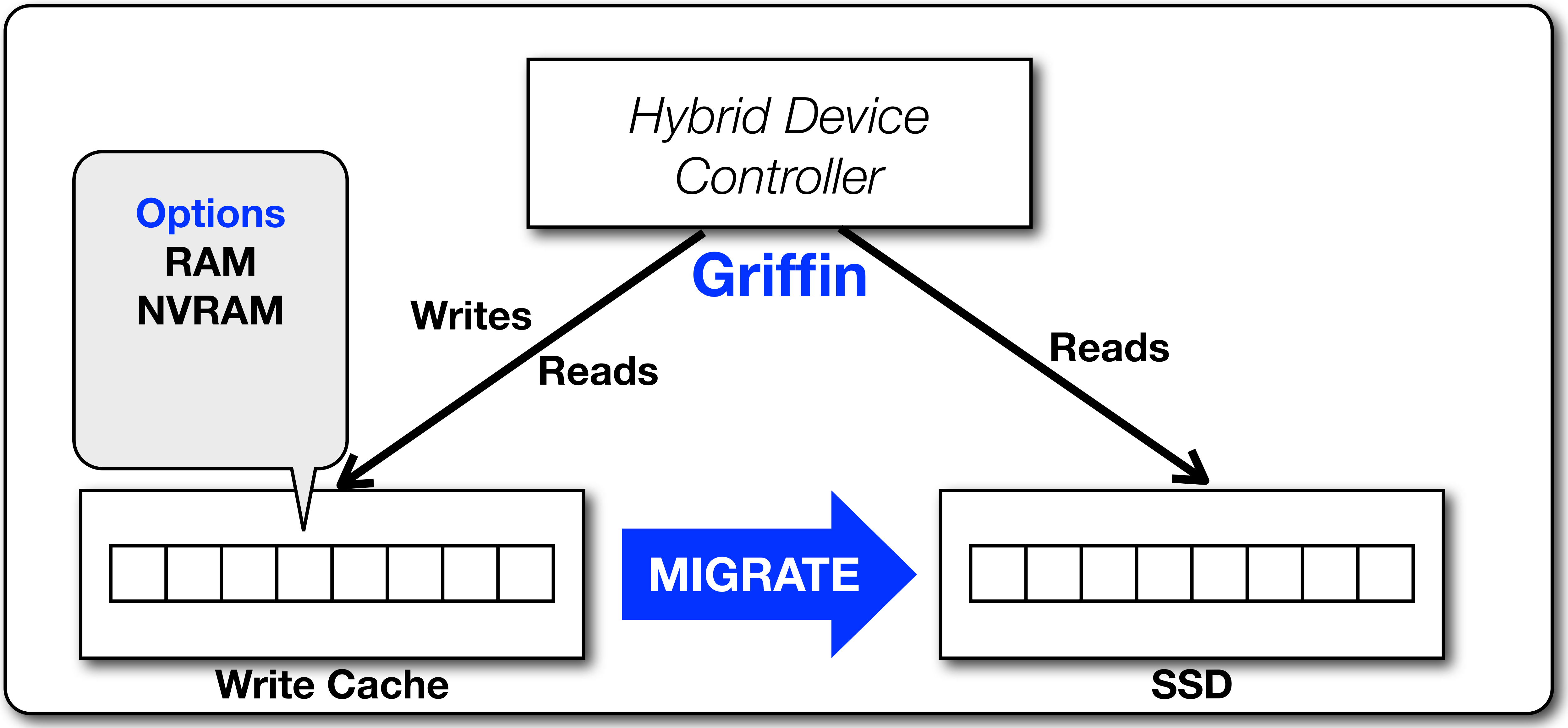
Griffin Hybrid Device



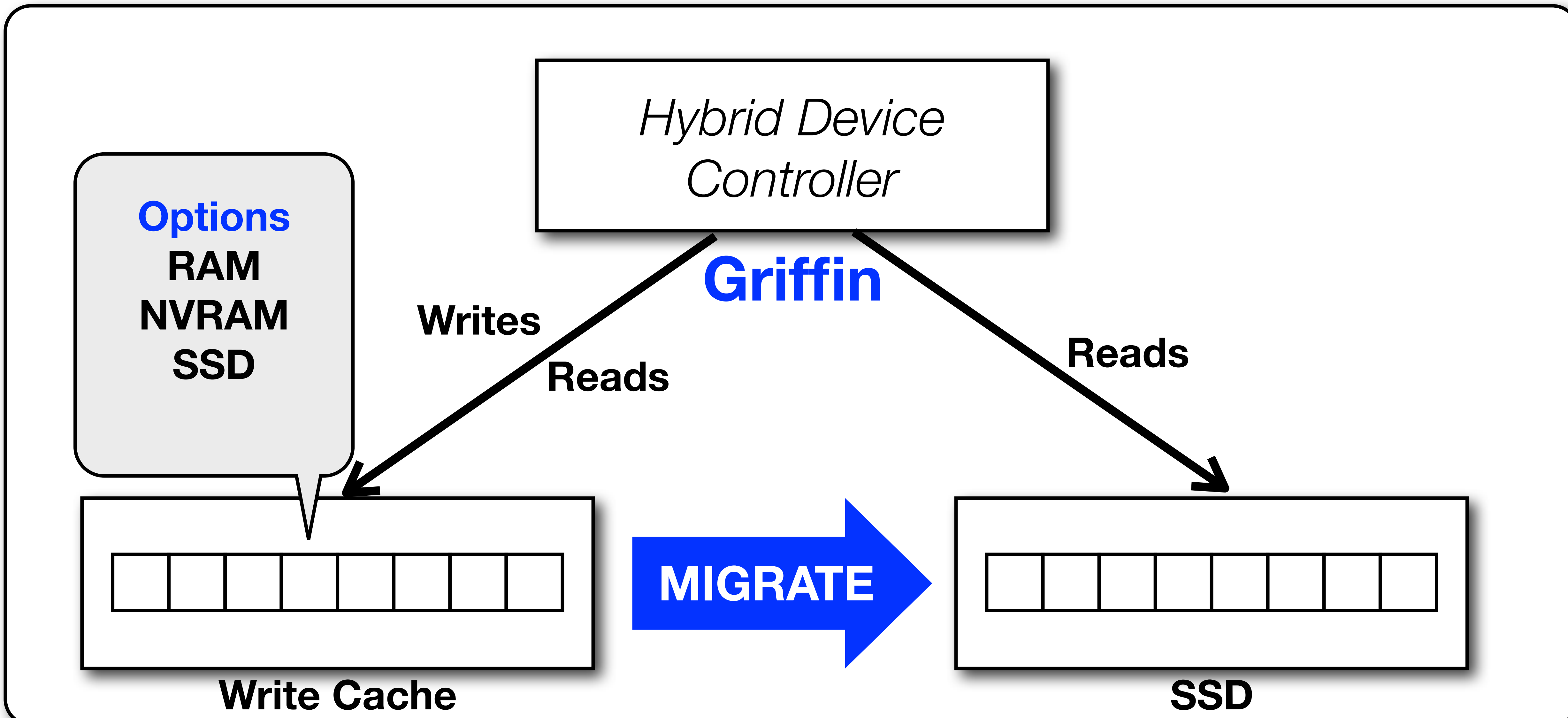
Griffin Hybrid Device



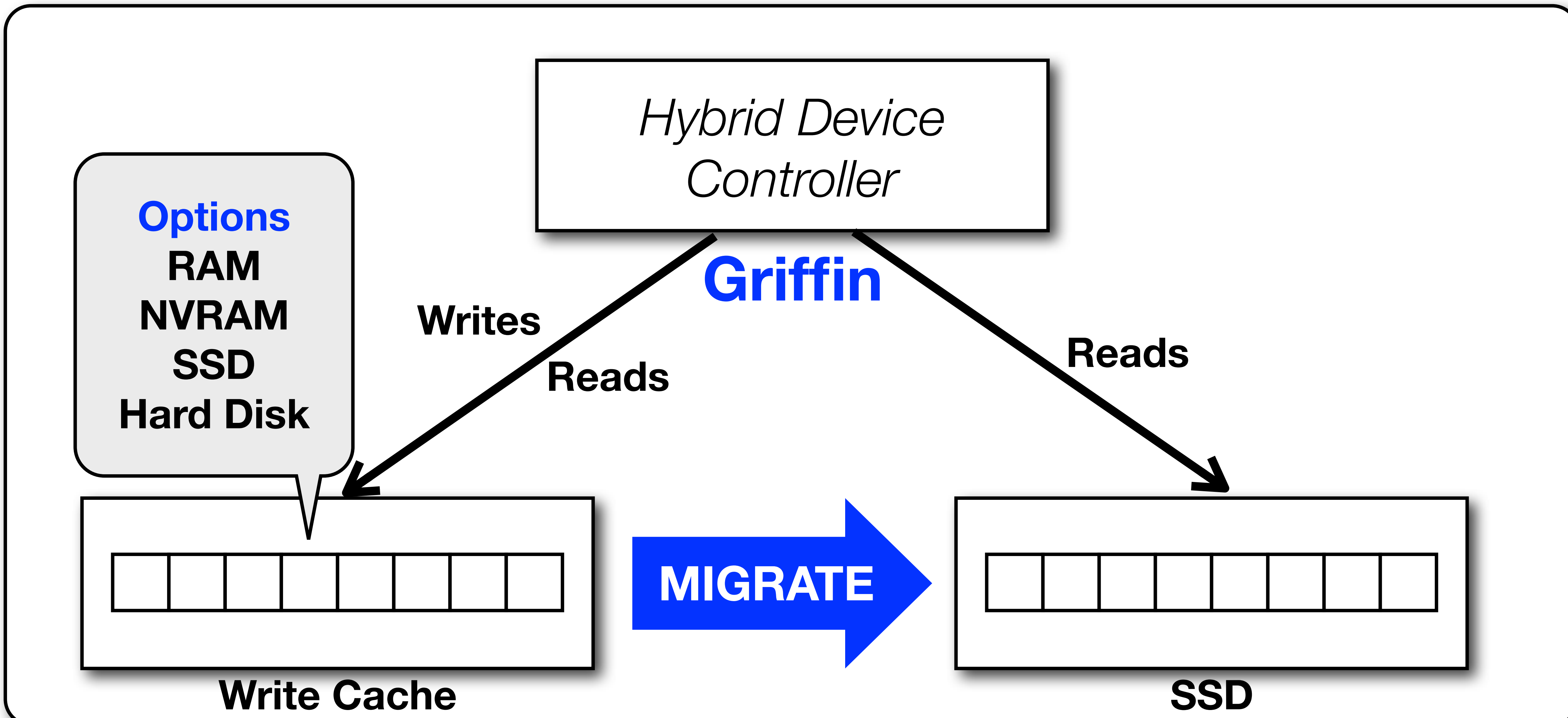
Griffin Hybrid Device



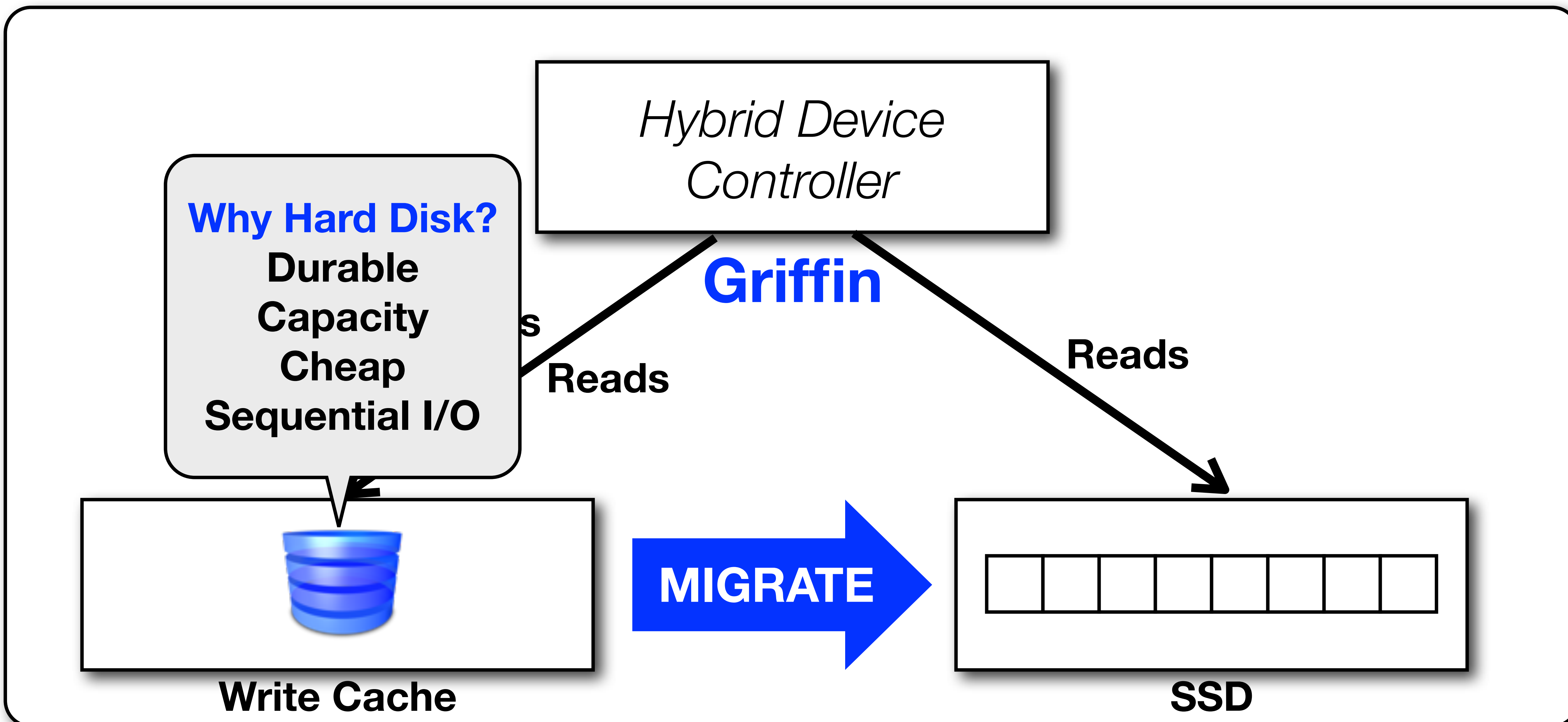
Griffin Hybrid Device



Griffin Hybrid Device



Griffin Hybrid Device



Contributions

Contributions

- **Characterized I/O patterns**
  - Found desktop/server traces contains many overwrites
  - Caching overwrites reduces writes to SSD by 52% ideally

Contributions

- **Characterized I/O patterns**
  - Found desktop/server traces contains many overwrites
  - Caching overwrites reduces writes to SSD by 52% ideally

- **Designed Griffin hybrid disk**
  - Uses a disk-based write cache
  - Cache data on hard disk, periodically move it back to SSD
  - Shows a 2x lifetime improvement (< 5% HDD reads)

Outline

- **Motivating Workload Characteristics**
  - Key features
- **Disk-based Write Caching**
- **Experimental Evaluation**
- **Conclusions**

I/O Workload Characterization

- **Examined various I/O traces**
  - Desktops (Internal Microsoft traces)
  - Servers (Narayanan et. al from MSR Cambridge)
    - Use only the write-intensive traces
  - Linux (Bhadkamkar et. al. from FIU)
    - Contains desktop, SVN, and web server
- **Trace descriptions**
  - Block I/Os collected below the filesystem buffer cache
  - Multi-hour traces of 5 hours to 176 hours
  - Between 209K to 543M I/O events per trace

Key Metrics

Key Metrics

- **Overwrites**
  - *Consecutive writes to a block without an intervening read*

Key Metrics

- **Overwrites**
  - *Consecutive writes to a block without an intervening read*

W: 100	W: 100	W: 100	W: 100	R: 100	W: 200	R: 200	W: 200	R: 200

Key Metrics

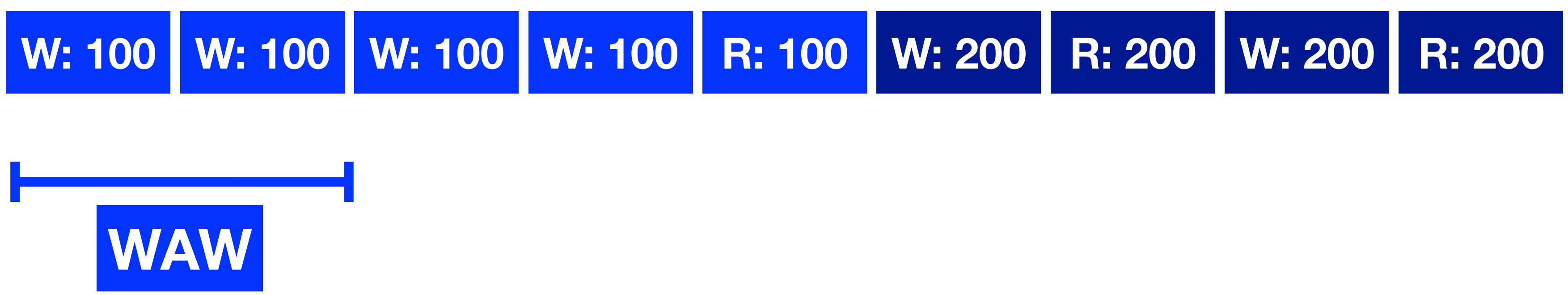
- **Overwrites**
  - *Consecutive writes to a block without an intervening read*

Key Metrics

- **Overwrites**
  - *Consecutive writes to a block without an intervening read*

Key Metrics

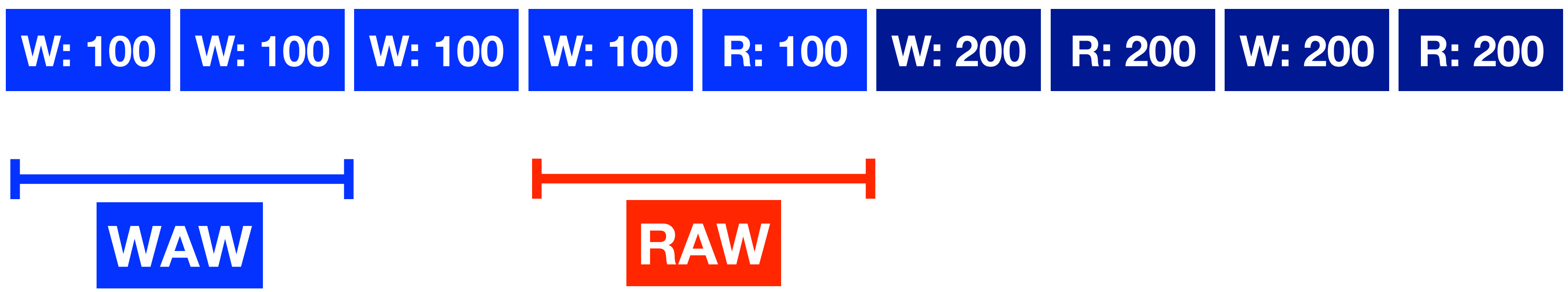
- **Overwrites**
  - *Consecutive writes to a block without an intervening read*



- **Write-after-Write (WAW) Times**
  - *Time between two consecutive writes to the same block*

Key Metrics

- **Overwrites**
  - *Consecutive writes to a block without an intervening read*

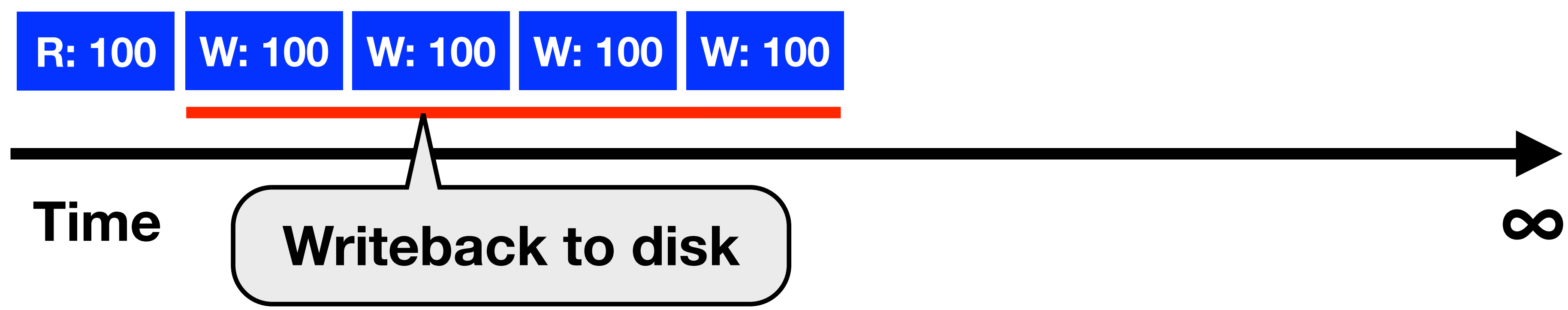


- **Write-after-Write (WAW) Times**
  - *Time between two consecutive writes to the same block*
- **Read-after-Write (RAW) Times**
  - *Time between a write and a subsequent read to the same block*

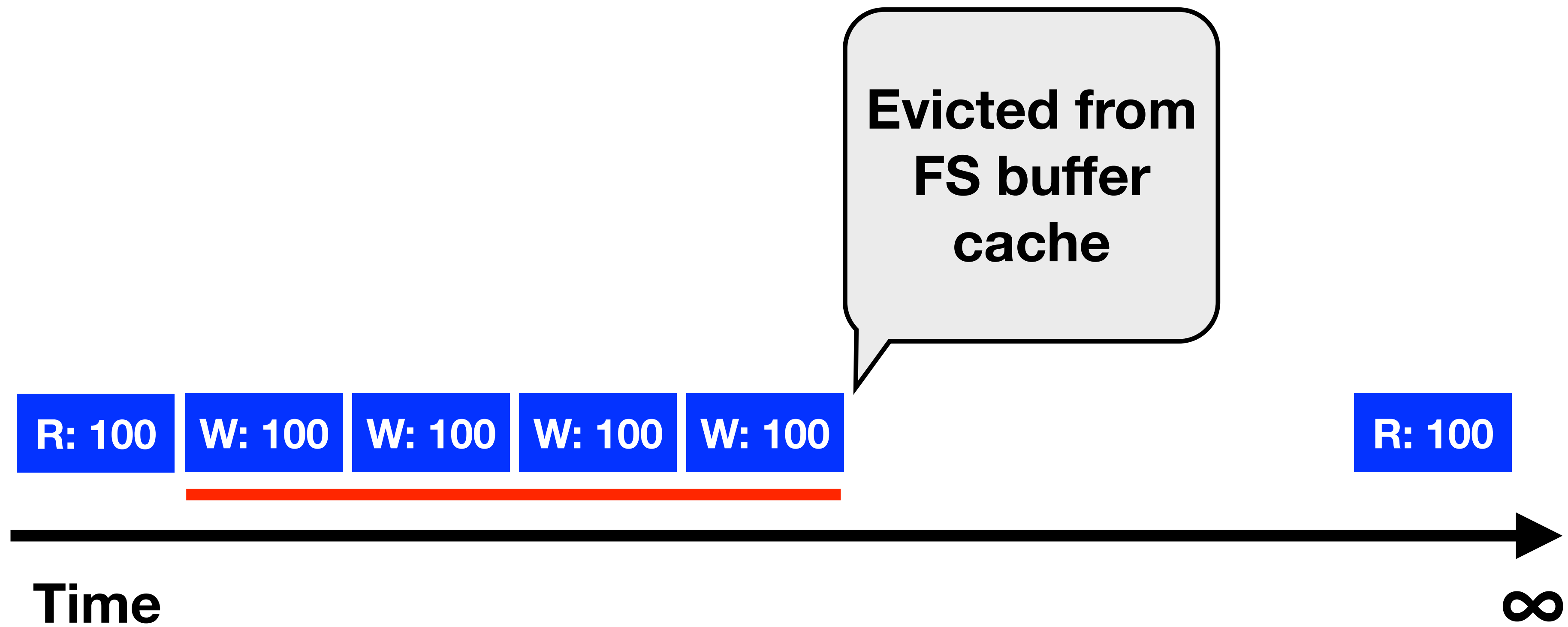
Why Overwrites? Lifetime of a Block

Why Overwrites? Lifetime of a Block

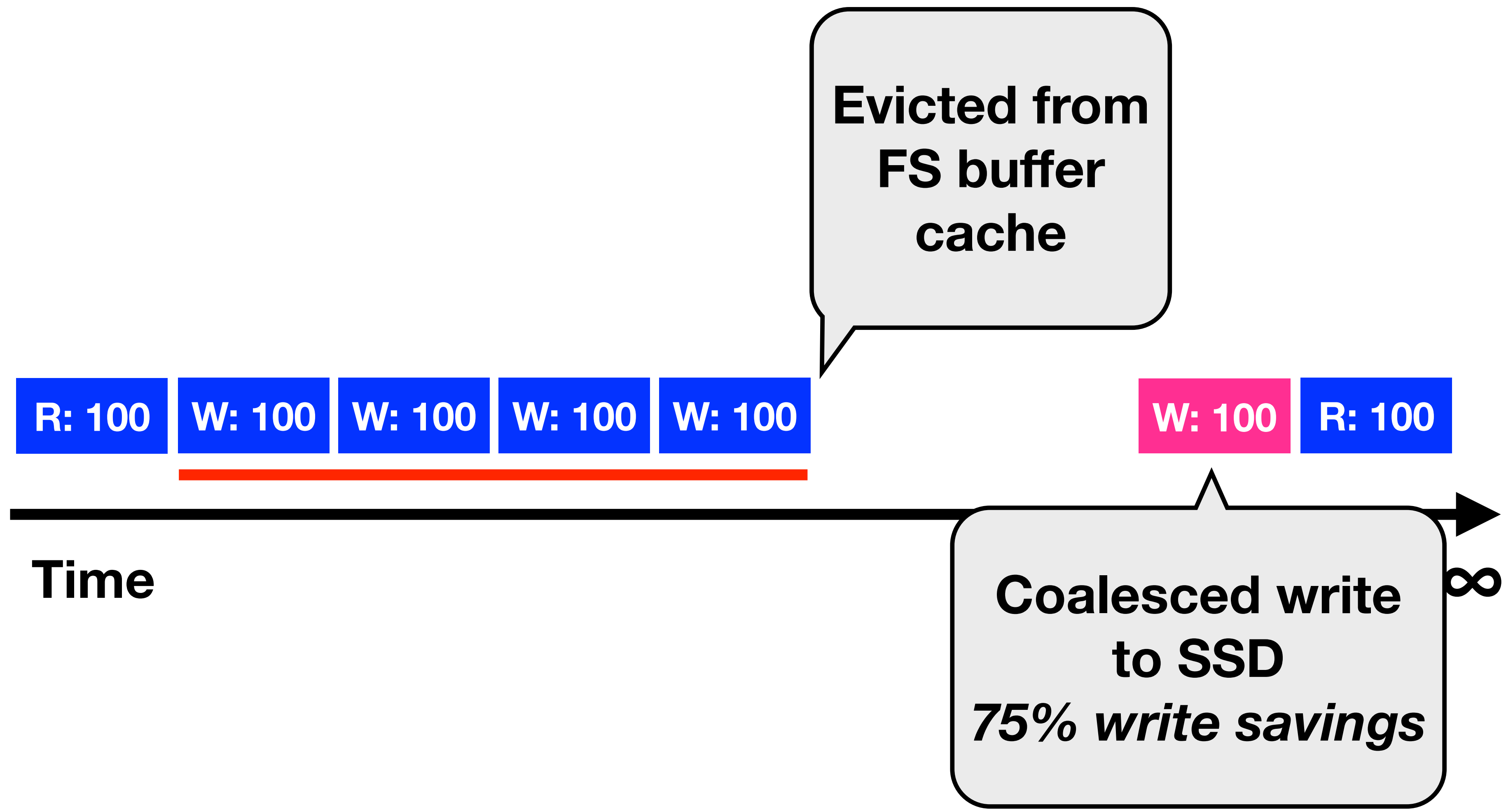
Why Overwrites? Lifetime of a Block



Why Overwrites? Lifetime of a Block



Why Overwrites? Lifetime of a Block

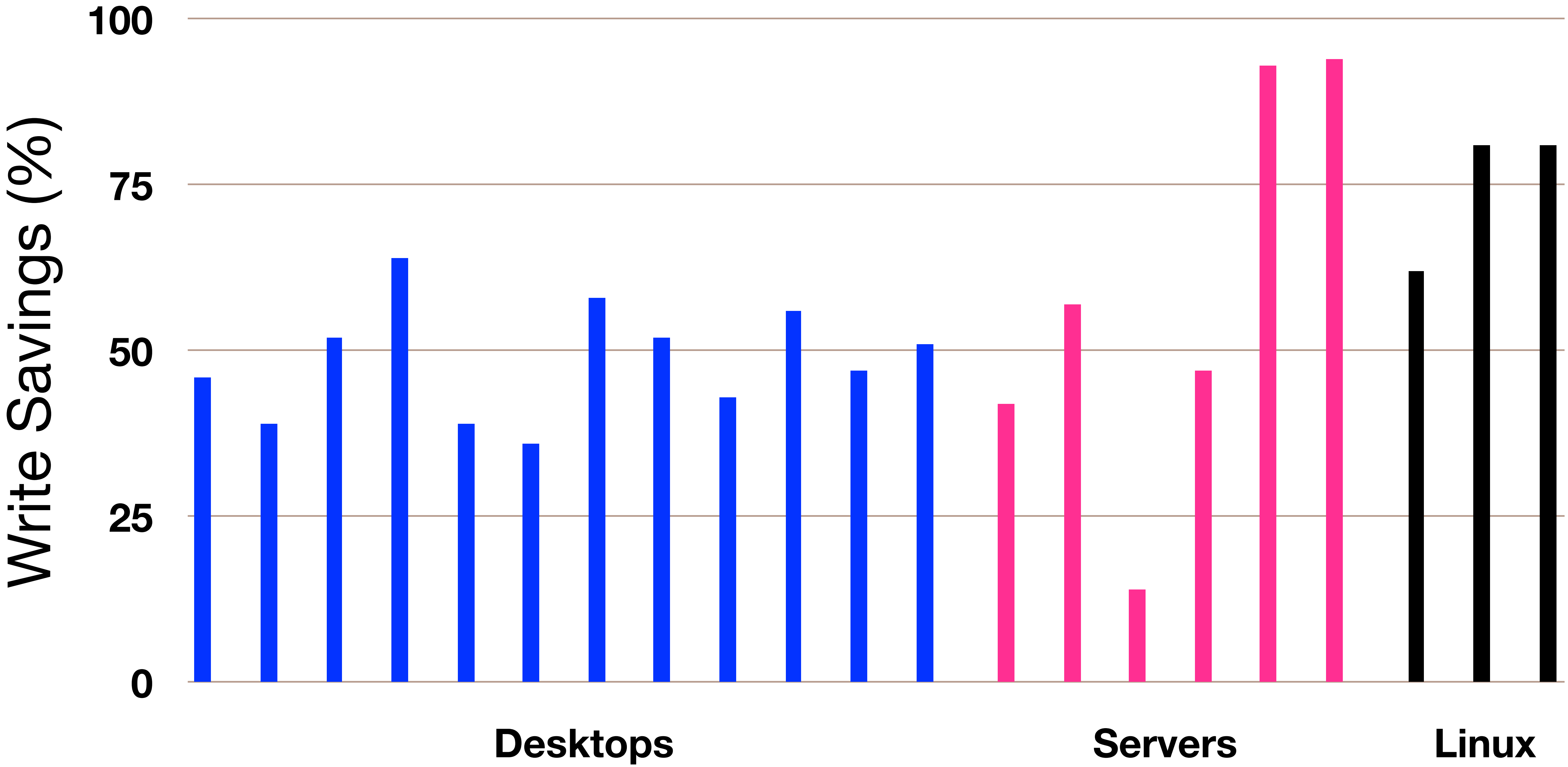


Top Overwritten Files in Desktops

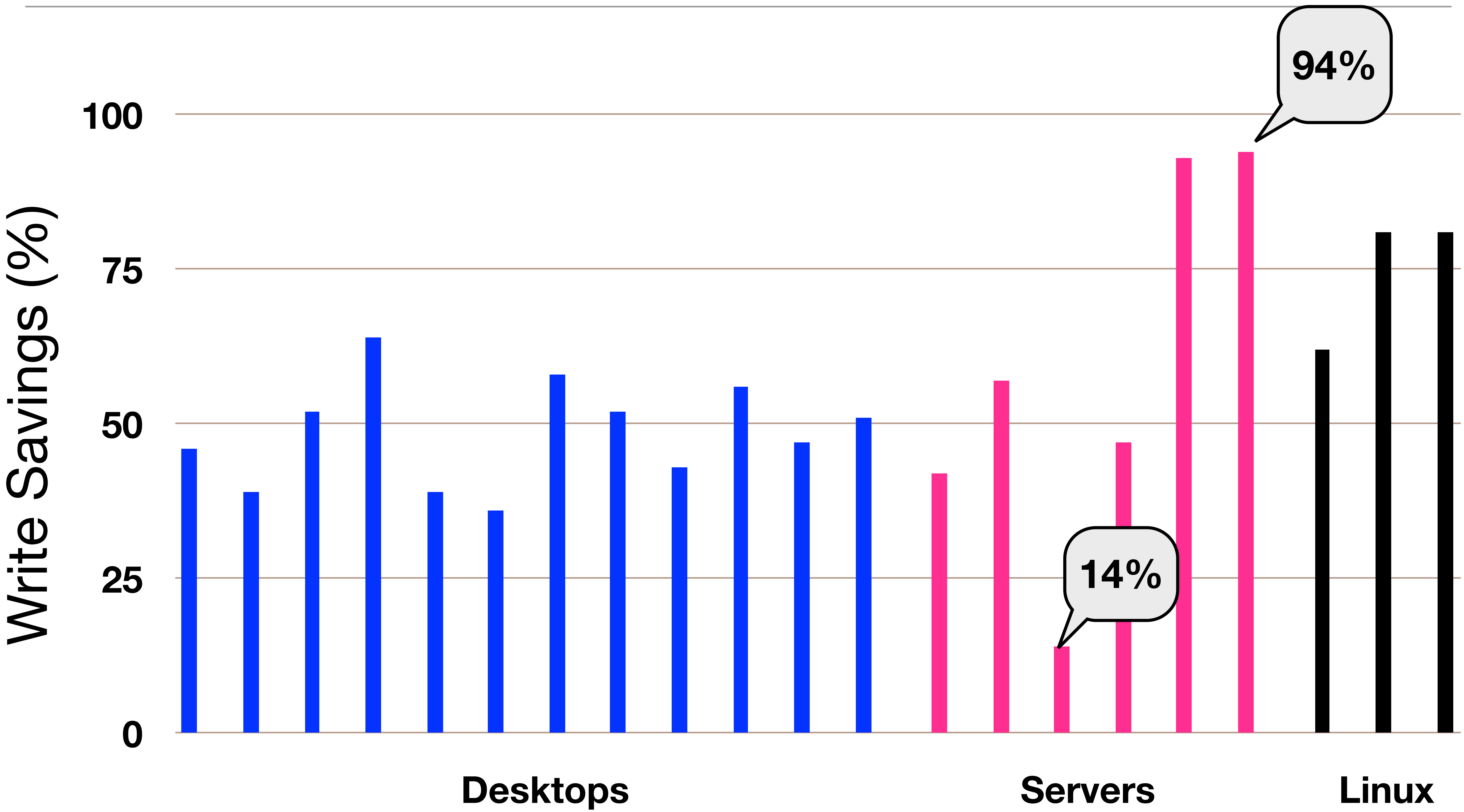
Rank	Filename
1	C:\Outlook.ost
2	C:\...**Search**\...\Windows.edb
3	C:\$Bitmap
4	C:\Windows\Prefetch\Layout.ini
5	C:\Users\<name>\NTUSER.DAT
6	C:\$Mft

Ideal Write Savings: Remove Overwrites

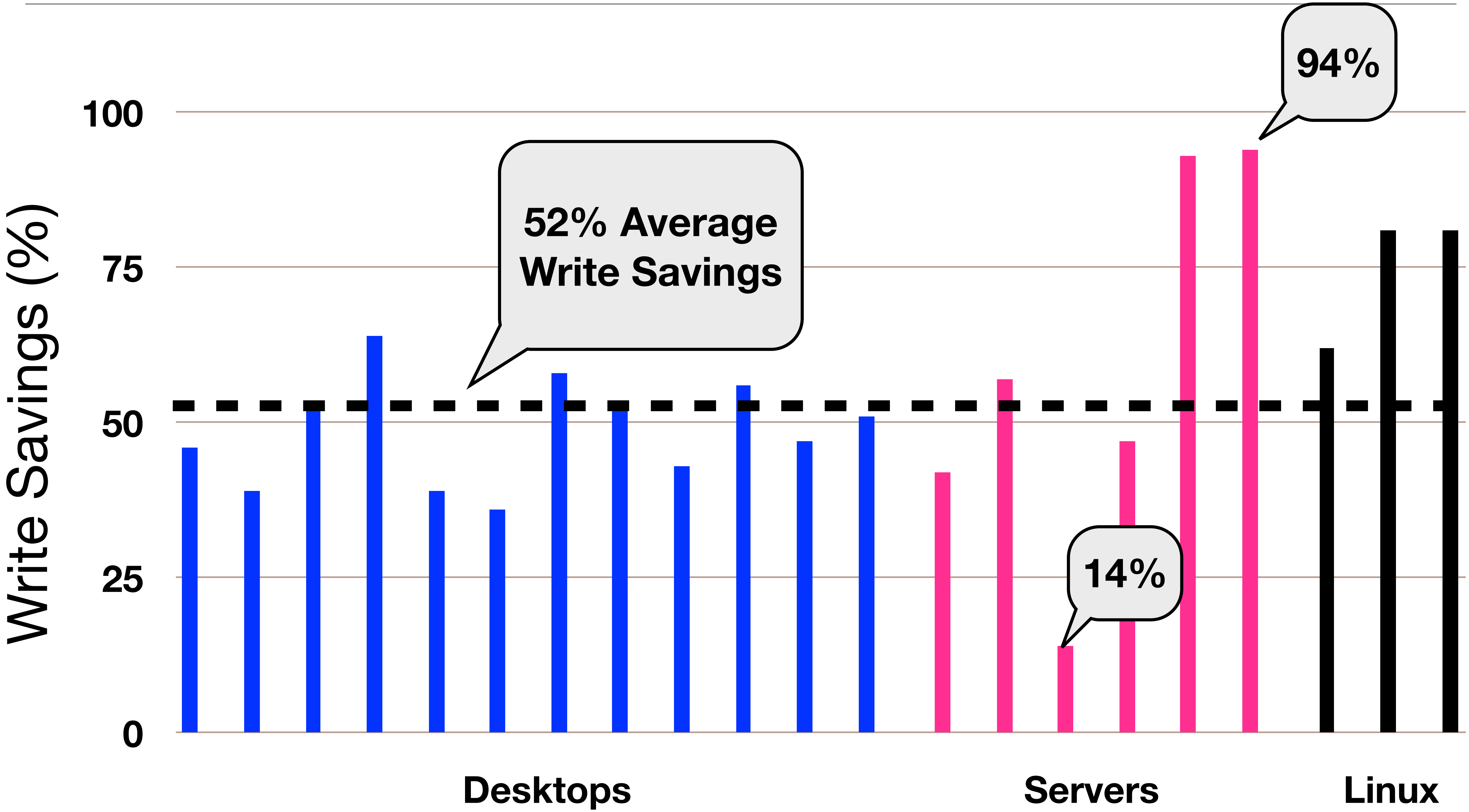
Ideal Write Savings: Remove Overwrites



Ideal Write Savings: Remove Overwrites

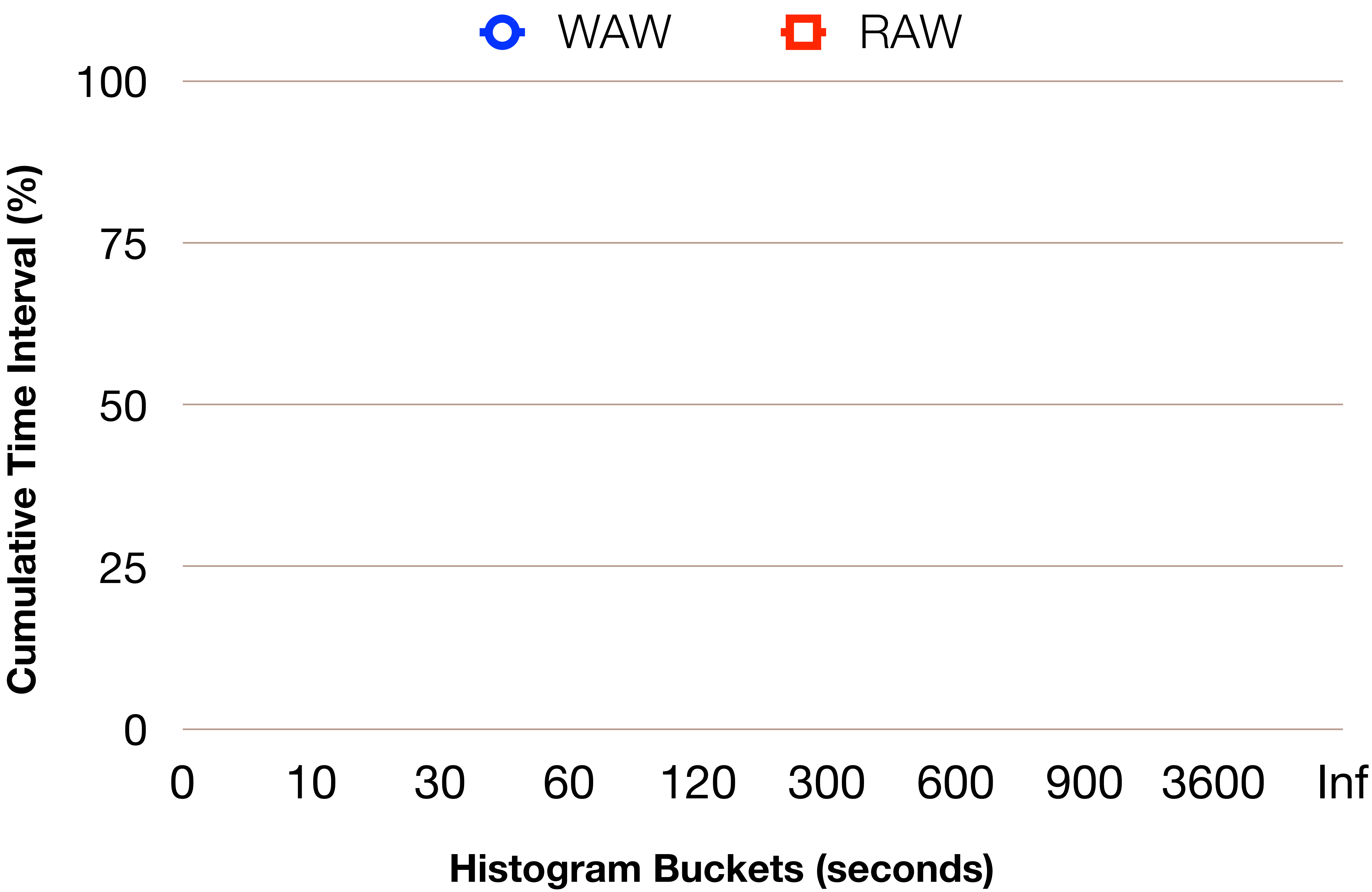


Ideal Write Savings: Remove Overwrites

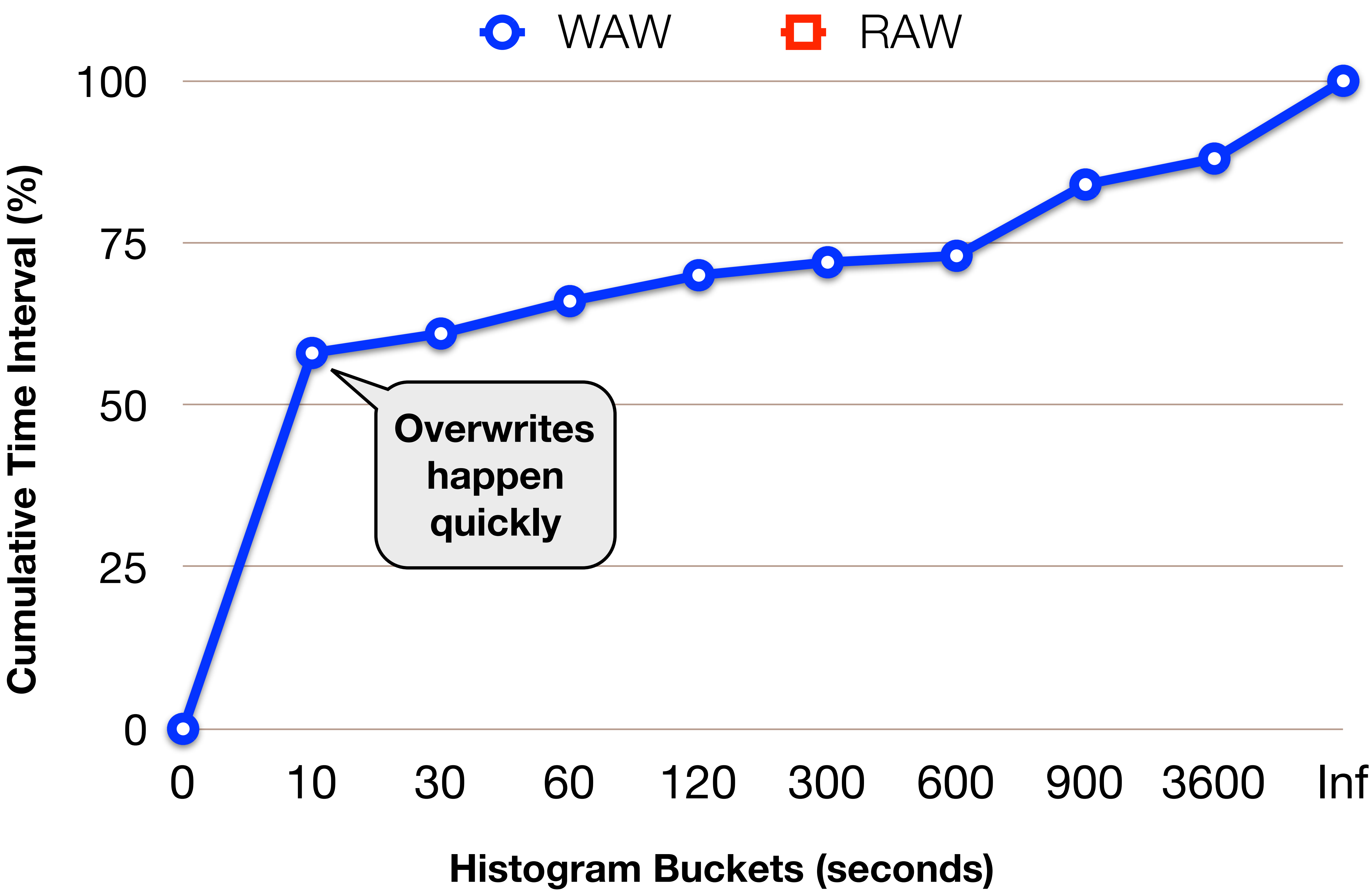


WAW/RAW Time Intervals

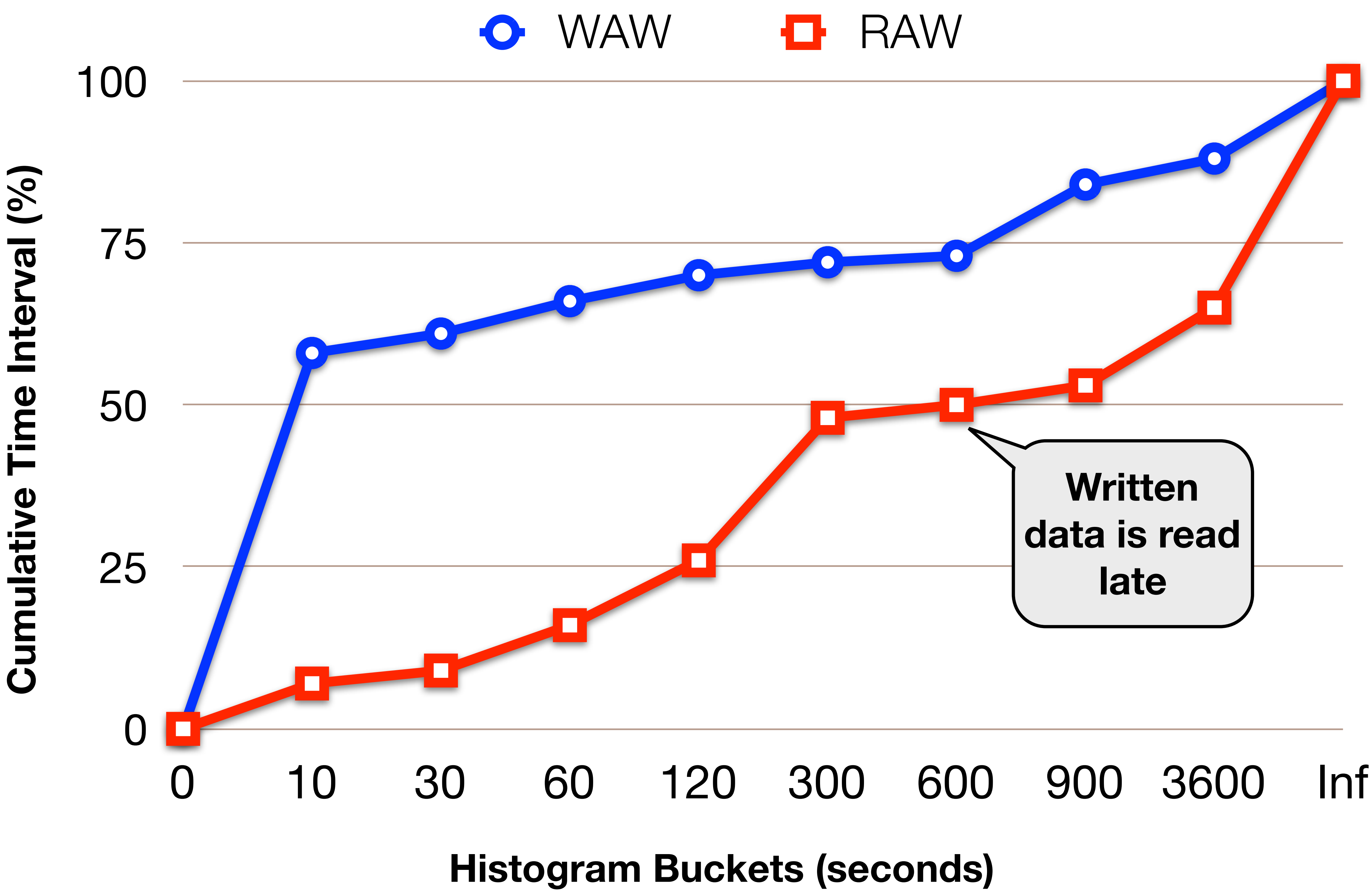
WAW/RAW Time Intervals



WAW/RAW Time Intervals



WAW/RAW Time Intervals



Summary of Observations

Summary of Observations

- **Large fraction of overwrites**
  - *Potential of 36% to 64% reduction for desktops*
  - *As much as 94% in server workloads (web server)*
  - *Linux: 62% in desktop and 81% in servers (web server)*

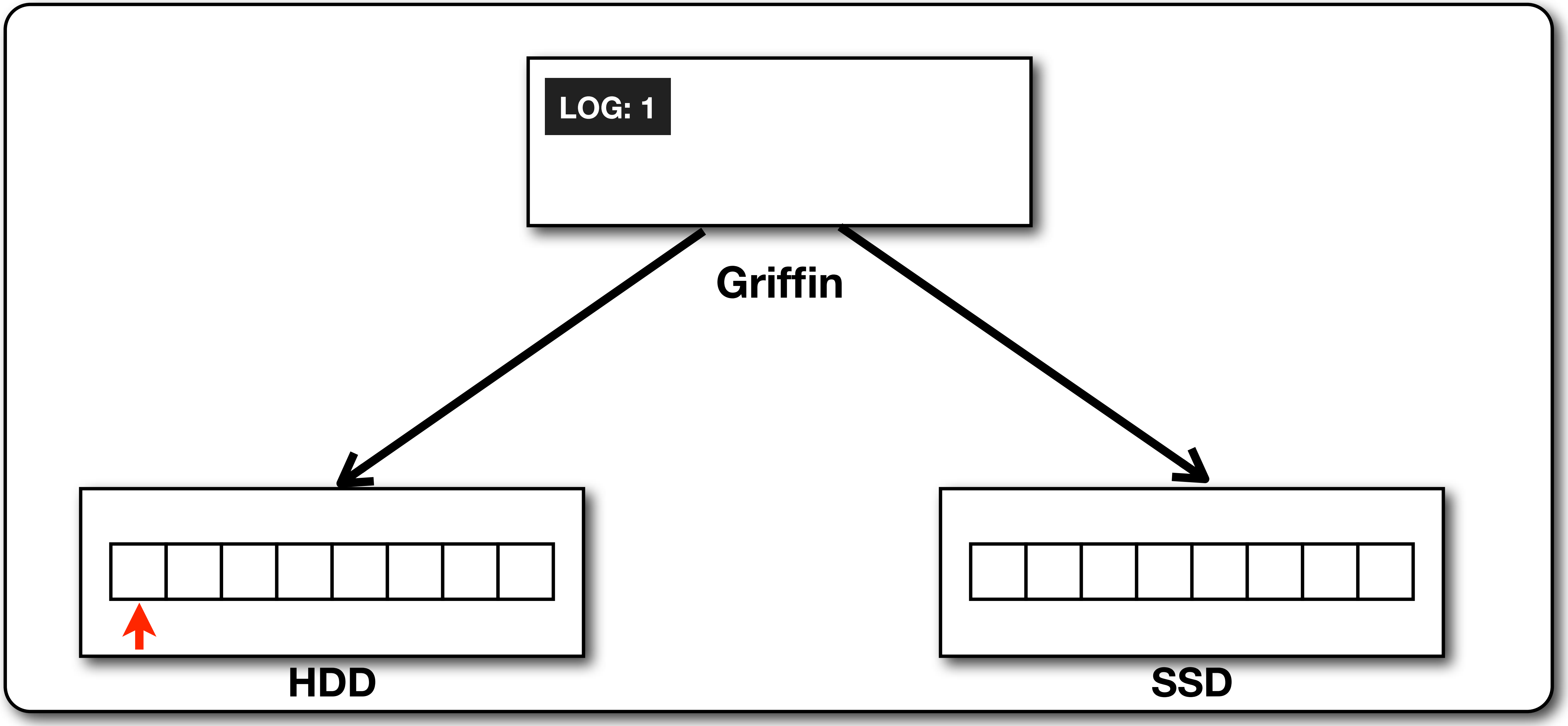
Summary of Observations

- **Large fraction of overwrites**
  - *Potential of 36% to 64% reduction for desktops*
  - *As much as 94% in server workloads (web server)*
  - *Linux: 62% in desktop and 81% in servers (web server)*
- **Overwrites happen quickly, reads after a long interval**
  - *Over 50% of overwrites within 30 seconds*
  - *Only 21% of written data read within 15 minutes*

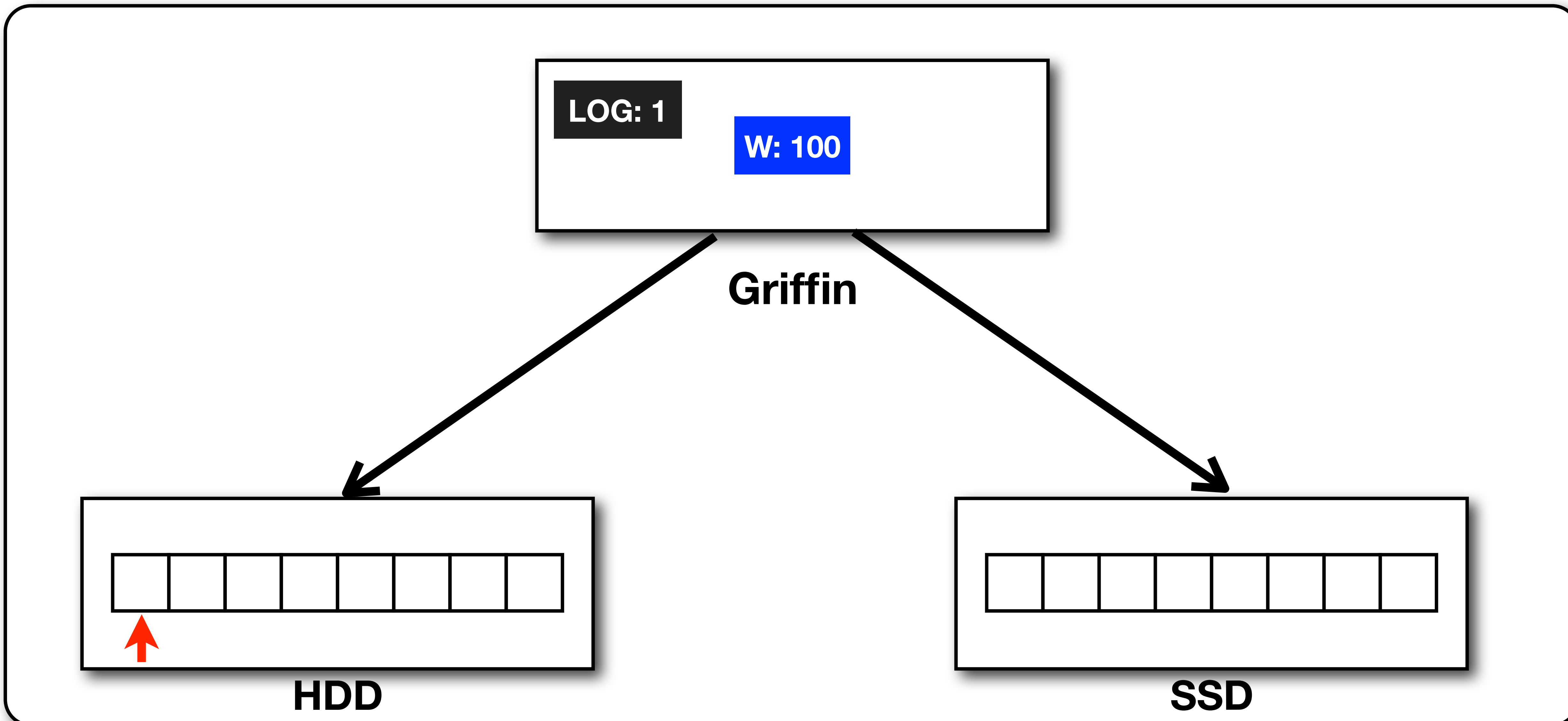
Outline

- **Motivating Workload Characteristics**
- **Disk-based Write Caching**
  - Basic algorithm
  - Performance tradeoffs
- **Experimental Results**
- **Conclusions**

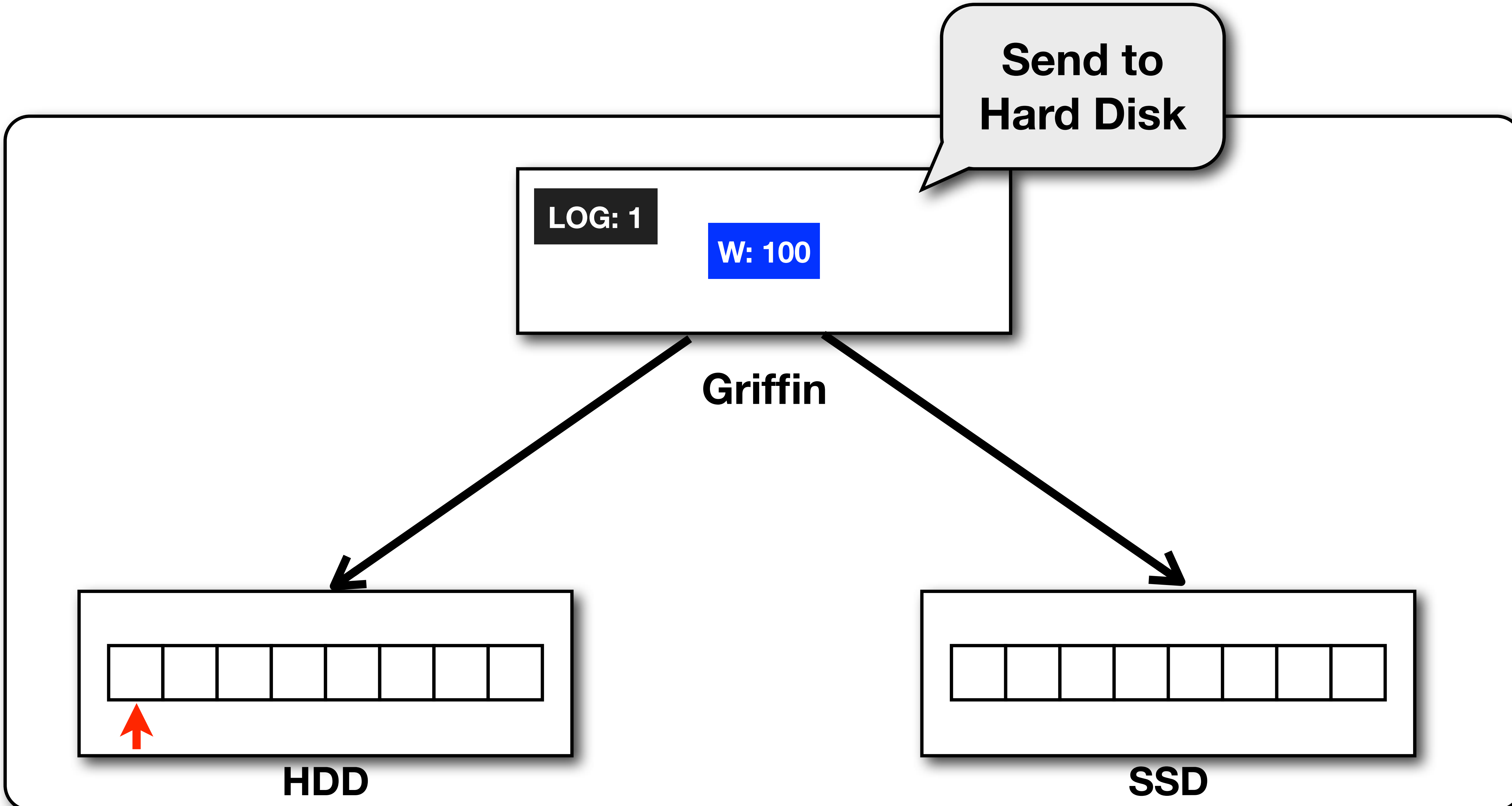
Griffin: Handling Writes



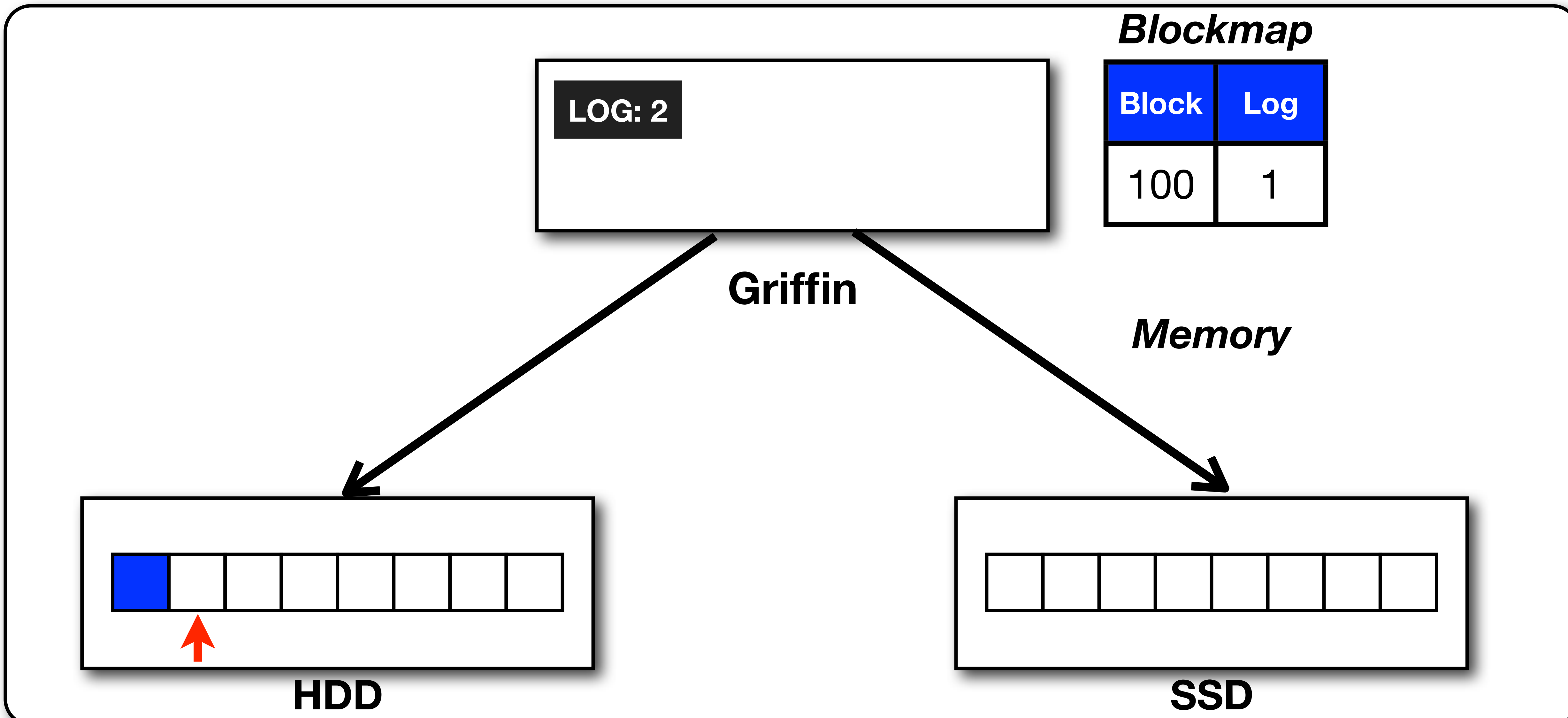
Griffin: Handling Writes



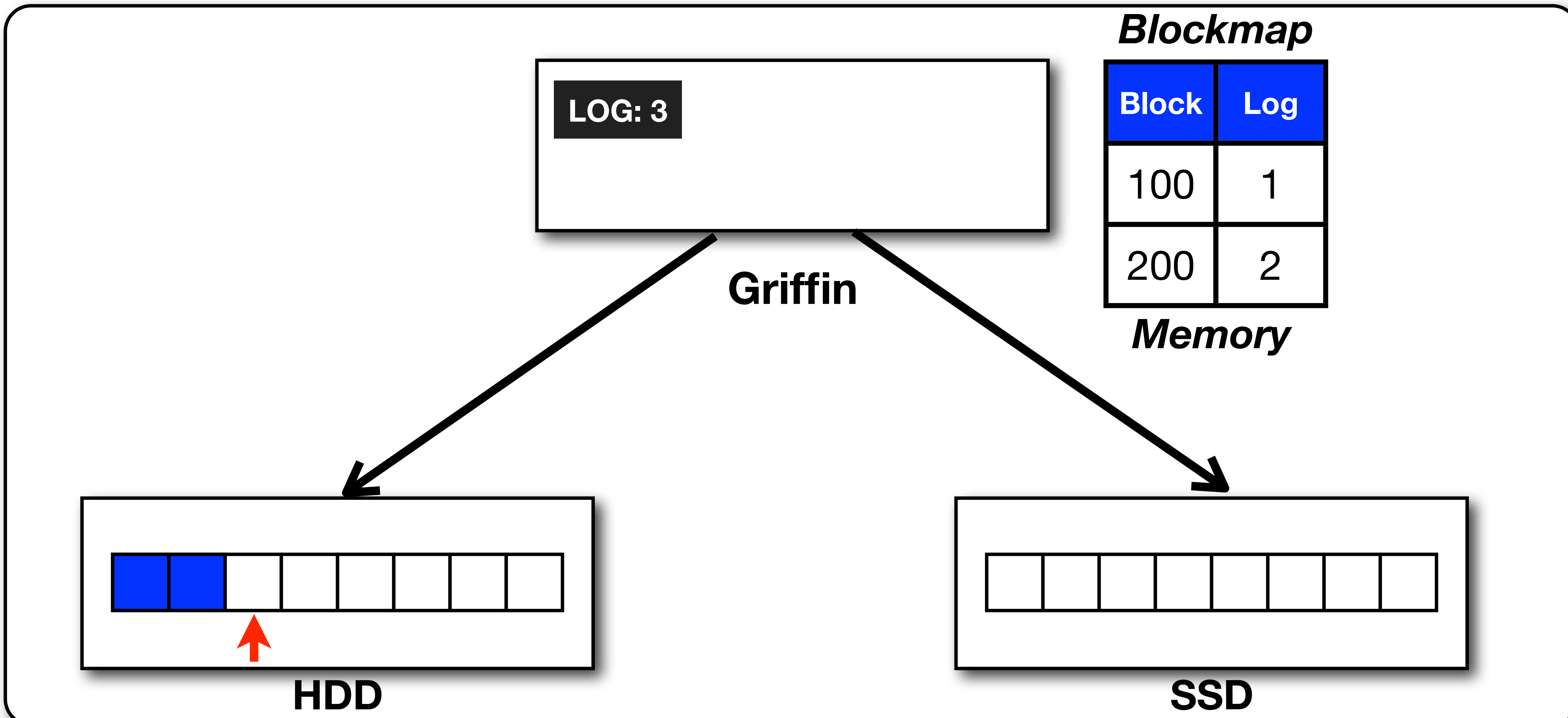
Griffin: Handling Writes



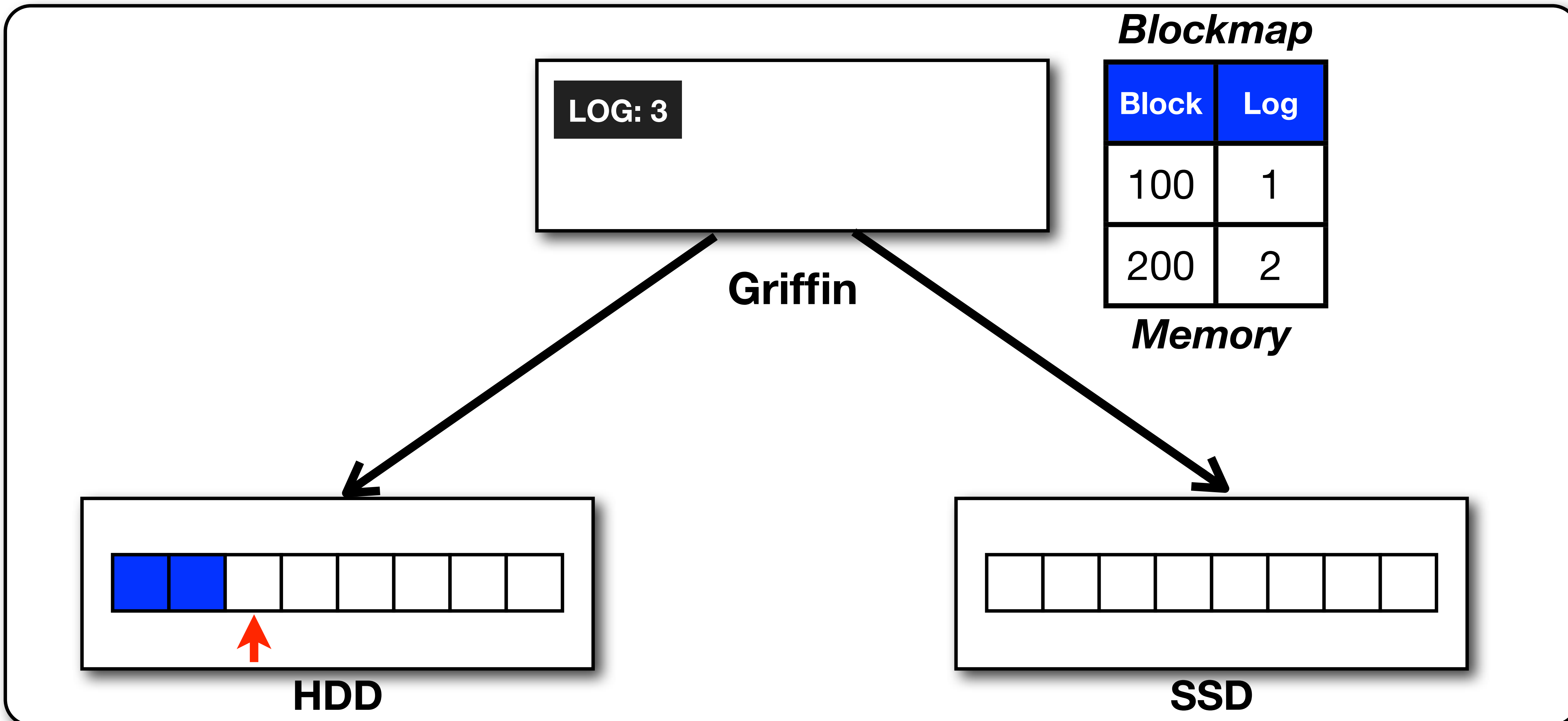
Griffin: Handling Writes



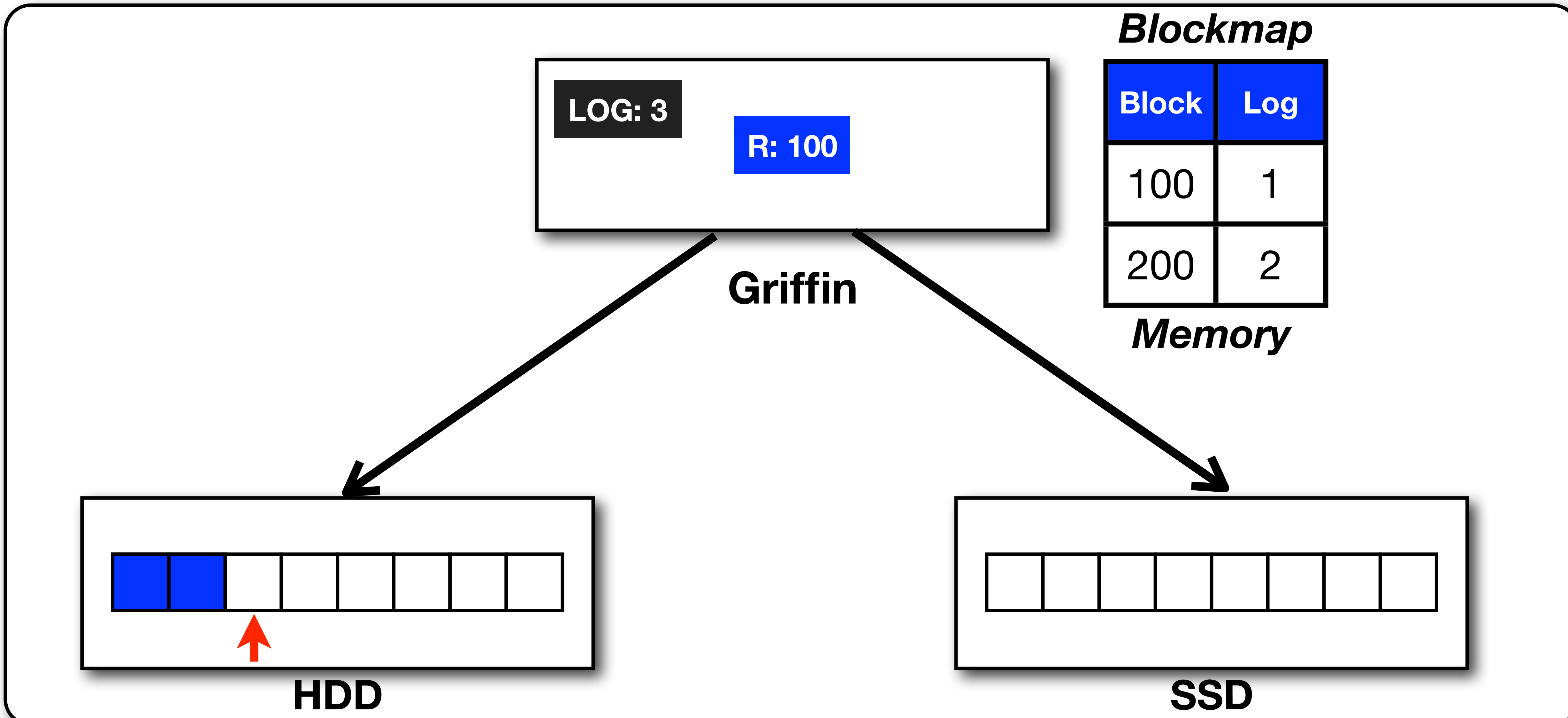
Griffin: Handling Writes



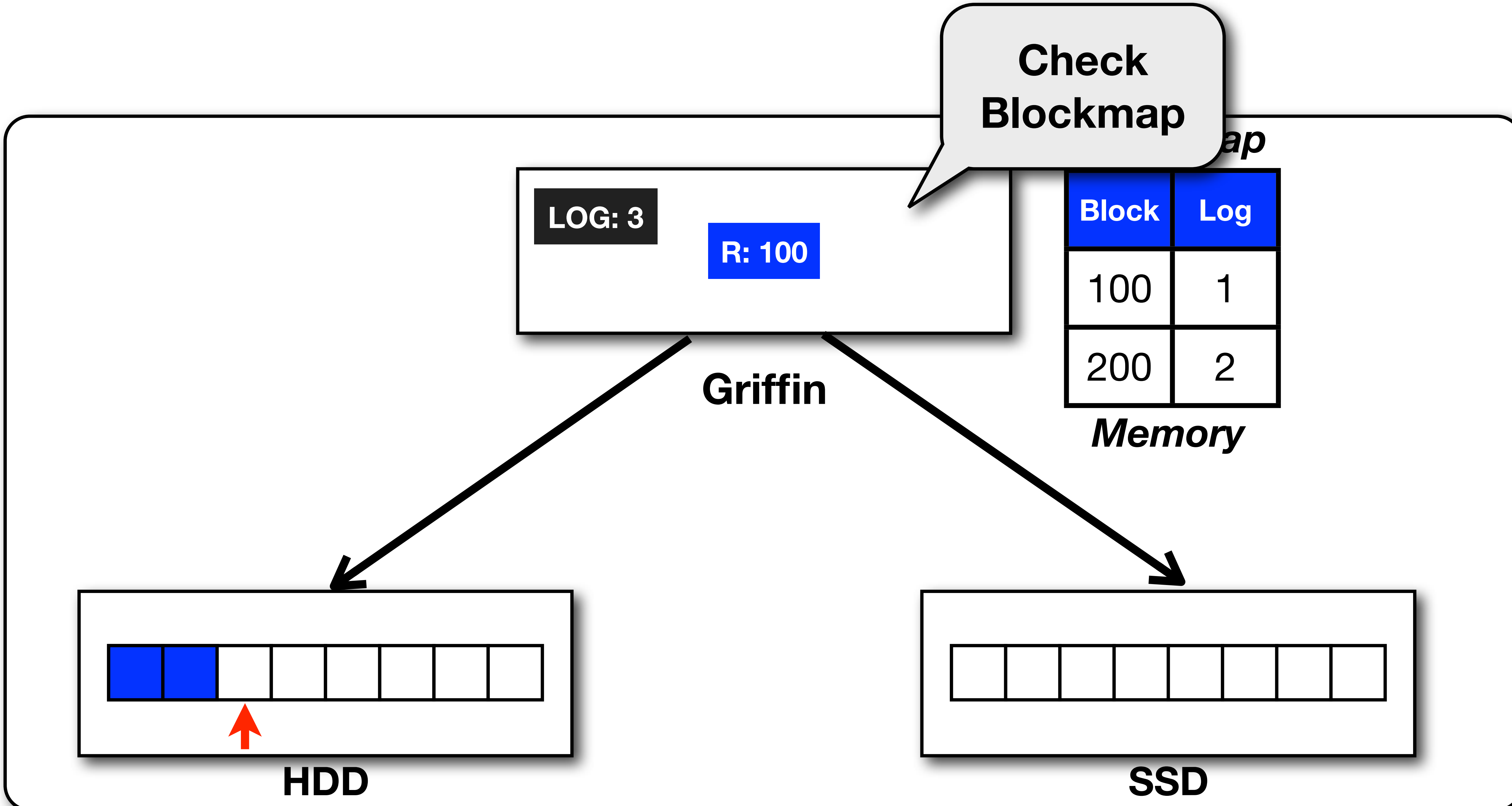
Griffin: Handling Reads



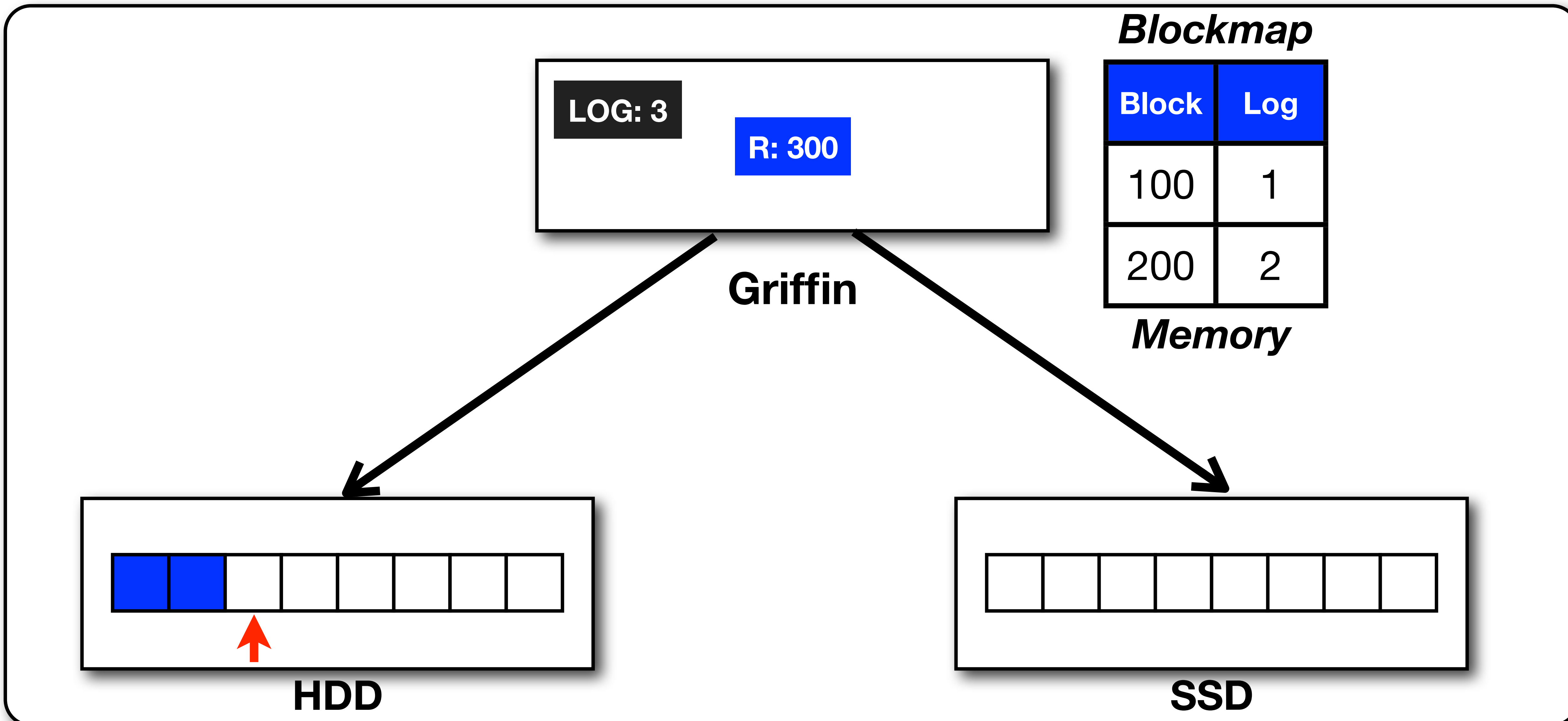
Griffin: Handling Reads



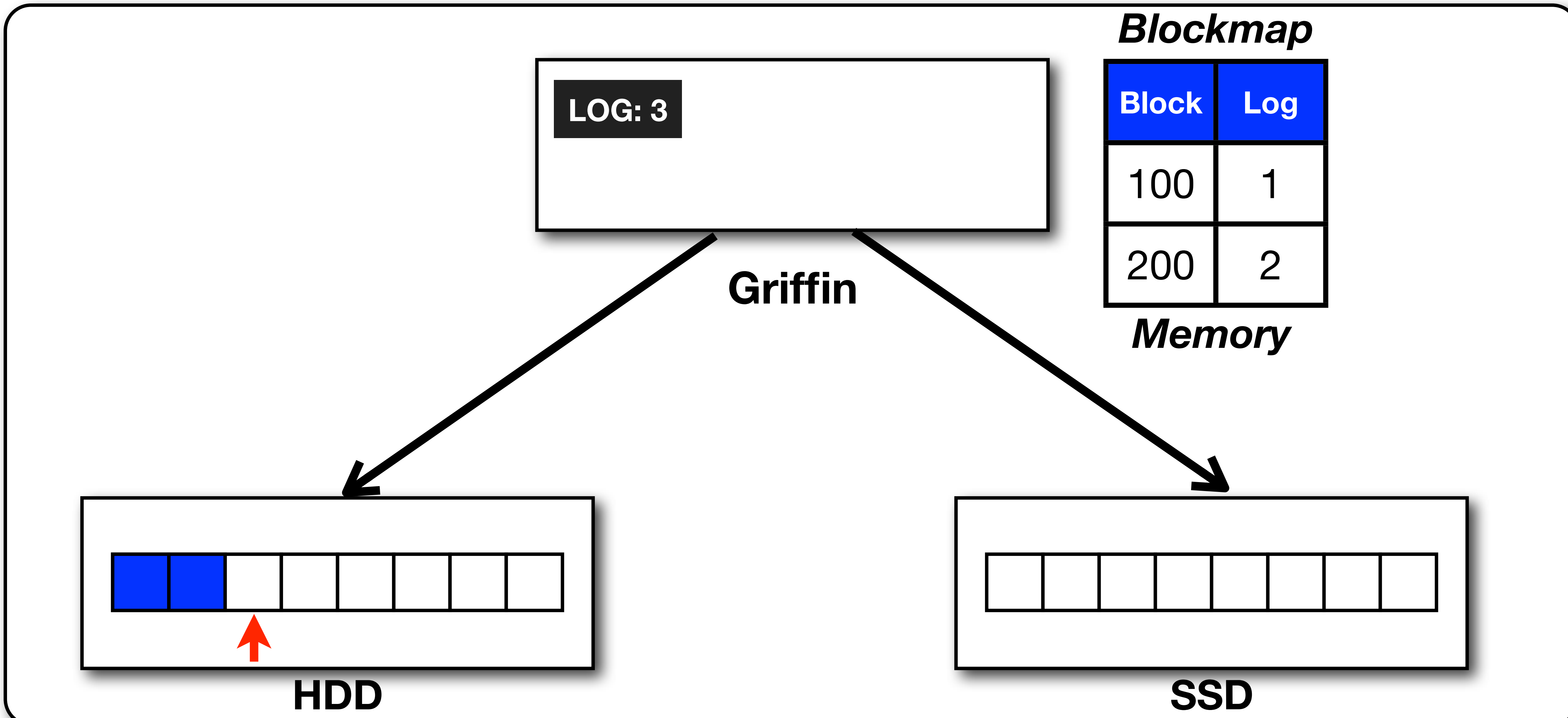
Griffin: Handling Reads



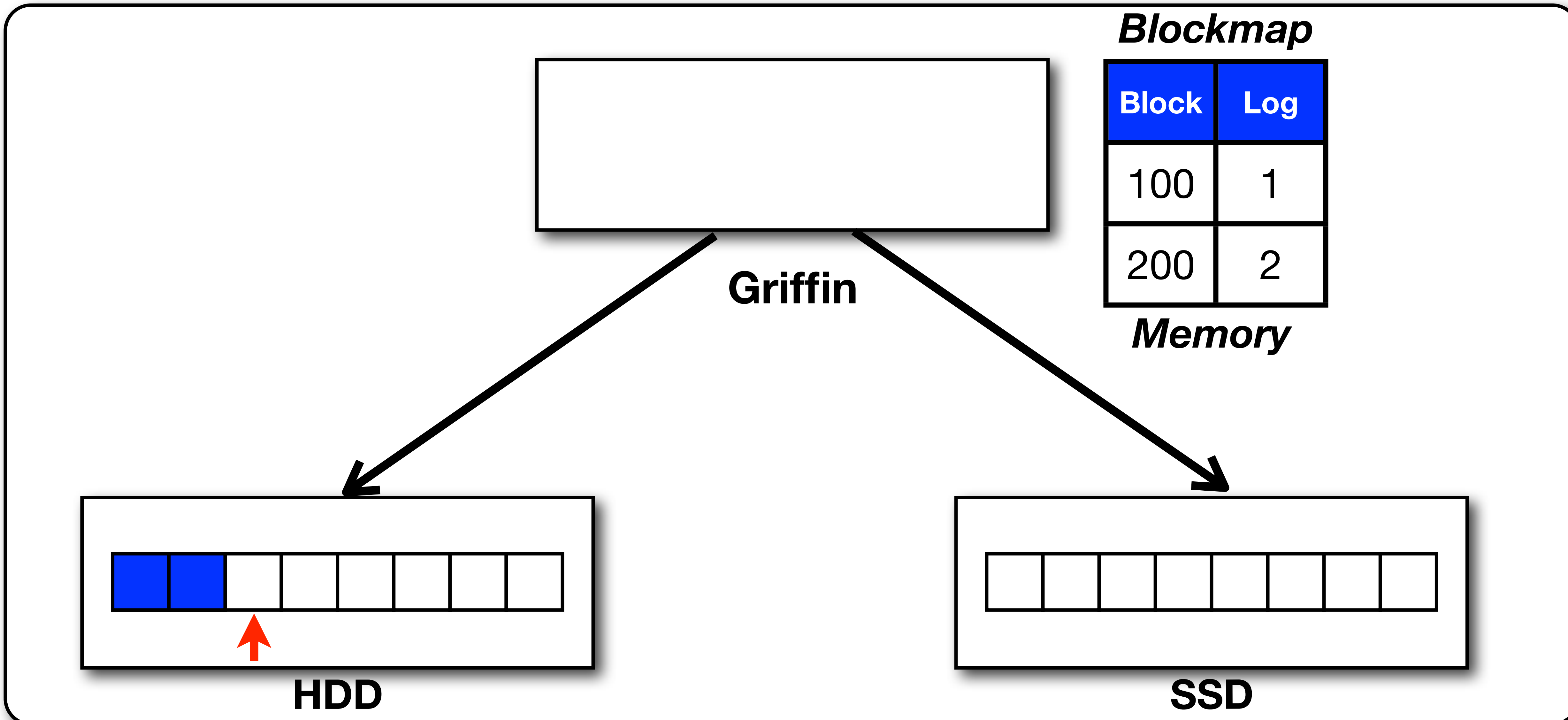
Griffin: Handling Reads



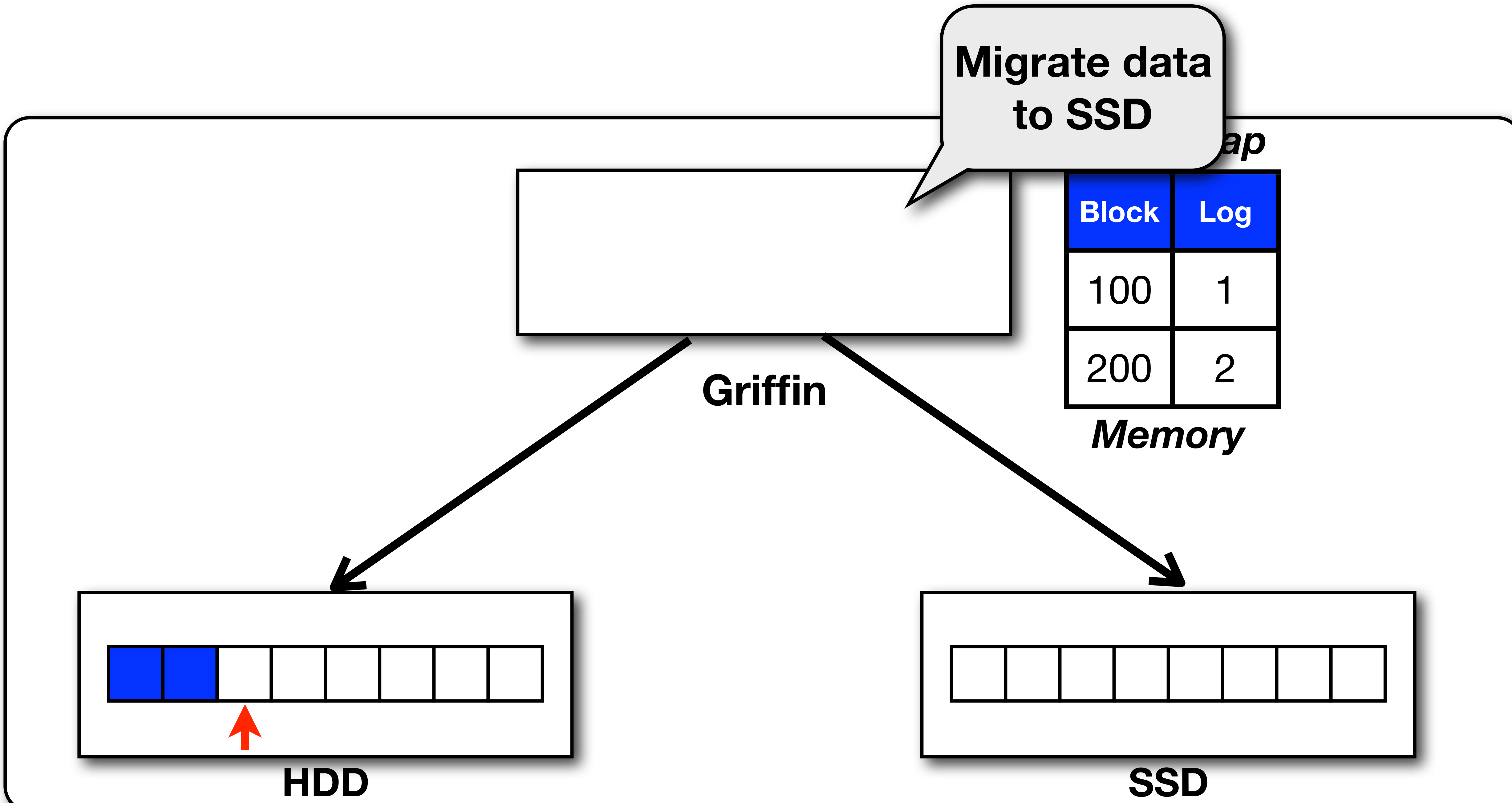
Griffin: Handling Reads



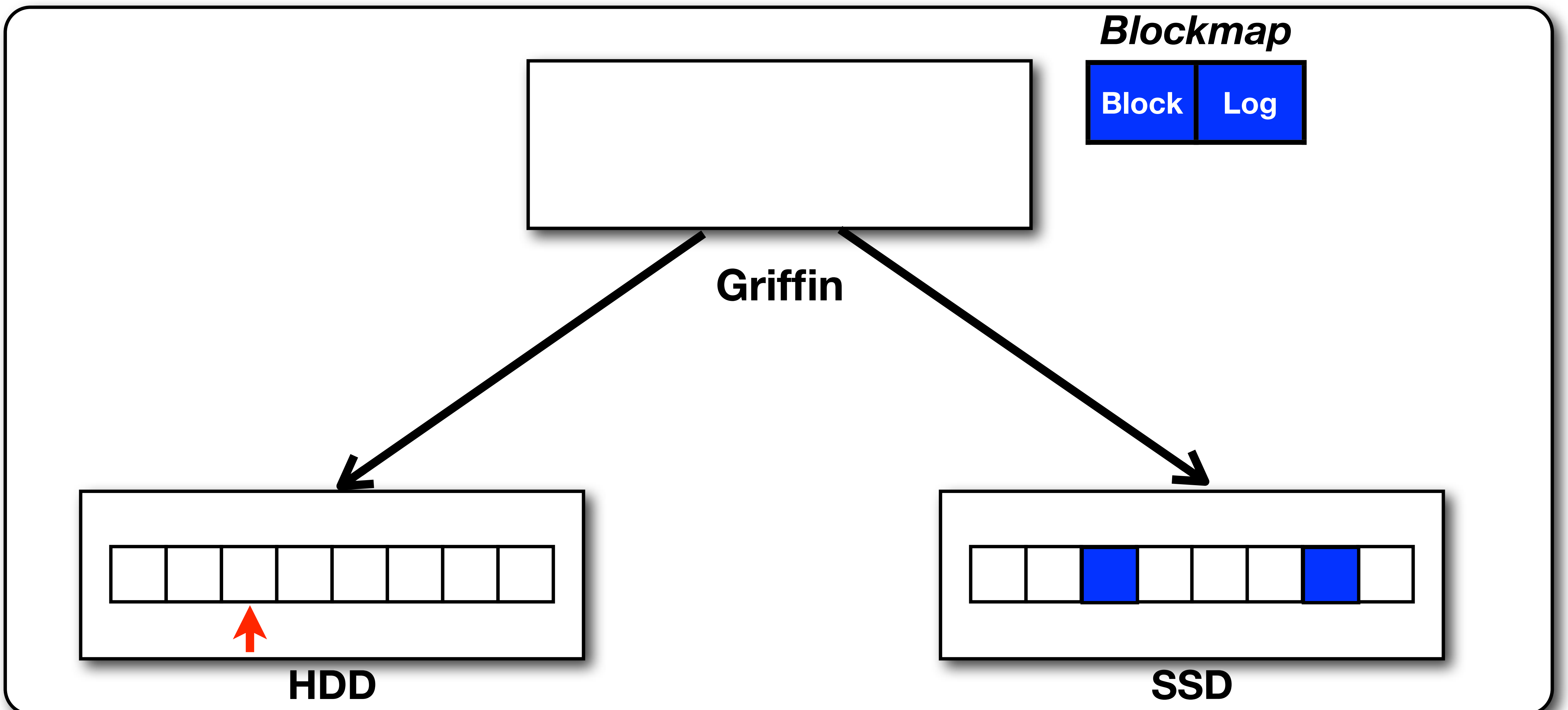
Griffin: Data Migration



Griffin: Data Migration

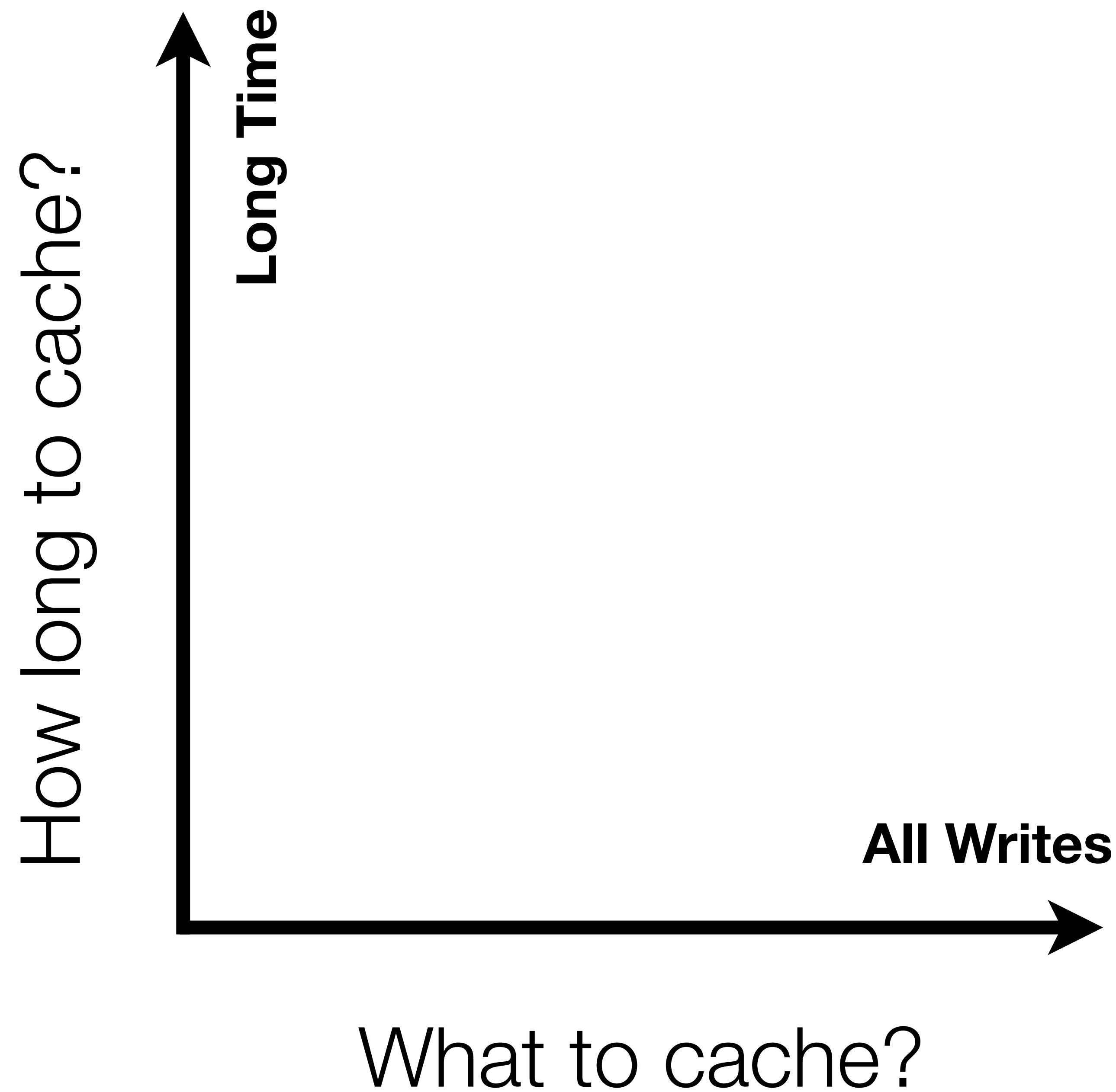


Griffin: Data Migration

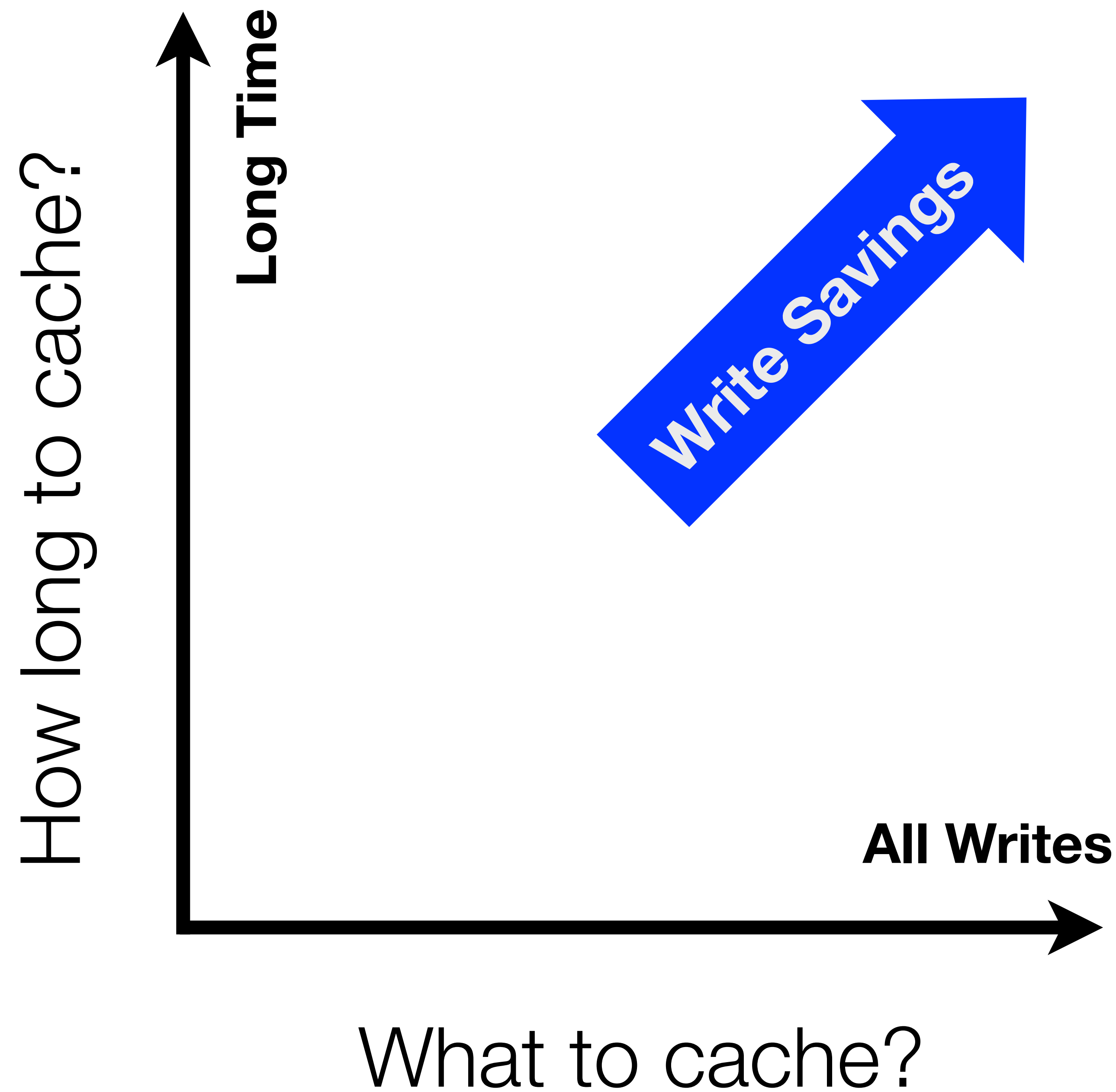


Tradeoff: Write Savings vs. Read Penalty

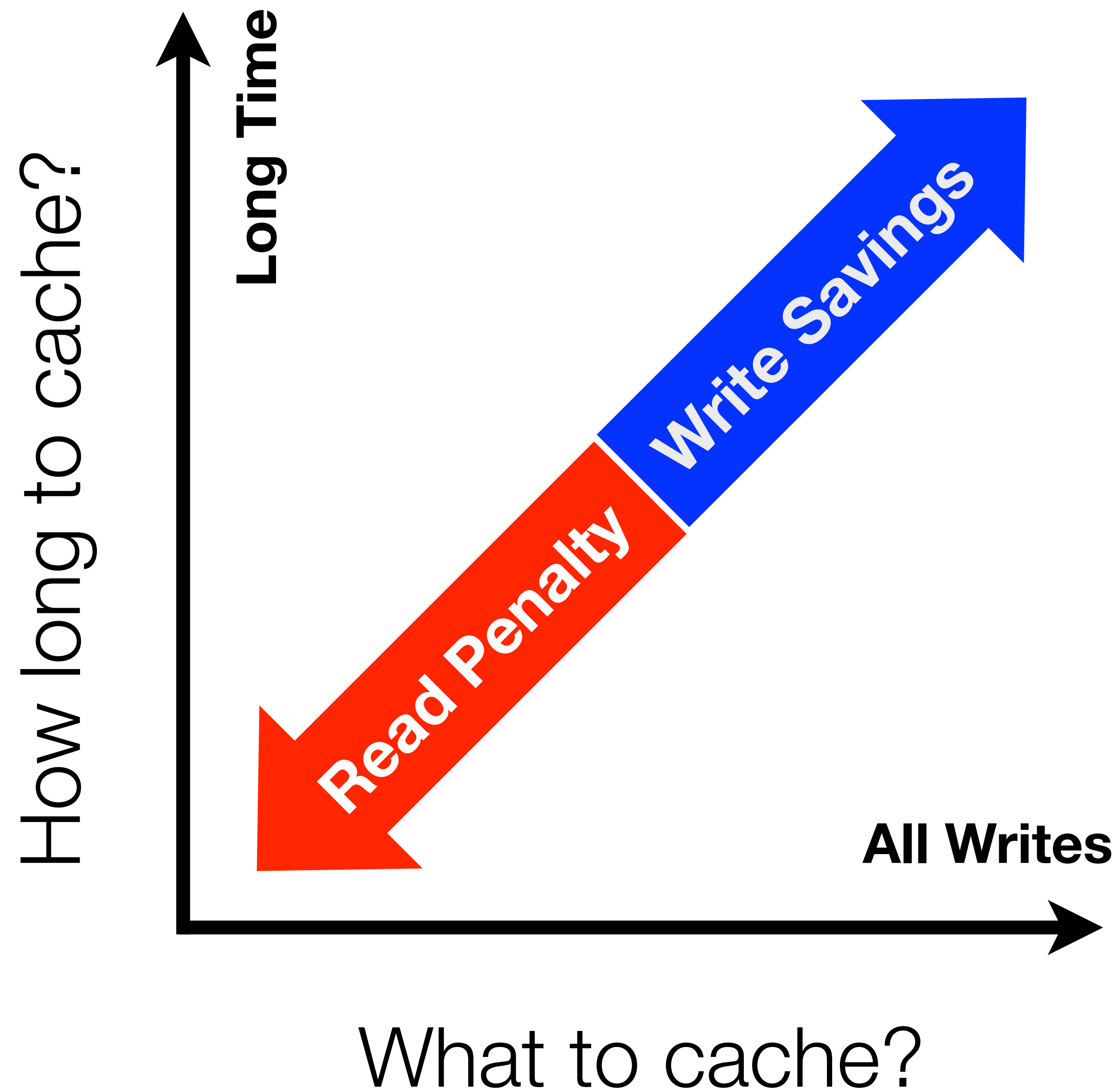
Tradeoff: Write Savings vs. Read Penalty



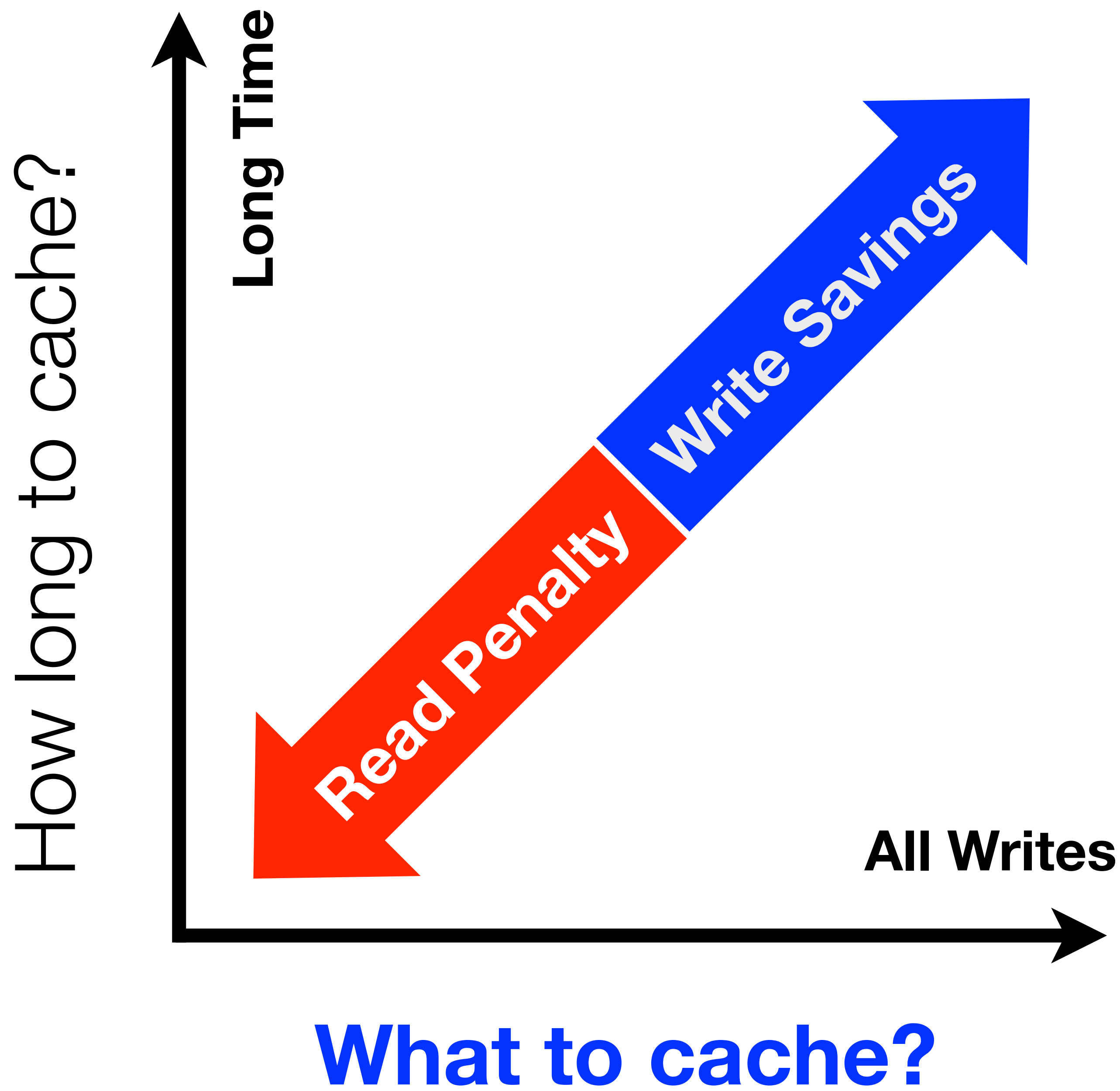
Tradeoff: Write Savings vs. Read Penalty



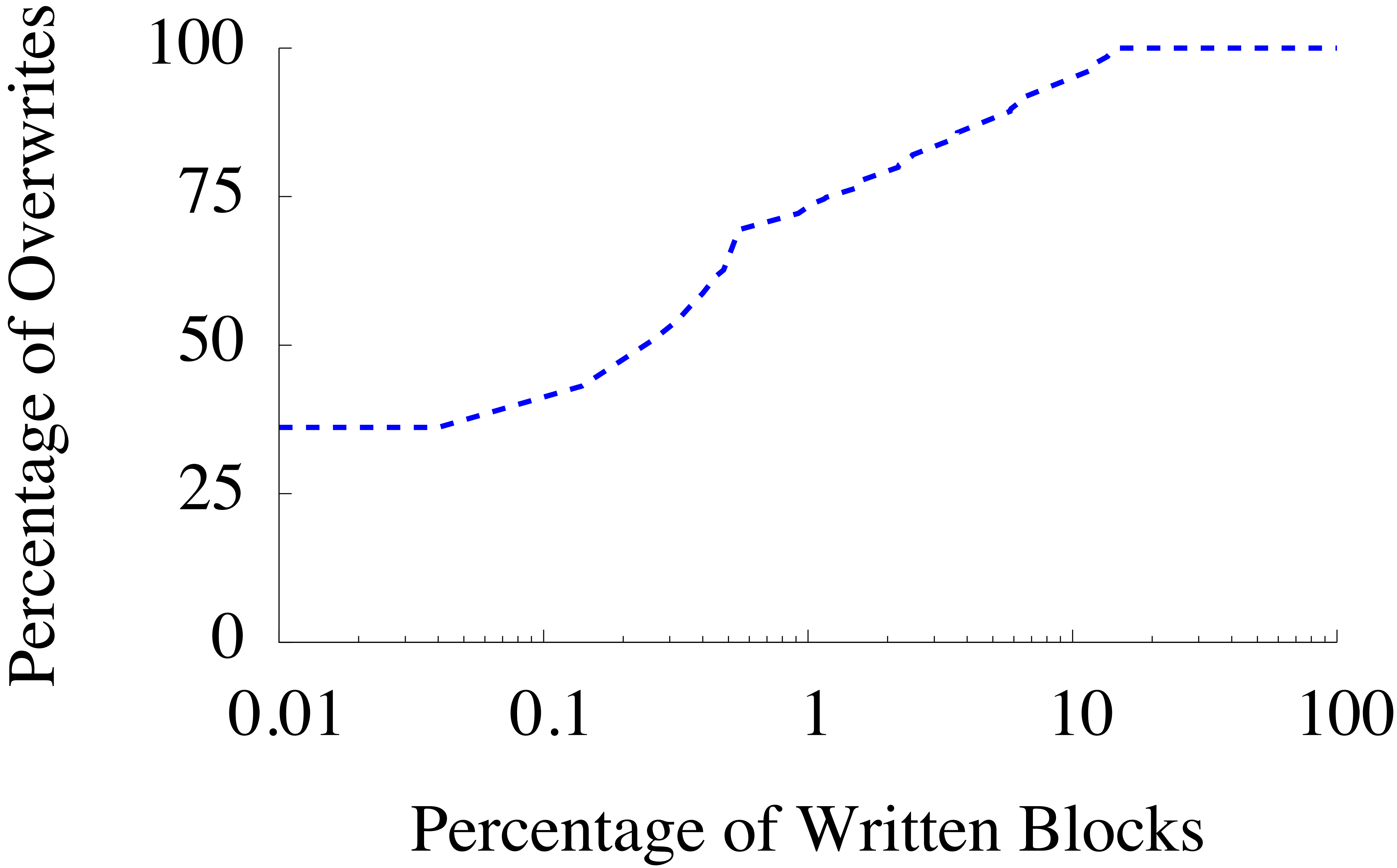
Tradeoff: Write Savings vs. Read Penalty



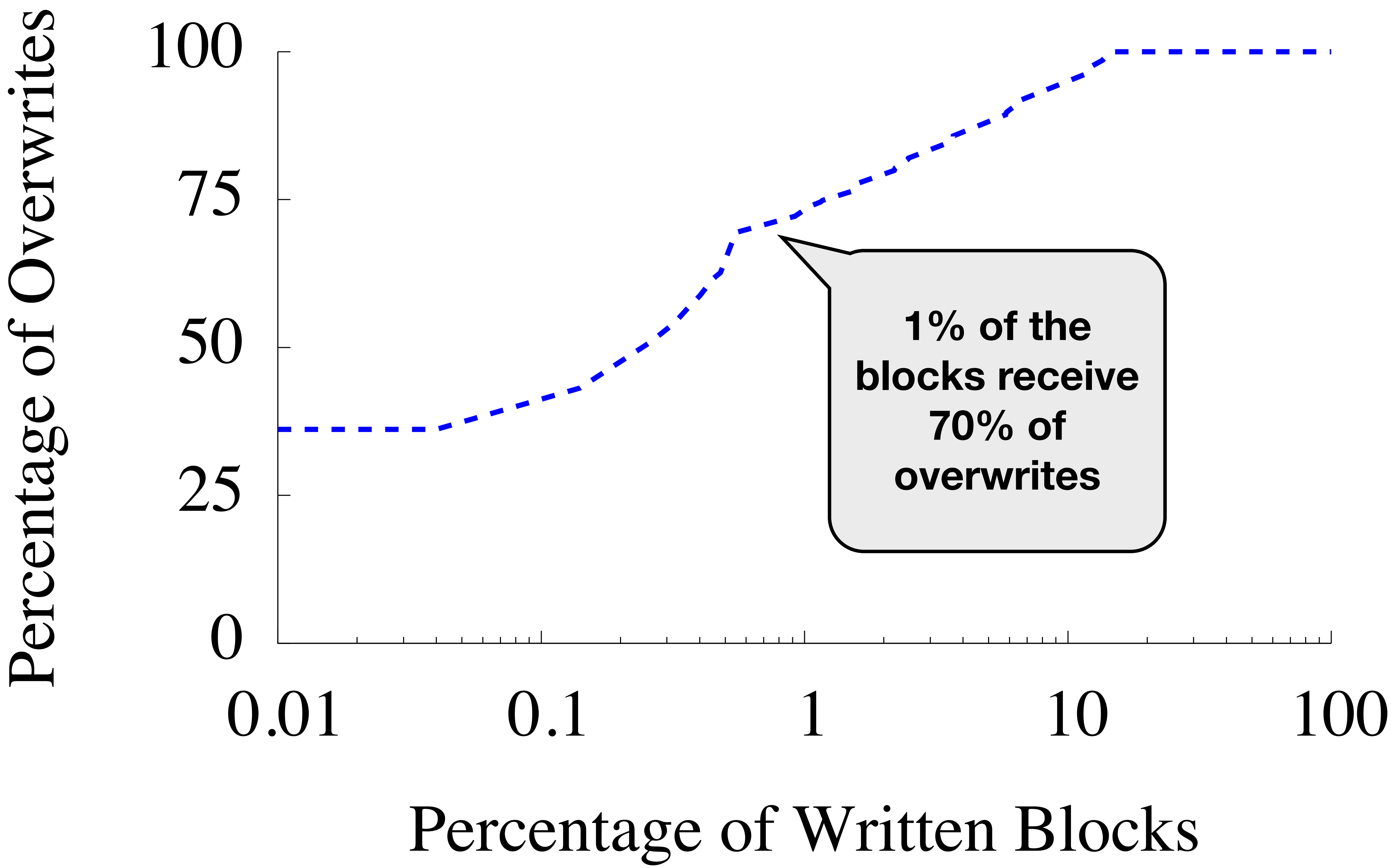
Tradeoff: Write Savings vs. Read Penalty



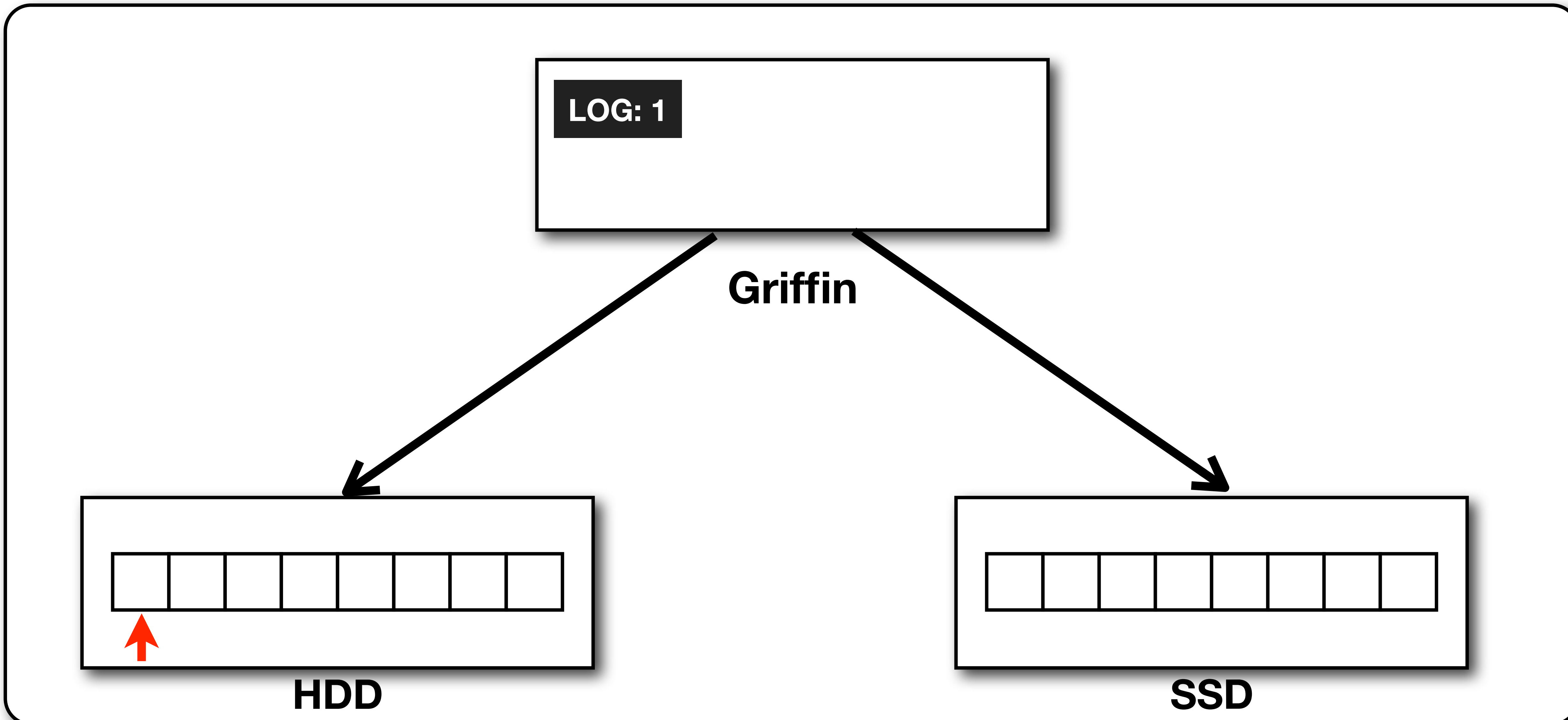
What to cache?: Overwrite Distribution



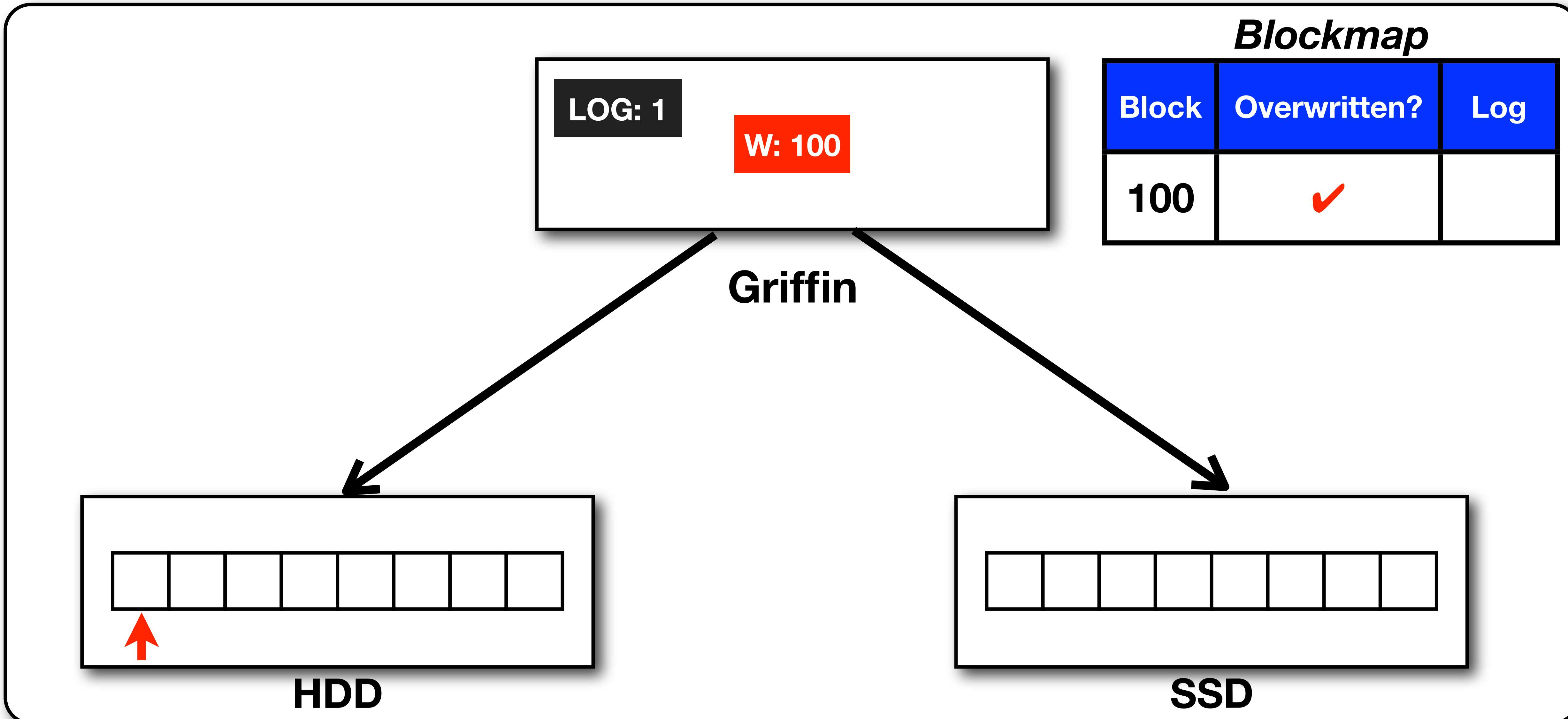
What to cache?: Overwrite Distribution



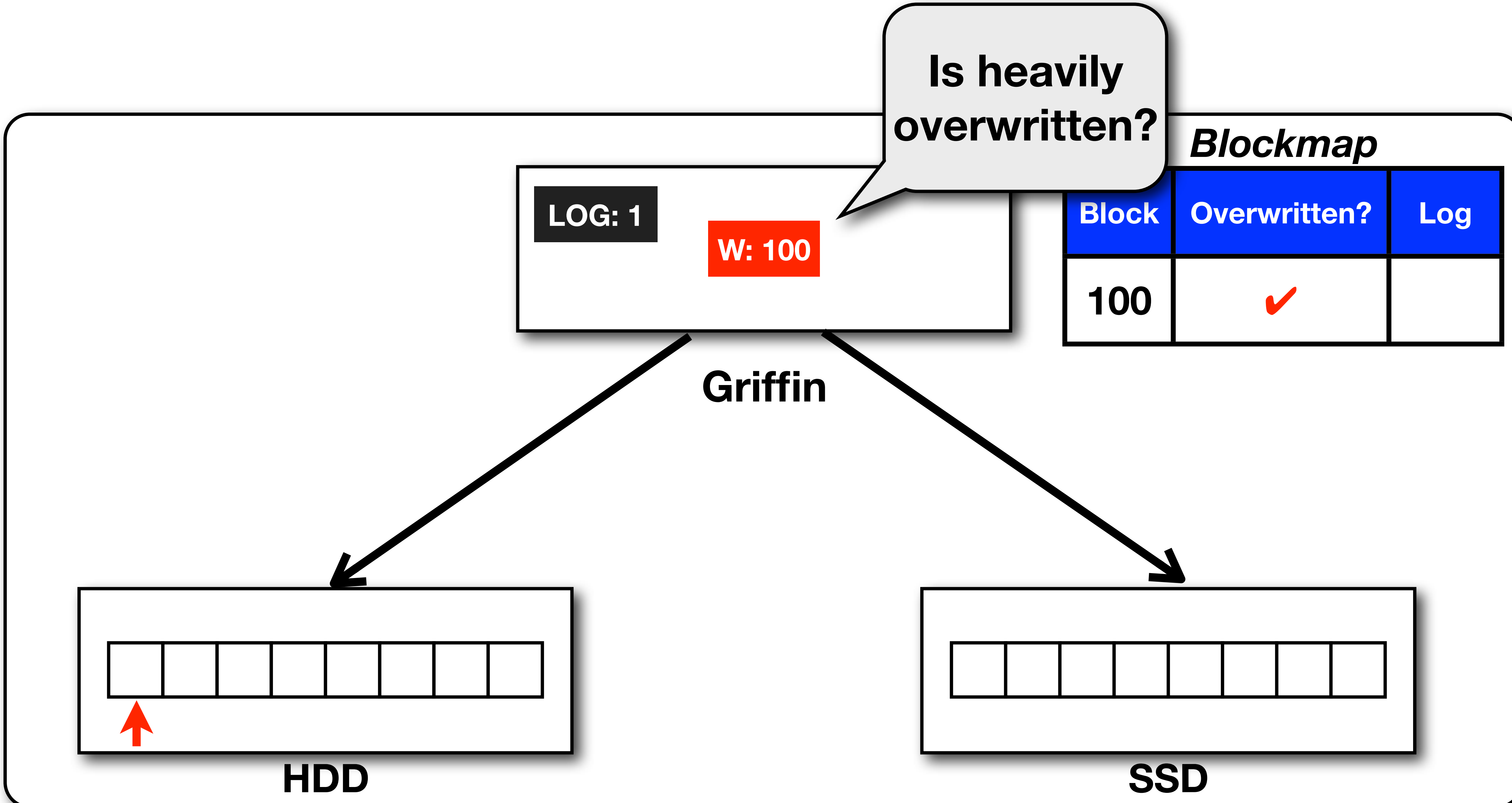
Selective Write-Caching



Selective Write-Caching

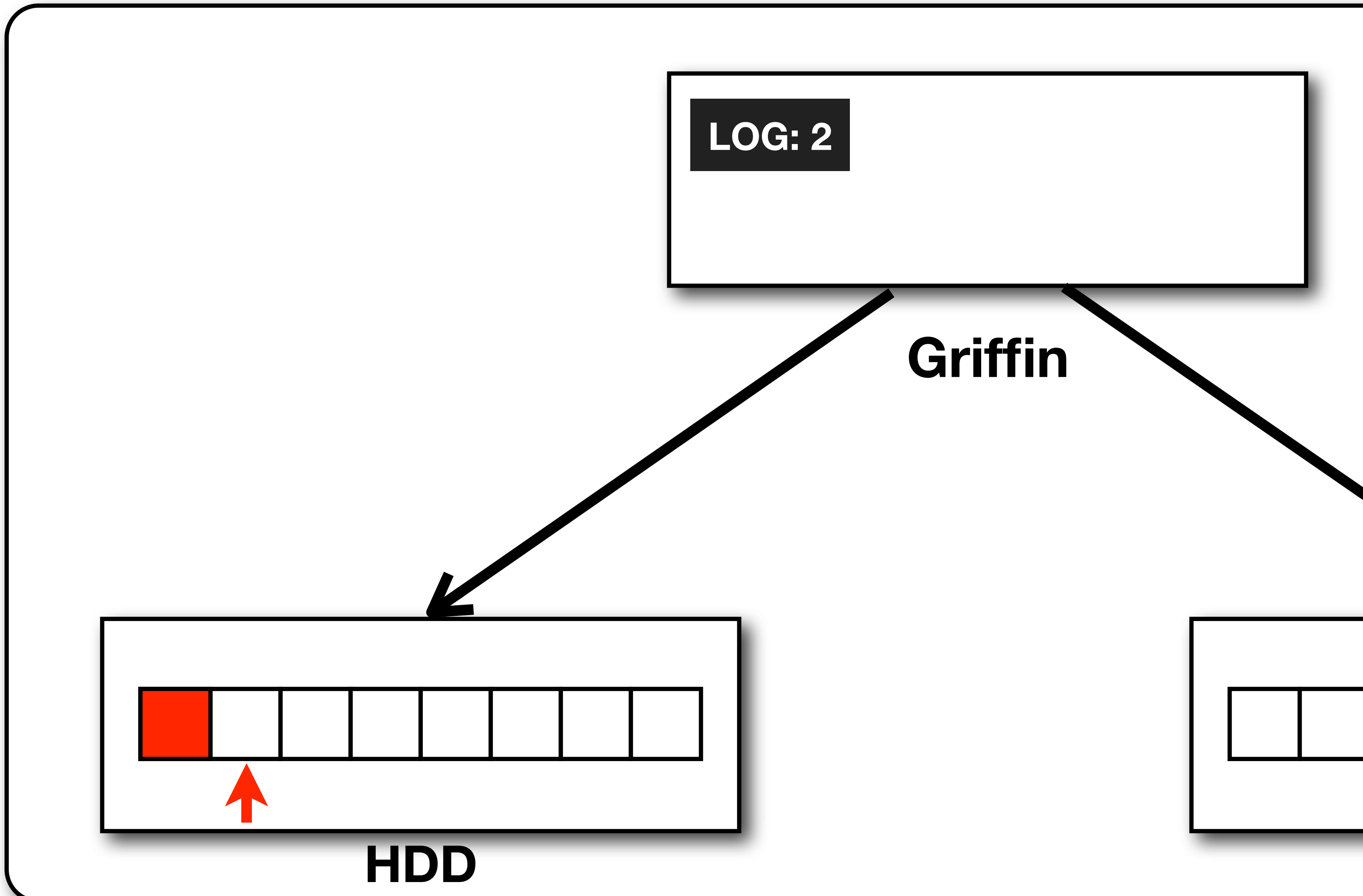


Selective Write-Caching



Block	Overwritten?	Log
100	✔	

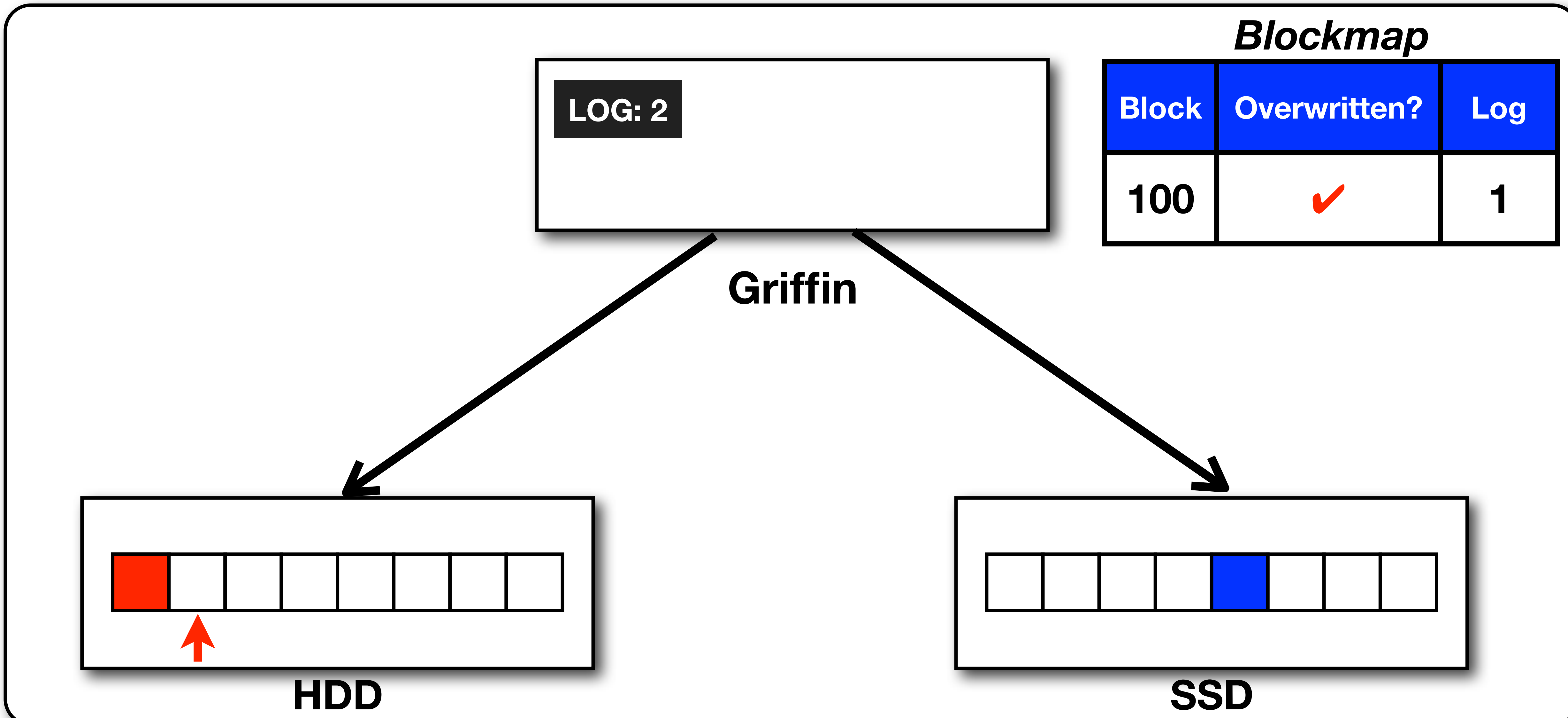
Selective Write-Caching



Blockmap

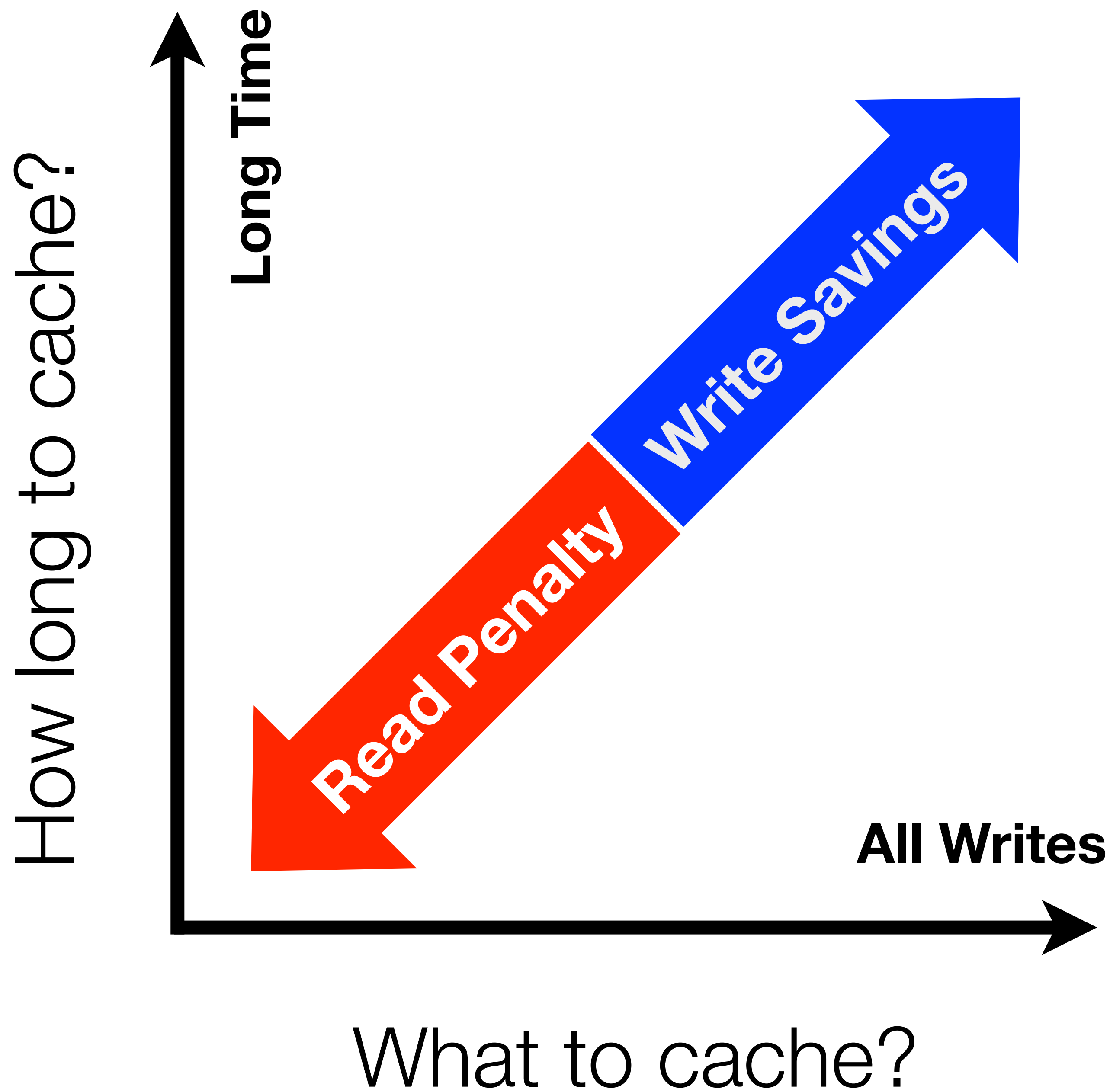
Block	Overwritten?	Log
100	✔	1

Selective Write-Caching

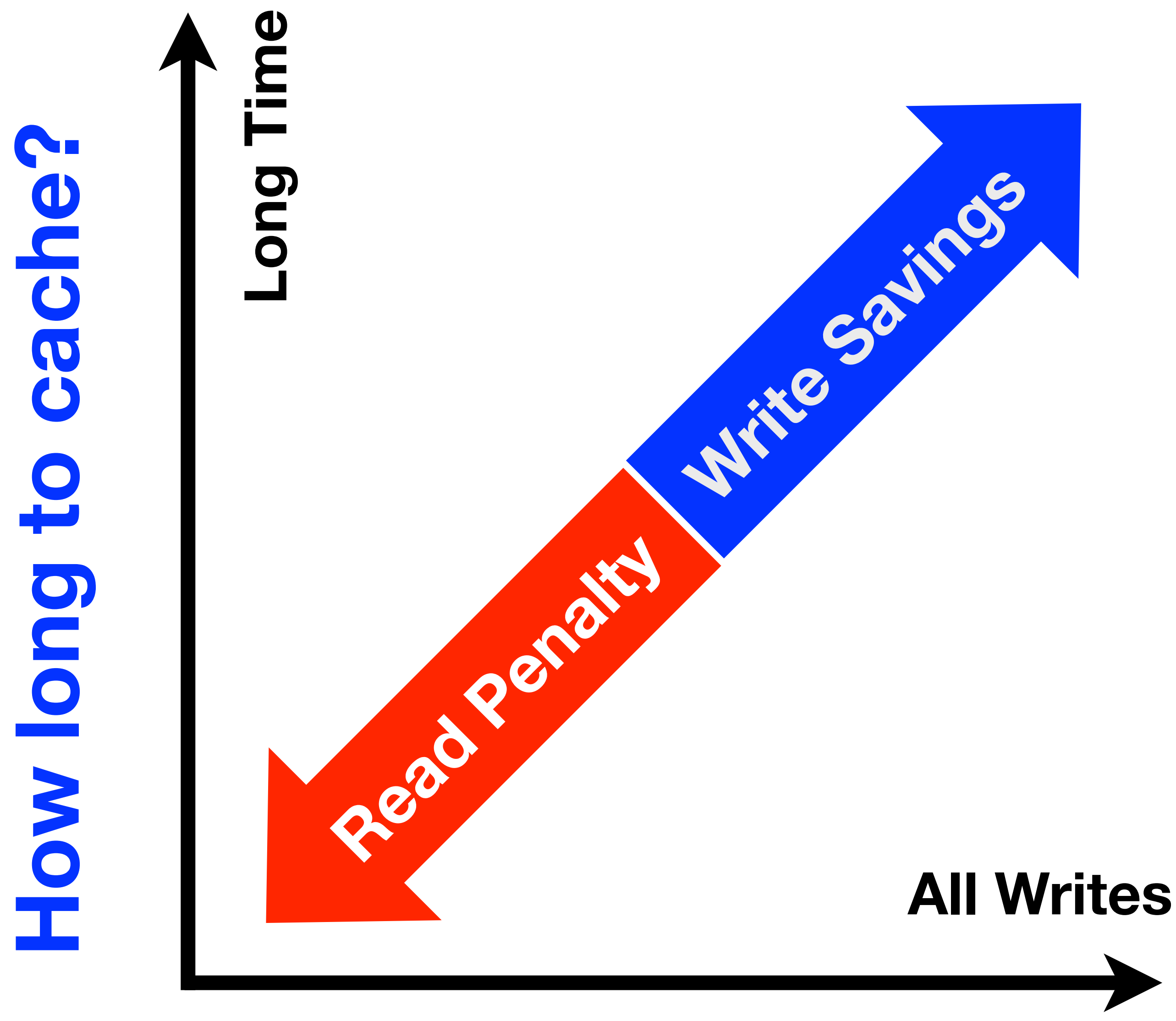


Block	Overwritten?	Log
100	✔	1

Tradeoff: Write Savings vs. Read Penalty

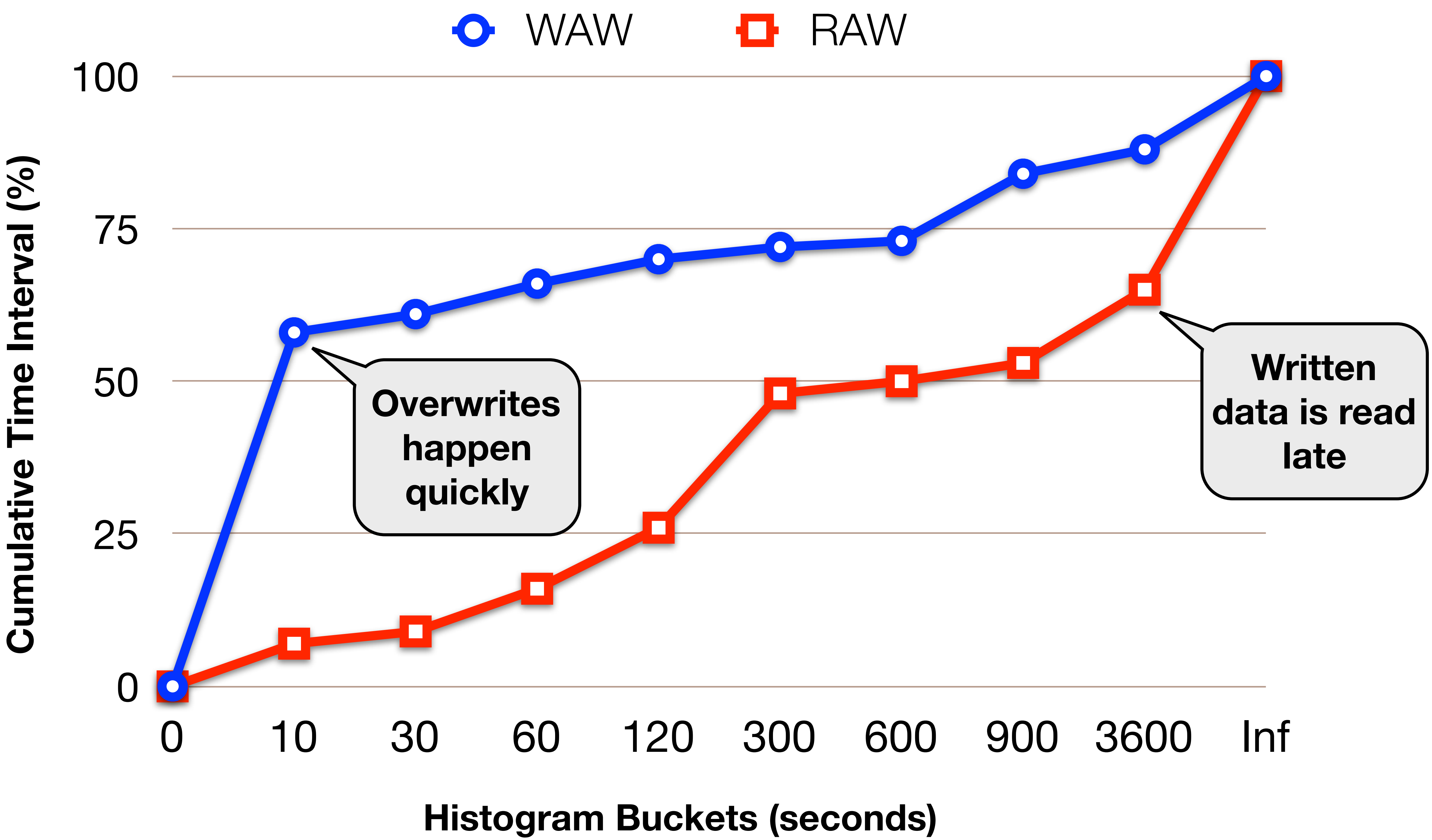


Tradeoff: Write Savings vs. Read Penalty

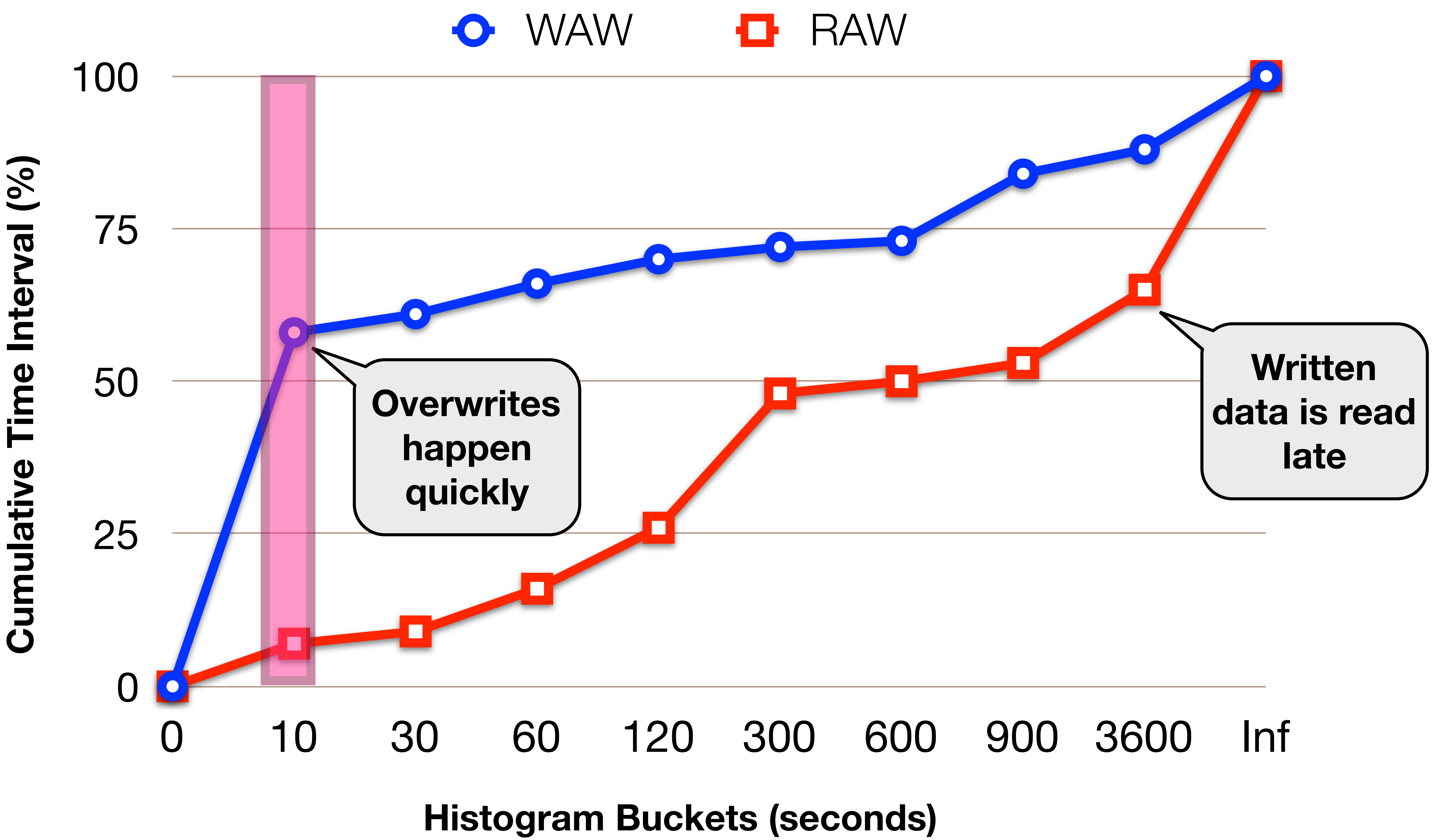


What to cache?

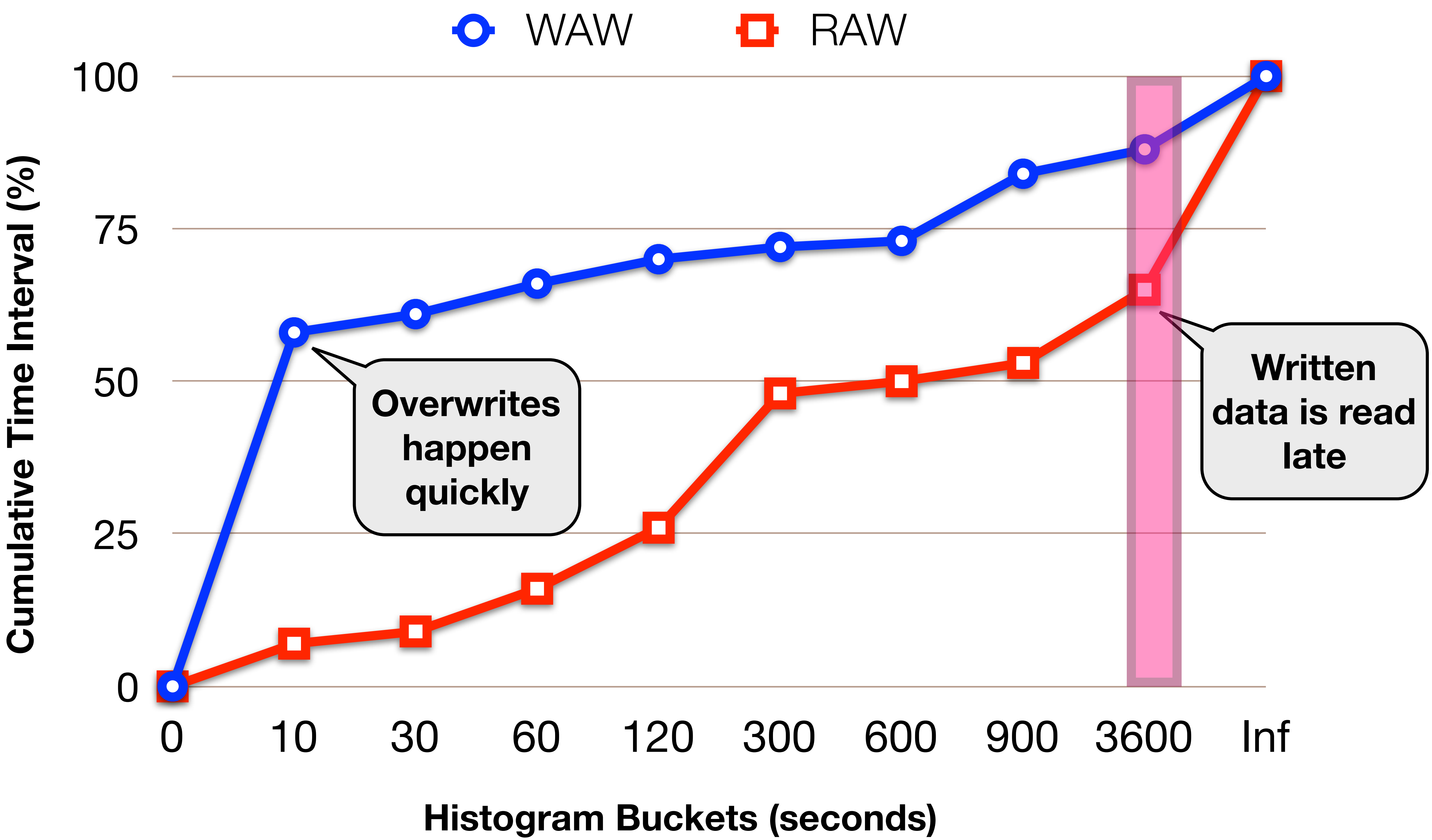
How long to cache?: Time Intervals



How long to cache?: Time Intervals



How long to cache?: Time Intervals



How long to cache?: Migration Triggers

How long to cache?: Migration Triggers

- **Timeout trigger**
  - ***Idea:*** *Fires after a certain time elapses*
  - Could have high read penalty

How long to cache?: Migration Triggers

- **Timeout trigger**
  - ***Idea:*** *Fires after a certain time elapses*
  - Could have high read penalty
- **Read-threshold trigger**
  - ***Idea:*** *Fires if HDD reads exceeds threshold since last migration*
  - Bounds HDD read fraction
  - Could have no migrations for a long time

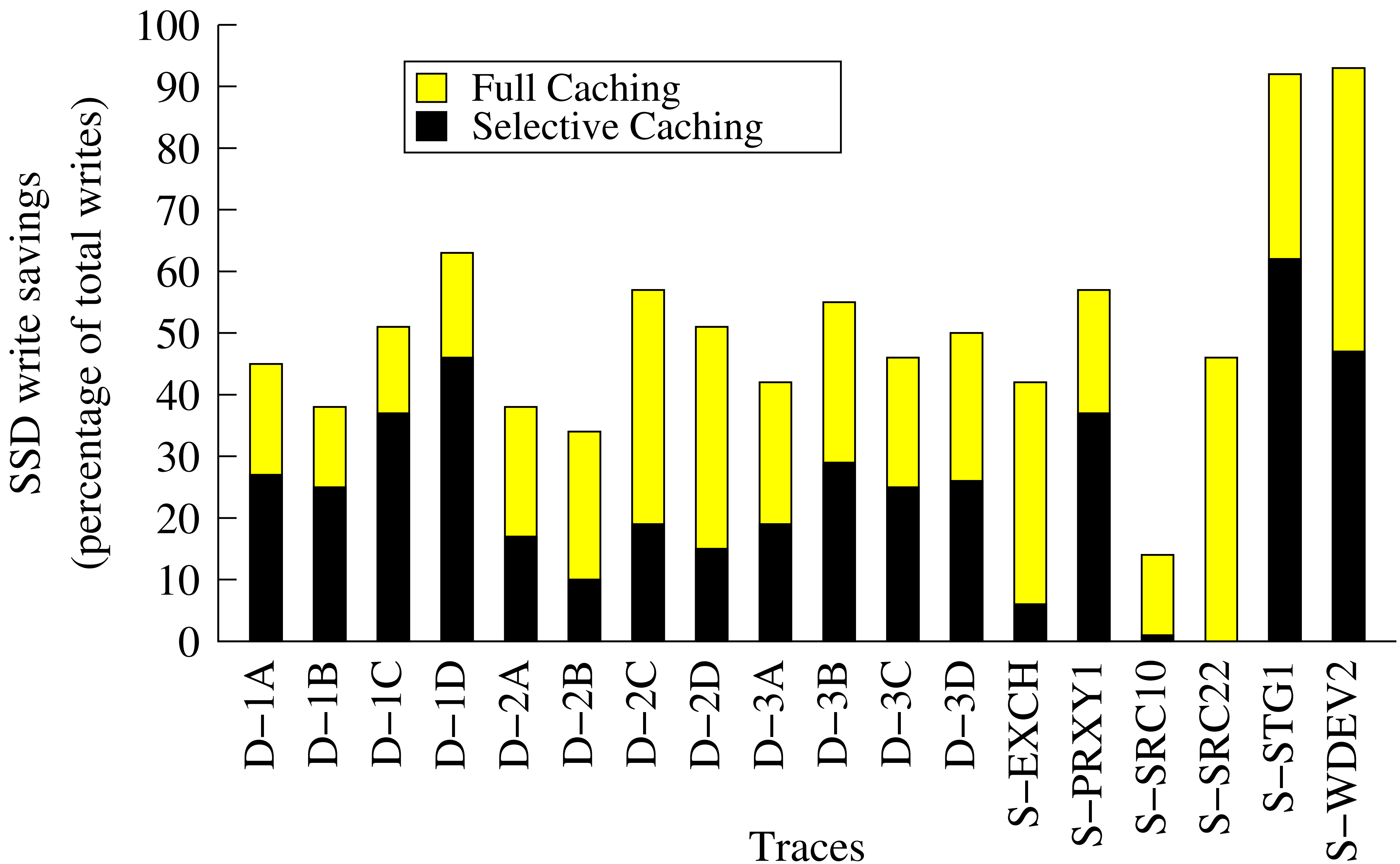
How long to cache?: Migration Triggers

- **Timeout trigger**
  - ***Idea:*** *Fires after a certain time elapses*
  - Could have high read penalty
- **Read-threshold trigger**
  - ***Idea:*** *Fires if HDD reads exceeds threshold since last migration*
  - Bounds HDD read fraction
  - Could have no migrations for a long time
- **Hybrid trigger**
  - Every 15 mins or read penalty > 5%

Outline

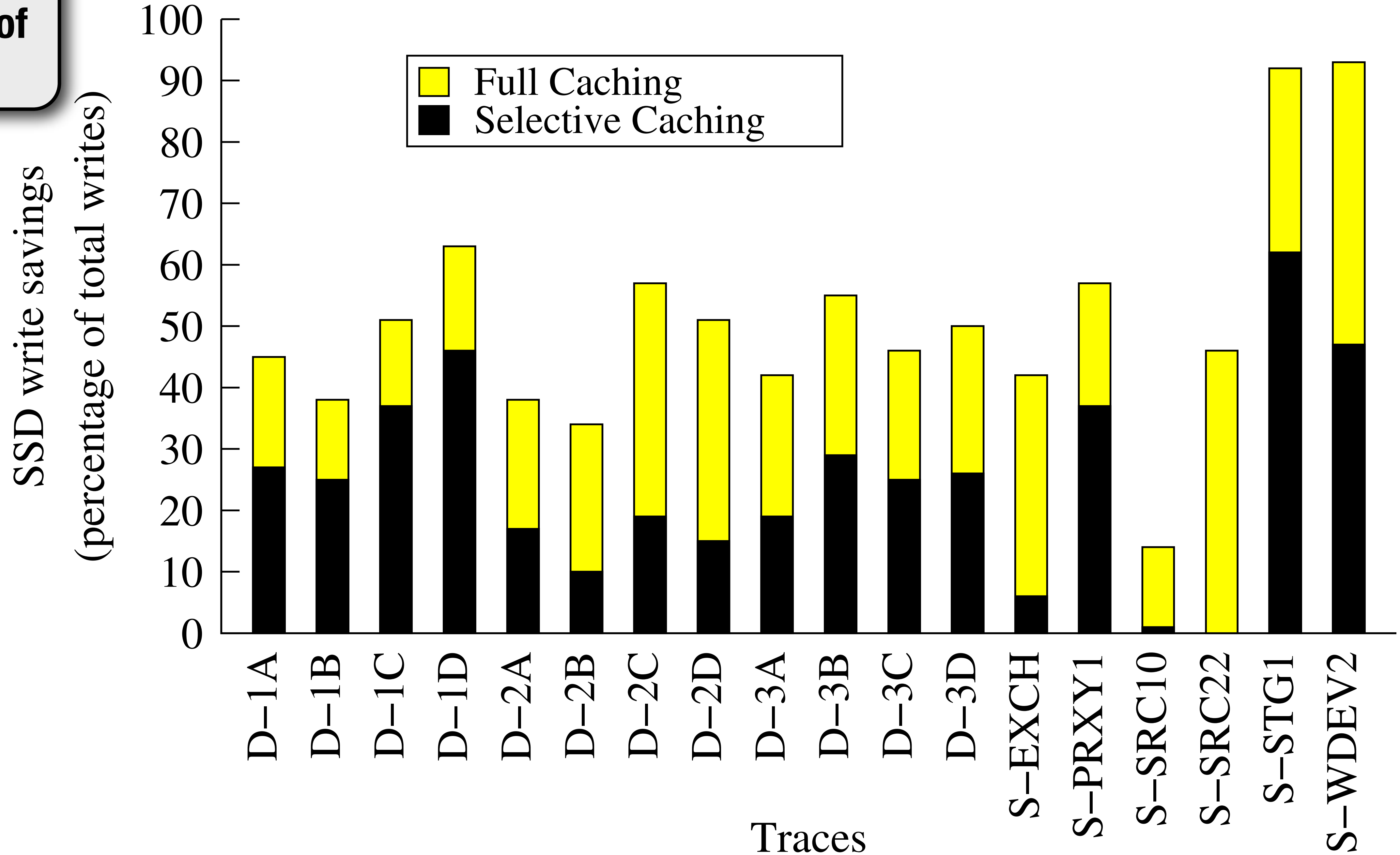
- **Motivating Workload Characteristics**
- **Disk-based Write Caching**
- **Experimental Evaluation**
  - Caching Policies: *What to cache?*
  - Migration Policies: *How long to cache?*
  - Performance: *Lifetime and Latency*
- **Conclusions**

What to Cache?: Write Savings



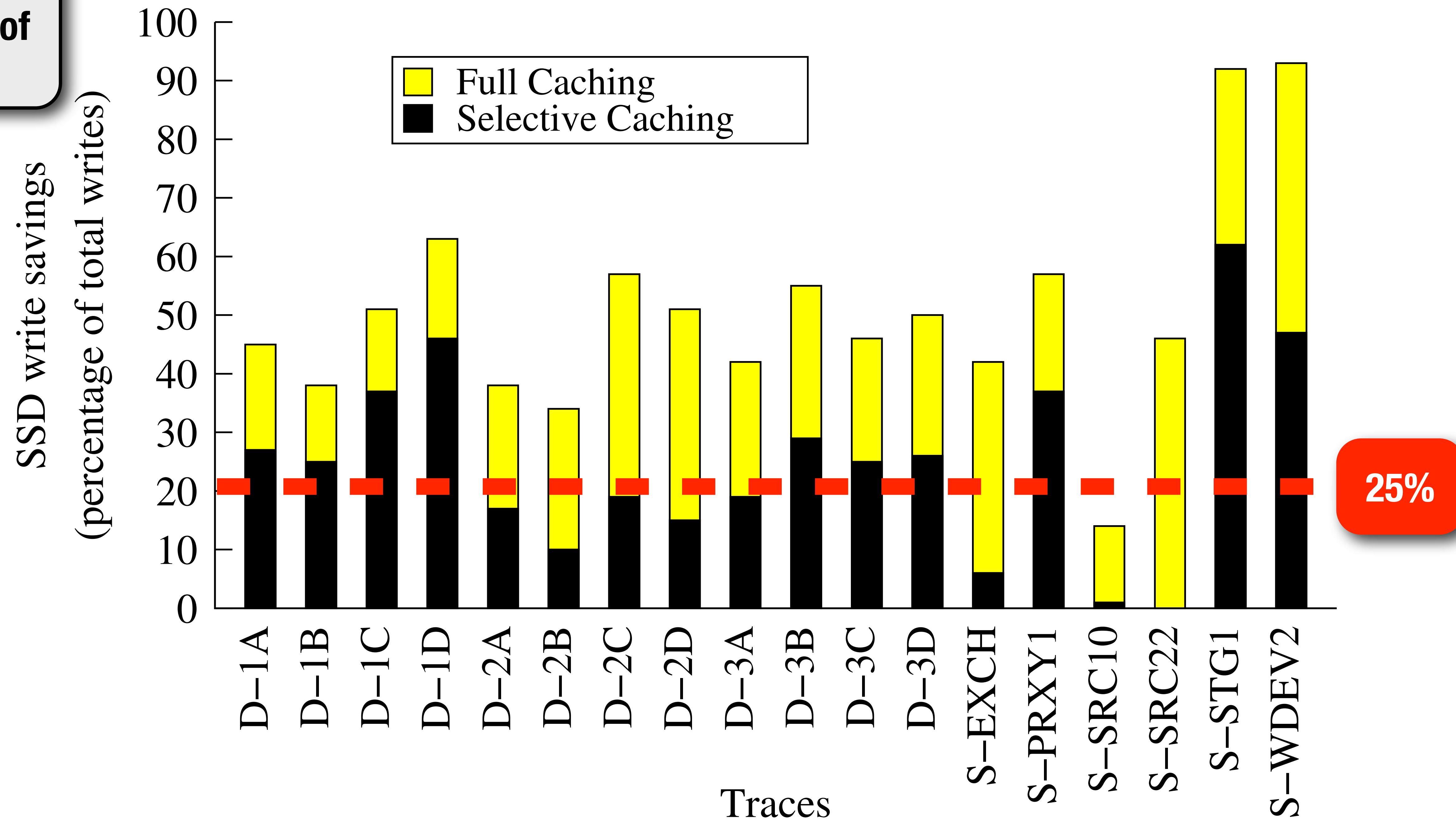
What to Cache?: Write Savings

Migrated at the end of trace



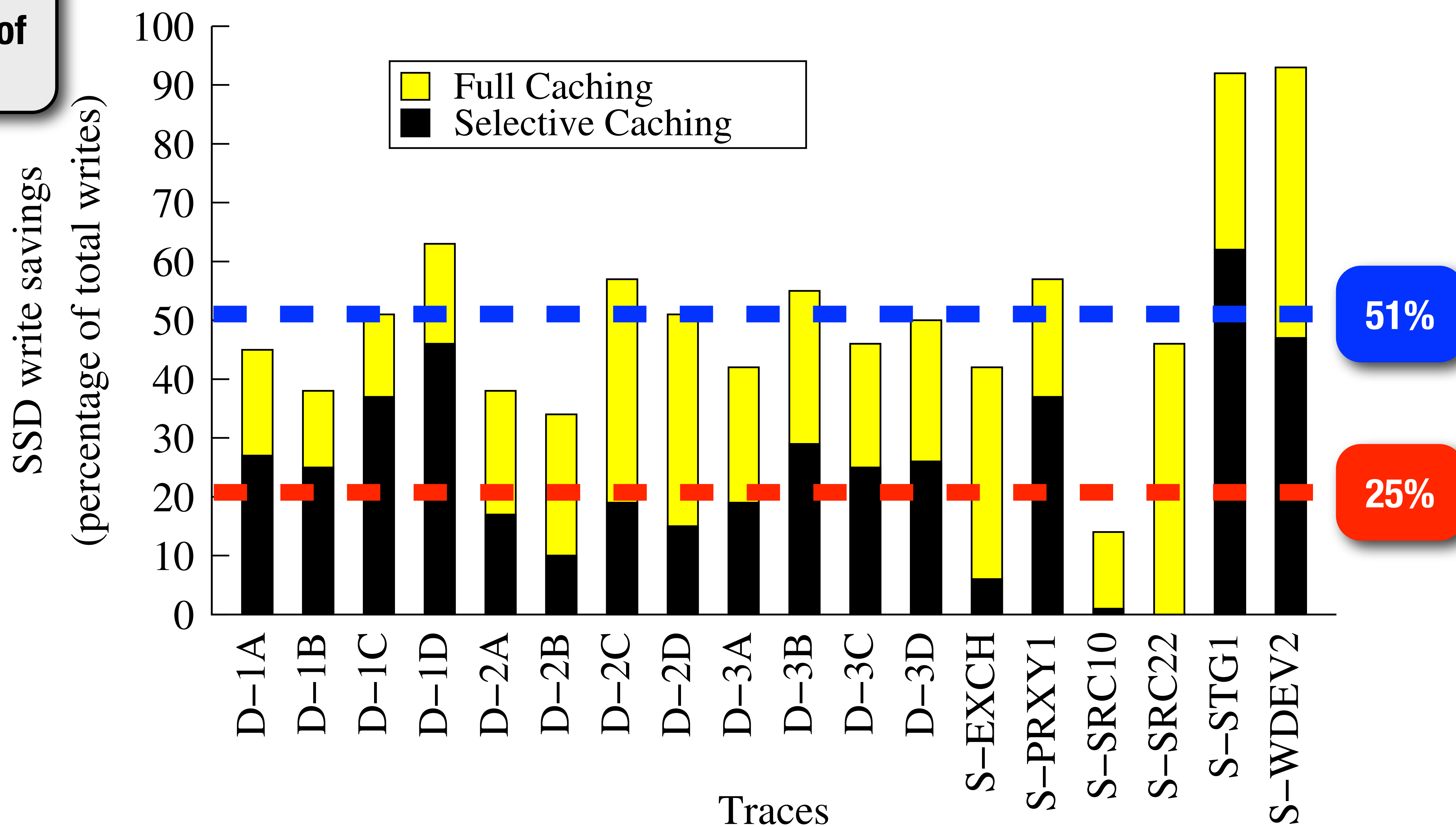
What to Cache?: Write Savings

Migrated at the end of trace

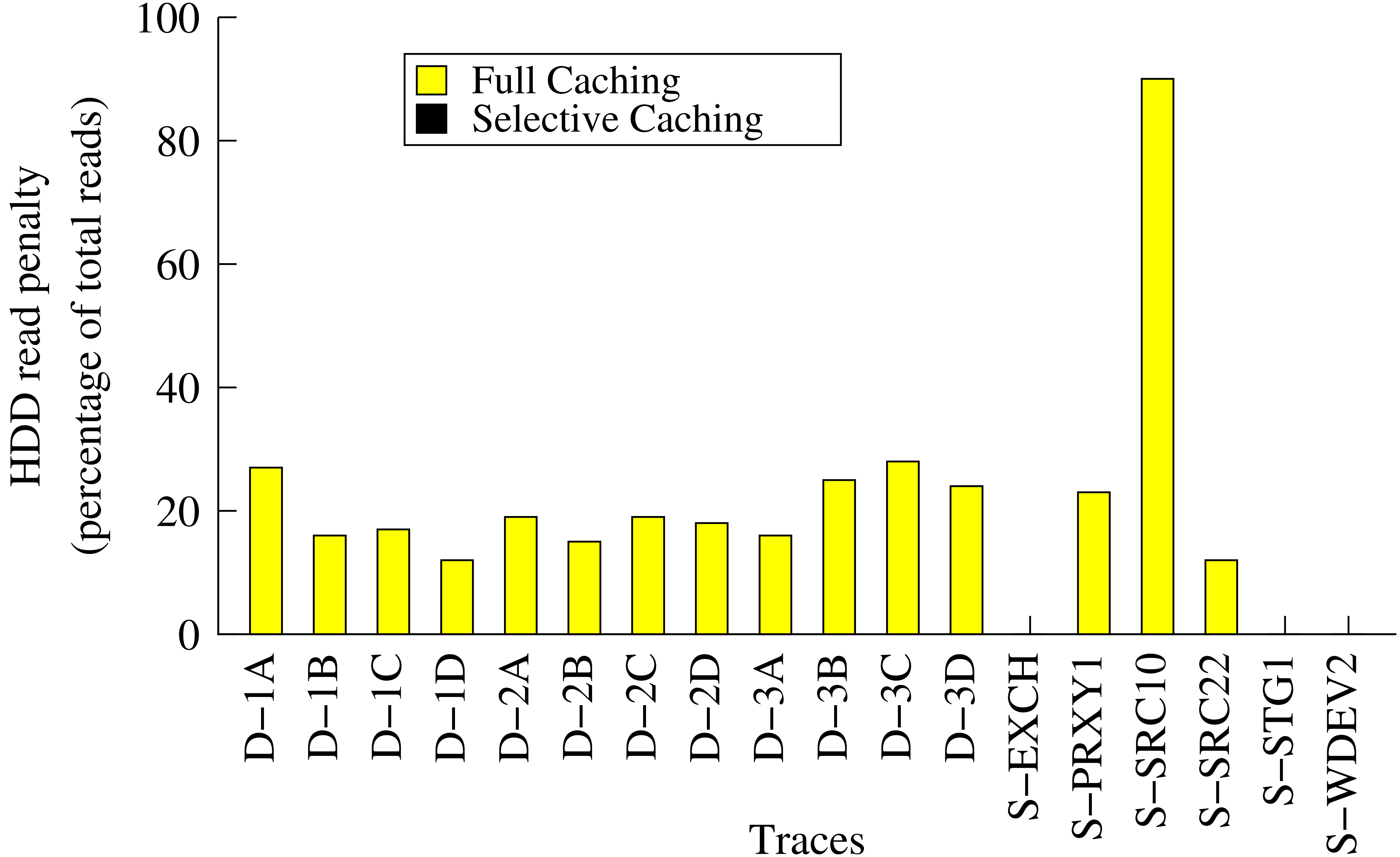


What to Cache?: Write Savings

Migrated at the end of trace

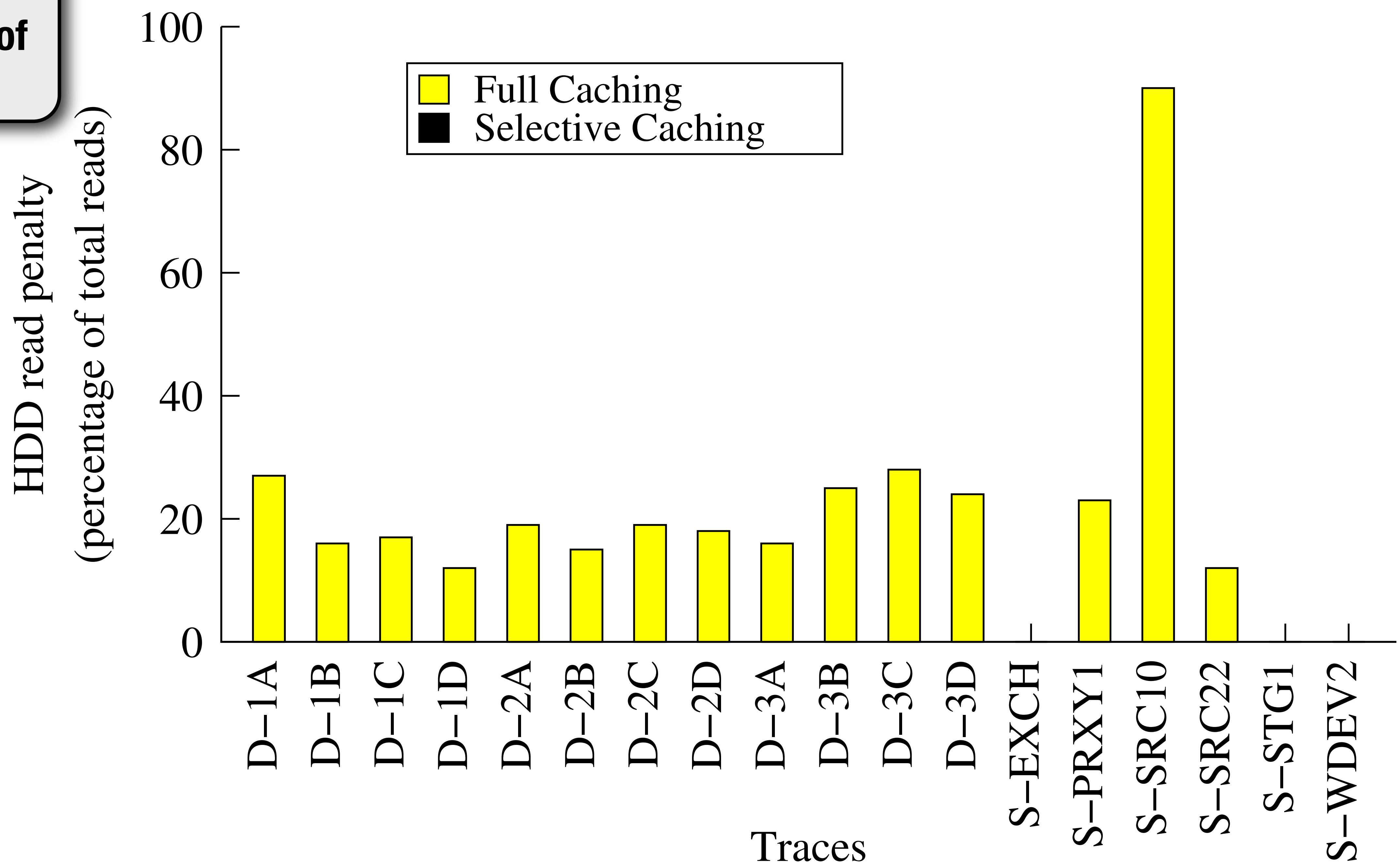


What to Cache?: Read Penalty

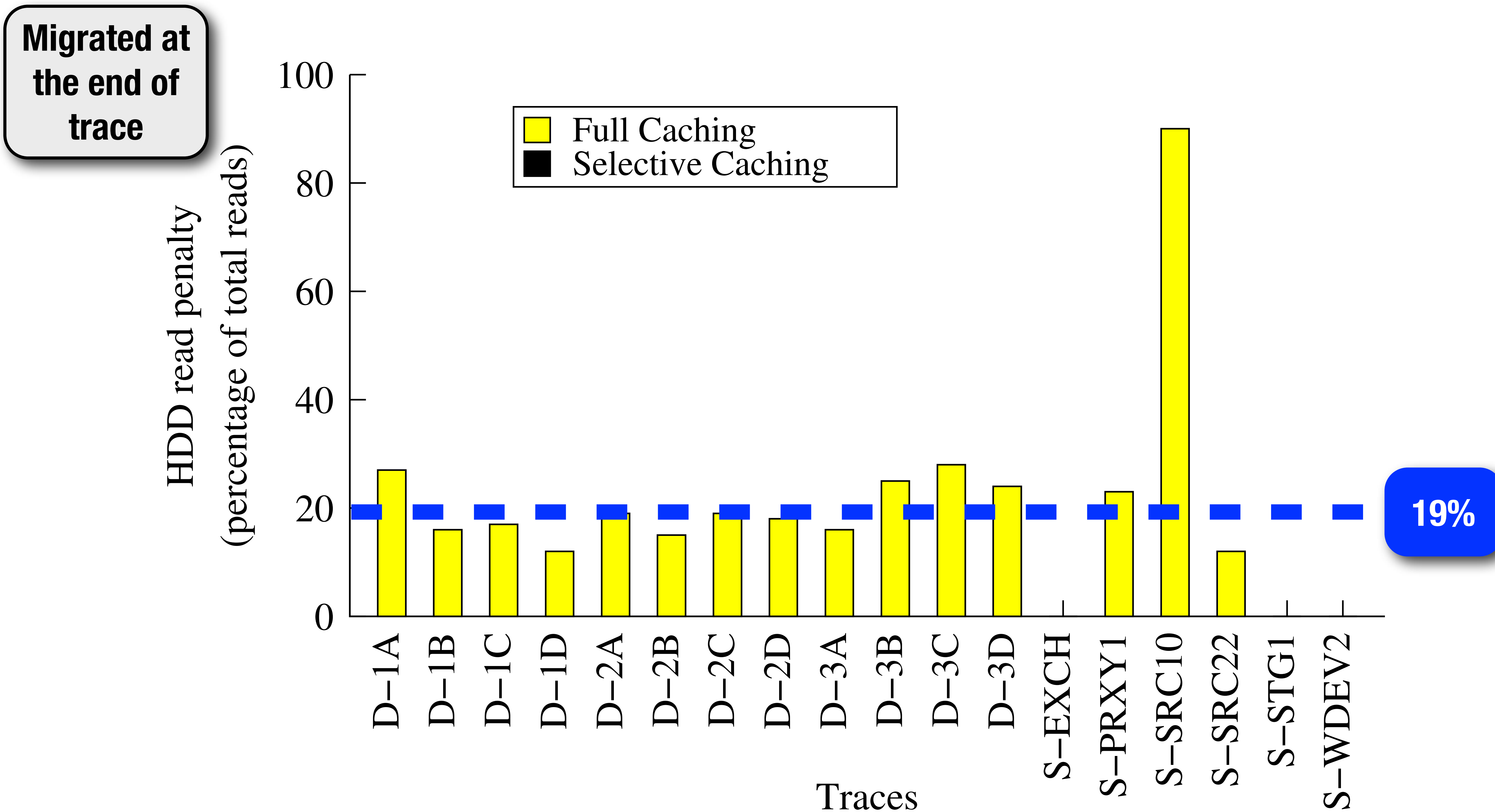


What to Cache?: Read Penalty

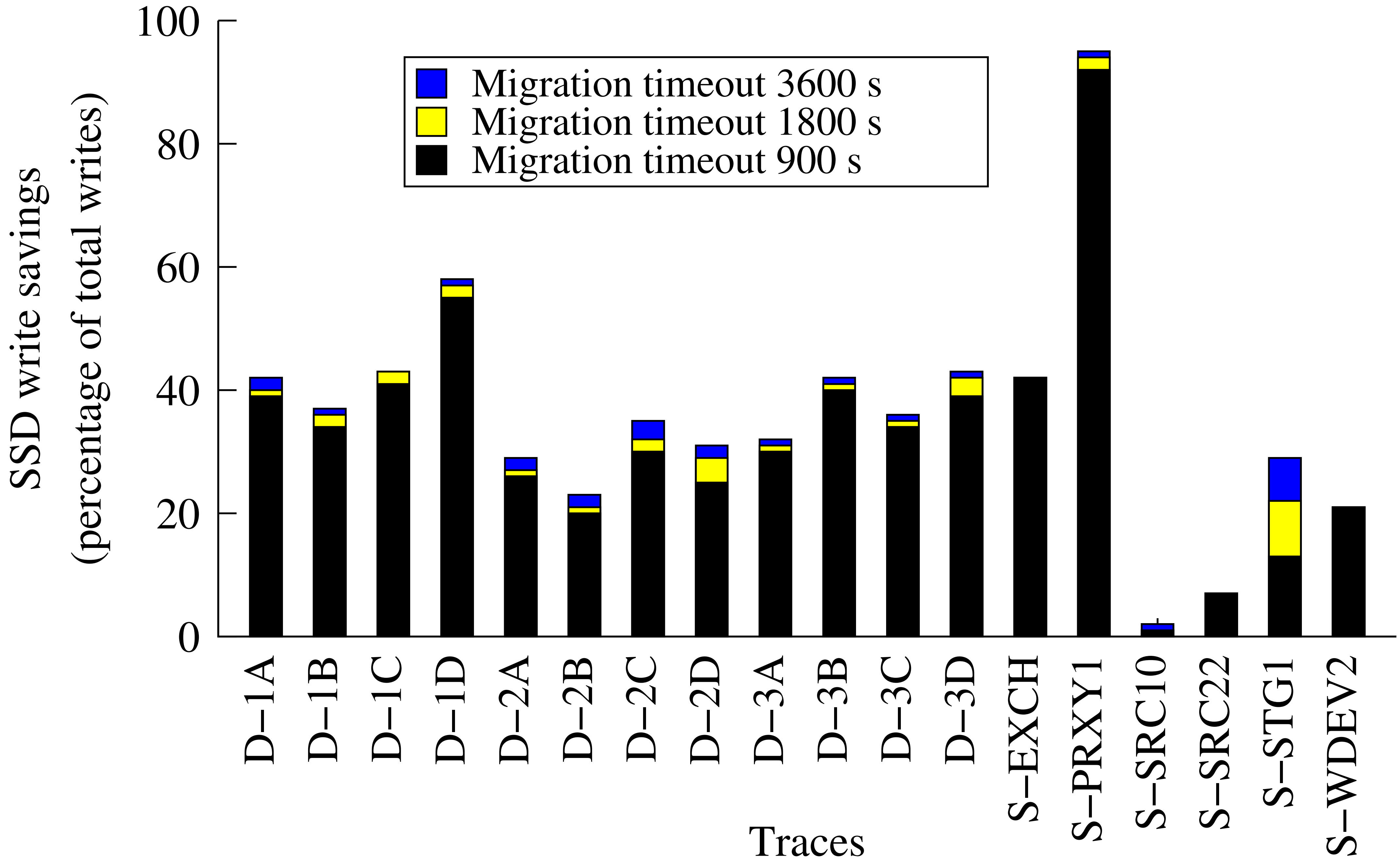
Migrated at the end of trace



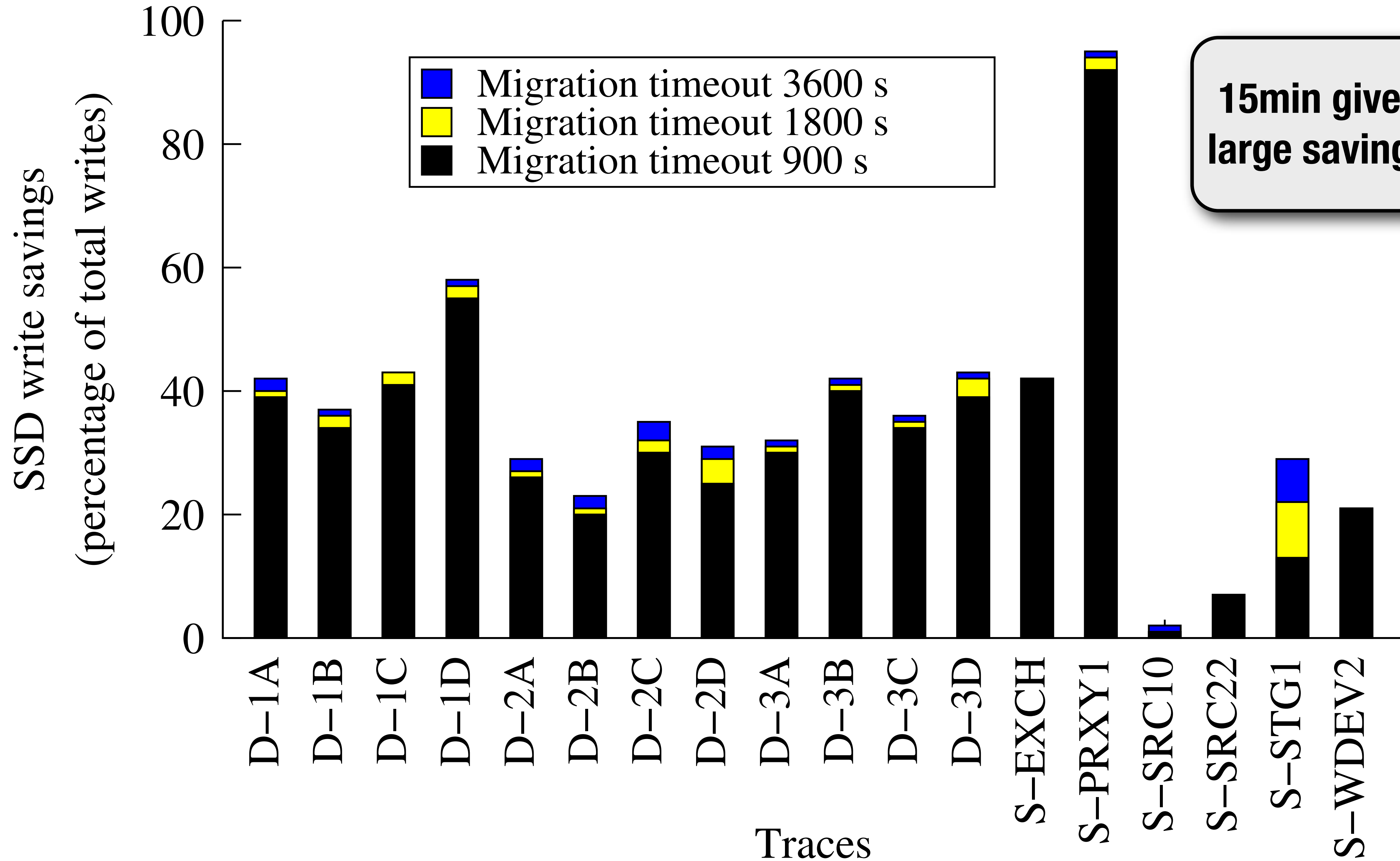
What to Cache?: Read Penalty



How Long to Cache?: Write Savings

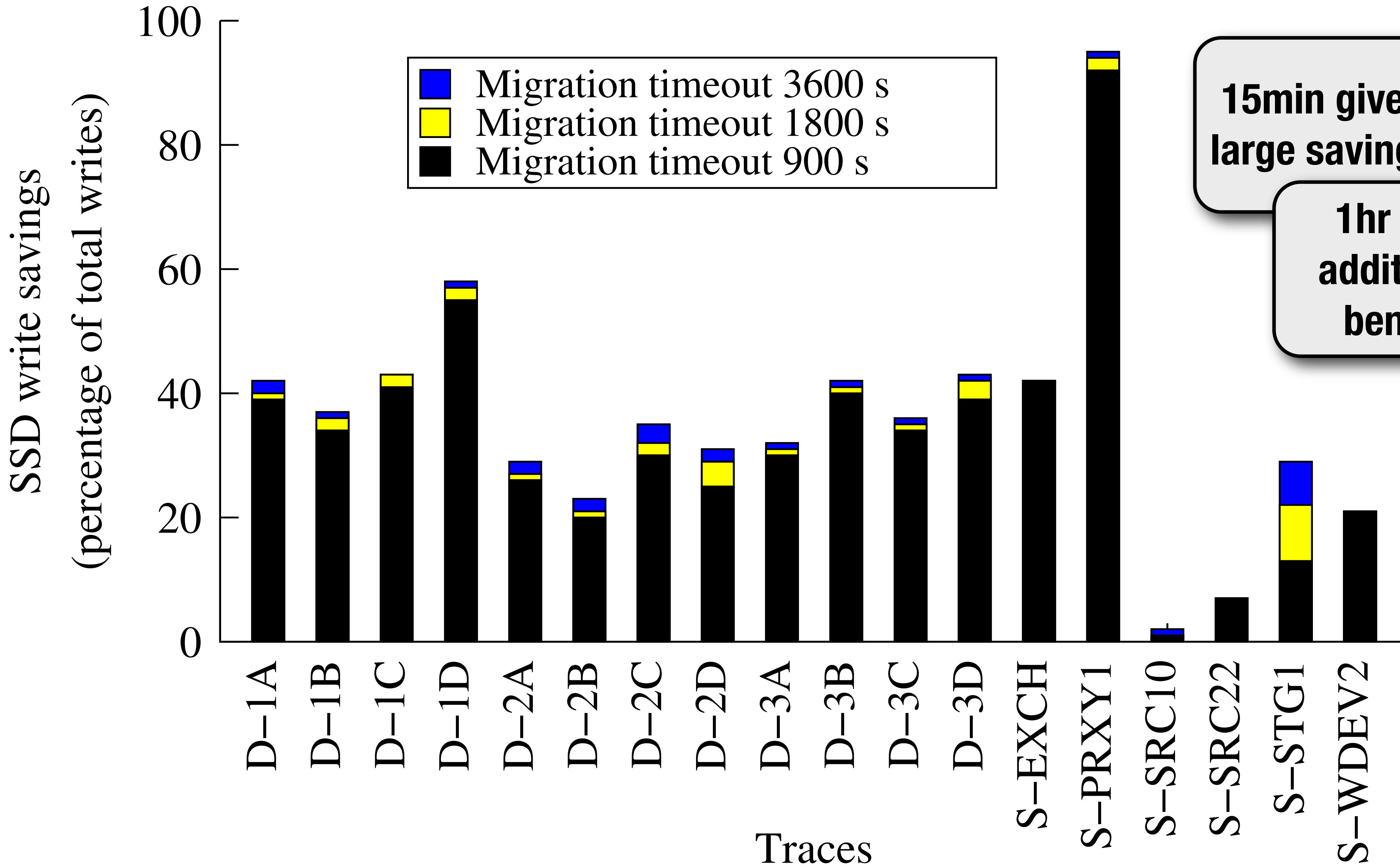


How Long to Cache?: Write Savings



15min gives large savings

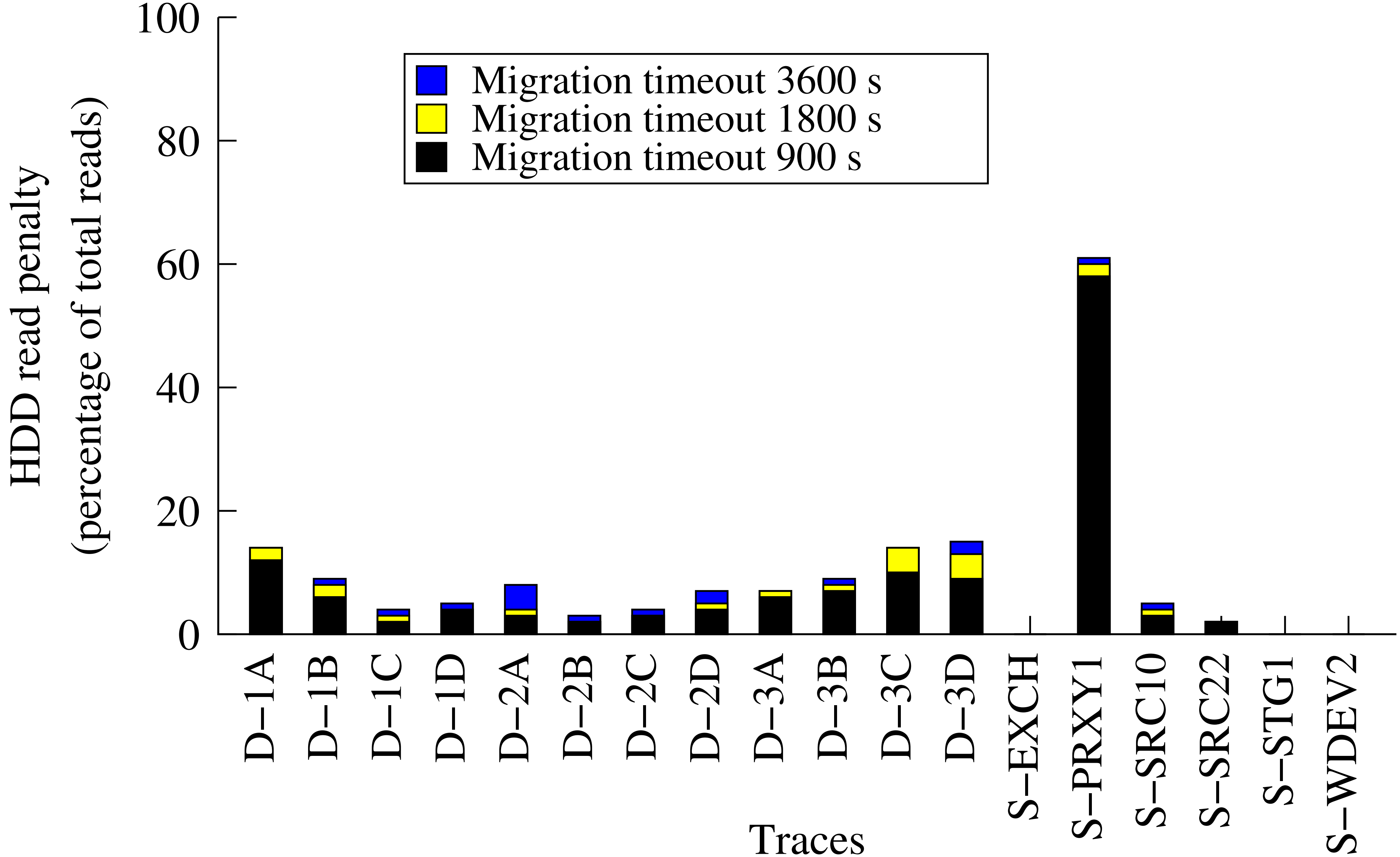
How Long to Cache?: Write Savings



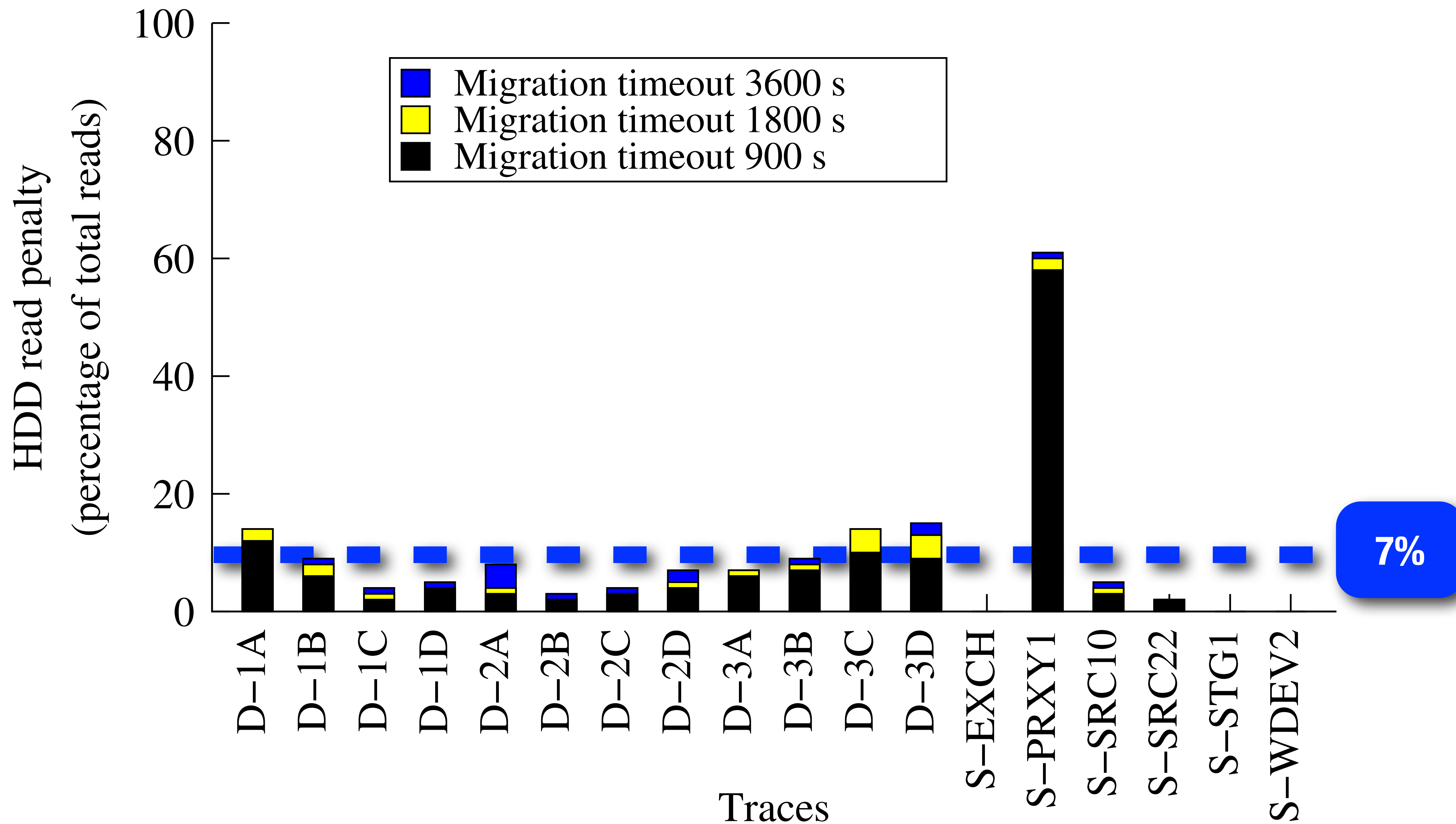
15min gives large savings

1hr less additional benefit

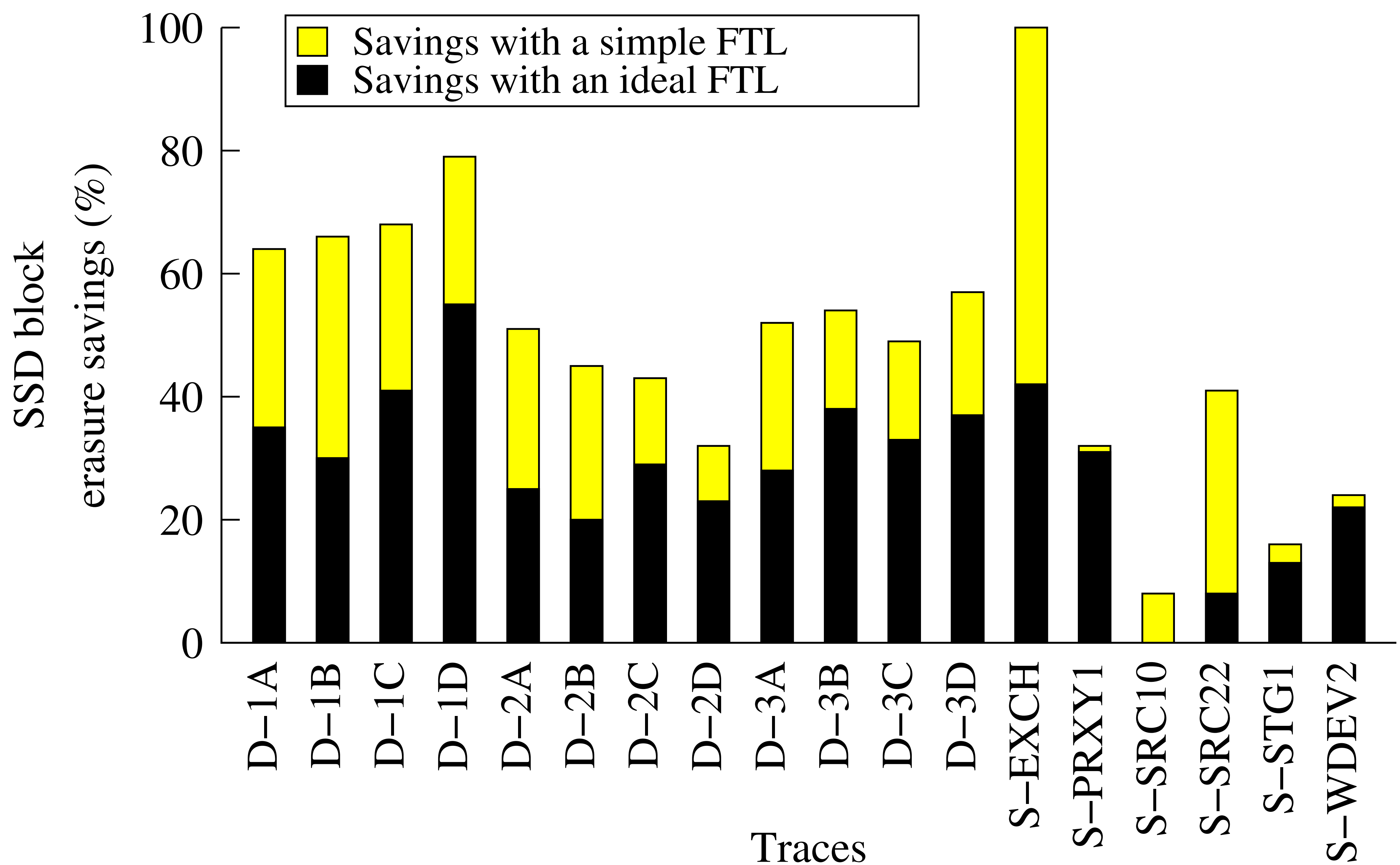
How Long to Cache?: Read Penalty



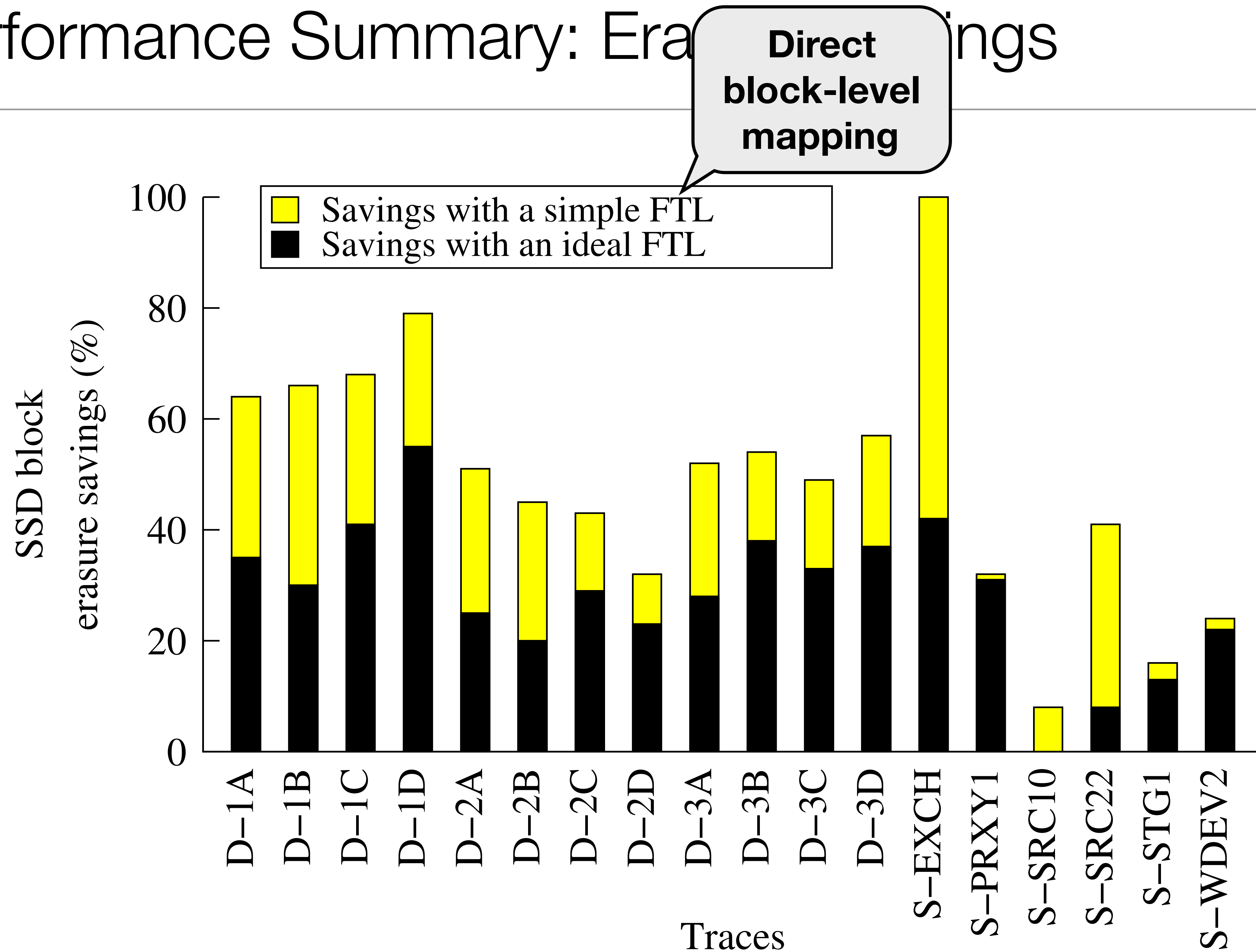
How Long to Cache?: Read Penalty



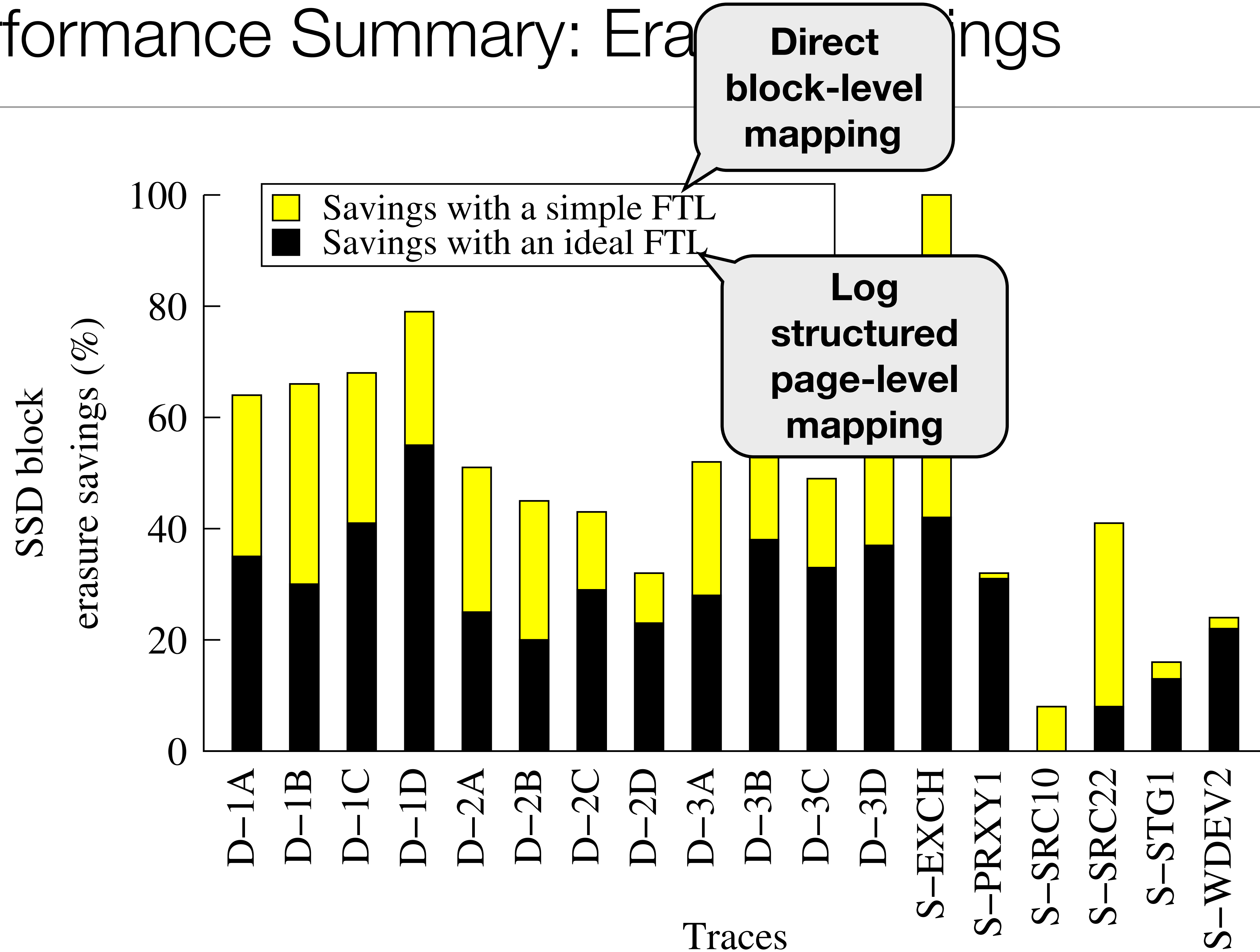
Performance Summary: Erasure Savings



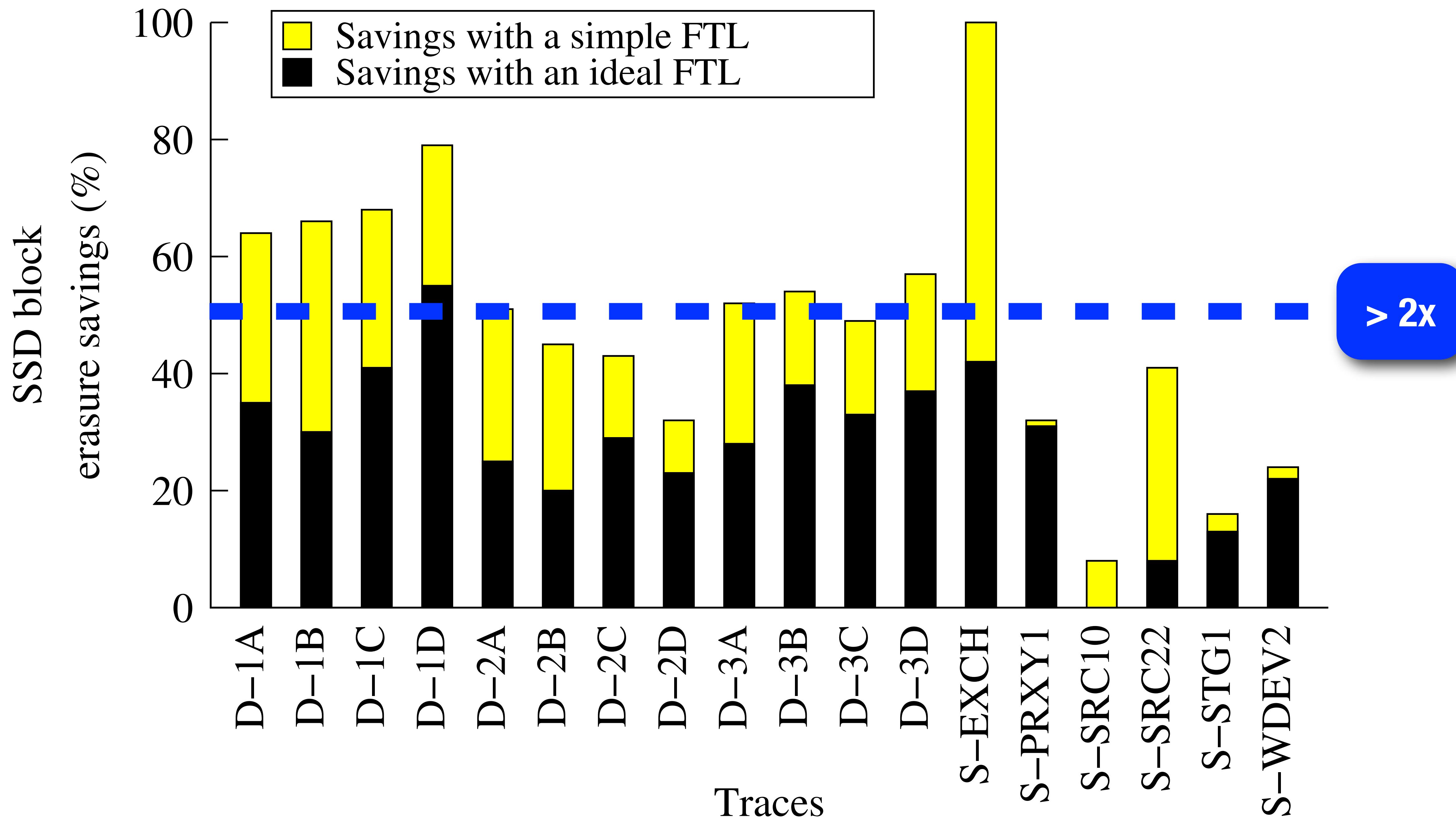
Performance Summary: Era[...]ings



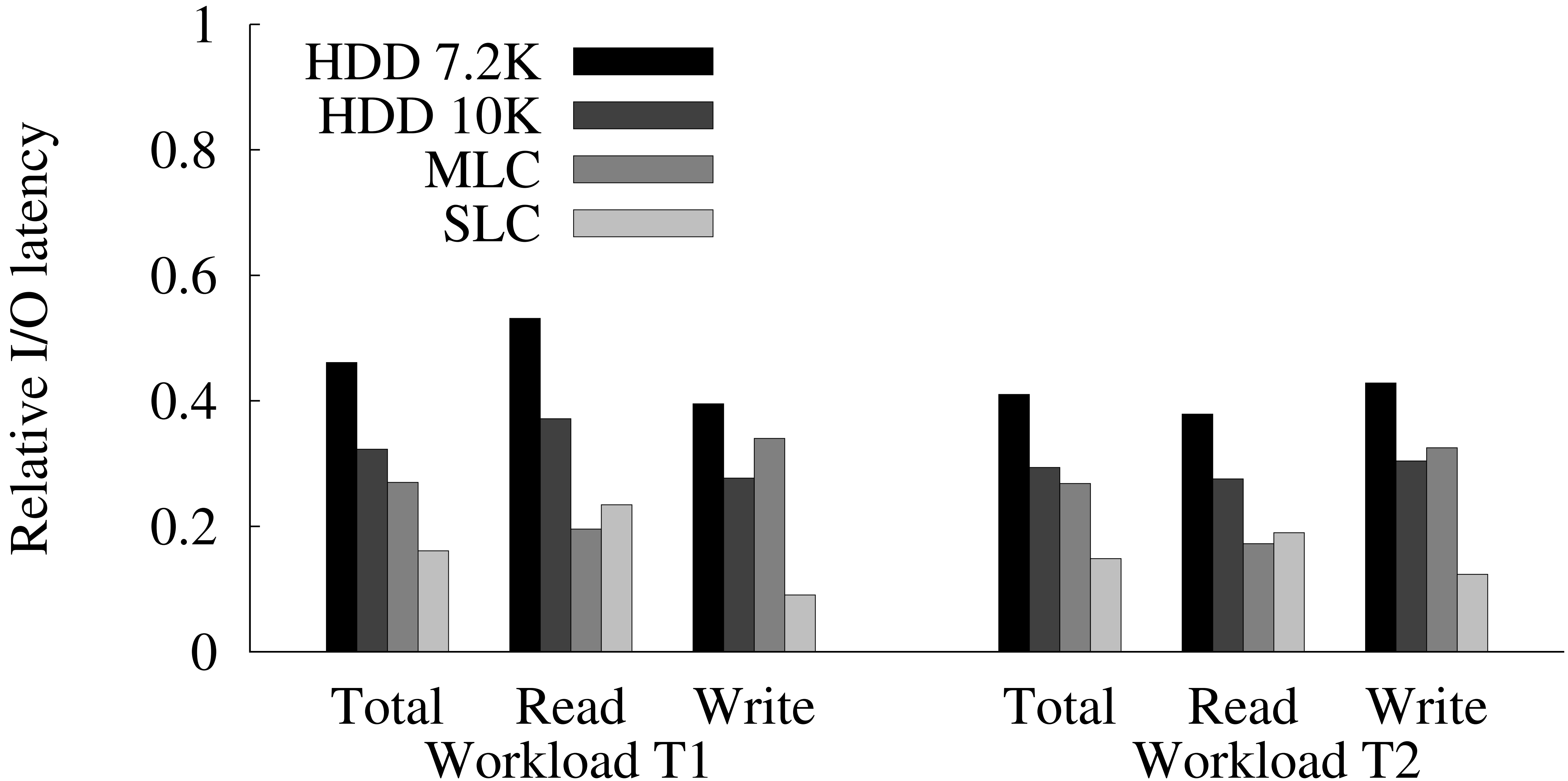
Performance Summary: Era[...]ings



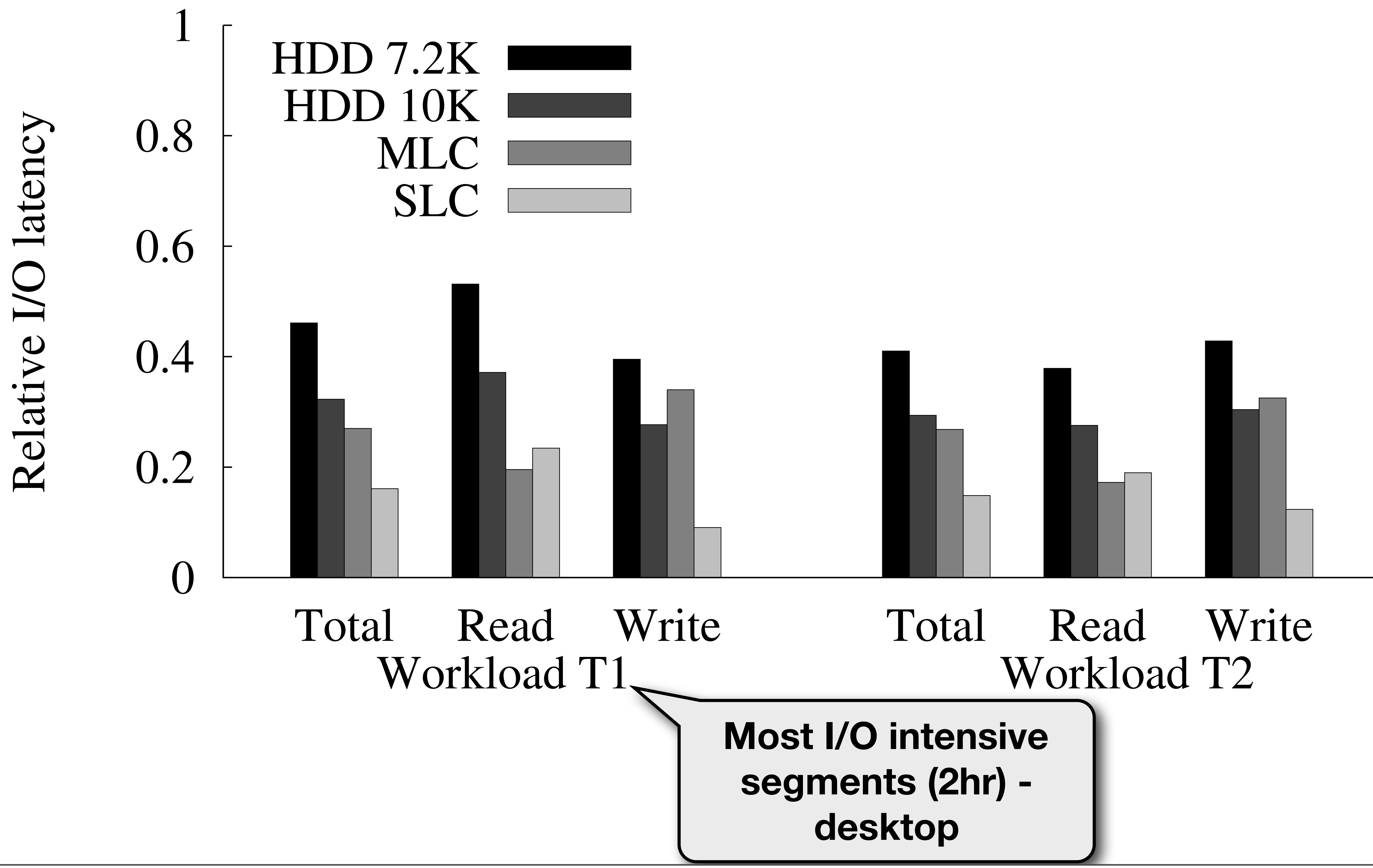
Performance Summary: Erasure Savings



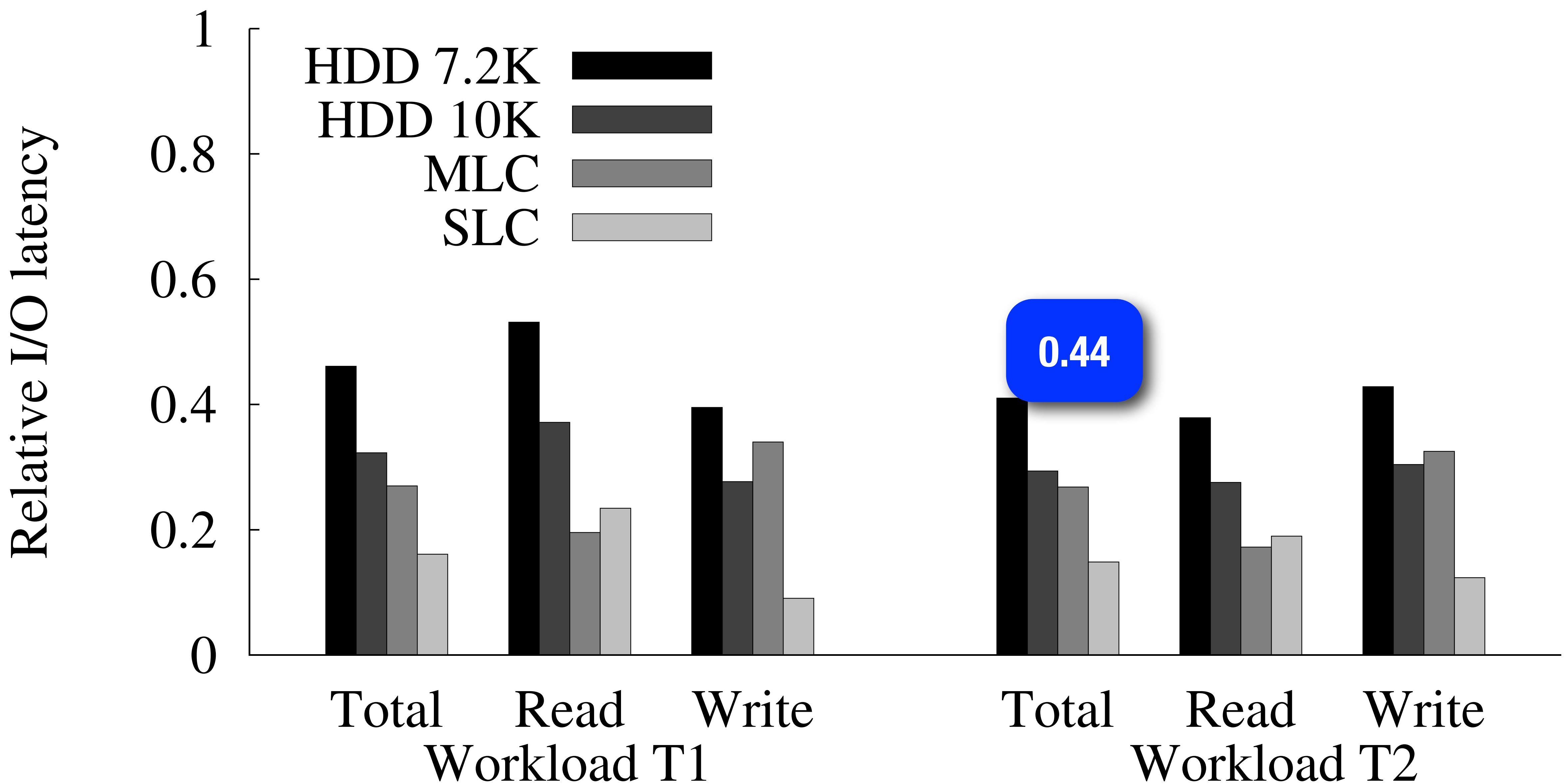
Performance Summary: I/O Latency



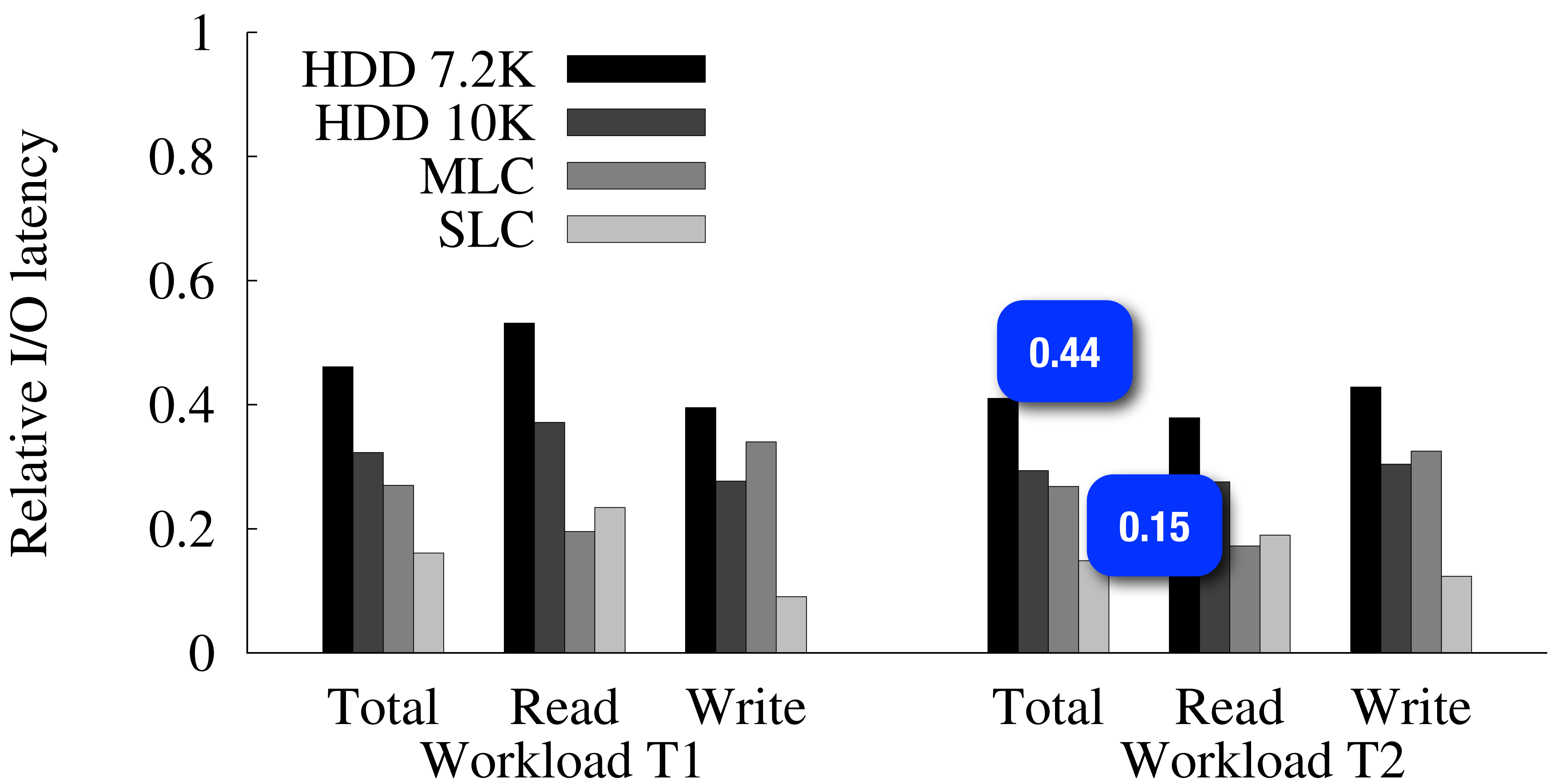
Performance Summary: I/O Latency



Performance Summary: I/O Latency



Performance Summary: I/O Latency



More in the Paper

More in the Paper

- **Evaluation of other policies**
    - Different migration triggers

More in the Paper

- **Evaluation of other policies**
    - Different migration triggers

- **Failure handling**
    - *Have state on two devices*
    - Recovery *more intricate*
        - Leverage existing journalling and recovery techniques
    - *More details in the paper*

Conclusions

- **SSDs starting to appear in desktops/laptops**
  - Contain more write-intensive workloads
  - Lifetimes limited due to limit of block erasures
- **Built Griffin hybrid disk**
  - Uses hard drive as a write-cache
- **Reduces writes while maintaining performance**
  - Reduces writes by 52% (< 5% HDD reads)
  - Improves lifetime by factor of 2
  - Reduces average I/O latency by 56%

Thank you

Griffin picture from:
http://www.e-wollmann.com/griffin.jpg

Related Work

- **SSD Lifetimes**
  - Shown to degrade over time
    - *Grupp et. al [ISM'09]*
    - *Desyoners [HotStorage'09], Boboila et. al [FAST'10]*
- **Hybrid drives**
  - Used SSD as cache for hard drive
    - *Kotsidas et. al [VLDB'08], Combo drive [WISH'09]*
  - Windows ReadyBoost
    - *caches data normally paged out the HDD*
  - Intel Turbo Memory and Sun ZFS+Flash

Improved Sequentiality