ZNSwap: un-Block your Swap

Shai Bergman, Technion; Niklas Cassel and Matias Bjørling, Western Digital; Mark Silberstein, Technion

USENIX ATC'22

2022. 07. 19

Presented by Yejin Han

yj0225@dankook.ac.kr

Introduction

- Data center applications exhibit large memory footprint

Memcached

TensorFlow

VOLTDB

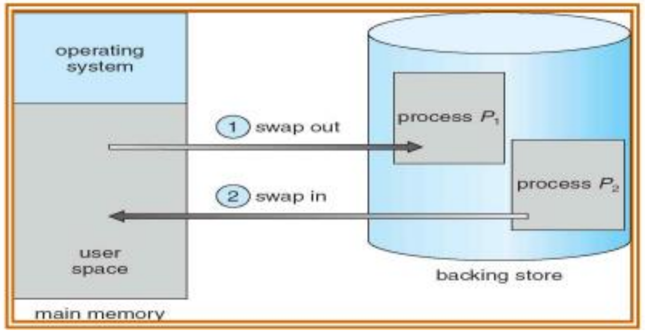
redis

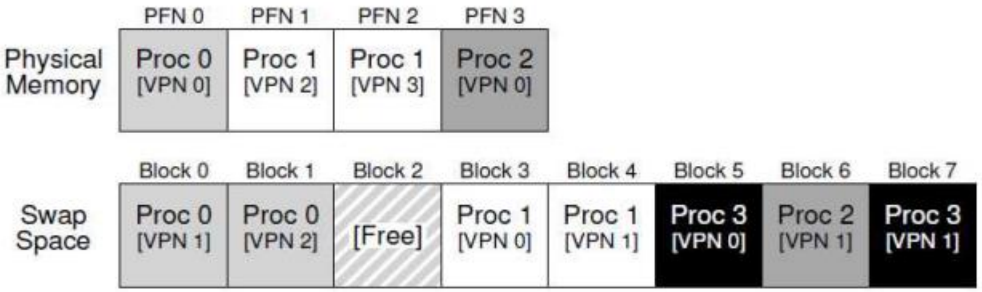
But, not all data is used frequently in the system!!

What is OS Swap?

- Space in disk for moving pages back and forth
  - To migrate data from memory to disk when available memory space is insufficient
  - Linux divides swap device into memory page-sized blocks called *swap-slots*

- Benefit
  - Allow to support the illusion of a large virtual memory for a process (usually larger than physical memory)



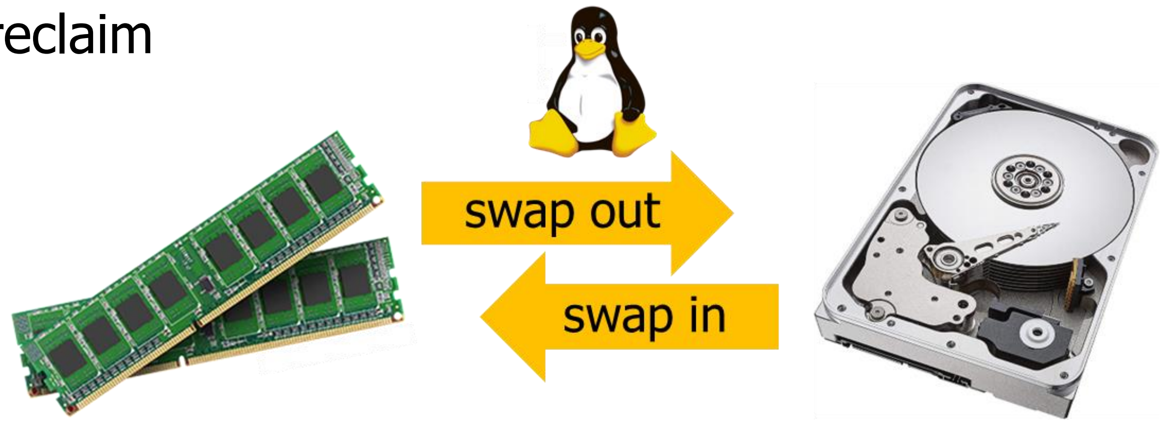


(Source: Lecture note 9. Paging and Beyond Physical Memory)

Why is Swap important?

- Swap is regaining interest from the academia and industry
  - Swap use in academia:
    - Maximizing memory utilization
    - Acting as memory extension
  - Swap use in industry:
    - Facebook's fbtax2 swap controls to improve system efficiency
    - Alibaba cloud: per-cgroup background reclaim

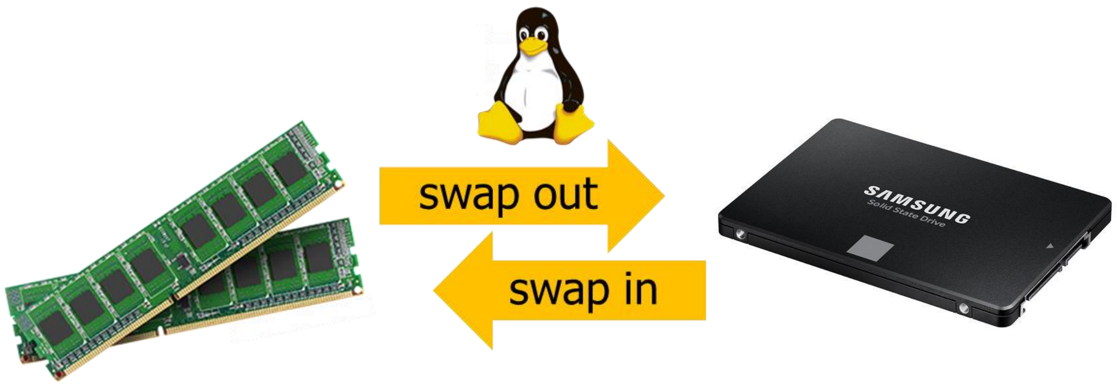
Swap: crucial system component

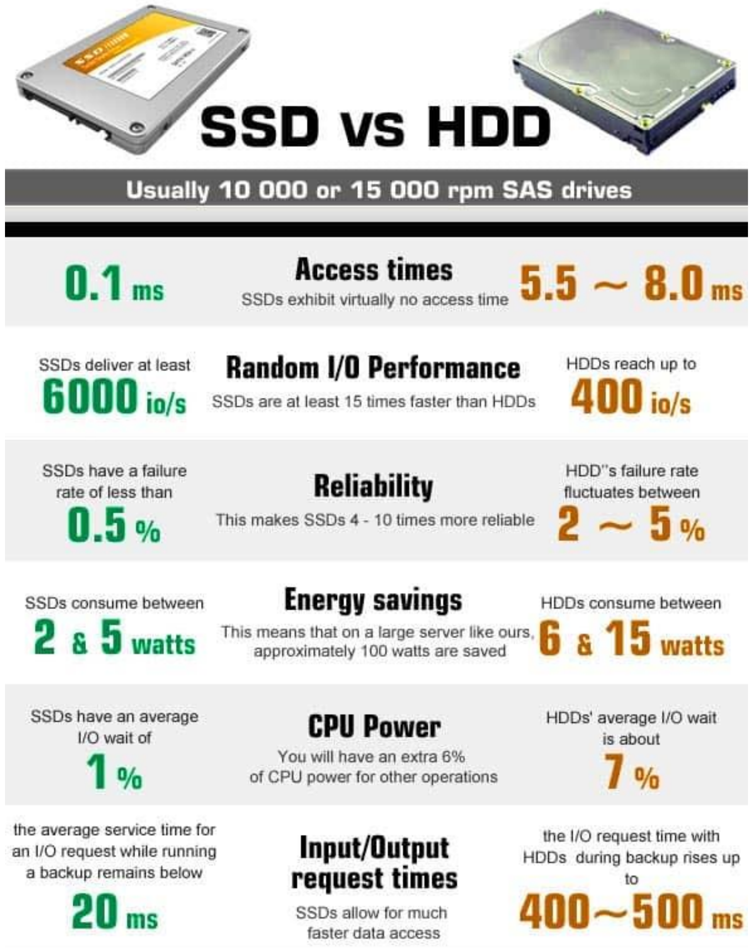


Swap on Traditional SSDs

- Flash technology is advancing:
  - Low latency NAND
  - Available Bandwidth increases

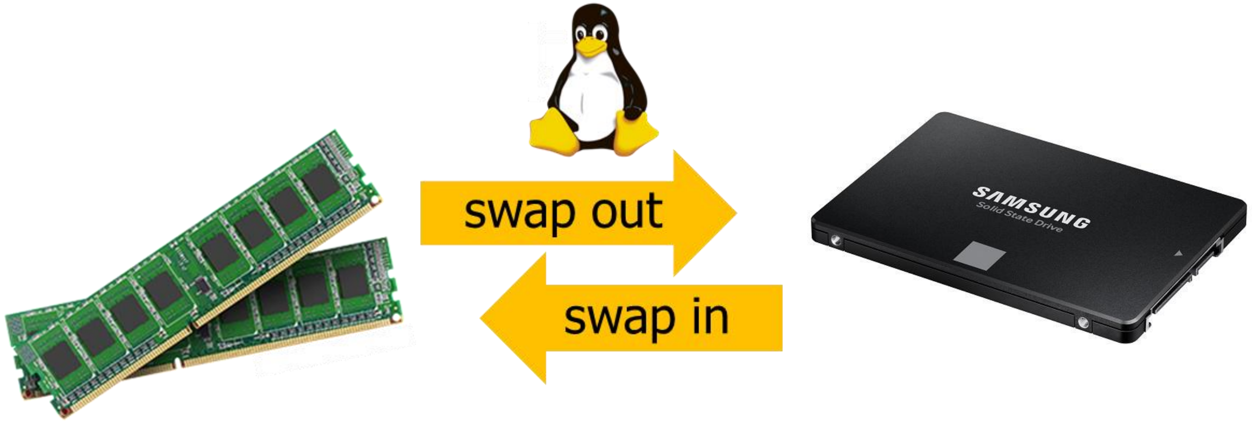
Great for memory swapping!





(Source: SSD vs HDD Speed and Performance Comparison 2022, https://windows101tricks.com/ssd-vs-hdd-which-is-better-for-you/)

Is it great for memory swapping?



Problem: Swap on Traditional SSDs

- Performance degradation as the swapped-out data occupies a larger part
  - Drastic swap bandwidth drop because the GC overheads grow

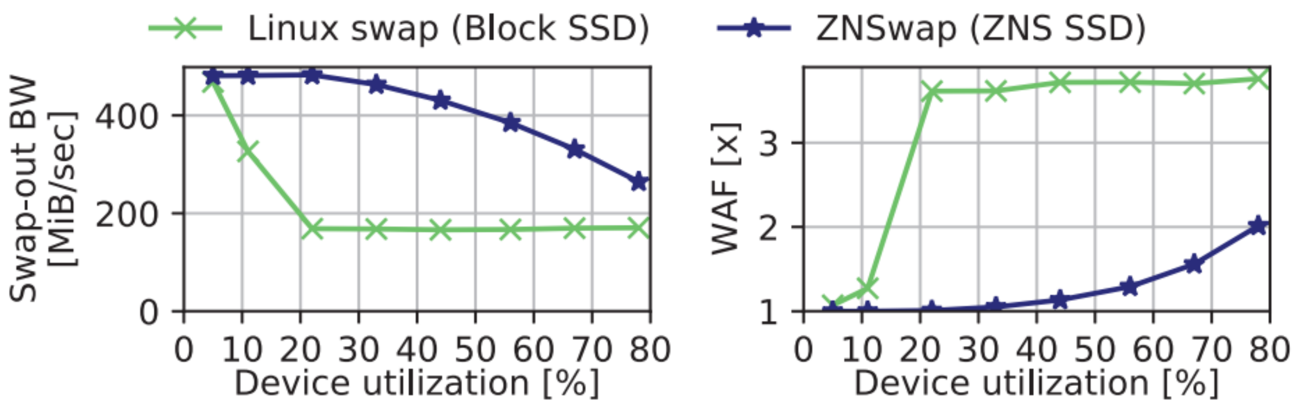
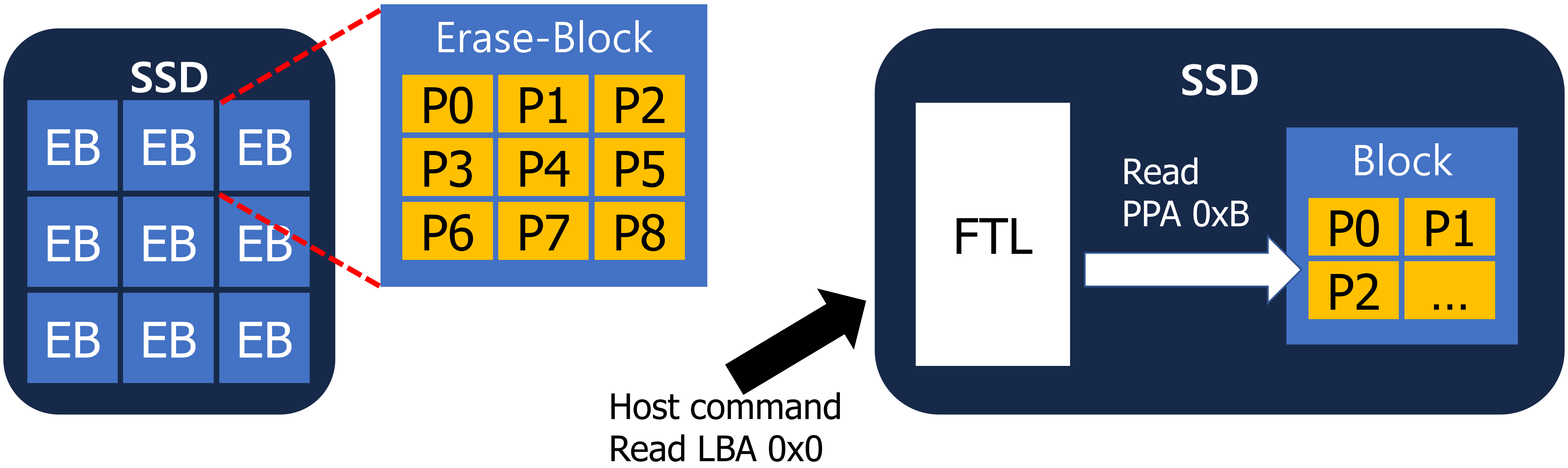


Figure 1: Swap-out bandwidth of random memory accesses (a common swap access pattern [43, 55]), with default Linux swap on Block SSD and ZNSwap on ZNS SSD. The two 1TB SSDs share the same hardware platform and media. WAF– Write Amplification Factor.

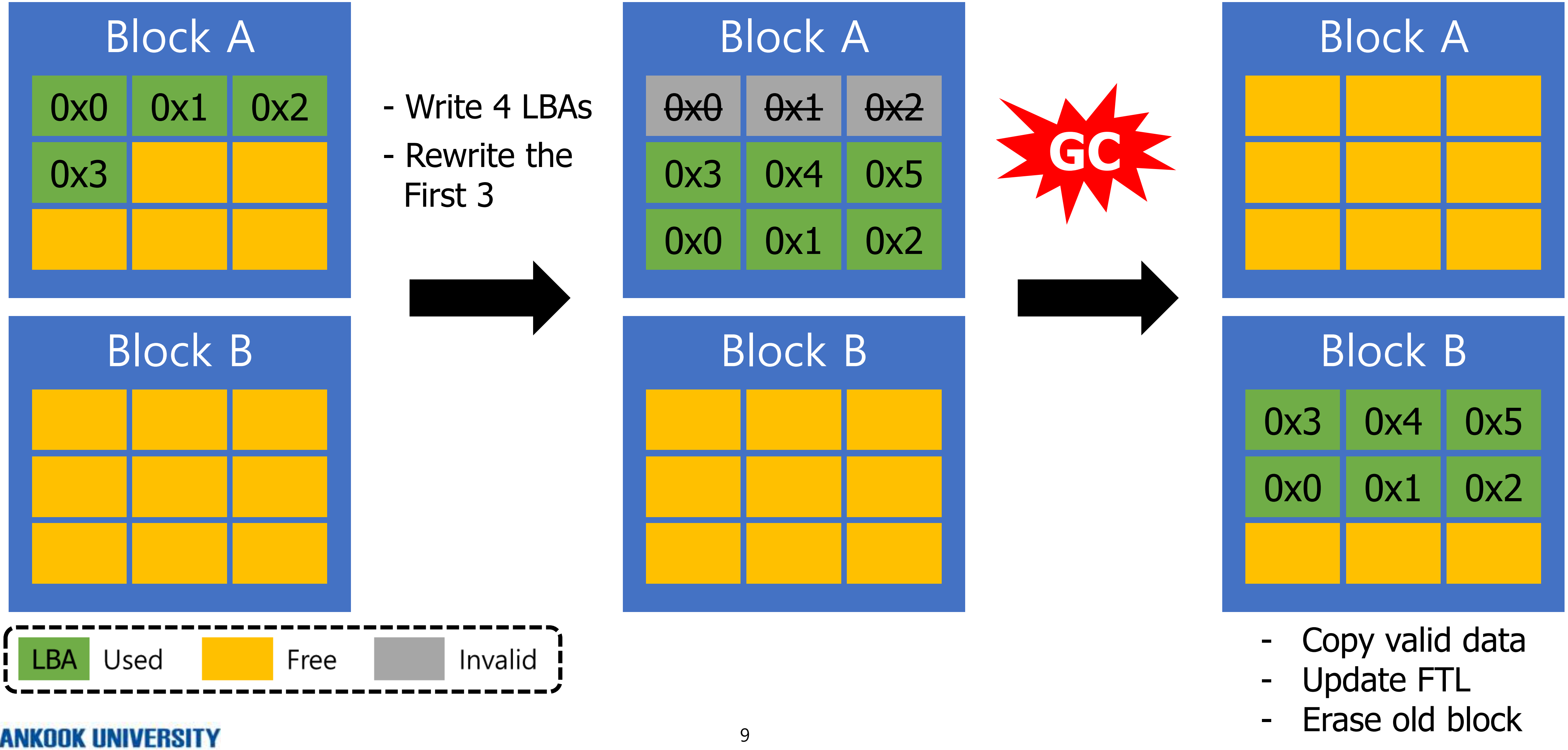
Swap performance drop caused by GC

Why: intrinsic structure of SSDs

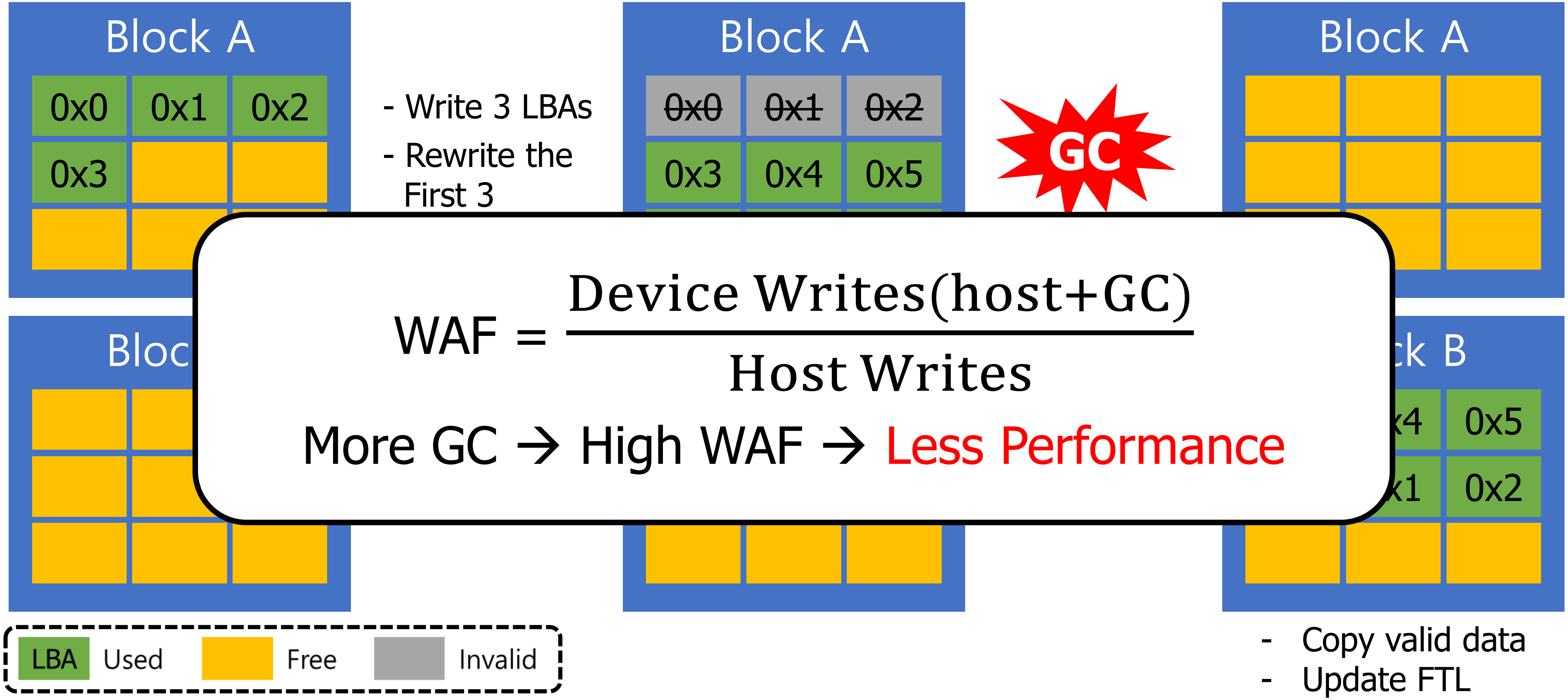
- Flash SSD's inherent mismatch between the block abstraction and the intrinsic properties



Why: Garbage collection & Write Amplification



Why: Garbage collection & Write Amplification



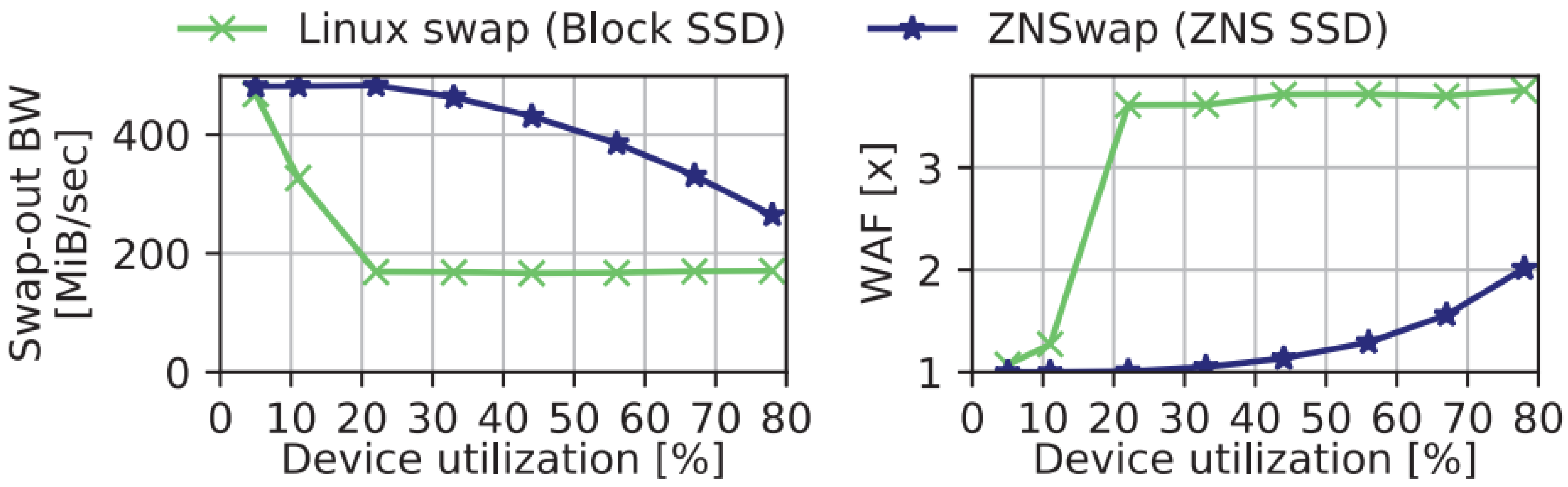
$$\text{WAF} = \frac{\text{Device Writes(host+GC)}}{\text{Host Writes}}$$

More GC → High WAF → Less Performance

- Copy valid data
- Update FTL
- Erase old block

Problem: Swap on Traditional SSDs

- Knowledge gap between SSD and OS
  - device-side GC is not aware of invalid swap data because OS does not notify SSD
  - GC copies unnecessary data
  - Performance decrease, WAF increase



Problem: Swap on Traditional SSDs

- How about TRIMs?
  - Hint by the host to invalidate a flash-page
  - GC will not copy invalidated pages

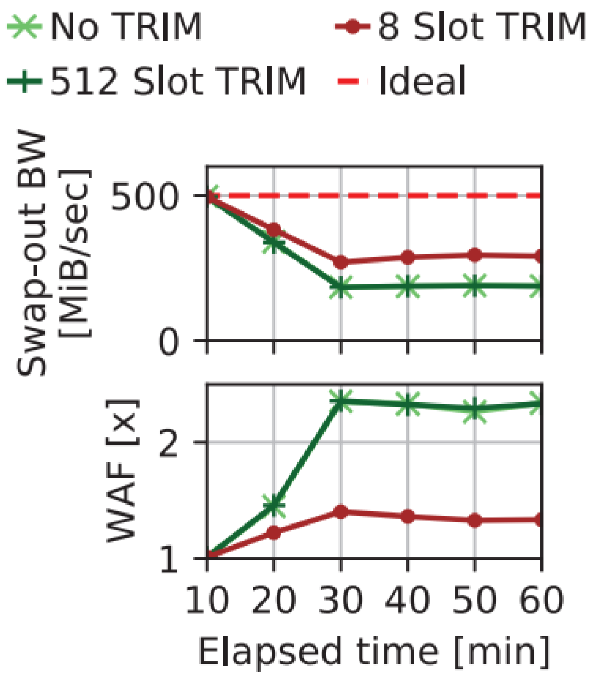


Figure 2: Swap-out bandwidth over time. Random memory writes using 40% of swap capacity.

TRIMs are not effective at lowering GC overheads for swap

Problem: Swap on Traditional SSDs

- Performance isolation cannot be guaranteed on TrSSD

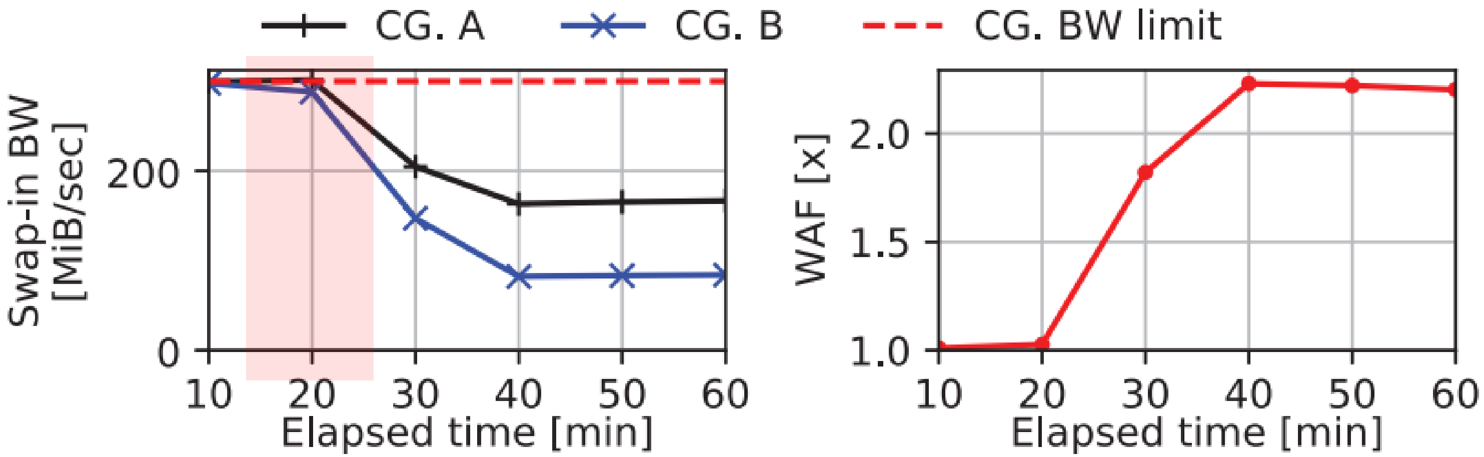


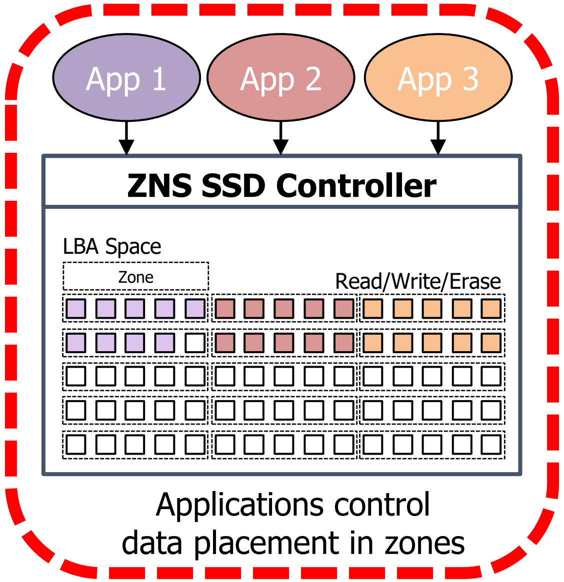
Figure 5: Swap-in bandwidth and WAF of 100%-random-read cgroup (A) and 50/50%-random-read/write cgroup (B) co-running together, each throttled to 300MiB/sec reads and 300MiB/sec writes.

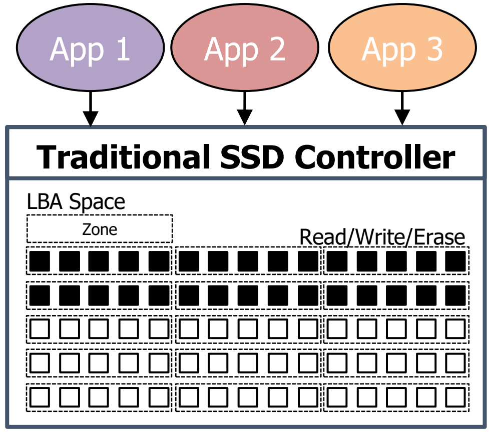
The GC impairs performance isolation dictated by the host OS

How about ZNS SSD?

- ZNS (Zoned Namespace): Tighter SSD-APP coupling
  - SSD is divided into zones
  - Each zone is written sequentially
  - Zones need to be reset before re-writing

  - No complicated FTL, no device GC
  - Higher degree of control over the device





Device controls data placement

ZNS + Swap = ZNSwap

ZNSwap Overview

- ZNSwap's main design

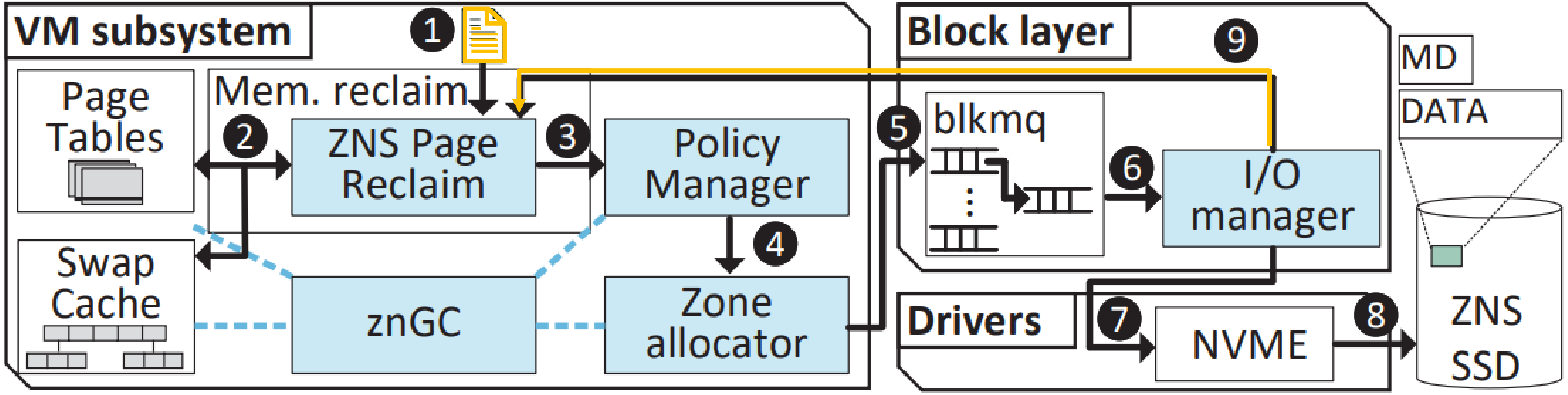
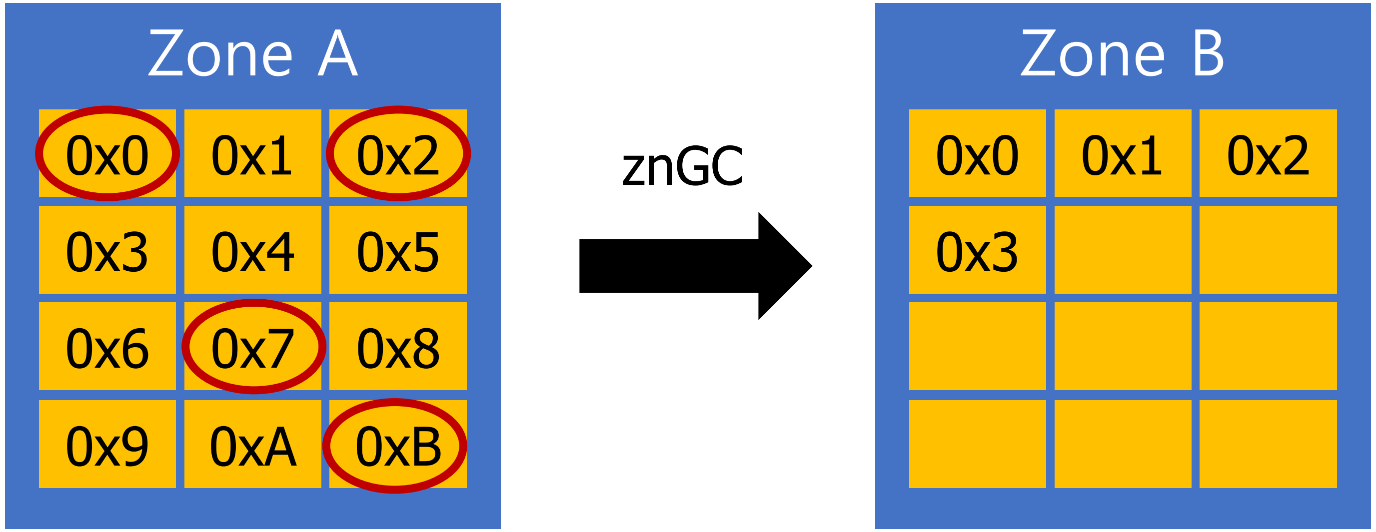


Figure 7: ZNSwap overview. Shaded shapes are internal ZN-Swap components.

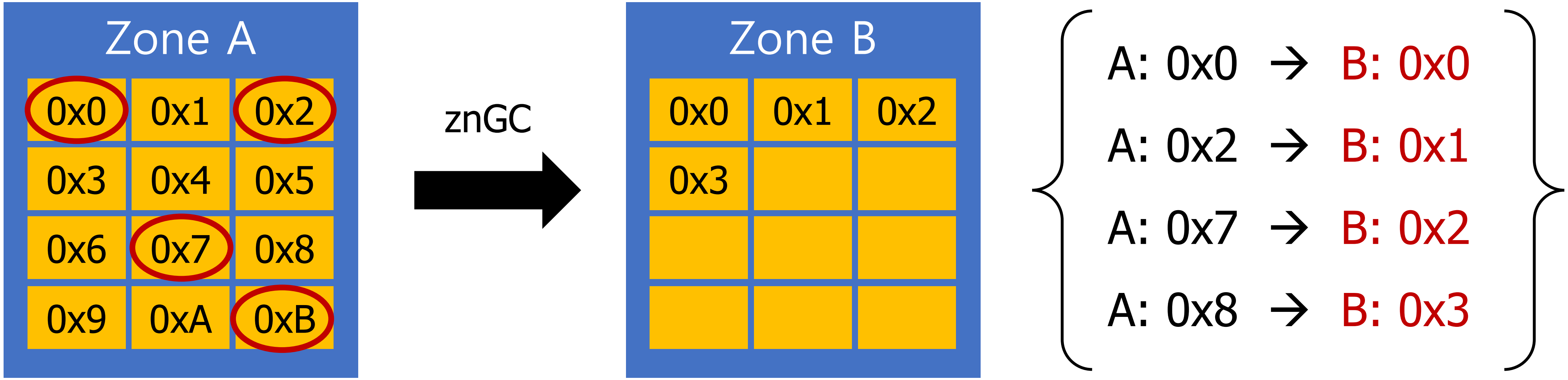
znGC

- Host-side GC for ZNS device eliminates:
  - TRIMs
  - uncertainty of GC
  - copy of invalid data



znGC

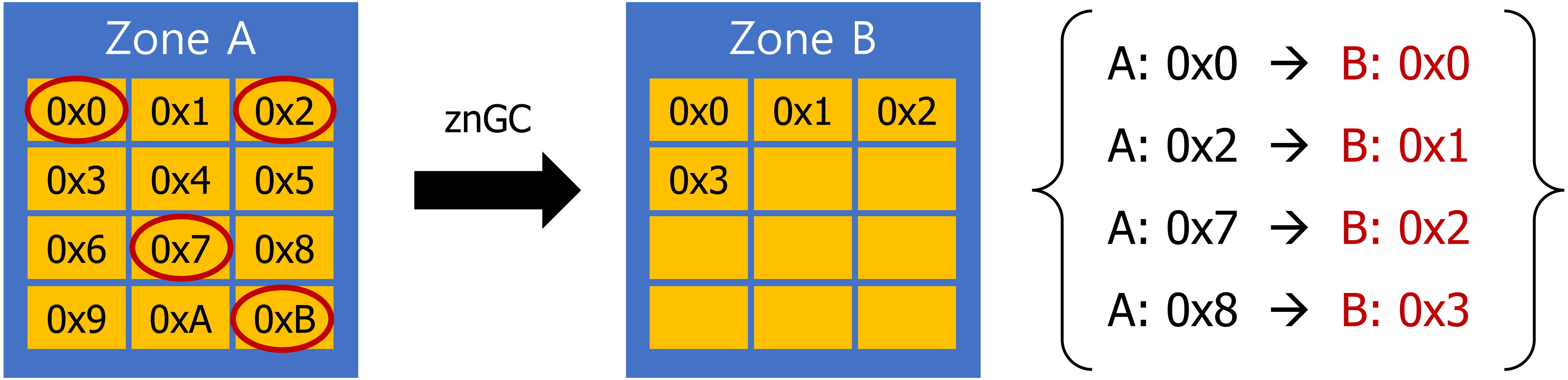
- Host-side GC moves valid swap data to new locations:



- Problem:
  - No FTL for indirection in ZNS
  - Page tables point to old locations in SSD

znGC

- Host-side GC moves valid swap data to new locations:

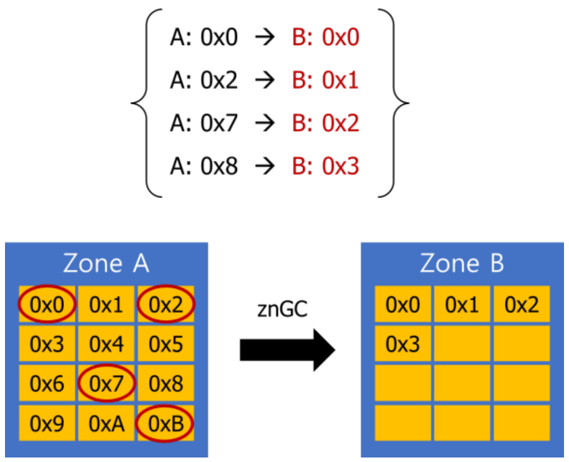


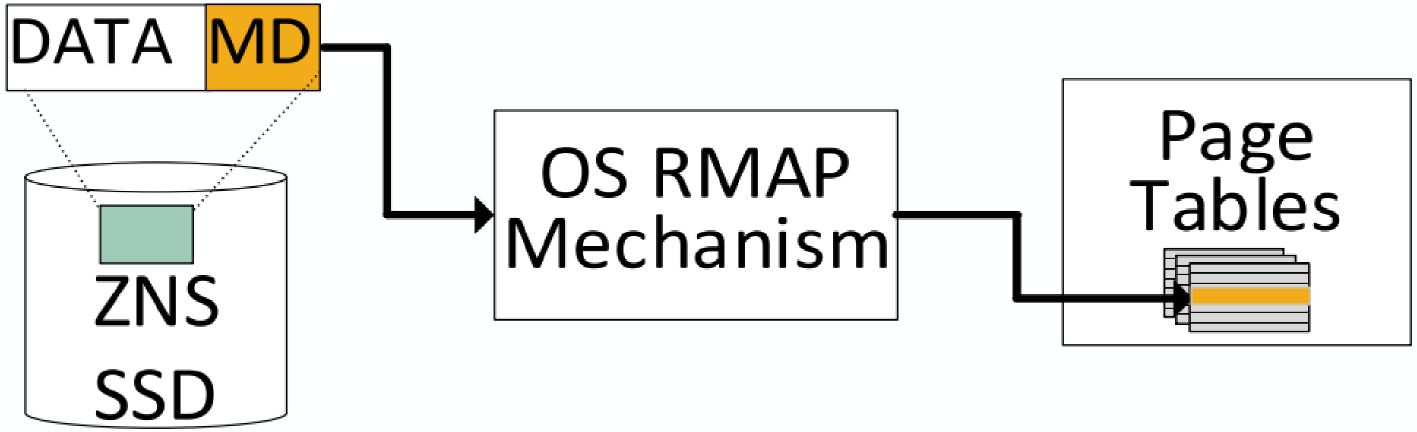
- Problem:
  - No FTL for indirection in ZNS
  - Page tables point to old locations in SSD

How to locate all page table entries?

znGC

- znGC Solution:
  - Store OS reverse-mapping info (anonymous VMA ptr + index)
  - Utilize NVMe per-block Metadata region

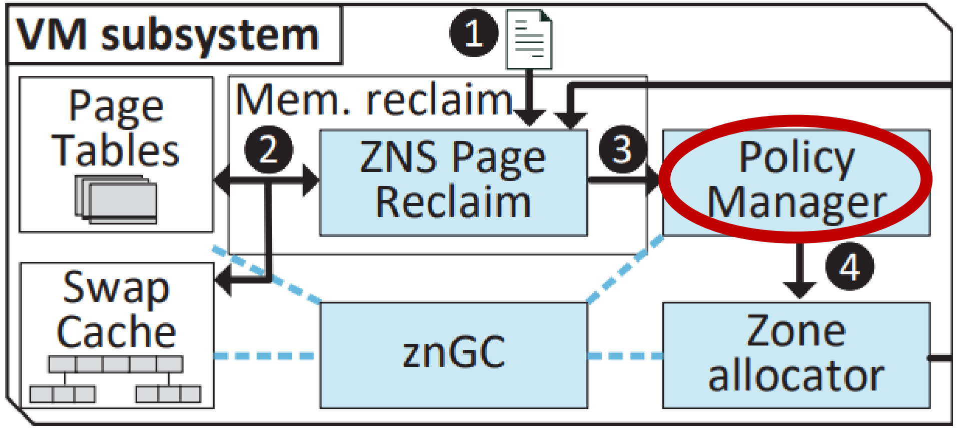




Efficiently update page tables with new location

znGC-swap Integration

- Three reclaim policies:
  - per-core policy
    - Assign a swap-zone per-CPU-core
  - Hot/cold policy
    - Assign hot and cold pages to different swap-zones
  - Cgroup policy
    - Assign a swap-zone per-cgroup



Evaluation

- Experiment Setup

CPU	2x Intel Xeon Silver 4216 CPU
Memory	512 GiB RAM (2x 256 GiB DDR4 2933Hz)
Kernel	Linux kernel 5.12.0
SSD	1TB Western ZN540 ZNSSSD / 1TB Equivalent Conventional SSD

Evaluation: synthetic benchmarks

- vm-scalability

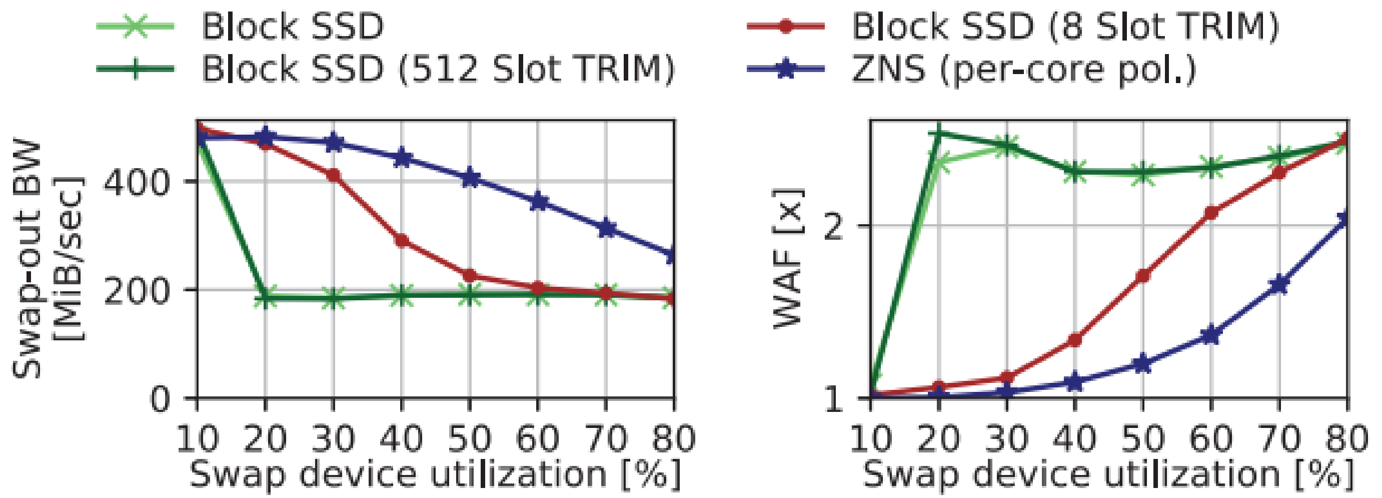


Figure 10: Swap-out bandwidth of `vm-scalability` with random memory writes. As expected, higher device utilization results in higher GC load.

ZNSwap avoids unnecessary data copies
50% util: 2x higher throughput, 2x lower WAF

Evaluation: synthetic benchmarks

- Cgroup Isolation
  - Cgroup A: 100% writes, Cgroup B: 100% reads

(swap bandwidth is limited via `blk-throttle` to 300Mib/sec)

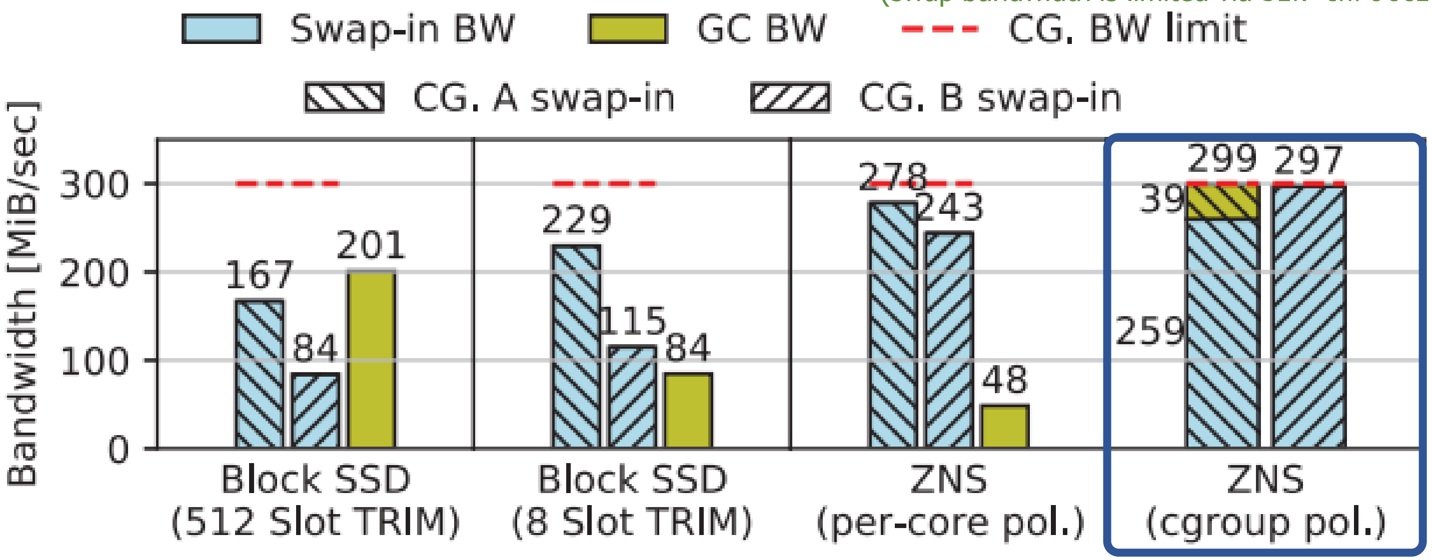


Figure 11: Bandwidth distribution among different cgroups, one reading and another writing data.

ZNSwap enables performance isolation

Evaluation: application benchmarks

- Memcached: Facebook ETC workload
  - random-skewed access pattern with 90% of requests accounting for 10% of the keys

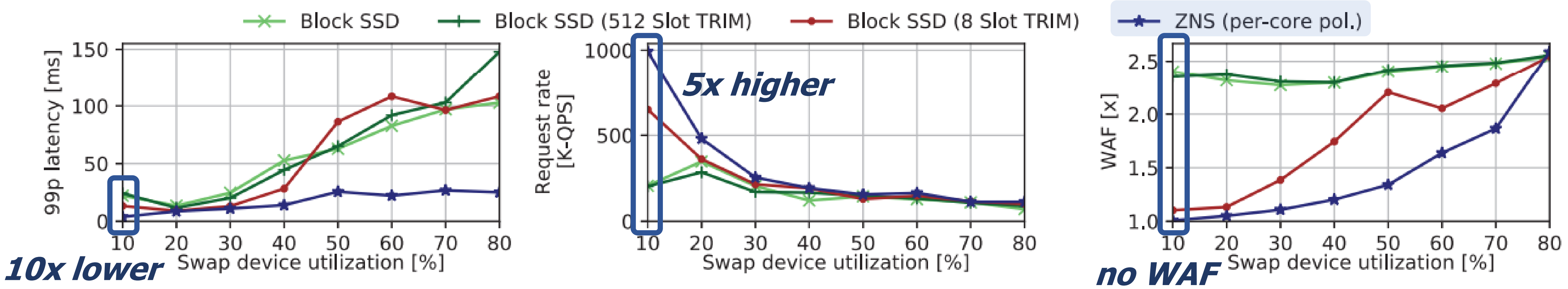


Figure 12: memcached Facebook ETC 99 percentile latency at the highest throughput

ZNSwap consistently outperforms Block SSD-based swap

Evaluation: application benchmarks

- Redis: YCSB workload
  - 50% read/ 50% updates in a 20-80 hotspot distribution (80% of accesses target 20% of working set)

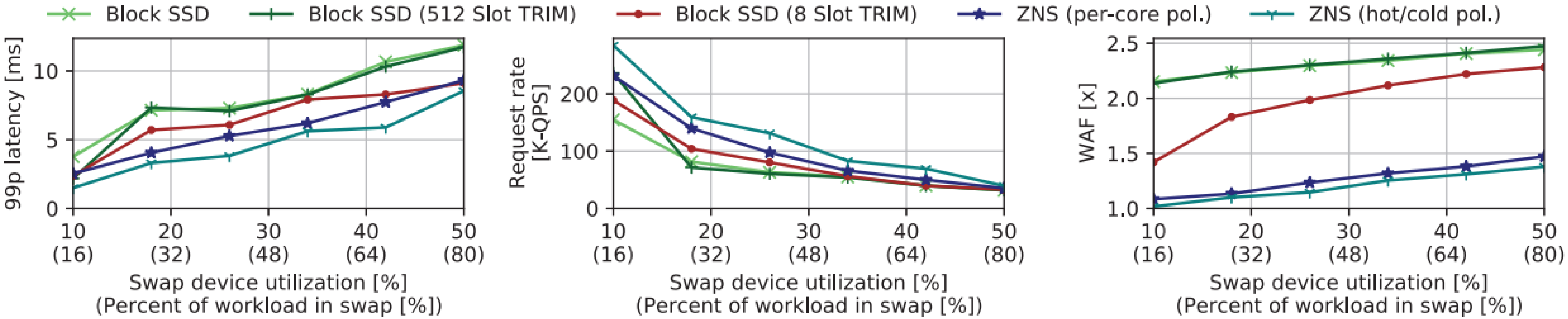


Figure 13: redis 20-80 hotspot distribution 50/50 read/write, 99p latency at maximum throughput

ZNS policies outperform Block SSD in all performance metrics

Conclusion

- Swap is regaining interest in academia and industry
- Swap on Traditional SSDs suffer from performance anomalies
- ZNSwap enables tight SSD <-> OS swap integration
  - Lowers WAF and higher performance benefits over swap on traditional SSDs

https://github.com/acsl-technion/znswap

ZNSwap: un-Block your Swap

Shai Bergman, Technion; Niklas Cassel and Matias Bjørling, Western Digital; Mark Silberstein, Technion

USENIX ATC'22

Thank You !

2022. 07. 19

Presented by Yejin Han

yj0225@dankook.ac.kr