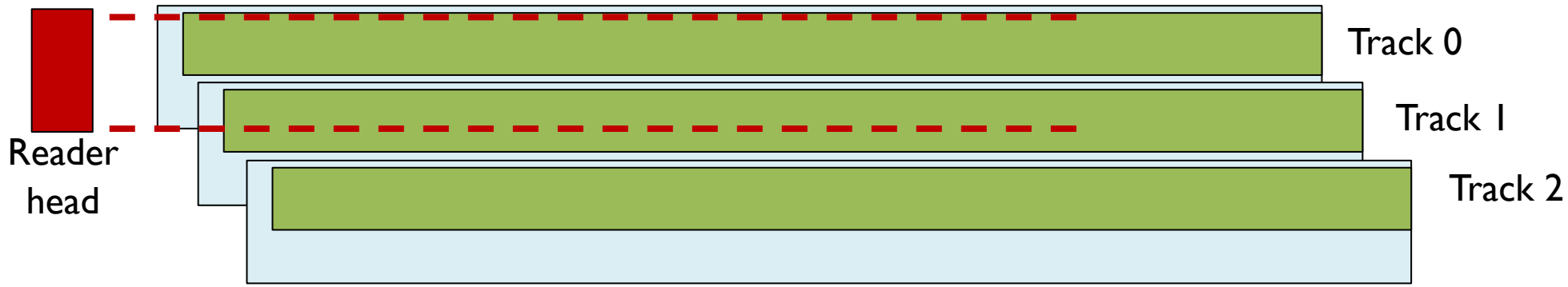
ZoneAlloy: Elastic Data and Space Management for Hybrid SMR Drives

Fenggang Wu Bingzhe Li Zhichao Cao Baoquan Zhang
Ming-Hong Yang Hao Wen David H.C. Du

HotStorage 19

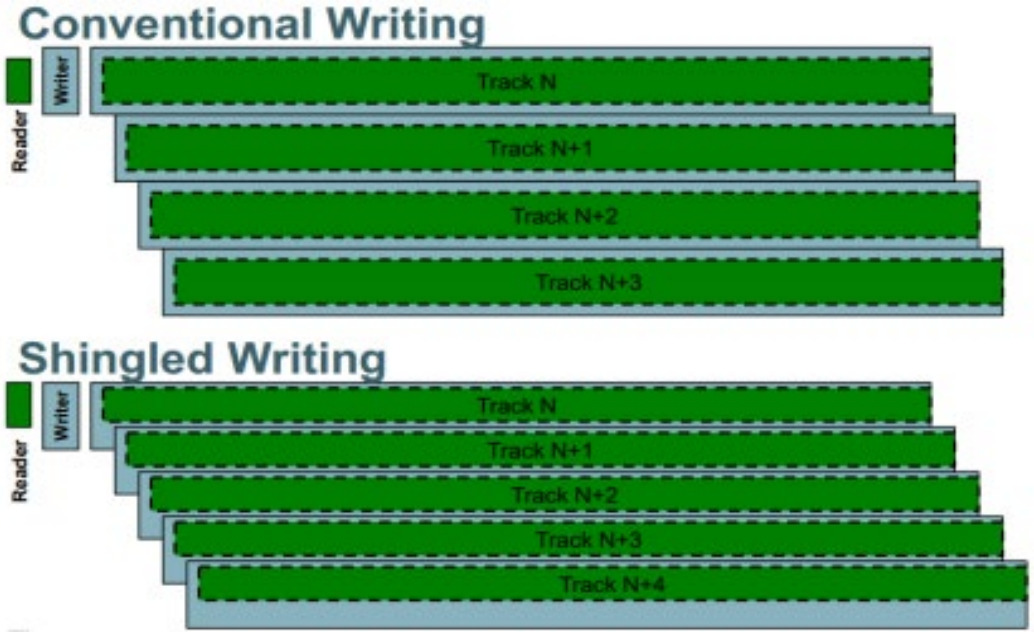
SMR HDDs

- SMR structure
  - Tracks are overlapped in a shingled fashion to achieve higher areal density on the same disk platter
  - Track width is smaller than the magnetic reader head → need **Read-Modify-Write** to update blocks in-place



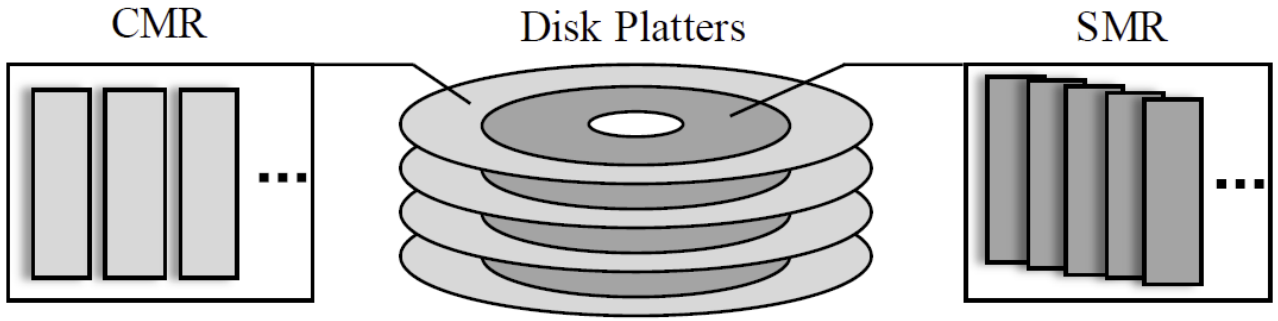
- The Problem: SMR Update Overhead

SMR, CMR and H-SMR HDDs



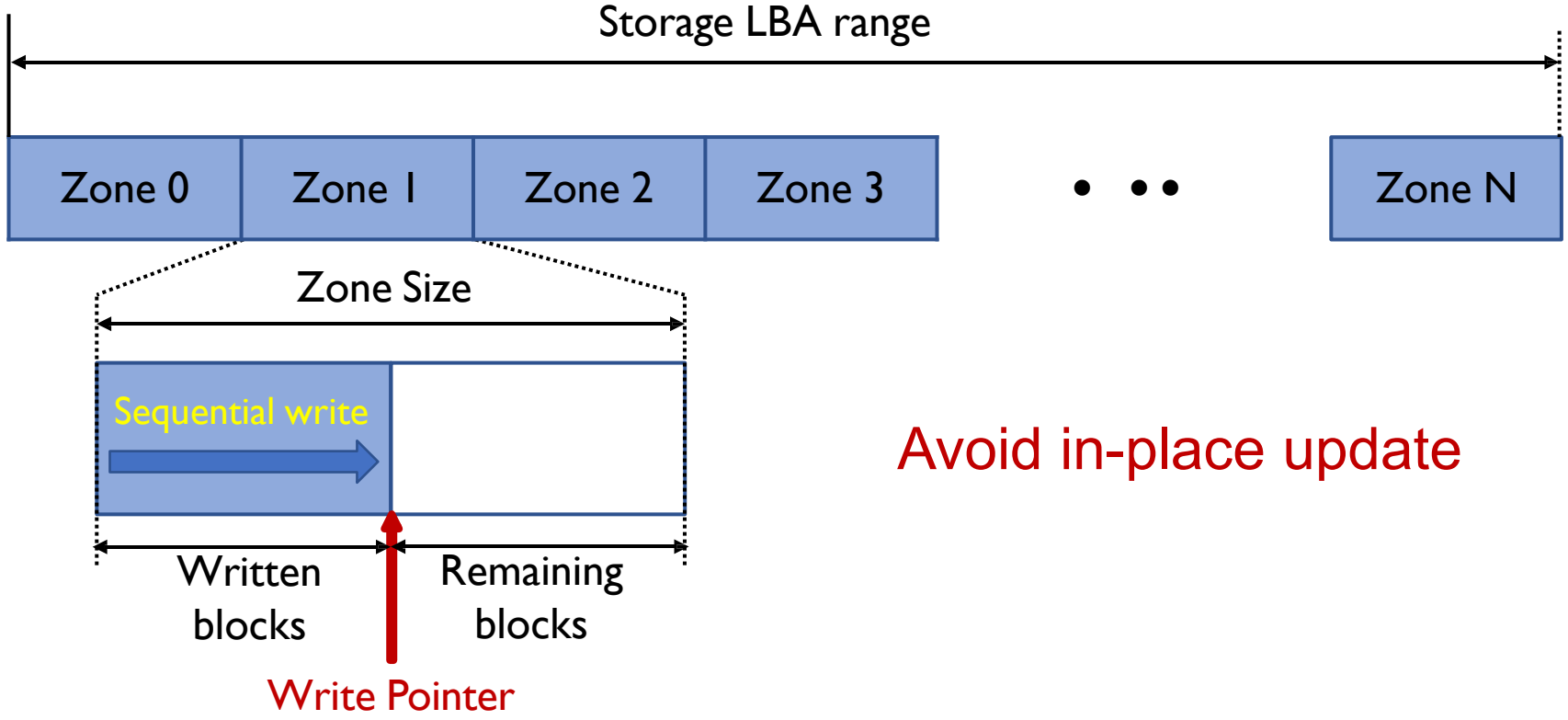
- CMR:
  - Low capacity; High performance
- SMR:
  - High capacity; Low performance
  - High update overhead (latency)

- H-SMR manages the **performance/capacity trade-off** on the disk



Zone Structure for H-SMR

- "Zone" is a consecutive LBA space with a size of 256MiB
  - **SMR zone** has a write pointer indicating where the next write should go to enforce sequential write.



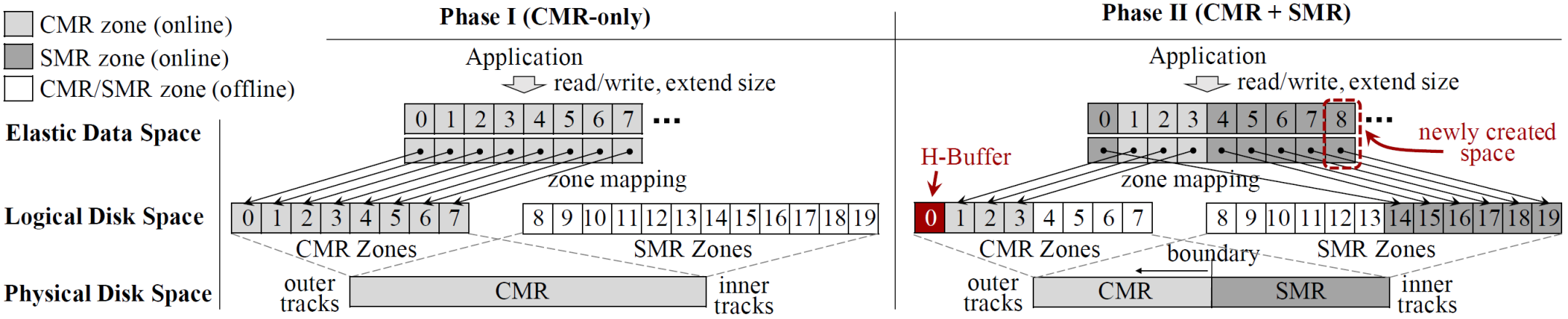
Avoid in-place update

Four Problems with H-SMR

- How to arrange the format layout and place data accordingly?
  - Coexistence of CMR and SMR
- How to perform format conversion efficiently?
  - SMR density is about 1.5 times that of CMR
- How to reduce SMR update overhead?
  - In-place update SMR data blocks
- How to adapt to dynamic workloads?
  - “Hot” and “Cold” data requires different latency

Zone Mapping and Two-phase Allocation

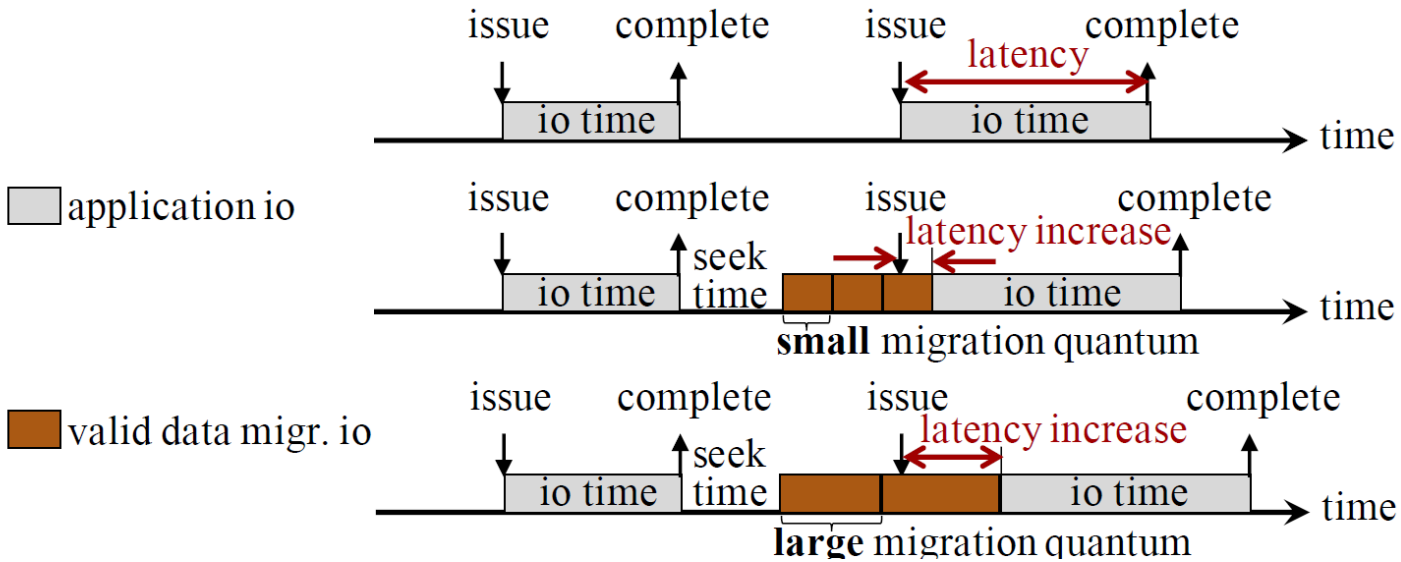
- H-SMR divide space into zones (CMR+SMR)
- H-SMR provide elastic data space
  - Full CMR (phase I) to CMR+SMR (phase II)



HDDs outer tracks have higher performance than inner tracks

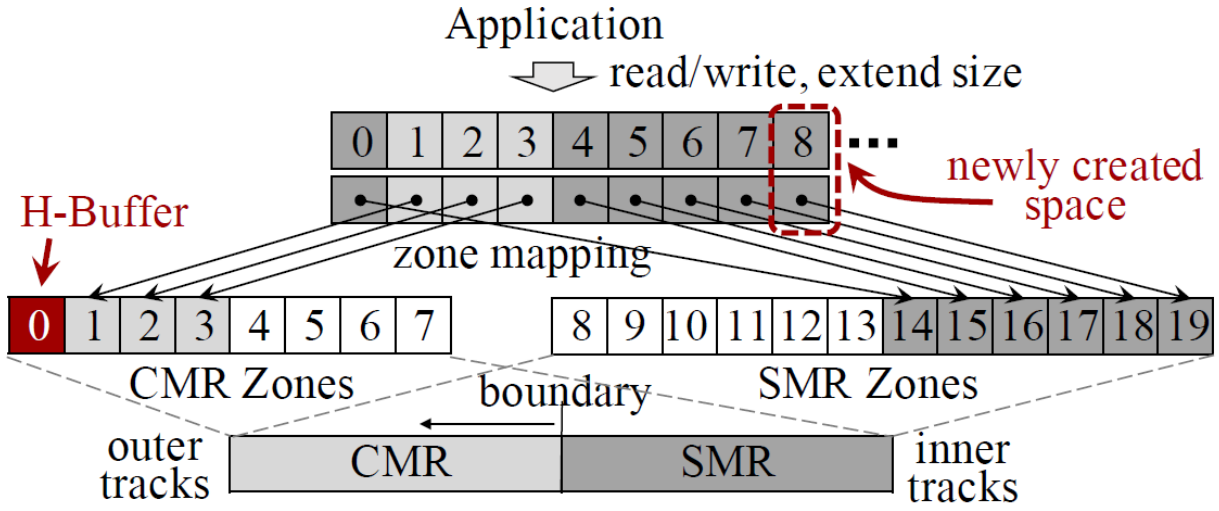
Quantized Migration

- Larger migration quantum →
  - conversion finish sooner
  - higher latency increase
- Application specifies the acceptable increase of latency

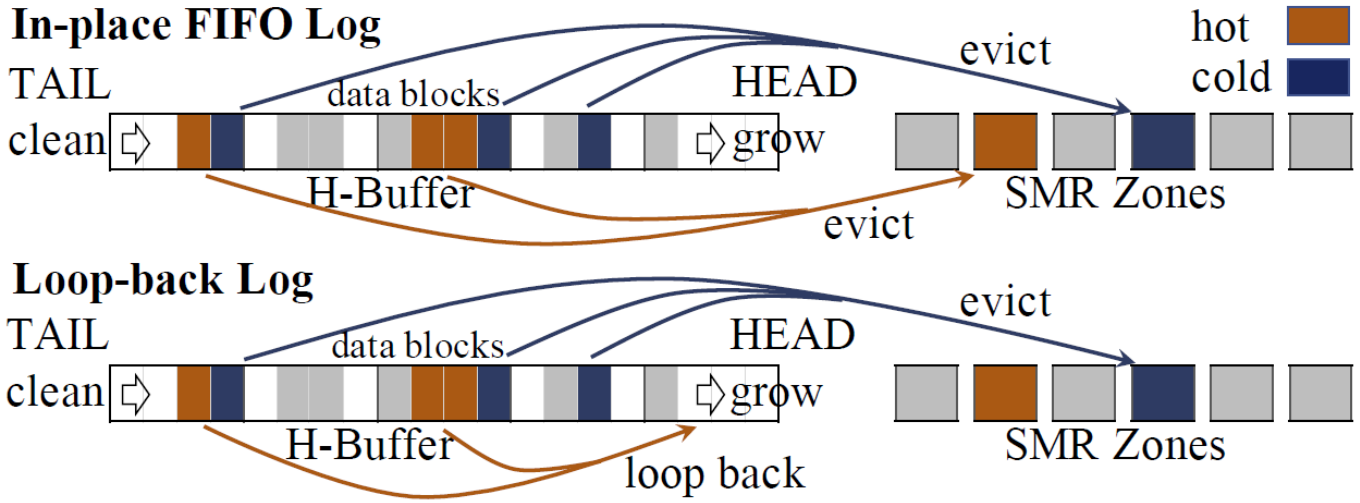


H-Buffer based SMR Update

- H-Buffer: small host-controlled CMR cache
  - Absorb and migrate SMR updates in batches
- Cache policy: LRU?
  - Block-based LRU leads to poor performance: zone update for each block
  - Zone-based LRU fragment free space of H-Buffer, causing random I/Os (Low performance)



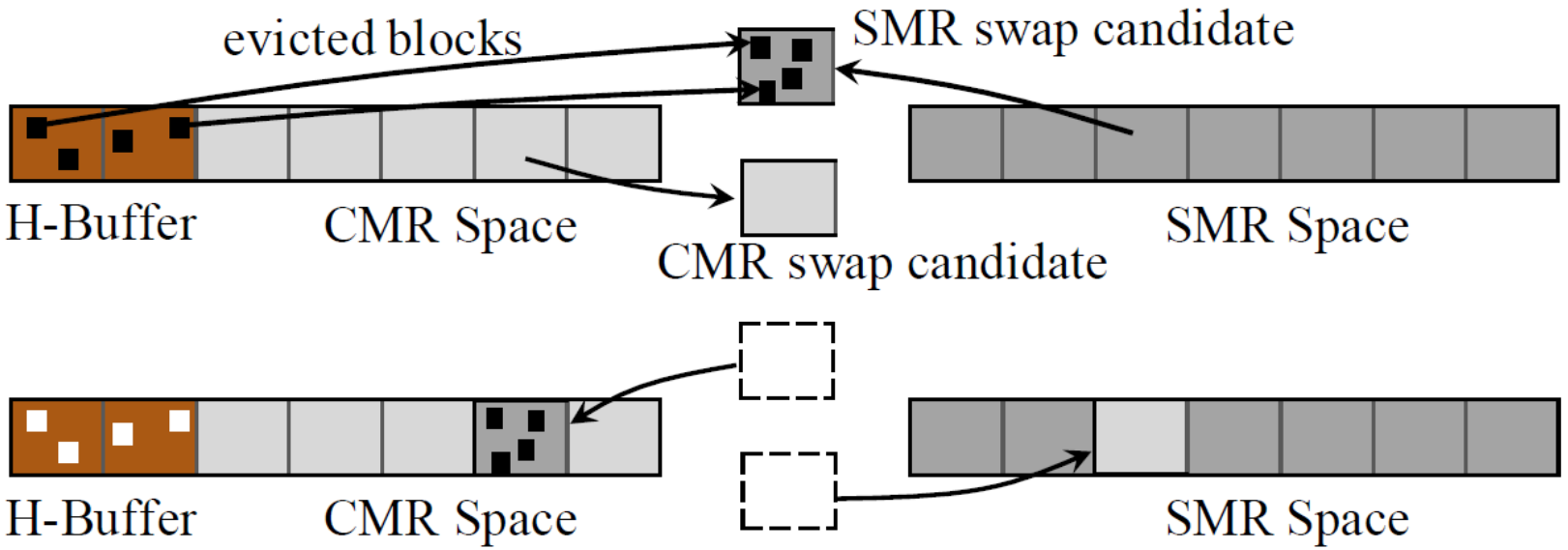
H-Buffer based SMR Update



- In-place FIFO Log
  - Evict blocks of same zone together→reduce zone update
  - Frequently updated data blocks will come back to H-Buffer soon
- Loop-back Log:
  - Use LRU to predict “hot” zones, then move blocks back to HEAD

Zone-Swap Scheme

- SMR swap candidates
  - Zones with an occupancy in H-Buffer(during last epoch) above a threshold
- CMR swap candidates
  - CMR zones that were not updated during last epoch

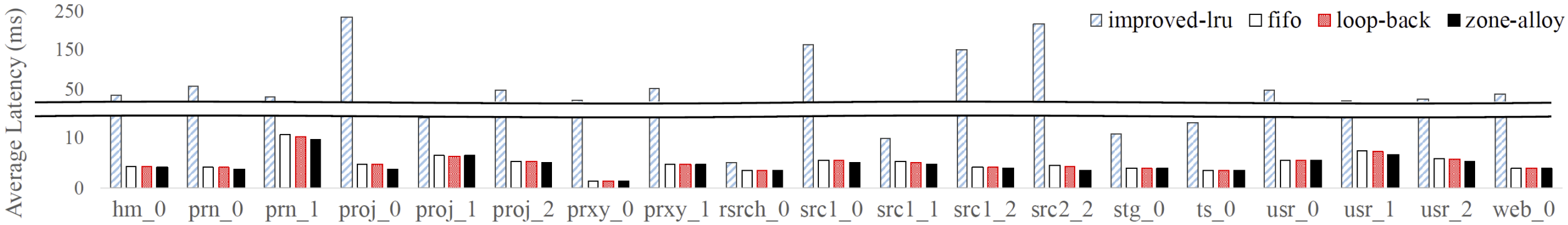


Evaluation

- Platforms & Traces
  - H-SMR simulator based on DiskSim.
  - Microsoft Research (MSR) Cambridge traces
- Methodology
  - SMR to CMR areal density ratio is set to 1.5:1
  - Only the traces with a write footprint greater than 10GB are used

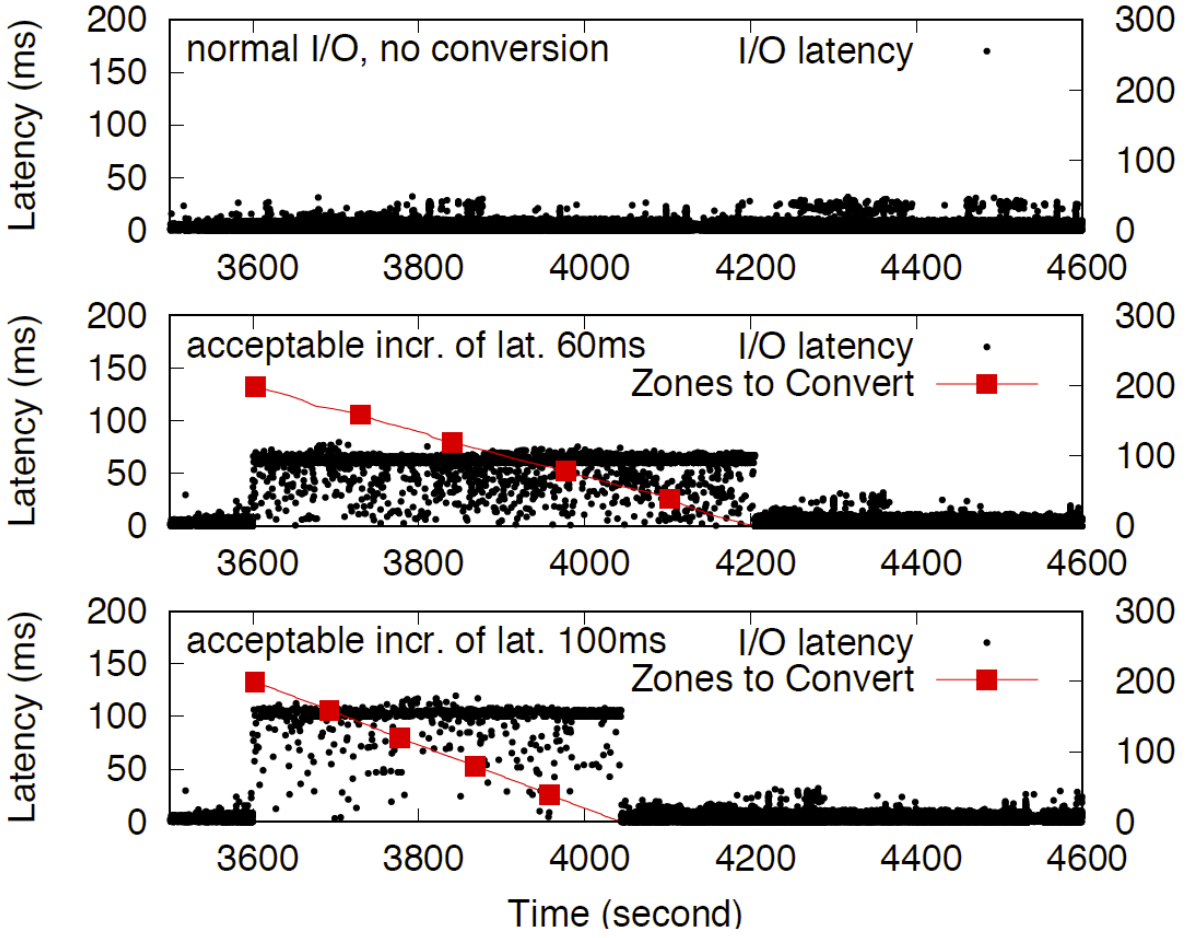
Overall Latency

- H-Buffer to SMR partition size ratio is set to as low as 0.02%, and the disk usage is set to 99.9%
  - Improved-lru: zone-based LRU H-Buffer



Zone-alloy achieves lowest latency, and effect of zone-swap is significant in proj_0/usr_1 datasets

Quantized Migration



- Application requests 25GiB of new space (200 zones to convert) with acceptable increase of latency set to 60 ms and 100 ms
  - The higher the acceptance delay, the faster the conversion
  - Delay not changed when no conversion

Conclusion

- ZoneAlloy, an elastic data and space management scheme that hides the H-SMR details and presents an elastic data space to the application.
  - H-Buffer and Zone-swap to reduce SMR update overhead
  - Two-phase allocation to support elastic data space
  - Trade-off to perform conversion efficiently while bounding their performance degradation