

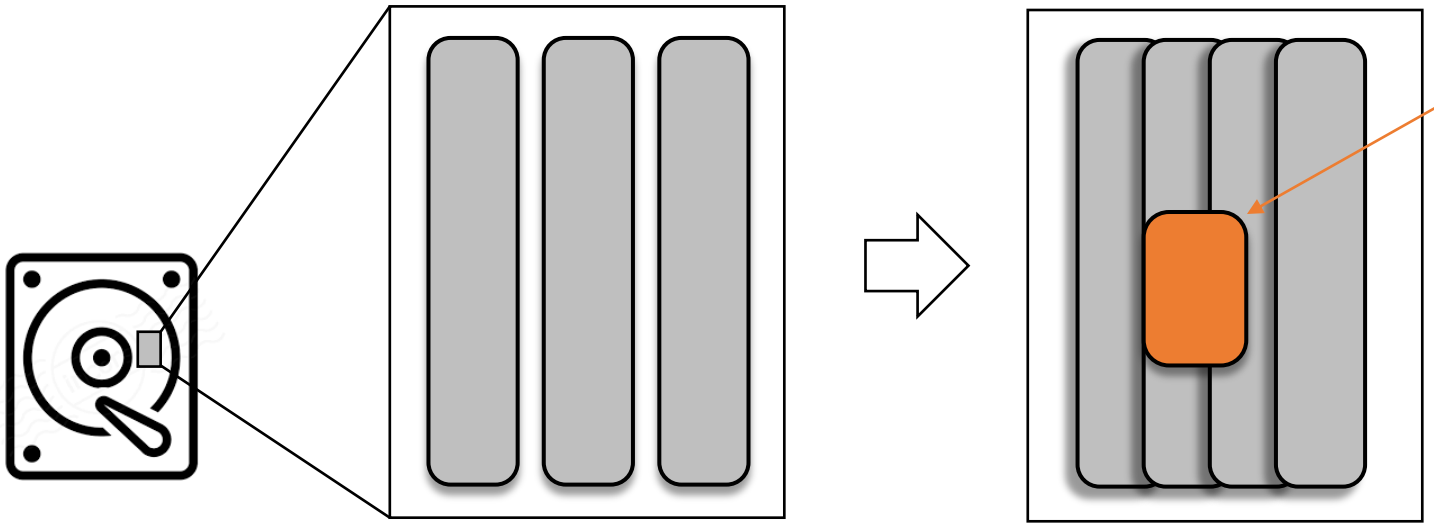
ZoneAlloy: Elastic Data and Space Management for Hybrid SMR Drives

Fenggang Wu, Bingzhe Li, Zhichao Cao, Baoquan Zhang
Ming-Hong Yang, Hao Wen, David H.C. Du
University of Minnesota, Twin Cities

Jul. 8, 2019. HotStorage'19

Center for Research in Intelligent Storage

University of Minnesota
Driven to Discover℠

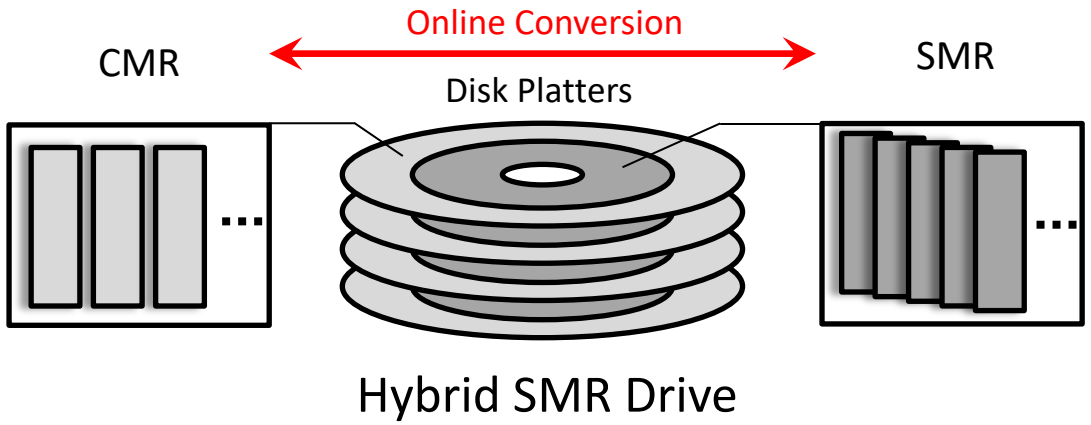


SMR: (+) more data density; (-) update overhead

CMR: (-) less data density; (+) no update overhead.

How about a **combination** of the two?

Emerging Hybrid SMR Drives



- **Hybrid SMR (H-SMR)**: mix of CMR and SMR; can be converted on line by H-SMR API.
- **Benefit**: utilize both IOPS and Capacity. Flexible and reconfigurable.

Objective: How do we efficiently manage the data and space for such Hybrid SMR drives?

Outline

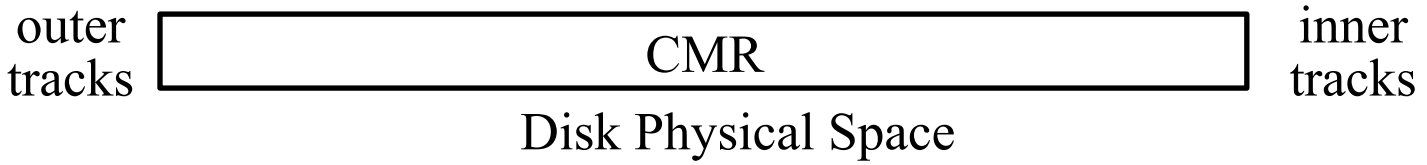
- Introduction
- Design Goals
- Background and Challenges
- Design and Evaluation
- Summary

Design Goal

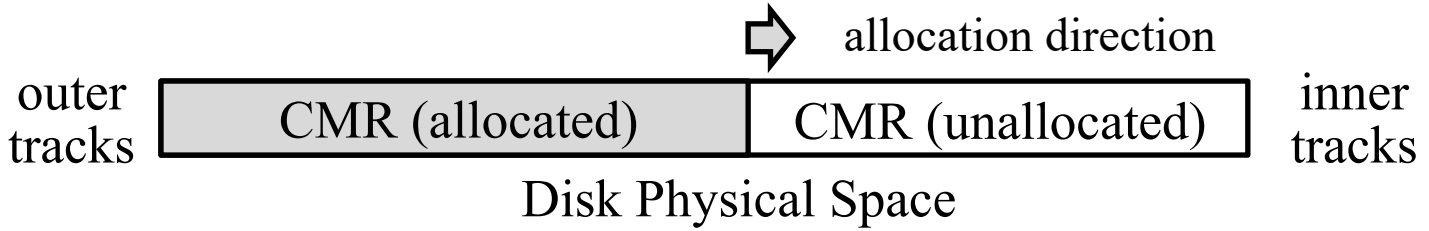
- Handle growing utilization.
- Reduce SMR update overhead.
- Adapt to dynamic workload.

Two-Phase Allocation

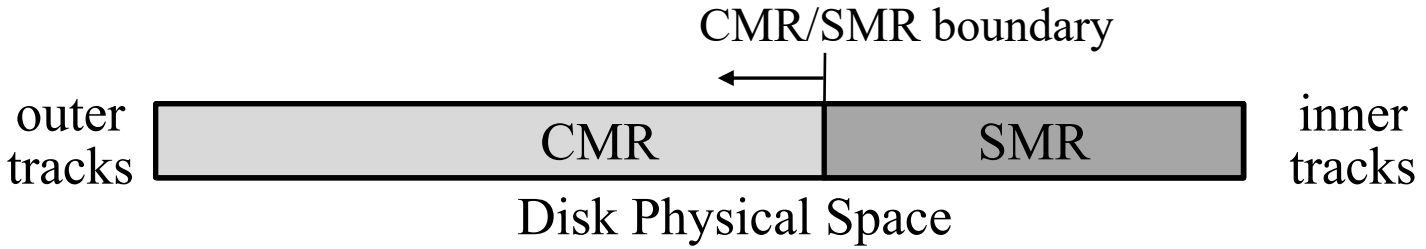
Initial State: Empty Disk



Phase I: CMR-only



Phase II: CMR + SMR

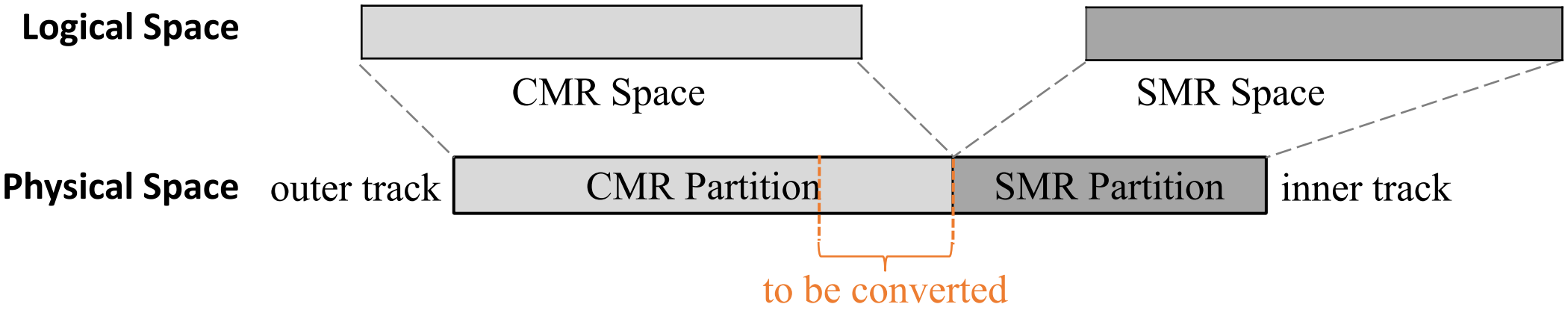


Intuition: use CMR first! then convert CMR to SMR when capacity is not enough.

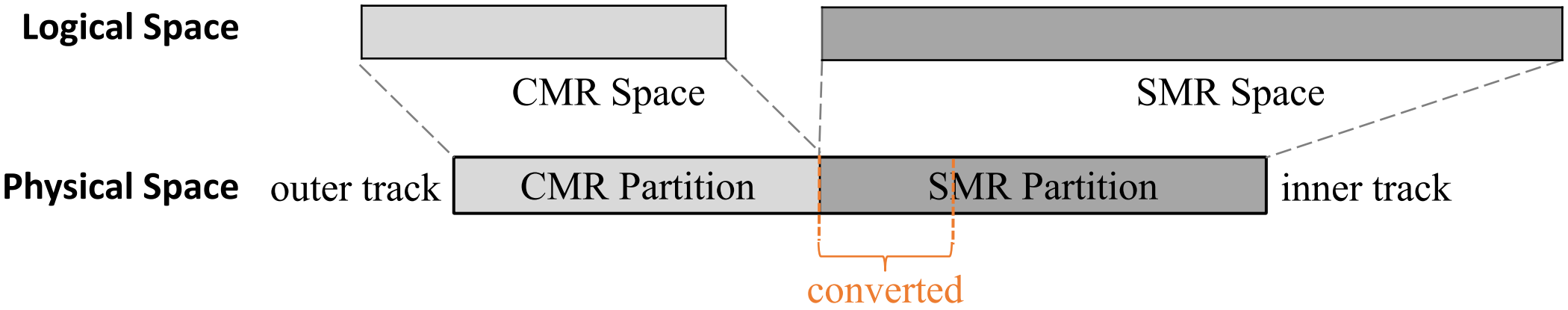
The devil is in the detail!

Buckle up. Challenge ahead!

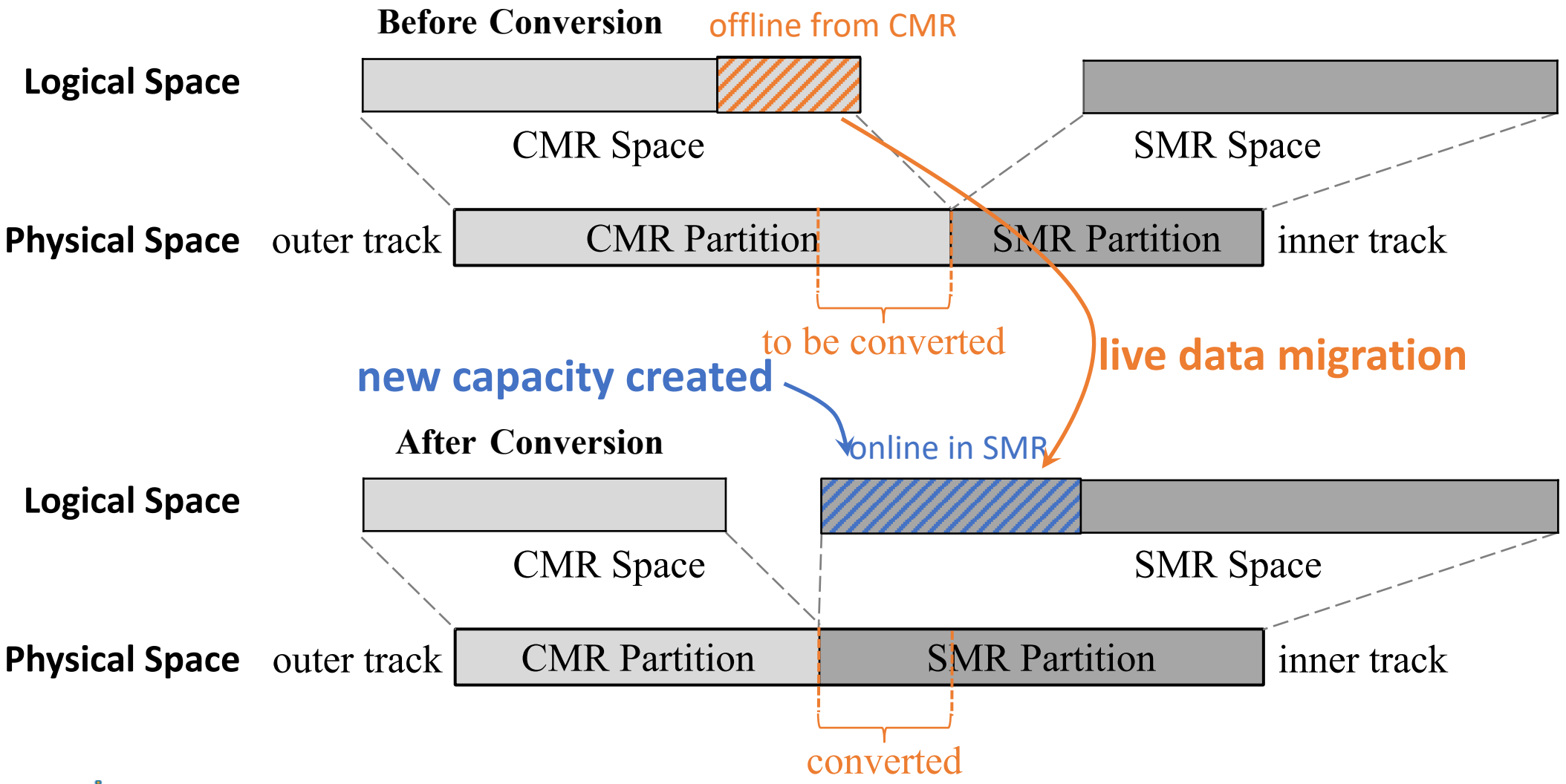
Background: Format Conversion



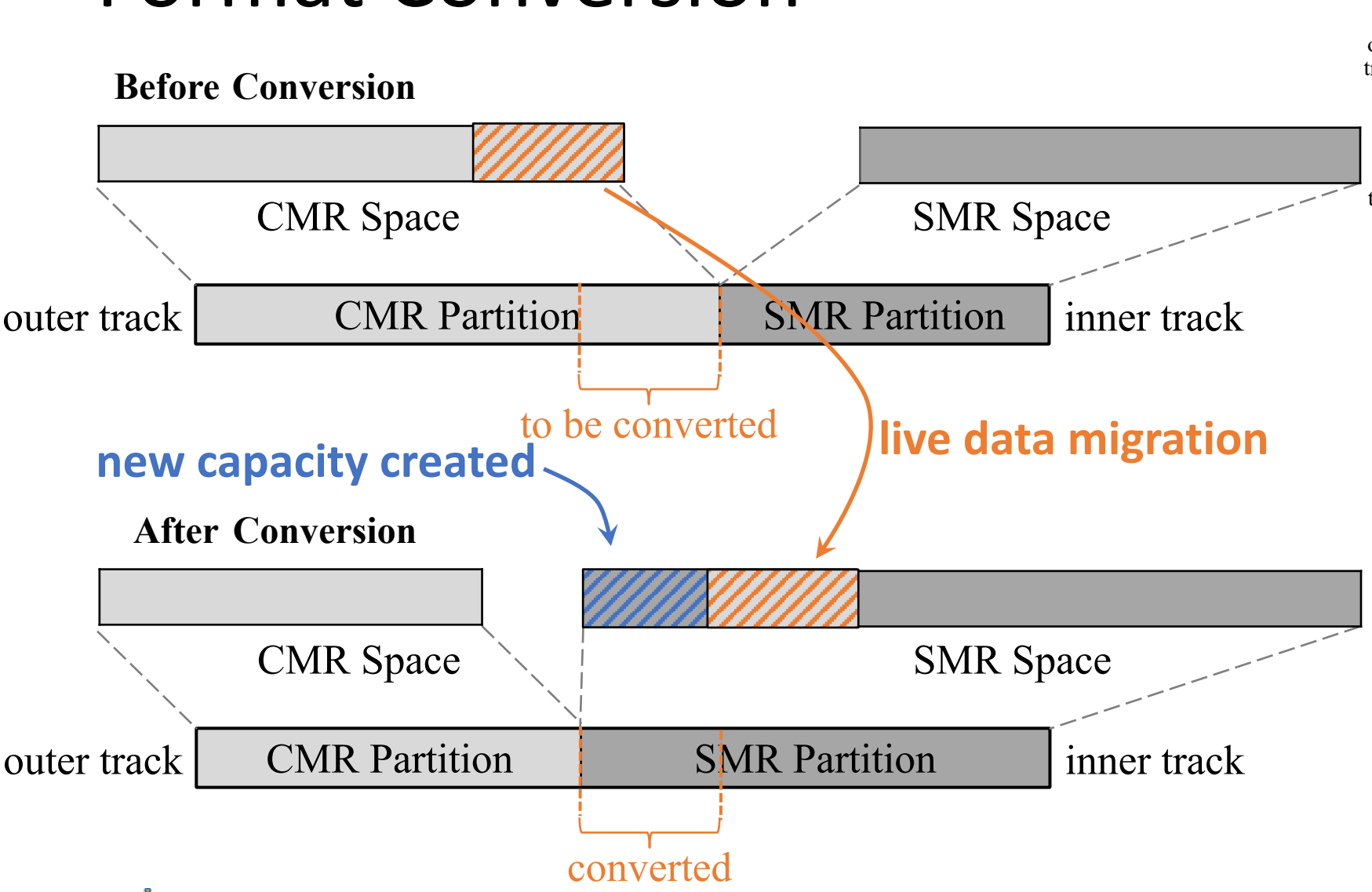
Background: Format Conversion

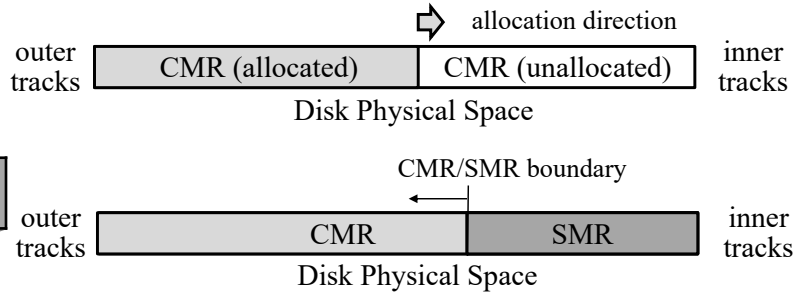


Background: Format Conversion



Format Conversion





Challenge:

- Address space / mapping
- SMR update overhead
- Live data migration cost

Challenges and Solutions

Challenge:
- Address space / mapping
- SMR update overhead
- Live data migration cost

Solution:
- **Elastic** Address Space with Zone-level **Mapping**
- **H-Buffer** and **Zone-Swap** to reduce SMR update overhead
- **Quantized Migration** to mitigate live data migration cost in the format conversion

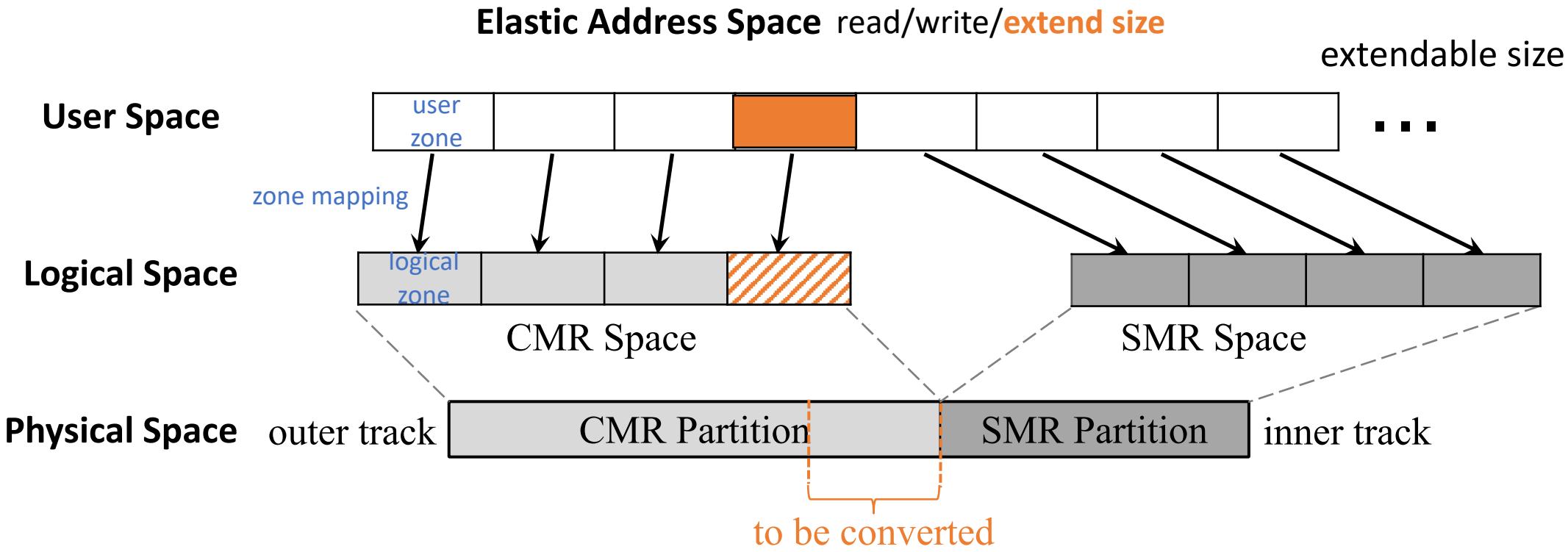
Challenges and Solutions

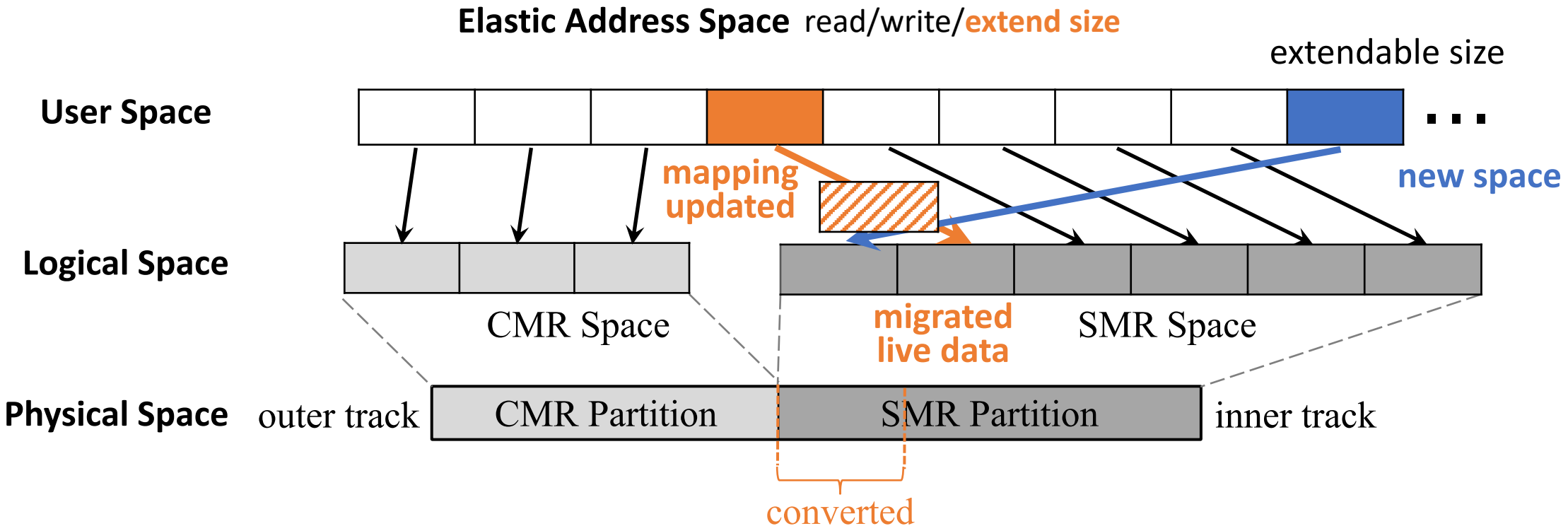
Challenge:
- Address space / mapping
- SMR update overhead
- Live data migration cost

Solution:
- **Elastic** Address Space with Zone-level **Mapping**
- **H-Buffer** and **Zone-Swap** to reduce SMR update overhead
- **Quantized Migration** to mitigate live data migration cost in the format conversion

DESIGN I: ADDRESS SPACE / MAPPING

Elastic Address Space with Zone-level Mapping

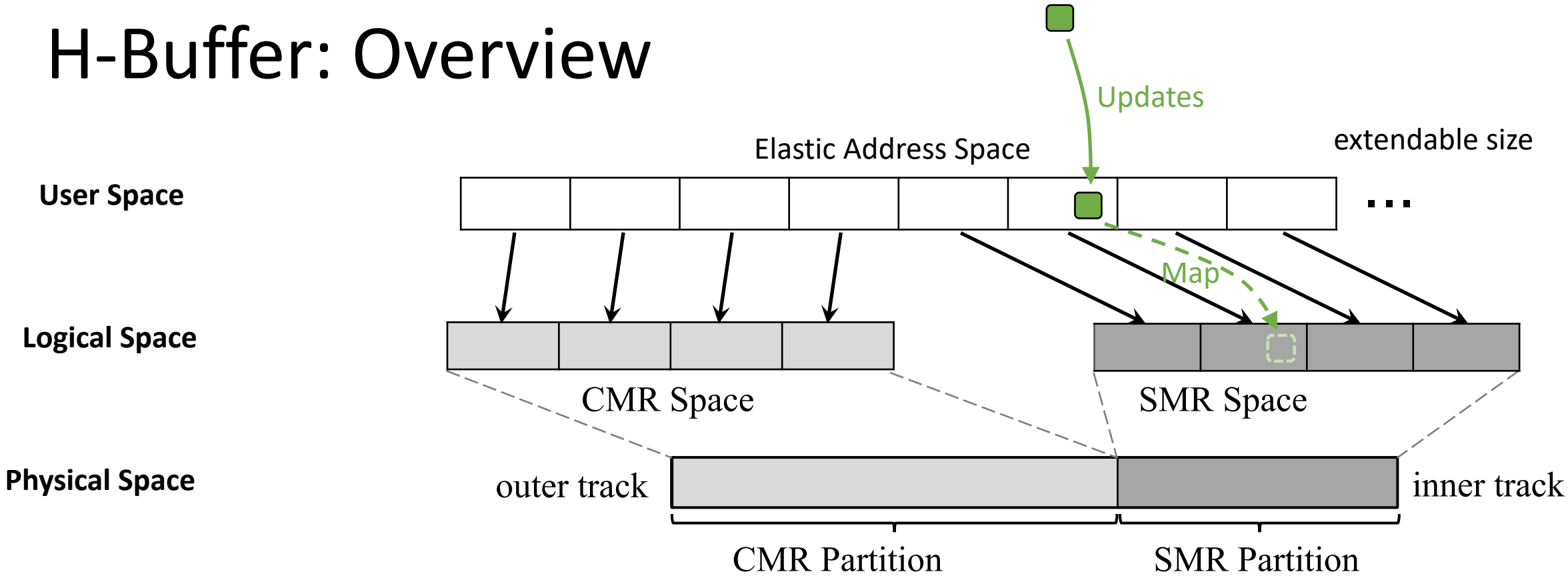

Elastic Address Space read/write/extend size
extendable size
User Space
user zone
zone mapping
Logical Space
logical zone
CMR Space
SMR Space
Physical Space
outer track
CMR Partition
SMR Partition
inner track
to be converted


Elastic Address Space read/write/extend size
extendable size
User Space
mapping updated
new space
Logical Space
CMR Space
migrated live data
SMR Space
Physical Space
outer track
CMR Partition
SMR Partition
inner track
converted

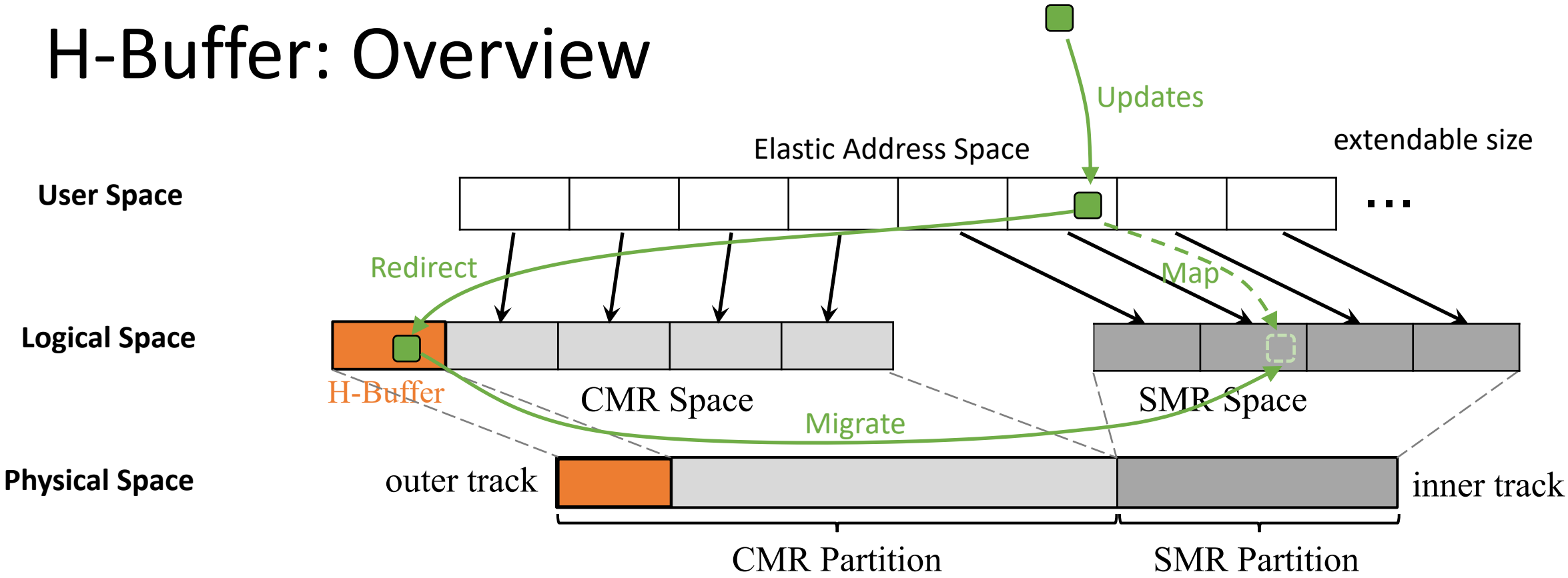
DESIGN II: REDUCING SMR UPDATE OVERHEAD

H-Buffer and Zone-Swap

H-Buffer: Overview



H-Buffer: Overview

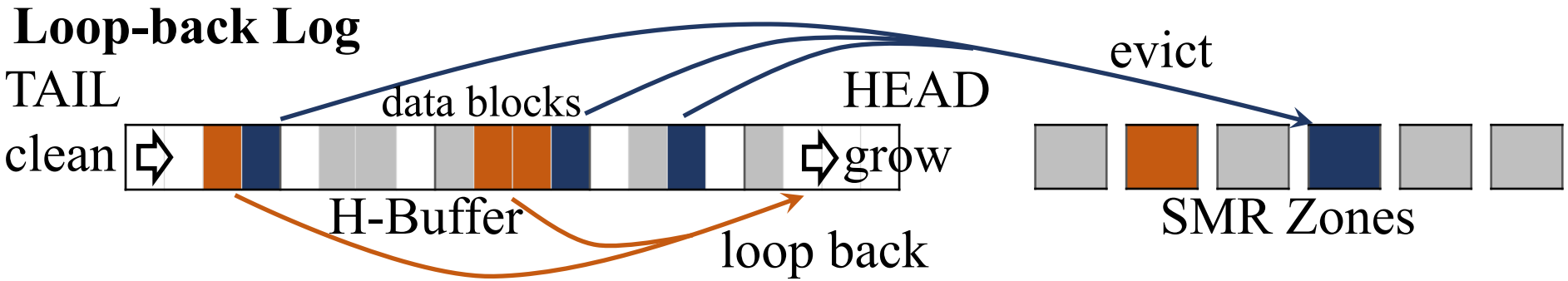


H-Buffer: Host-controlled Buffer

Basic Idea: using some reserved CMR space to buffer SMR updates and migrate to SMR zones later.

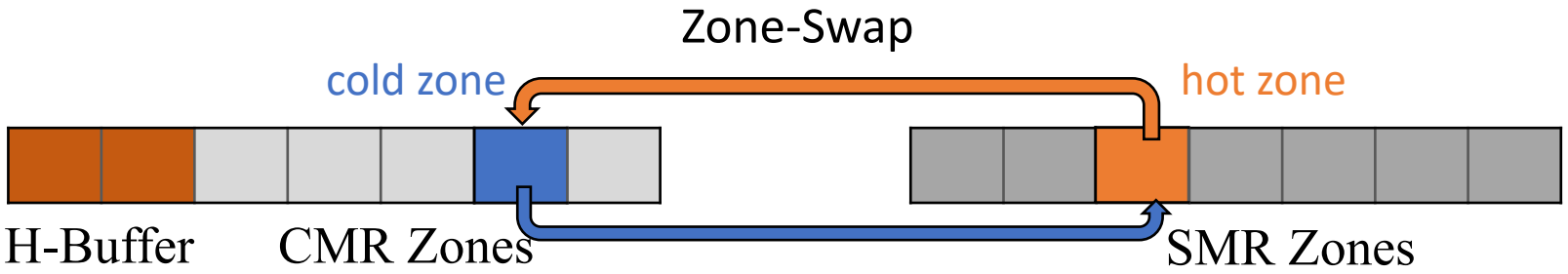
H-Buffer Management: Alternatives

- Block-based (e.g. LRU) Problem: Random I/O in redirecting/cleaning
- Log-based
  - In-place FIFO
  - Loop-back Log (with hot/cold classification)



Idea: **Re-queue the hot data blocks** to the log head without evicting to SMR zones.

Zone-Swap: Overview

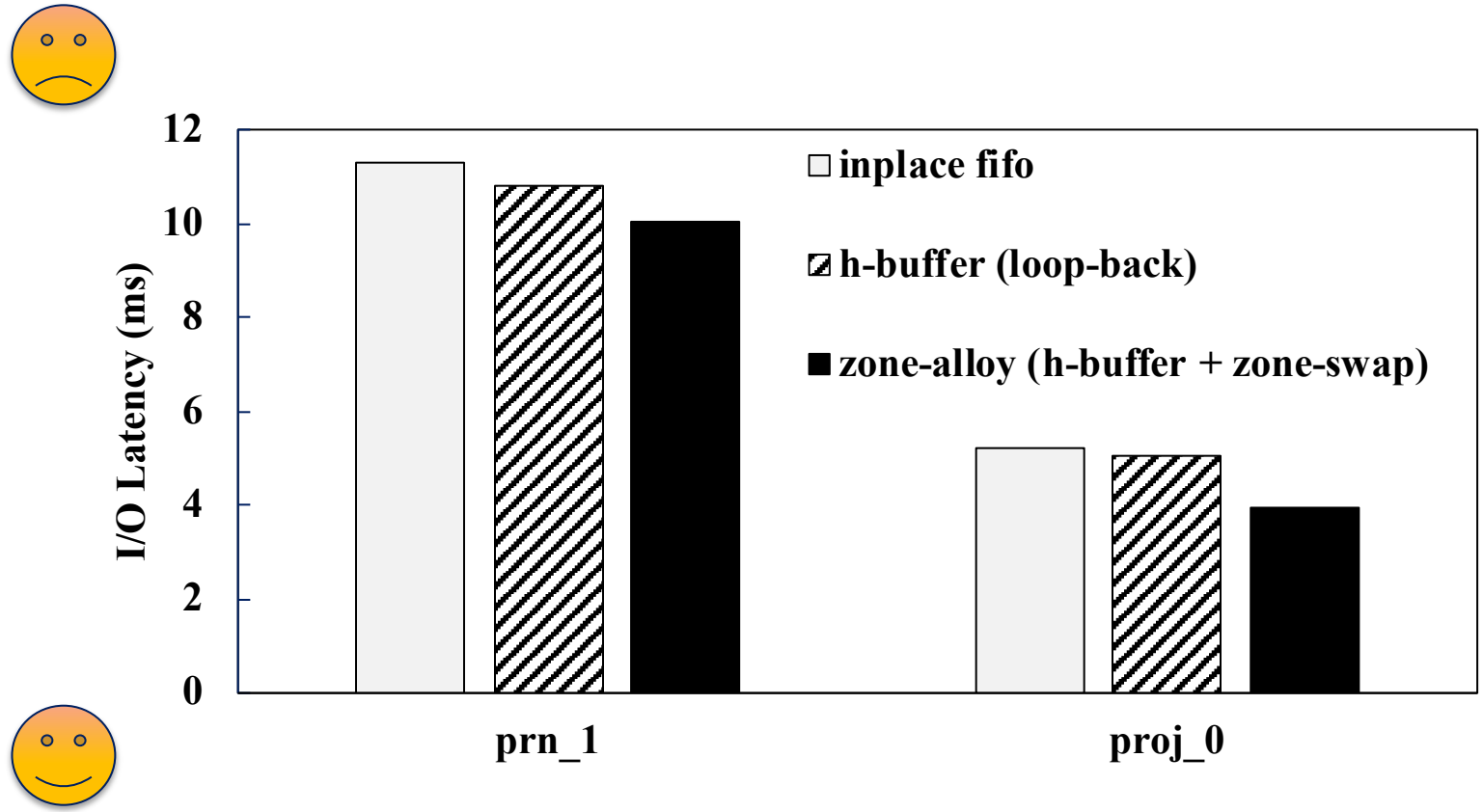


Co-design with H-Buffer:

- Swapping happens when H-Buffer evicts.
- H-Buffer eviction choice also depends on Zone-Swap decision.

Basic Idea: **swap** hot zones (heavily updated ones) from SMR to CMR to reduce SMR update overhead.

H-Buffer and Zone-Swap Evaluation

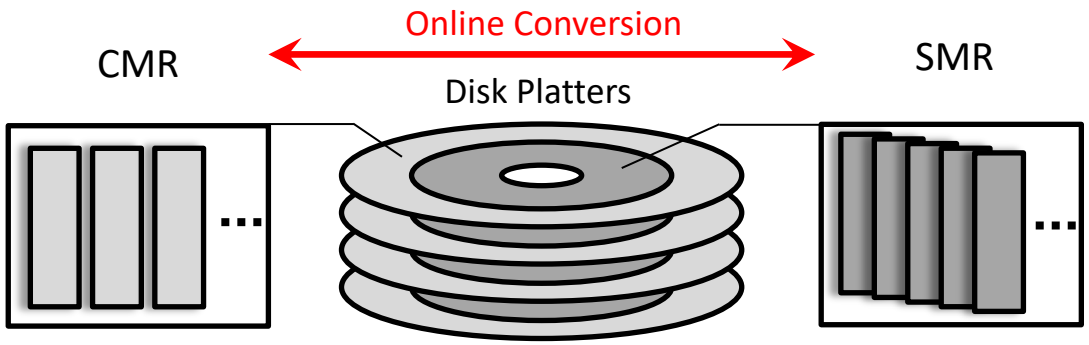


More result: in the poster session.

Summary: ZoneAlloy

-- Data management for Hybrid-SMR

Background



Hybrid SMR Drive

Problem

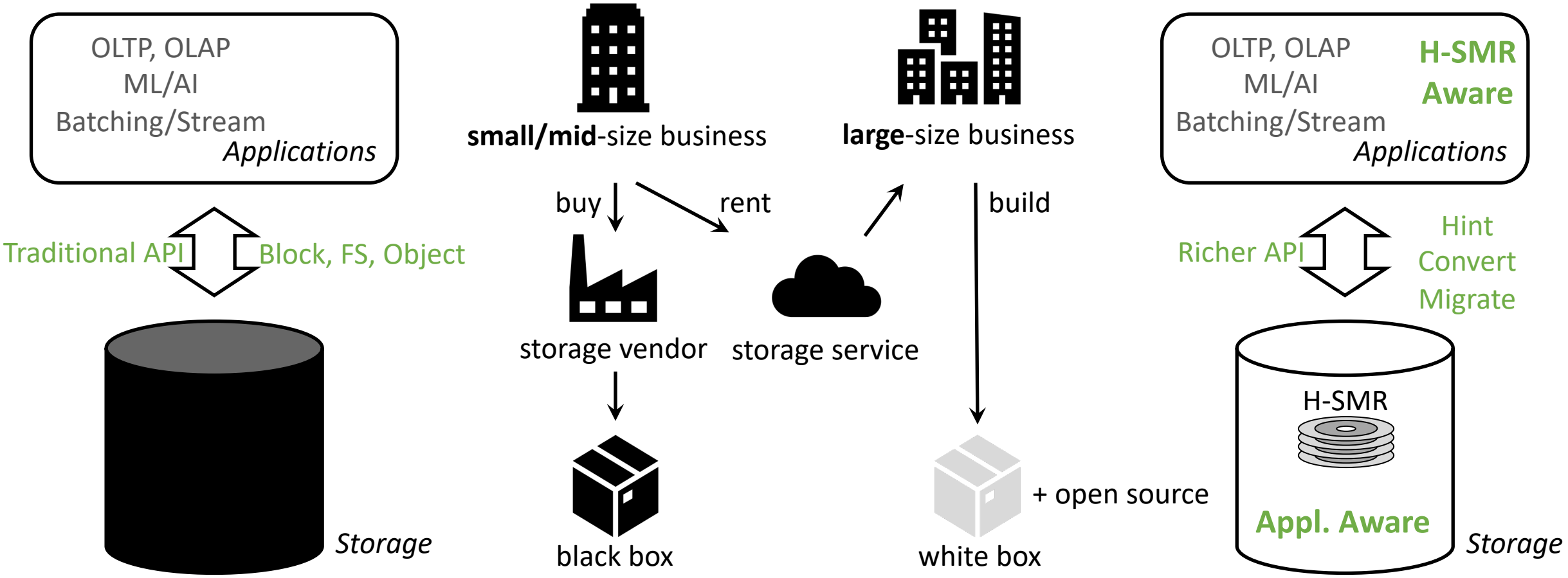
Data and Space Management in Hybrid SMR drives

Solutions/Contributions

- Two-Phase Allocation
- Elastic Address Space/ Zone-level Mapping
- H-Buffer
- Zone-Swapping
- Quantized Migration

I/O stack change & API (ask for feedback)

Question: Which layer(s) of the I/O stack should do the heavy-lifting?



Thank You!
Questions?

Fenggang Wu
wuxx0835@umn.edu

