HITACHI
Inspire the Next

Indirection Systems for Shingled-Recording Disk Drives

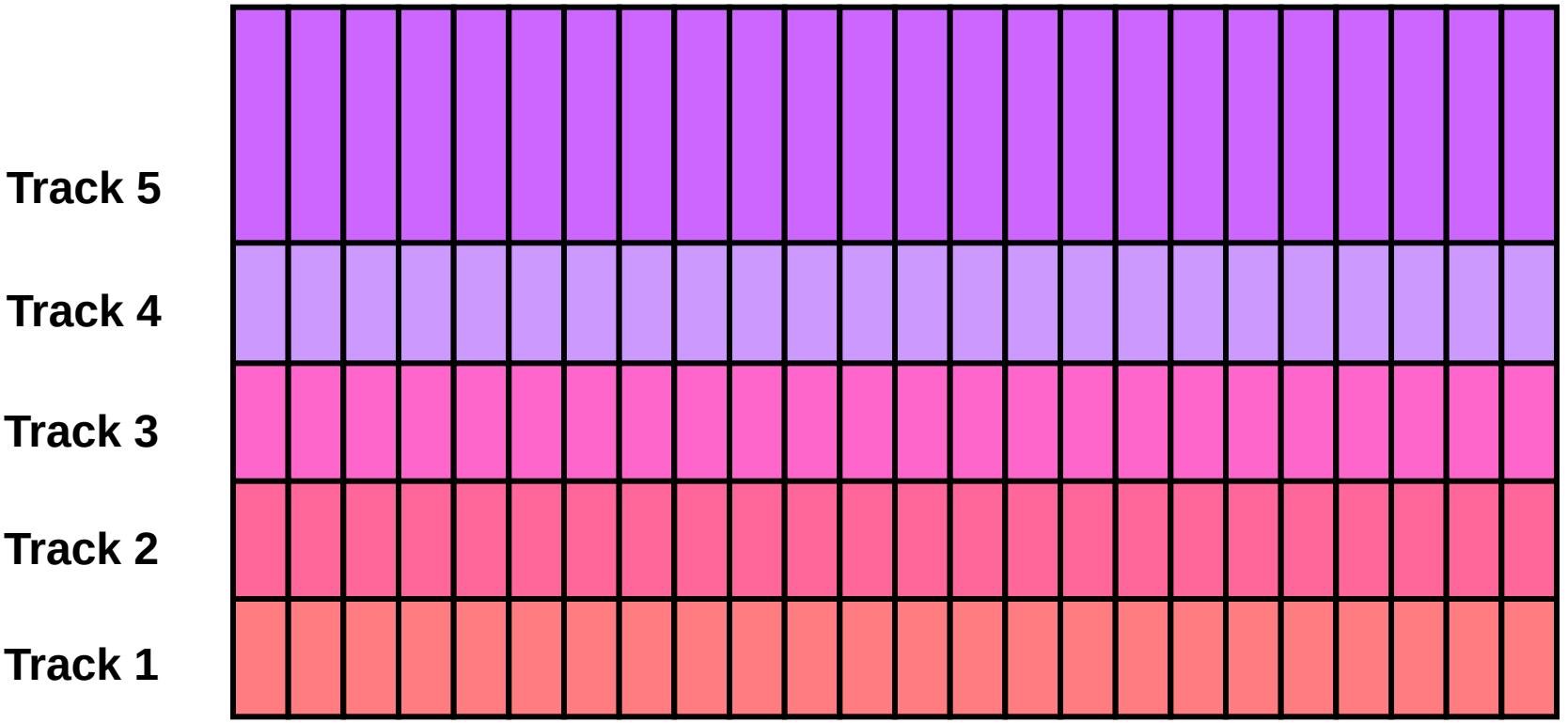
Yuval Cassuto

with co-authors:
M. Sanvido, C. Guyot, D. Hall and Z. Bandic

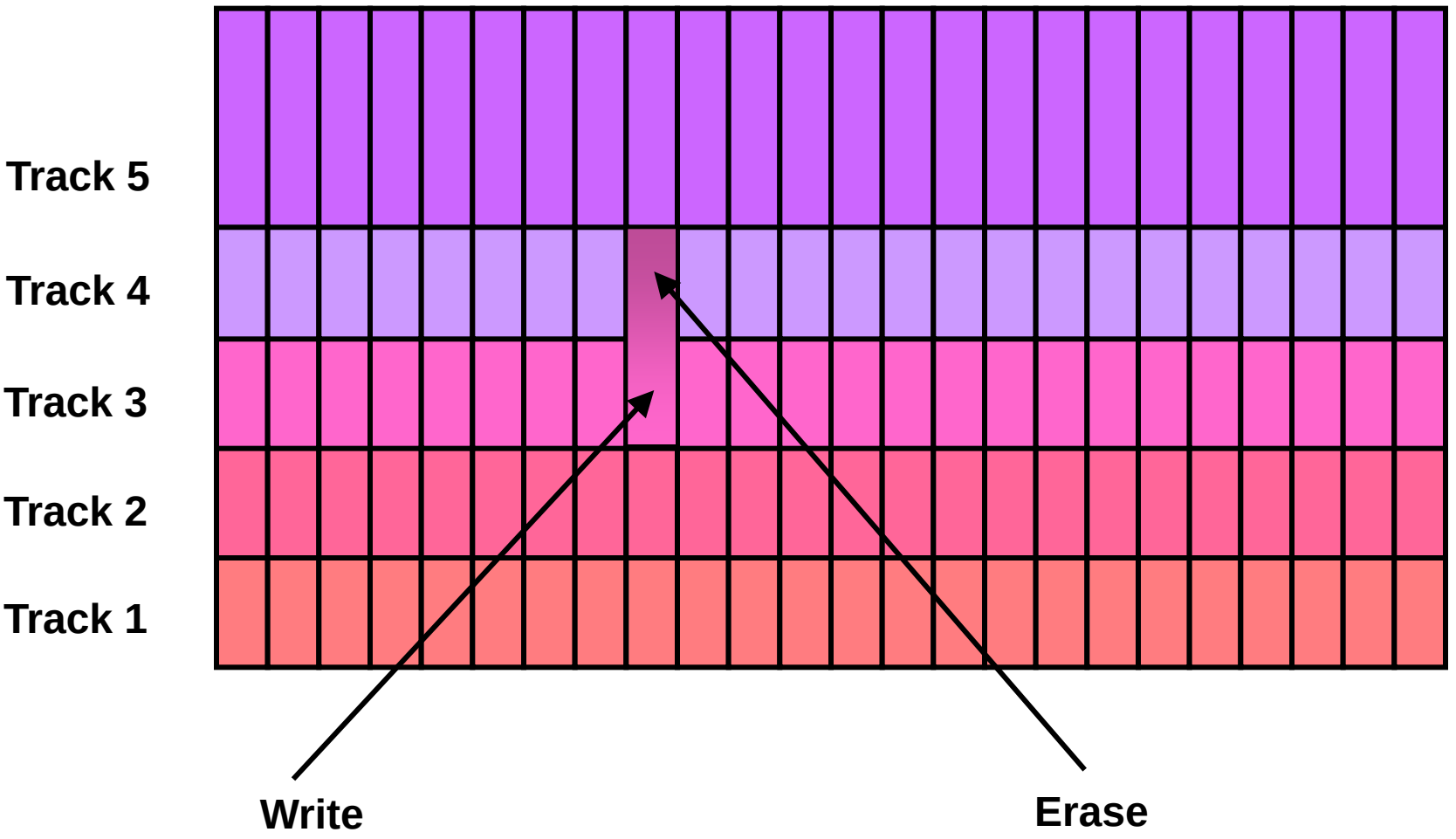
Hitachi Global Storage Technologies

MSST 2010, May 7

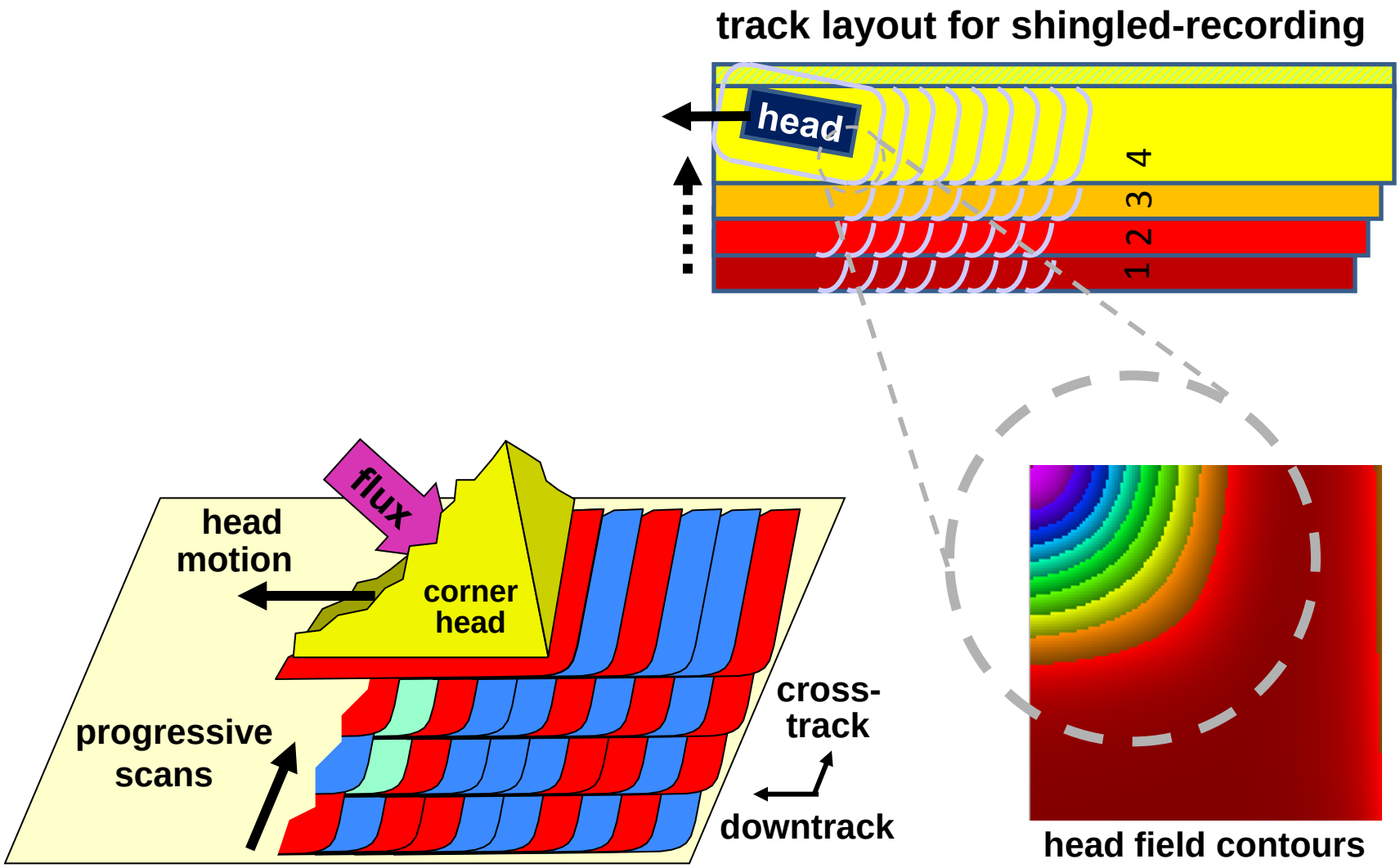
Shingled Recording



Shingled Recording – No Random Write



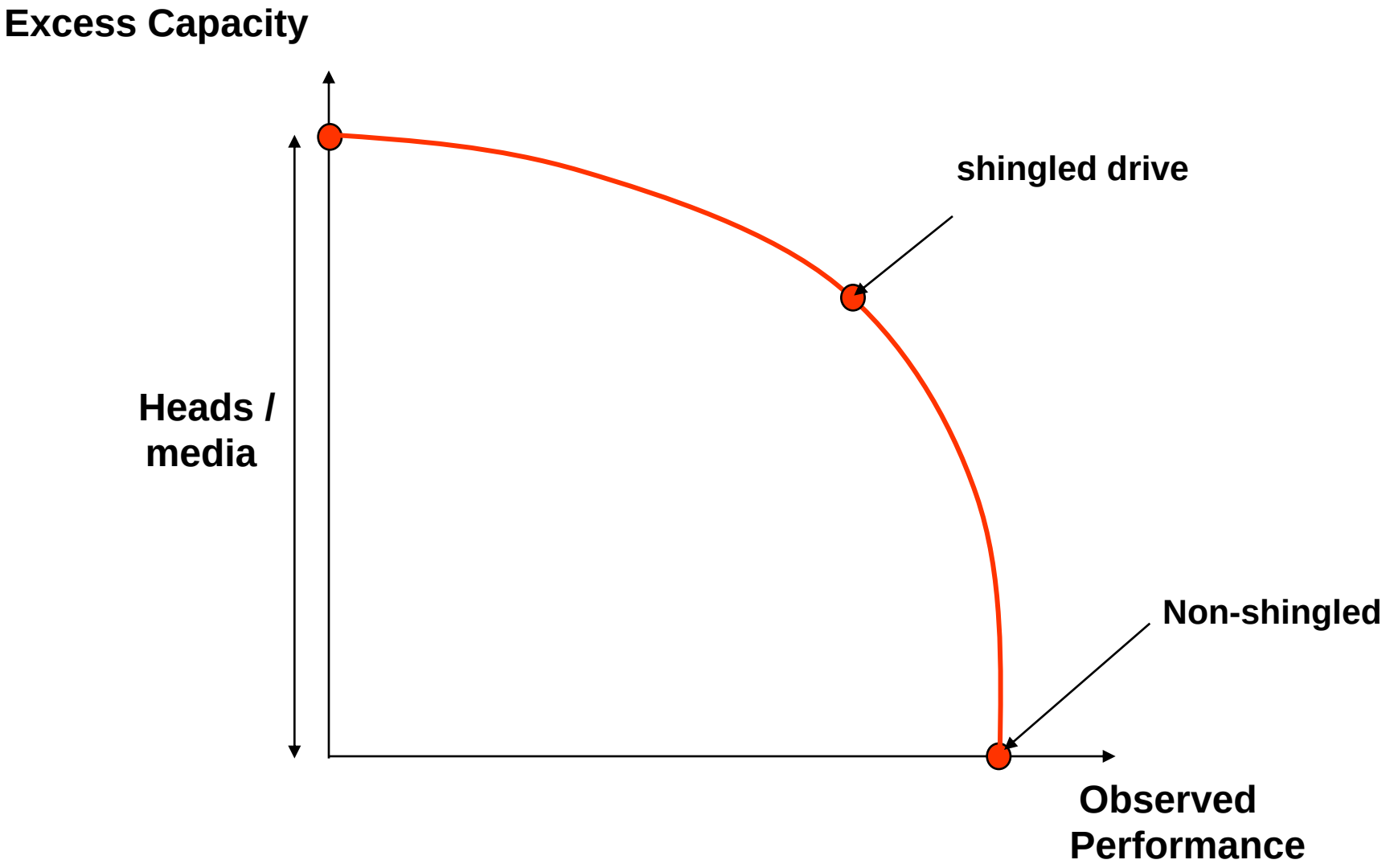
Shingled Recording Magnetics



Shingled Recording Tradeoff

- More: Capacity
- Less: Functionality → Performance

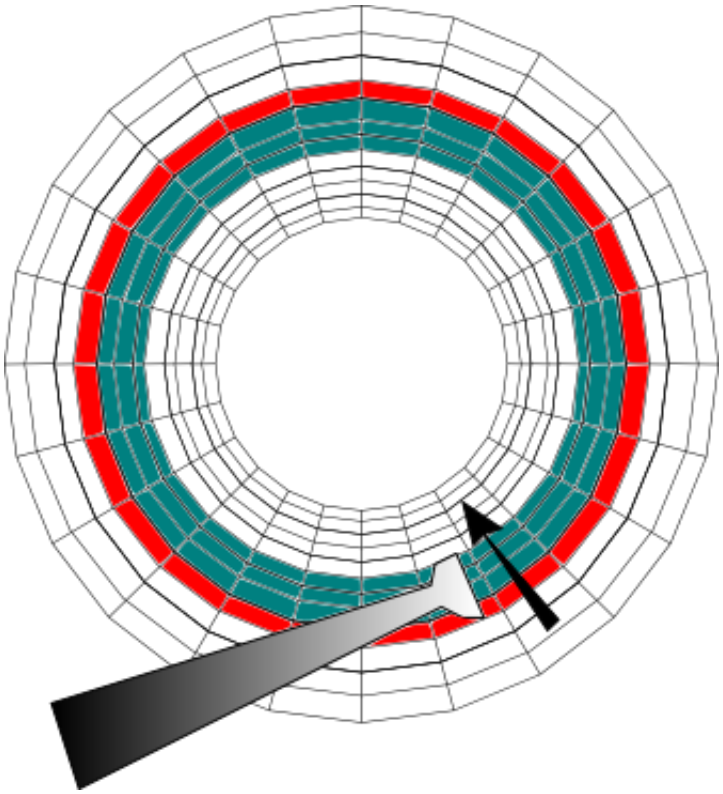
The Optimization Envelope

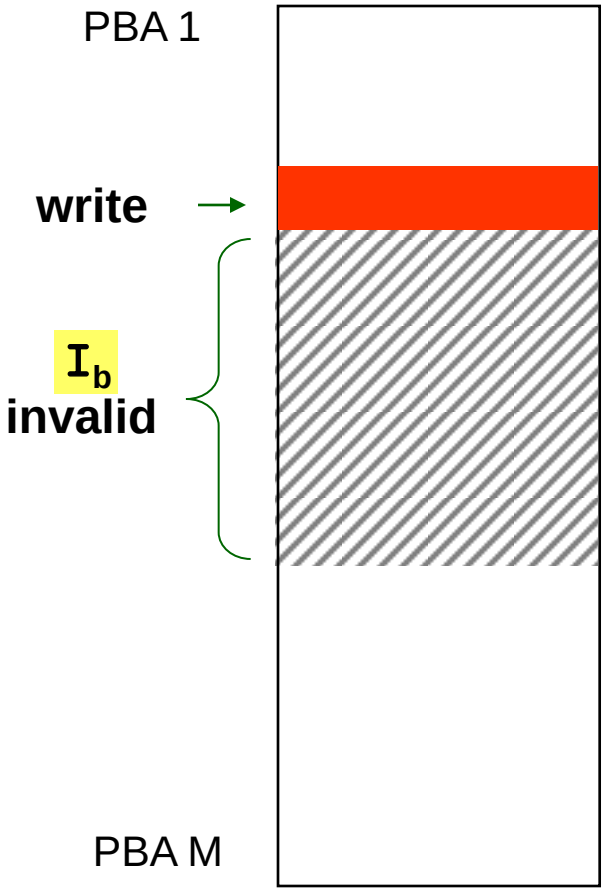


From Magnetics to Systems

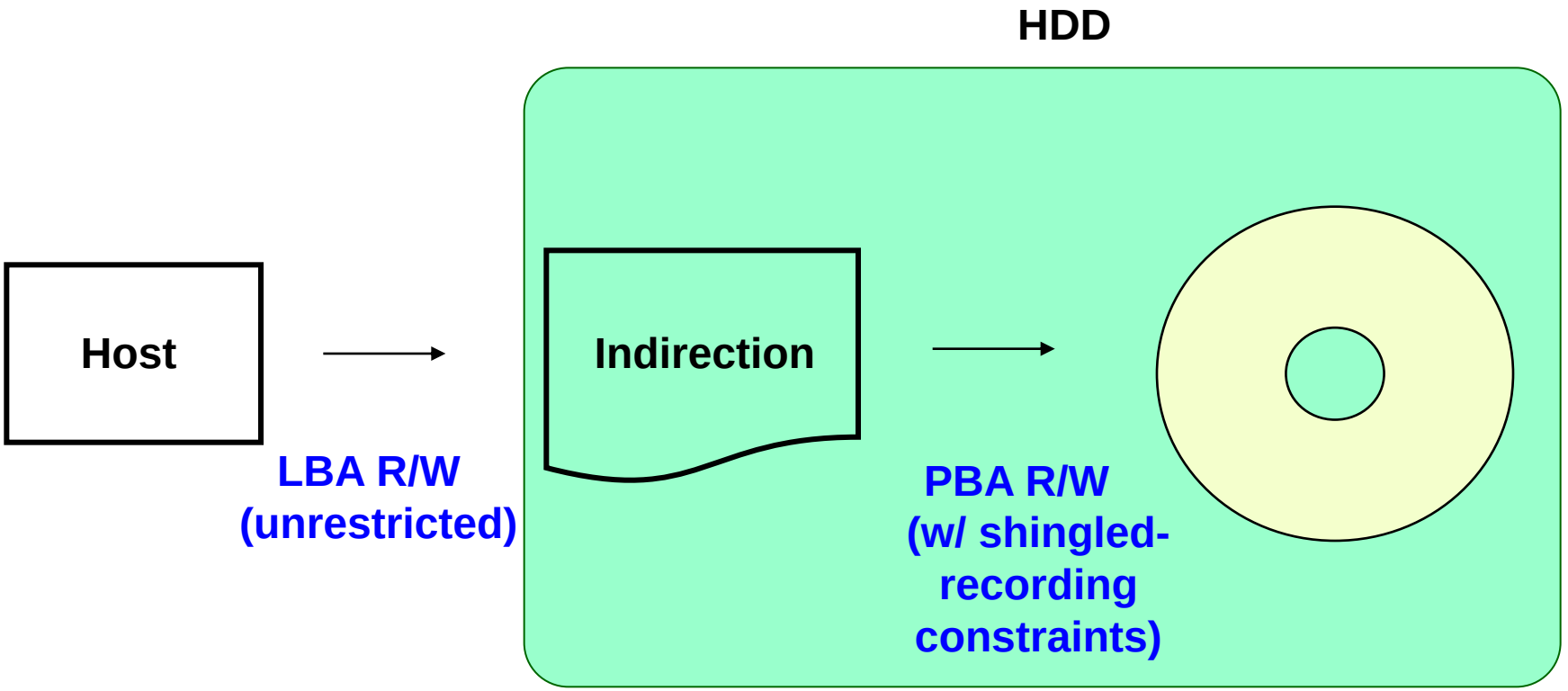
HITACHI
Inspire the Next

Track Interference → Block Interference





Drive Based Shingled Recording Indirection



Drive vs. Host Indirection

HITACHI
Inspire the Next

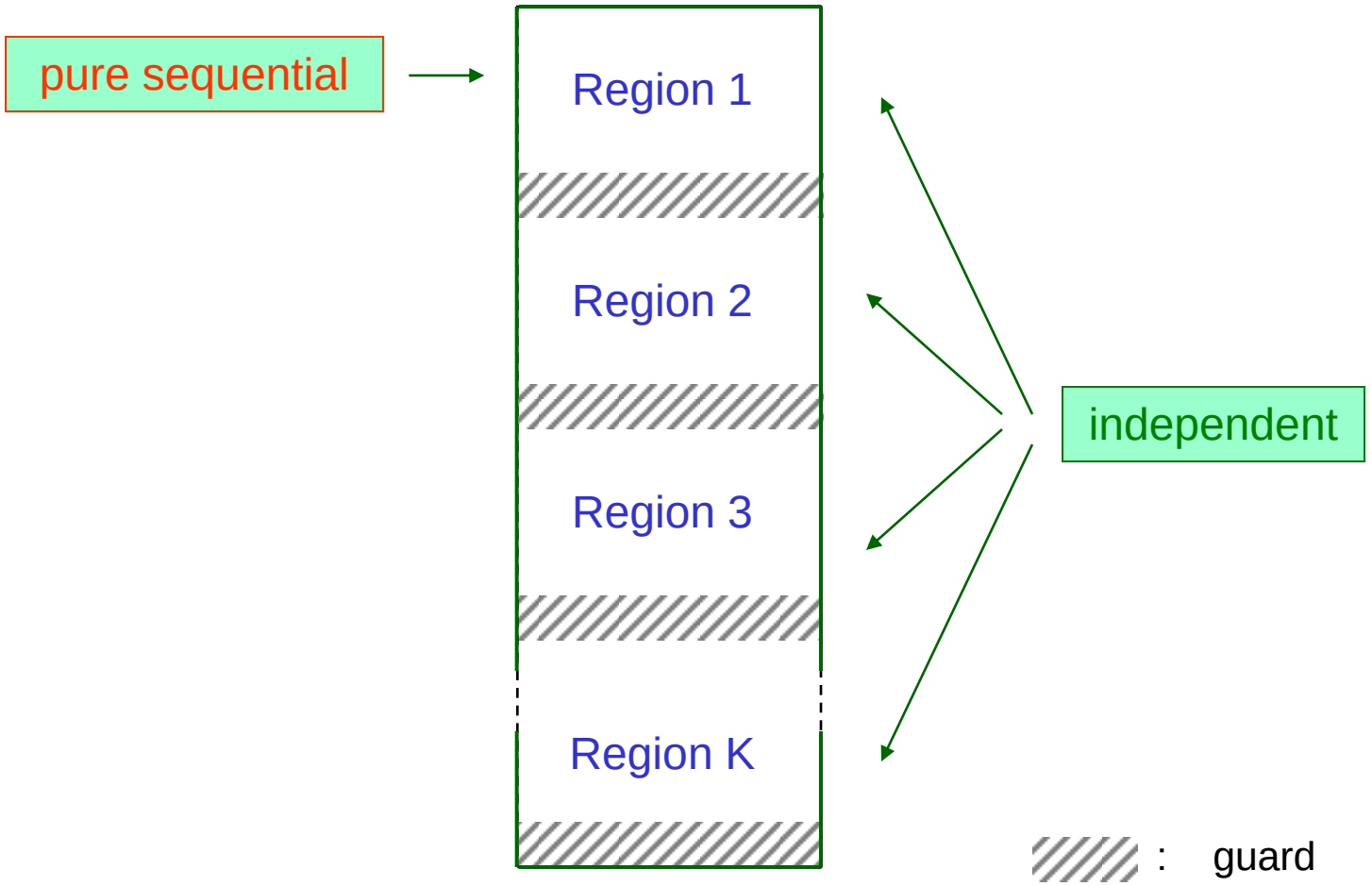
Why on the drive?

- Transparent to Host
- Complete knowledge of physical layout

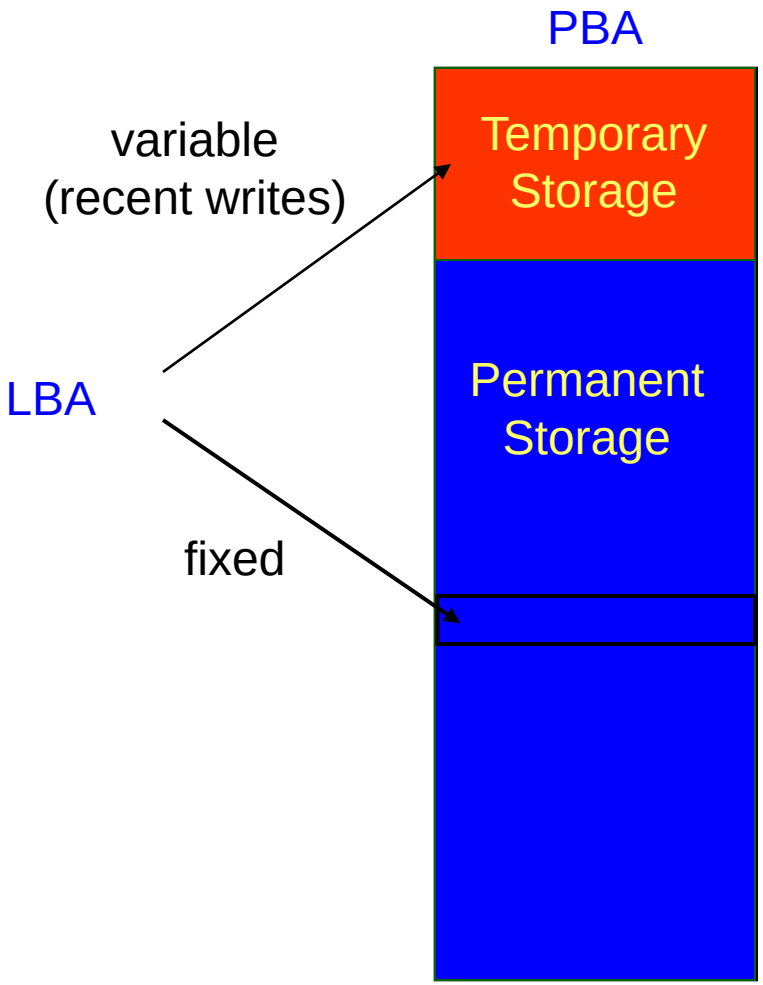
Why on the host?

- “Shingle aware” access and allocation
- System specific performance optimization

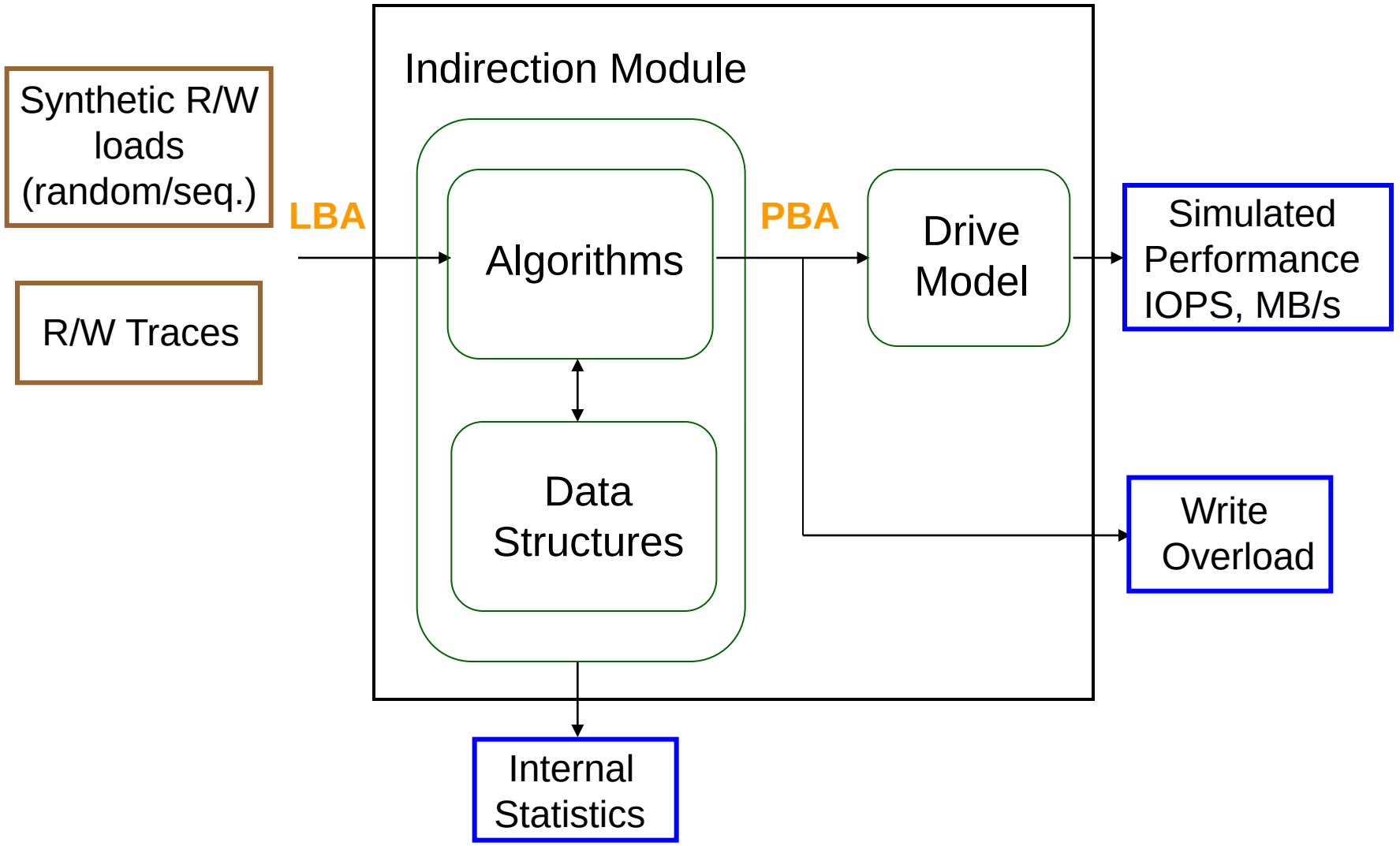
- Append/Read-Modify-Write with Shingled Regions



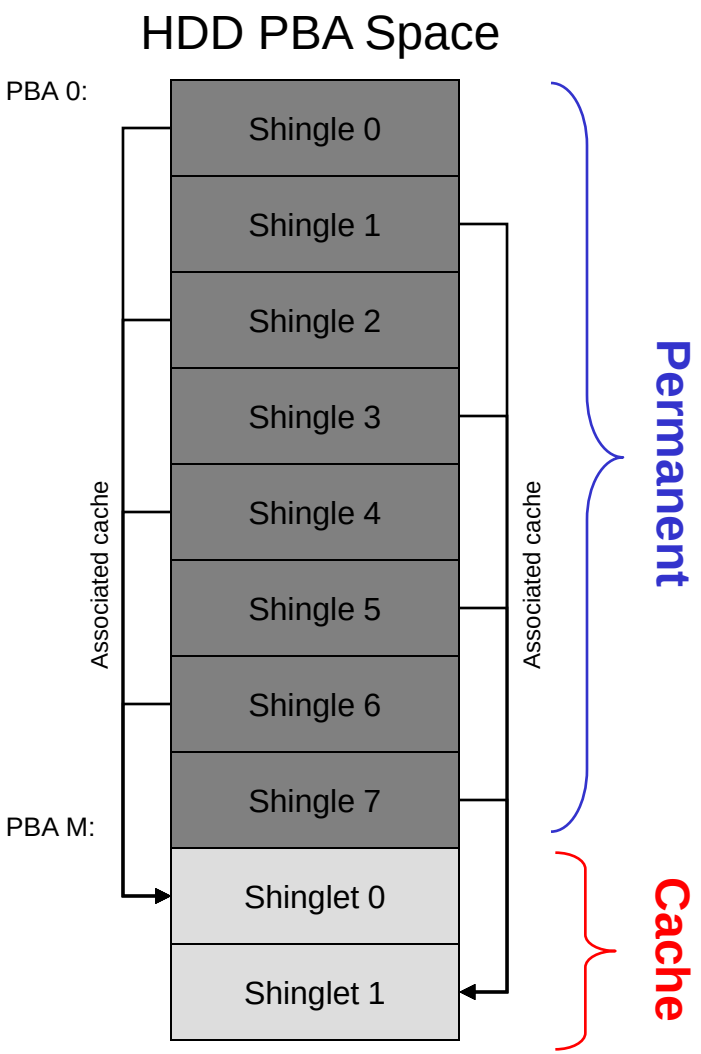
General Method for LBA↔PBA Mapping



Evaluation Platform

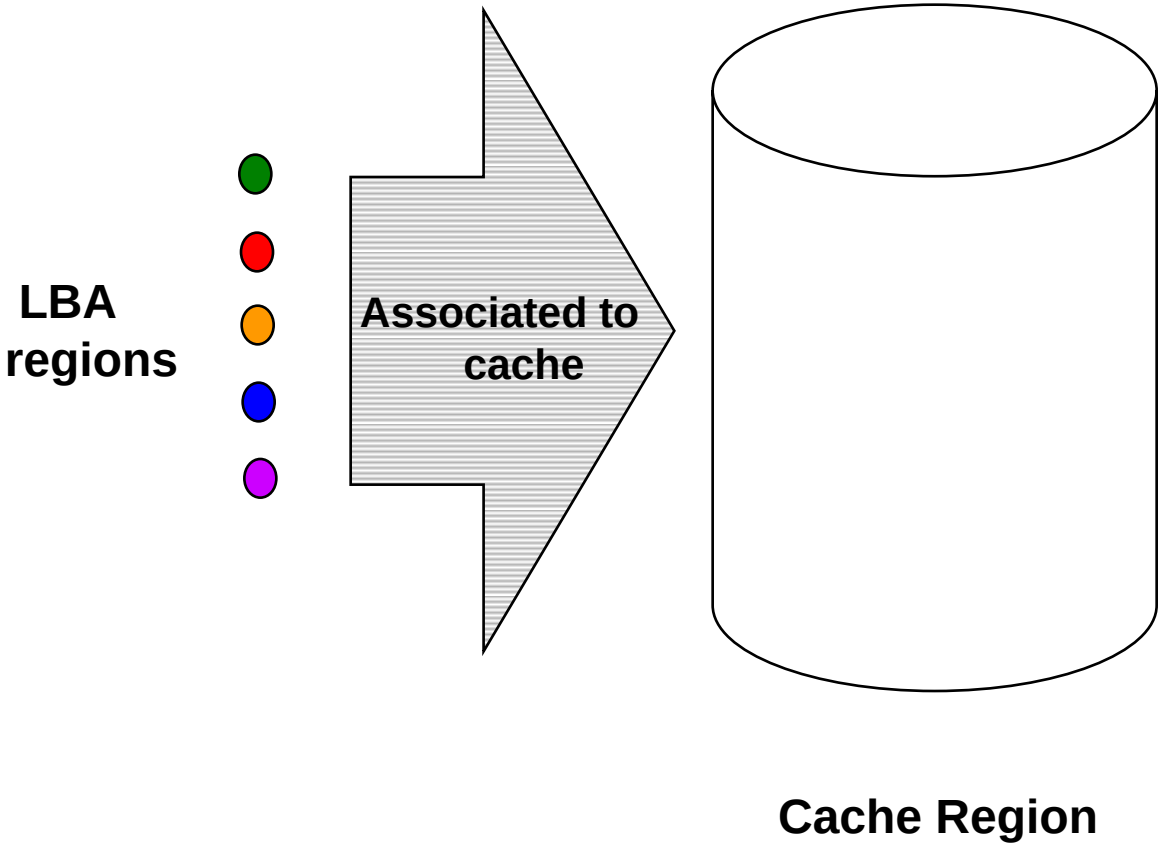


Set-Associative Disk Cache Architecture



- Disk is partitioned to LBA shingled regions and cache shingled regions
- Each LBA region (shingle) is associated with one cache region (shinglet)
- Multiple LBA regions are associated with each cache region (set-associative cache)

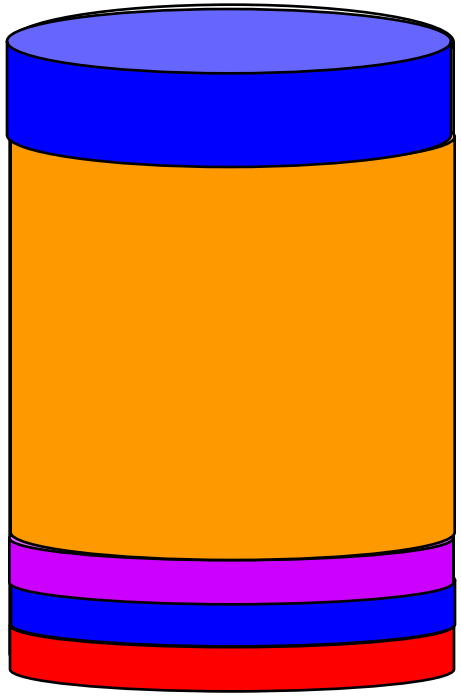
Set-Associative Disk Cache: Write Path



Write to Disk Cache

HITACHI
Inspire the Next

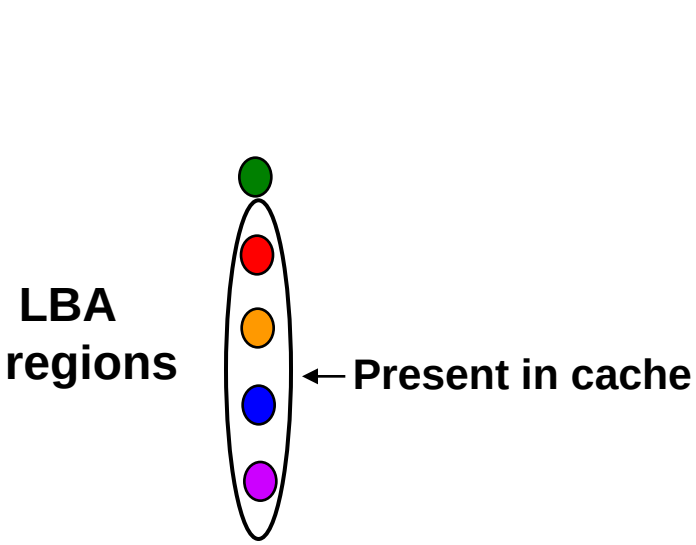
LBA regions

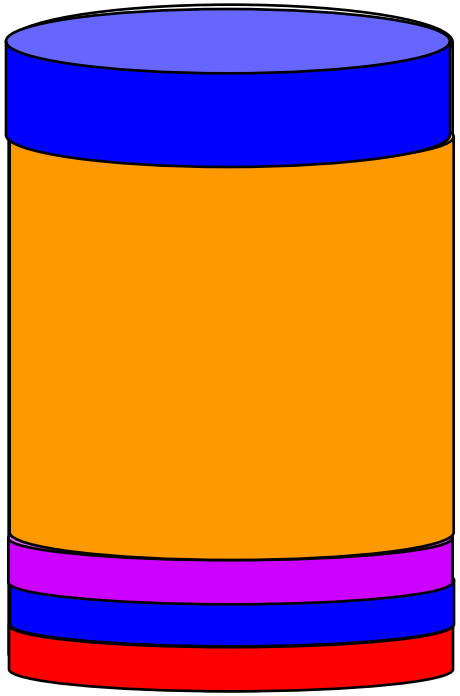


FULL!

Cache Region

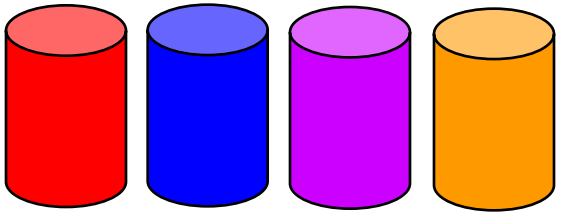
Garbage Collection





Cache Region

- Read cache
- For each LBA region present in cache: RMW* full region

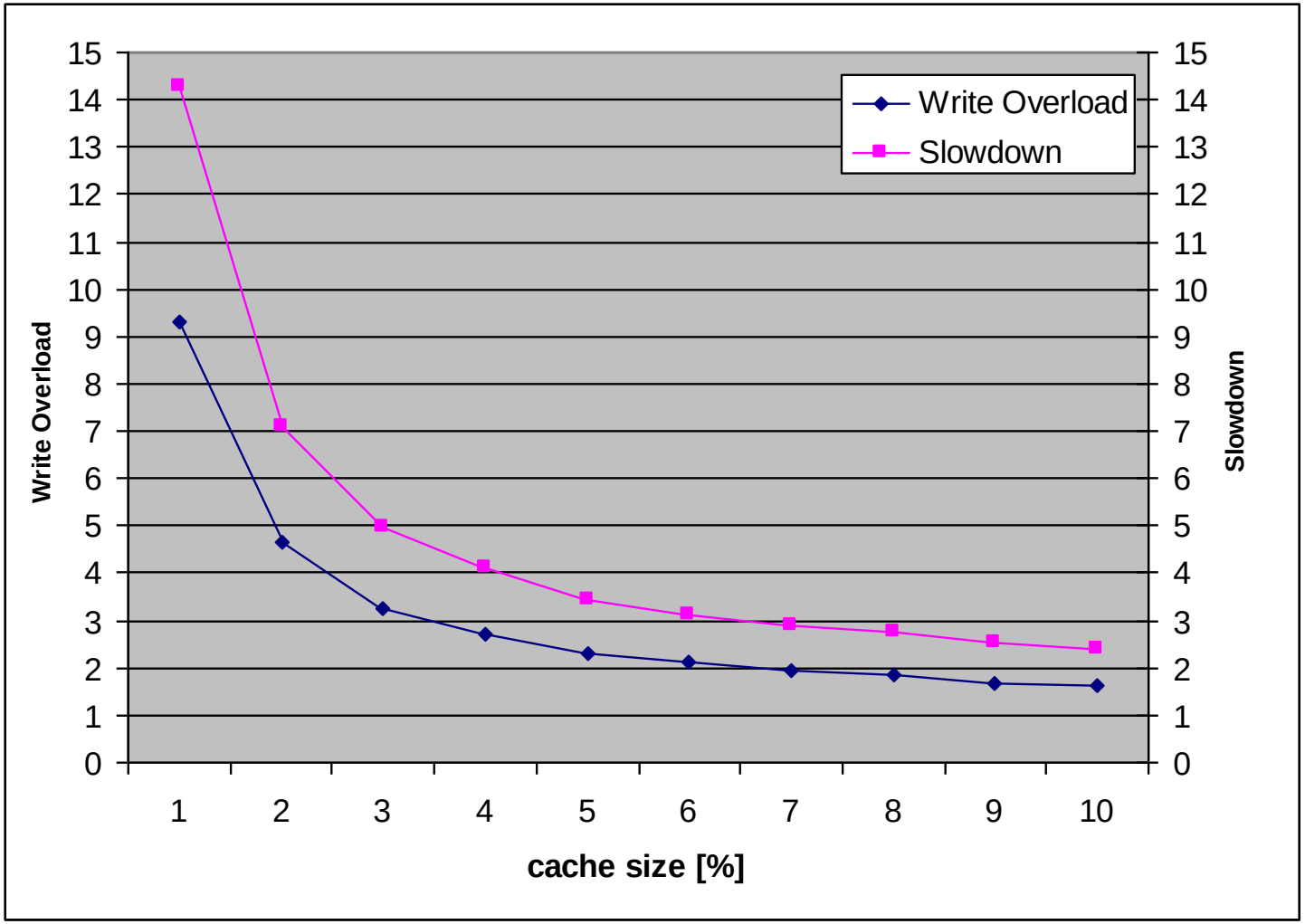


(Read+Write) x #Present

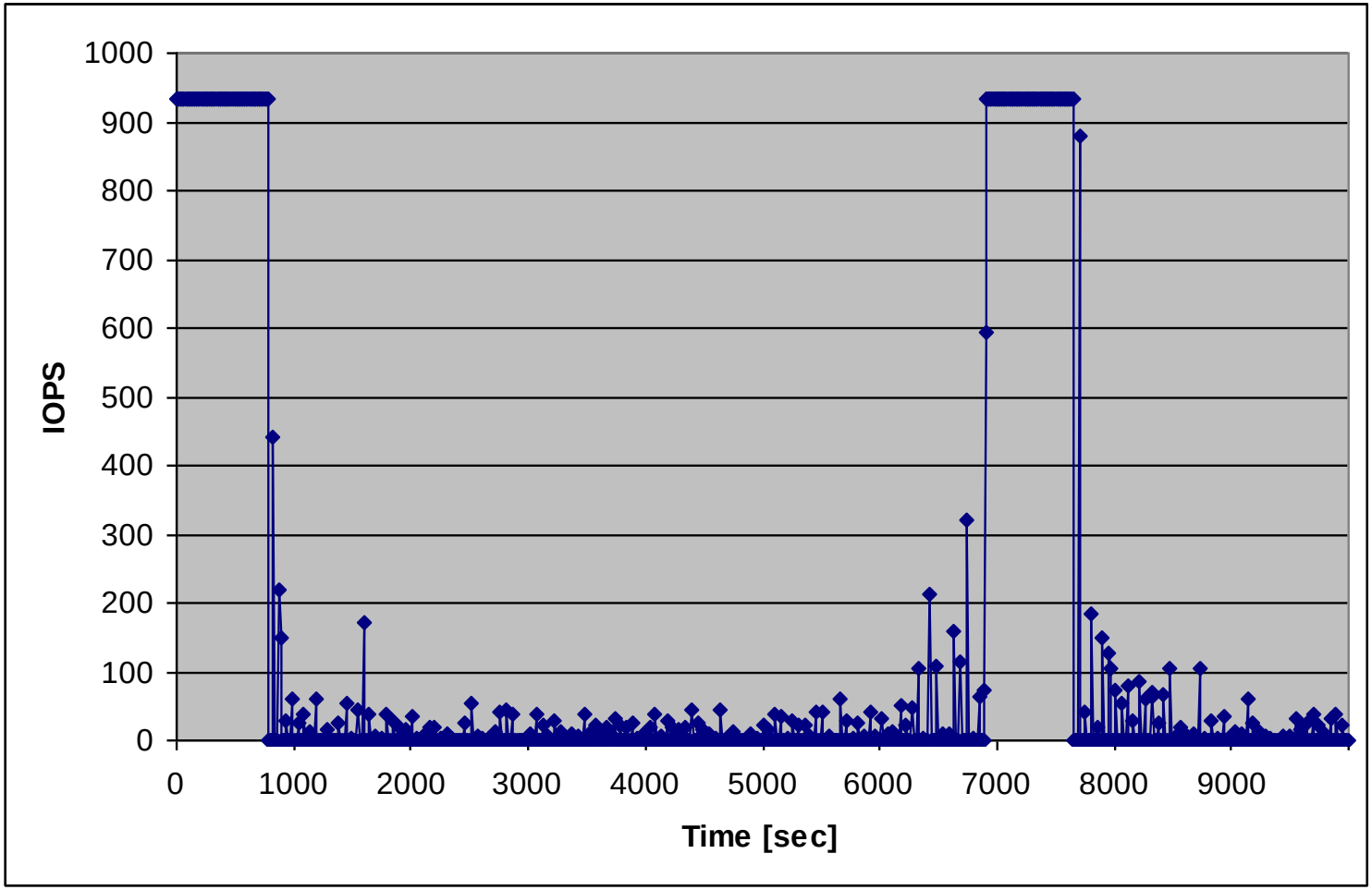
- Full cache available again

*** RMW = Read-Modify-Write**

Results from PC Traces



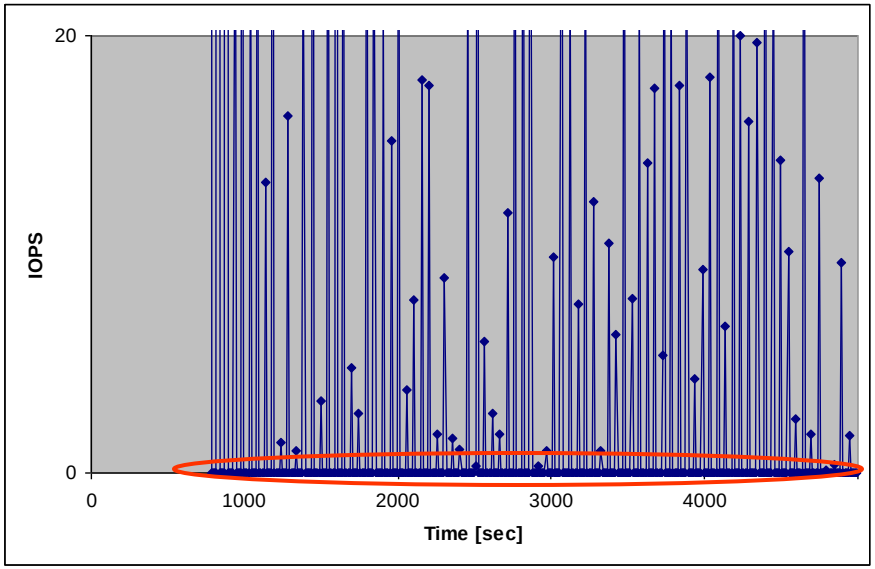
4K Random IOPS



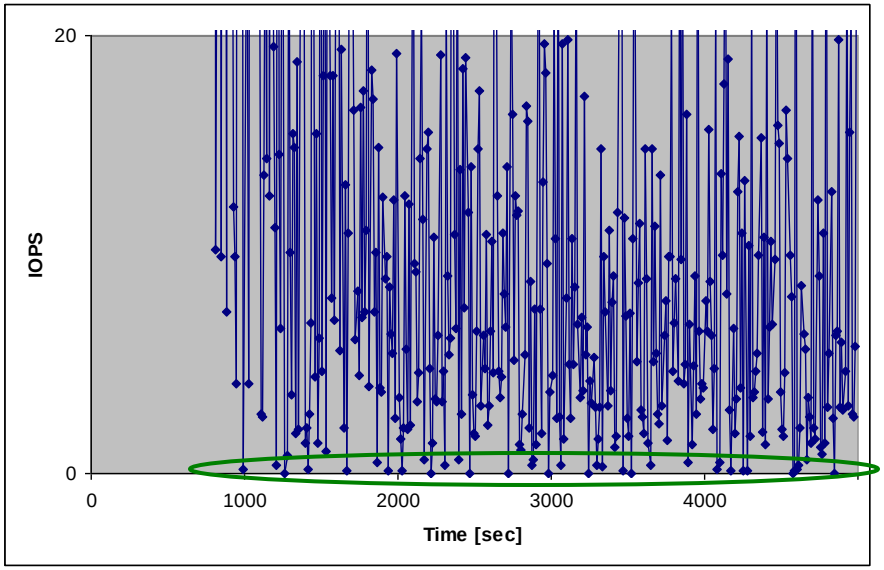
Effect of Region Size

HITACHI
Inspire the Next

Size 50000



Size 10000



Set-Associative Disk Cache: Score Sheet

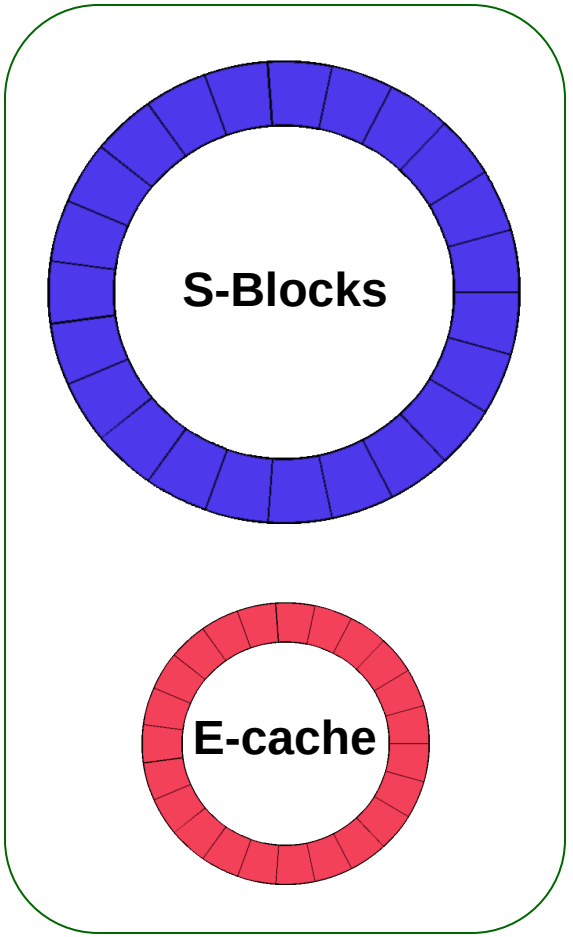
HITACHI
Inspire the Next

- Significant improvement over plain Read-Modify-Write
- Simple to implement

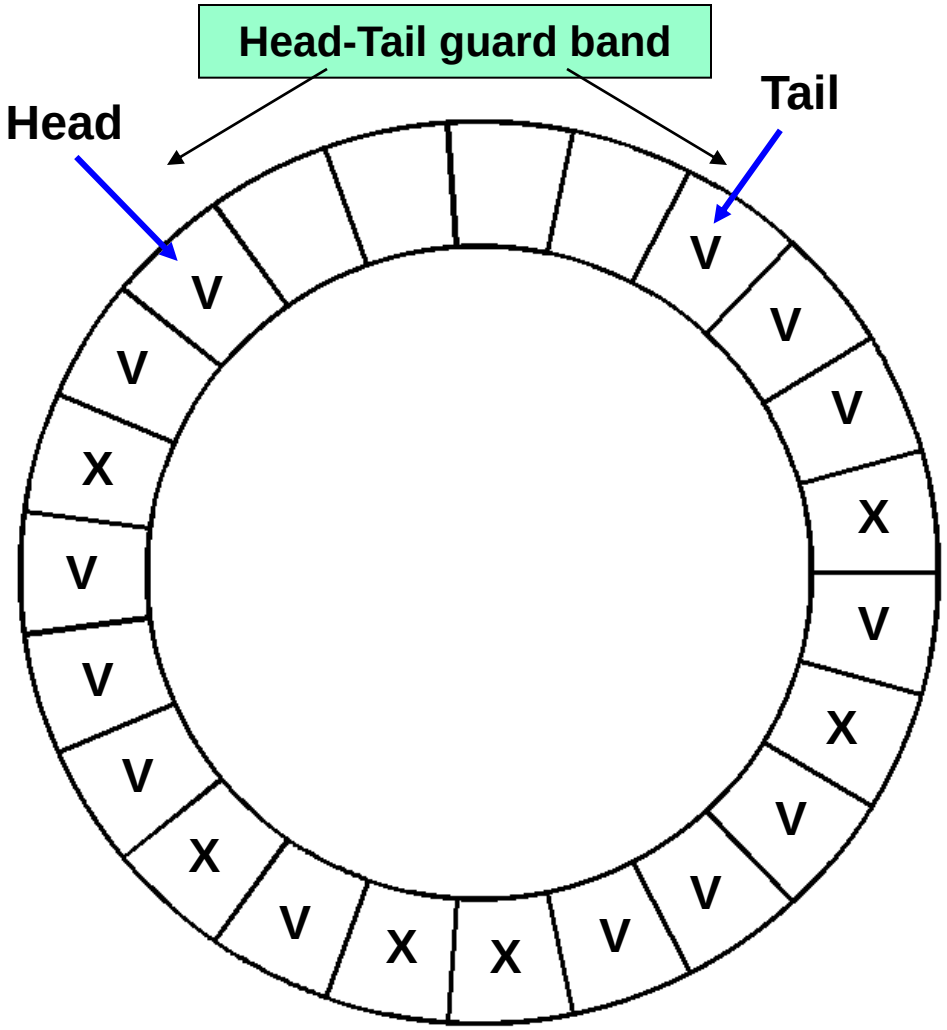
- Large dips in performance due to long garbage collections
- Region updated in place → consistency issues

The S-Blocks Architecture

- Temporary (red) and permanent (blue) storage managed as ring buffers
- S-Block: intermediate unit between sector and region



Ring Buffer

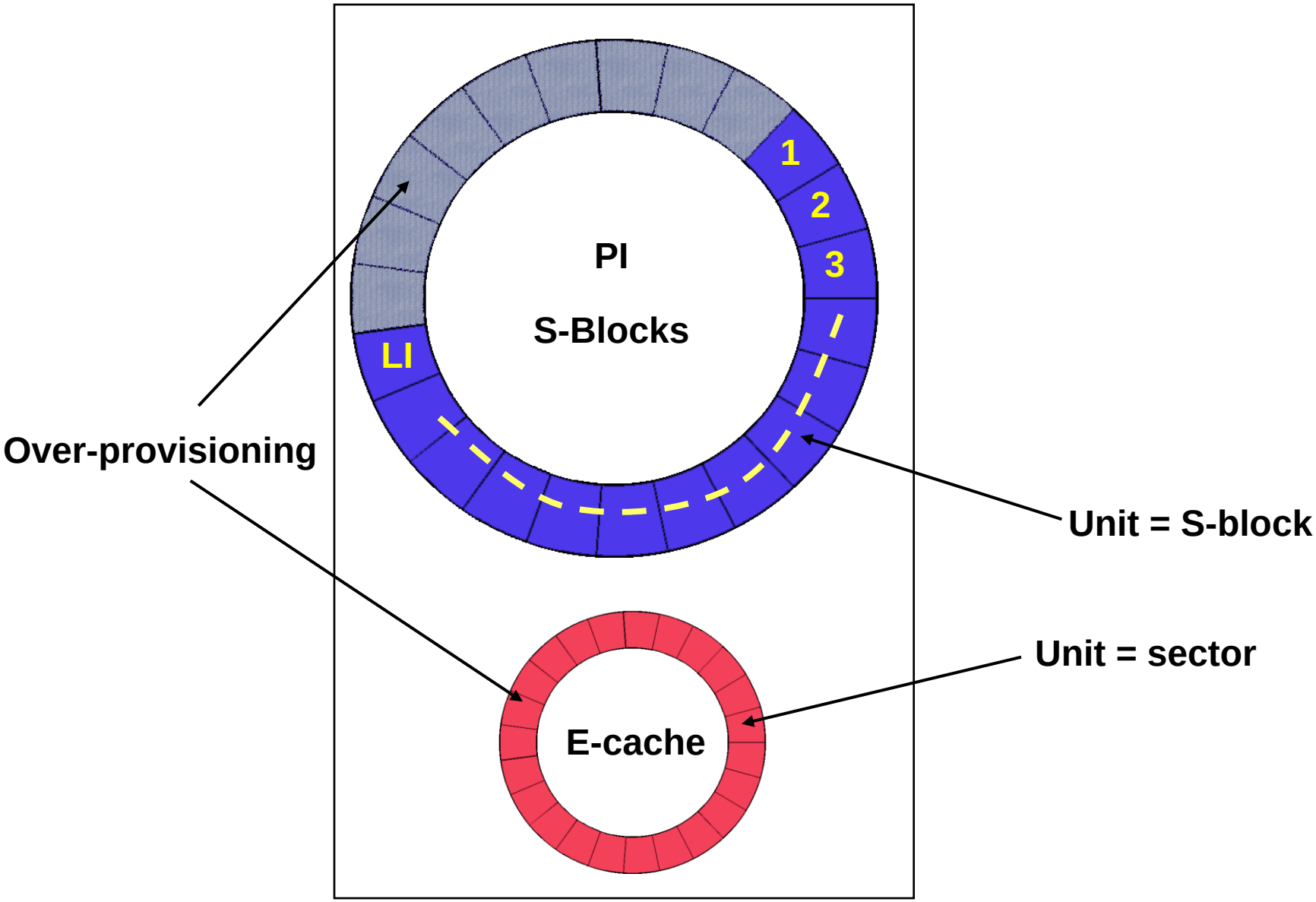


V Used, valid

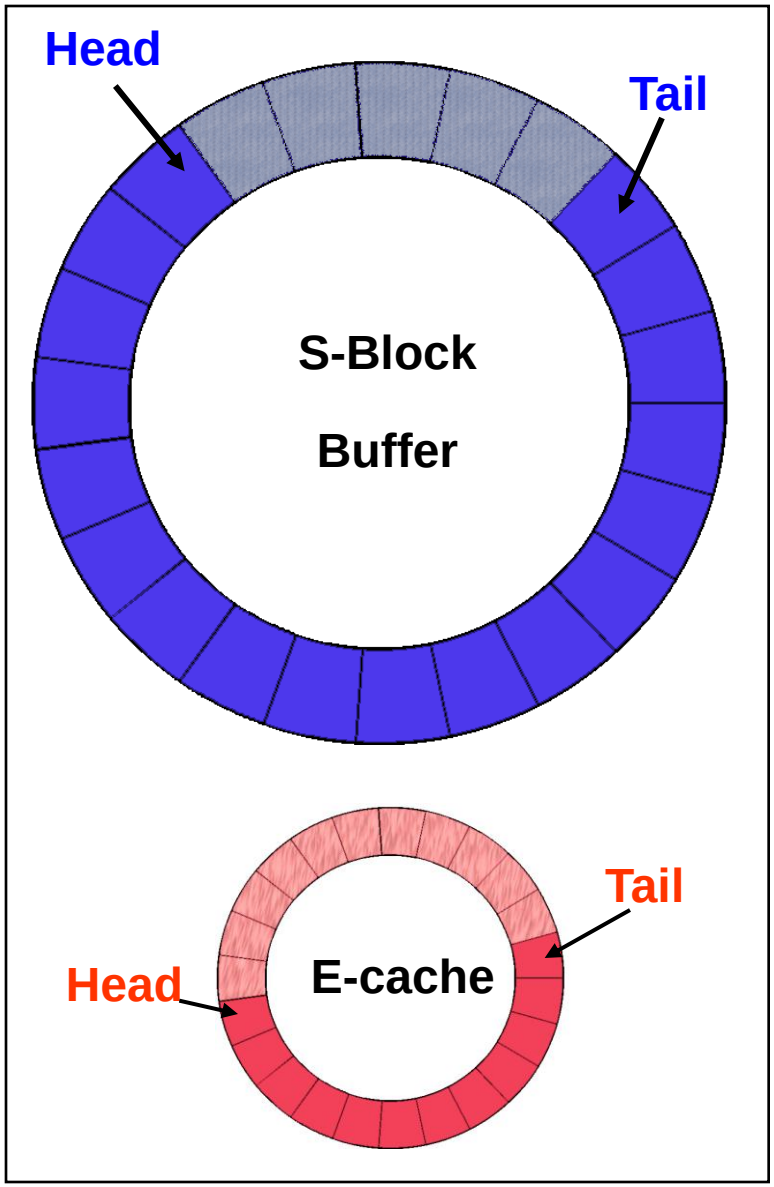
X Used, invalid

Unused

Initial Conditions

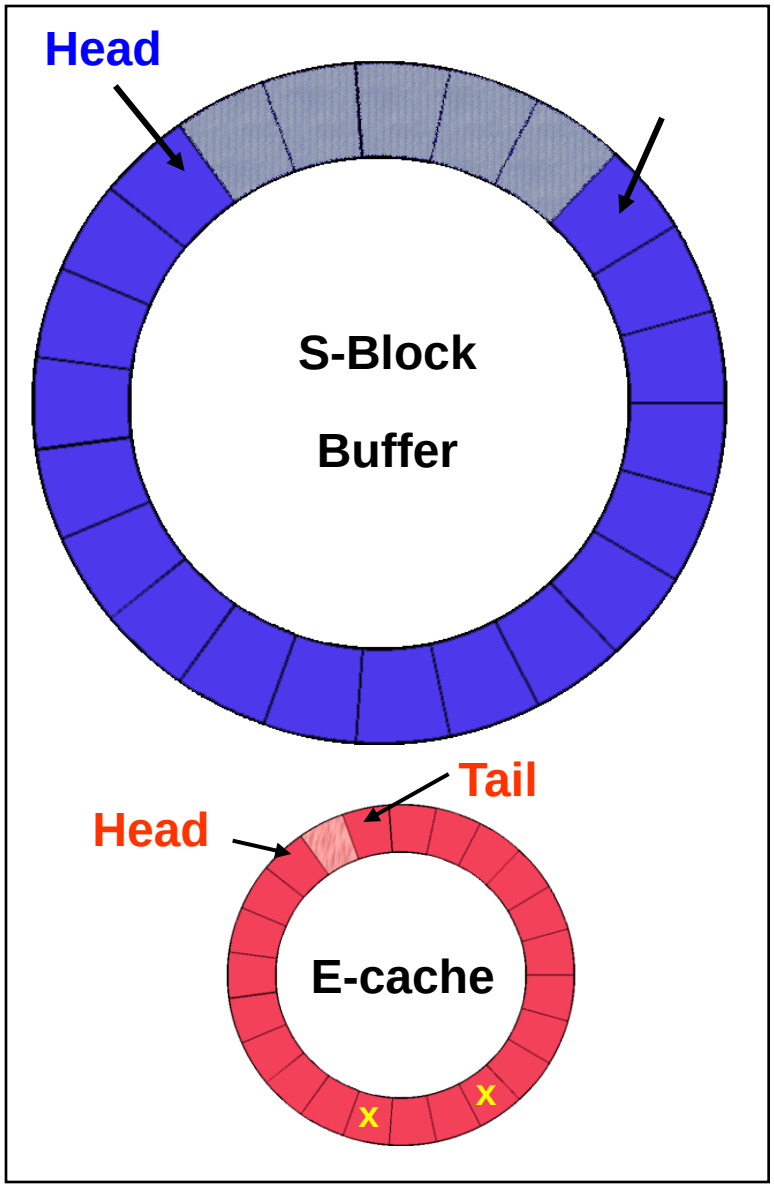


Write Operation



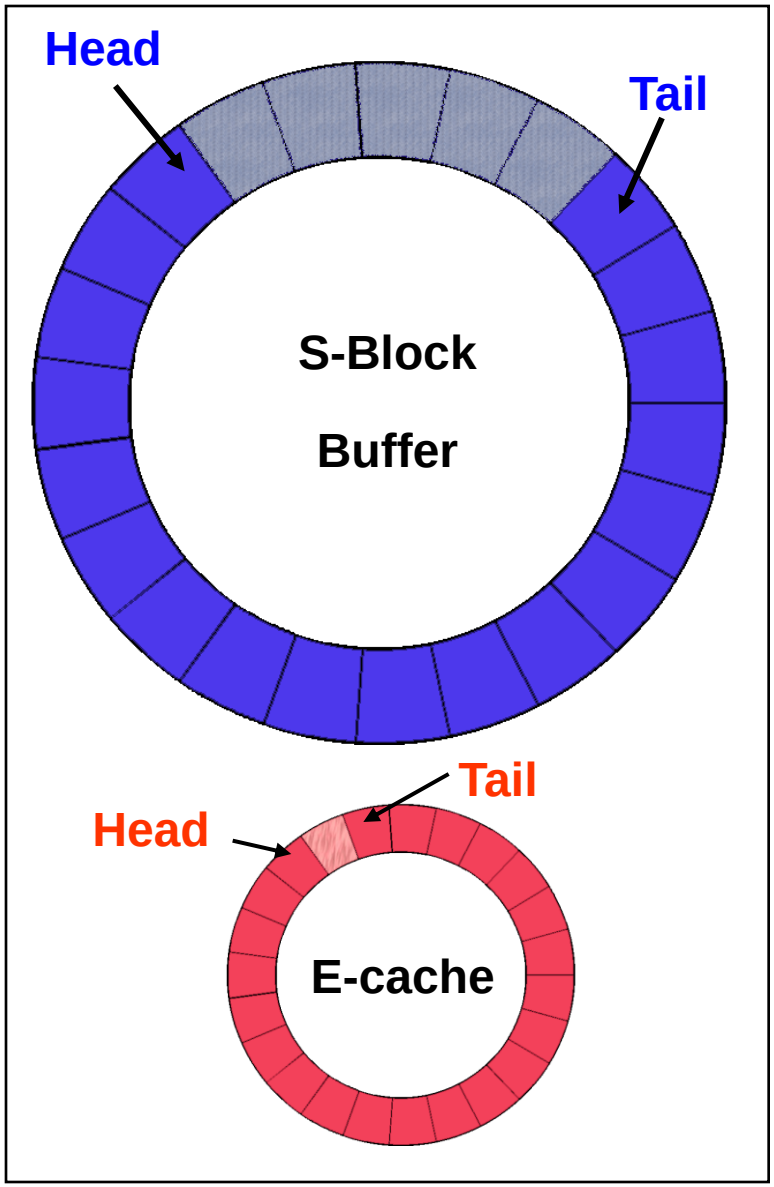
- If full S-Block: write directly in S-Block buffer
- If smaller than full S-Block: write in E-cache

E-cache Full



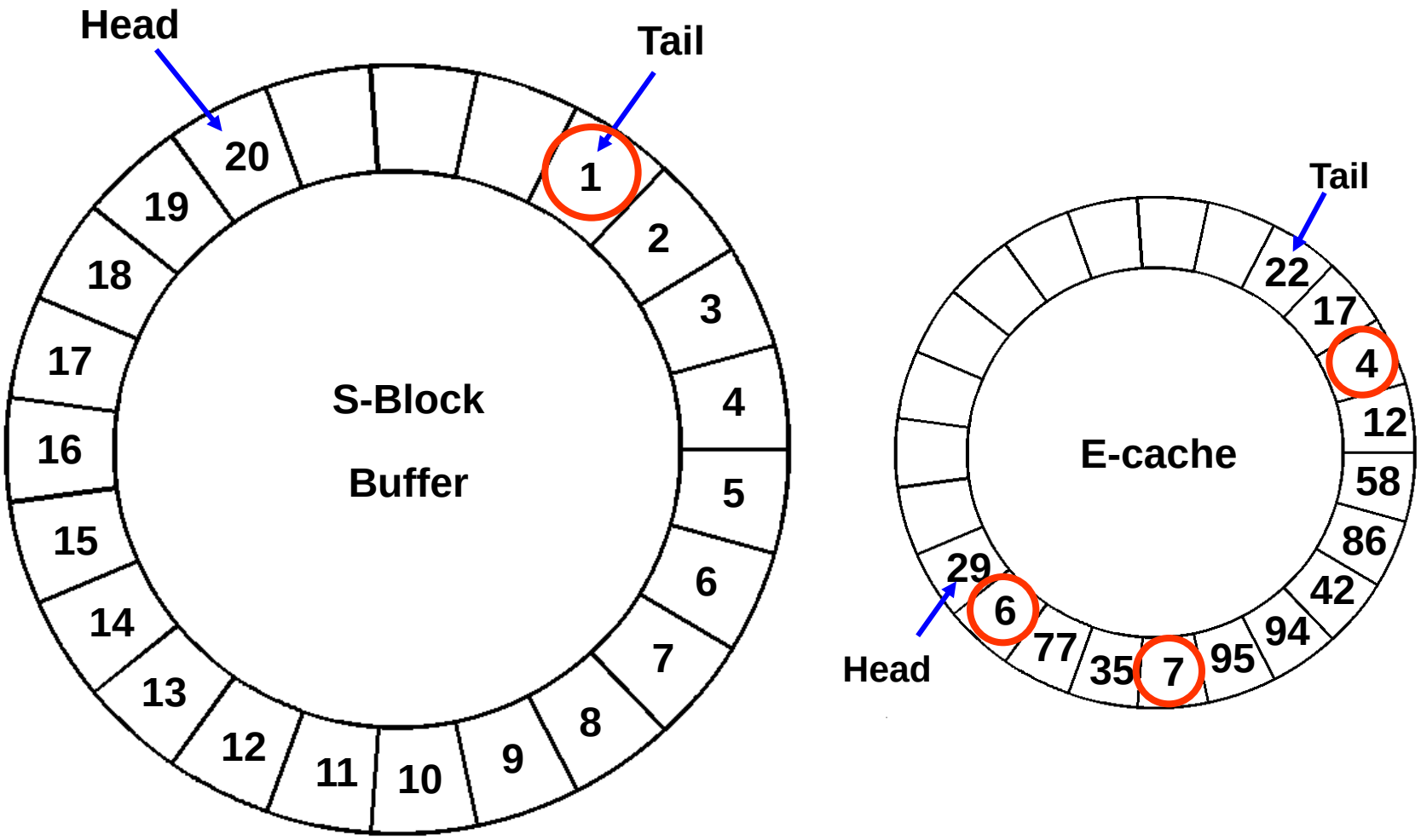
- If no room for incoming write: reclaim invalid exceptions in E-cache (defrag)

Defrag(E-cache)



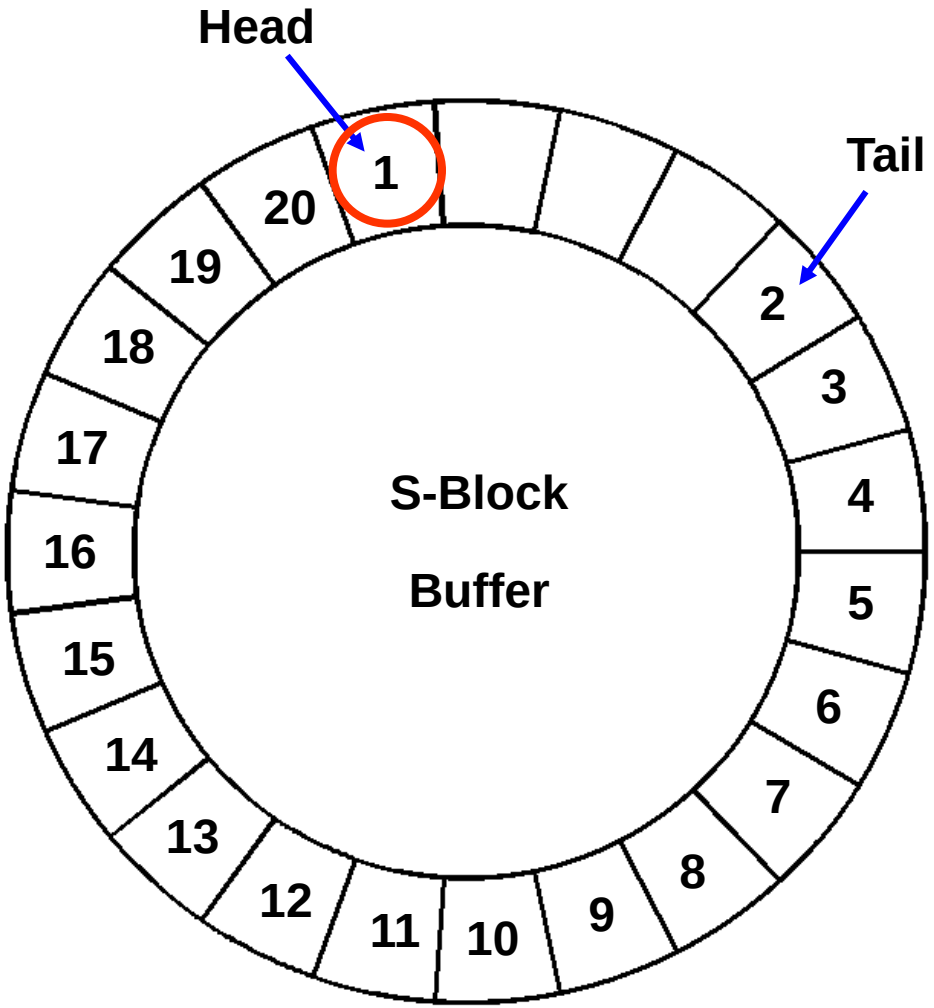
- If not enough invalid exceptions:
  - Choose S-Block
  - Destage(S-Block)

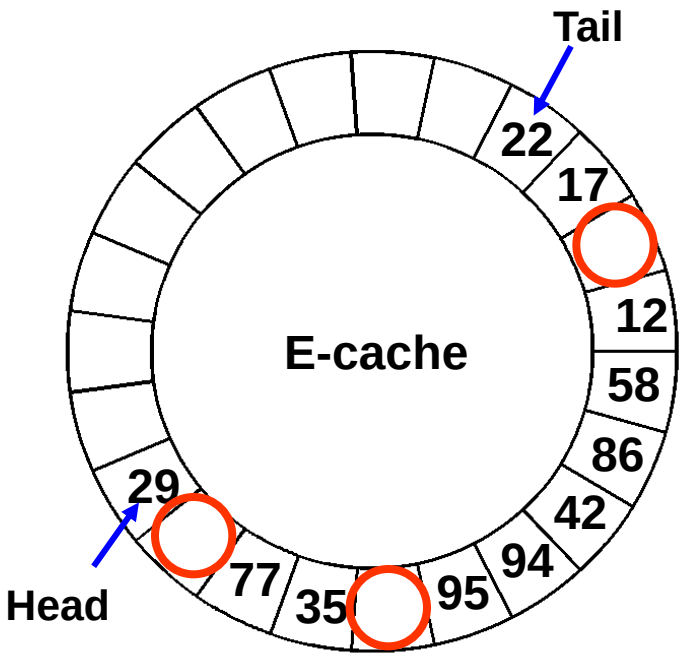
Destage(S-Block)



S-Block 1 → sectors 1-10

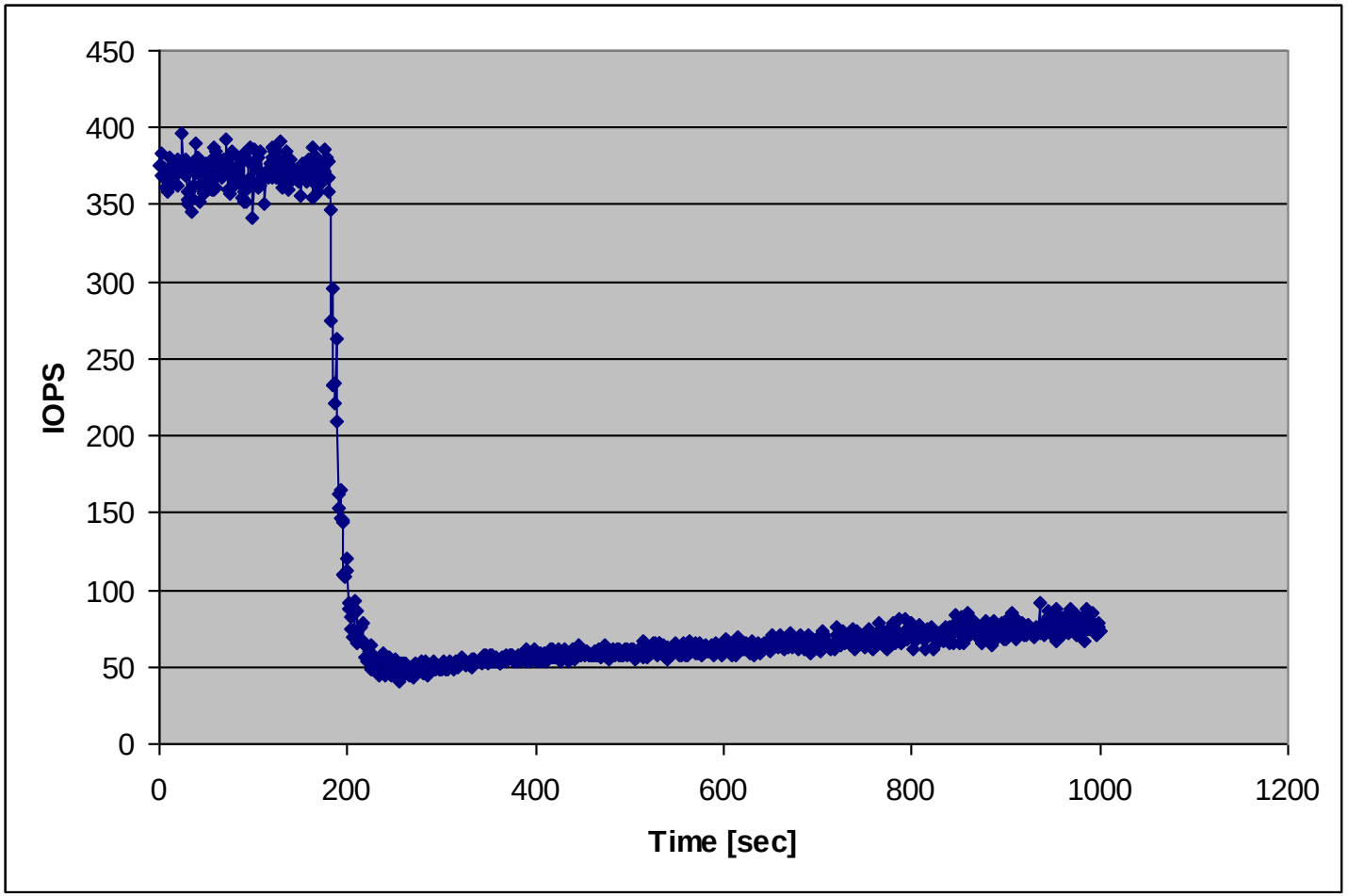
Destage(S-Block)





S-block writes in E-cachce are not needed after destage

4K Random IOPS



Choosing S-Block to Destage

HITACHI
Inspire the Next

Optimal Destage

- Choose S-Block with highest exception count
- Best amortization of S-Block rewrite
- Good for biased workloads (e.g. hotspots)

Optimal Defrag

- Choose S-Block closest to the tail
- Least amount of copying in S-Block defrag
- Good for random workloads

Comparison for Random Workload

HITACHI
Inspire the Next

Optimal Destage

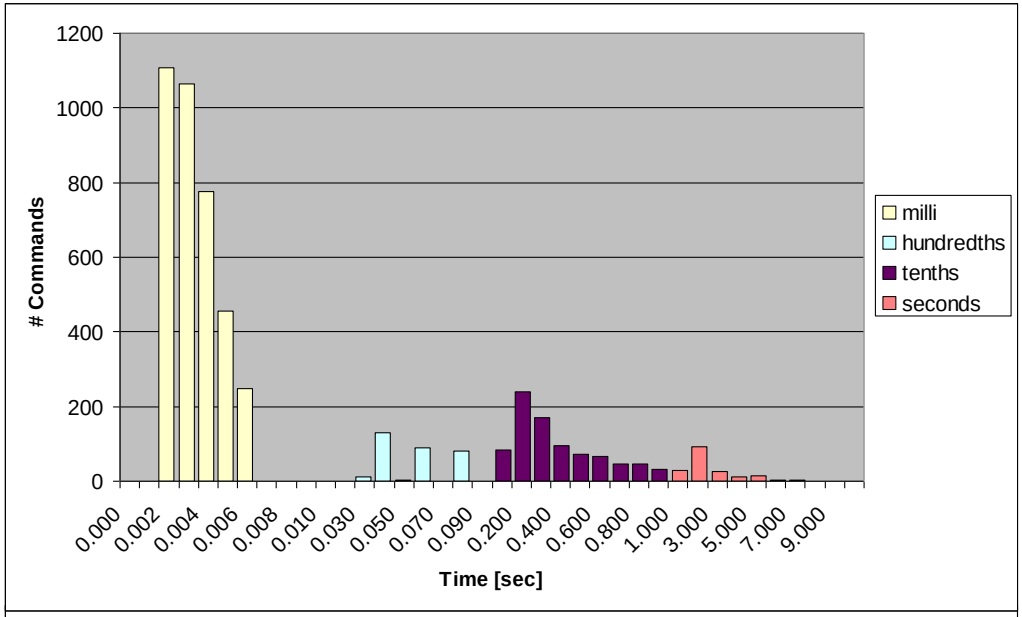
- S-Block destage average # invalidated exceptions 55.26
- S-Block defrag average copy count 1744.64
- average IOPS = 73.44

Optimal Defrag

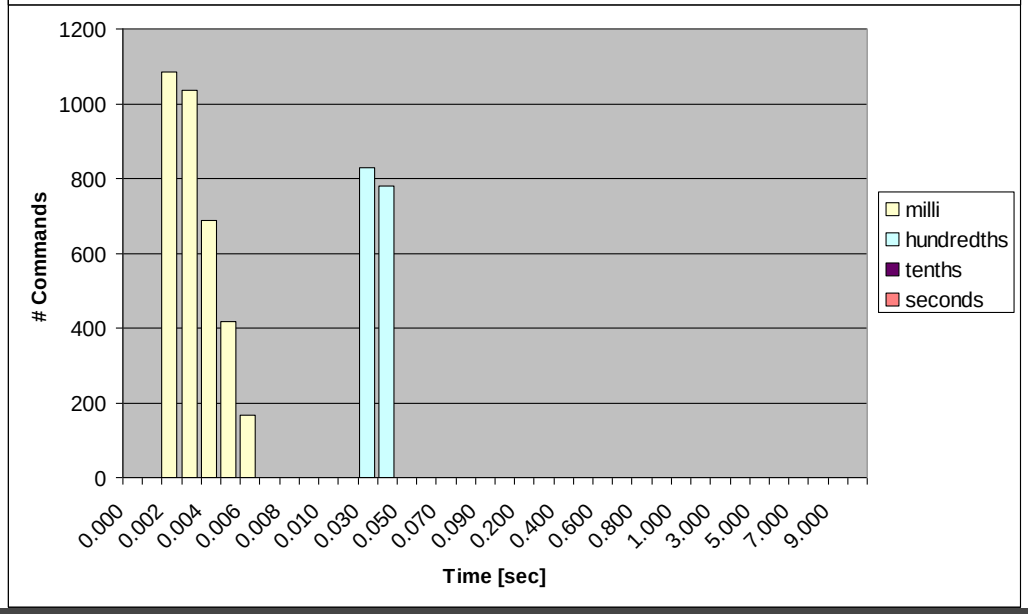
- S-Block destage average # invalidated exceptions 12.52
- S-Block defrag average copy count 0.00
- average IOPS = 121.54

Effect of S-Block Choice on Performance

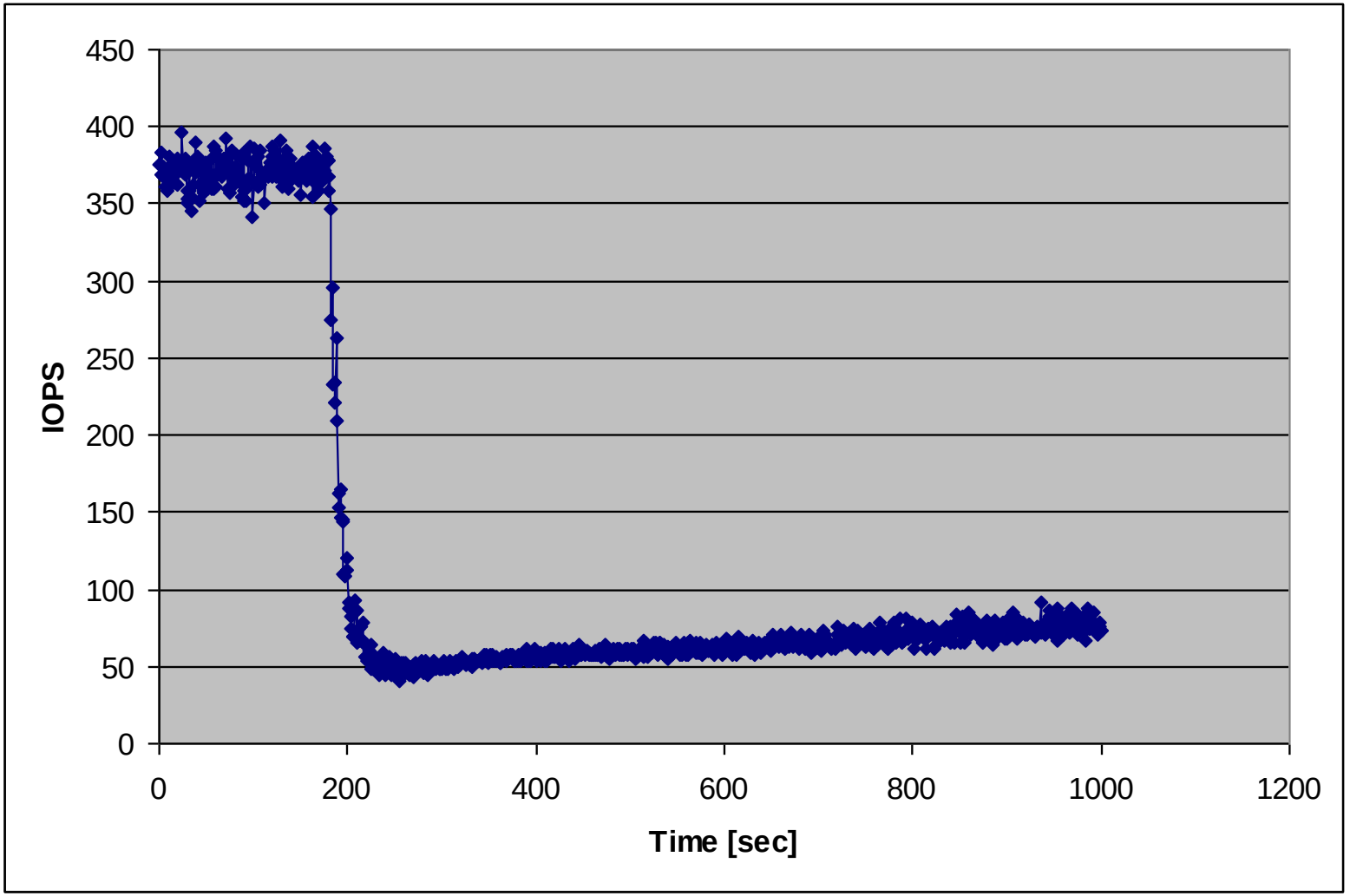
Optimal Destage



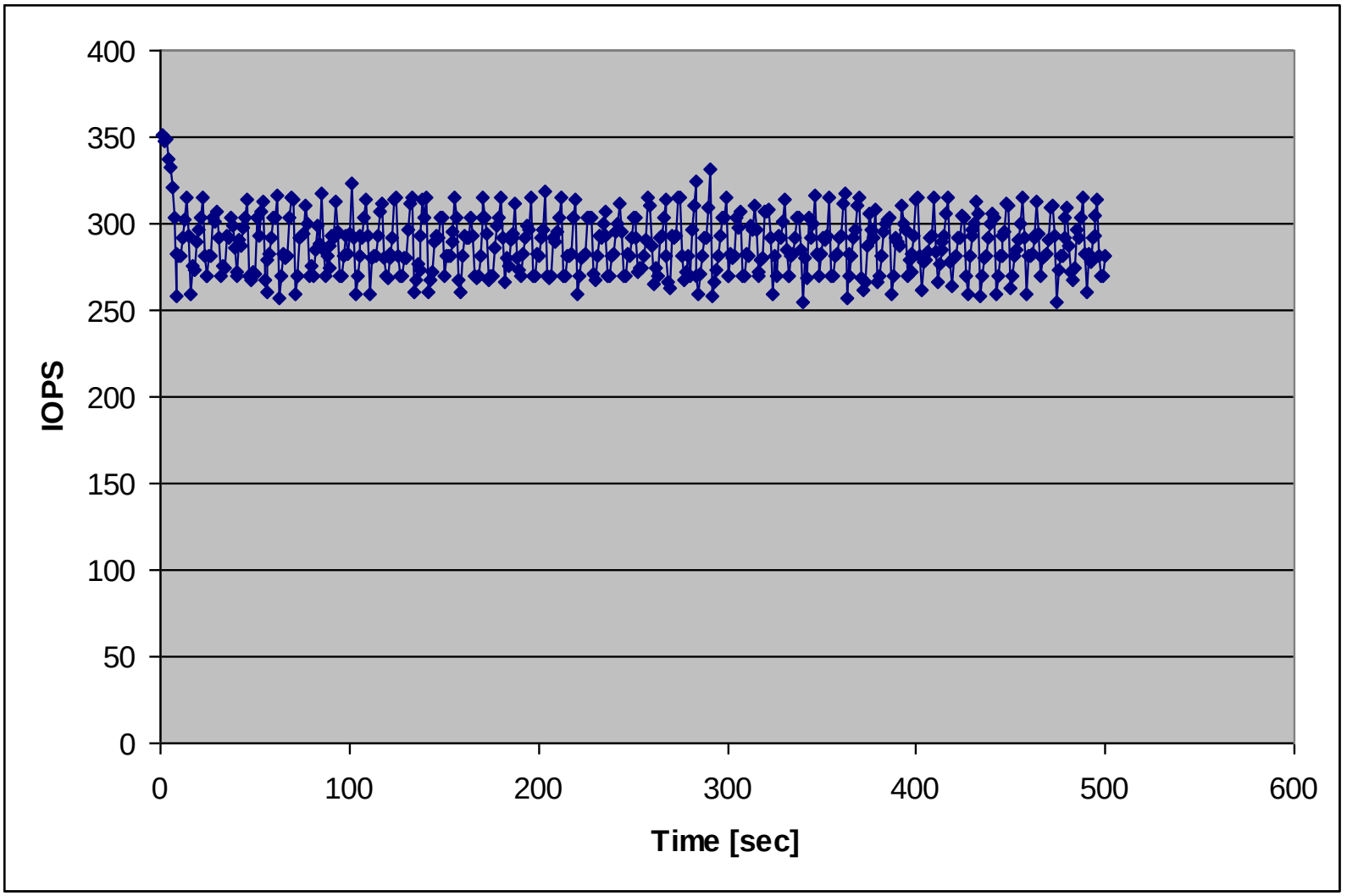
Optimal Defrag



Lazy Garbage Collection



Constant Garbage Collection



S-Blocks: Score Sheet

HITACHI
Inspire the Next

- Good sustained random write performance
- Append update → no consistency issues
- Flexible to handle different workload types
- Good sequential performance with direct S-block writes

- Non-trivial implementation

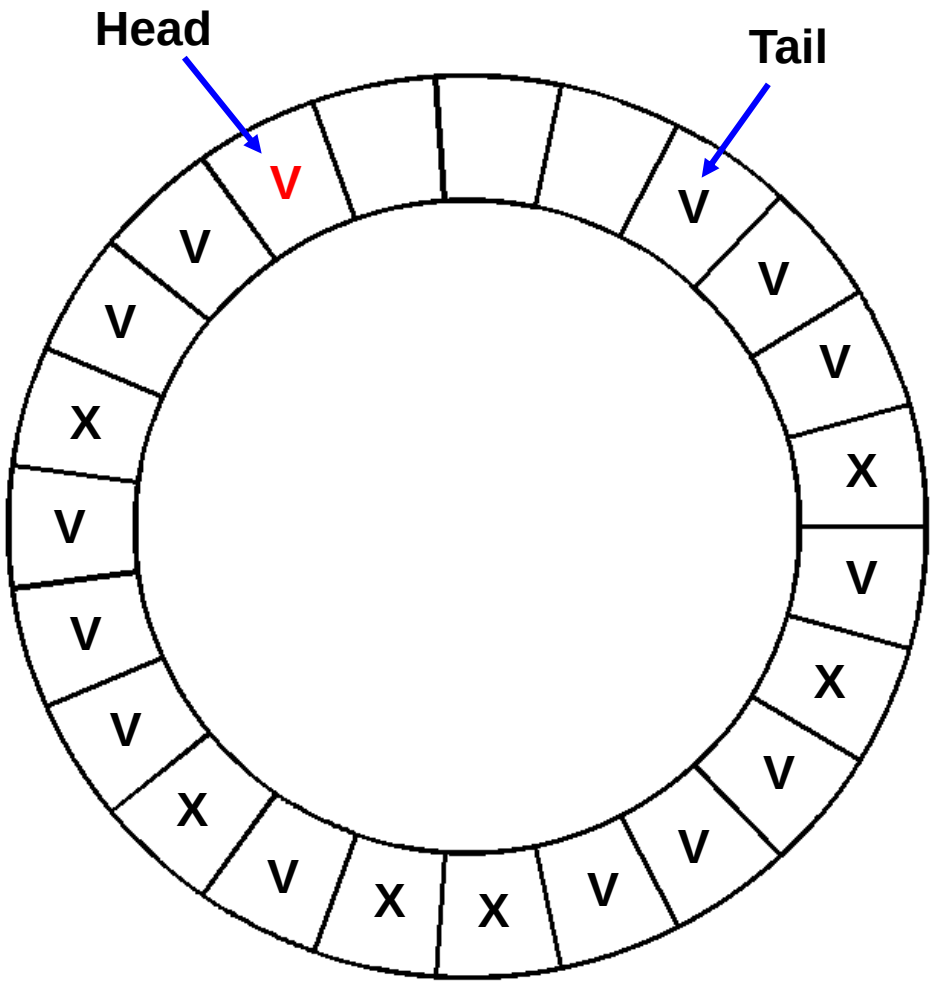
Summary

- Shingled magnetic recording opens a rich area of systems research
- Good understanding of main issues and tradeoffs
- Proposed architectures likely basis for real implementations
- Significant research needed in performance optimization

Thank YOU !

Backup Slides

Add to Ring Buffer

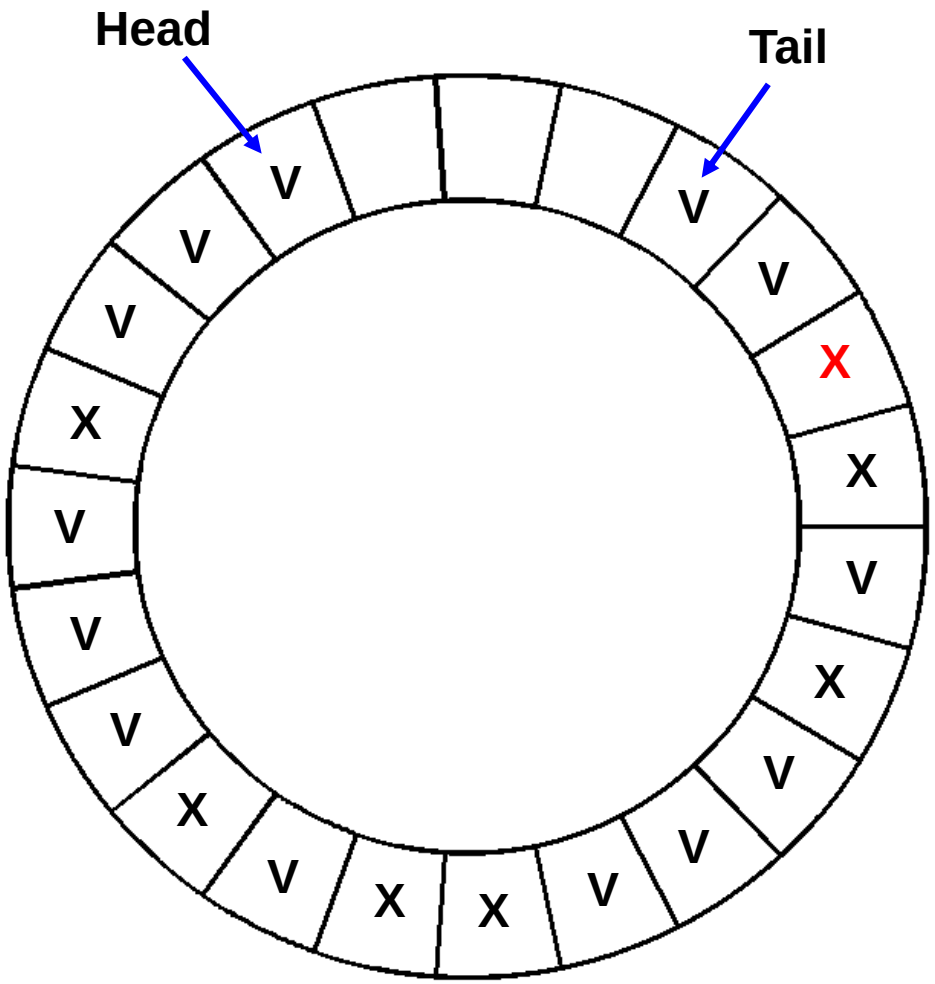


V Used, valid

X Used, invalid

Unused

Invalidate in Ring Buffer

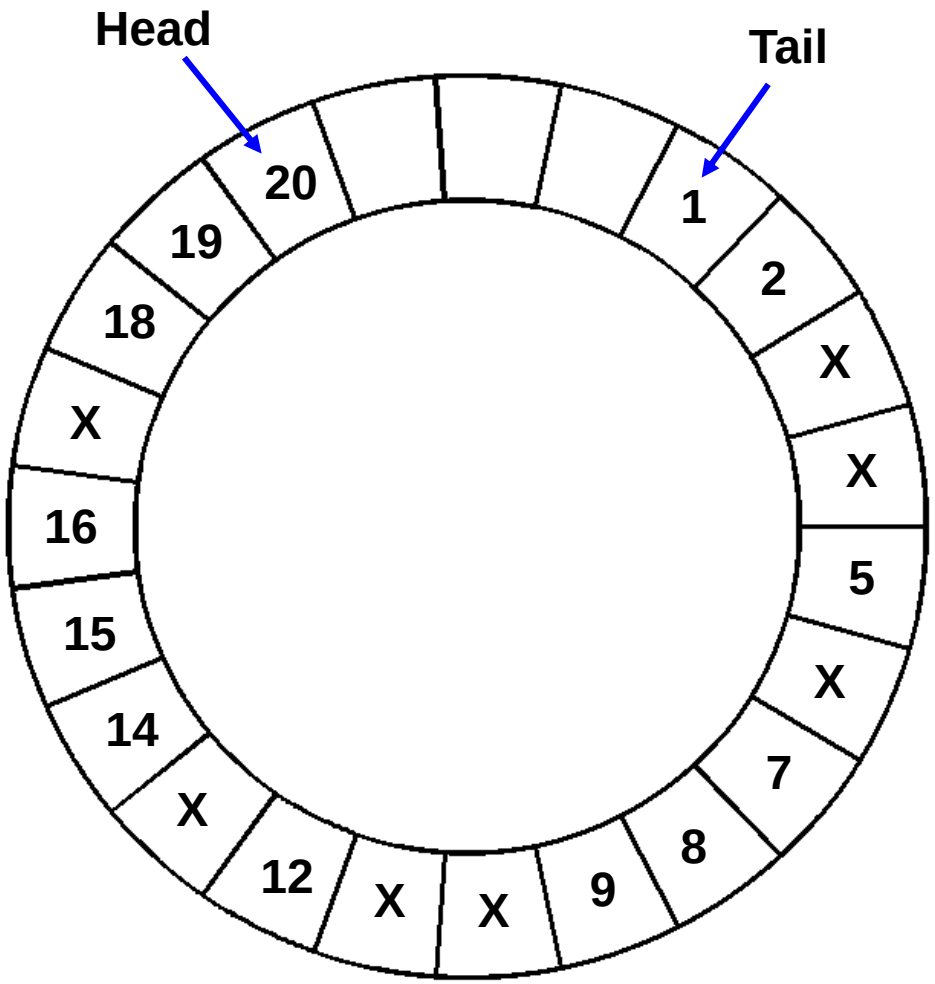


V Used, valid

X Used, invalid

Unused

Defrag Ring Buffer

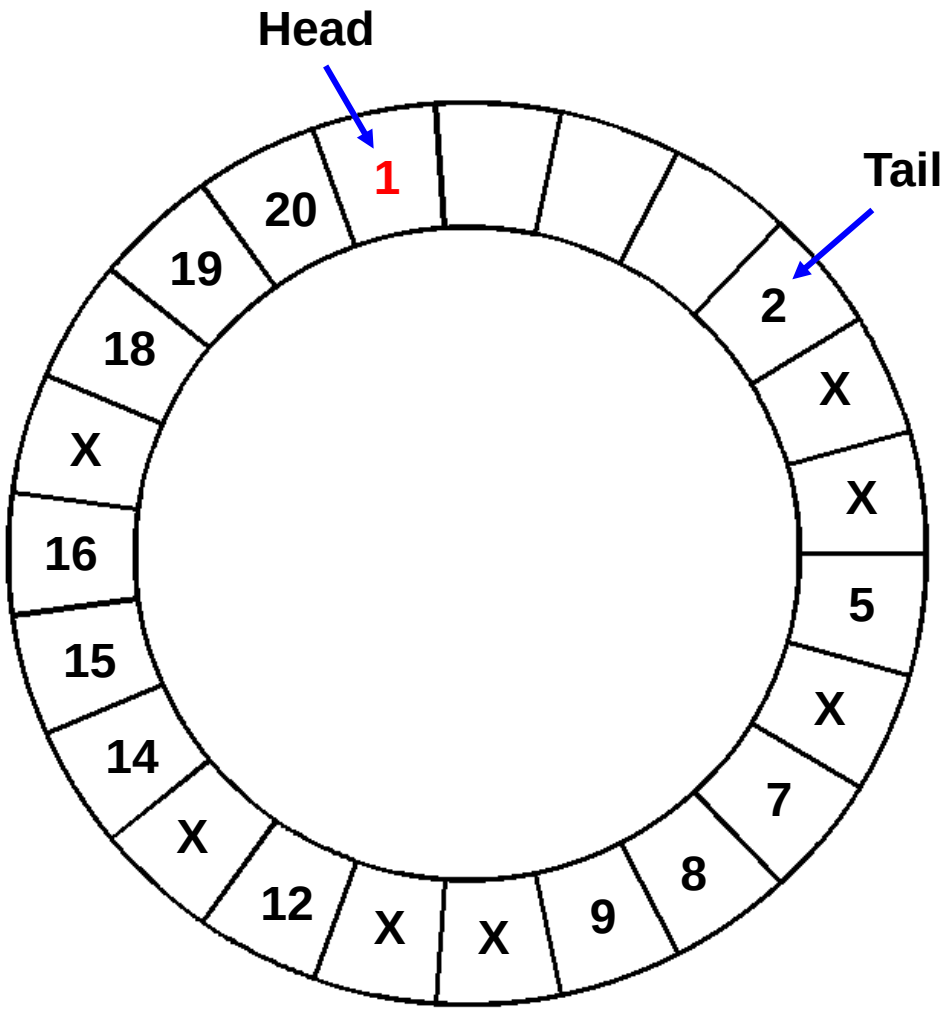


Used, valid

X Used, invalid

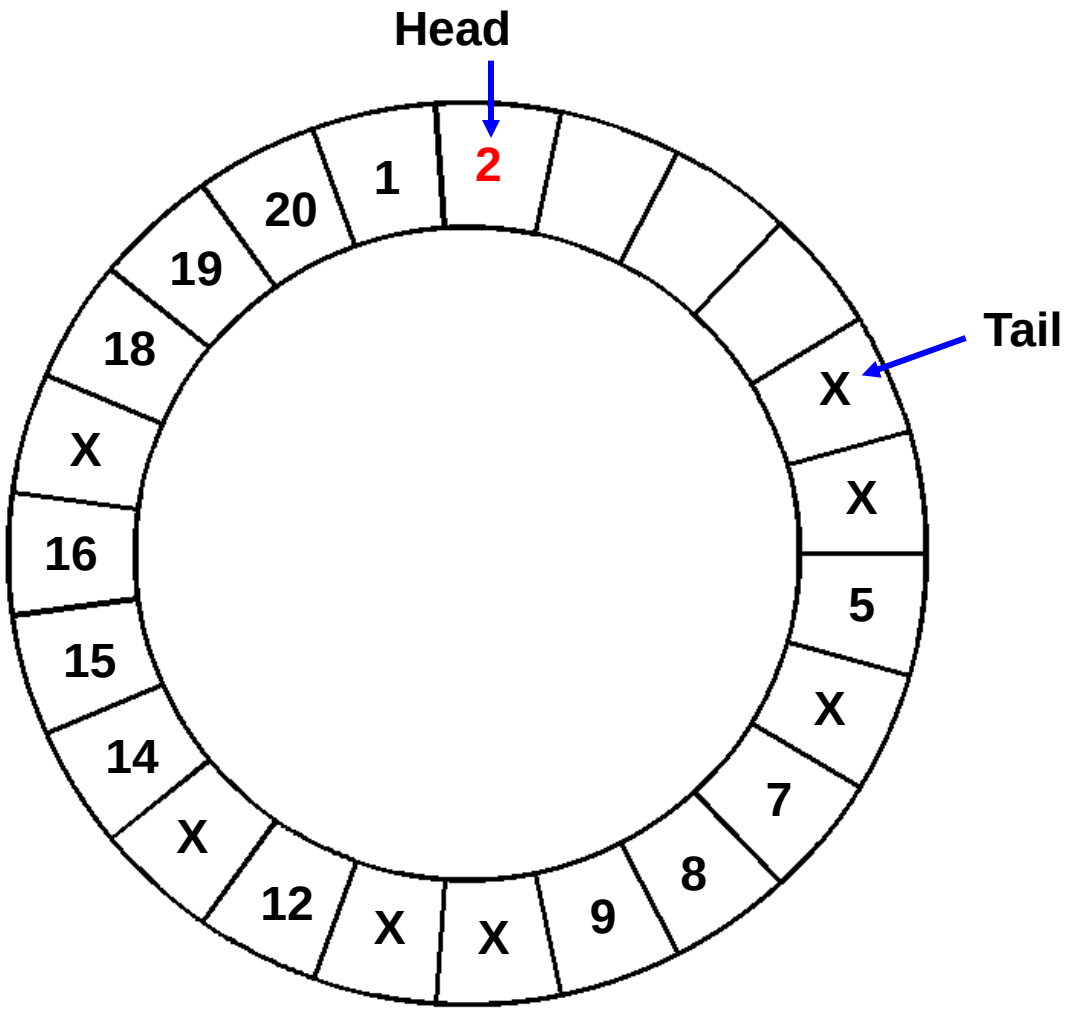
Unused

Defrag Ring Buffer



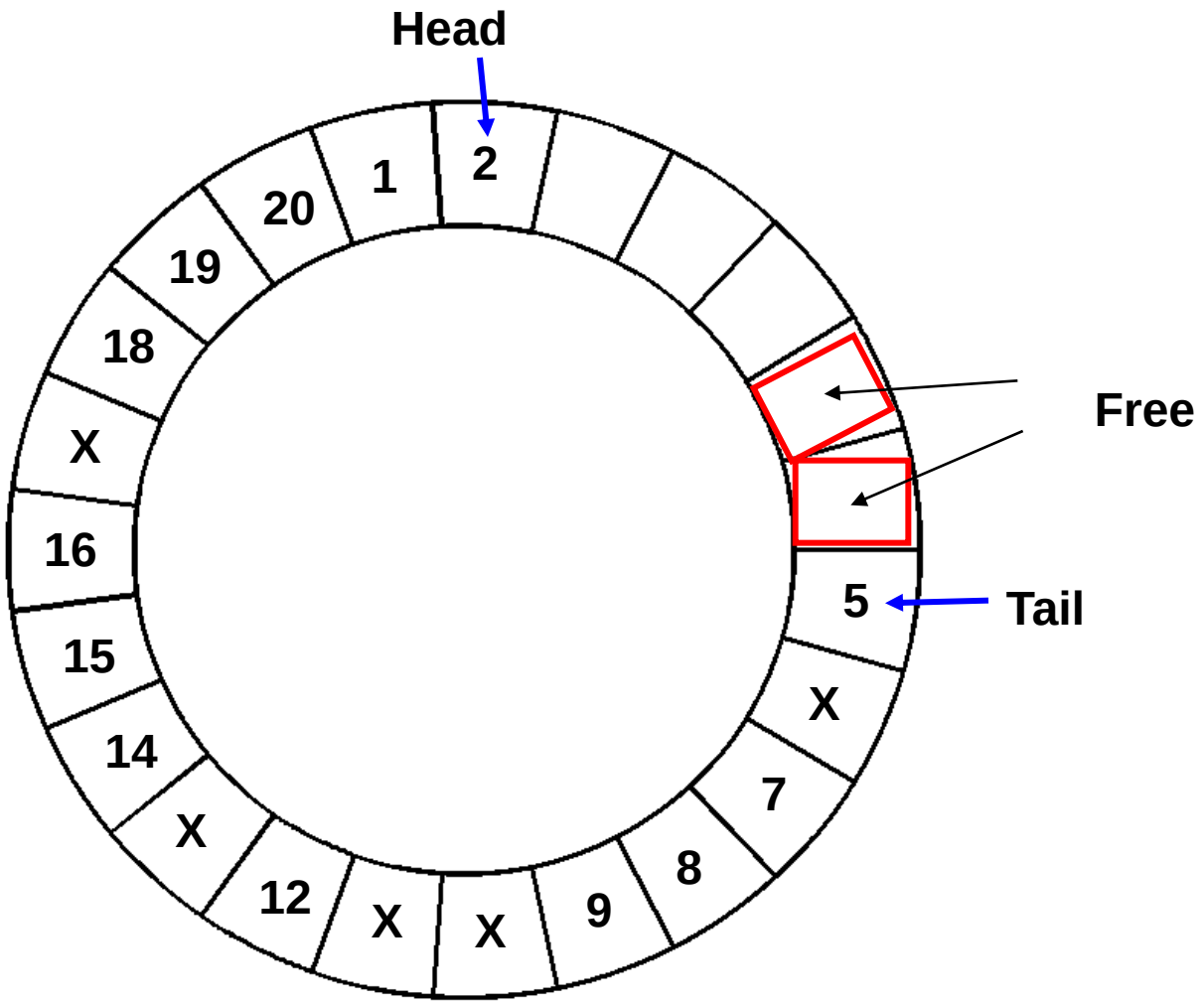
#	Used, valid
X	Used, invalid
	Unused

Defrag Ring Buffer



#	Used, valid
X	Used, invalid
	Unused

Defrag Ring Buffer



#	Used, valid
X	Used, invalid
	Unused