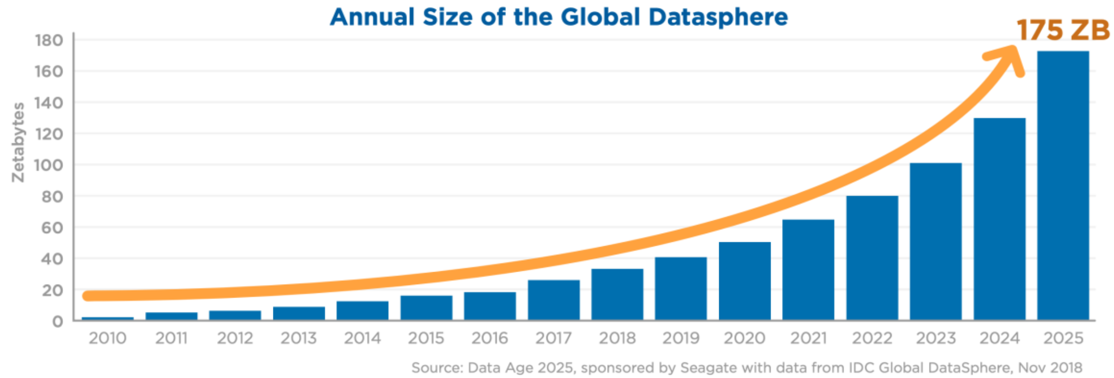
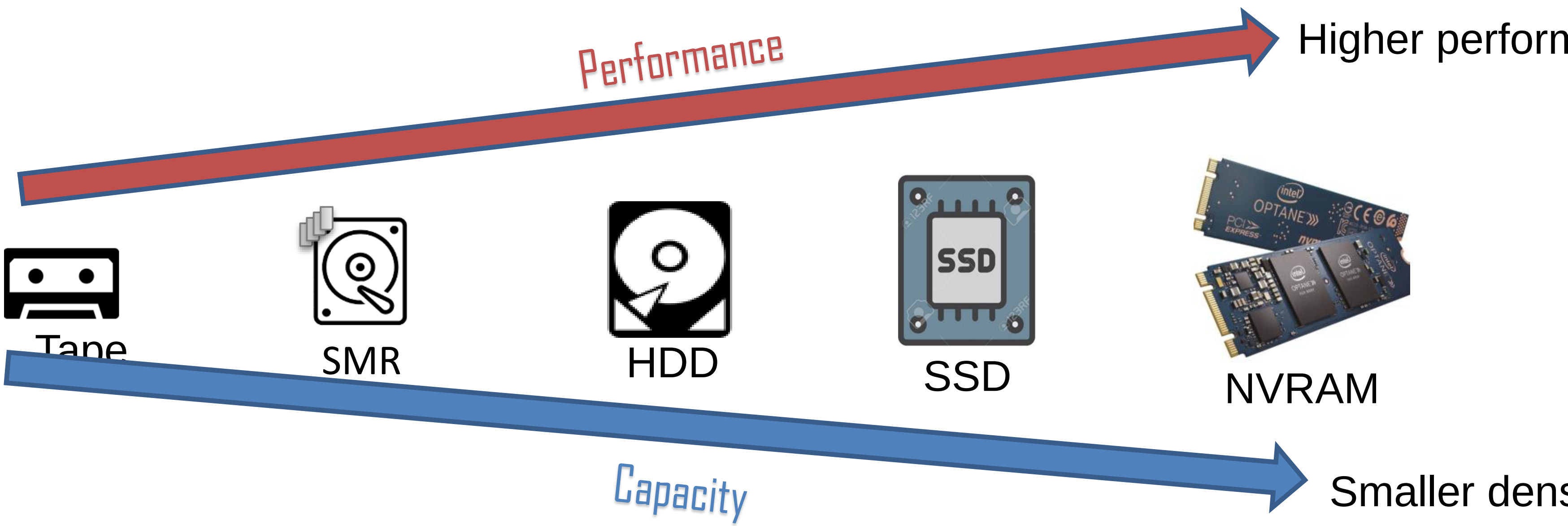
Csci 5980 Spring 2020

New Storage Technologies/Devices

University of Minnesota


BIG
DATA
ANALYTICS
COMPLEX
STORAGE
INTERNET
MANAGEMENT
DATABASES
NETWORKS

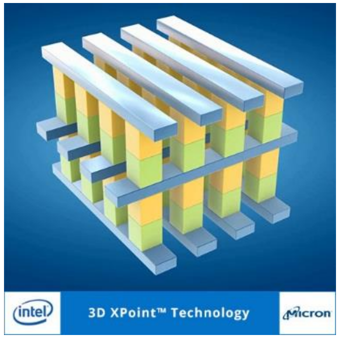

Annual Size of the Global Datasphere
175 ZB
Zetabytes
0
20
40
60
80
100
120
140
160
180
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
Source: Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018


Performance
Higher perforn
Tape
SMR
HDD
SSD
NVRAM
Capacity
Smaller dens

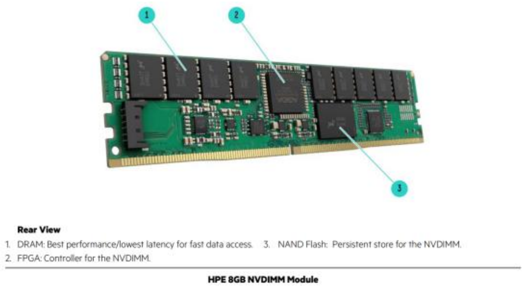
Non-Volatile Memory
NVRAM

University of Minnesota

Examples of
non-volatile memory (NVRAM)



3D Xpoint
(By Intel and Micron)



NVDIMM
(By HPE)

STT-MRAM
(By
Everspin)

Summary of Memory Technologies

	HDD	DRAM DIMM	Flash SSD	PCM (25nm)
Density (μm^2/bit)	0.00006	0.00380	0.00210	0.00250
Read Latency (ns)	3,000,000	55	25,000	48
Write Latency (ns)	3,000,000	55	200,000	150
Read Energy (pJ/bit)	2,500	12.5	250	2
Write Energy (pJ/bit)	2,500	12.5	250	19.2
Static Power	Yes	Yes	No	No
Endurance	$>10^{15}$	$>10^{15}$	10^4	10^8
Nonvolatility	Yes	No	Yes	Yes

Summary of Different Memory Technologies

	SRAM	DRAM	Flash	FRAM	MRAM	ReRAM
Read Speed	Fast	Medium	Medium	Fast	Fast	Medium
Write Speed	Fast	Medium	Slow	Fast	Medium	Medium
Array Efficiency	High	High	Medium	Medium	High	High
Scalability	Good	Limited	Limited	Limited	Medium	Good
Cell Density	Low	High	High	Medium	Medium	High
Volatile?	Yes	Yes	No	No	No	No
Endurance	Infinite	Infinite	Limited	Limited	Infinite	Limited
Current Consumption	Low/High	High	Low	Low	Low	Low
Low-Voltage	Yes	Limited	Limited	Limited	Yes	Yes
Process Complexity	Low	Medium	Medium	Medium	Complex	Medium

(Source: Objective Analysis)

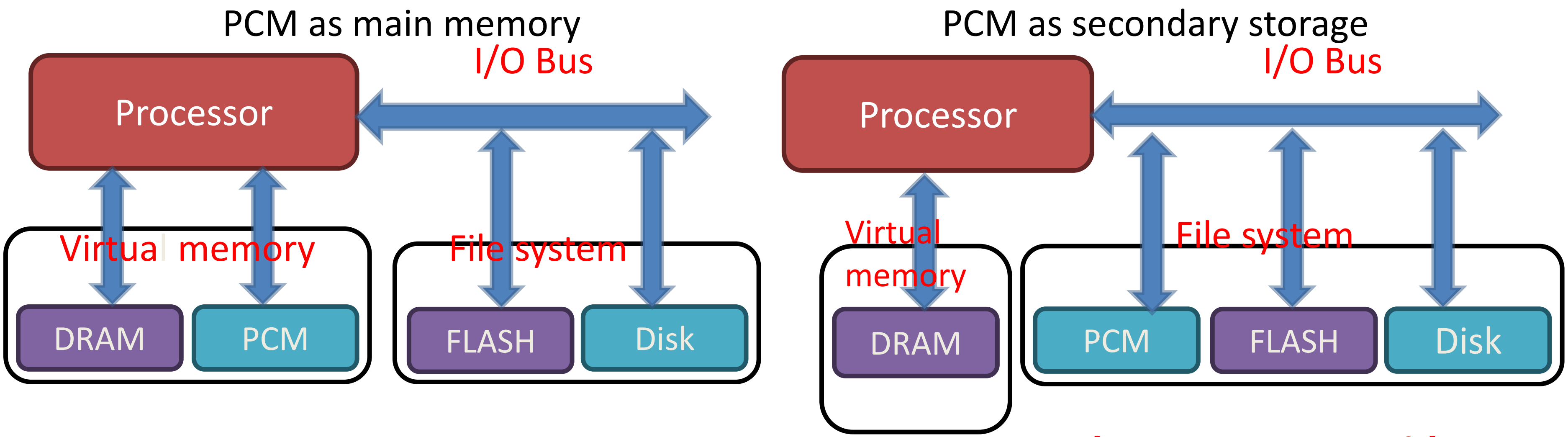
How to innovate our software, architecture and systems to exploit NVRAM technologies?

- ✓ Non-volatile
- ✓ Low power consumption
- ✓ Fast (close to DRAM)
- ✓ Byte addressable
- ✓ Memory or Storage?

NVM Research Issues

- Data Consistency and Durability against Systems and Application failures
  - Solutions: ACID (Atomicity, Consistency, Isolation, and Durability) Transactions, Appended Logs, and Shadow Update
  - Challenges: Guarantee Consistency and Durability While Preserve Performance
- Memory Allocation, De-allocation & Garbage Collection
- New Programming Models

New Memory/Storage Hierarchy



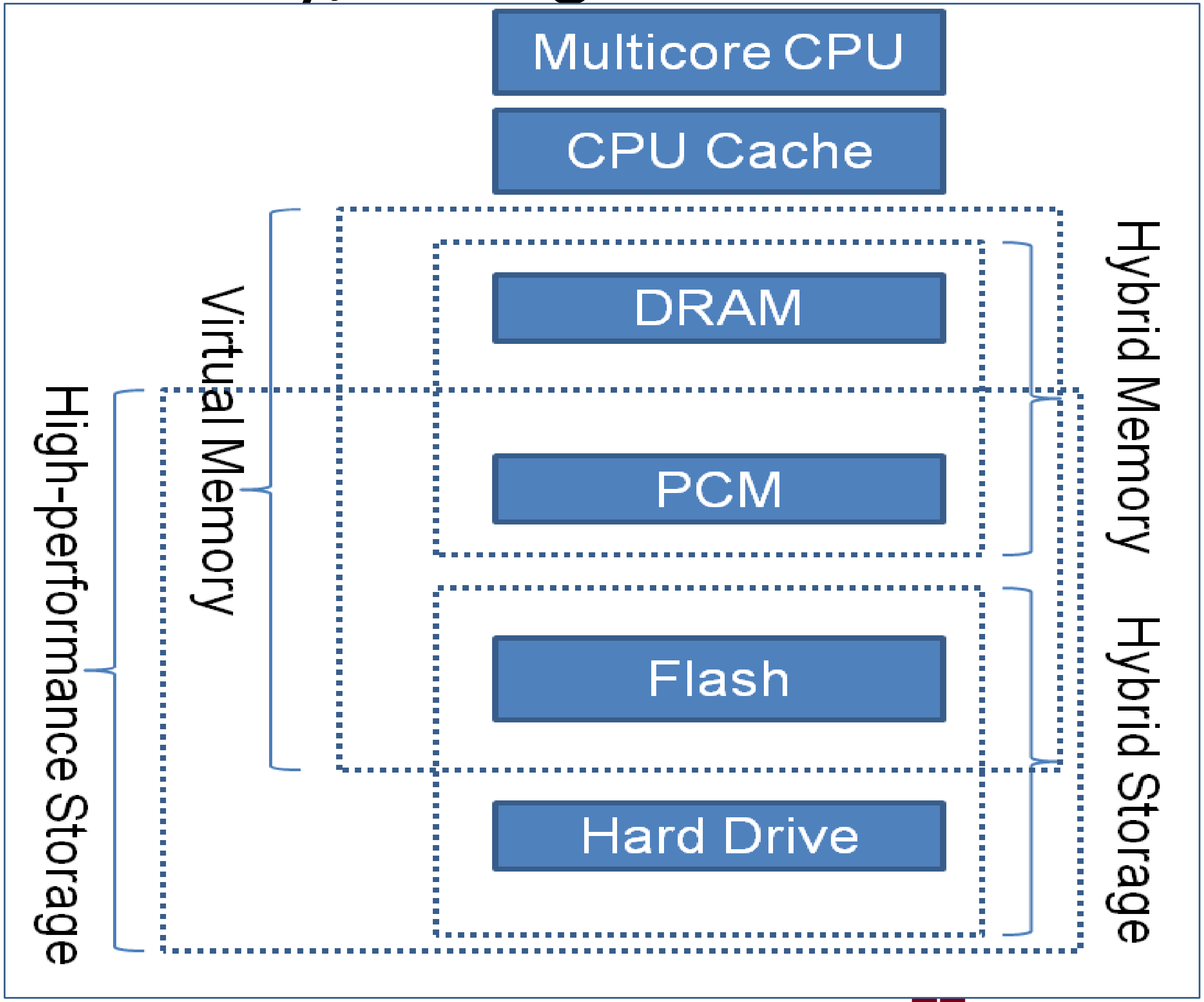
PCM as main memory provides:

1) **High capacity**
2) **standby power**

PCM as secondary storage provides:

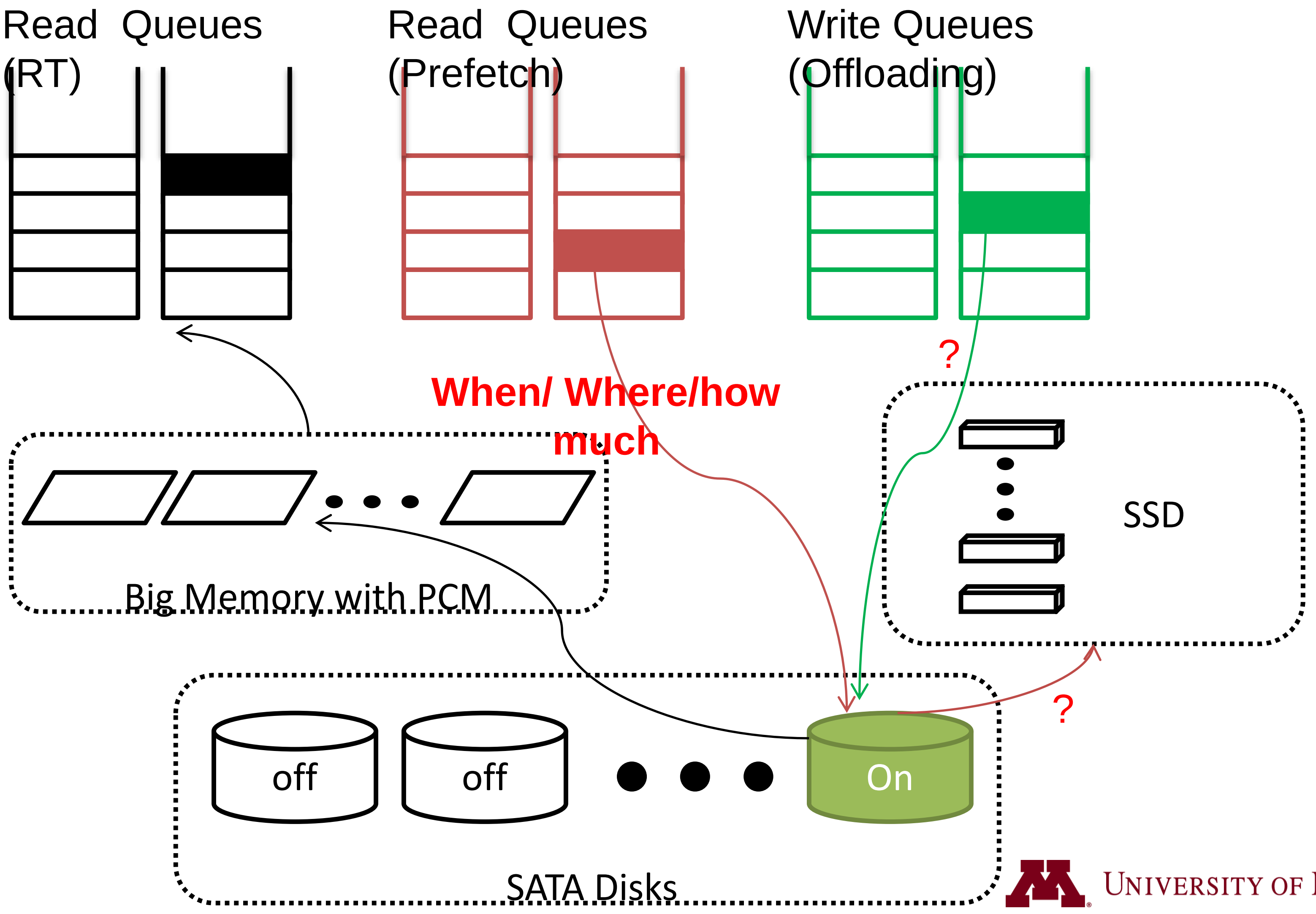
1) **Low access latency**

How to Integrating PCM and Flash Memory into Memory/Storage Hierarchies?



Storage Layer Management and Caching

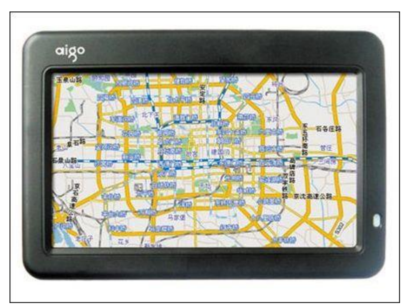
How this can be done in a HEC environment?



Flash Memory-based Solid State Drives

Why Flash Memory?

- Diversified Application Domains
  - Portable Storage Devices
  - Consumer Electronics
  - Industrial Applications
  - Critical System Components



Flash-based SSD Characteristics

- Random read is the same as sequential.
- Read and write by the unit of pages
- Does not allow overwrite. Need erase before writes. Erase is performed in blocks
- Typical block size is 128 K and page size 2K
- Write is slower than read. Erase is a very slow operation
- Read takes 25 microseconds, write takes 200 microseconds, and erase takes 1500 microseconds
- Limited number of writes per cell. 100 K for SLC and 10K for MLC.
- Flash Translation Layer (FTL) sits in between file system and SSD. FTL provides remapping and wear-leveling

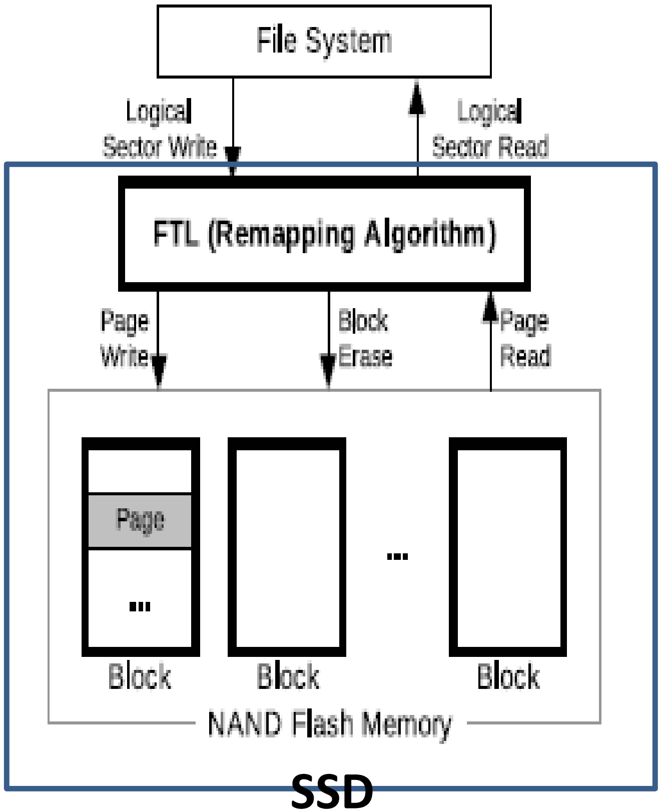
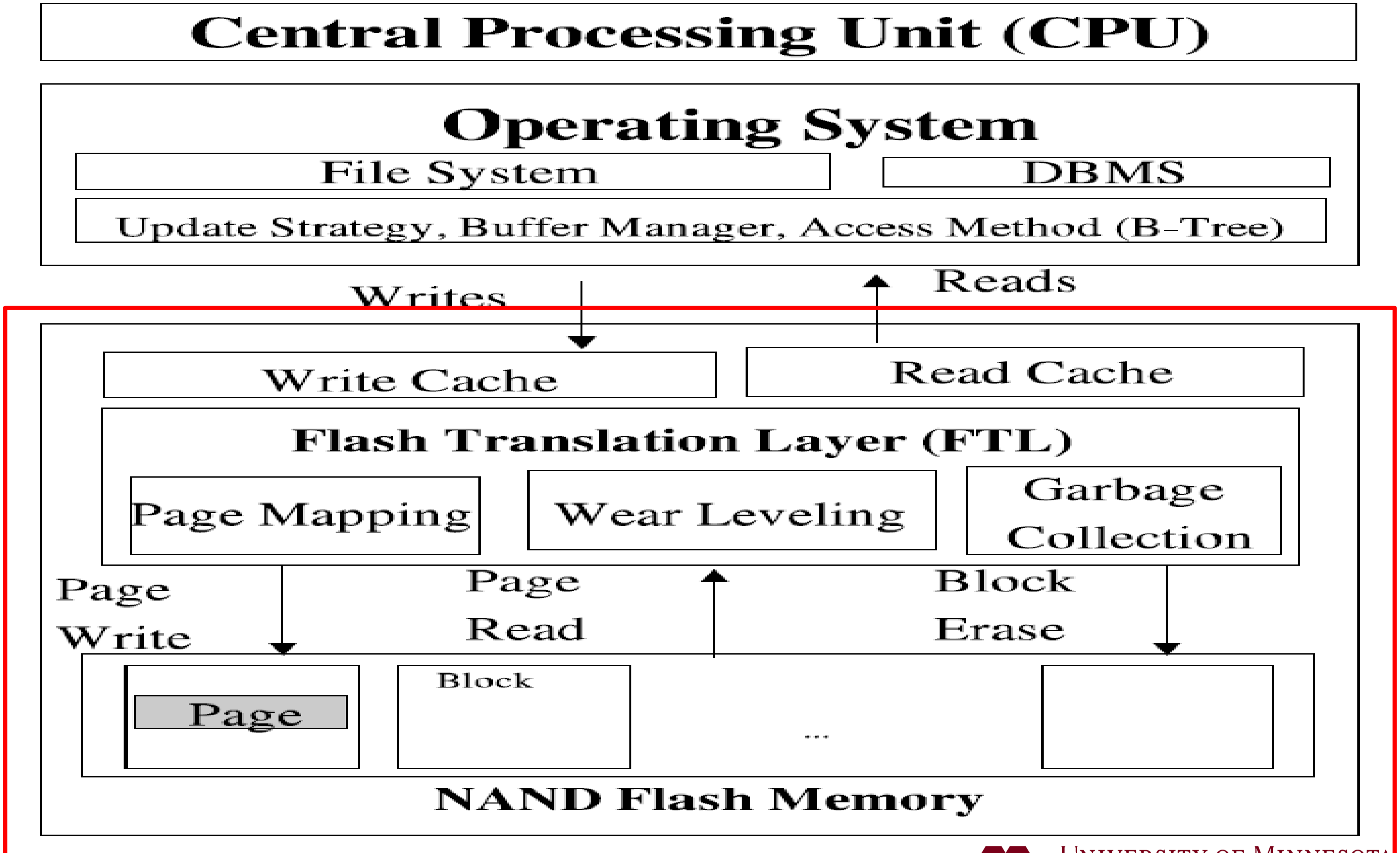


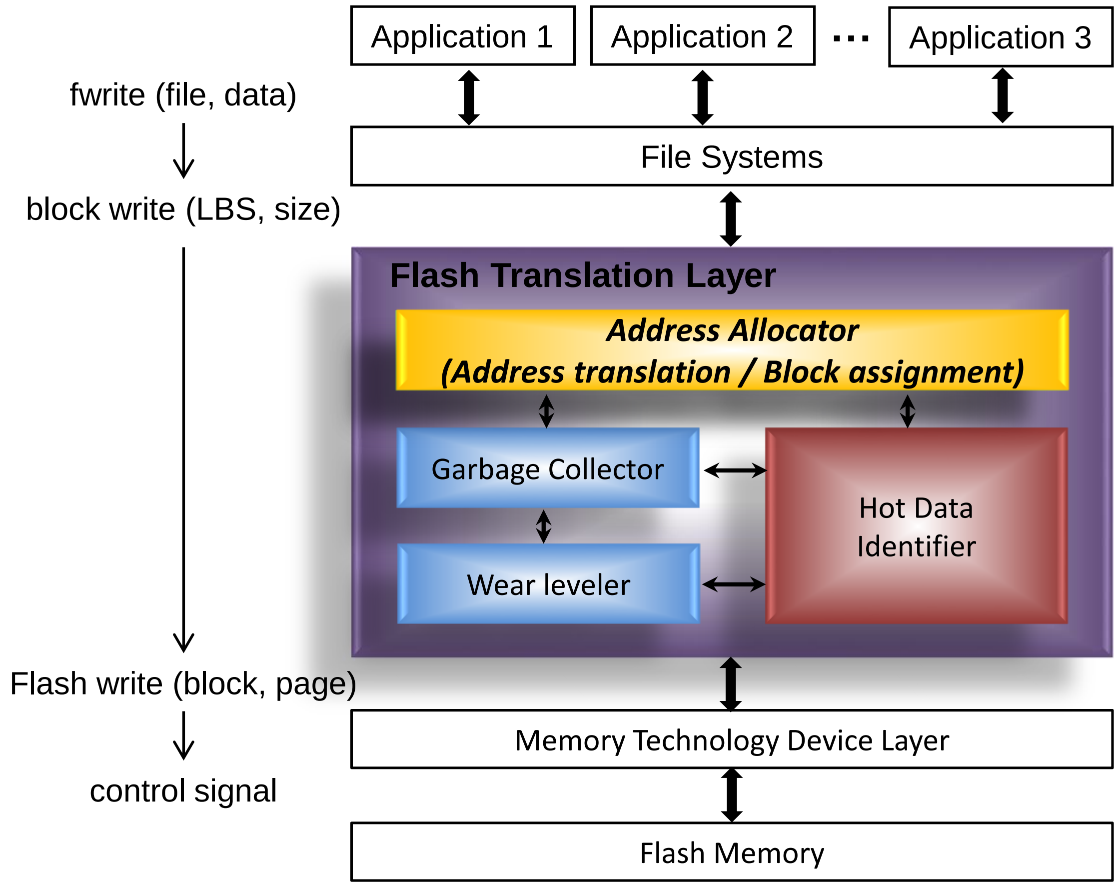
Figure Source: "**BPLRU: A Buffer Management Scheme for Improving Random Writes in Flash Storage", Hyojun Kim and Seongjun Ahn**, FAST 2008

High-Level View of Flash Memory Design



FTL (Flash Translation Layer)

Flash Translation Layer (FTL)

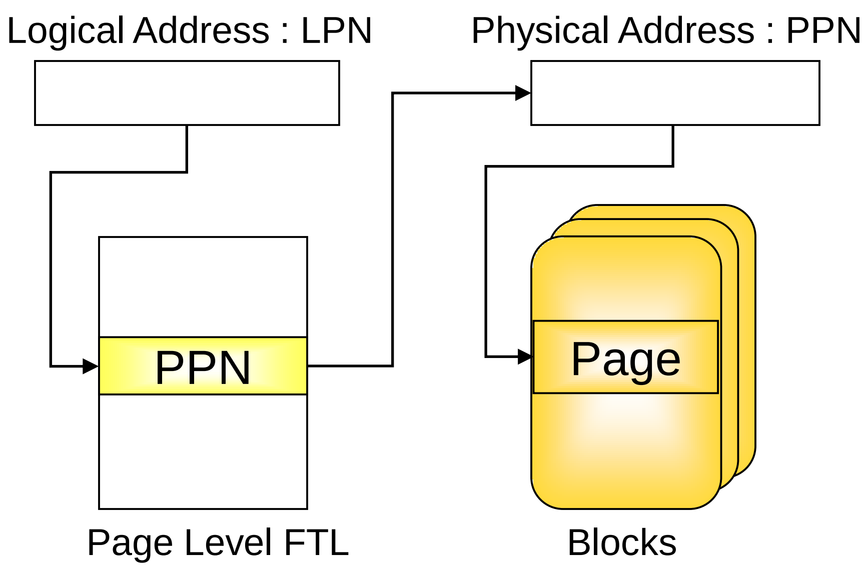


Flash Translation Layer (FTL)

- Flash Translation Layer
  - Emulates a block device interface
  - Hides the presence of erase operation/erase-before-write
  - Address translation, garbage collection, and wear-leveling
- Address Translation
  - Three types
    - Page-level, block-level, and hybrid mapping FTL
  - Mapping table is stored in small RAM within the flash device

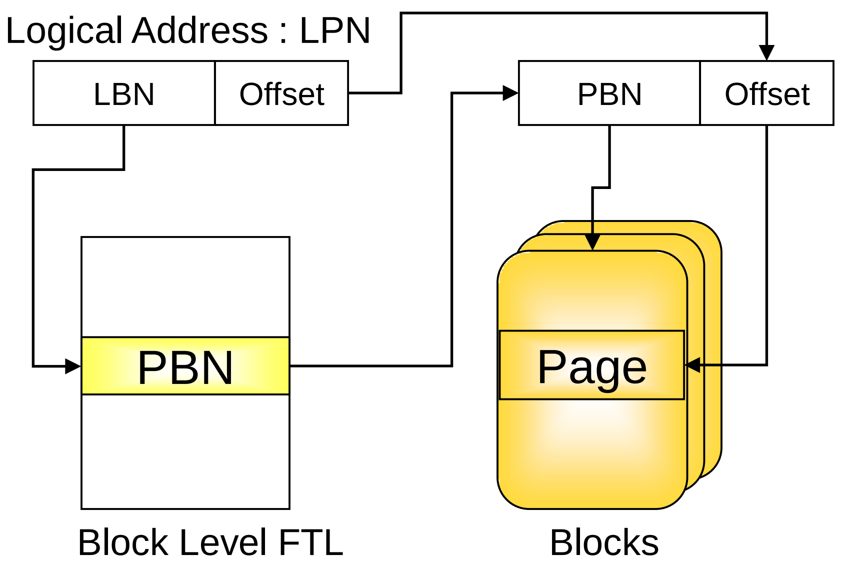
Page vs. Block Level Mapping

Page Level Mapping



Flexible but requires a lot of RAM (e.g., 2MB for 1GB SSD)

Block Level Mapping

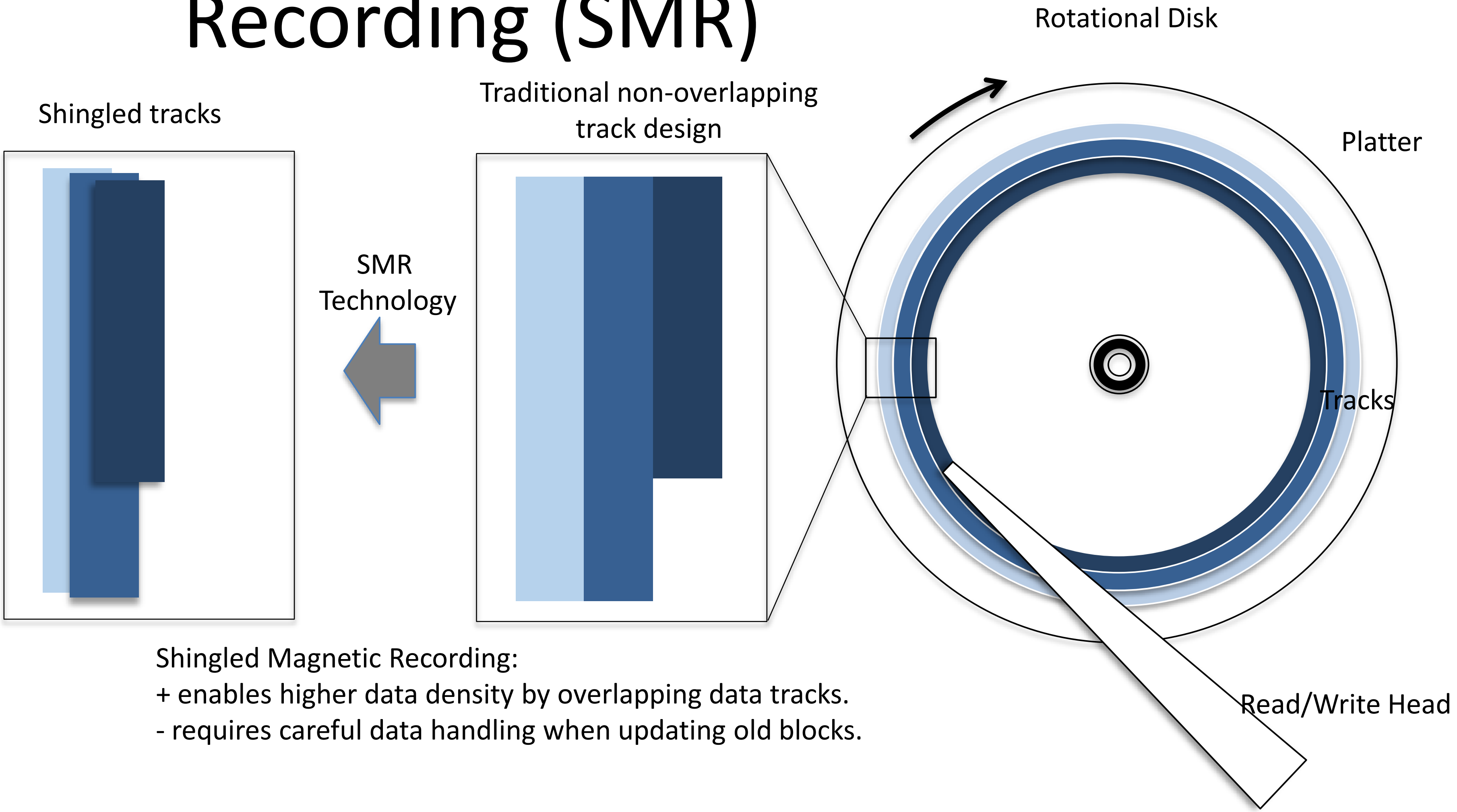


Less RAM (e.g., 32K for 1GB SSD), but inflexible in content placement

Emerging Disk Drives Including Shingled Magnetic Recording (SMR) Drives and Interlaced Magnetic Recording (IMR) Drives

University of Minnesota

Shingled Magnetic Recording (SMR)



Shingled Magnetic Recording:
+ enables higher data density by overlapping data tracks.
- requires careful data handling when updating old blocks.

University of Minnesota

T10 SMR Drive Models

- Drive Managed
  - Black box/drop-in solution: the drive handles all out-of-order write operations.
- Host Managed
  - White box/application modification needed: the drive reports zone layout information; out-of-order writes will be rejected.
- Host Aware
  - Grey box: the drive reports zone layout information; out-of-order writes will still be handled internally.
  - Applications can use HA-SMR drive as is, and also have the opportunity for zone-layout aware optimizations.

Hybrid SMR Basics

- Google's Proposal
  - 100GiB Volume creation. < 200ms, typically < 50ms. Query time < 50ms
- Seagate Flex API
  - In a basic unit of one zone. Or a consecutive zone extent.
- WD Realm API
  - 100GiB, same SMR size, but different CMR size.

Google's Proposal [Brewer'16, Tso '17]

Disks for Data Centers

White paper for FAST 2016

Eric Brewer, Lawrence Ying,
Lawrence Greenfield, Robert Cypher, and Theodore Ts'o
Google, Inc.

February 23, 2016
Version 1.1, revised February 29, 2016

Online at: http://research.google.com/pubs/pub44830.html

Hybrid-SMR Product Requirements Proposal for OCP

Theodore Ts'o

November 13, 2017

Introduction

This document describes proposed requirements for a Hybrid SMR drive where portions of the storage media can be dynamically converted between Conventional Magnetic Recording (CMR) and Shingled Magnetic Recording (SMR).

Must be usable as 100% CMR drive by Legacy Software

In the factory-default state, the HDD must be 100% backwards compatible with traditional CMR HDD's. For this reason, the number of accessible sectors from IDENTIFY DEVICE shall return the lesser size between: 100% CMR disk size or the size from SET ACCESSIBLE MAX ADDRESS EXT. In addition, GET NATIVE MAX ADDRESS EXT should return the 100% CMR disk size. The CMR capacity should conform to the industry specification SFF-8447 on HDD sizes.

Short-stroking compatibility

As with current CMR HDD's available on the market today, larger LBA sectors should roughly correspond with locations closer to the inner diameter (ID). This should be true for both LBA sector numbers in the CMR and SMR space; the smallest CMR or SMR sector numbers should correspond to physical sectors at the outer diameter, while the largest CMR or SMR sector numbers should correspond to physical sectors at the inner diameter. Replacement sectors in response to grown defects are an exception to this rule.

The reason for this requirement is so that the "hottest" CMR data can be located at the OD. Since SMR data tends to be cold, the SMR region will be located at the ID. (See below for more details about the CMR->SMR conversion.)

Support for other advanced HDD features

The Hybrid SMR drive must support other advanced HDD features, including Head Depop, TCG/Opal Storage Specification, ATA Security, GPL, NCQ, SMART, Sense Data Reporting, Write-Read-Verify, etc. Interactions with the SMR zones will be based on the ZAC specification, potentially with modifications as required to support the Hybrid SMR feature.

Google Document for OCP

- Must be usable as 100% CMR drive by Legacy Software
- SMR->CMR conversion
  - must be able to support converting a 100 GiB SMR volume back to CMR. OD->ID sequence is sufficient.
- CMR / SMR sector addressing (see fig.)
- CMR->SMR conversion
  - Must support the creation of 100 GiB SMR volumes (400 SMR zones)
  - May support smaller granularity
  - ID -> OD. SMR volume will be adjacent to previous one
- Performance Requirements
  - 100GiB SMR Volume Creation < 200ms
  - with typical conversion time < 50ms
  - Conversion back to CMR equally quick.
  - Query response < 50ms.
- Conversion Atomicity

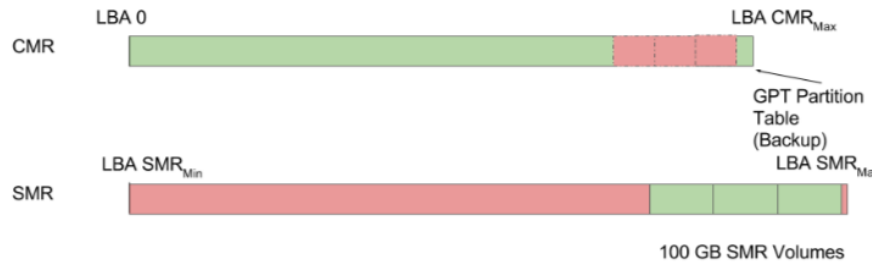
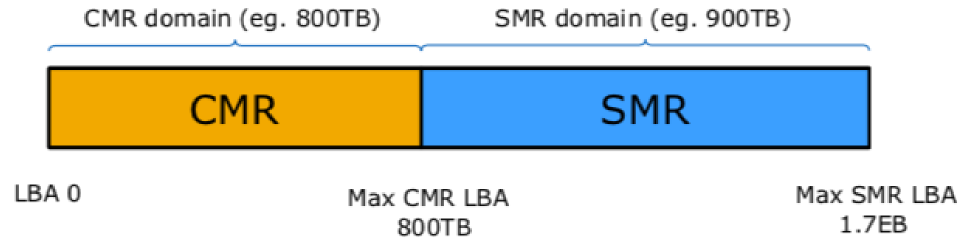


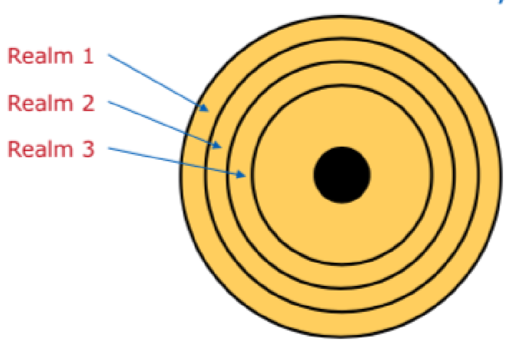
Fig. CMR / SMR sector addressing [Tso ‘17]

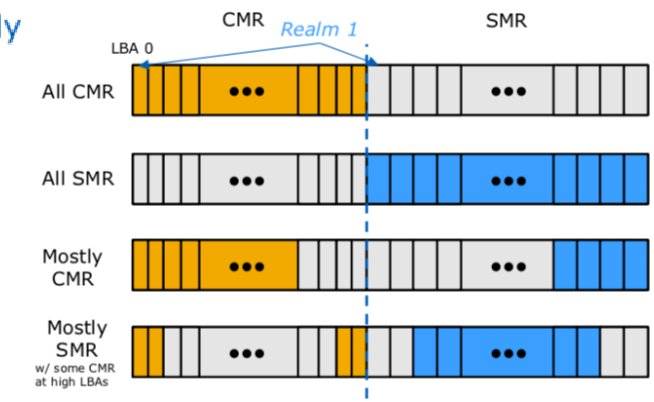
WD's Realm API [Boyle'17]



Realms

- A Realm is a physical portion of the device that stores user data
- A Realm is intended to be an allocation unit for the application client
- Each Realm is either CMR or SMR recording technology
  - CMR recording will provide less capacity than SMR recording
- A Realm is the conversion unit size between CMR and SMR modes
- All Realms in SMR mode will have the same capacity
- CMR Realms will likely vary in size but will always be less than the SMR size
- Realms can be converted between CMR <-> SMR by the host dynamically



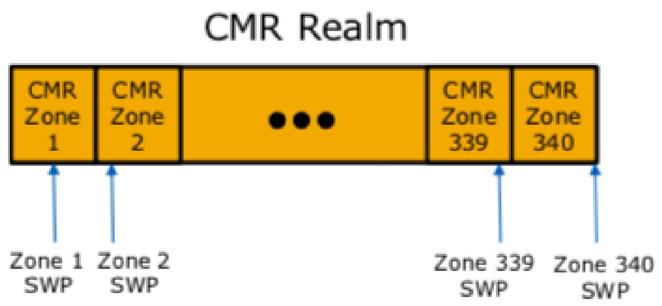


Western Digital

UNIVERSITY OF MINNESOTA

CMR Sequential Write Pointer

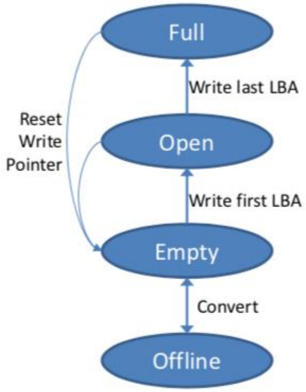
- To solve CMR Realm initialization issue, a CMR SWP is introduced
- Each CMR Zone will have a SWP to keep track of the progress of block initializations (written).
- Incoming command checking (optionally turned on by host)
  - Write commands are allowed to start <= SWP (aborted otherwise)
  - Read commands must address only LBAs that have been written
    - If checking is turned off, read commands succeed and return finishing pattern for unwritten data
- If command checking is disabled then drive will allow full random writes and reads with potential for longer latency first write commands of blocks.



Seagate Flex API [Feldman'17, Feldman'18]

Flex Zones

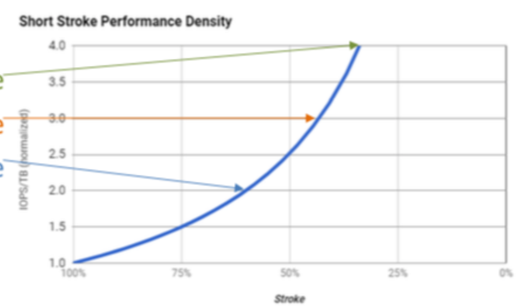
- Each zone is a 256-MiB extent of LBAs
  - Both the CMR and SMR spaces are made up of abutting zones
  - Each zone has its own write pointer indicating its write frontier
- Zones are the granularity of conversion
  - Each zone is online or offline
  - Online zones are provisioned with media; offline zones are not
  - A conversion command specifies a single extent of zones to take offline or to bring online
  - Application allocation unit is any integer number of zones



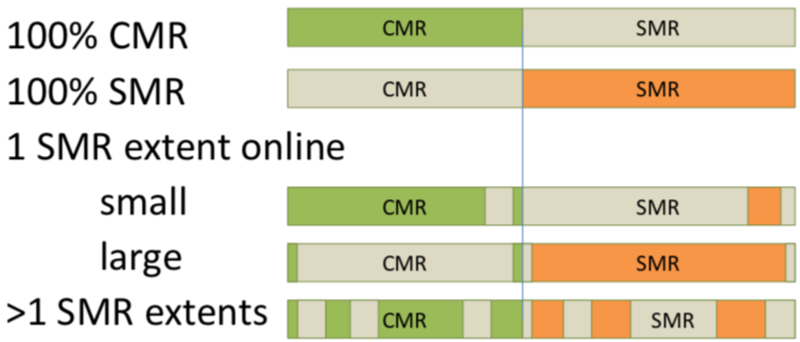
Problem Statement

- Hot data performance
  - Short stroking is powerful
    - 4x performance at 33% stroke
    - 3x performance at 44% stroke
    - 2x performance at 60% stroke

    but unused media represent wasted capital



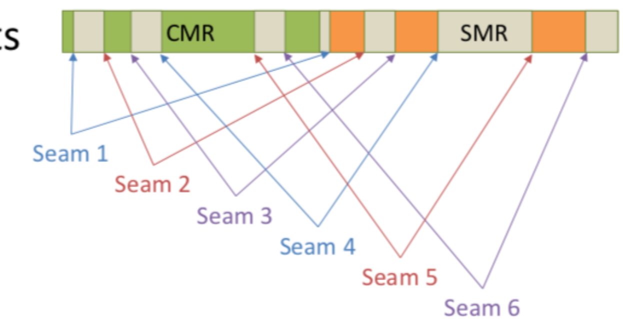
Flex Configuration Examples

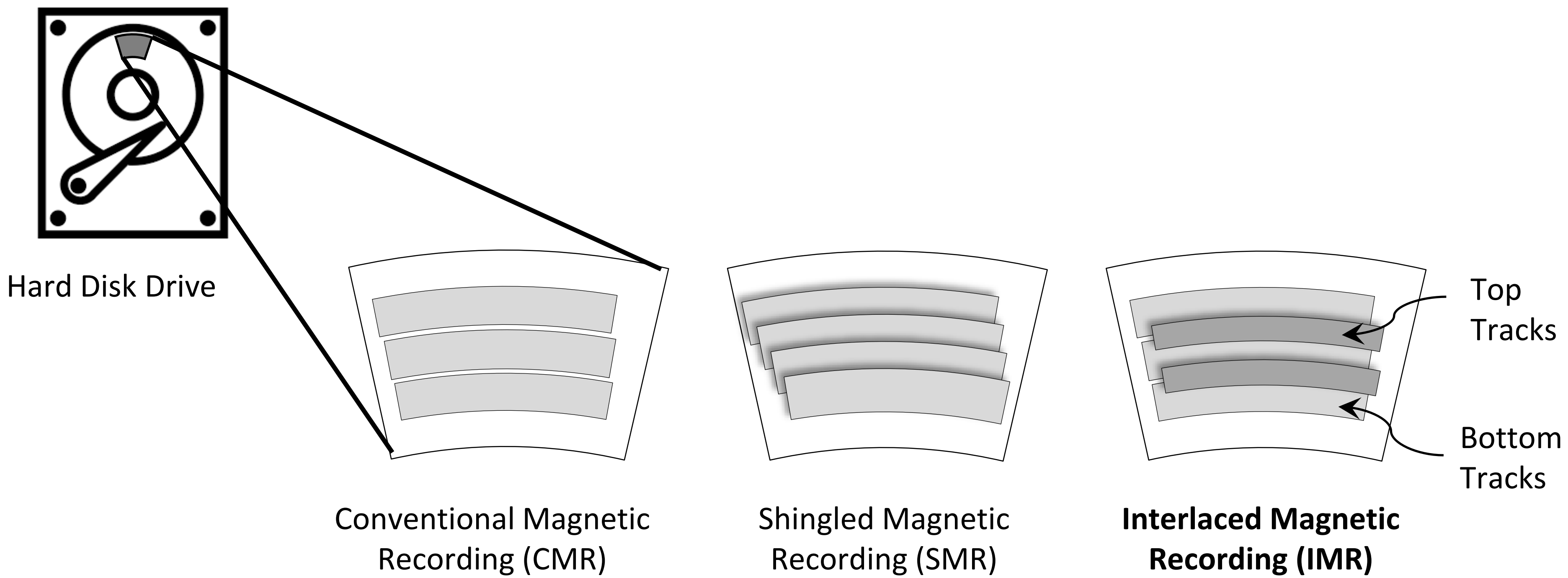


Flex Multiple Seam Example

3 SMR extents

6 seams

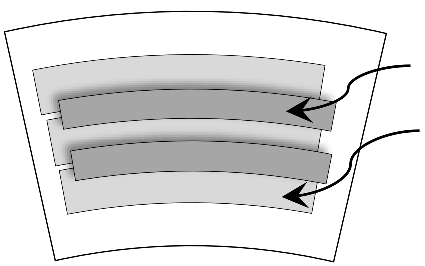




IMR: Higher areal data density than CMR, lower write amplification (WA) than SMR.

HDD icon image: https://www.flaticon.com/

IMR Tracks	Width	Laser Power	Data Density	Data Rate	Track Capacity
Bottom Tracks	wider	higher	higher(+27%)[1]	higher	higher
Top Tracks	narrower	lower	lower	lower	lower



Updating top tracks has no penalty

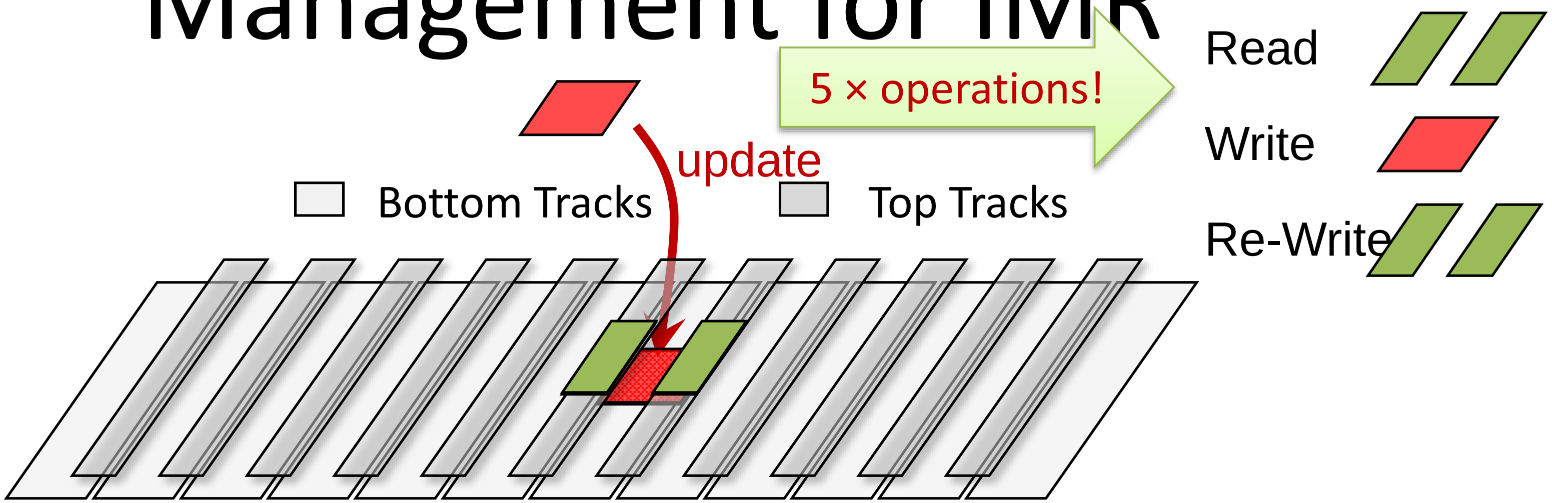
Updating bottom tracks causes Write Amplification (WA)

Only using bottom tracks when disk is not full may reduce WA.

I/O Performance depends on **disk usage**, and **layout design**.

[1]Granz et. al, 2017

TrackPly: Data and Space Management for IMR



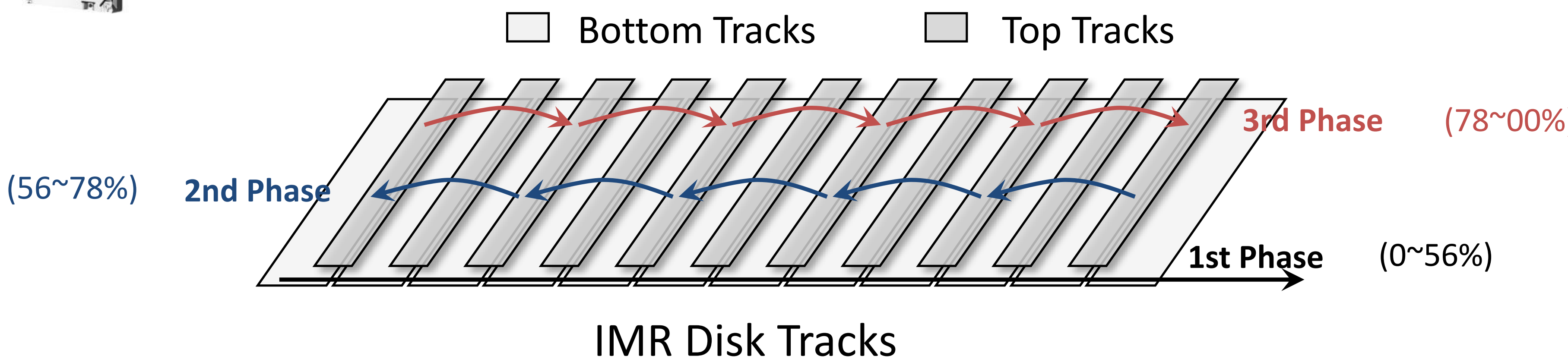
IMR Disk Tracks

Question: How serious is the **update overhead**?

Problem: how to efficiently use IMR drives and alleviate the update overhead?

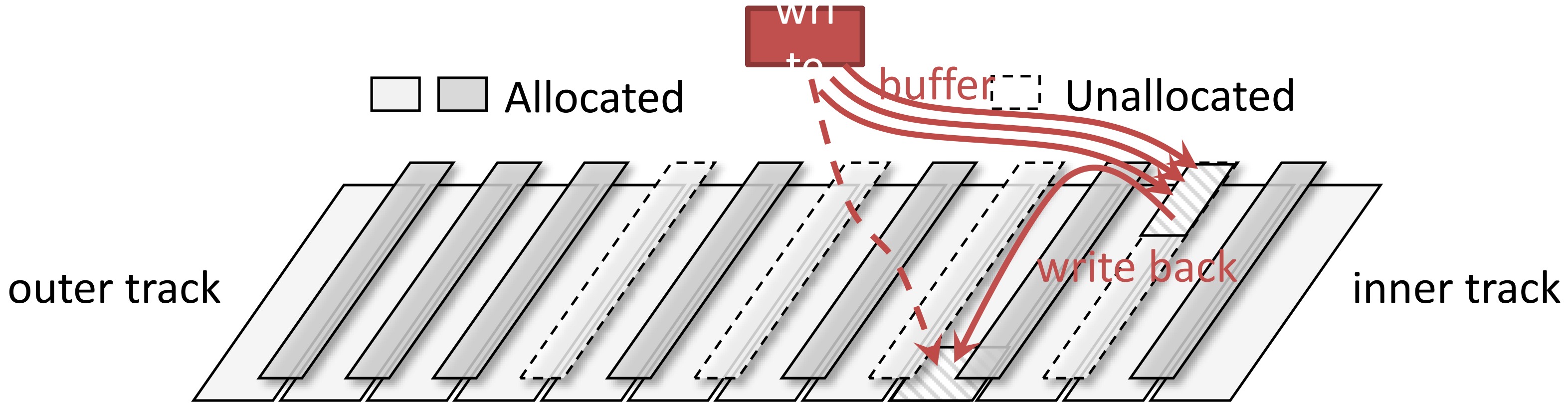
Design (1/3): Zigzag Allocation

Key Idea: the **data management** should depend on disk **usage** in High-Capacity HDDs.



Design (2/3): Top-Buffer

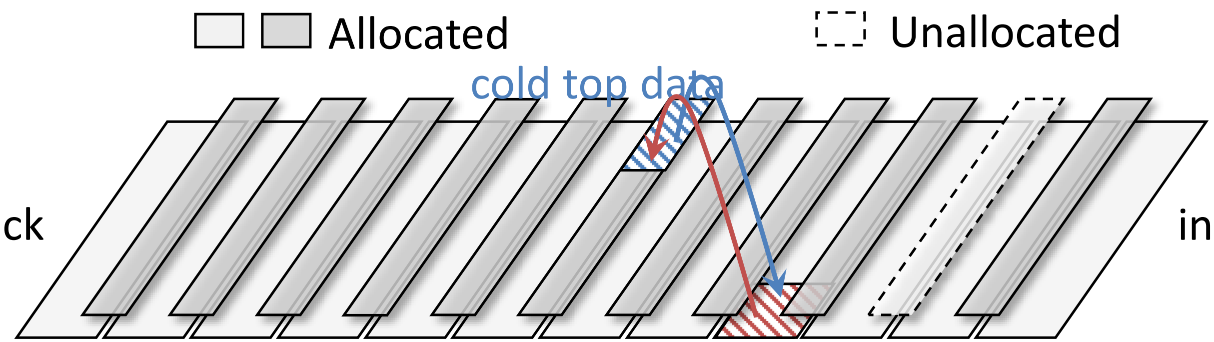
The idea: **buffer** -> **accumulate multiple** -> **writeback**



IMR Disk Tracks

Design (3/3): Block-Swap

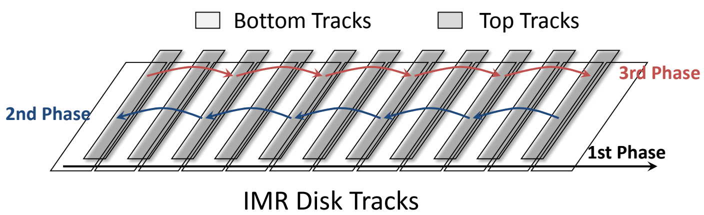
The idea: swap **hot** bottom-track data with **cold** top-track data.



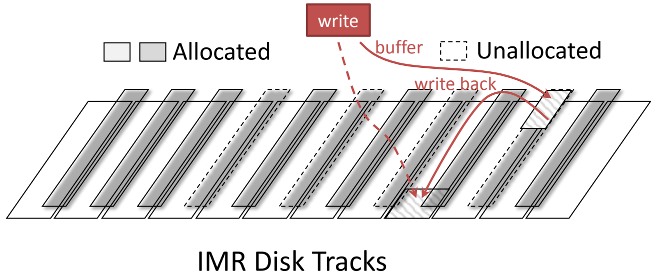
IMR Disk Tracks

University of Minnesota

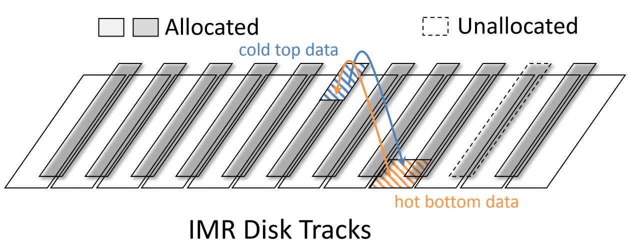
36



Zigzag Allocation: the **data management** should depend on disk **usage** in High-Capacity HDDs.



Top-Buffer: buffer and accumulate bottom-write requests into unallocated top tracks



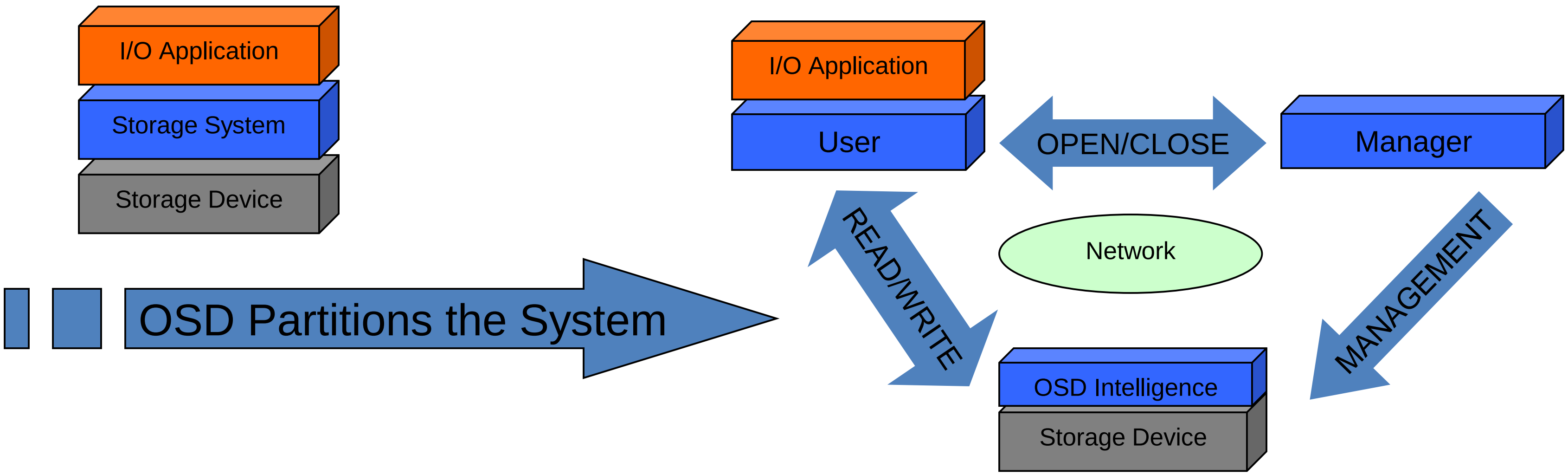
Block-Swap: swap **hot** bottom-track data with **cold**

Object Oriented Store
and
Active Storage

CRIS

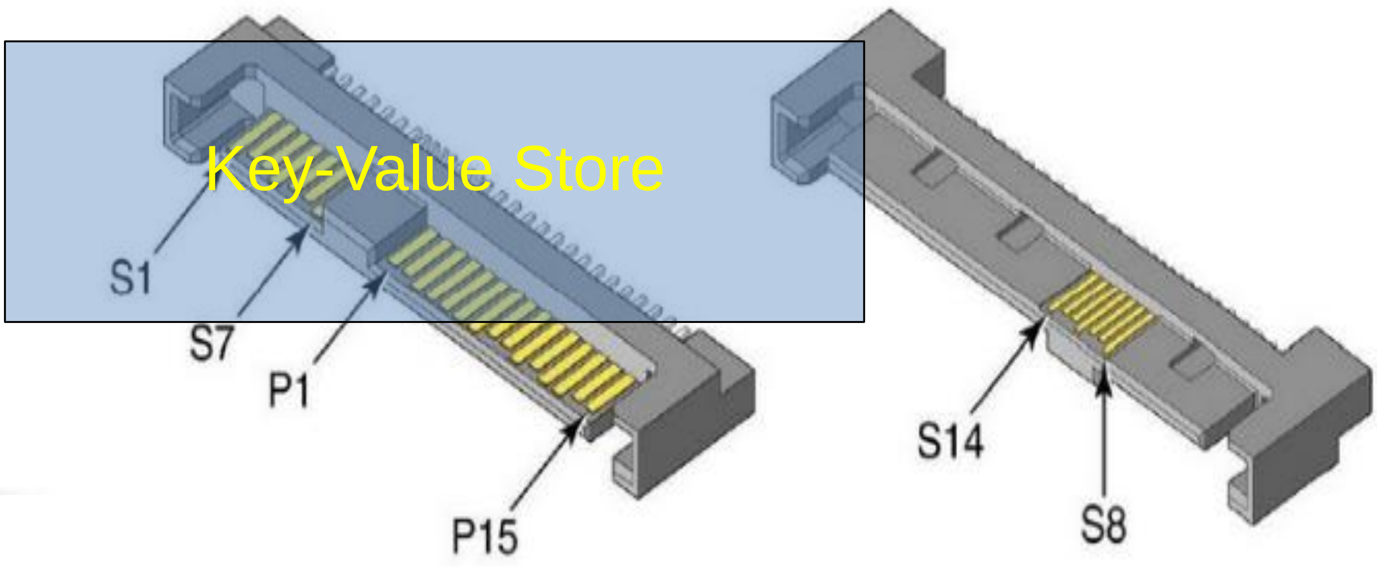
UNIVERSITY OF MINNESOTA

Active/Object Storage Device System Architecture (Internet Model)



The Manager is *not* in the data path.

Kinetic Drives
Implementing An Application on Storage Device



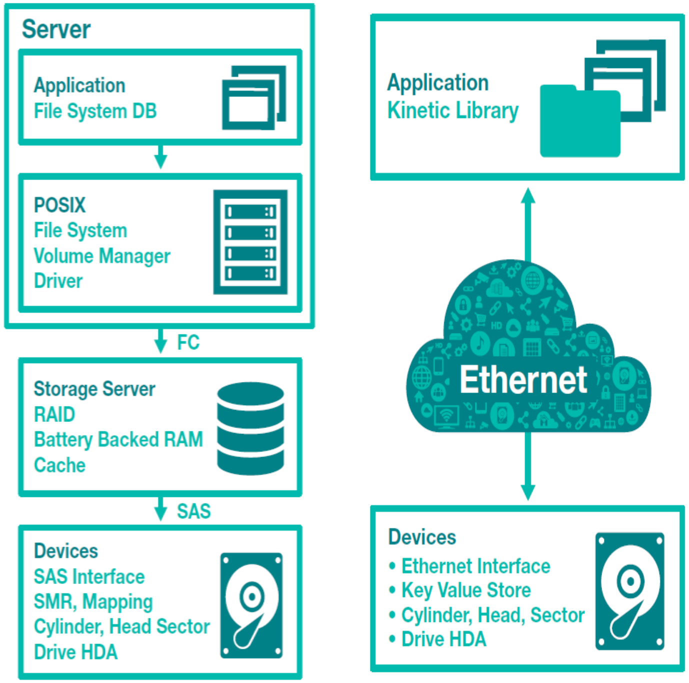
Kinetic Drives (Key-Value Store)

- Nowadays, Key-value store is becoming popular (e.g., Amazon, Facebook, LinkedIn).

Key	Value

- Kinetic Drives provide storage for key-value based operations via direct Ethernet connections without storage servers, which can reduce the management complexity.

- It is important to scale the Kinetic Drives to a global key-value store system which can provide service for worldwide users.

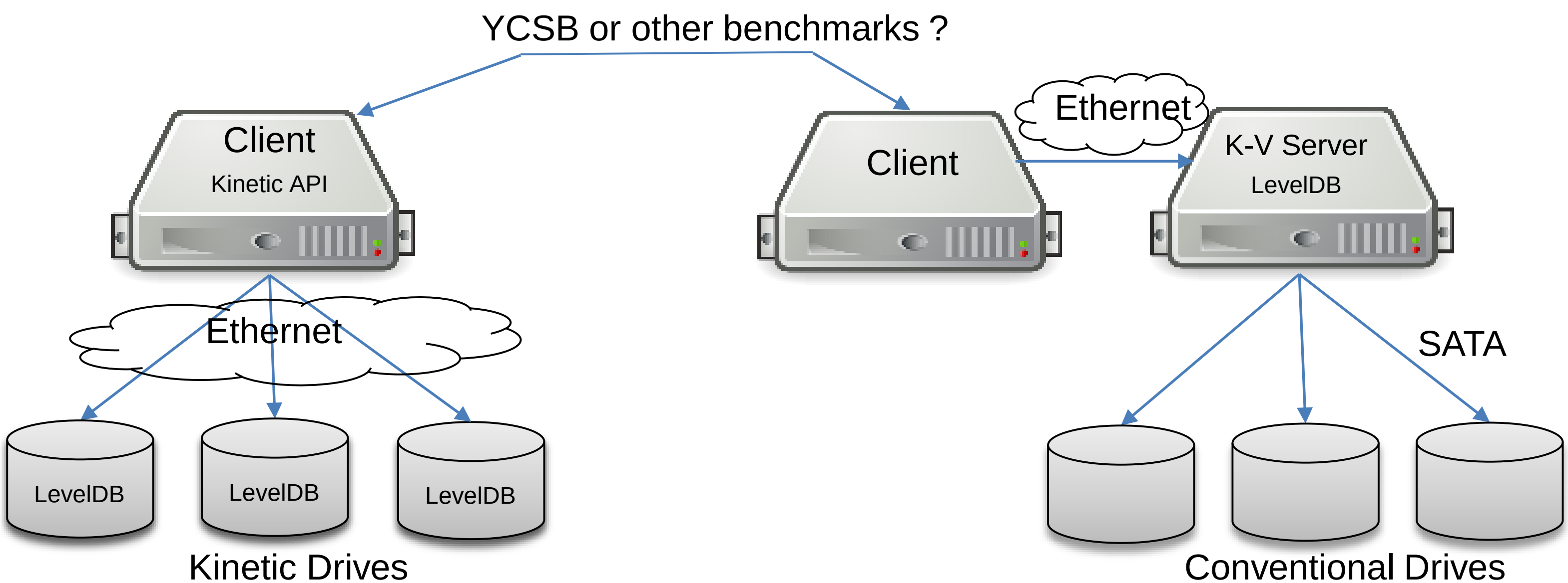


Traditional Storage Stack

Kinetic Storage Stack

Measure Performance of LevelDB

- Install LevelDB on a server with conventional drives
- Run a common benchmark and test the performance
  - YCSB?
  - Other benchmarks?
- Performance metrics – Throughput , Latency, Reads, or Writes?



New Type of Tape Drives

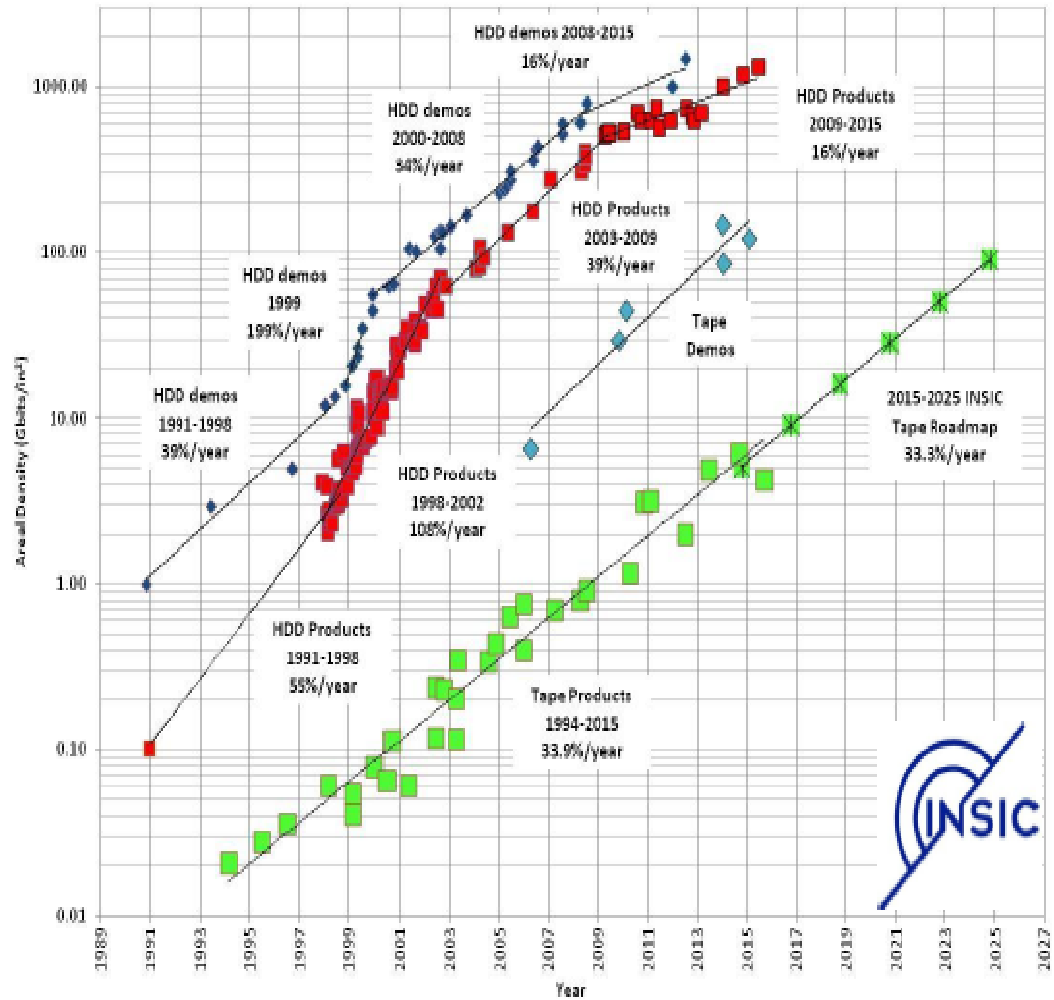
Why Tape Drives

Tape vs Disk

- Cost effective at scale (50-75% less than disk)
- Requires no energy at rest
- 10-100 times more reliable
- Strong density road map

Research Motivation

- Gap in research
- New tape features
- Data explosion, cost factor
- Latency issue, big improvement potential

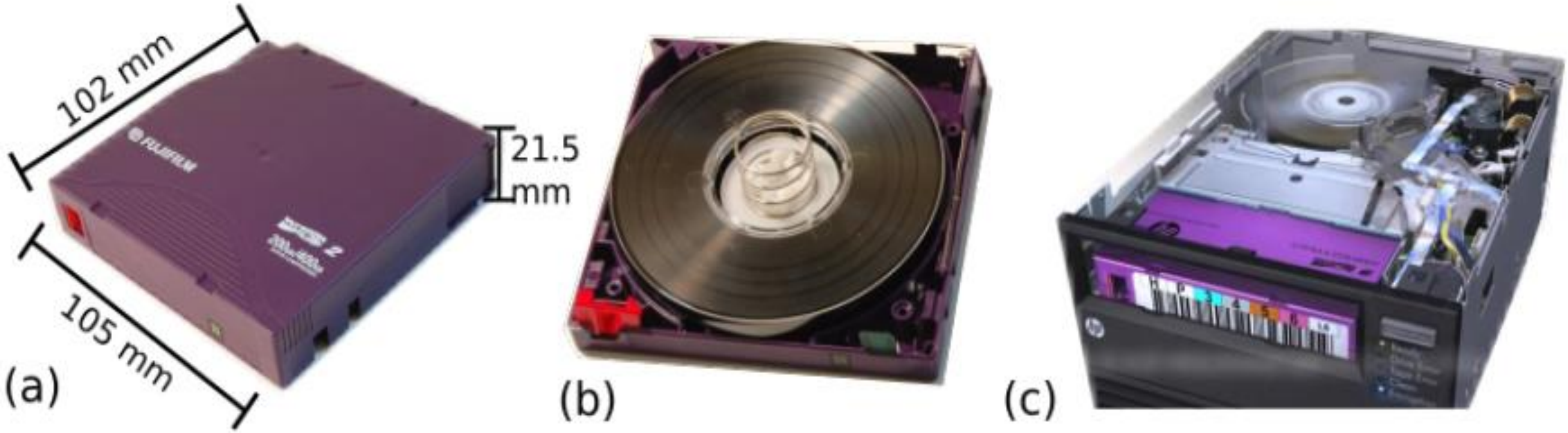


www.insic.org/news/2015 roadmap/15 index.html

Archival Storage Devices

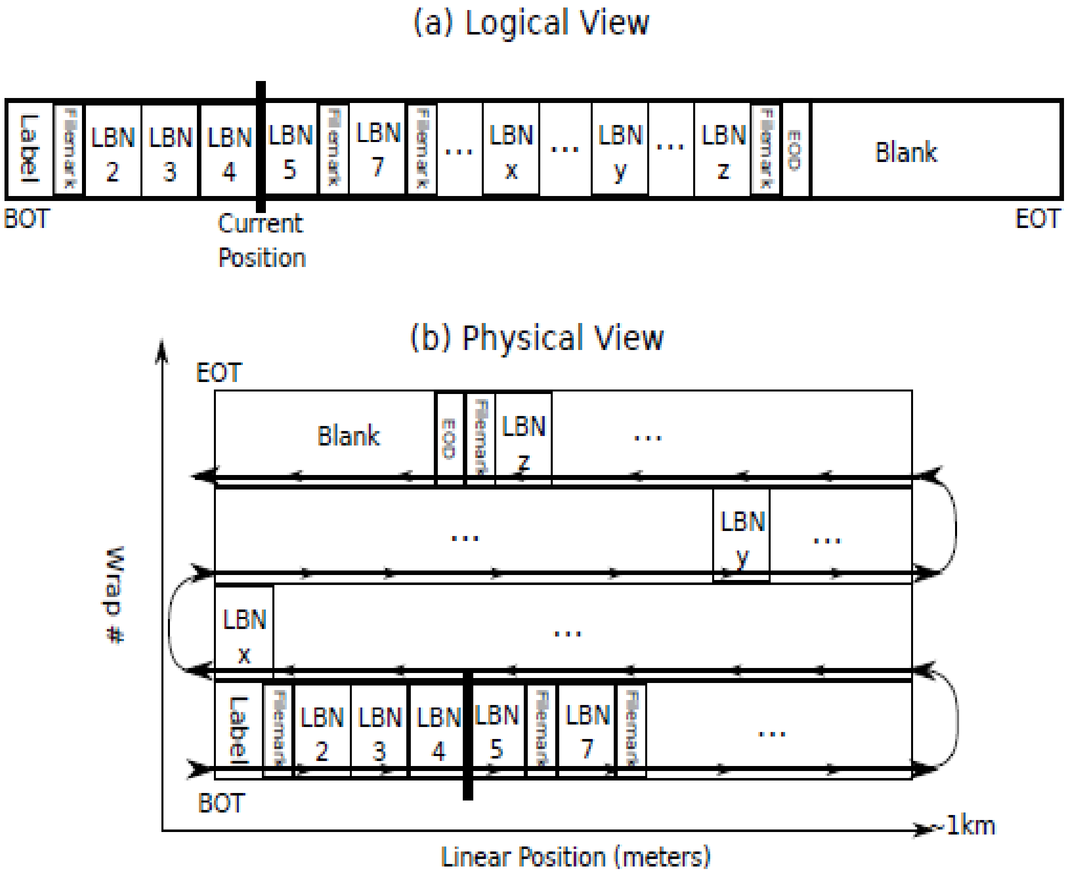
	Device Properties	Latency	MB/s	$/GB	TB
LTO-8 Tape	Removable, sequential	60 s	360	.015	12
Blu-ray BD-R Disc	Removable, random	180 ms	54	.049	.1
Archive SMR HDD	Fixed, sequential writes	4 ms	233	.029	15
Enterprise HDD	Fixed, random	4 ms	248	.031	12
SATA SSD	Fixed, random	100 µs	500	.199	4

Tape Cartridge



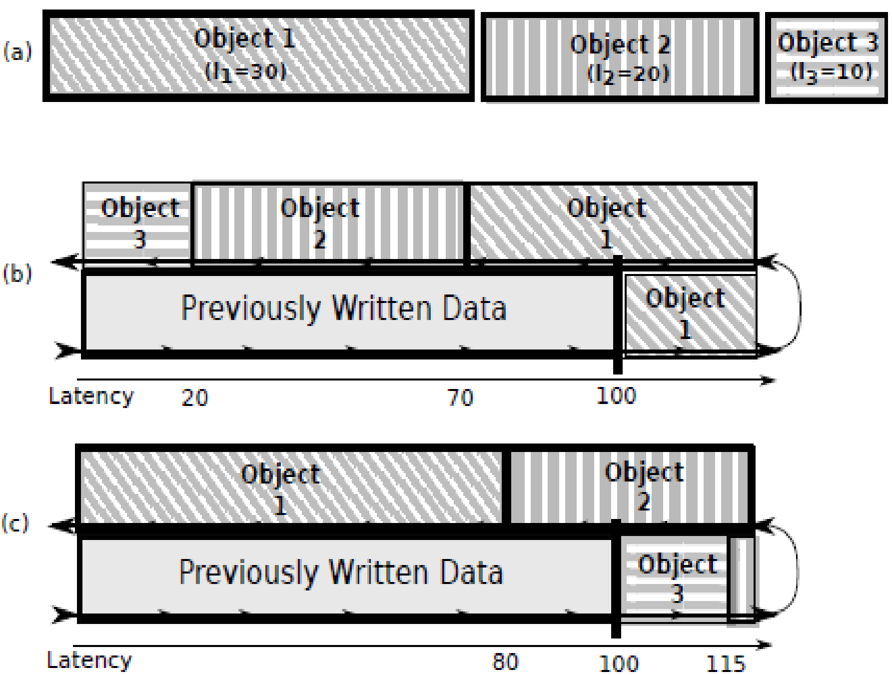
- **a** LTO cartridge (~4 in x 4 in x .75 in)
- **b** Internal view of cartridge (~1 km of tape)
- **c** Cartridge inserted into tape drive

Tape Model



- **a** Logical API
  - Block Numbers (LBNs)
  - Filemark-based Records
  - Seek Records or LBNs
- **b** Logical to Physical
  - Multiple Record Passes
  - Logical Order $\neq$ Linear Position Order

Write Order Optimization



Write Order Motivation

- Linear position determines latency
- Place objects near required latency
- Cost is difference between required and expected latency

Write Order Example

a. Three objects required latency $l_1 = 30, l_2 = 20, l_1 = 10$.
   Current Tape Position Latency: 100
b. Order: (1,2,3)
   Cost: $70 + 50 + 10 = \mathbf{150}$
c. Order: (3,2,1)
   Cost: $50 + 95 + 90 = \mathbf{235}$

Thank You!

Questions?