Asymmetric RAID: Rethinking RAID for SSD Heterogeneity

Ziyang Jiao, Bryan S. Kim

Syracuse University

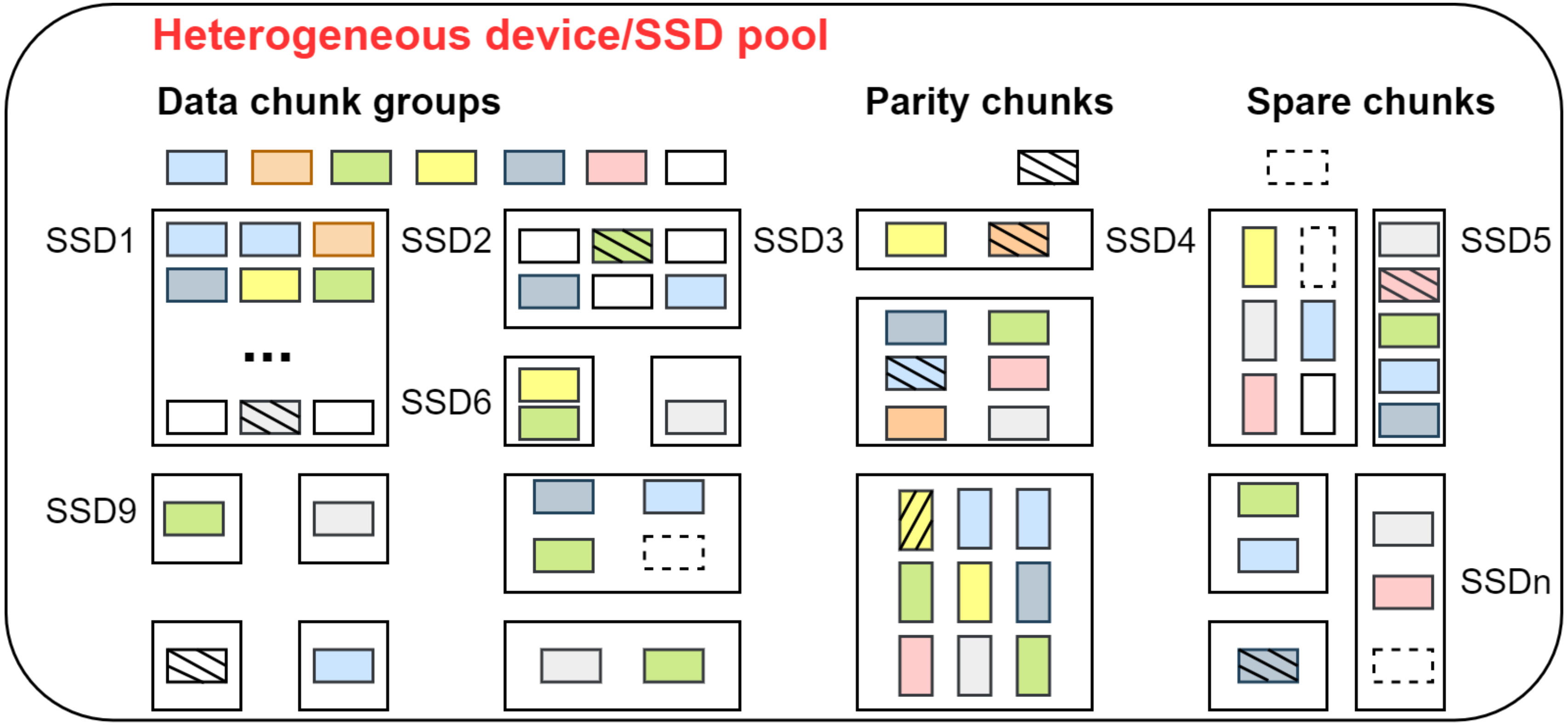
Syracuse University
College of Engineering
& Computer Science

HotStorage 2024

The 16th ACM Workshop on Hot Topics in Storage and File Systems (HotStorage'24)

Asymmetric RAID

- Optimize storage utilization by leveraging a mix of heterogeneous devices
- Asymmetrically distribute data across the disk array



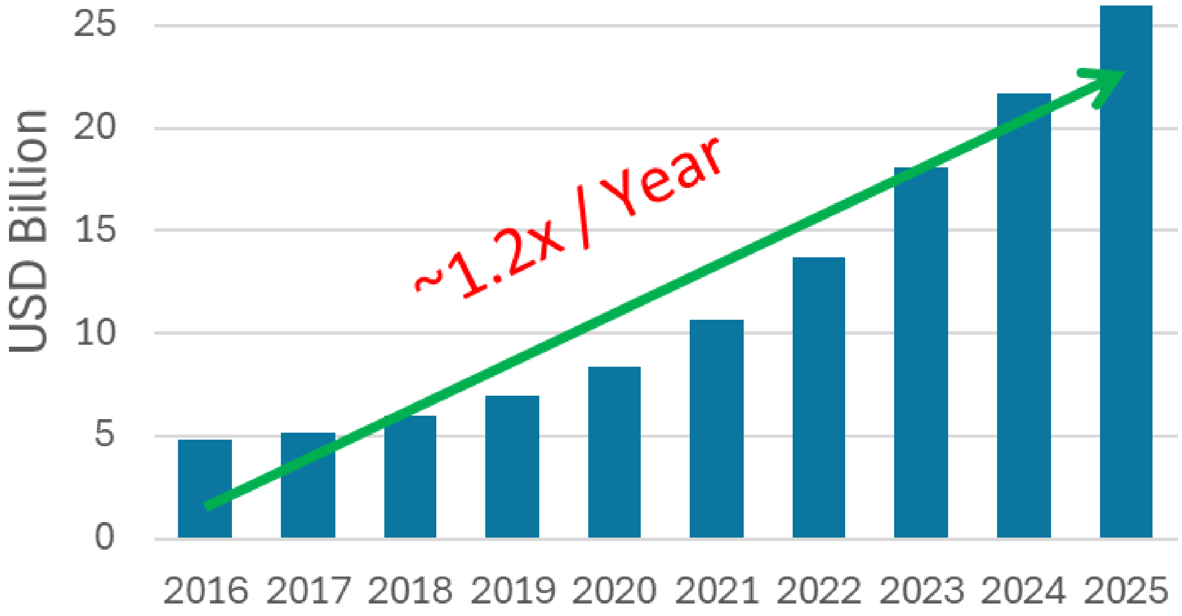
Outline

- All-Flash Array Systems
- AFA with Heterogeneous Devices
- Asymmetric RAID
- Ongoing Work
- Conclusion

All-flash arrays (AFAs)

- Storage infrastructure that uses only SSDs
  - High performance
  - Low latency
  - Better reliability

- Global AFA market

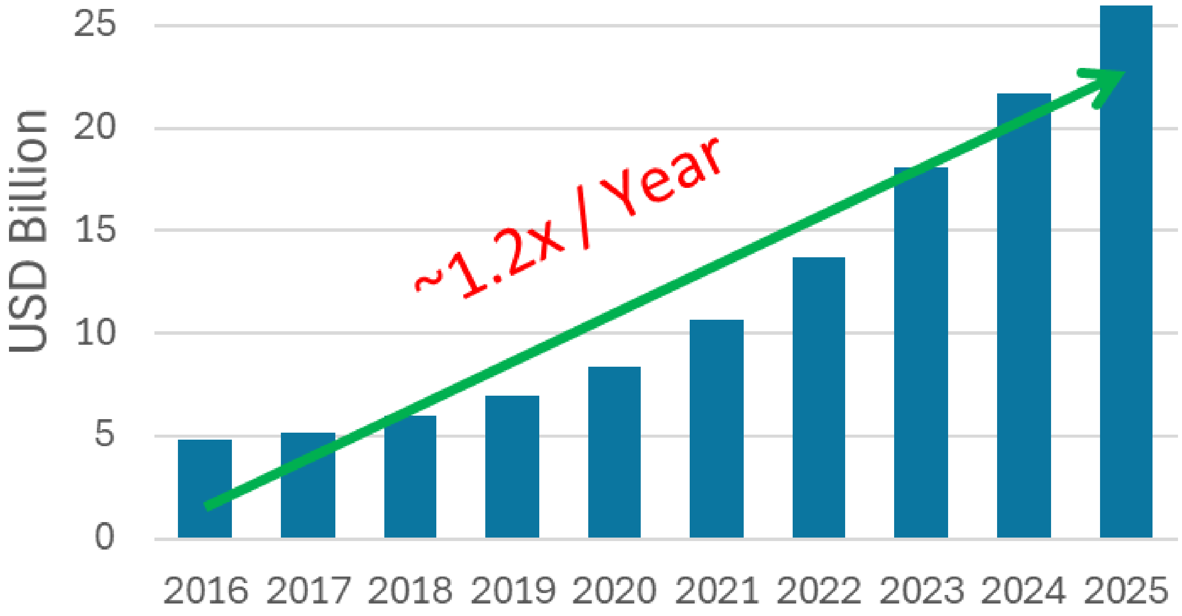


- Data source: FusionRAID [FAST'21] & www.marketsandmarkets.com
- Images from Google search

All-flash arrays (AFAs)

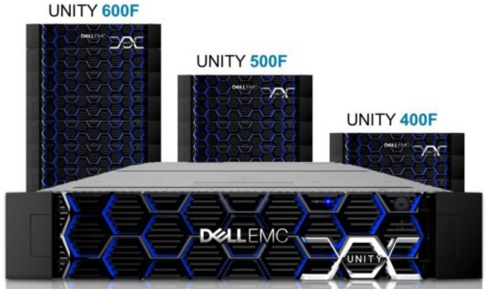
- Storage infrastructure that uses only SSDs
  - High performance
  - Low latency
  - Better reliability

- Global AFA market

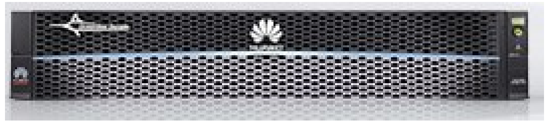


PURESTORAGE

DELL Technologies



HUAWEI



NetApp

- Images from Google search

Existing AFA solutions

- Existing AFA solutions spread I/O to the disk pool in a balanced manner.
  - ✓ I/O parallelism
  - ✓ Throughput
  - ✓ Data reliability

	Write Strategy	Disk Organization	Issue tackled
Linux-MD	In-place write	RAID	—
SWAN [ATC '19]	Log write	2D Array	GC interference
IODA [SOSP '21]	In-place write	RAID-5/6	GC interference
RAID+ [FAST '18]	In-place write	MOLS-based	Disk partitioning
FusionRAID [FAST '21]	Log write	Pool	I/O determinism
StRAID [ATC '22]	In-place write	RAID	I/O concurrency
Diff-RAID [EuroSys '10]	In-place write	RAID	Correlated failures
HeART [FAST '19]	In-place/log write	Pool	System reliability
Pacemaker [OSDI '20]	In-place/log write	Pool	System reliability
Tiger [OSDI '22]	In-place/log write	Pool	System reliability

Existing AFA solutions

- Existing AFA solutions spread I/O to the disk pool in a balanced manner
  - ✓ I/O parallelism
  - ✓ Throughput
  - ✓ Data reliability

→ Assume that storage components are homogeneous
- Performance and capacity

	Write Strategy	Disk Organization	Issue tackled
Linux-MD	In-place write	RAID	—
SWAN [ATC '19]	Log write	2D Array	GC interference
IODA [SOSP '21]	In-place write	RAID-5/6	GC interference
RAID+ [FAST '18]	In-place write	MOLS-based	Disk partitioning
FusionRAID [FAST '21]	Log write	Pool	I/O determinism
StRAID [ATC '22]	In-place write	RAID	I/O concurrency
Diff-RAID [EuroSys '10]	In-place write	RAID	Correlated failures
HeART [FAST '19]	In-place/log write	Pool	System reliability
Pacemaker [OSDI '20]	In-place/log write	Pool	System reliability
Tiger [OSDI '22]	In-place/log write	Pool	System reliability

Existing AFA solutions

- Existing AFA solutions spread I/O to the disk pool in a balanced manner
  - ✓ I/O parallelism
  - ✓ Throughput
  - ✓ Data reliability

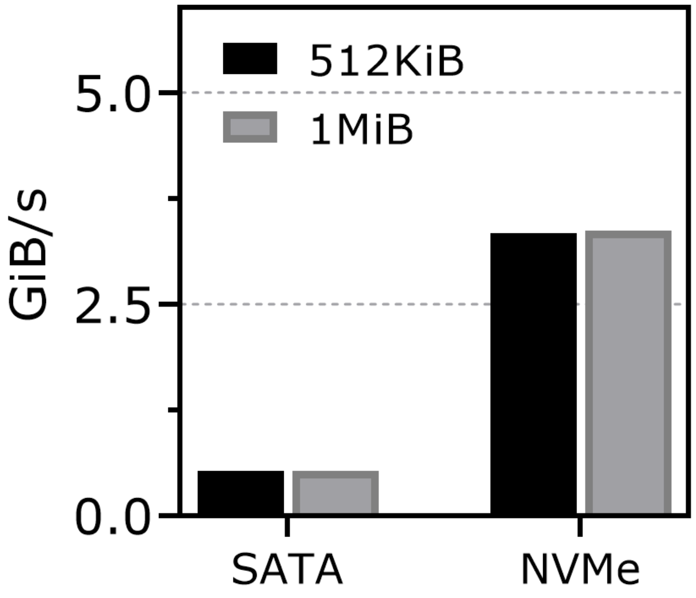
→ Assume that storage components are homogeneous
- Performance and capacity

	Write Strategy	Disk Organization	Issue tackled	Disk Heterogeneity
Linux-MD	In-place write	RAID	—	Low
SWAN [ATC '19]	Log write	2D Array	GC interference	Low
IODA [SOSP '21]	In-place write	RAID-5/6	GC interference	Low
RAID+ [FAST '18]	In-place write	MOLS-based	Disk partitioning	Low
FusionRAID [FAST '21]	Log write	Pool	I/O determinism	Low
StRAID [ATC '22]	In-place write	RAID	I/O concurrency	Low
Diff-RAID [EuroSys '10]	In-place write	RAID	Correlated failures	Low
HeART [FAST '19]	In-place/log write	Pool	System reliability	Medium
Pacemaker [OSDI '20]	In-place/log write	Pool	System reliability	Medium
Tiger [OSDI '22]	In-place/log write	Pool	System reliability	Medium

Low disk utilization when considering disk heterogeneity

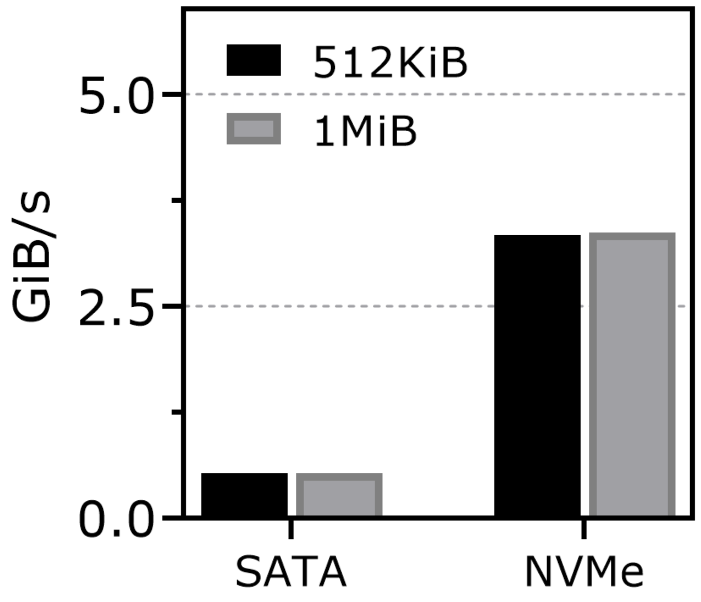
AFA with heterogeneous devices

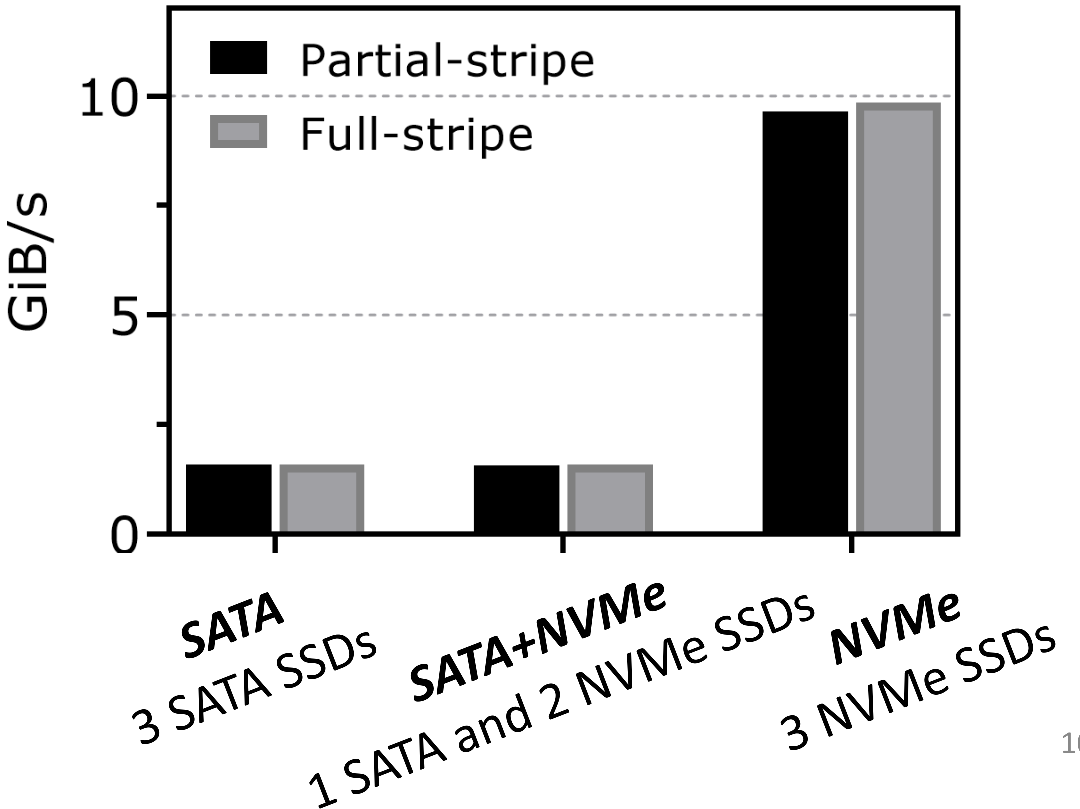
- What if include a heterogeneous mix of devices?
    - NVMe: Samsung PM9A3
    - SATA: Samsung PM1645



AFA with heterogeneous devices

- What if include a heterogeneous mix of devices?
  - NVMe: Samsung PM9A3
  - SATA: Samsung PM1645

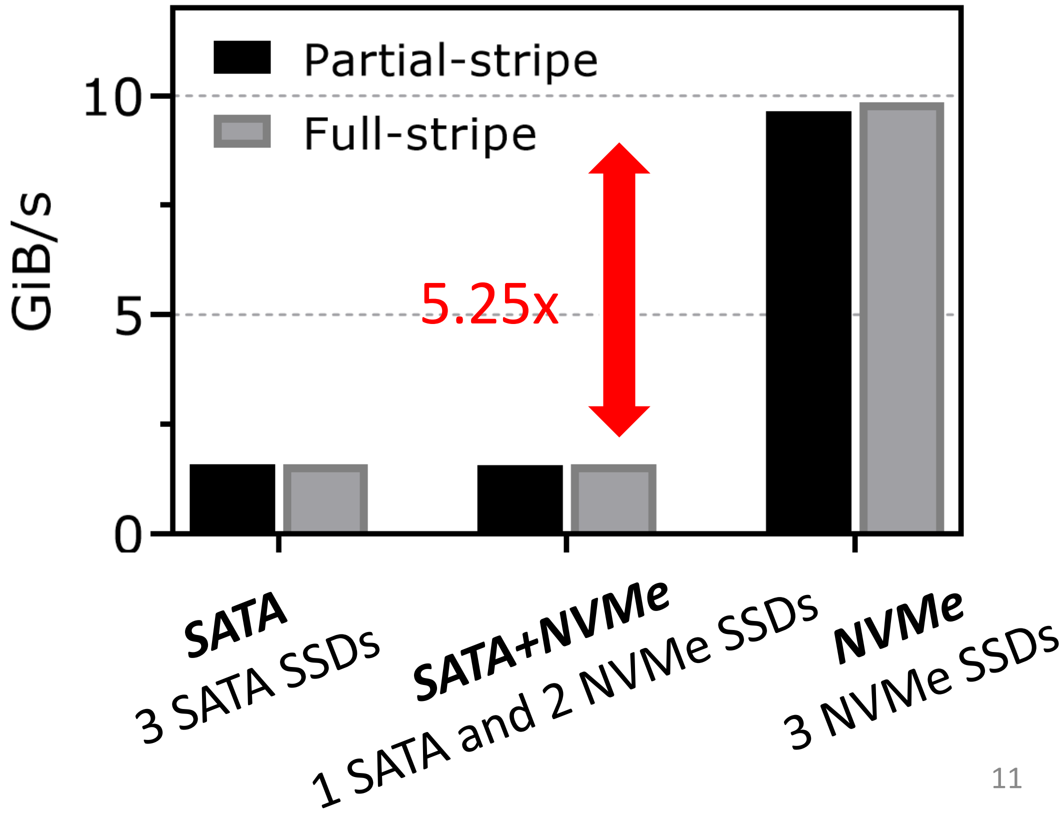




AFA with heterogeneous devices

- What if include a heterogeneous mix of devices?
  - NVMe: Samsung PM9A3
  - SATA: Samsung PM1645

Significant storage under-utilization:
- Performance is bottlenecked by the poor-performing drives;
- Capacity is determined by the minimal capacity device.



Is this a common issue?

- Heterogeneous storage devices are ubiquitous
  - Linux-MD: supporting arrays with more than 384 component devices
  - NetApp: SSDs with varying deployment times [FAST ‘20]
  - Alibaba Cloud: 12 to 18 SSDs from multiple vendors [ATC ‘19]
  - …

Is this a common issue?

- Heterogeneous storage devices are ubiquitous
  - Linux-MD: supporting arrays with more than 384 component devices
  - NetApp: SSDs with varying deployment times [FAST ‘20]
  - Alibaba Cloud: 12 to 18 SSDs from multiple vendors [ATC ‘19]
- The challenge persists even among disks of identical models
  - Performance variability from manufacturing

Is this a common issue?

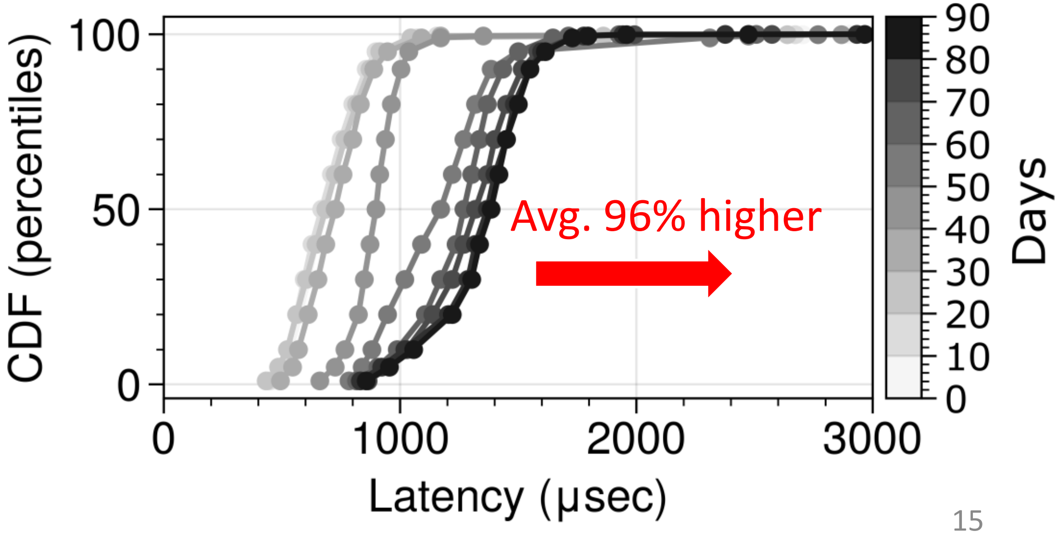
- Heterogeneous storage devices are ubiquitous
  - Linux-MD: supporting arrays with more than 384 component devices
  - NetApp: SSDs with varying deployment times [FAST ‘20]
  - Alibaba Cloud: 12 to 18 SSDs from multiple vendors [ATC ‘19]
- The challenge persists even among disks of identical models
  - Performance variability from manufacturing
  - Device aging
    - Dell Datacenter NVMe Drive
    - 3D TLC NAND

Is this a common issue?

- Heterogeneous storage devices are ubiquitous
  - Linux-MD: supporting arrays with more than 384 component devices
  - NetApp: SSDs with varying deployment times [FAST '20]
  - Alibaba Cloud: 12 to 18 SSDs from multiple vendors [ATC '19]
- The challenge persists even among disks of identical models
  - Performance variability from manufacturing
  - Device aging
    - Dell Datacenter NVMe Drive
    - 3D TLC NAND

Aging phase:
- ~100 TB random writes/day

Measuring phase:
- Read-only workload with high IO depth
- Avoid the impact of GC and host
- Fail-slow symptoms

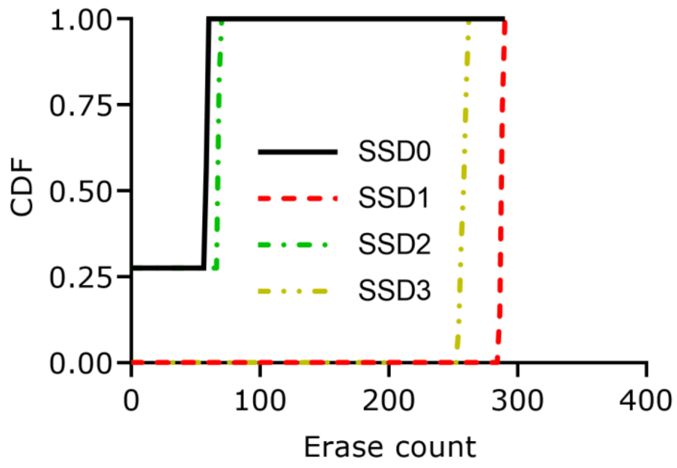


Is this a common issue?

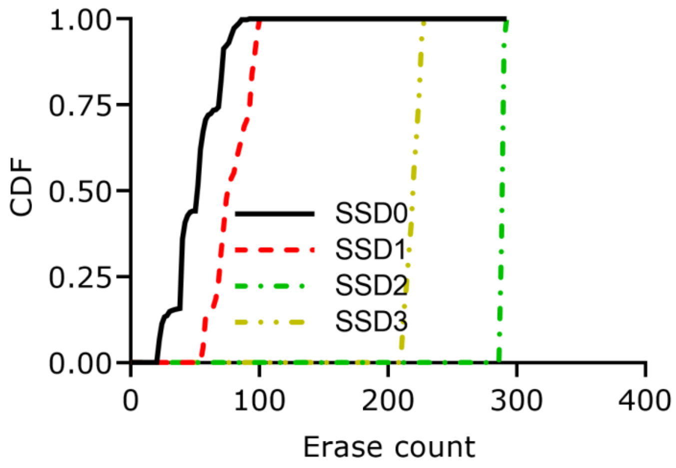
- SSDs can experience varying levels of degradation within RAID configurations.
  - E.g., with skewed/partial-stripe workloads

Is this a common issue?

- SSDs can experience varying levels of degradation within RAID configurations.
  - E.g., with skewed/partial-stripe workloads
- Experiments using FEMU
  - RAID: RAID-5 with 4 identical SSDs.
  - SSD: 32 GiB physical capacity (OP = 14%).



Zipfian with utilization of 30%



Zipfian with utilization of 70%

Asymmetric RAID

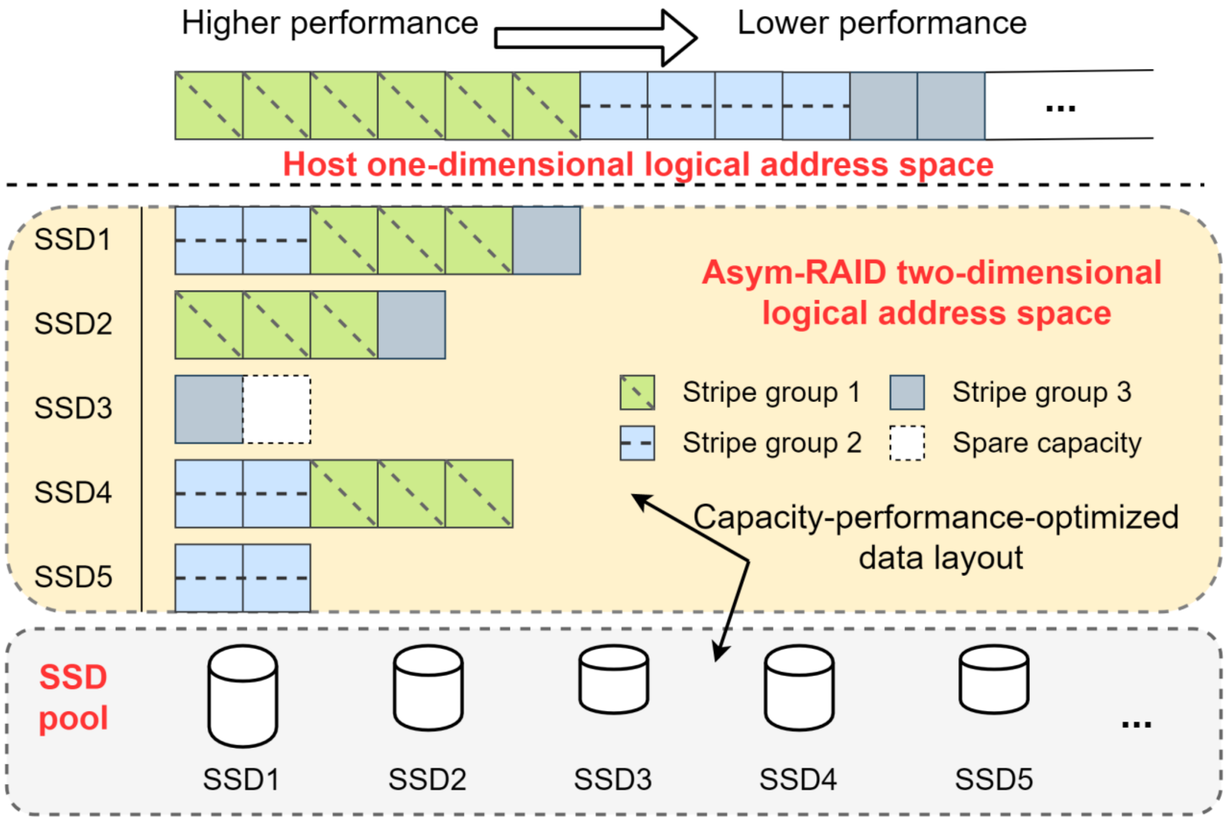
- Goal
  - Optimize system performance and storage utilization by leveraging a mix of heterogeneous devices

- High-level idea
  - Asymmetrically distribute data across the disk array

- Approach
  - Capacity → heterogeneity-aware data distribution
  - Performance → performance-optimized data placement
  - L2P addressing → mapping table/learned models

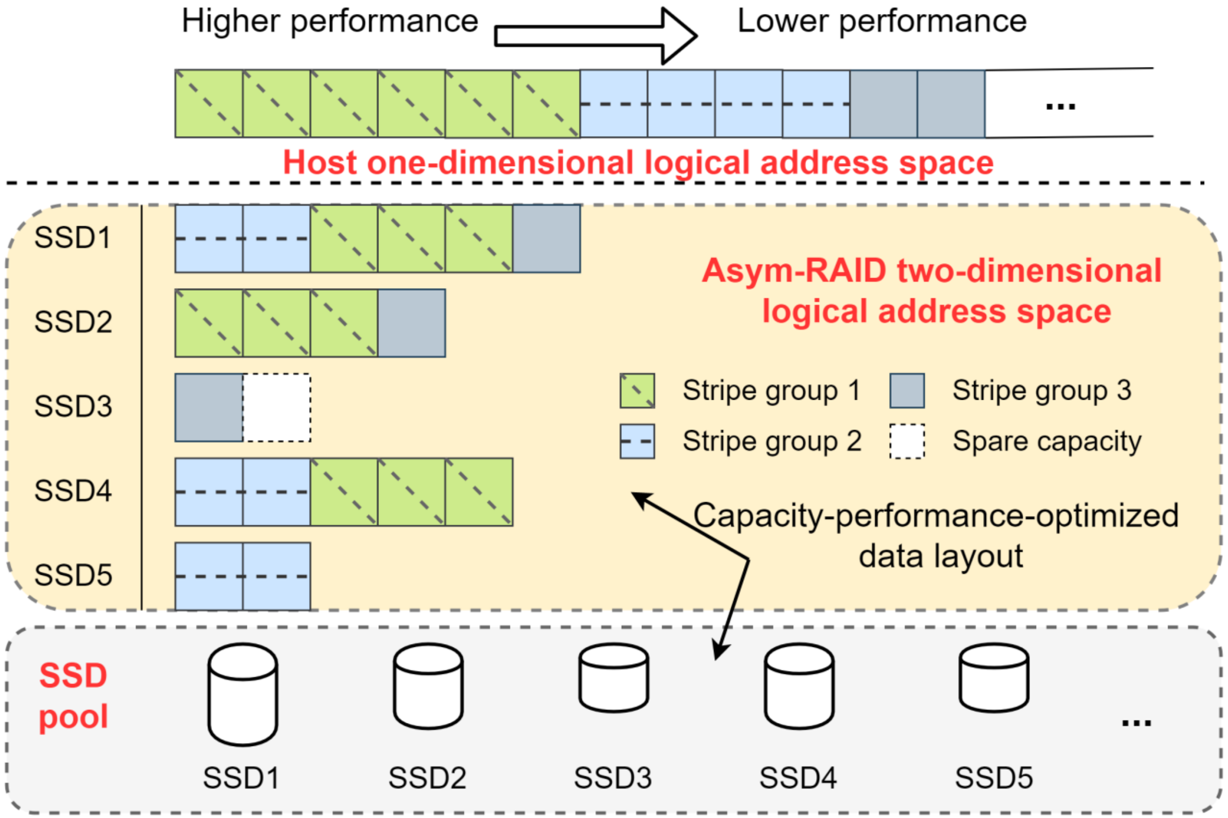
Asymmetric RAID

- A simple (2+1) RAID-5 configuration
  - 2 data chunks and 1 parity chunk from a 5-disk array



Asymmetric RAID

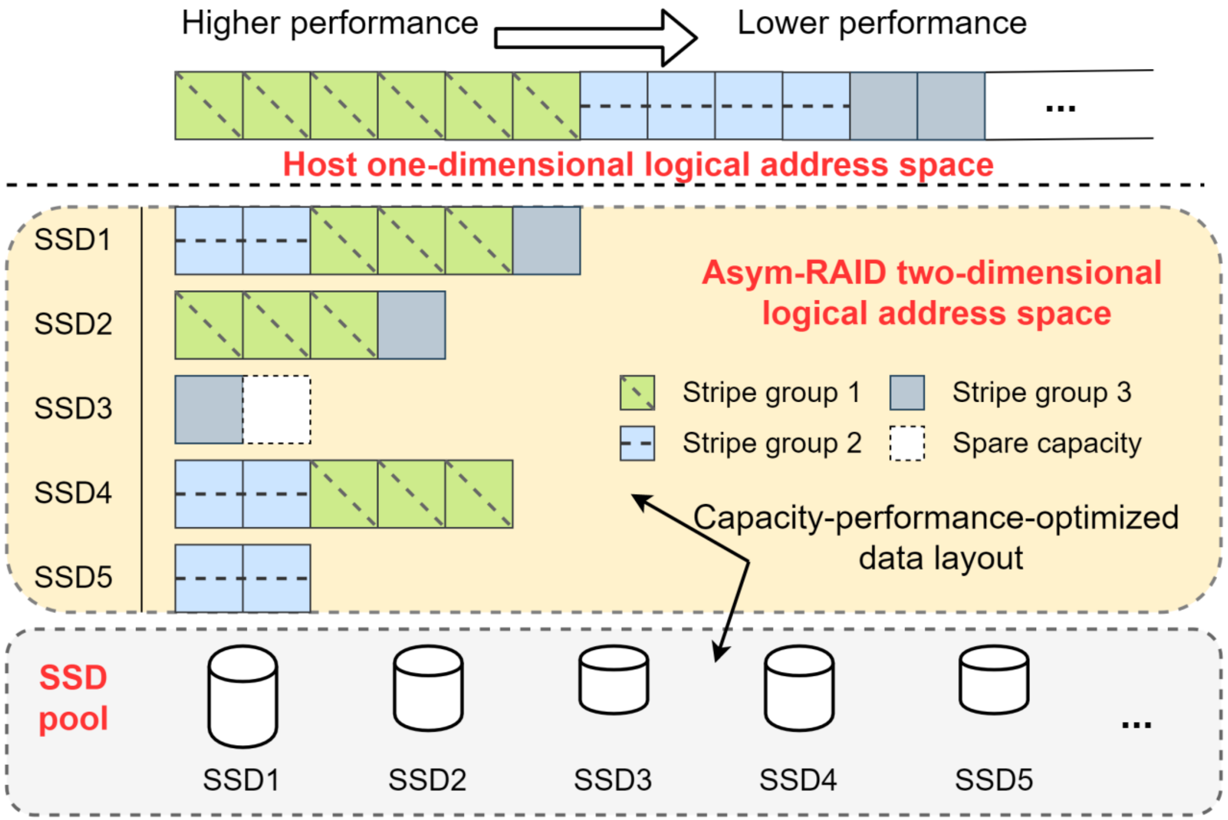
- A simple (2+1) RAID-5 configuration
  - 2 data chunks and 1 parity chunk from a 5-disk array



Challenge 1:
Maximize aggregate logical capacity for devices with different capacity

Asymmetric RAID

- A simple (2+1) RAID-5 configuration
  - 2 data chunks and 1 parity chunk from a 5-disk array

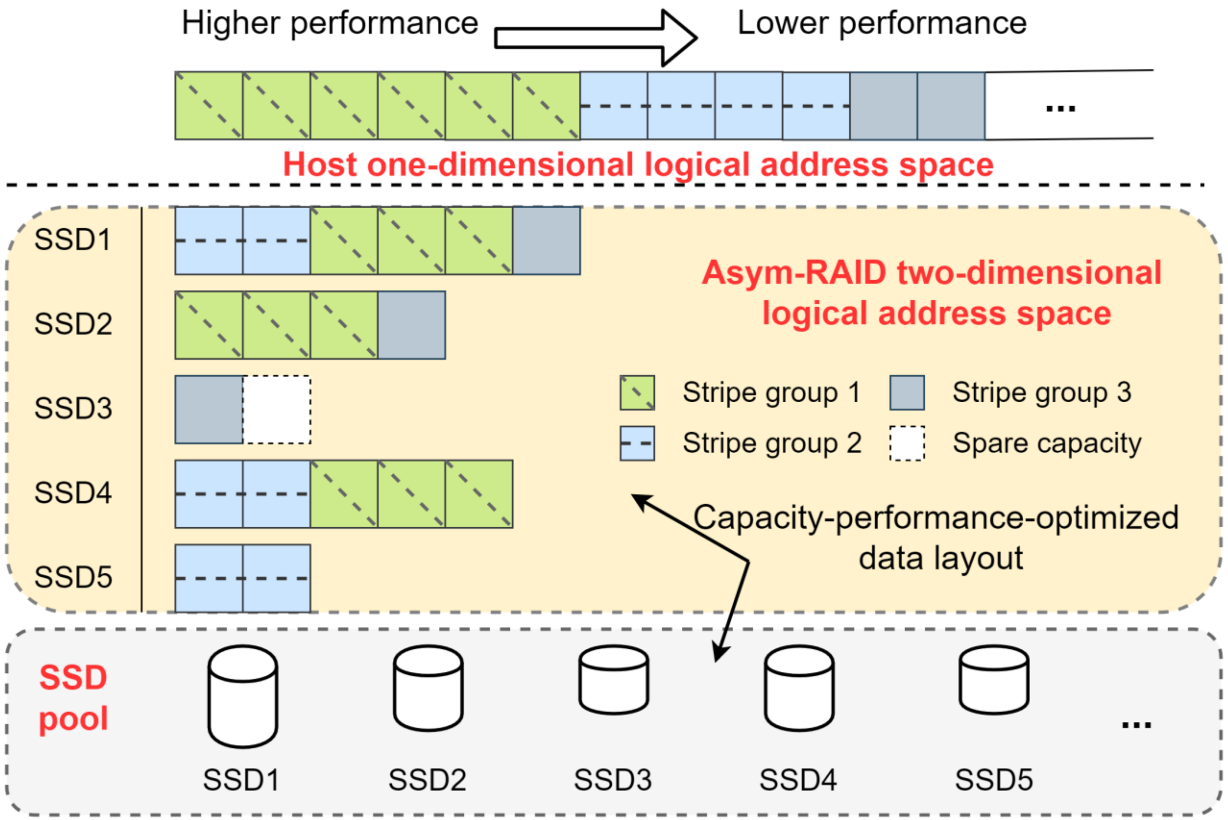


Challenge 1:
Maximize aggregate logical capacity for devices with different capacity

Challenge 2:
Optimize the usage of higher performance devices

Asymmetric RAID

- A simple (2+1) RAID-5 configuration
  - 2 data chunks and 1 parity chunk from a 5-disk array



Challenge 1:
Maximize aggregate logical capacity for devices with different capacity

Challenge 2:
Optimize the usage of higher performance devices

Challenge 3:
Achieve efficient address translation between user, AFA, and devices

Heterogeneity-aware data distribution

- Maximize the available logical capacity exported to the host

- Mathematical modeling
  - Parameters: disk pool size $\boldsymbol{N}$, disk sizes $\boldsymbol{S_i}$ $(1 \leq i \leq N)$, data stripe width $\boldsymbol{k}$ (k < N), and chunk size $\boldsymbol{C}$.
  - Binary decision variable $\boldsymbol{x_{ijk}}$: representing whether chunk k of data stripe j is assigned to disk i.
  - Objective function $\boldsymbol{D}$: maximize the number of complete k-width data stripes.

Heterogeneity-aware data distribution

- Maximize the available logical capacity exported to the host

- Mathematical modeling
  - Parameters: disk pool size $\boldsymbol{N}$, disk sizes $\boldsymbol{S_i}$ $(1 \leq i \leq N)$, data stripe width $\boldsymbol{k}$ (k < N), and chunk size $\boldsymbol{C}$.
  - Binary decision variable $\boldsymbol{x_{ijk}}$: representing whether chunk k of data stripe j is assigned to disk i.
  - Objective function $\boldsymbol{D}$: maximize the number of complete k-width data stripes.

- Constraints inherited from the RAID
  - Each chunk in a data stripe is assigned to exactly one disk
  - No two chunks in a stripe are on the same disk
  - Disk capacity limits the number of chunks it can hold

Heterogeneity-aware data distribution

- Maximize the available logical capacity exported to the host

- Mathematical modeling
  - Parameters: disk pool size $\boldsymbol{N}$, disk sizes $\boldsymbol{S_i}$ $(1 \leq i \leq N)$, data stripe width $\boldsymbol{k}$ (k < N), and chunk size $\boldsymbol{C}$.
  - Binary decision variable $\boldsymbol{x_{ijk}}$: representing whether chunk k of data stripe j is assigned to disk i.
  - Objective function $\boldsymbol{D}$: maximize the number of complete k-width data stripes.

- Constraints inherited from the RAID
  - Each chunk in a data stripe is assigned to exactly one disk
  - No two chunks in a stripe are on the same disk
  - Disk capacity limits the number of chunks it can hold

$$\text{Maximize} \quad D$$

$$\sum_{i=1}^{N} x_{ijk} = 1, \quad \forall j, \forall k$$

$$x_{ijk} + x_{ijk'} \leq 1, \quad \forall i, \forall j, \forall k' \neq k$$

$$\sum_{j=1}^{D} \sum_{k=1}^{k} C \cdot x_{ijk} \leq S_i, \quad \forall i$$

Solved by integer linear programming

Performance-optimized data placement

- Build a performance-aware logical volume
  - Imbue performance info into logical blocks

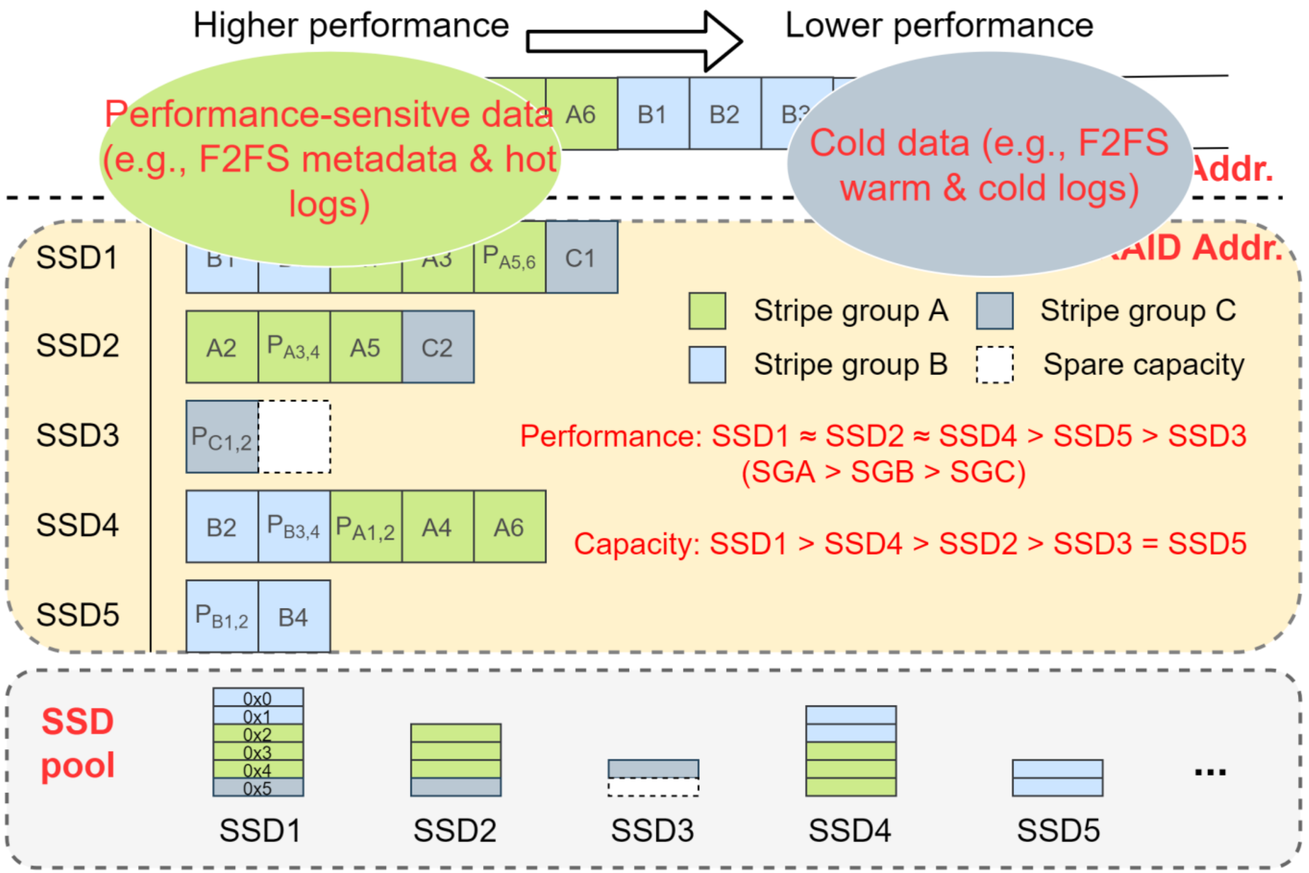
Performance-optimized data placement

- Build a performance-aware logical volume
  - Imbue performance info into logical blocks

Allow the system to differentially use logical blocks with low overhead

Performance-optimized data placement

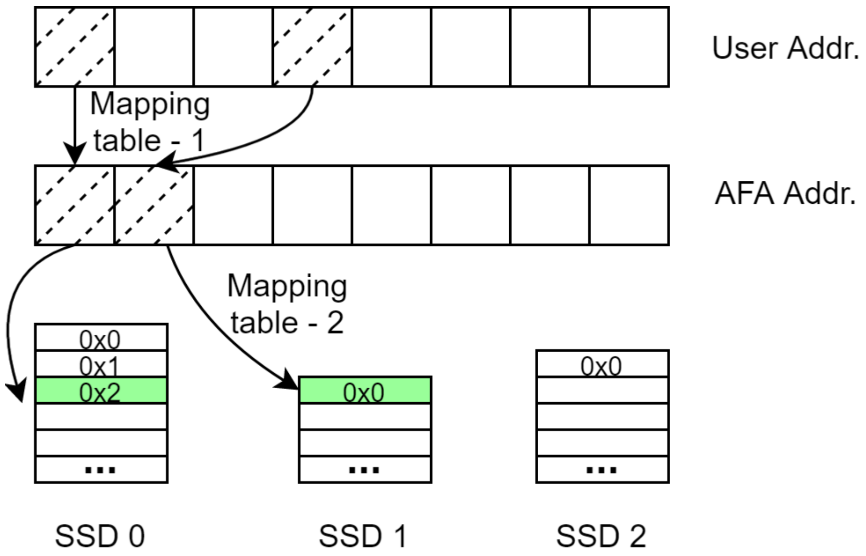
- Build a performance-aware logical volume
  - Imbue performance info into logical blocks



Allow the system to differentially use logical blocks with low overhead

L2P addressing

- Asym-RAID requires a logical-to-physical mapping for each stripe group
  - 25 bytes for each entry



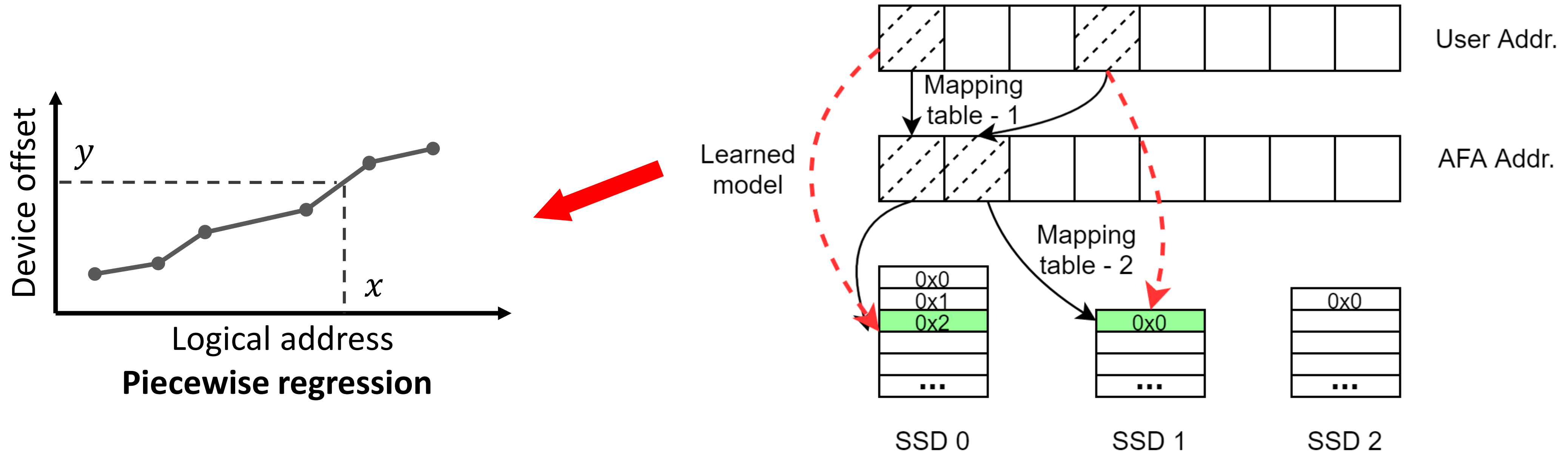
1. User LBA
2. Group ID & offset
3. Stripe ID (group)
4. Disk ID & LBA

	Start LBA	Length	Disk components	Disk offsets
Stripe_group 1	0	6	(1, 2, 4)	(2, 0, 2)
Stripe_group 2	6	4	(1, 4, 5)	(0, 0, 0)
Stripe_group 3	10	2	(1, 2, 3)	(5, 3, 0)

One-to-one mapping table:
~0.1% space overhead worst case

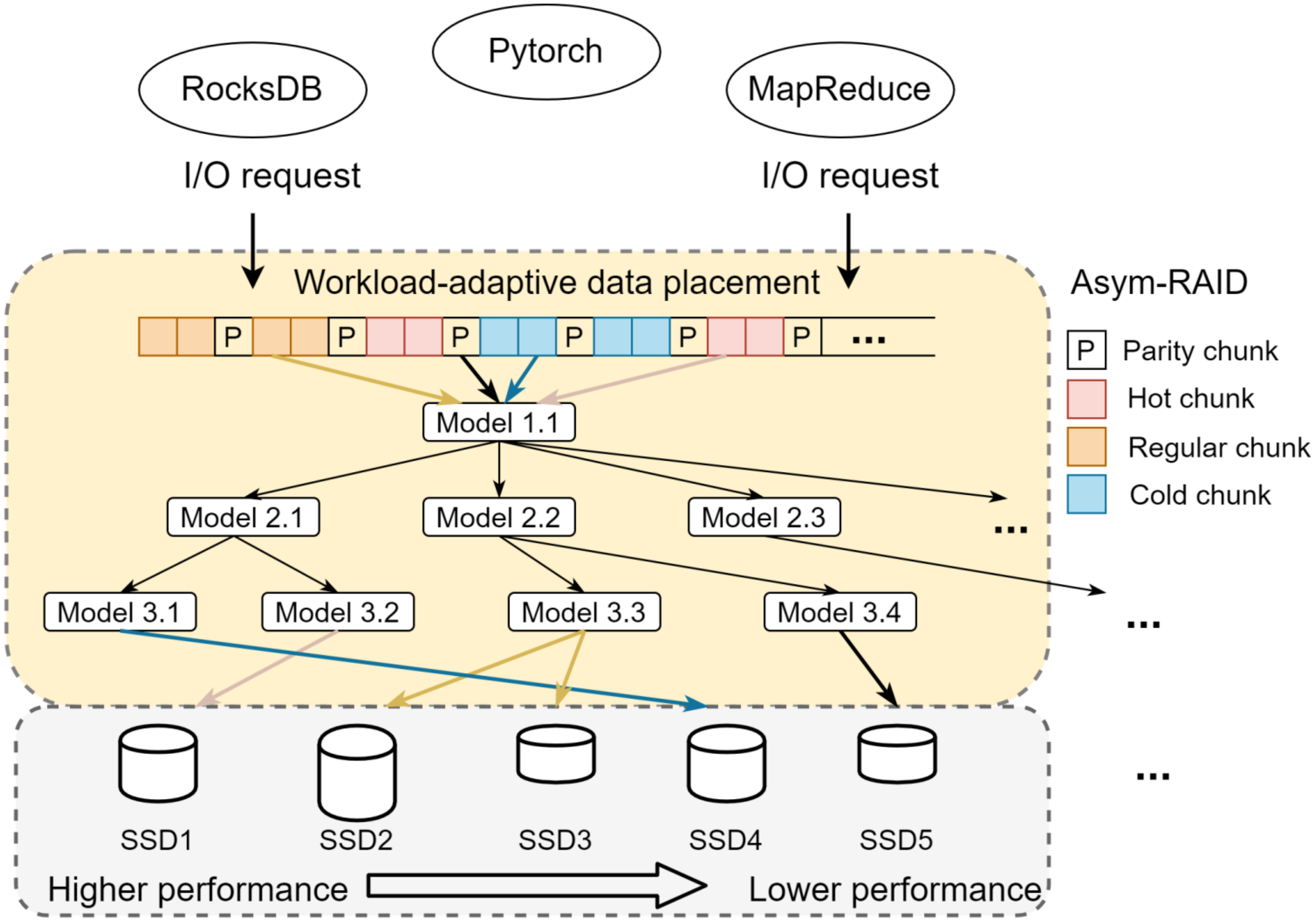
Learned models for addressing

- ϵ-bounded piecewise linear model
  - $\mathcal{M} = sl, ic, LPA_{Start}$, $y = sl \cdot x + ic$
  - ϵ-bound: $|y_{pred} - y_{real}| < \epsilon$

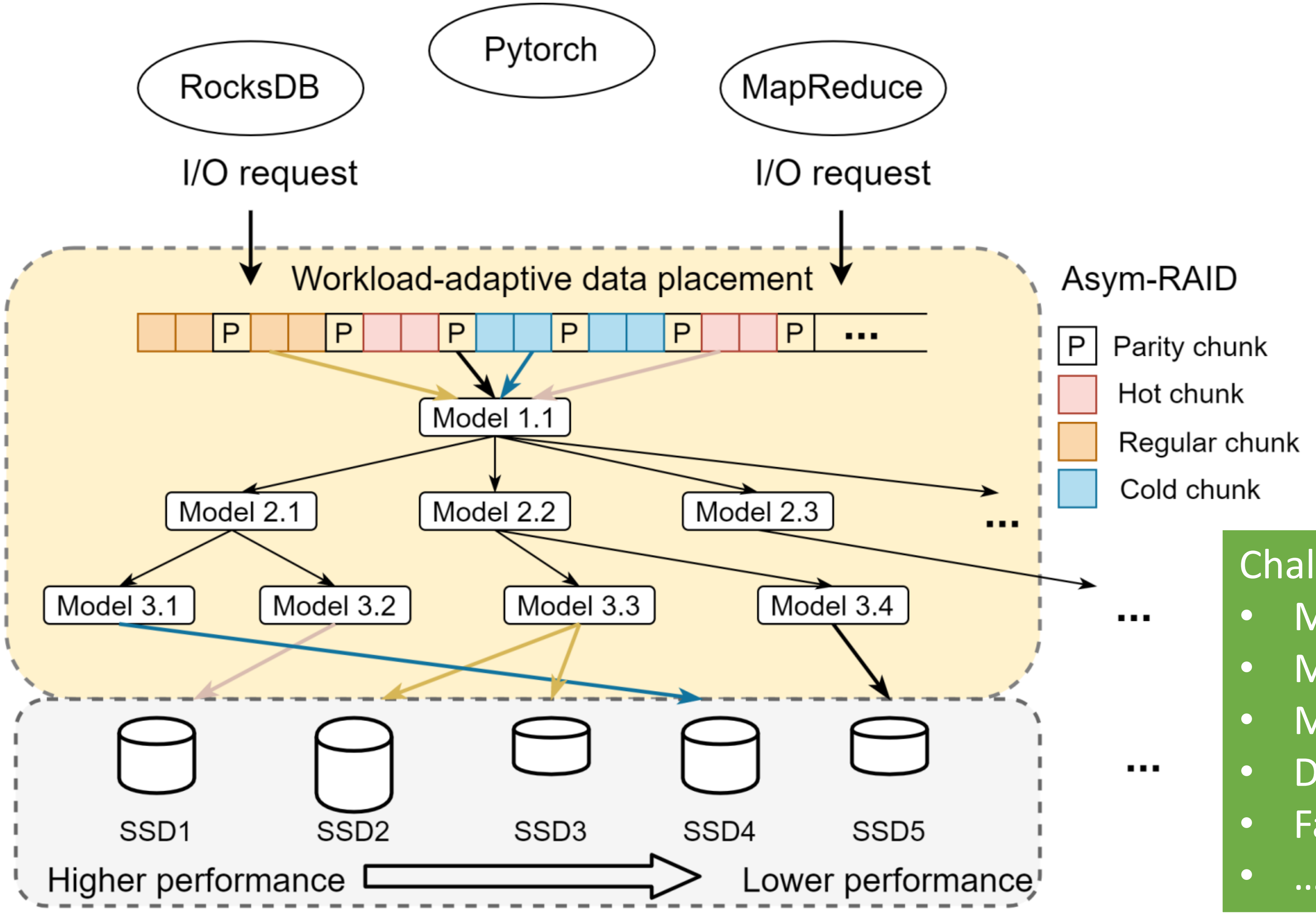


Piecewise regression

Workload-adaptive data placement



Workload-adaptive data placement



Challenges:

- Misprediction
- Metadata persistence
- Model retraining
- Data redistribution
- Fault tolerance
- …

Conclusion

- Existing AFA solutions lead to significant disk underutilization when considering device heterogeneity

- Asym-RAID asymmetrically distributes data across the array to fully utilize the capacity of each SSD
  - Capacity → determine data layout through mathematical modeling
  - Performance → imbue performance info into logical blocks

- Ongoing work
  - Adaptive data layout for dynamic disk heterogeneity
  - Learned index models for addressing
  - RAID over disaggregated storage

Conclusion

- Existing AFA solutions lead to significant disk underutilization when considering device heterogeneity
- Asym-RAID asymmetrically distributes data across the array to fully utilize the capacity of each SSD
  - Capacity → determine data layout through mathematical modeling
  - Performance → imbue performance info into logical blocks
- Ongoing work
  - Adaptive data layout for dynamic disk heterogeneity
  - Learned index models for addressing
  - RAID over disaggregated storage

Thank you!
Q&A
Contact: zjiao04@syr.edu