SNIA DEVELOPER CONFERENCE
SDC 25
By Developers FOR Developers

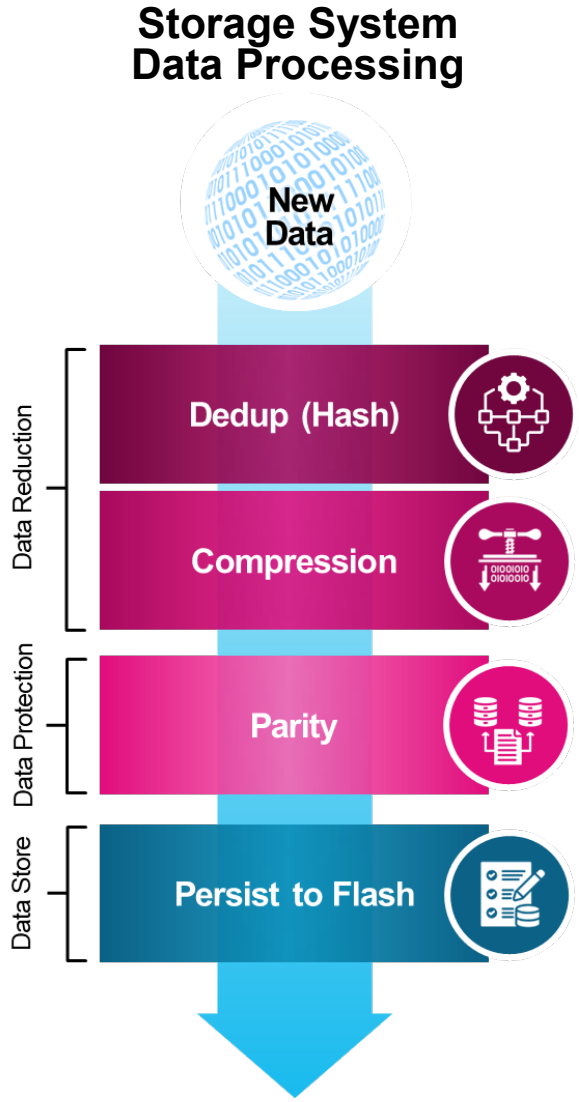
Hyatt Regency Santa Clara, CA
September 15-17, 2025

Offload Fixed Function Storage Services to Storage Subsystem

Mahinder Saluja,
Director of SSD Technology and Pathfinding
KIOXIA America, Inc.
September 2025

Storage Systems Data Management Scalability Cha

Slide was previ... approved this year for FMS '25



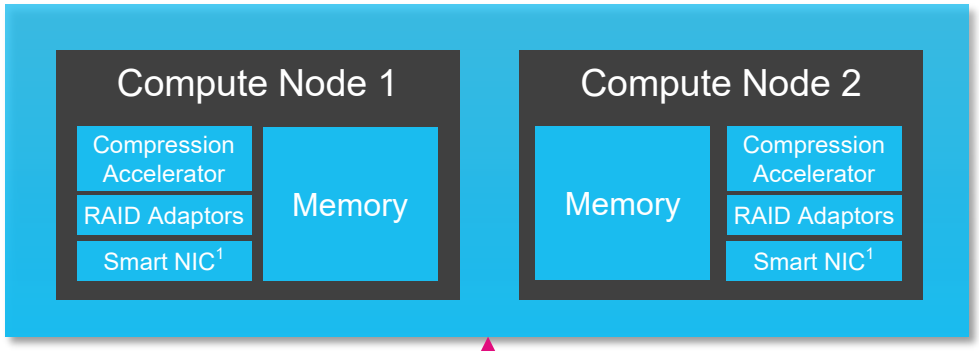
- **Data ingestion in a storage system involves a series of compute intensive operations**
- **With increased SSD performance, data management tasks require increased processing power usually solved by**
  - *General purpose CPU cores*
    - Current compute architecture and memory hierarchy increase total cost of acquisition and total cost of ownership
    - Overprovisioning cores and memory
  - *Accelerators*
    - Data processing units
    - Performance limited by number of PCIe® lanes assigned or system memory
    - Cannot scale with added SSDs
    - Consumes additional power from system slots

1. Deduplication (dedupe) hashes are unique identifiers generated from data blocks using hashing algorithms to identify duplicate data within a system. All images and/or graphics within this slide are the property of KIOXIA America, Inc. (KAI) and are reproduced with the permission of KAI. PCIe is a registered trademark of PCI-SIG

Storage Systems Data Management Compute Reso

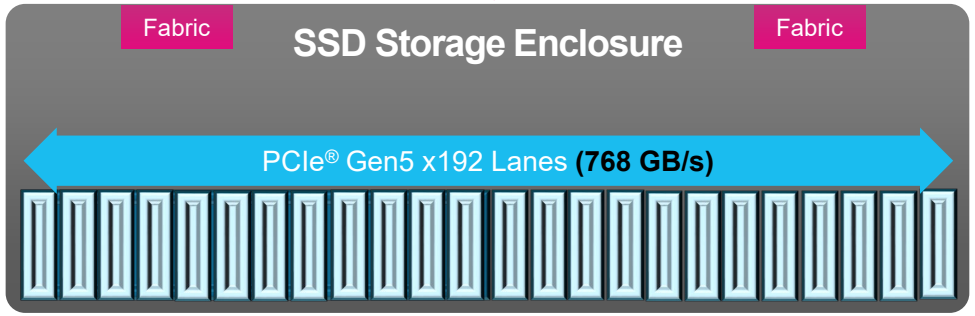
Slide was previ... approved this year for FMS '25

Compute for Storage Functions



Fabric
- Performance configuration to match SSD perf
- Capacity Configuration

Resources
- Expensive Compute Node
- Wasted PCIe bandwidth



48 NVMe™ SSDs
(each SSD with Gen5 x4) enclosure

Compute Node Requirements

- For Performance, will require extremely high memory bandwidth and compute resources, accelerators to augment compute
- For Capacity, Still require significant compute resources and unutilized bandwidth in enclosures

Storage Enclosure 48 NVMe SSDs

- **Available Resources for Offload**
  - Ingress(Write) vs egress(Read) SSD bandwidth
  - Useable interface bandwidth (assuming 32 PCIe lanes): 32x4 = **128 GB/s**
  - PCIe bandwidth (Gen5) (48 Gen5 x4 Lanes) : **48x**4x4 = **768 GB/s**
    (SSDs*Lanes*per lane bandwidth)
  - **Spare PCIe bandwidth: 768-128 = 680 GB/s**

Host Orchestrated Compute Offload Building

Slide was previously approved this year for FMS '25

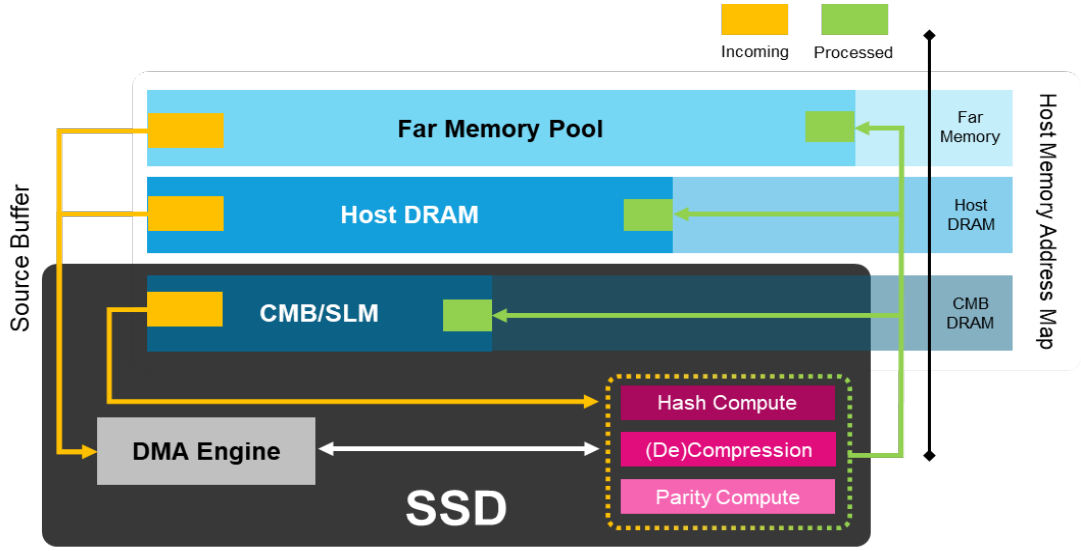
Power-efficient compute engines

DRAM bandwidth saving

Host orchestrated standard based

Applications of Offload Engines

- **Hash/CRC[3]:** Dedupe, Object/File signature/scrubbing, buffer integrity
- **(De)Compression:** Compression with levels, decompress and filter
- **Parity Compute**: Erasure code (EC), compare, **Data scrubbing, RAID Rebuild**



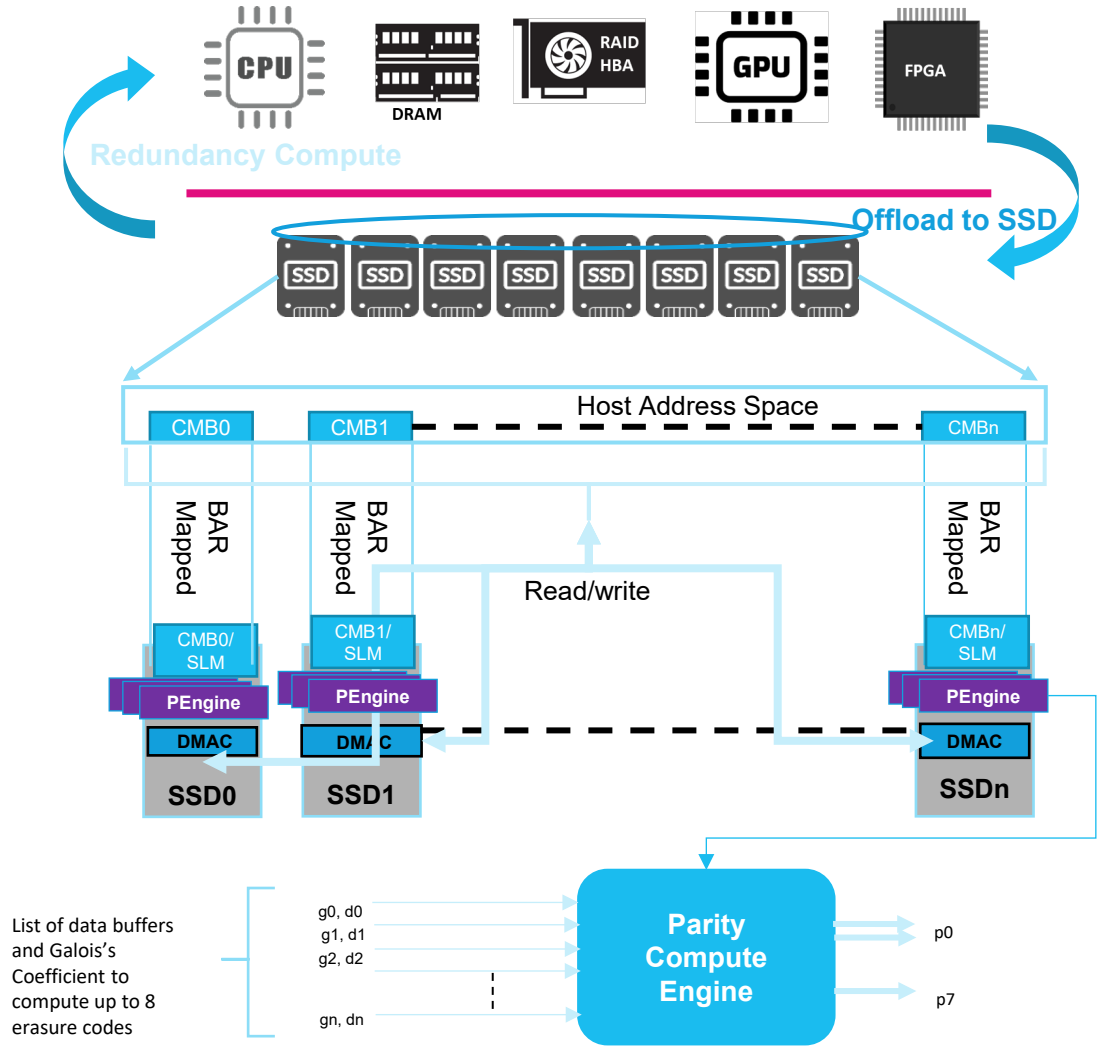
Host places data in appropriate buffer (host memory/ far memory/ CMB[1]) → Host issues command to the SSD → Compute engine on the SSD executes the command → Processed data is placed into the destination buffer → Command completion returns key attributes

1. Controller memory buffer (CMB). 2. Subsystem local memory (SLM). 3. Cyclic redundancy check (CRC). All images and/or graphics within this slide are the property of KIOXIA America, Inc. (KAI) and are reproduced with the permission of KAI.

RAID Offload

RAID Offload

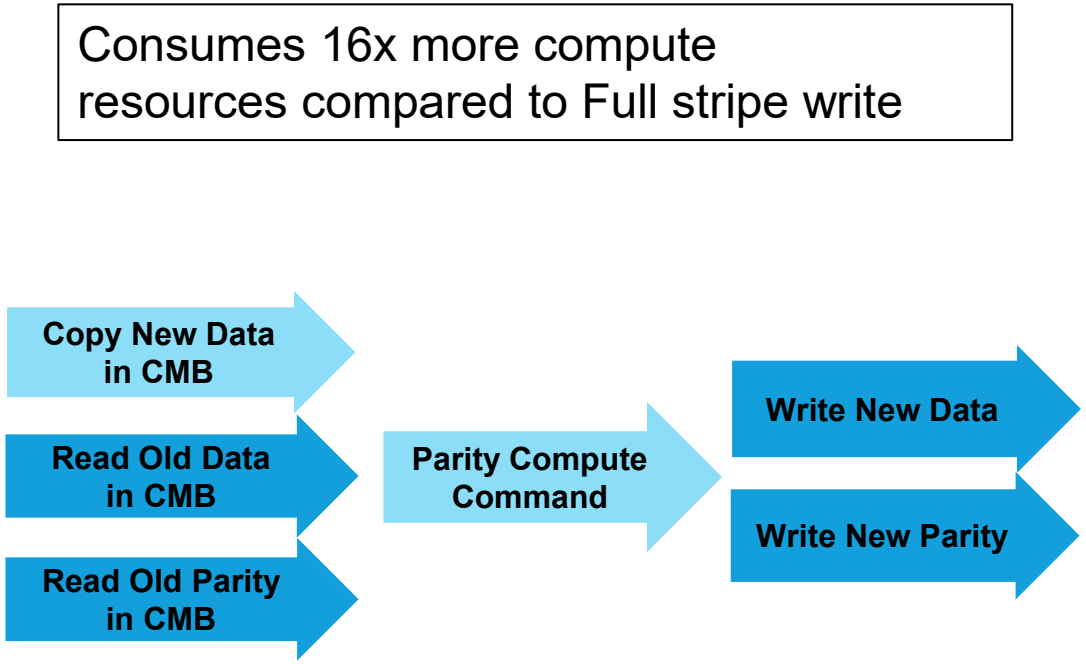
- Host orchestrated parity/EC(Erasure Coding) compute and memory bandwidth offload to SSD
  - Leverage NVMe™ subsystem attached memory such as CMB/SLM for memory bandwidth offload
  - Parallel parity compute function on SSD
  - DMA (direct memory access) engine for buffer-to-buffer copy for mapped addresses
  - Commands to control the functions
  - RAID applications use commands to offload and continue to handle faults



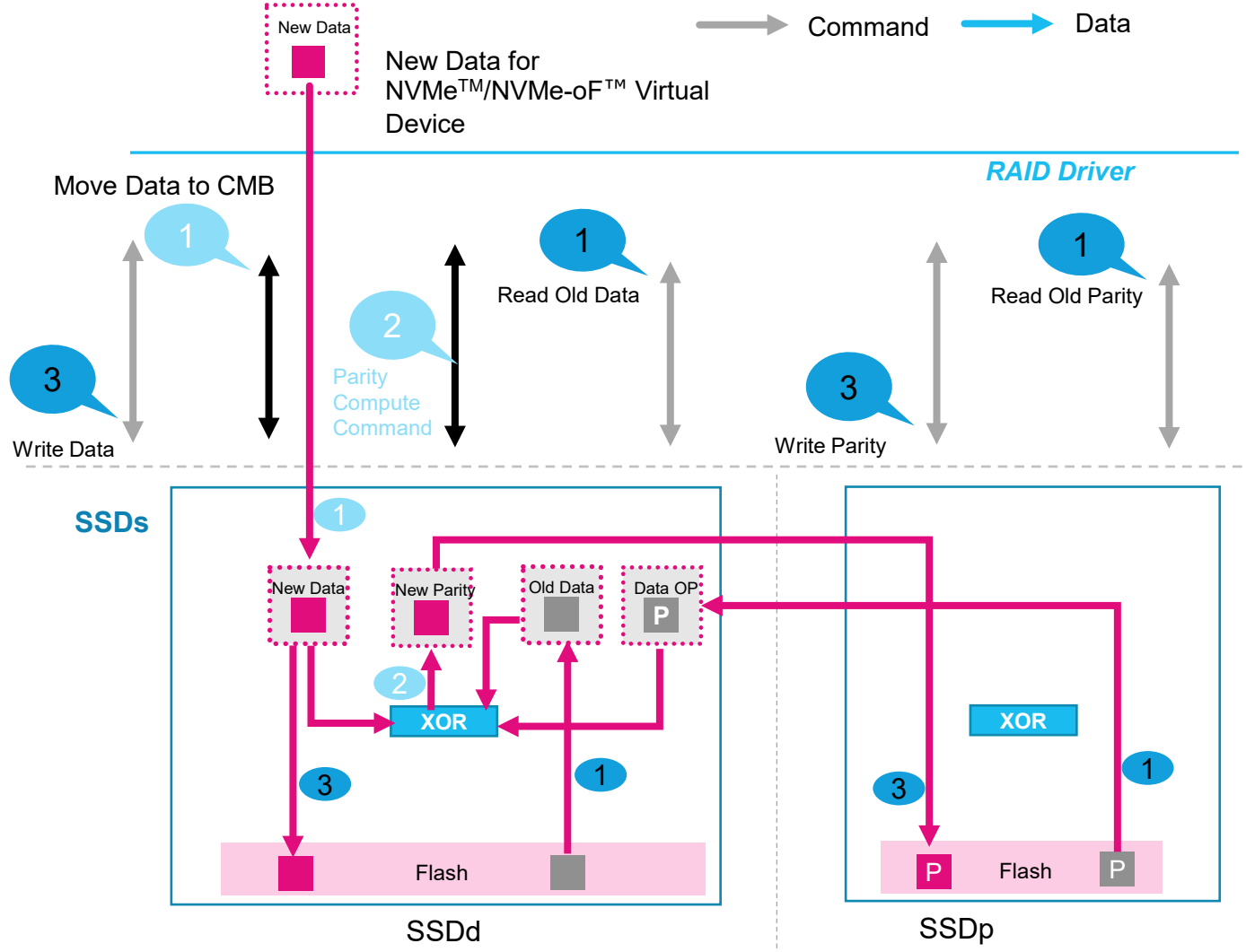
Command and Data Flow : RAID Partial Stripe

Slide was previ... approved last year for SDC '24

Consumes 16x more compute resources compared to Full stripe write

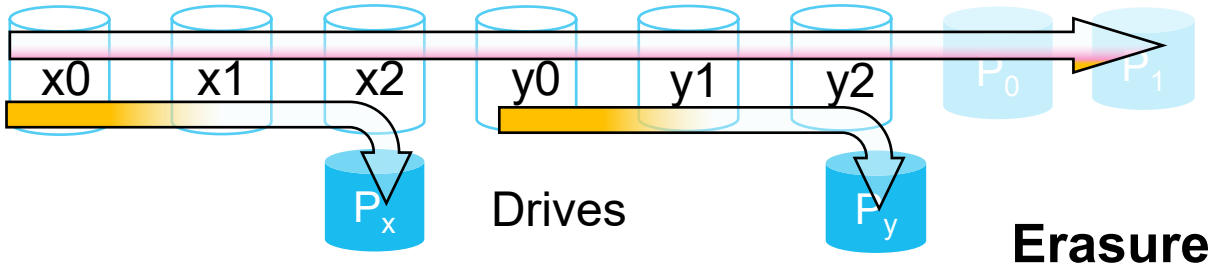


RAID 5 – Read / Modify / Write Sequence



Parallel XOR: Example Build 4 EC in One Com

Slide was previ... approved last year for SDC '24



Erasure Code command for **4 parity** compute

Parity P_x (XOR)	
Src buf	x0, x1, x2
Galois coefficient	1,1,1,1
Output buffer address	Px
Operation type	XOR

Parity P_y (XOR)	
Src buf	y0,y1,y2
Galois generator	1,1,1,1
Output buffer address	Py
Operation type	XOR

Parity P_0 (Weighted XOR)	
Src buf	x0, x1, x2,y0,y1,y2
Galois coeffficient	$\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1, \beta_2$
Output buffer address	P_0
Operation type	XOR

Parity P_1 (Weighted XOR)	
Src buf	x0, x1, x2,y0,y1,y2
Galois coefficient	$\alpha_0^2, \alpha_1^2, \alpha_2^2, \beta_0^2, \beta_1^2, \beta_2^2$
Output buffer address	P_1
Operation type	XOR

Compression and Hash Offload

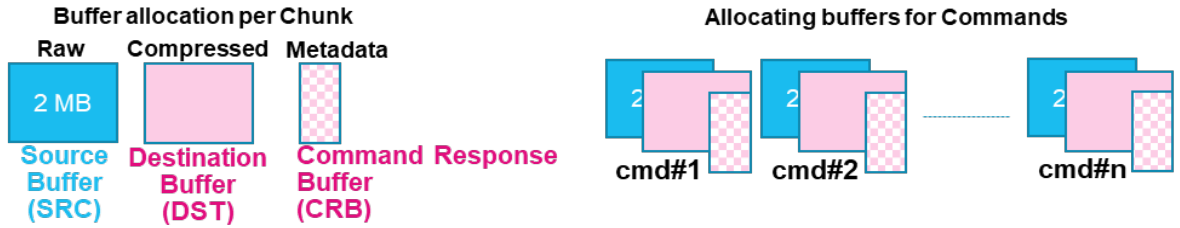
Multiple Block Compression/Hash Offload

1 A stream of uncompressed data (multiple blocks of varying sizes) is ingested into the storage pipeline

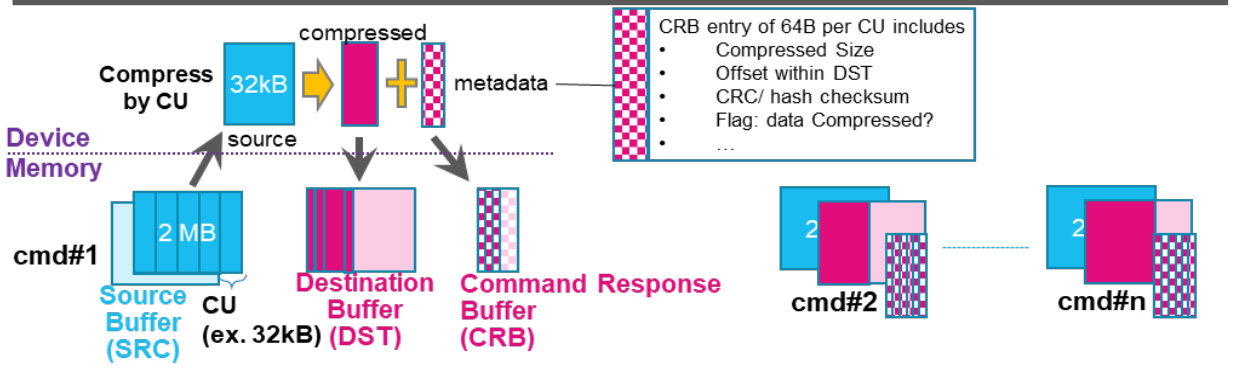
Data

2 Host allocates working area for each Command (SRC, DST, and CRB), and issues NVMe™ Compression Command

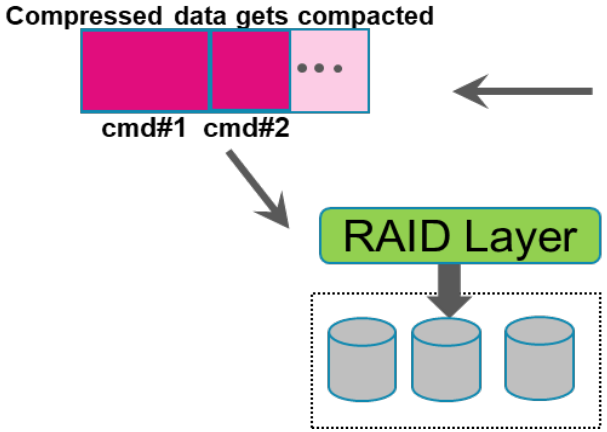
*DST size = SRC size; CRB = Num of CU Blocks * 64B*



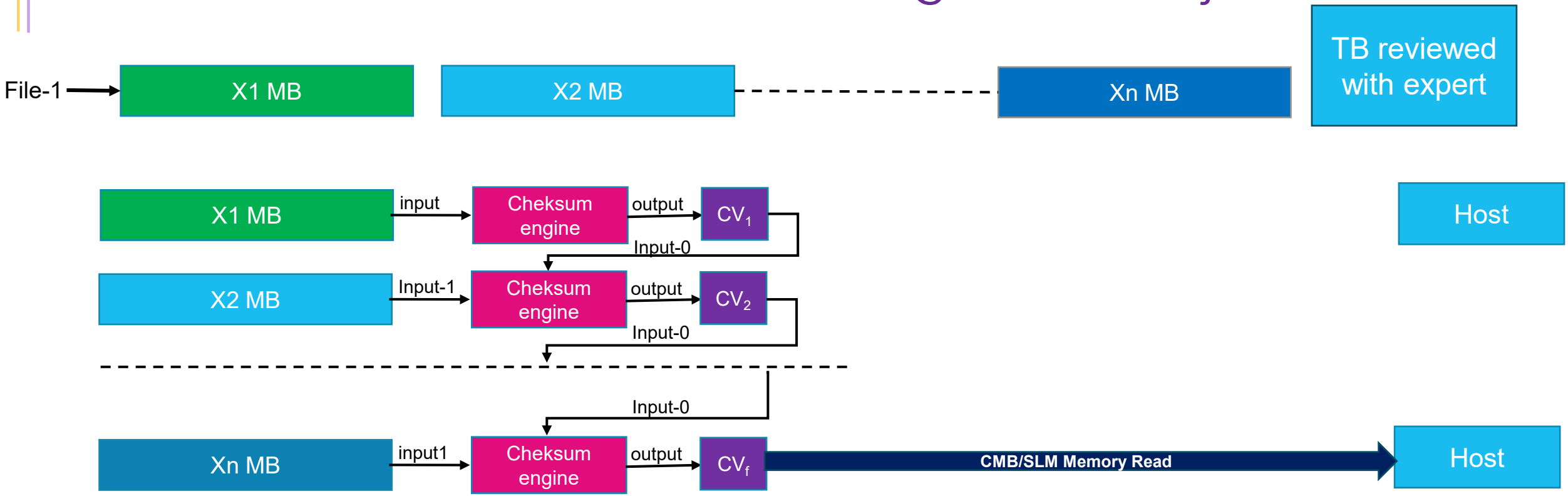
3 SSD
1) Chops incoming data into Compression Unit (CU) chunks
2) Compresses each CU individually
3) CRB entry for each CU is populated in CRB (compressed size, data placement, CRC, etc.)



4 Host has the data and metadata (compressed size, location) of compressed data. Next, it can compact, compute parity and persist data on the storage.



Checksum Calculation for Small/Large Files/Objects/Blocks



Example shows checksum (SHA256, CRC32 or CRC16) calculation of file object

1. Source data is not physically contiguous, and they are not of same size.
2. Size shown in MB is for illustration purpose only. It can be of any size (DWs, KBs, MBs ..GBs)
3. In checksum calculation, total object size should be known
4. Content of CV can differ as per algorithm used in checksum computation

Data Ingestion Pipeline with Offload

Slide was previ… approved this year for FMS '25

Storage Controller

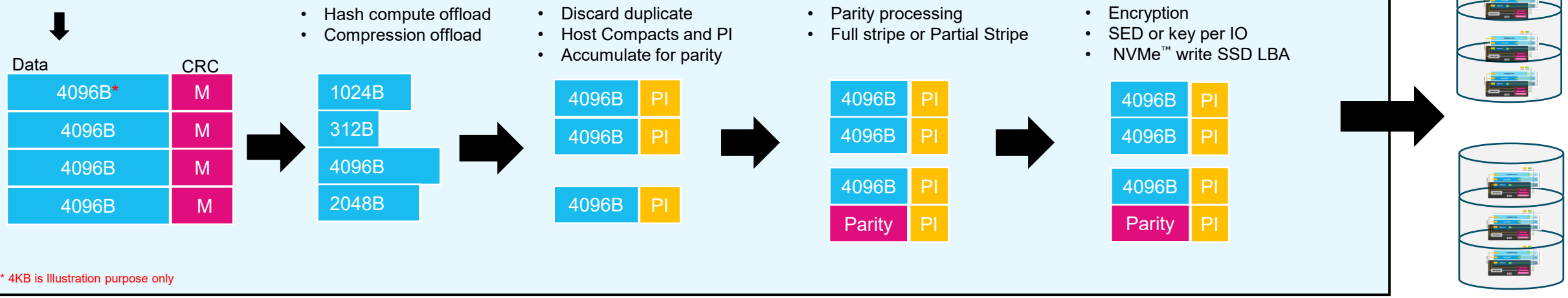
Step1: Host places data in appropriate buffer

Step2: Data Reduction and Compaction

Step3: Parity Compute Offload

Step4: Store Data to SSD

Host Memory or CMB[1] or SLM[2] Namespace



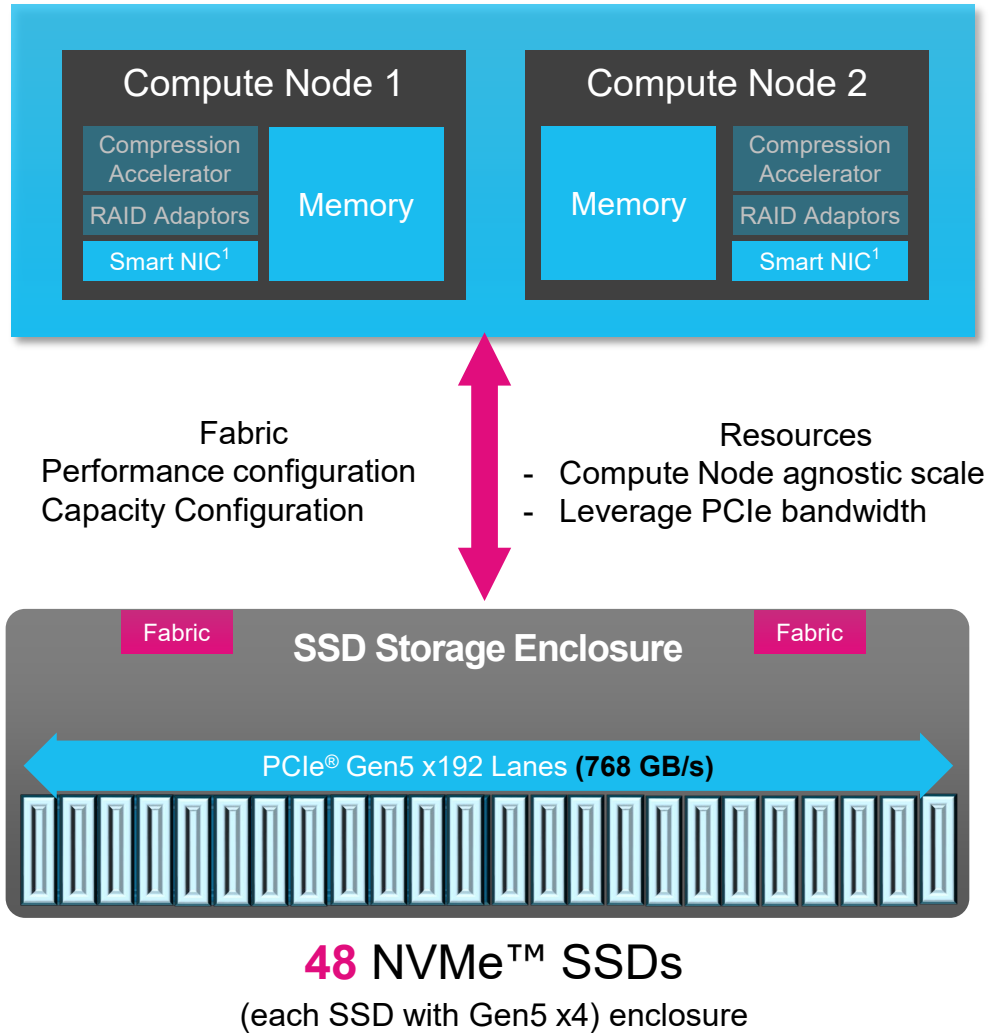
* 4KB is Illustration purpose only

 1. Controller Memory Buffer (CMB). 2. Subsystem Local Memory (SLM). Definition of capacity: KIOXIA Corporation defines a megabyte (MB) as 1,000,000 bytes, a gigabyte (GB) as 1,000,000,000 bytes, a terabyte (TB) as 1,000,000,000,000 bytes and a petabyte (PB) as 1,000,000,000,000,000 bytes. A computer operating system, however, reports storage capacity using powers of 2 for the definition of 1Gbit = 230 bits = 1,073,741,824 bits, 1GB = 230 bytes = 1,073,741,824 bytes, 1TB = 240 bytes = 1,099,511,627,776 bytes and 1PB = 240 bytes = 1,125,899,906,842,624 bytes and therefore shows less storage capacity. Available storage capacity (including examples of various media files) will vary based on file size, formatting, settings, software and operating system, and/or pre-installed software applications, or media content. Actual formatted capacity may vary.

Storage Systems Unexploited PCIe® Bandwid

Slide was previ… approved this year for FMS '25

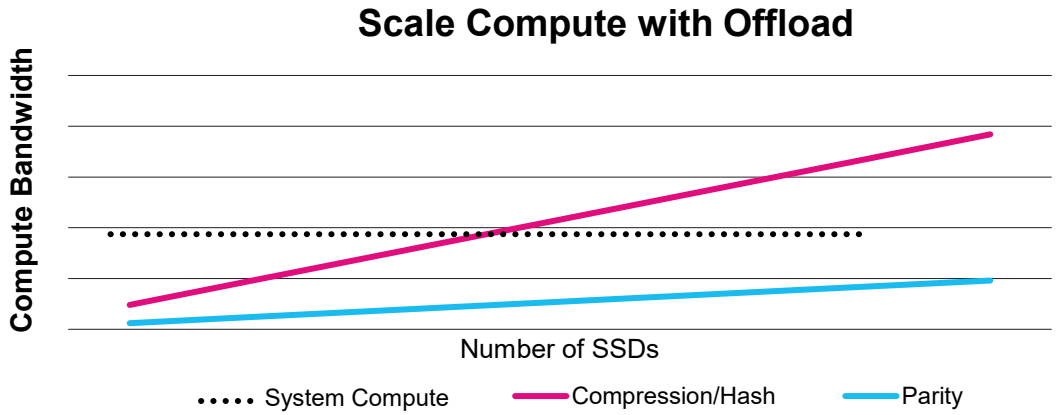
Compute for Storage Functions



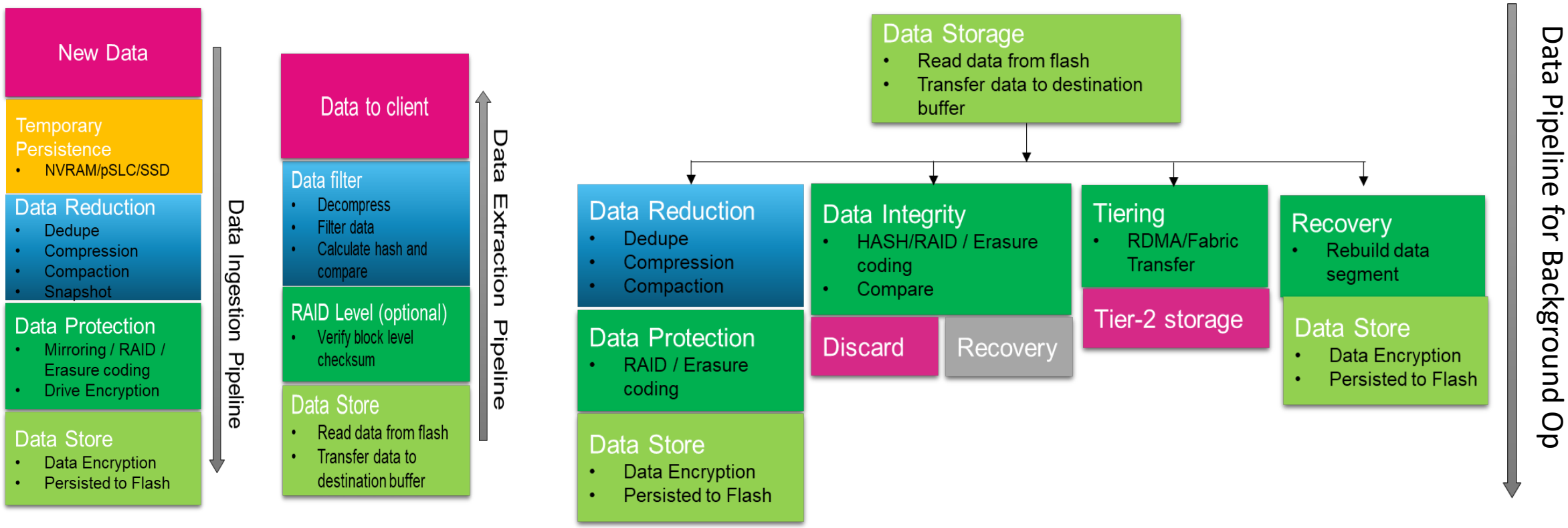
Compute Node

- Use SSD's accelerators and reduce memory footprint
- Scale up/out with every added SSD
- Leverage storage enclosure unused PCIe Bandwidth write path
- Cost effective and power efficient

- **With Offload Capable SSDs** (for illustration, not official specs.)
  - CMB/SLM memory bandwidth @ 10 GB/s/ SSD :480 GB/s
  - Parity compute bandwidth @ 2 GB/s/ SSD : 96 GB/s
  - (De)compression @ 10 GB/s :480 GB/s



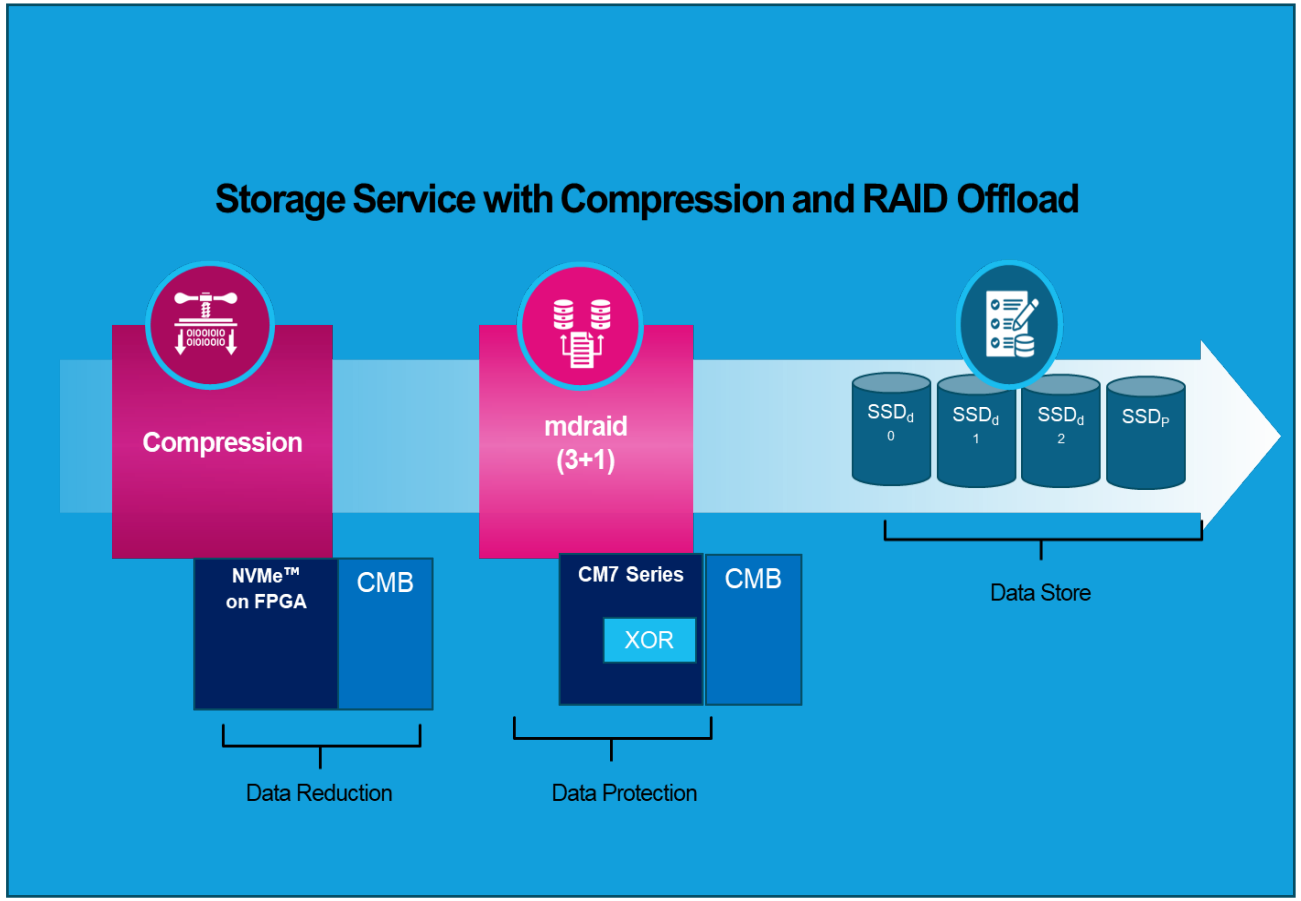
Storage System Data Management Flows



Storage Data Flows

1. Data ingestion/extraction pipeline offloaded to SSD
2. Background tasks can leverage offload without moving data out of storage systems

Data Pipeline Offload PoC Results



	No Offload	Offload	% Benefit
Write Bandwidth	~140 MB/s	~140 MB/s	-
Compression ratio	2.5x	2.5x	-
Compression (gzip) CPU core	200%	~1%	~199%
RAID CPU Resources	4%	4%	-
DRAM Bandwidth	~600 MB/s	**~160 MB/s**	3.8x

Results from RAID Offload Shared

Slide was previ… approved this year for FMS '25

RAID Offload: PoC Results (with KIOXIA CM7 Series SSDs and mdraid 5)

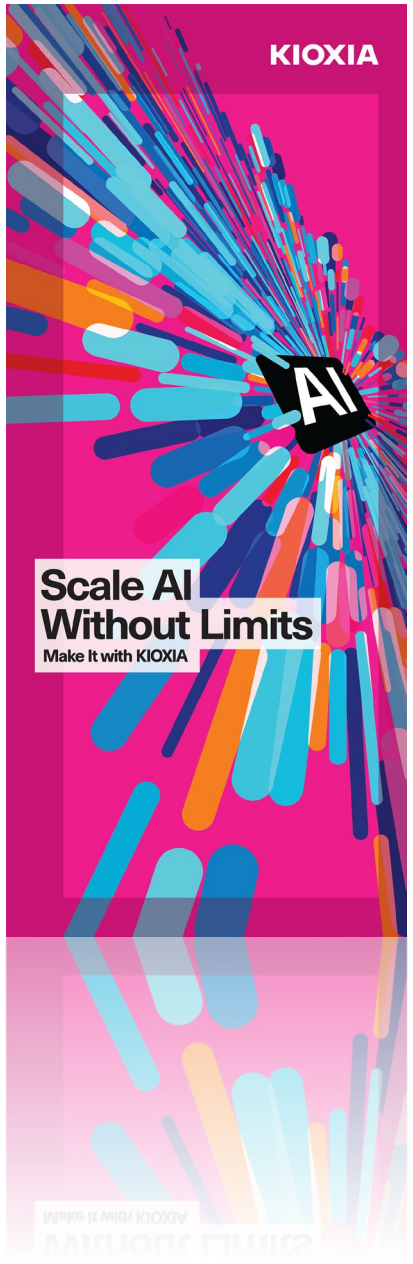
System	KIOXIA CM7 Gen4 x4 – mdRAID 5	RAID Offload	% Benefit
CPU Utilization	42	37	12% Reduction
DRAM Bandwidth (in MiB^2/s)	3450	340	91% Reduction

System: DELL® PowerEdge™ R650xs Intel Xeon® Gold 6338N 2.2GHz (2 Socket, 32 Cores) PCIe® 4.0 , SSDs: 5xCM7 Gen4 (1.92 TB)

I/O workload: FIO[1] 512K Random Write @ 950 MB/s

Data Scrubbing PoC Results

	Offload Disabled	Offload Enabled
Scrubbing Time	**129s**	**91s**
DRAM Bandwidth	**10.24 GB/s**	**1.43 GB/s**
Total CPU Utilization	**99.5%**	**~70%**
L3 Cache Misses	**14.7M**	**4M**
Total PCIe® Write (MB/s)	**3694 MB/s**	**159 MB/s**



Summary

Slide was previ... approved this year for FMS '25

- **Storage systems data services is series of compute functions and ready to offload**
- **The discarded PCIe® bandwidth can be leveraged efficiently**
- **Host managed standard fixed compute functions can be integrated to existing applications with nominal effort**
- **Scale out/up storage systems sustainably with every added SSDs**

Thank you for attending!

Please remember to rate this session. You get access to presentations at
http://sniadeveloper.org/conference