NVMe Over CXL

How CXL Lets Us Do

Controller Memory Buffers the Right Way

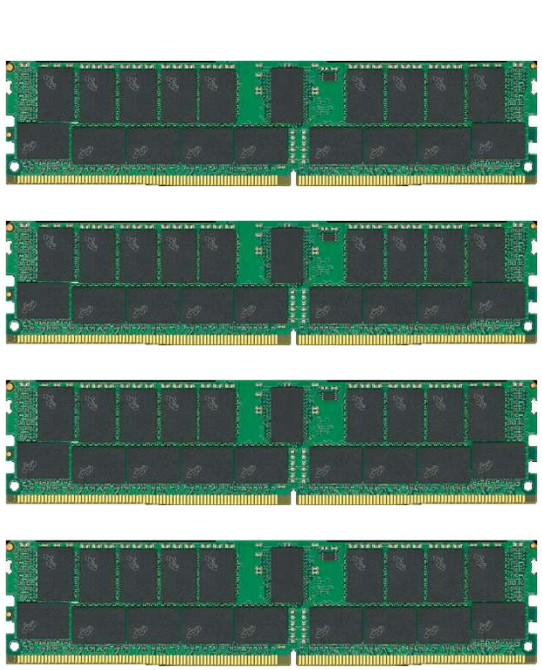
Bill Gervasi, Principal Systems Architect

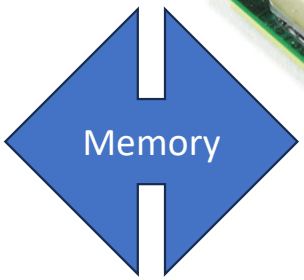
bilge@wolleytech.com

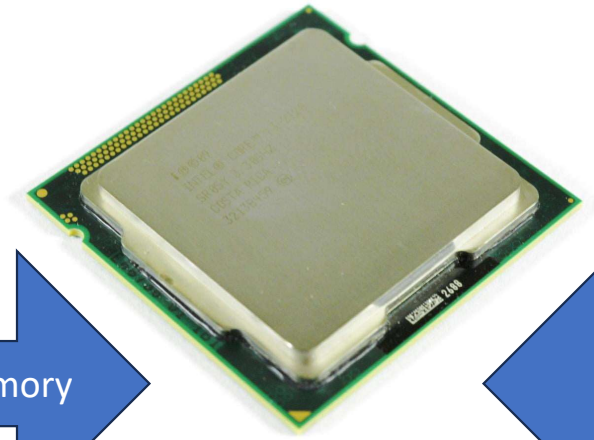
San Chang 張育銘, Director of Engineering

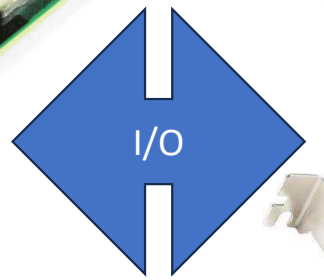
san@wolleytech.com

What millennium are we in?



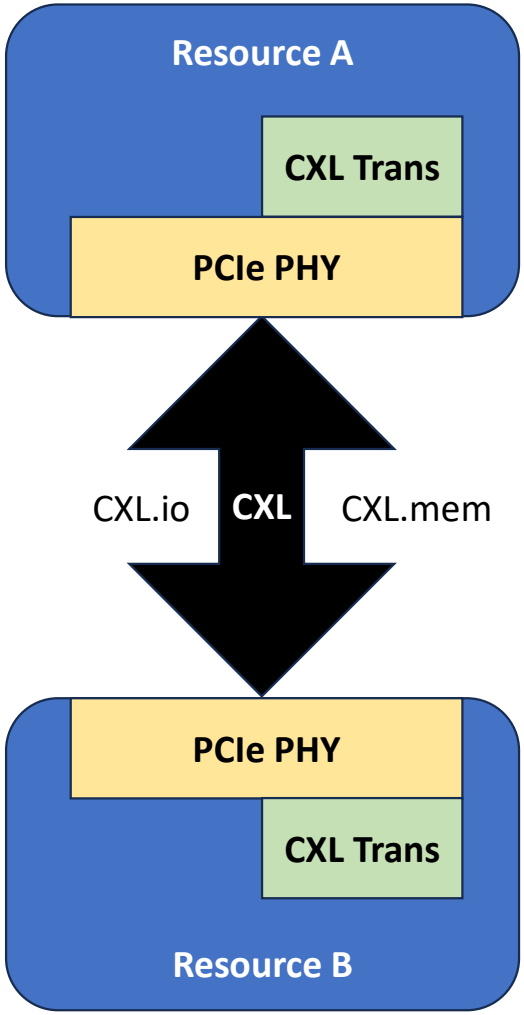






Memory and I/O have traditionally been separated

This has been our basic system architecture for thousands of years



CXL is more than just another I/O bus

CXL allows the blending of processing, memory, storage, and I/O over a consistent protocol

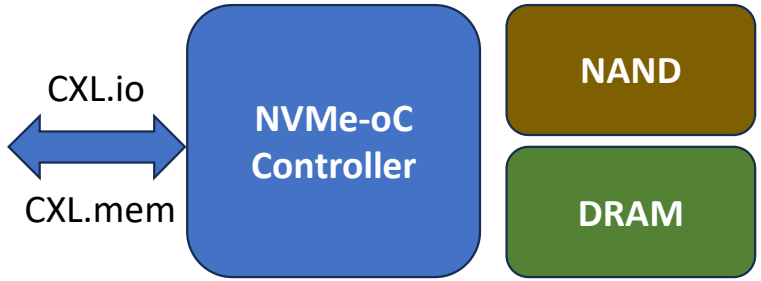
This enables virtualizing resources in interesting new ways

PCIe SSD

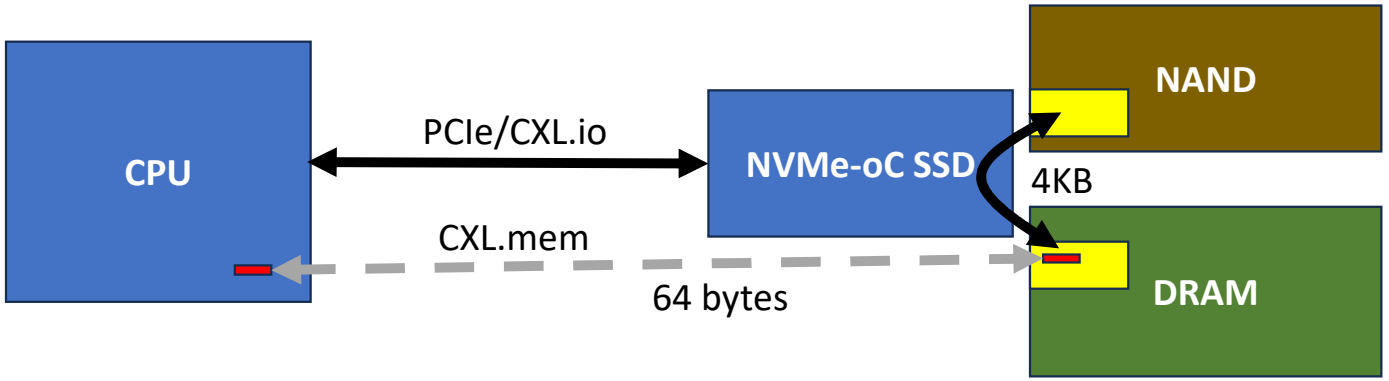
CXL Memory Module

NVMe-Over-CXL™ Merges memory and storage into a unified interface

NVMe-Over-CXL



The NVMe Over CXL Solution: Only grab the FLITs you need



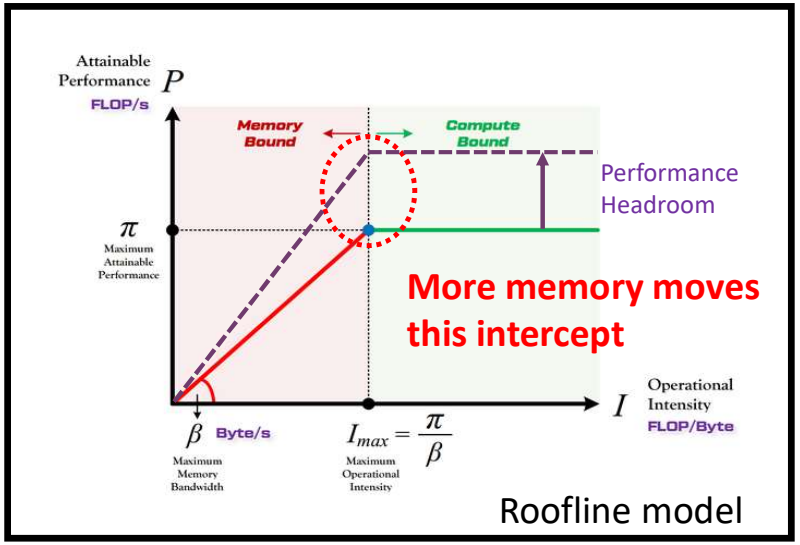
NVMe is just a cache protocol between NAND and DRAM

NVMe-oC places the controller memory buffer (CMB) in CXL space (HDM)

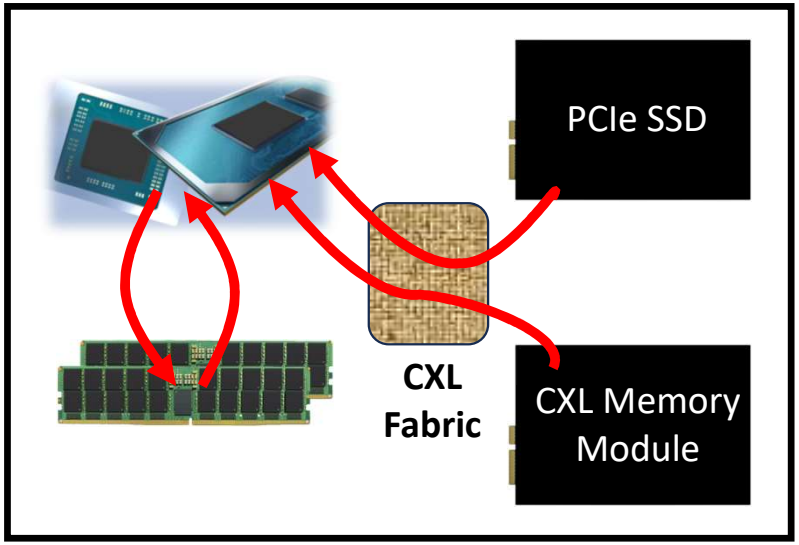
Processor grabs only the FLITs needed using CXL.mem

The rest of the CMB data (on average, 97%) remains where it is

This cache management scheme is expanded to create Virtual HDM



Roofline model

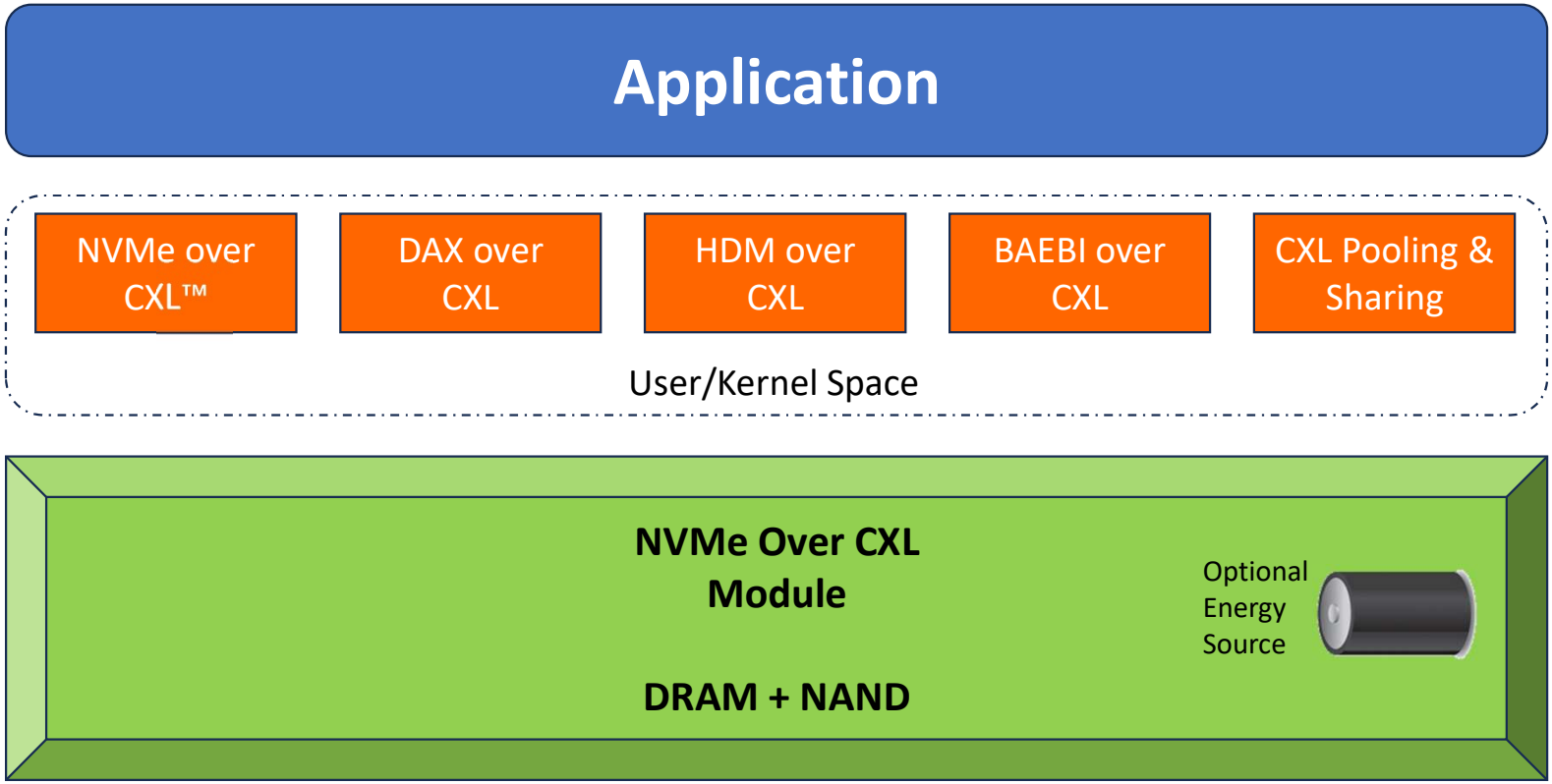


NVMe-oC addresses the memory wall which limits AI

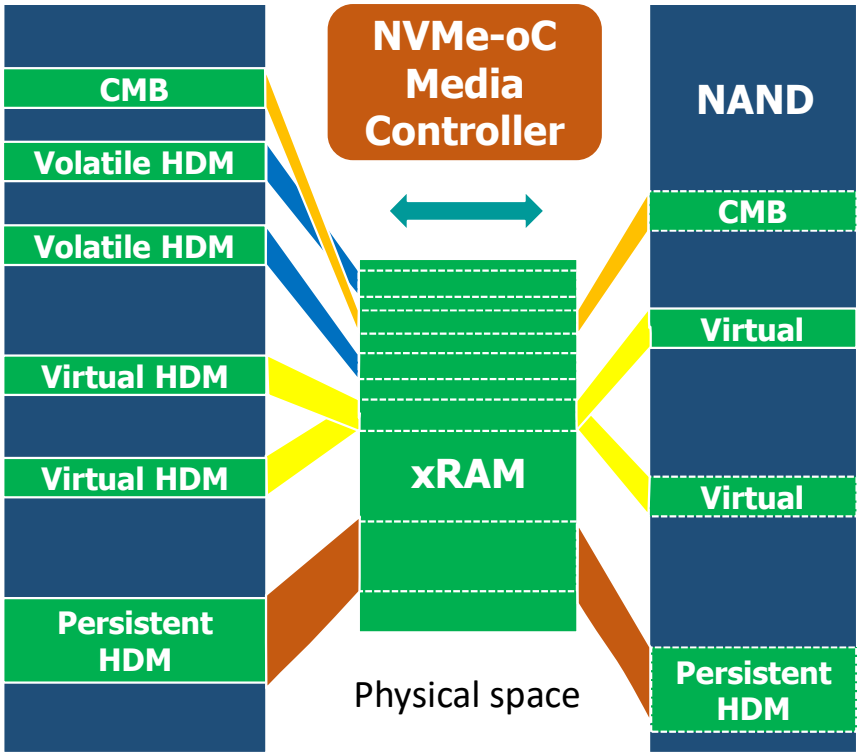
Always let the Host decide where data belongs

NVMe-oC reduces wasted data traffic over the fabric by 30x or more

NVMe-oC supports persistence, allowing checkpoint elimination



Use existing software APIs where possible



NVMe-oC operates in all access modes simultaneously

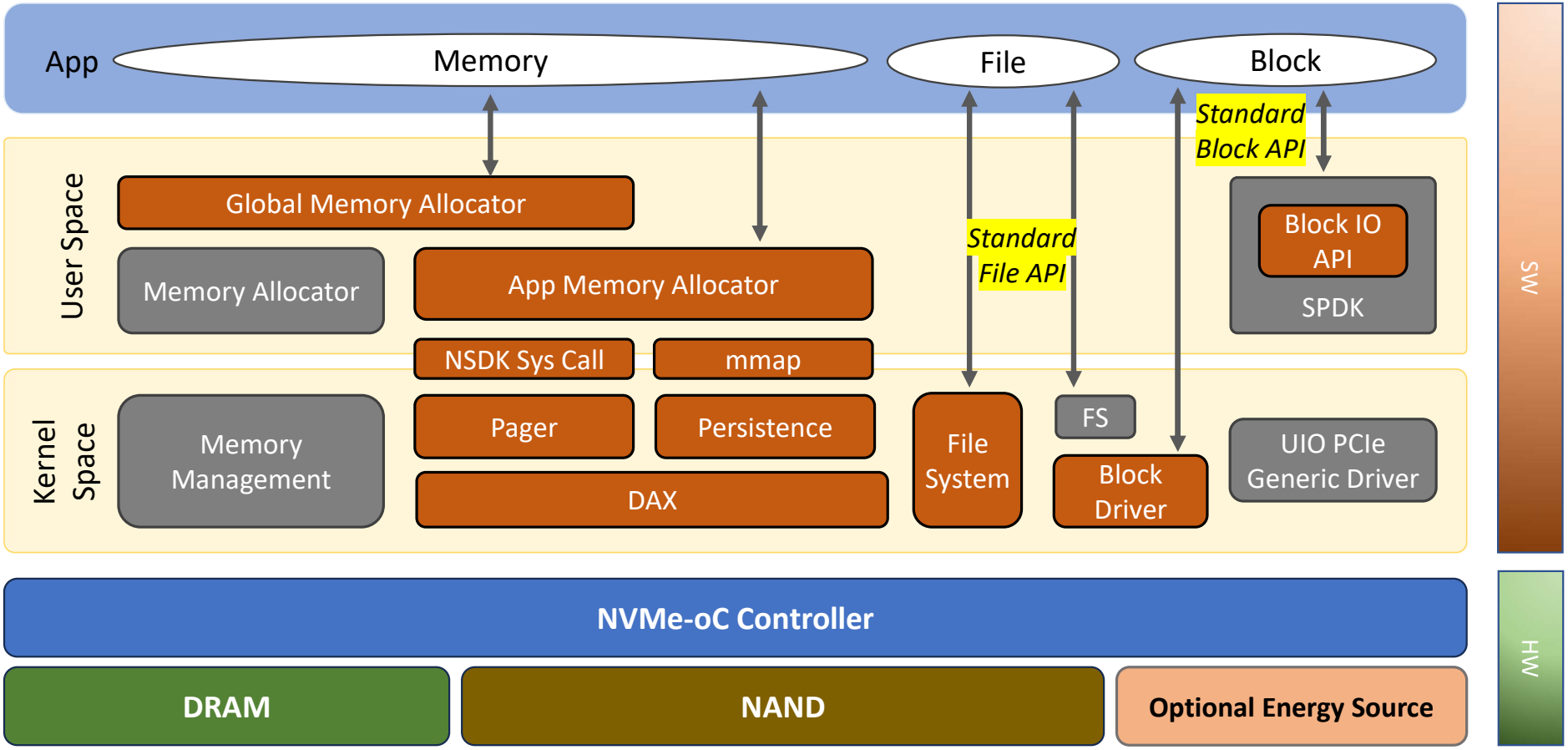
xRAM always accessed as HDM

CMB, DAX, HDM all allowed

NAND to xRAM transfer schemes driven by host using NVMe commands

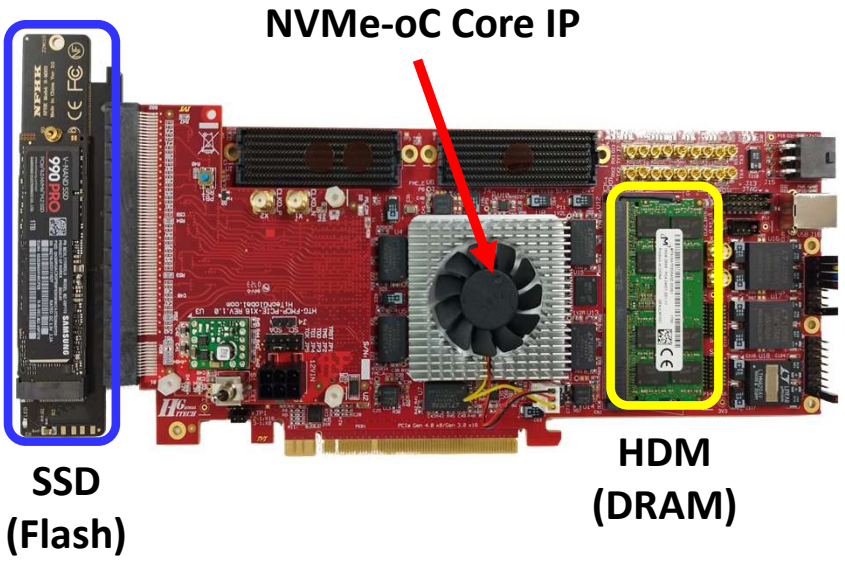
Persistence regions can be partial

NVMe-oC Software Development Kit (NSDK)



NVMe-oC Demonstration Platform – Booth 952

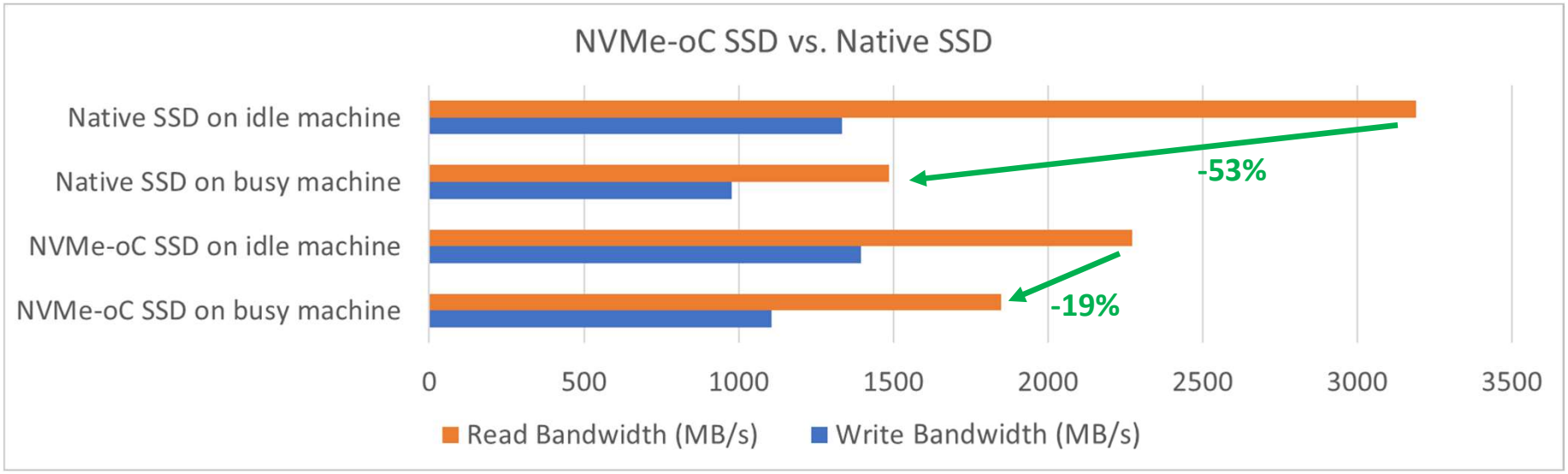
Device	
Host Interface	CXL 1.1/2.0 Gen3x8
HDM	16GB (DDR4-2000)
SSD	128GB ~ 1TB
System Clock	250MHz
NVMe	2.0
Operation Mode	Memory / Storage



Demonstrating Virtual HDM mode using NVMe-oC

Host	
CPU	Intel Granite Rapids, 2 processors, 288-cpu
Memory	128GB DRAM 6400MT
OS	Fedora release 40 (Forty)
Kernel	6.9.5

NVMe-OC Versus Traditional SSD

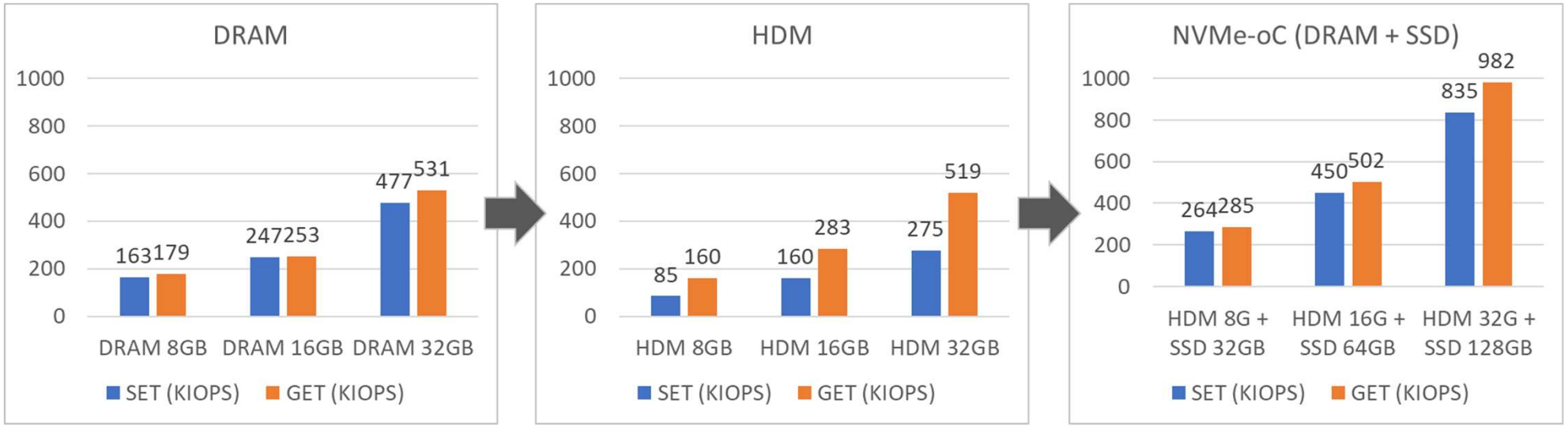


STREAM (memory benchmark) with 256 background threads

Results:

2.8X reduced impact on read performance

Virtual HDM Mode Redis Performance Versus HDM



Unmodified Redis (In-memory Key Value Store)

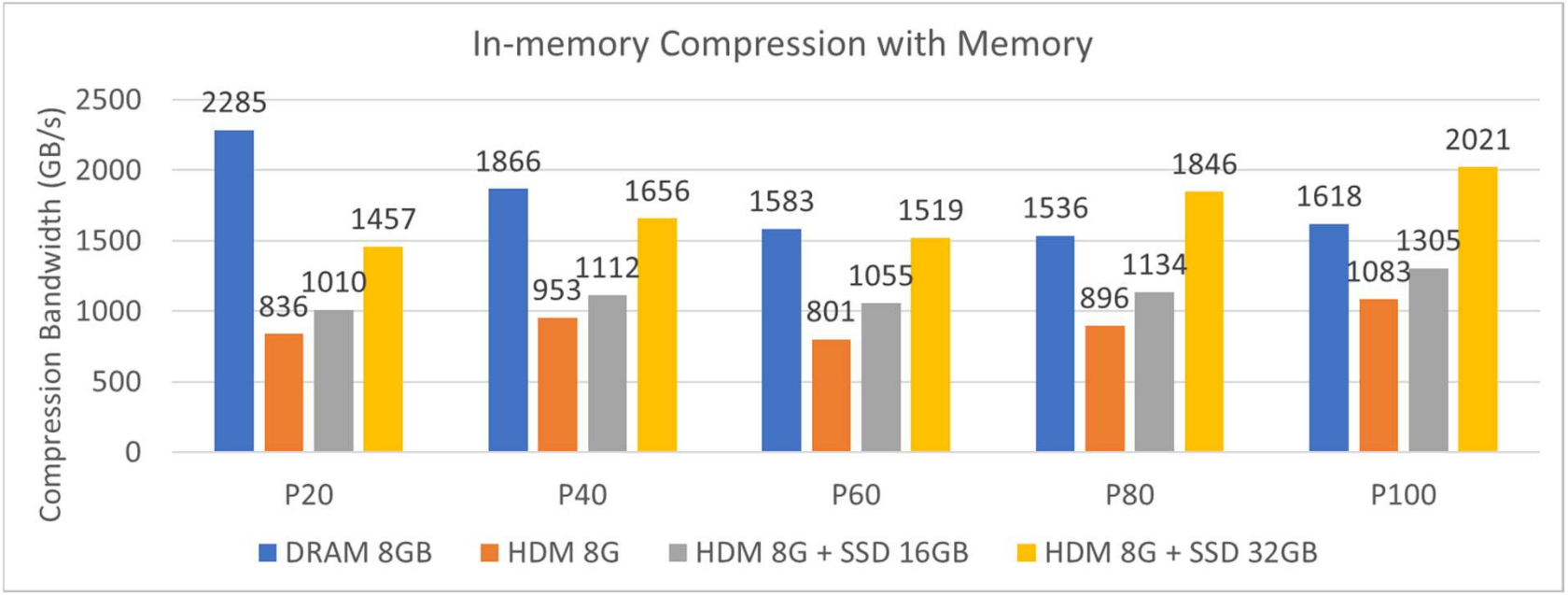
Results:

4X memory capacity

2X performance

90% cost reduction

Virtual HDM Mode Compression Performance Versus HDM



LZ77 lossless data compression method

Results:

4X memory compression

Same performance as DRAM

More compression threads executed

Thank you for your time

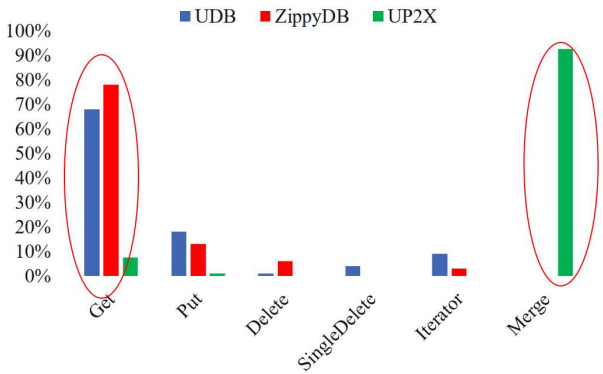
Bill Gervasi, Principal Systems Architect
bilge@wolleytech.com

San Chang 張育銘, Director of Engineering
san@wolleytech.com

Backup data

Why HDM CMB?*

Facebook RocksDB

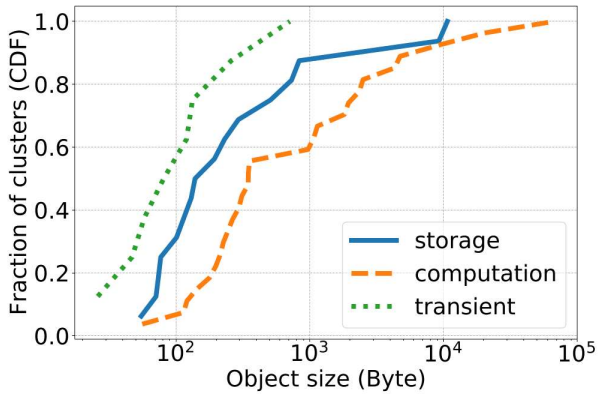


The average key size (AVG-K), the standard deviation of key size (SD-K), the average value size (AVG-V), and the standard deviation of value size (SD-V) of UDB, ZippyDB, and UP2X (in bytes)

	AVG-K	SD-K	AVG-V	SD-V
UDB	27.1	2.6	126.7	22.1
ZippyDB	47.9	3.7	42.9	26.1
UP2X	10.45	1.4	46.8	11.6

Cao, Zhichao, et al. "Characterizing, Modeling, and Benchmarking RocksDB Key-Value Workloads at Facebook" 18th USENIX Conference on File and Storage Technologies (FAST 20). 2020.

X (Twitter) Twemcache



Yang, Juncheng, Yao Yue, and K. V. Rashmi. "A Large-scale Analysis of Hundreds of In-memory Key-value Cache Clusters at Twitter" ACM Transactions on Storage (TOS) 17.3 (2021): 1-35.

* Translation: why use the DRAM on the device as CXL memory?

Majority of data accesses are between 50 and 300 bytes with median ~100 bytes (key values, objects)

With NVMe-oC, PCIe traffic reduction > 97%

NVMe-oC *Virtual HDM* Mode: Joint Management of DRAM + NAND by DAX

DAX software concept is introduced in Persistent Memory Programming Model for byte-addressable memory (e.g., NVDIMM, Intel® Optane™ persistent memory)

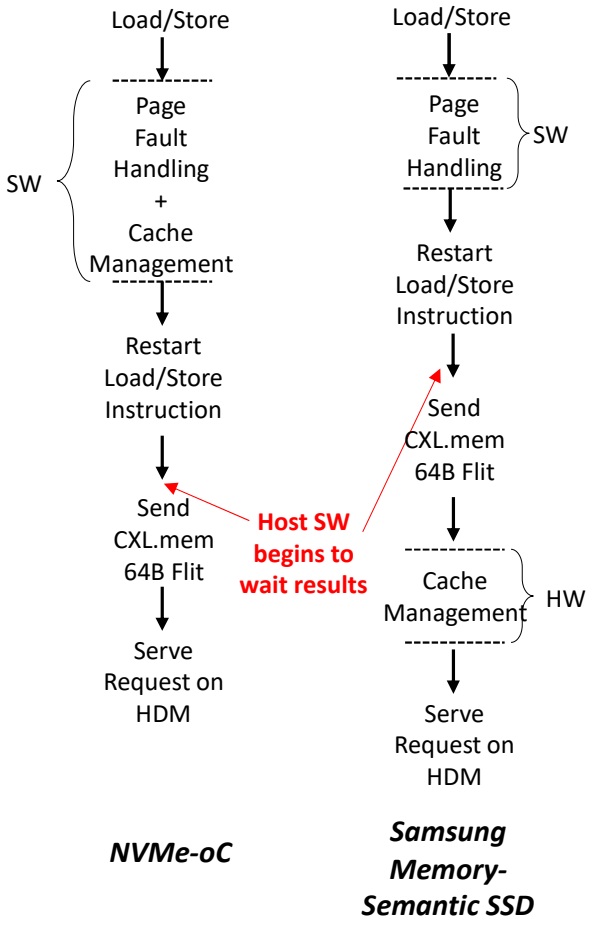
- Eliminate I/O stack software overhead to shorten latency for small and random I/O or memory operations

High-performance Memory Virtualization

- NVMe-oC DAX allows applications to run on large "memory" address space backed by SSD

SW-managed Cache vs. HW-managed Cache

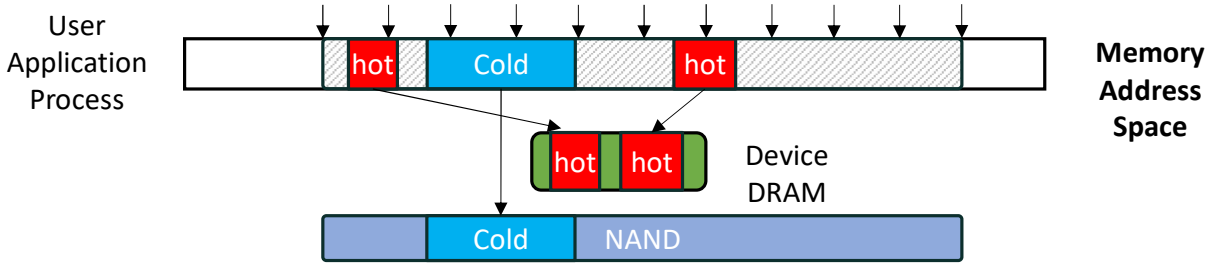
- NVMe-oC DAX manages page allocation/eviction on HDM
- NVMe-oC DAX reacts to load/store in time after sending requests to HW, while Samsung Memory-Semantic SSD may experience unexpected latency and result in generic software time-out



Host Intelligence in NVMe-oC DAX

Hit Rate Optimization on Device DRAM (as a cache)

- OS-managed Intelligence
  - Built-in cache strategy in NVMe-oC DAX
- App-managed Intelligence
  - Users can optimize their applications with additional hint to the cache management
- Prefetching



Samsung CMM-H solution relies on the OS to provide hints to memory-semantic SSDs, where the hardware independently manages cache based on those hints.

- NVMe-oC host-side cache management can precisely control over data allocation and eviction by software, offering the greatest flexibility and adaptability