SNIA DEVELOPER CONFERENCE
SDC 25
By Developers FOR Developers

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

Host Addressable Subsystem Local Memory

Jason Molgaard

www.sniadeveloper.org

Agenda

- Why Host Addressable NVMe® Subsystem Local Memory (SLM)
- SLM Theory of Operation
- Benefits of Host Addressable SLM
- Computational Storage: a Use Case for Host Addressable SLM

Why Access SLM Using Host Addressing?

Why Access Subsystem Local Memory (SLM) Using Host Addressing

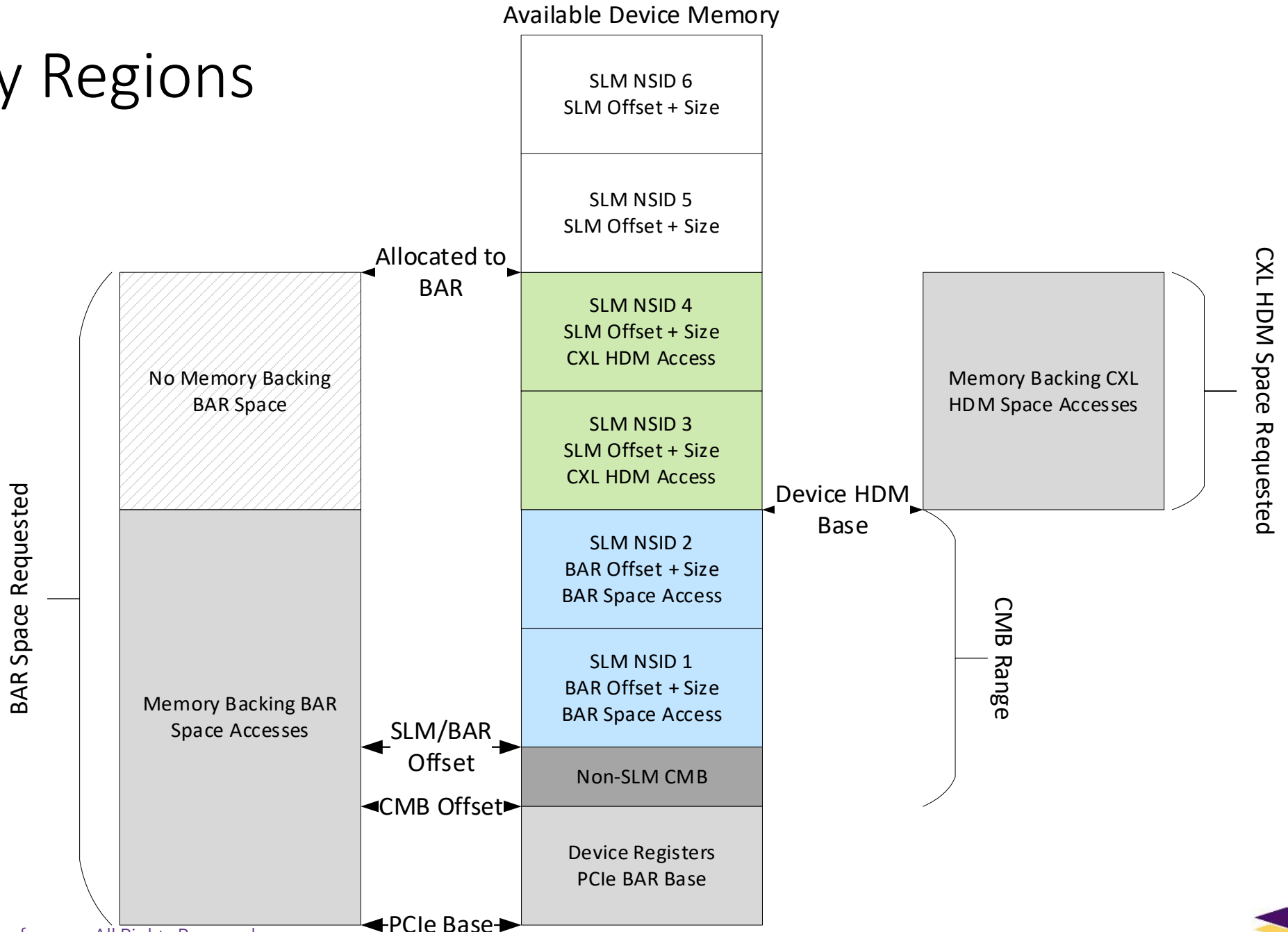
- NVMe® devices are providing host accessible memory in the form of SLM
  - Accessing this memory via a memory protocol is more efficient
  - Allows peer-to-peer communication using a memory model
  - If using CXL®, allows cache coherency of that memory
- Computational Storage Use
  - Computational Storage Drives have more host accessible memory than a traditional Storage Device
  - Benefits from peer-to-peer communication

How Does Host Addressable SLM work?

SLM Theory of Operation

- SLM namespaces may be accessed by one of three mechanisms:
  - SLM command access only
  - SLM command access and PCIe Base Address Registers (BAR) access
  - SLM command access and CXL® Host-managed Device Memory (HDM) access
- At boot time, the mechanism of access for each namespace is determined
  - The device requests BAR space for all SLM namespaces that may be BAR accessible
  - The CXL aware NVMe® driver calls CXL kernel services to discover and configure any CXL HDM SLM memory

Memory Regions



Host Addressable SLM – NVMe® TP4184

- NVMe TP4184 is in the Architecture Definition phase
- SLM memory is accessible by the host and the device
- SLM is addressed at a Host Physical Address (HPA)
  - PCIe® BAR; or
  - CXL®
- SLM can be read/written with:
  - SLM commands
  - Host Load/Store commands
  - CXL.mem commands
- SLM memory can have a virtual mapping for host applications
  - Host application does not have to switch contexts for SLM memory access
- Compute is triggered with Computational Programs commands
  - Utilizes the host addressable SLM
- Allows P2P data movement based upon HPA
- SLM is still accessible using the SLM Command Set Memory Read and Memory Write commands

Benefits of Host Addressable SLM

Benefits of Host Addressable SLM

- Multiple access methods
- Load/Store access provides low latency, fine granularity path to access SLM
- CXL® support enables coherency with SLM
- Avoid Copies and data movement
- Bypass the host for P2P

PCIe® BAR access versus CXL HDM access

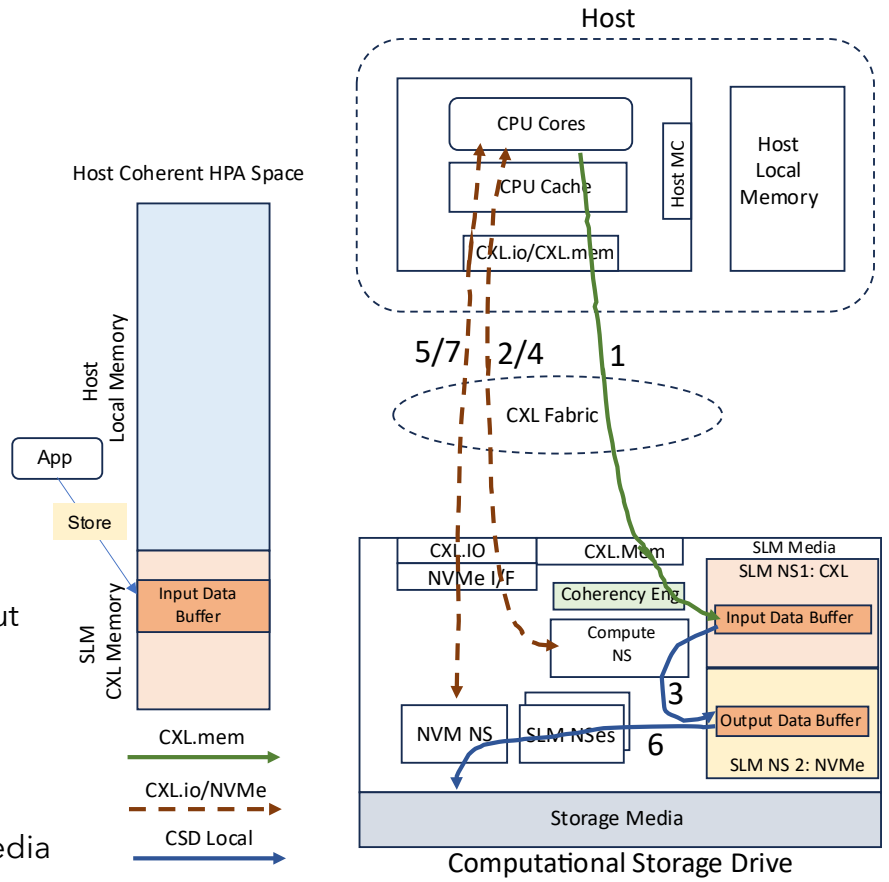
CXL allows both coherency with host memory and Memory-mapped I/O (MMIO) space

PCIe BAR allows host Load/Store access over PCIe® architecture using uncached MMIO space

Use Cases

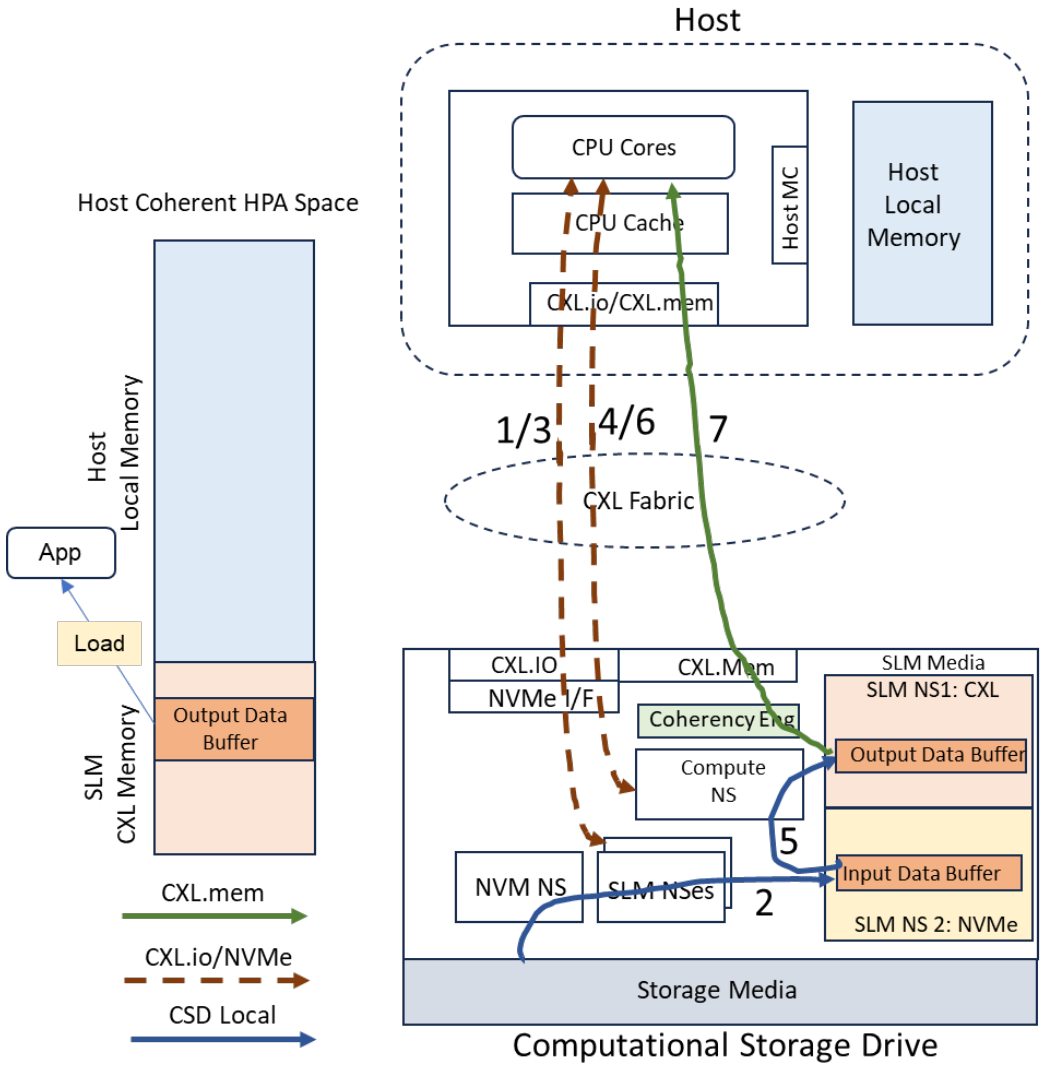
Use Case 1: Data Post-Processing (Before Writing to Storage)

- Value Proposition
  - Avoid copying data using DMA from/to Host Memory
  - Lower latency CXL® based direct Load/Store access, especially for small input data
- Configuration
  - Input Data Buffer is in SLM CXL memory address space
  - Output Data Buffer is in SLM
- Example Use Case
  1. Application writes (Store) Input Data Buffer using CXL.mem
     - Some or all data may reside in Host Cache on completion
  2. Host issues NVMe® Execute Program command to Compute Namespace
  3. Compute Namespace Operates on data in Input Data Buffer and stores results in Output Data Buffer
     - Uses CXL BI Snoop protocol to keep Host Cache coherent with Input Data Buffer
  4. CQE is posted for the Compute Namespace
  5. Host issues NVMe Copy command to copy data from Output Data Buffer to Storage Media
  6. Data is copied to Storage Media from Output Data Buffer
  7. CQE is posted for the NVM Namespace



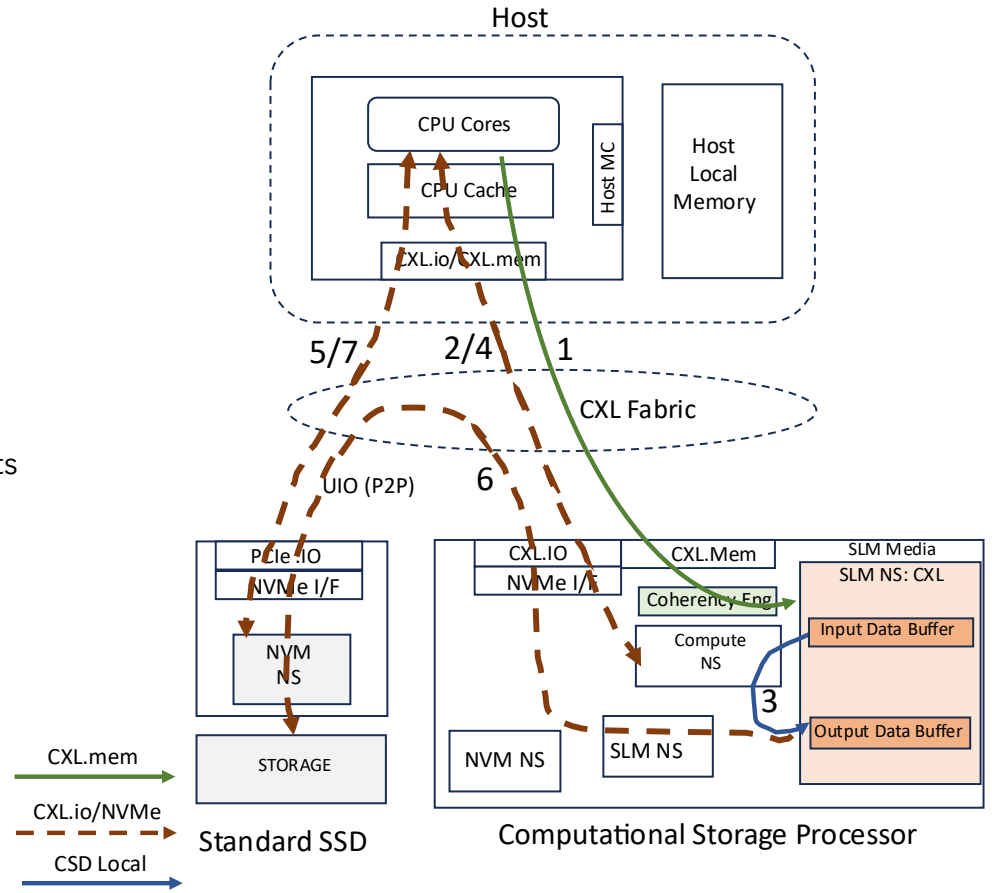
Use Case 2: Data Pre-Processing (before sending to host)

- Value Proposition
  - Avoid copying data using DMA from/to Host Memory
  - Lower latency CXL based direct Load/Store access, especially for small output data
- Configuration
  - Input Data Buffer is in SLM
  - Output Data Buffer is in host addressable SLM
- Example Use Case
  1. Host issues NVMe Memory Copy command to SLM NS
  2. Data copied from NVM NS to Input Data Buffer
  3. CQE is posted for SLM NS
  4. Host issues NVMe Execute Program command to Compute NS
  5. Compute NS operates on data in Input Data Buffer and stores results in Output Data Buffer
     - ➢ Uses CXL BI Snoop protocol to keep Host caches coherent with Output Data Buffer
  6. CQE is posted for Compute NS
  7. Application reads (ld/st) Output Data Buffer using CXL.mem



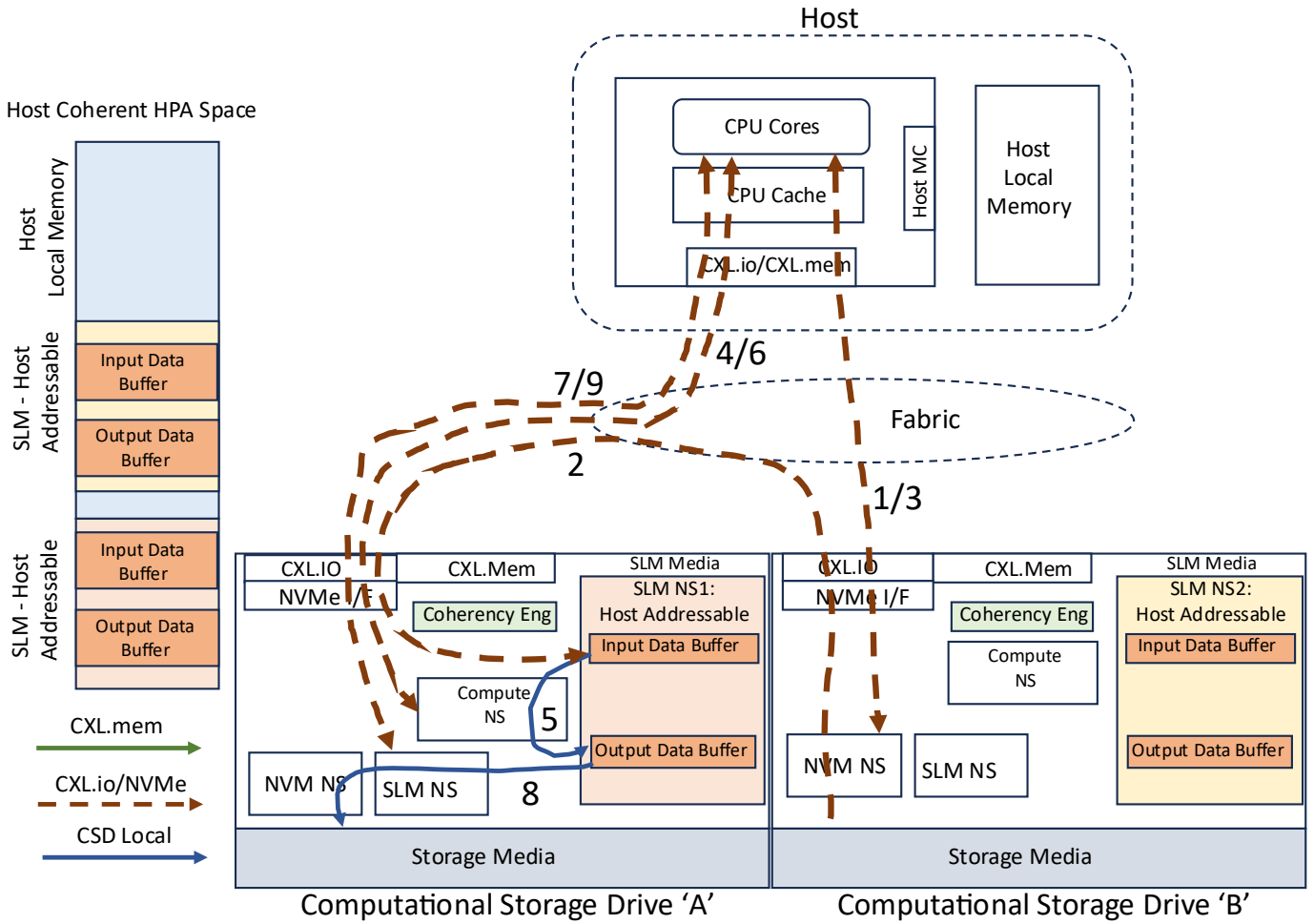
Use Case 3: Data Post-Processing with a Standard SSD

- Value Proposition
  - Bypass data movement through Host memory
- Configuration
  - Input Data Buffer is in SLM CXL® memory address space
  - Output Data Buffer is in SLM CXL memory address space
- Example Use Case
  1. Application writes (Store) Input Data Buffer using CXL.mem
     - Some or all data may reside in Host Cache on completion
  2. Host issues NVMe® Execute Program command to Compute Namespace
  3. Compute Namespace operates on data in Input Data Buffer and stores results in Output Data Buffer
     - Uses CXL BI Snoop protocol to keep Host Cache coherent with Input Data Buffer and Output Data Buffer
  4. CQE is posted for Compute Namespace
  5. Host generates IO Write to SSD NVM Namespace
     - Data Pointer points to Output Buffer in SLM (HDM)
  6. SSD uses PCIe® UIO for direct P2P from HDM space and writes to storage media
     - Since output buffer is in CXL HDM space, UIO can't use BAR space for P2P
  7. CQE is posted for NVM Namespace



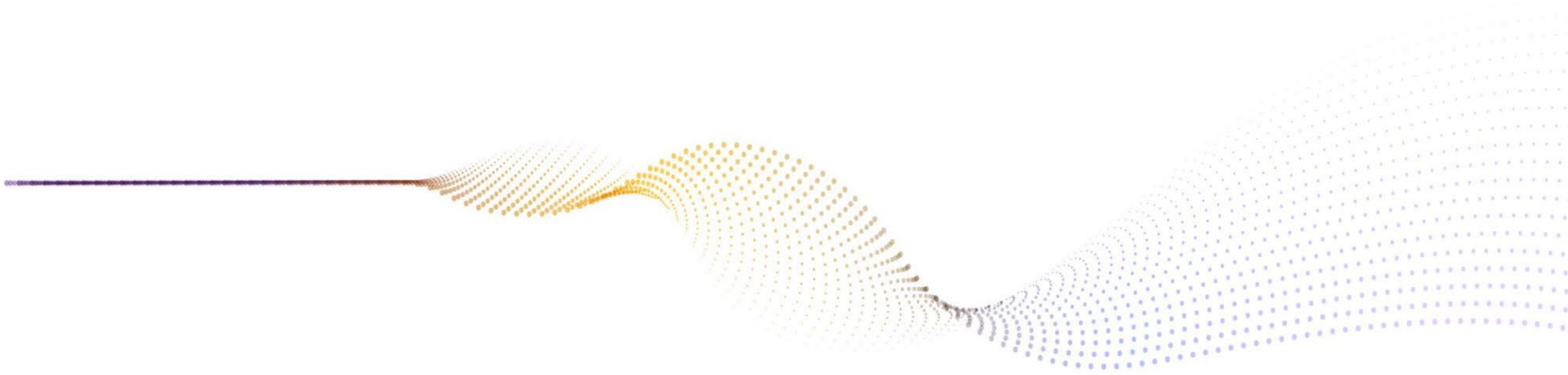
Use Case 4: Peer Data Processing

- Value Proposition
  - Bypass data movement through host memory
- Configuration
  - Data resides in peer device and is moved to Input Data Buffer
  - Input Data Buffer is in host addressable SLM
- Example Use Case
  1. Host issues NVMe® Read command to Drive B with the destination is Input Data Buffer on Drive A
  2. Data is transferred P2P from Drive B to Drive A
  3. CQE is posted for READ command
  4. Host issues NVMe Execute Program command to Compute Namespace on Drive A
  5. Compute Namespace on Drive A operates on data in Input Data Buffer and stores results in Output Data Buffer
  6. CQE is posted for the Compute Namespace
  7. Host issues NVMe Copy command to copy data from Output Data Buffer to Storage Media
  8. Data is copied to Storage Media from Output Data Buffer
  9. CQE is posted for the NVM Namespace



Summary and Next Steps

- NVMe® technology is being enhanced to add host addressing to SLM
  - The work is defined by TP4184
  - There will be three access methods for SLM
- Benefits
  - Load/Store access to NVMe® SLM
  - Potential coherency between device SLM and host
  - Lower latency for small data transfers between host and SLM
  - Since SLM is addressable via HPA, no need to copy data from host memory to SLM
  - Bypassing the host for peer-to-peer data movement
- Looking Ahead
  - Enhancing NVMe SLM to be host addressable enables new use cases for Computational Storage



Thank you for attending!

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

