Enabling Remote Persistent Memory

Idan Burstein, Mellanox Technologies, Inc.

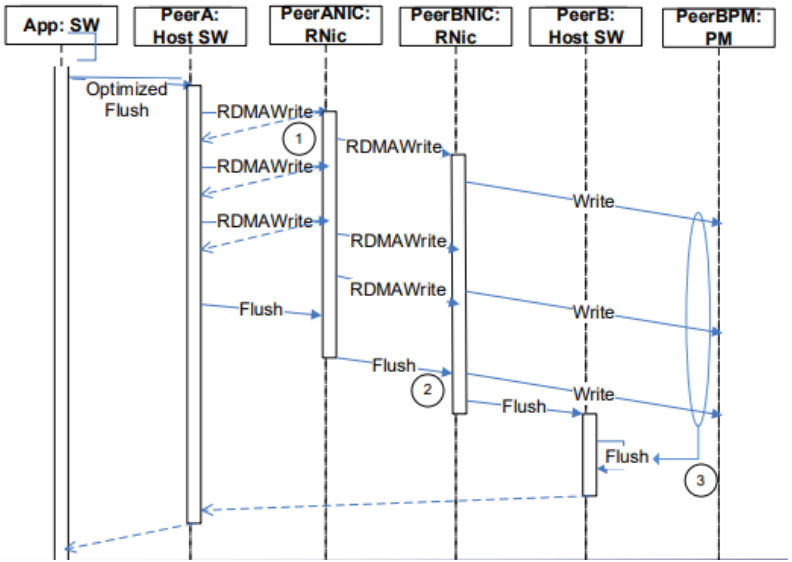
Agenda

- RDMA Remote Persistent Memory Workload
- RDMA Memory Placement Extensions
- RDMA Memory Management
- Summary

Remote Persistent Memory Workloads

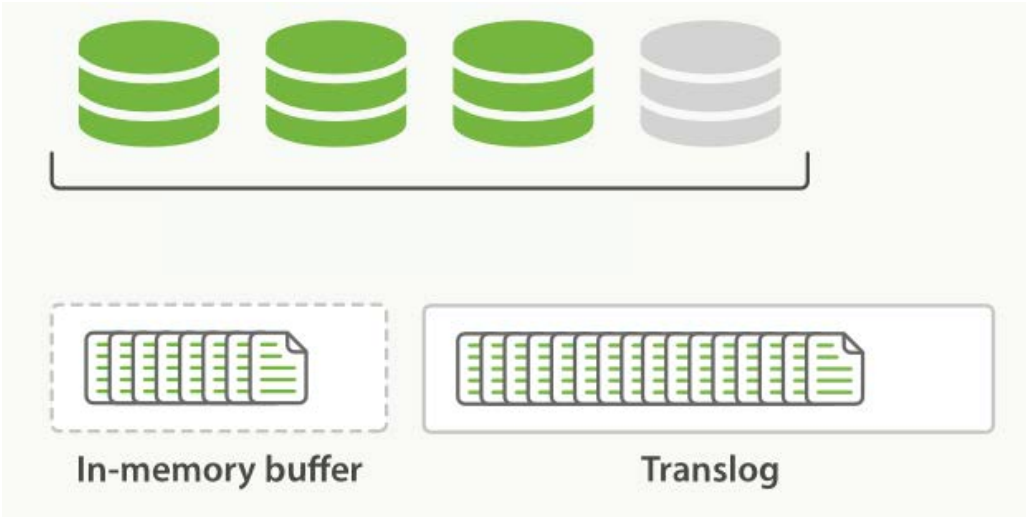
Replication

- Replicate persistent memory files / blocks
  - Mirror scatter-gather of updates to remote persistent memory
  - Async / sync draining of the data
  - Verify persistency
- May require
  - Integrity checking
  - Data at rest encryption



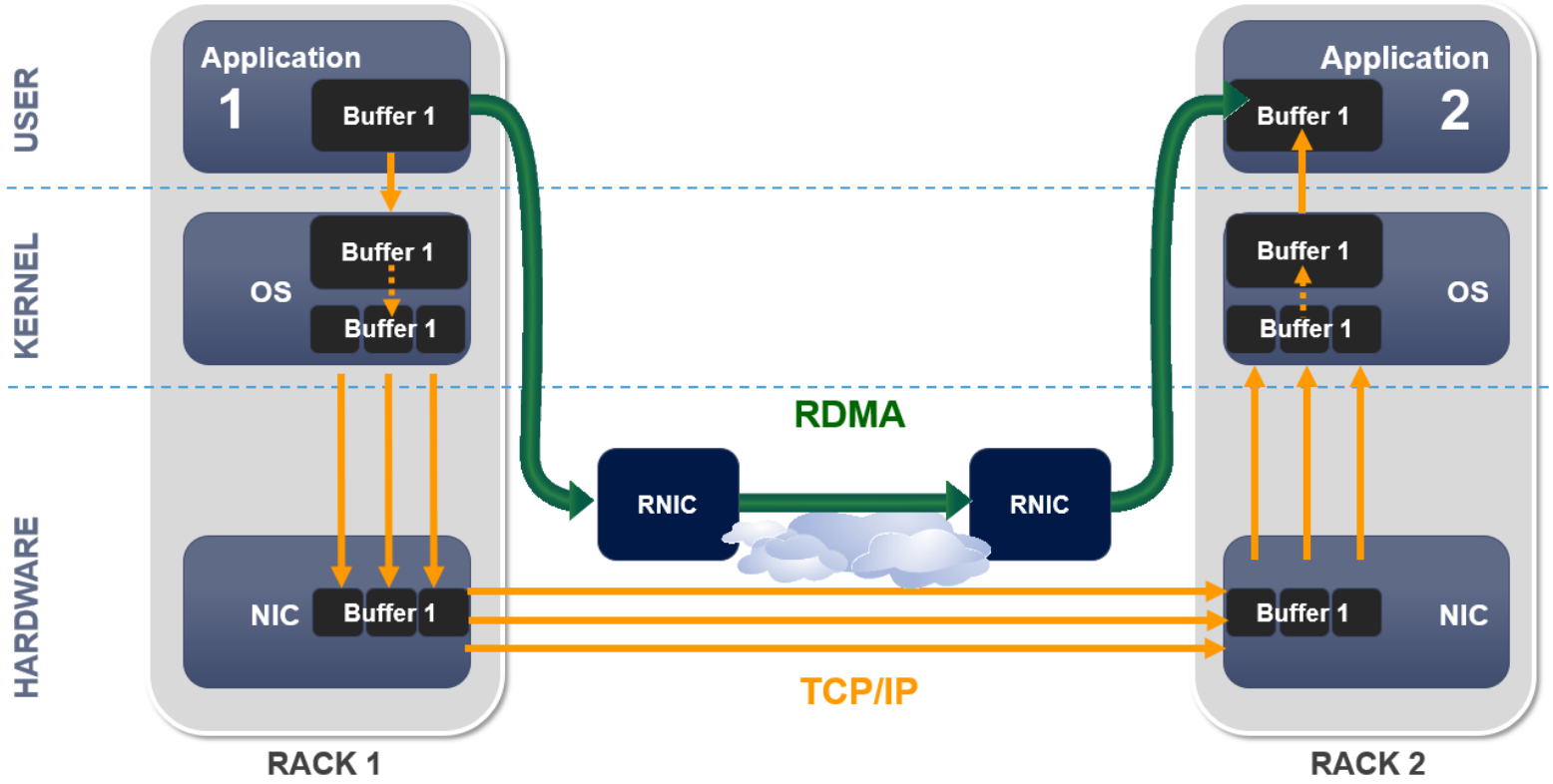
Two Phase Commit

- Write data, commit to memory and update flag/pointer
- Typical workload in file system and databases
- Flag/pointer must be committed to media after data is committed



RDMA Memory Placement Extensions

RDMA – How does it Work



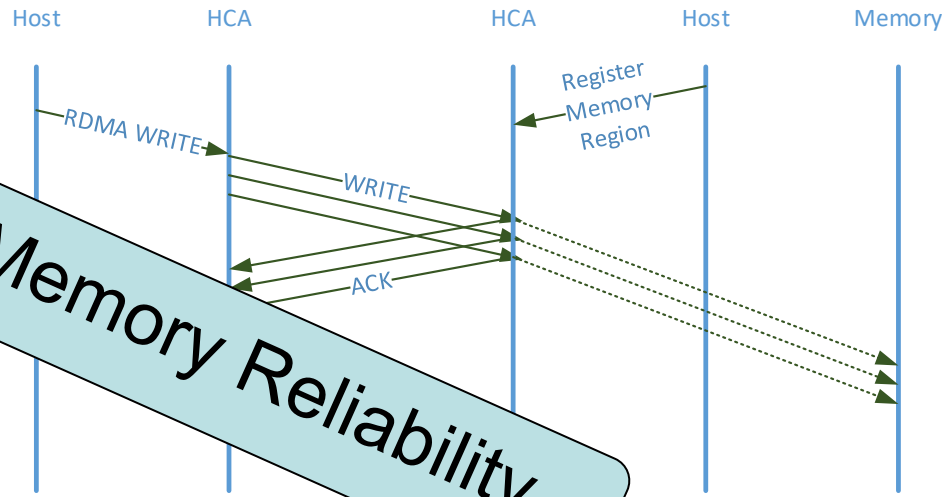
RDMA Principles

- Transport built on simple primitives deployed for 15 years in the industry
  - **Queue Pair (QP)** – RDMA communication end point
  - **Connect** for establishing connection mutually
  - RDMA **Registration** of memory region (REG_MR) for enabling virtual network access to memory
  - **SEND** and **RCV** for reliable two-sided messaging
  - RDMA **READ** and RDMA **WRITE** for reliable one-sided memory to memory transmission

- Reliability
  - Delivery
  - Once
  - In order

RDMA WRITE Reliability Scope

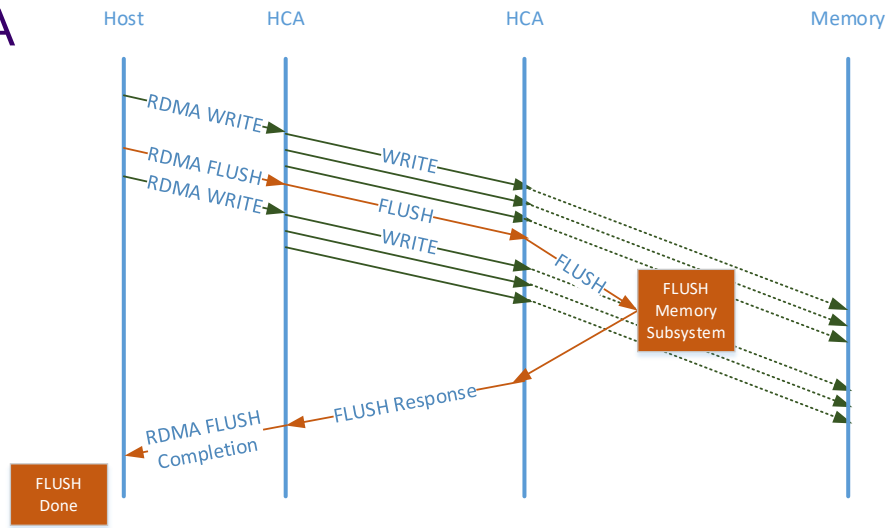
- RDMA Ackn (and Completion)
  - Guarant en successfully d for execution by the re
  - Doesn't guarantee data ha remote host memory
  - Doesn't guarantee the data can be visible/durable for other consumers accesses (other connections, host processor)
- Further Guarantees Implemented by ULP



New Concept – Memory Reliability

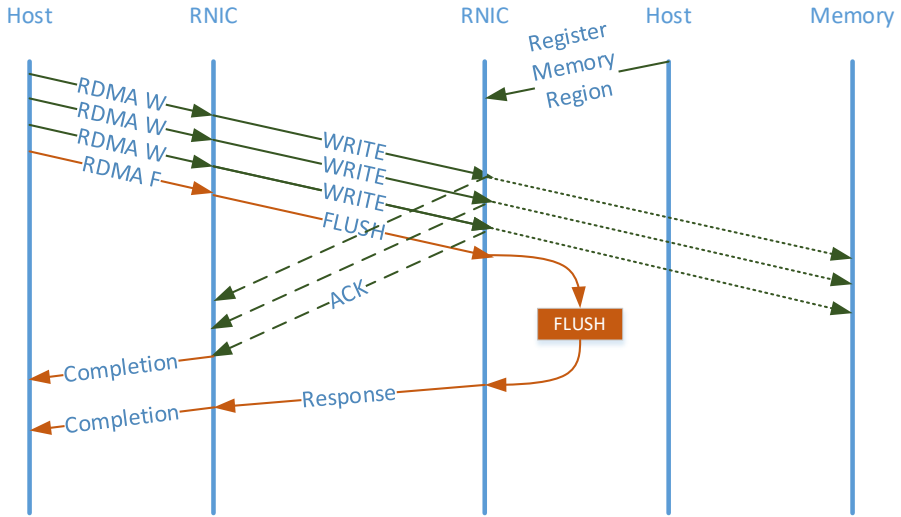
RDMA FLUSH

- New transport operation – RDMA FLUSH
  - To provide memory placement guarantees to the upper layer software
- RDMA memory operations remain unchanged



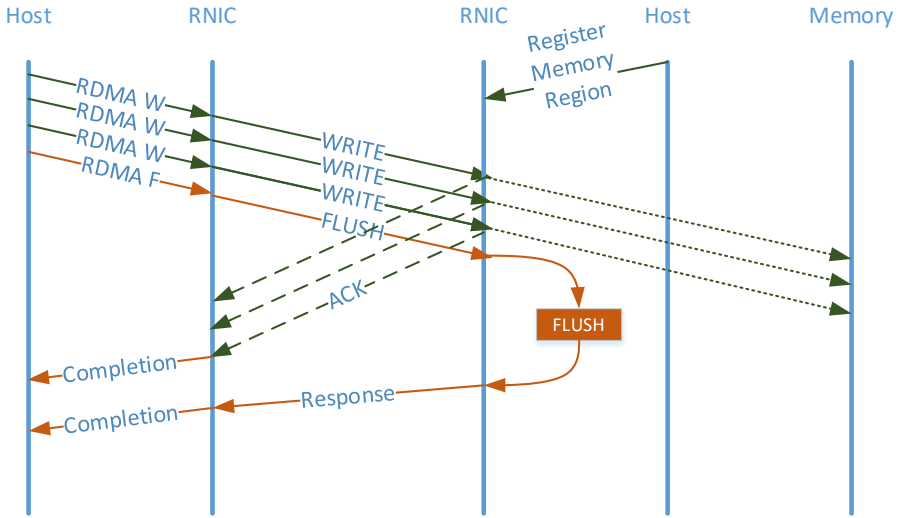
RDMA Flush - Requirements

- Performance Requirements
  - In Transport Latency → “Non-Posted” / “Delayed”
  - Selectiveness → “Apply on certain ranges”
  - Pipelining → “No need to fence the network”
  - Amortization → “One on many WRITEs”
- System level implication may be:
  - Caching efficiency
  - Persistent memory bandwidth / durability
- Types
  - Global Visibility
  - Global Visibility & Persistency



RDMA Flush - Overview

- New transport operation for providing memory placement guarantees
  - Non posted (execution could be delayed)
  - Explicit response
- Associated with memory key range
- Memory placement types
- Selectivity levels



RDMA Flush - Types

- Global Visibility
  - FLUSH type global visibility shall ensure the placement of the preceding data accesses in the memory domain which **visible for reading for the responder platform**

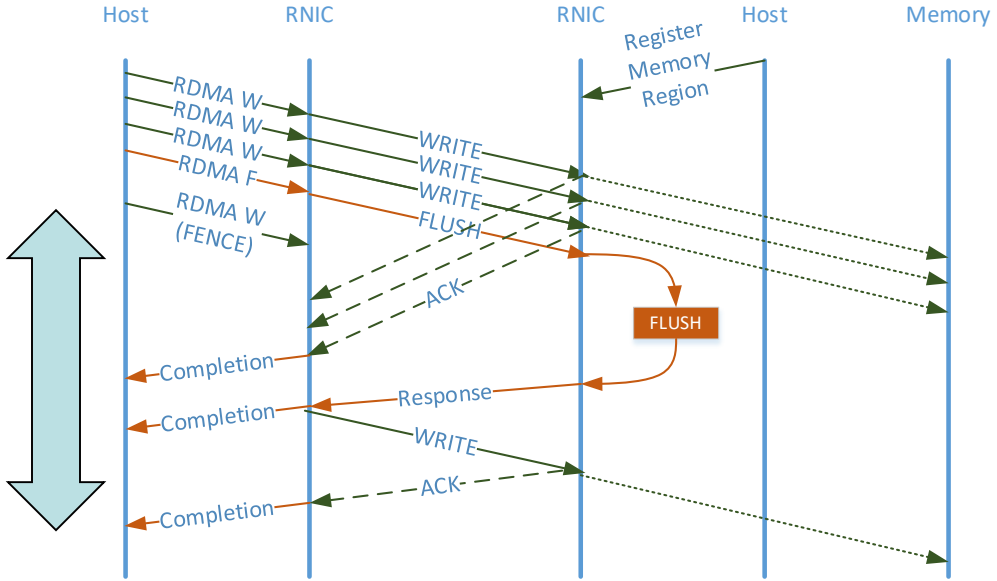
- Persistency
  - FLUSH type persistency shall ensure the placement of preceding data accesses in a memory that **persists the data across power cycle and globally visible**, response shall be send only after successful completion in the responder.

RDMA Flush – Selectivity Level

- Memory Region Range
  - FLUSH preceding data access within the RETH range {RKEY, VA, Length} within the QP
- Memory Region
  - FLUSH preceding data access within the RETH.RKEY within the QP

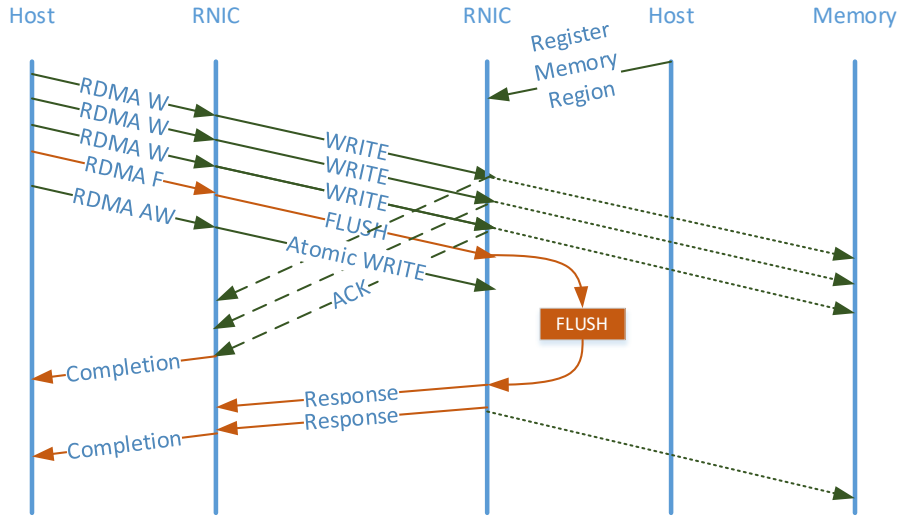
Enabling Efficient Two Phase Commit

- RDMA ordering rules requires the application should wait for FLUSH response before updating the flag/pointer
  - Either with fencing the QP (bad for bandwidth)
  - Or by waiting for completion (hence software interrupt)
- Anyway, round trip latency is added for committing transaction



New Transport Operation – Atomic WRITE

- Proposal: Atomic WRITE
  - New transport function
  - Atomic update of 8B
  - Non posted, ordered to non-posted operations completions (e.g. FLUSH)
- Two phase commit ordering will be done in the responder



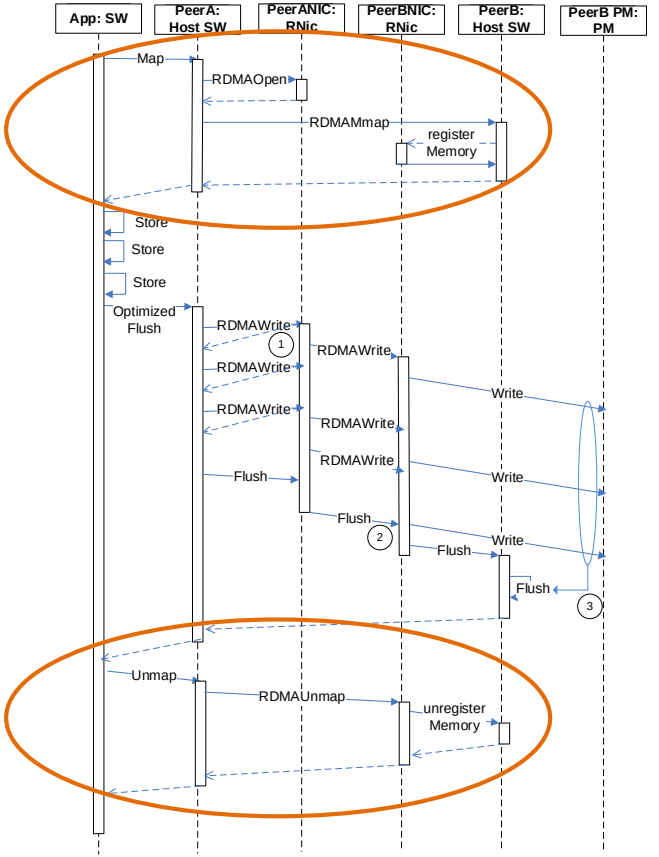
RDMA Memory Management

Memory Region

- Accessibility permissions
  - E.g. Remote, Local, READ, WRITE, FLUSH
- Protection Domain (PD)
  - Protection between applications
- Memory layout (scatter gather)
  - Scatter-gather in memory
- Integrity checking (t10dif)
  - Verify integrity of data on reads / write

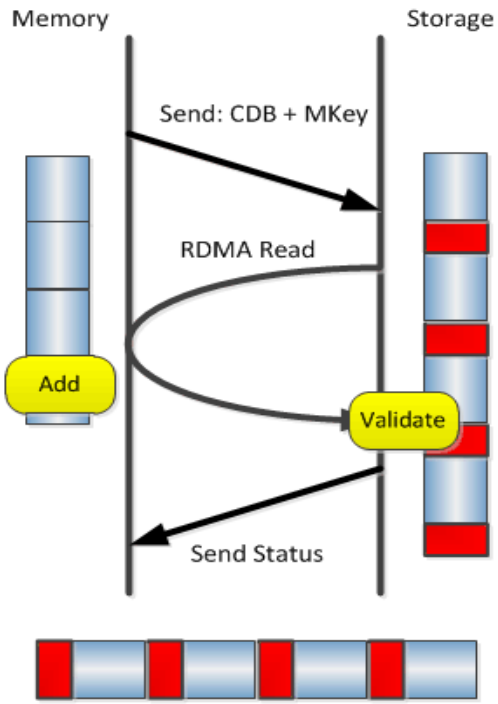
Control Plane for Memory Management

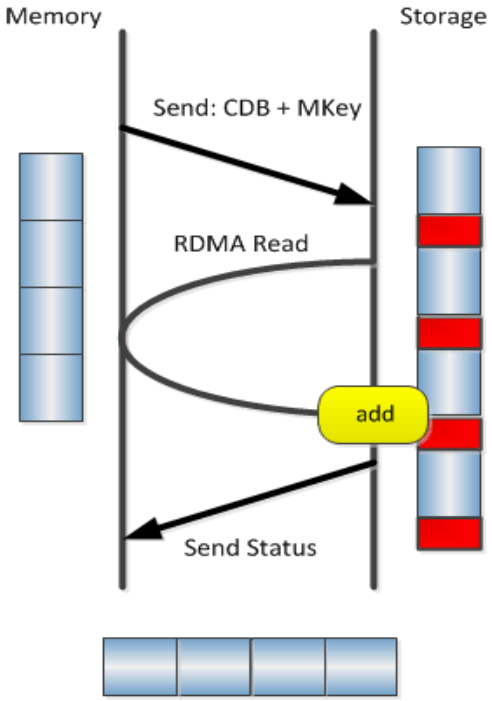
- File system or database responsibility
  - E.g. map, unmap for HA
- Allocate / deallocate
  - → Registration
- Protection
  - →Protection Domain



Memory Region Based Integrity Checking

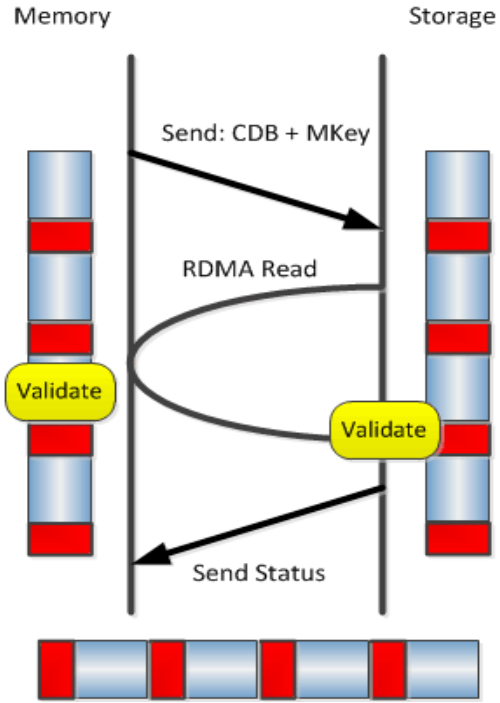
- Memory region is attributed for the signature
  - Type
  - Block size
  - App/Ref tag
- With persistent memory, there is no ULP on the storage side
  - There is a need to query the signature result

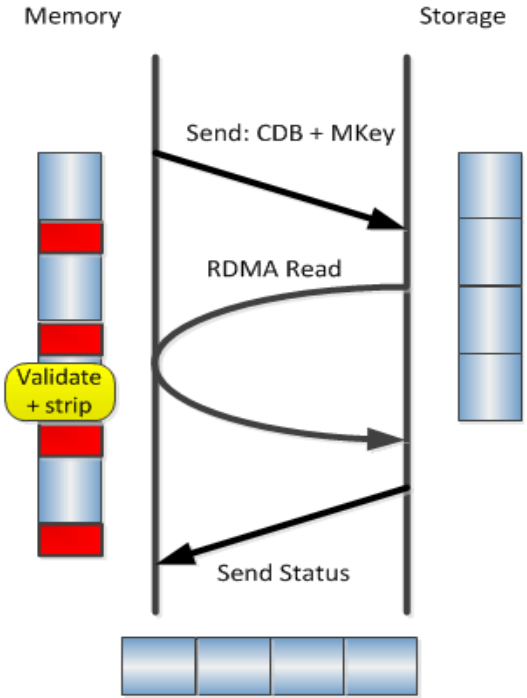




Memory Region Based Integrity Checking

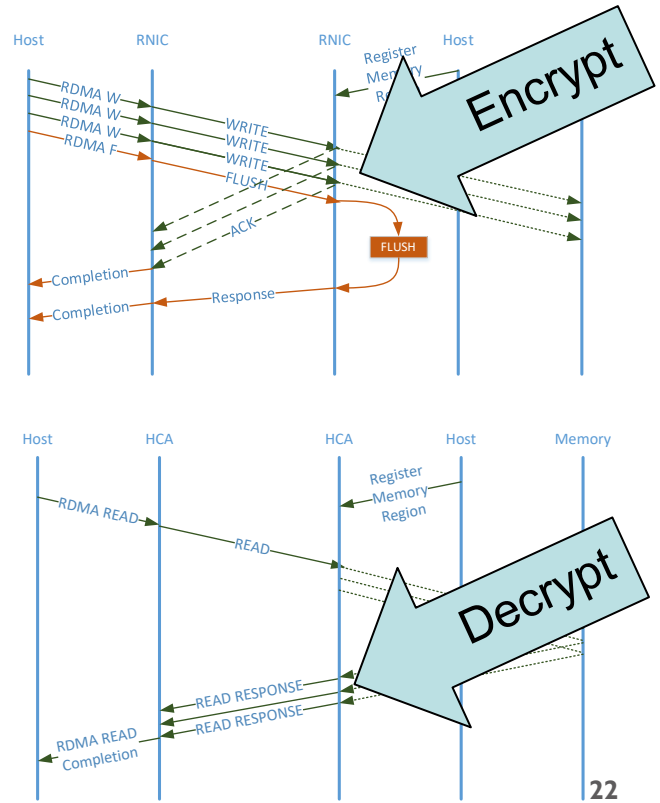
- Memory region is attributed for the signature
  - Type
  - Block size
  - App/Ref tag
- With persistent memory, there is no ULP on the storage side
  - There is a need to query the signature result





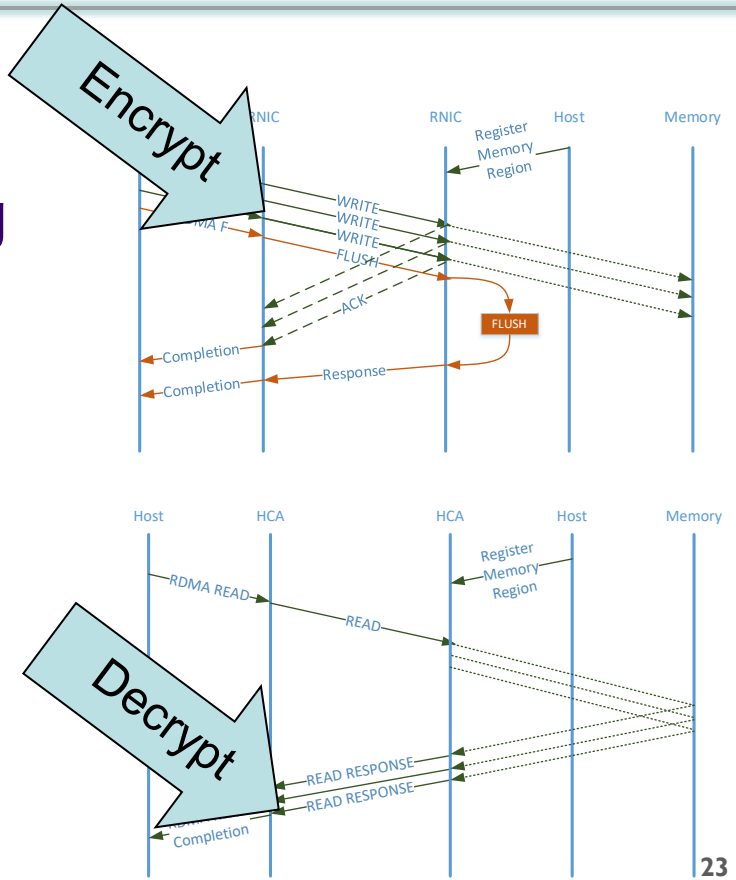
Memory Region Based Encryption

- With remote PMEM data at rest encryption becomes challenging
- Proposal: associate a memory key with encrypting key to encrypt decrypt data
  - Data must be encrypted before reaching the media
  - AES-XTS



Memory Region Based Encryption

- With remote PMEM data at rest encryption becomes challenging
- Proposal: associate a memory key with encrypting key to encrypt decrypt data
  - Data must be encrypted before reaching the media
  - AES-XTS



Summary

- InfiniBand is building the foundations for remote access to persistent memory
  - Memory reliability, security, data integrity
  - Work in progress in IBTA to produce an Annex
- To make a complete story on the system level these challenges should be addressed in other standard interfaces of memory
  - E.g PCIe