FUJITSU

shaping tomorrow with you

Unveil Remote Persistent Memory Access

Oct 24, 2021

Yang Xiao, Software Engineer, Fujitsu Nanda

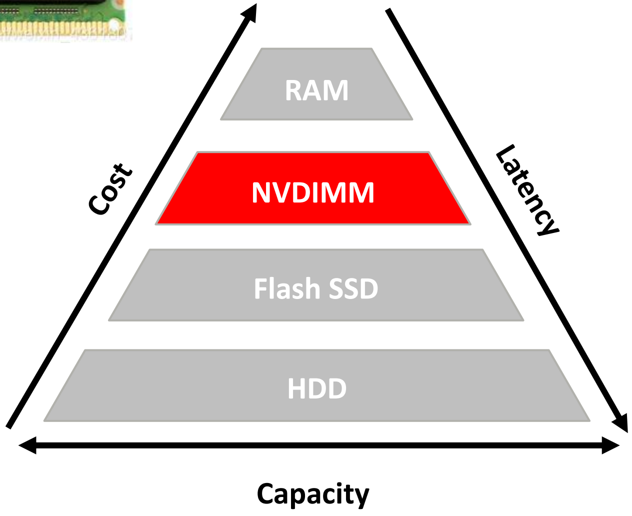
Agenda

- Persistent Memory
- Remote Direct Memory Access
- Combining RDMA with PMEM
  - Motivation
  - Shortcomings
- Remote Persistent Memory Access library
- RDMA extension
  - New RDMA FLUSH and RDMA ATOMIC WRITE operations
  - New RDMA FLUSH and RDMA ATOMIC WRITE packages over SoftRoCE
- Conclusion and future work

Persistent Memory

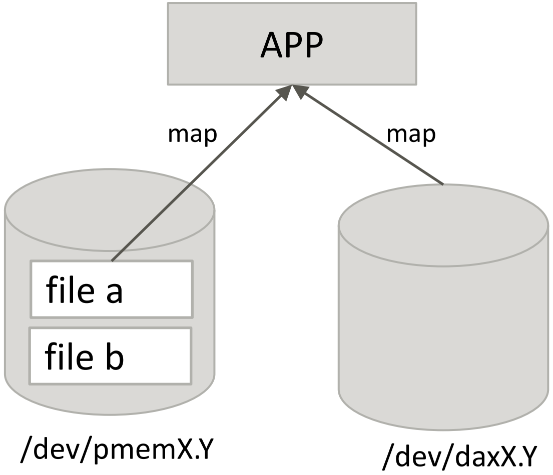
- Persistent Memory（PMEM）is a high-performance and byte-addressable memory device which resides on the memory bus.

- Advantages
  - data is not volatile after power interruption
  - have nearly the same speed and latency of DRAM
  - provide large capacity like flash SSD
  - cheaper than DRAM



Operating modes of PMEM

- Memory mode (as memory)
- App Direct mode (as storage)
  - fsdax, devdax, sector and raw

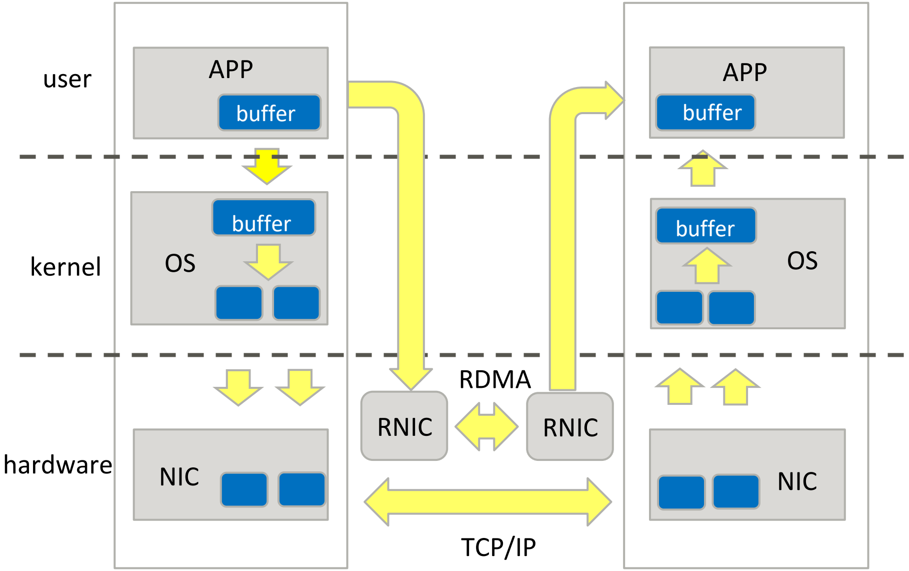


- Filesystem-DAX (fsdax)
  - Create a block device(/dev/pmemX.Y)
  - Bypass the page cache
  - Allow mmap() to establish direct mappings to files on PMEM
- Device-DAX (devdax)
  - Create a character device (/dev/daxX.Y)
  - Allow mmap() to establish a direct mapping to the whole PMEM

Remote Direct Memory Access

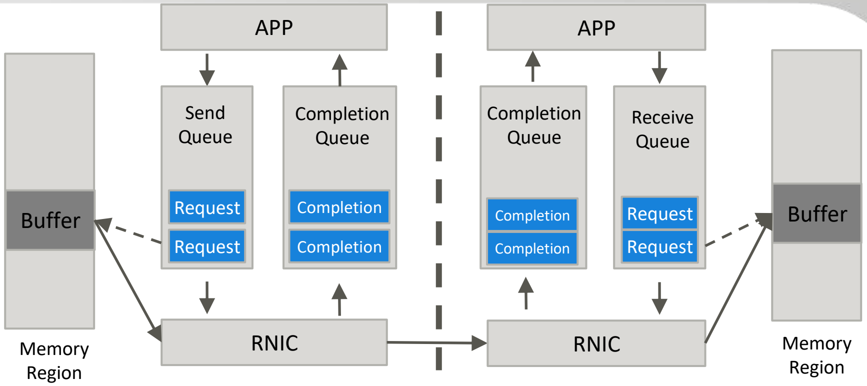
- Remote Direct Memory Access（RDMA）is a technology that enables computers in a network to exchange data in the main memory without involving operating system of either computer.

- Advantages
  - Zero copy between kernel space and user space
  - Bypass the host system’s software TCP/IP stack
  - Move data without CPU involvement by DMA engine

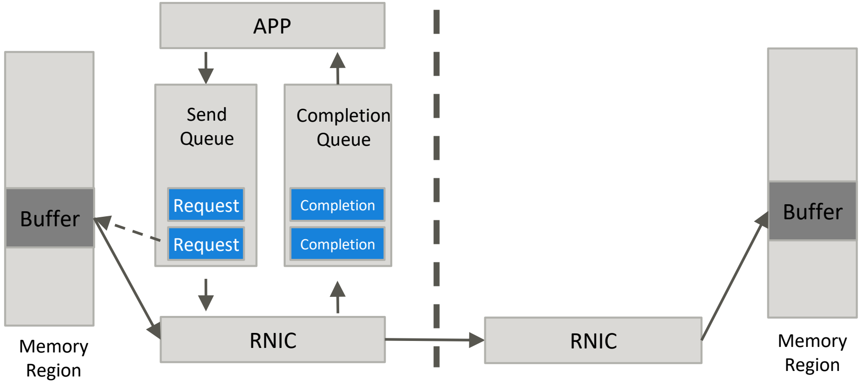


RDMA operation flow

- RDMA operations
  - RDMA two-sided operations
    - RDMA SEND
    - RDMA RECEIVE
  - RDMA one-sided operations
    - RDMA READ
    - RDMA WRITE
    - RDMA ATOMIC
- RDMA elements
  - Memory Region
  - Queue pair (Send Queue & Receive Queue)
  - Completion Queue
  - Work Request/Work Queue entry (Request)
  - Work Completion/Completion Queue entry (Completion)



RDMA SEND & RDMA RECEIVE

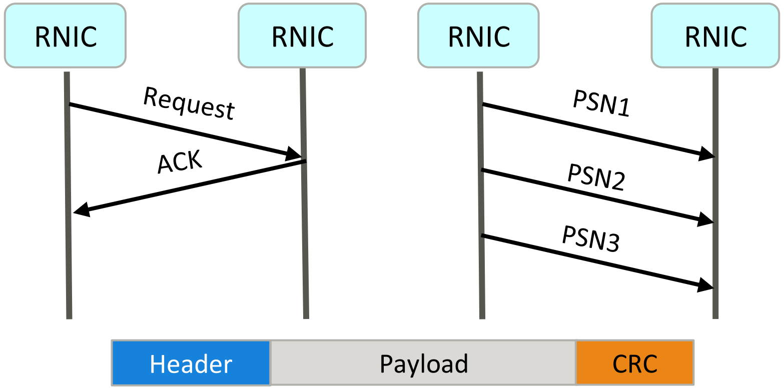


RDMA WRITE

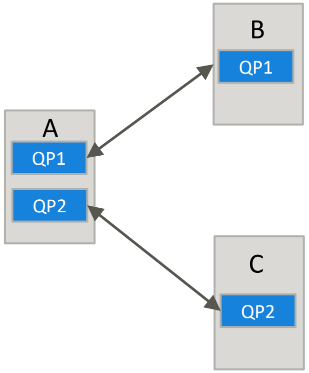
Services of RDMA

- Types of Service
  - Reliable Service
  - Unreliable Service
  - Connection oriented Service
  - Datagram Service

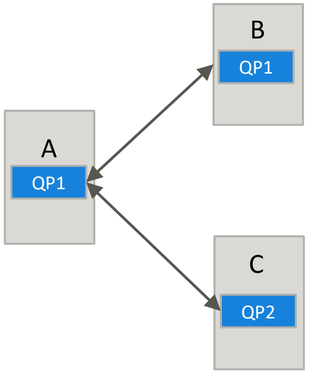
	Connection oriented	Datagram
Reliable	**Reliable Connection (RC)**	Reliable Datagram (RD)
Unreliable Service	Unreliable Connection (UC)	Unreliable Datagram (UD)



Reliable Service



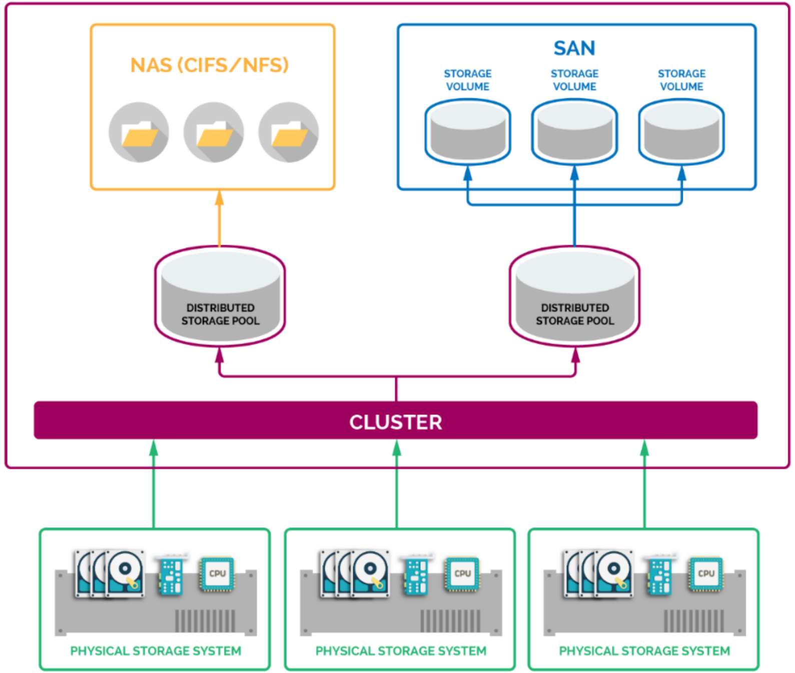
Connection oriented Service



Datagram Service

Motivation of combining RDMA with PMEM

- Use case
  - Distributed storage system
  - Distributed database
- Performance
  - Improve performance of IO by PMEM on local storage node
  - Improve the performance of network by RDMA between storage nodes



Distributed storage system[1]

[1] https://docs.bmc.com/docs/discovery/contentref/distributed-storage-model-concept-and-principles-997880678.html

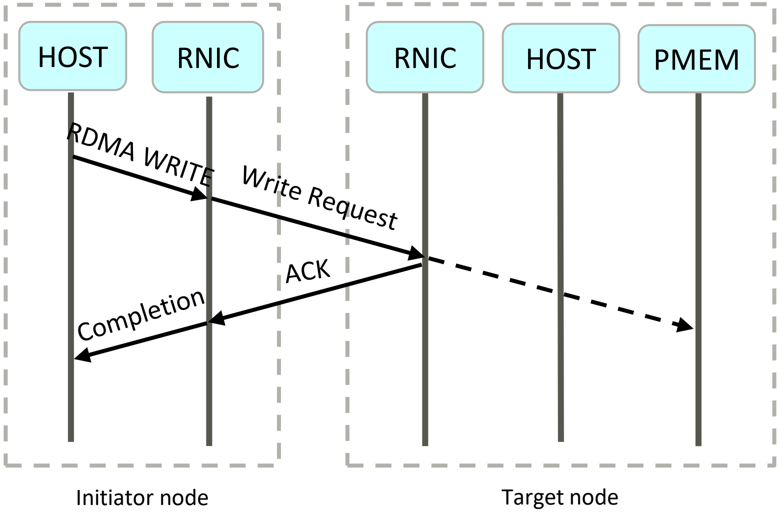
Shortcomings of combining RDMA with PMEM(1/2)

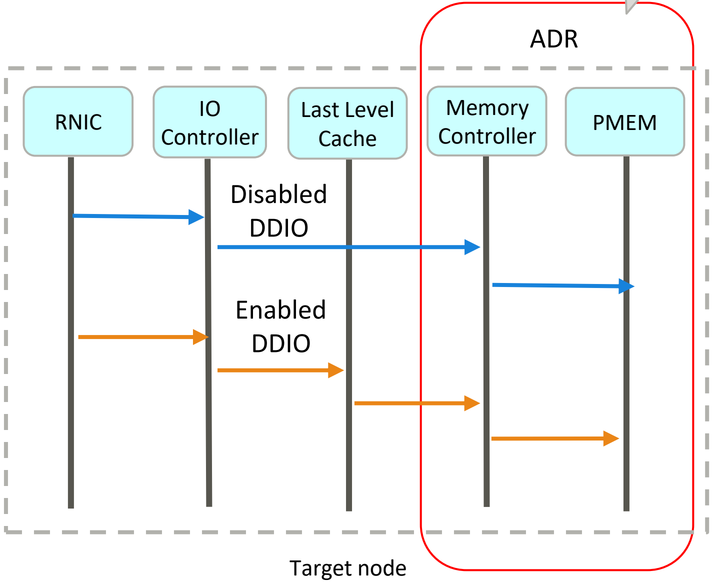
Data persistency

- RDMA WRITE operation can only ensure that the written data reaches the remote RNIC.
- Flushing the preceding written data into the remote PMEM is influenced by DDIO.

※ DDIO (Data Direct I/O Technology) makes NICs and controllers talk directly to the processor cache (LLC) without a detour via system memory.

Require FLUSH operation!



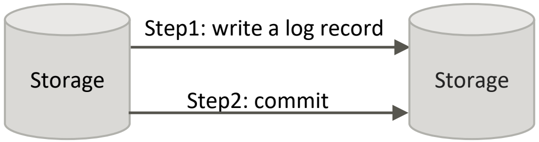


Shortcomings of combining RDMA with PMEM(2/2)

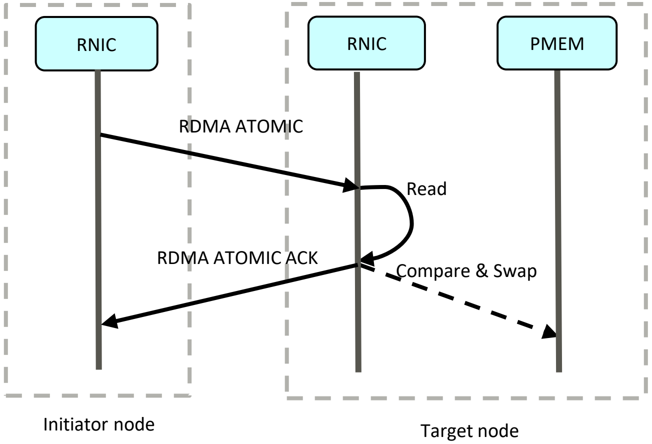
FUJITSU

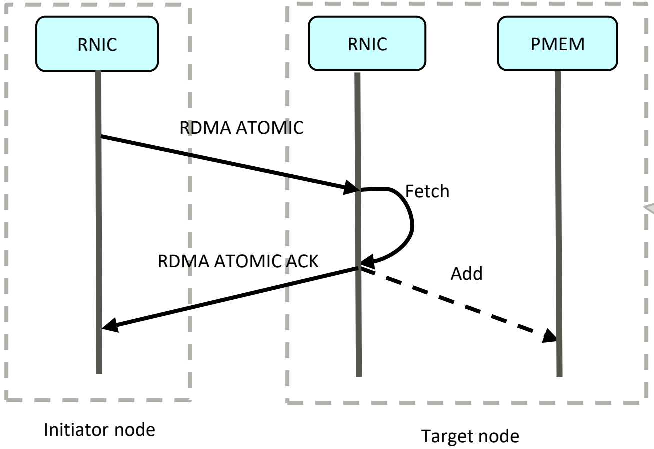
Data availability

- No dedicated ATOMIC WRITE operation
  - Two phase commit for databases or log-based filesystems
    - Step 1: Write a log record through multiple packages
    - Step 2: Mark the log record as valid by update a write pointer (8 bytes) atomically
- RDMA ATOMIC operation (Cmp/Swap or Fetch/Add) is too heavy



Two phase commit





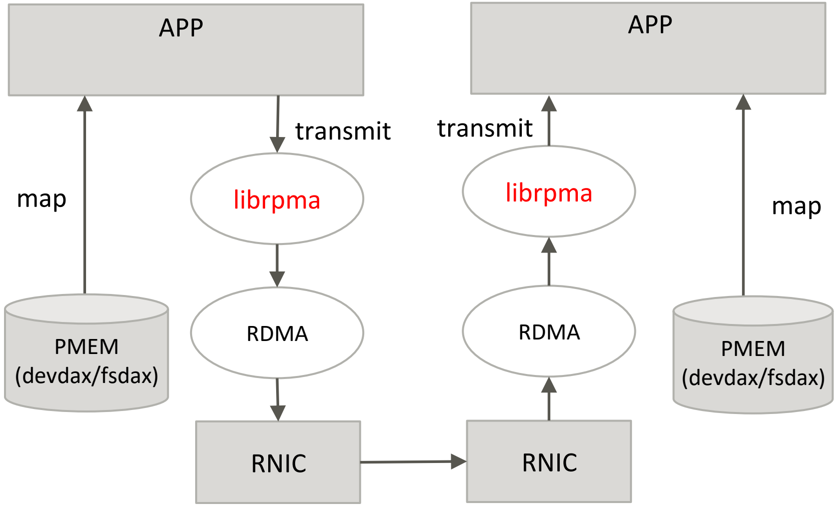
Require ATOMIC WRITE operation!

Remote Persistent Memory Access library

- Remote Persistent Memory Access library (librpma)[1] is designed to support accessing Remote PMEM over RDMA.

- Solve the shortcomings of RDMA with PMEM
  - FLUSH operation
  - ATOMIC WRITE operation
- Simplify the usage of RDMA

- Main contributors
  - Intel, Fujitsu



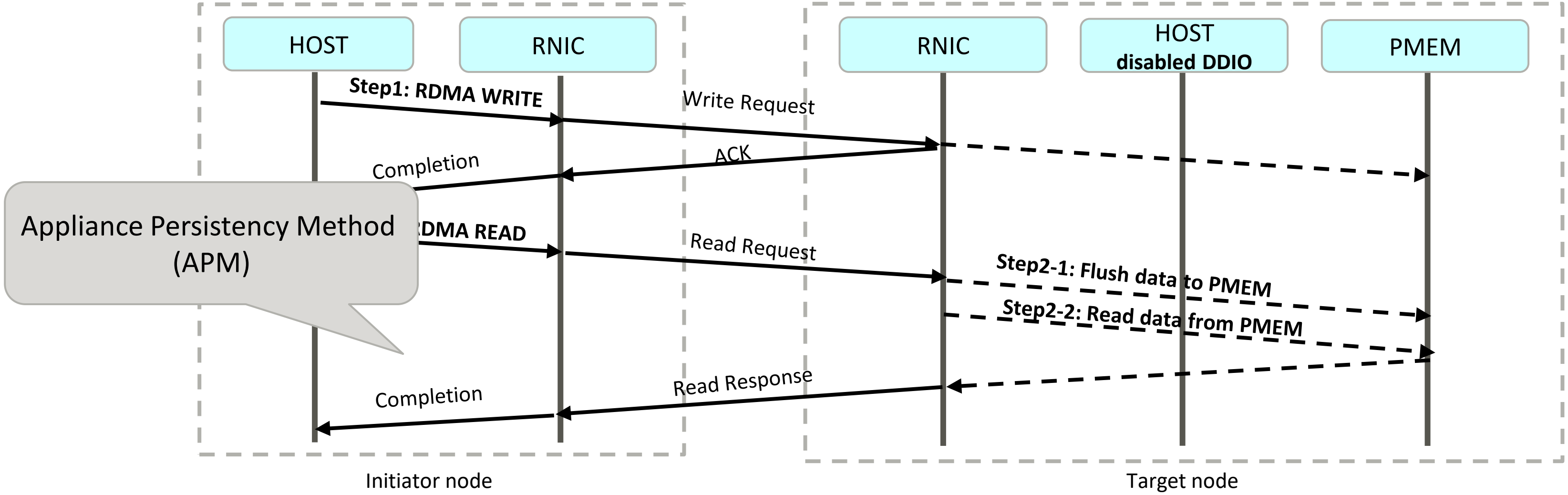
[1] https://github.com/pmem/rpma

Implement FLUSH operation on librpma(1/2)

FUJITSU

Implement FLUSH operation by RDMA READ

- Situation: DDIO is off that always skips LLC(Last Level Cache)
  - Step 1: Do a sequence of RDMA WRITE operations
  - Step 2: Do a following RDMA READ operation
    - Step 2-1: Flush all written data from RNIC to the remote PMEM
    - Step 2-2: Read data from the remote PMEM

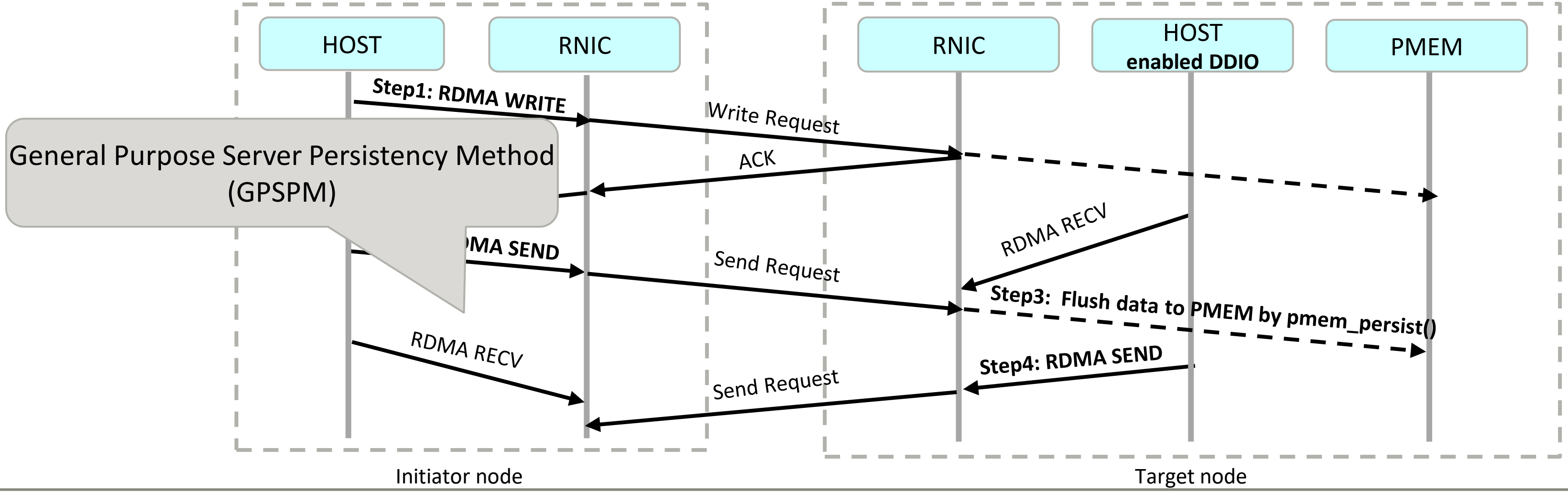


Appliance Persistency Method (APM)

Implement FLUSH operation on librpma(2/2)

Implement FLUSH operation by RDMA SEND and RECEIVE

- Situation: DDIO is on that always use LLC(Last Level Cache).
  - Step 1: Do a sequence of RDMA WRITE operations
  - Step 2: Do a following RDMA SEND operation
  - Step 3: On target node, flush all written data from LLC to PMEM according to the contents received
  - Step 4: Another RDMA SEND operation notifies the Initiator node that data has been written into PMEM

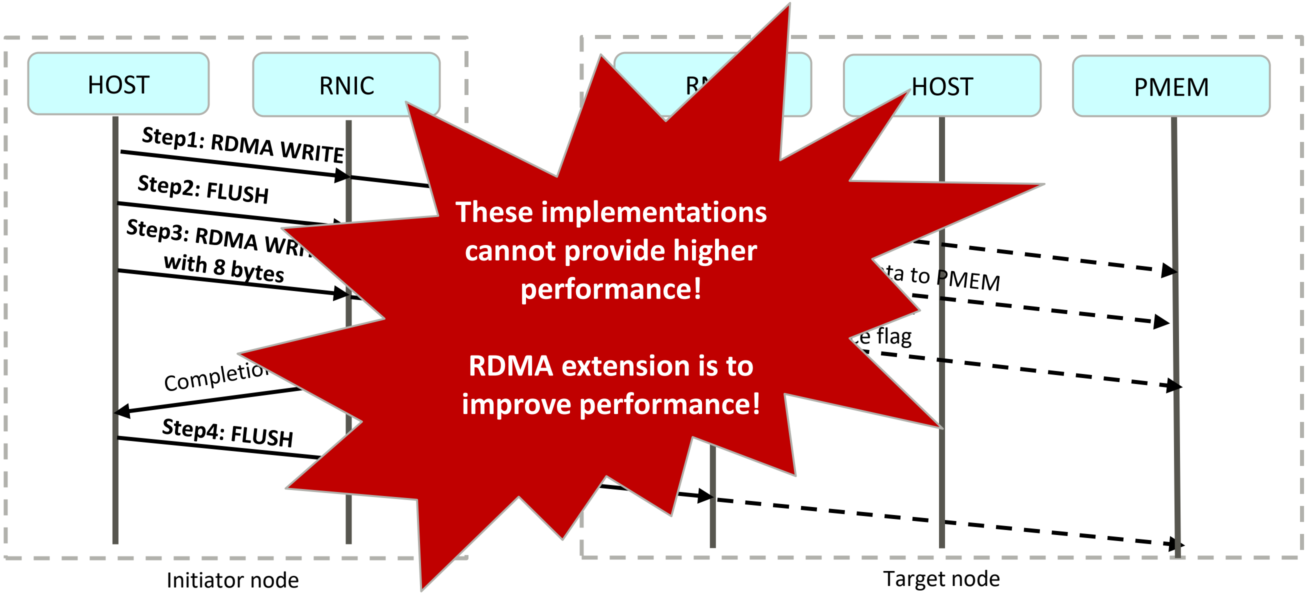


General Purpose Server Persistency Method (GPSPM)

Implement ATOMIC WRITE operation on librpma

Implement ATOMIC WRITE operation by RDMA WRITE with 8 bytes

- Step 1: Do a sequence of RDMA WRITE operations
- Step 2: Do a following FLUSH operation
- Step 3: Do a RDMA WRITE operation with 8 bytes to update the write pointer
- Step 4: Do a following FLUSH operation



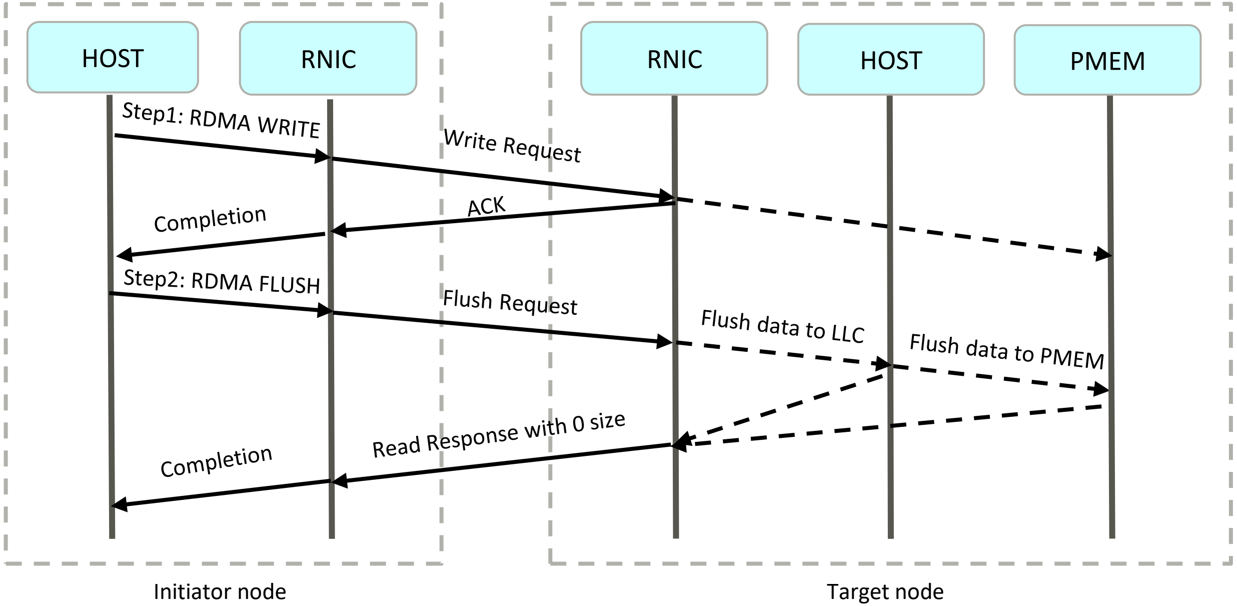
These implementations cannot provide higher performance!

RDMA extension is to improve performance!

A new RDMA FLUSH operation

FUJITSU

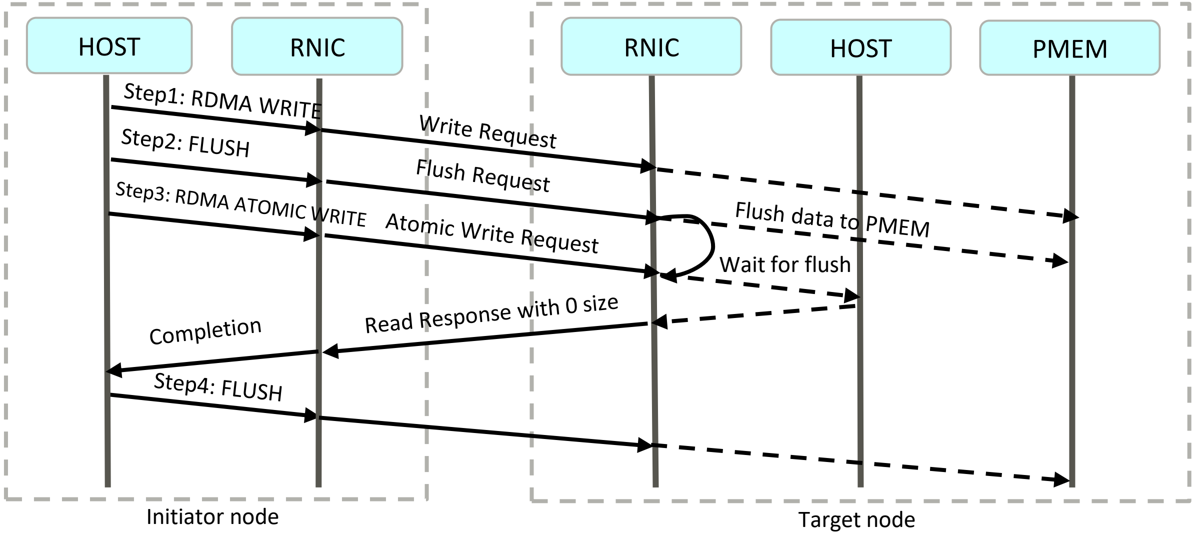
- Usage of RDMA FLUSH
  - Step1: Do a sequence of RDMA WRITE operations
  - Step2: Do a following RDMA FLUSH operation with two options(type and range)
- Type
  - Visibility
  - Persistence
- Range
  - A specified range
  - Entire memory region



A new RDMA ATOMIC WRITE operation

Usage of RDMA ATOMIC WRITE

- Step 1: Do a sequence of RDMA WRITE operations
- Step 2: Do a following FLUSH operation
- Step 3: Do a RDMA ATOMIC WRITE operation to update the write pointer
- Step 4: Do a following FLUSH operation

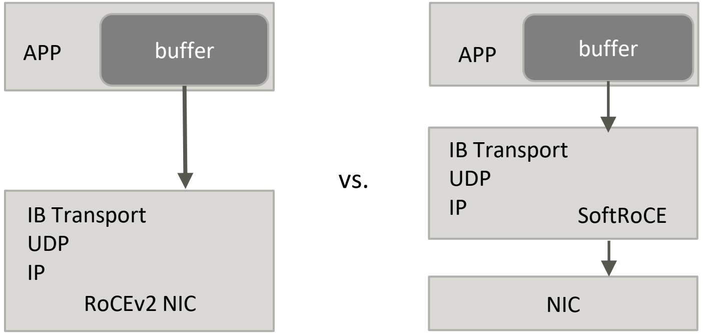


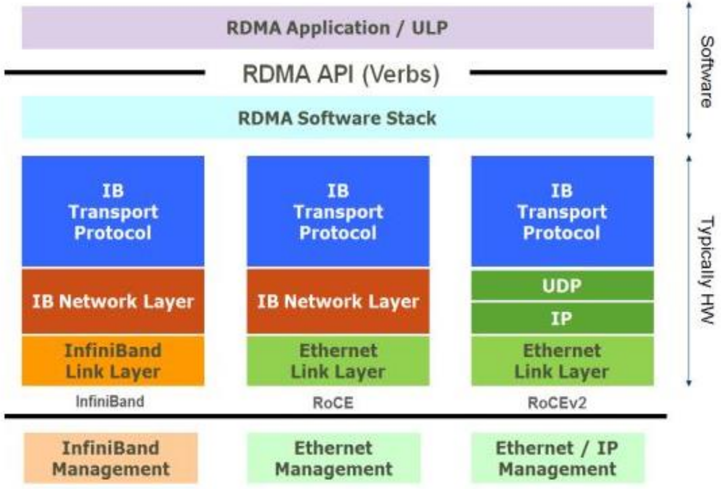
Network protocols supporting RDMA

- InfiniBand（IB）
- RDMA over Converged Ethernet (RoCEv1)
- IP ROUTABLE RDMA over Converged Ethernet (RoCEv2)
  - RoCEv2 packet format

Ethernet header	IP header	UDP header	IB Transport header	IB Payload	ICRC	FCS

  - Hardware RoCEv2 vs. Software-based RoCEv2 (SoftRoCE)



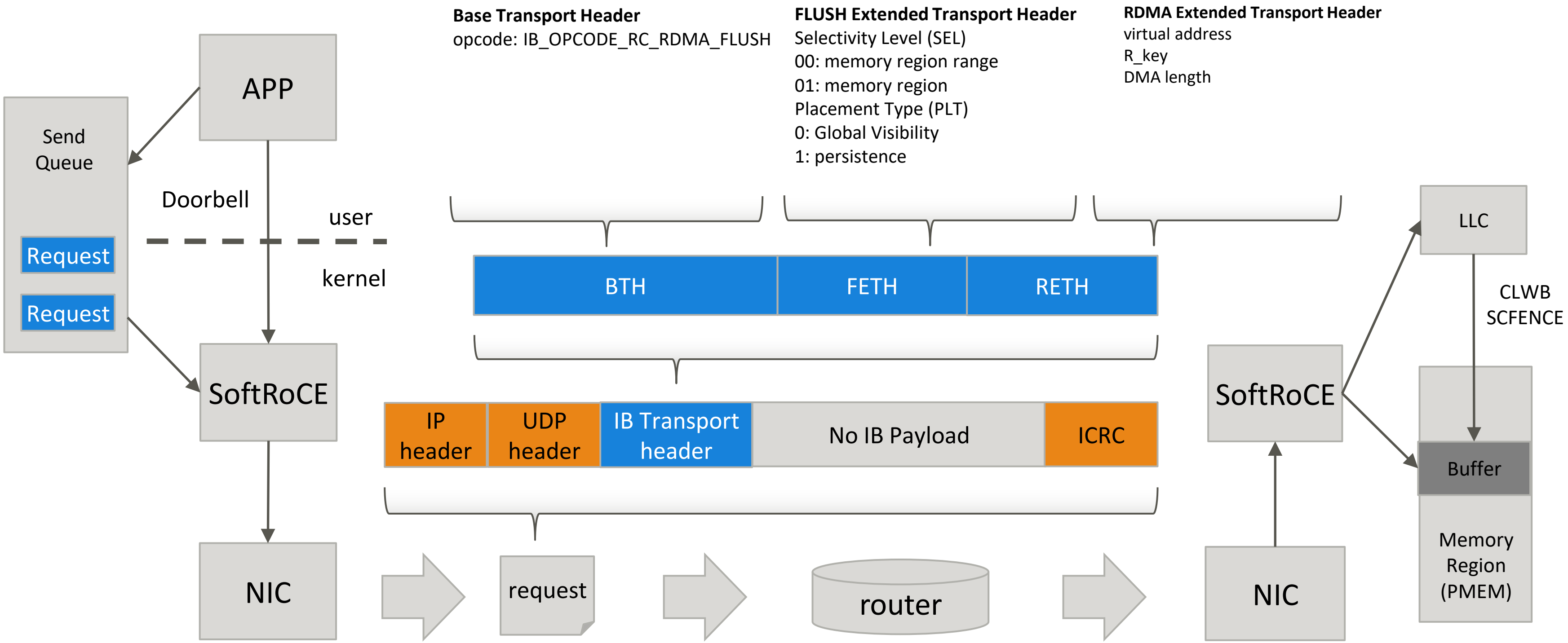


Protocol Stack[1]

[1] InfiniBand™ Architecture Specification Volume 1 Release 1.5, P1958

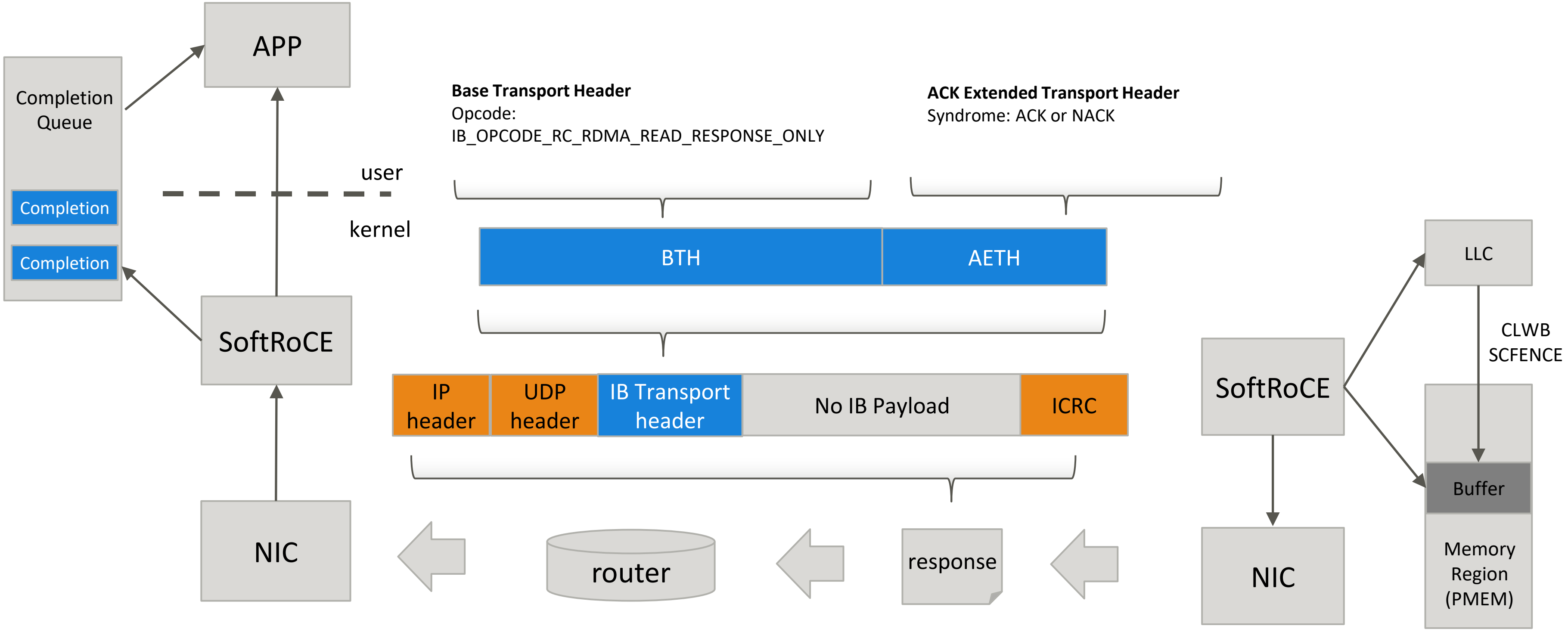
New RDMA FLUSH packages (1/2)

- Implement RDMA FLUSH Request over SoftRoCE (RC service)



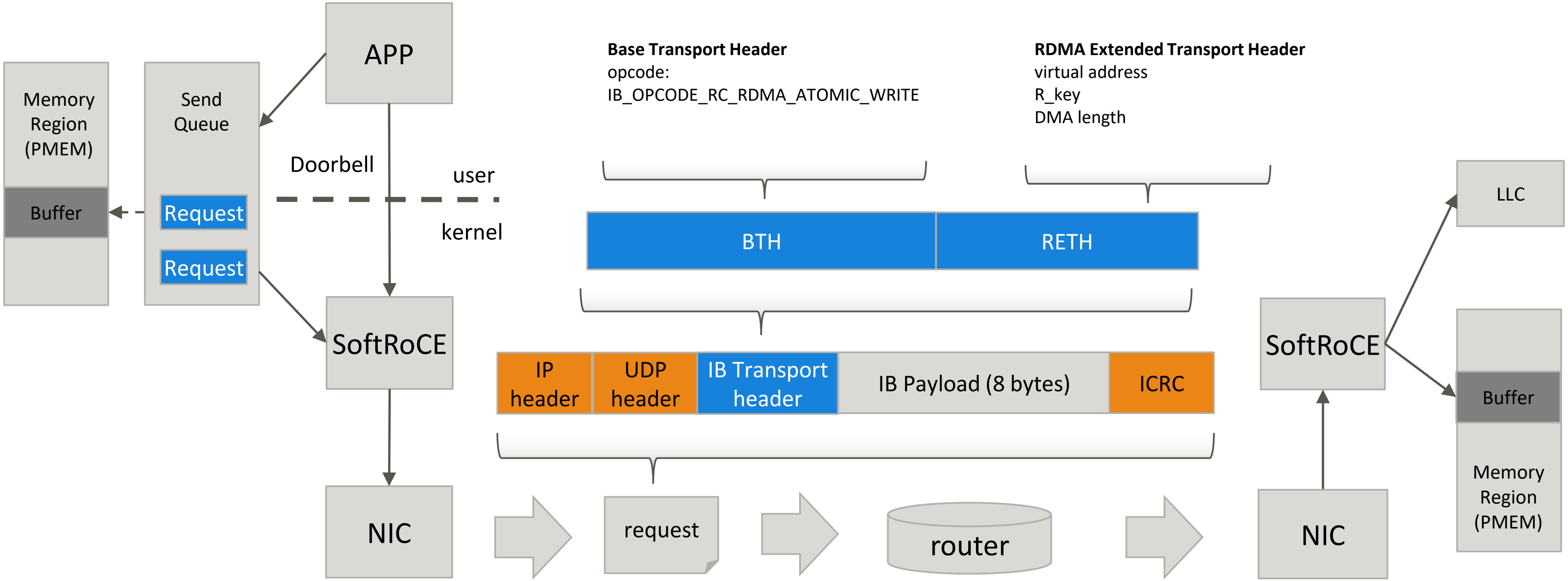
New RDMA FLUSH packages (2/2)

- Implement RDMA FLUSH Response over SoftRoCE (RC service)



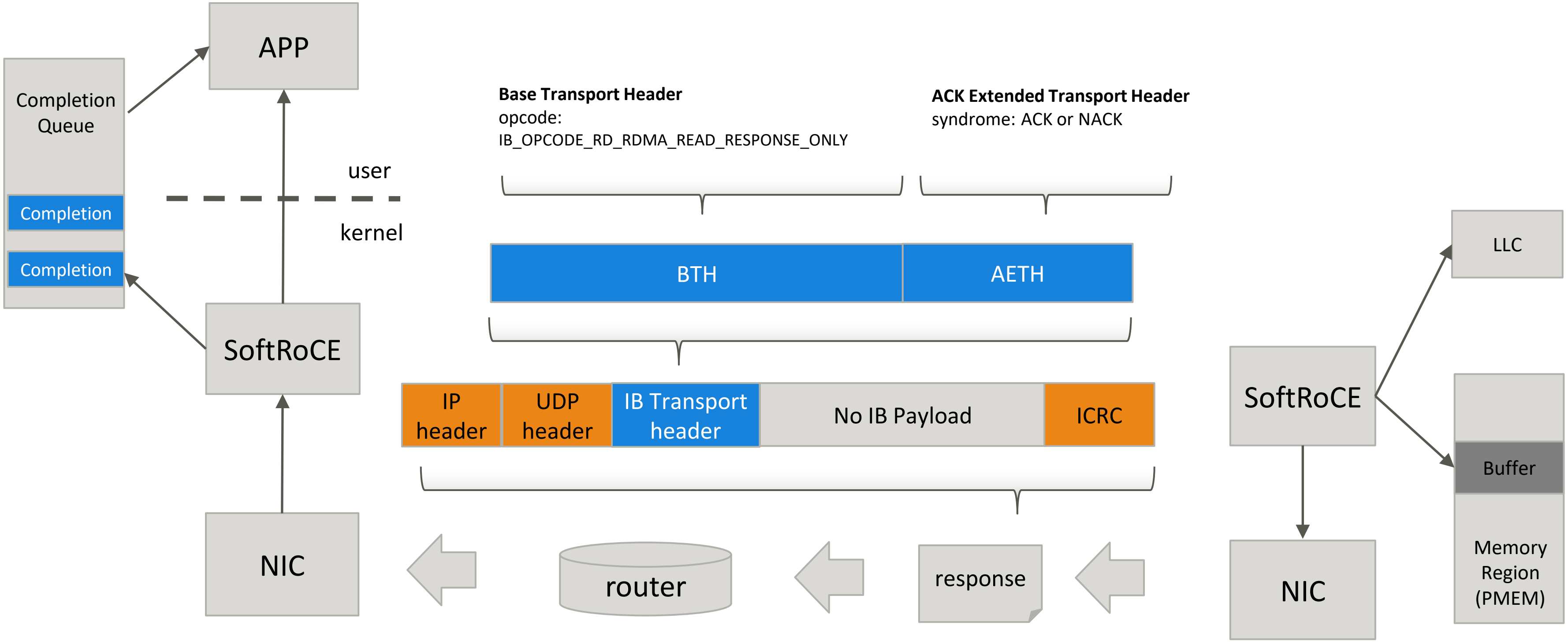
New RDMA ATOMIC WRITE packages (1/2)

- Implement RDMA ATOMIC WRITE Request over SoftRoCE (RC service)



New RDMA ATOMIC WRITE packages (2/2)

- Implement RDMA ATOMIC WRITE Response over SoftRoCE (RC service)



Conclusion and future work

FUJITSU

- Conclusion
  - Basis of PMEM and RDMA
  - Shortcomings of combining RDMA with PMEM
  - Basis of librpma
  - Implement RDMA FLUSH and RDMA ATOMIC WRITE over SoftRoCE

- Future work
  - Make librpma support RDMA FLUSH and RDMA ATOMIC WRITE
  - Implement RDMA VERIFY over SoftRoCE
  - Make librpma support RDMA VERIFY

FUJITSU

shaping tomorrow with you