InfiniBand/RoCE RDMA Specification Update

Overview of August 2021 updates to the IB Specification, including the new Memory Placement Extensions

Chet Douglas, Intel
Specification update overview

- Volume 1, Release 1.5, published August 6, 2021
- The specification defines InfiniBand and RoCE
- Available to IBTA Members

- 2038 pages
- 22 issues addressed
- 57 new sections/features added

- InfiniBand NDR speeds
- QoS and bandwidth enhancements
- Virtualization section updated
- Memory Placement Extensions
Support Enhanced Speeds

- 1.5 spec includes support for InfiniBand NDR speeds
- Including split support (2x)

<table>
<thead>
<tr>
<th>Link Speed</th>
<th>Lane Speed</th>
<th>Signaling</th>
<th>2X throughput</th>
<th>4X throughput</th>
<th>8X throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDR</td>
<td>25.78125 Gb/s</td>
<td>NRZ</td>
<td>51.5625 Gb/s</td>
<td>103.125 Gb/s</td>
<td>206.25 Gb/s</td>
</tr>
<tr>
<td>HDR</td>
<td>53.125 Gb/s</td>
<td>PAM4</td>
<td>106.25 Gb/s</td>
<td>212.5 Gb/s</td>
<td>425 Gb/s</td>
</tr>
<tr>
<td>NDR</td>
<td>106.25 Gb/s</td>
<td>PAM4</td>
<td>212.5 Gb/s</td>
<td>425 Gb/s</td>
<td>850 Gb/s</td>
</tr>
</tbody>
</table>

- RoCE follows the Ethernet Physical & Link Layer Standards
  - RoCE fully supports 400 Gb/s and future speeds
Minimum Bandwidth Added to Enhanced Port Arbiter

- New minimum bandwidth setting capabilities
- Allow setting a dynamic bandwidth increase
- Use cases:
  - Traffic that requires low latency and low jitter.
    - Management
    - Time sync
    - Heartbeat protocol
Virtual Ports QoS

- Introduce bandwidth sharing and rate limiting configuration for \{Vport,SL\} on top of Standard or Enhanced InfiniBand port arbiter.
Memory Placement Extensions

“MPE”
Memory Placement Extensions

**Motivation**

- Performance demonstration compared round trip latency for RDMA to an SSD vs RDMA to PMEM
- Because the PMEM is byte addressable and is attached to the memory subsystem it is possible to transfer data to the final persistence domain via RDMA, with minimal CPU involvement

![RDMA Benefits with PMEM](image-url)
Memory Placement Extensions

Motivation/problem statement

- With the advent of PMEM, storage is now connected to the memory subsystem and can be directly accessed using RDMA.
- However, existing RDMA did not provide RDMA Write or Atomics reliability guarantees
  - RDMA Write completion does not guarantee data has reached remote host memory.
- With pre-1.5 specification, any guarantees that the data has reached the persistence domain must be implemented by a ULP. This requires interrupts, pipeline stalls, adding additional latency to the transaction, and may not be scalable.
Memory Placement Extensions
Motivation/problem statement details

- RDMA Acknowledge (and Completion)
  - Guarantee only that Data has been successfully received and accepted for execution by the remote HCA
  - Doesn’t guarantee data has reached remote host memory
- Further guarantees are out of the scope of the standard
- There are platform specific ways to implement further guarantees
  - E.g. messaging + SW cache flushing
  - E.g. RDMA READ
Memory Placement Extensions

- Two new command opcodes in the 1.5 specification
  - Flush
    - Flush all previous writes or specific regions, per QP
    - Provides acknowledgement that volatile writes have made it to Global Observability
    - Provides acknowledgement that persistent memory writes have made it to the power fail safe persistence domain
    - Pipelined operation
  - Atomic Write
    - Writes an aligned 8-byte value atomically
    - Provides guarantees for remote pointer updates to persistent or volatile memory
- Fully supported by InfiniBand and RoCE
Memory Placement Extensions
Synchronous Log Writing with MPE

- Synchronous log writing: Requester Host application wishes to send an 8 byte pointer update to PMEM only if the log data (write 1) was written to PMEM first
- Requester Host application issues a FLUSH operation after RDMA Write 1 to force the Responder HCA to flush the writes before completing the FLUSH response
- Without waiting for the FLUSH completion, the Requester Host application can queue an ATOMIC WRITE operation to update the 8 byte pointer
- Ordering rules prevent the Responder HCA from executing the ATOMIC WRITE until the previous FLUSH has completed
- This allows the Requester Host application to queue the RDMA WRITE, FLUSH, ATOMIC WRITE, in a pipelined manner, without waiting for this to be done by an ULP, and without stalling the pipeline
Memory Placement Extensions
Scenarios where MPE can improve performance

- **Synchronous Replication**
  - Multiple physical copies of all data are replicated on several systems before the original data is considered Highly Available (HA)
  - High priority network latency sensitive scenario for datacenters adopting PMEM
  - Advantageous to perform consistency checks on replica data in-line, pipelined with the remotes writes & flushes
Memory Placement Extensions
Scenarios where MPE can improve performance

- **Synchronous Log Writing**
  - Shared data is distributed amongst multiple systems
  - Log Files keep track of transactions applied to the data
  - High priority network latency sensitive workload for SQL adoption of PMEM
  - Advantageous to perform consistency checks on log data in-line, pipelined with the remote writes & flushes
MPE Details

FLUSH and ATOMIC WRITE
Memory Placement Extensions

- MPE Initialization

Allocate persistent memory buffer on Responder

Registering memory with Responder HCA:
- Address
- Length
- PMEM – Used for GV/PMEM FLUSH selector
- FLUSHABLE – Indicates FLUSH of the region is allowed

Handle for memory registration made available to Requester application
Memory Placement Extensions

- **FLUSH to Global Visibility (GV)**

![Diagram showing the process of FLUSH to Global Visibility (GV) with Requester and Responder.]
Memory Placement Extensions

- **FLUSH to Persistence (PMEM)**

![Diagram of Memory Placement Extensions]

- Application to DRAM to HCA to PMEM
- Writing data to Responder persistent memory
- Requester to Responder
- TPT: Handle
  - PMEM, FLUSHABLE
- Flushing previously written data to Responder persistent memory
Memory Placement Extensions

- Pipelined Log Writing Example

![Diagram of Memory Placement Extensions]

- Writing data to Responder persistent memory
- Flushing previously written data to Responder persistent memory
- Updating Log pointer in Responder persistent memory
Next steps

- Future version of spec considering:
  - Opcode extension to support additional operations
  - VERIFY operation to provide remote integrity checks
  - Continued integration of extensions into base document text
For more information

https://www.infinibandta.org/ibta-specification/

- **RDMA vendors:**
  - Implement MPE in your InfiniBand and RoCE adapter(s)

- **RDMA users:**
  - Enhance your application(s) and ULP(s) to leverage MPE
Please take a moment to rate this session.

Your feedback is important to us.