Proposed MPI Library Enhancements for Improving Latency and Rate for Small Messages

Hemal Shah, Distinguished Engineer and Architect
Broadcom Inc.

Date: August 22, 2023
MPI Message Characterization – where small messages are used?

- Small messages – focus of this talk
  - Synchronization
  - PGAS
  - Control packets
  - Collectives
  - …

- Medium (1KB to 10s-100s of KB)

- Large (> 100s KB)
Communication Overhead for Small RoCE Messages

As RoCE scales from 200G→400G→800G→1.6T, how do we improve MPI layer?

- MPI layer adds significant latency if not optimized
- CPU overhead per MPI message may bound message rate

- Optimized verbs library adds a few 100s of ns of the latency
- PCIe latency a few 100s ns
- Hardware RoCE transport, QoS, and congestion control
  - sub 1us latency
  - 100s of Million msg/sec
- Cable/optics latency 100-200 ns
- Multi-Stage Ethernet Fabric latency ~ few 100 ns
Latency and Message Rate Considerations

• CPU cycles spent per message
• Scaling of message rate with number of CPU cores
Copy Optimization

- Small MPI send messages may incur multiple copies

- Small MPI recv message incur at least one CPU copy

- Potential MPI library enhancements
  - Use of vector mode instructions for copies → improves latency and message rate
  - MPI buffer pool per core (pinned in core cache) – reduces CPU overhead for copies
MPI Message Coalescing Enhancements

• Coalescing of MPI messages into a single RoCE packet
  – Improves MPI msg rate as RoCE pkt overhead is amortized
  – Adds complexity at the MPI layer
  – Impacts MPI message latency

• Posting of MPI messages as a list of WRs in a single verbs call
  – Amortizes the cost of verbs layer processing
  – Reduces doorbell rate → reduces MMIO overhead → improves MPI message rate
  – Impacts MPI message latency

• MPI libs typically have env variables to control coalescing (all or nothing)

• Enhancement: selectively bypass coalescing for latency sensitive messages
  – Example 1: MPI_SendRecv → expected to be latency sensitive
  – Example 2: Small messages of low latency Class of Service (CoS)
**MPI Buffer Management**

- **MPI buffer pinning and caching**
  - MPI buffer pool per core (pinned in cache) → reduces CPU overhead for copies

- **Shared ownership of buffers between MPI and verbs layers**
  - Avoids intermediate copies
  - Buffer ownership is transferred during calls and completions
MPI Library Hints

• MPI lib hints for pending WRs (to be posted) on the QPs
  – Helps with moderating doorbell rates

• MPI lib hints for low latency/high message rate QPs
  – Allows verbs library to optimize WRs processing per QP for low latency and/or message rate

• Use of thread domain verbs
  – Provides hint to verbs library that access to resources within a thread domain is thread safe
  – Helps in avoiding internal locking in the verbs library
  – ibv_alloc_td: a thread safe alternative to ibv_alloc_pd (thread unsafe)
Use of Multiple Class of Service Queues (CoSQs)

- MPI traffic separation in different CoSQs
  - Small Send/Recv (latency sensitive)
  - Small Send/Recv (high message rate)
  - RDMA Read/Write (throughput)
  - Small message-oriented collectives e.g. AllReduce OR operation on 1-byte
  - ...

- Use of separate CoSQs for low latency and high-rate small message traffic
  - Enables different message processing policies at MPI layer
  - Allow MPI layer to provide verbs layer hints for WRs and QPs
Summary

- MPI layer overheads impact small message latency and message rate
- MPI enhancements improve message latency and message rate scaling
- Copy optimization, selective coalescing, buffer caching, hints, CoSQs help

- Call to action: Investigate propose MPI lib enhancements