Design Guidelines for High Performance RDMA Systems

Anuj Kalia (CMU)
Michael Kaminsky (Intel Labs)
David Andersen (CMU)
RDMA is cheap (and fast!)

Mellanox Connect-IB
- 2x 56 Gbps InfiniBand
- ~2 µs RTT
- RDMA
- $1300

Problem
Performance depends on complex low-level factors
Background: RDMA read

- **RDMA read request**
- **RDMA read response**
- **DMA read**
- **PCI Express**
- **L3**
- **CPU**
- **NIC**
How to design a sequencer?
Which RDMA ops to use?

Remote CPU bypass (one-sided)
- Read
- Write
- **Fetch-and-add**
- Compare-and-swap

Remote CPU involved (messaging, two-sided)
- Send
-Recv

Perf? 2.2 M/s
How we sped up the sequencer by 50X
Large RDMA design space

Operations

- READ
- WRITE
- ATOMIC
- SEND, RECV

Remote bypass (one-sided)

Two-sided

Transports

- Reliable
- Unreliable
- Connected
- Datagram

Optimizations

- Inlined
- Unsigned
- Doorbell batching
- WQE shrinking
- 0B-RECVs
## Guidelines

NICs have multiple processing units (PUs)
- Avoid contention
- Exploit parallelism

PCI Express messages are expensive
- Reduce CPU-to-NIC messages (MMIOs)
- Reduce NIC-to-CPU messages (DMAs)
High contention w/ atomics

Fetch&Add(A, 1)

Latency ~500ns
Throughput ~2 M/s
Reduce contention: use CPU cores

[HERD, SIGCOMM 14]
Sequencer throughput

Throughput (M/s)

- Atomics: 2.2
- RPC (1 core): 7

50x
Reduce MMIOs w/ Doorbell batching

Push

CPU

SEND

SEND

NIC

MMIOs $\Rightarrow$ lots of CPU cycles

Pull

CPU

SEND

SEND

NIC

DMA
RPCs w/ Doorbell batching

Push

CPU → NIC: Requests → Responses

Pull (Doorbell batching)

CPU → NIC: Requests

CPU → NIC: Responses
Sequencer throughput

Throughput (M/s)

- Atomics: 2.2
- RPC (1 C): 7
- +Dbell batching: 16.6

50x
Exploit NIC parallelism w/ multiQ

SEND (RPC resp)
Sequencer throughput

Throughput (M/s)

- 2.2 (Atomics)
- 7 (RPC (1 C))
- 16.6 (+Dbell batching)
- 27.4 (+3 queues)

50x

16
Sequencer throughput

Bottleneck = PCIe DMA bandwidth (paper)
Reduce DMA size: Header-only

CPU

NIC

SEND

Header

Imm

Size

Data

Unused

64B

4B

8B

52B

0

64

128

Move payload

0

64
Sequencer throughput

Throughput (M/s)

<table>
<thead>
<tr>
<th>Atomics</th>
<th>RPC (1 C)</th>
<th>+4 Queues, Dbell batching</th>
<th>+6 cores</th>
<th>+Header-only</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>7</td>
<td>27.4</td>
<td>97.2</td>
<td>122</td>
</tr>
</tbody>
</table>

50x
Evaluation

• Evaluation of optimizations on 3 RDMA generations

• PCIe models, bottlenecks

• More atomics experiments
  • Example: atomic operations on multiple addresses
RPC-based key-value store

Throughput (M/s)

Number of cores

Baseline

+Doorbell Batching

HERD [SIGCOMM 14]

16B keys, 32B values, 5% PUTs

14 resps/doorbell

9 resps/doorbell
Conclusion

NICs have multiple processing units (PUs)

Avoid contention
Exploit parallelism

PCI Express messages are expensive

Reduce CPU-to-NIC messages (MMIOs)
Reduce NIC-to-CPU messages (DMAs)

Code: https://github.com/anujkaliaiitd/rdma_bench