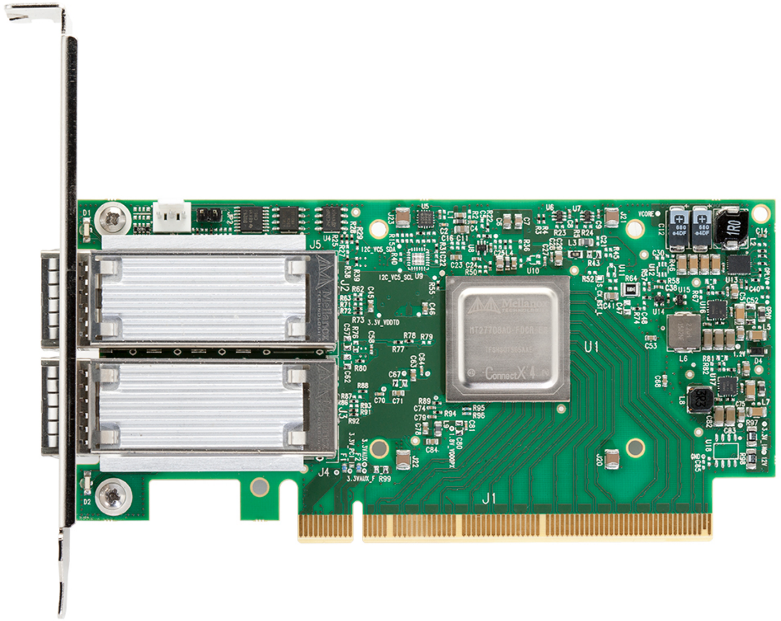
Design Guidelines
for
High Performance RDMA Systems

Anuj Kalia (CMU)
Michael Kaminsky (Intel Labs)
David Andersen (CMU)

RDMA is cheap (and fast!)



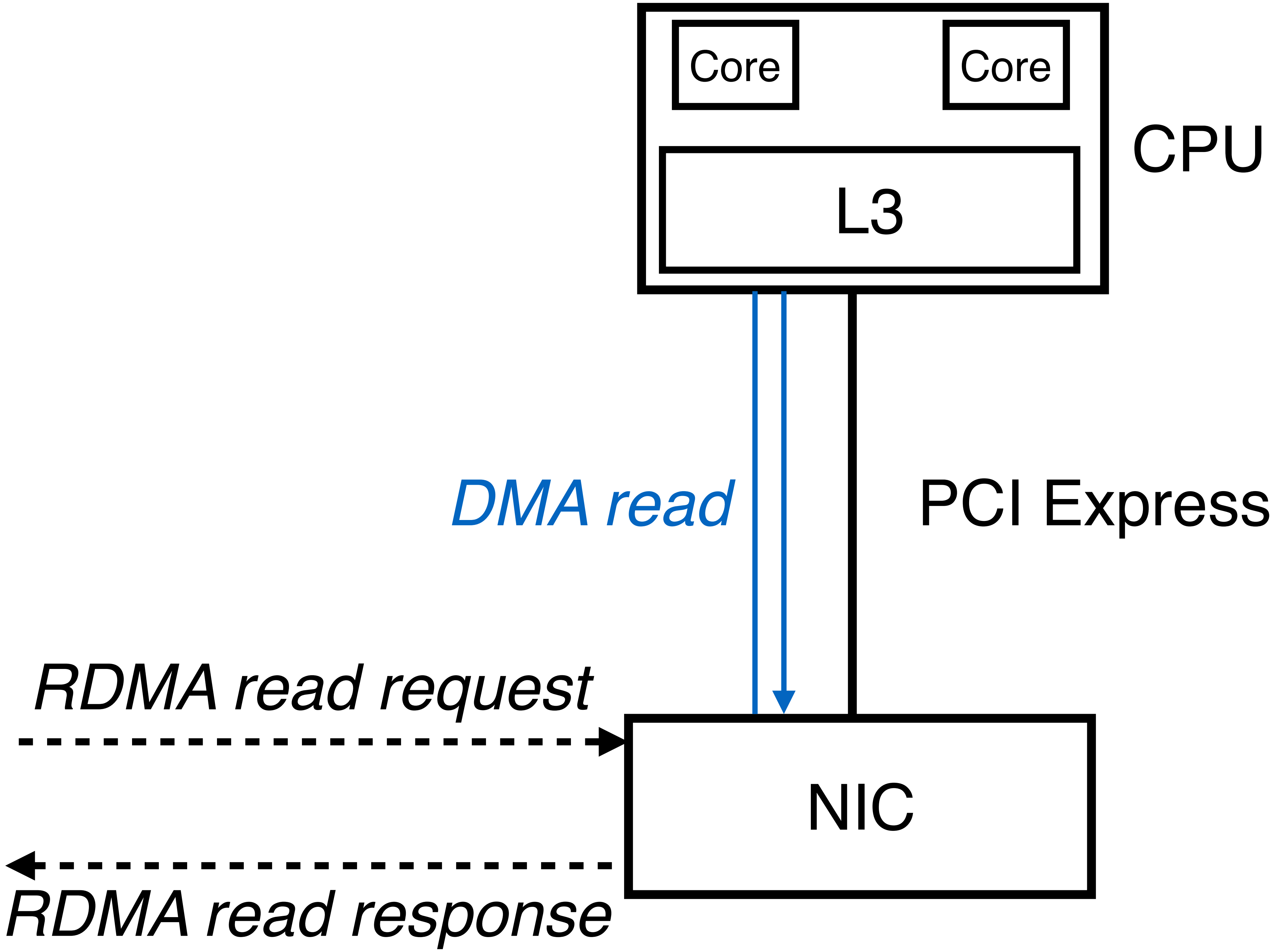
Mellanox Connect-IB

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

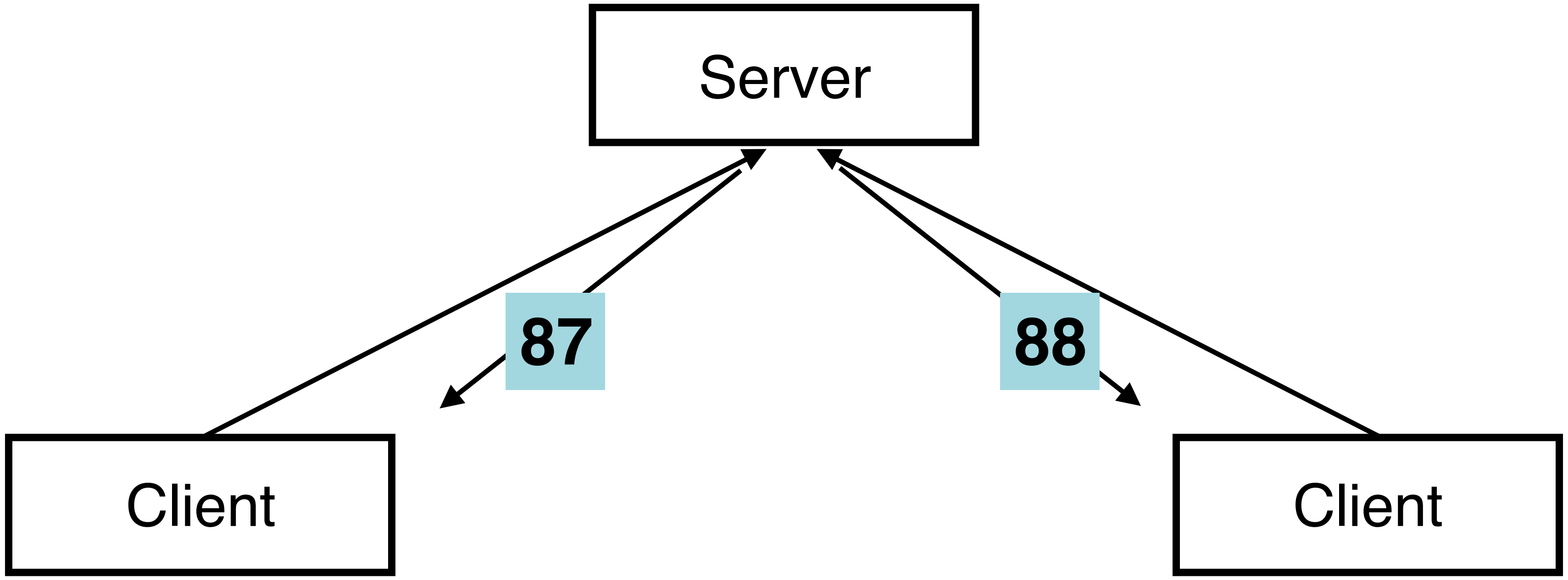
Problem

Performance depends on complex low-level factors

Background: RDMA read



How to design a sequencer?



Which RDMA ops to use?

Remote CPU bypass (one-sided)

- Read
- Write
- Fetch-and-add ☺ Perf? → 2.2 M/s ☹
- Compare-and-swap

Remote CPU involved (messaging, two-sided)

- Send
- Recv

How we sped up the sequencer

by 50X

Large RDMA design space

Operations

READ | WRITE | ATOMIC

Remote bypass (one-sided)

SEND, RECV

Two-sided

Transports

Reliable | Unreliable | Connected | Datagram

Optimizations

Inlined | Unsignaled | Doorbell batching

WQE shrinking | 0B-RECVs

Guidelines

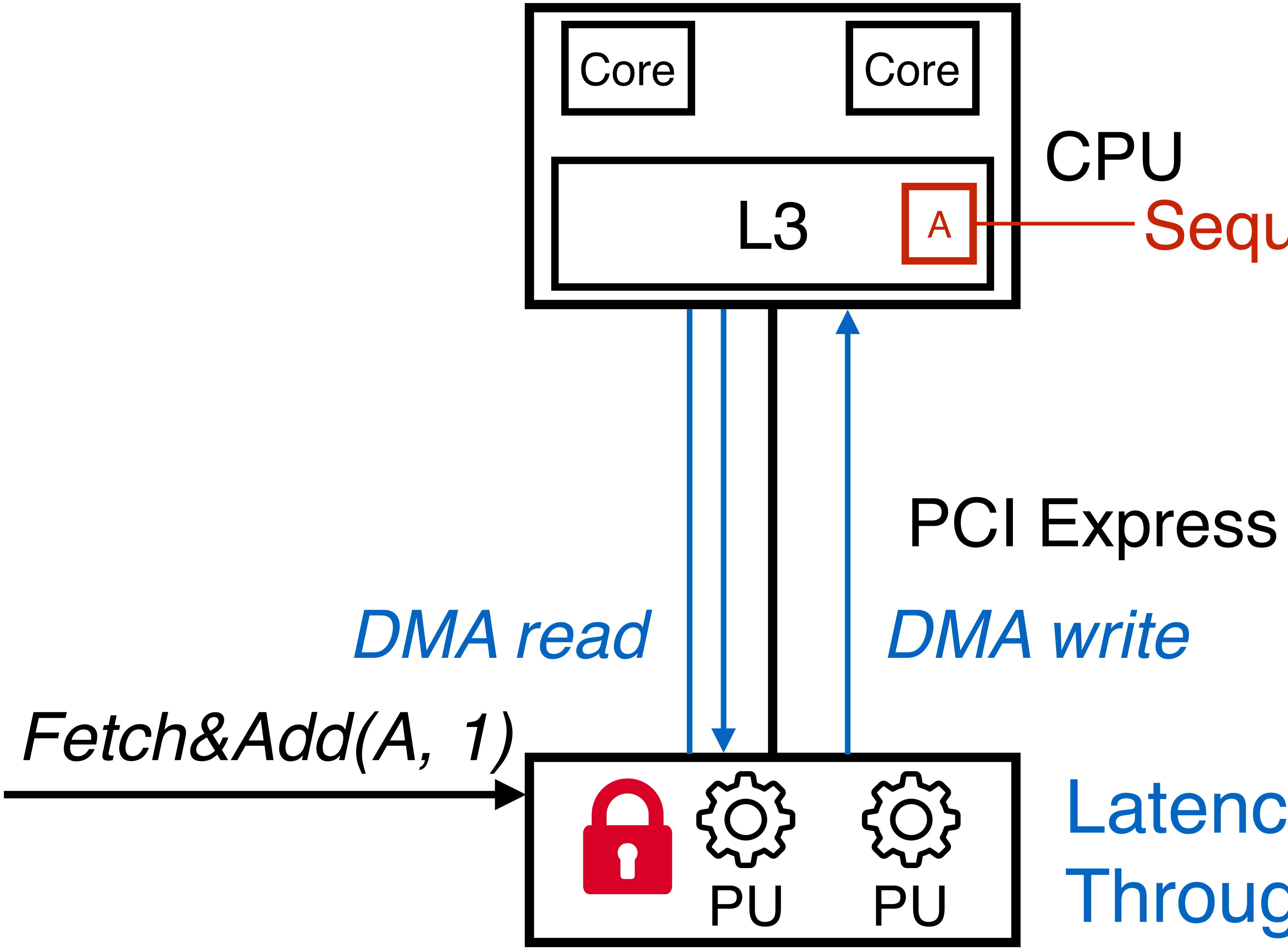
NICs have multiple processing units (PUs)

Avoid contention
Exploit parallelism

PCI Express messages are expensive

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

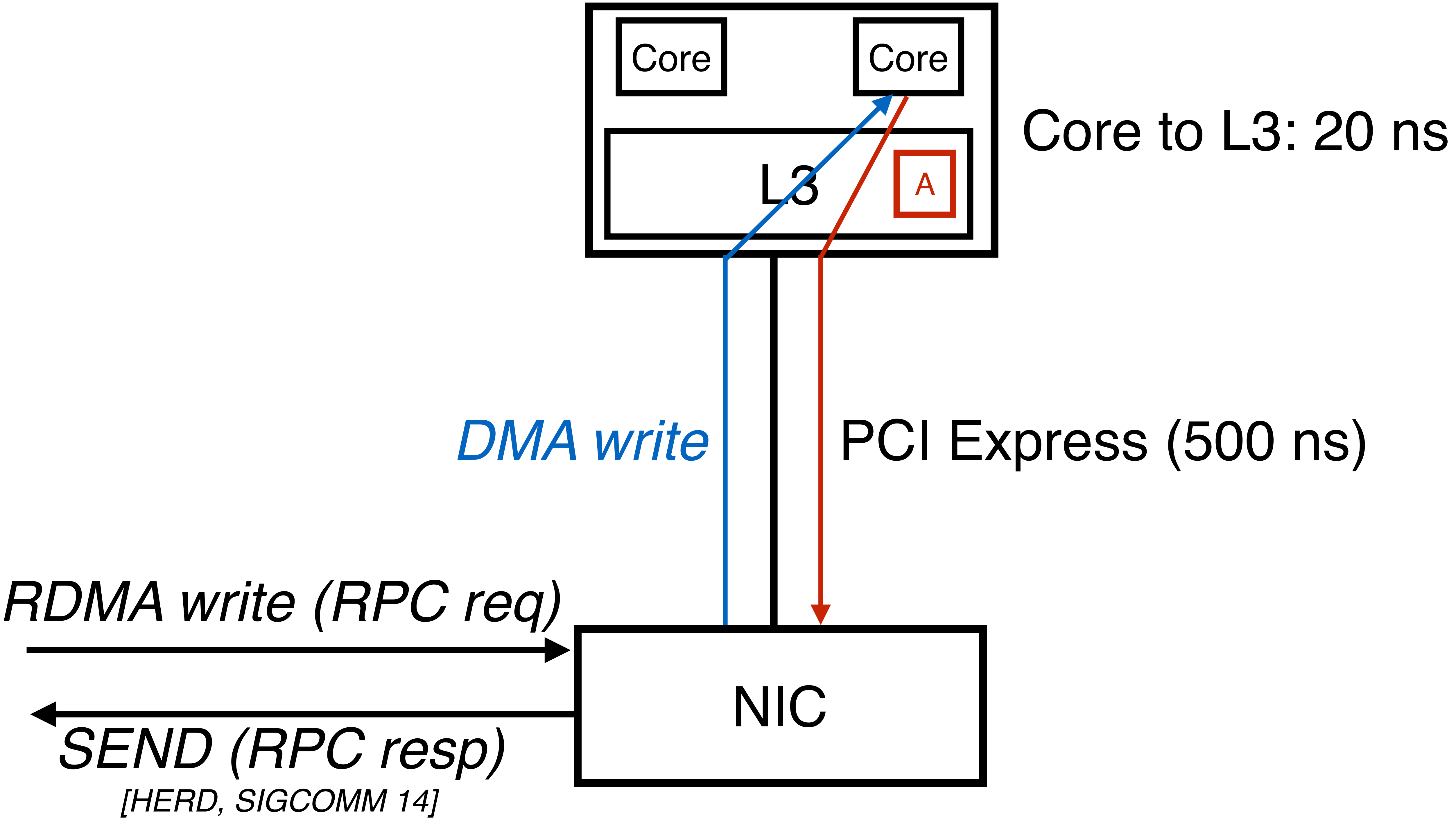
High contention w/ atomics



Sequence counter

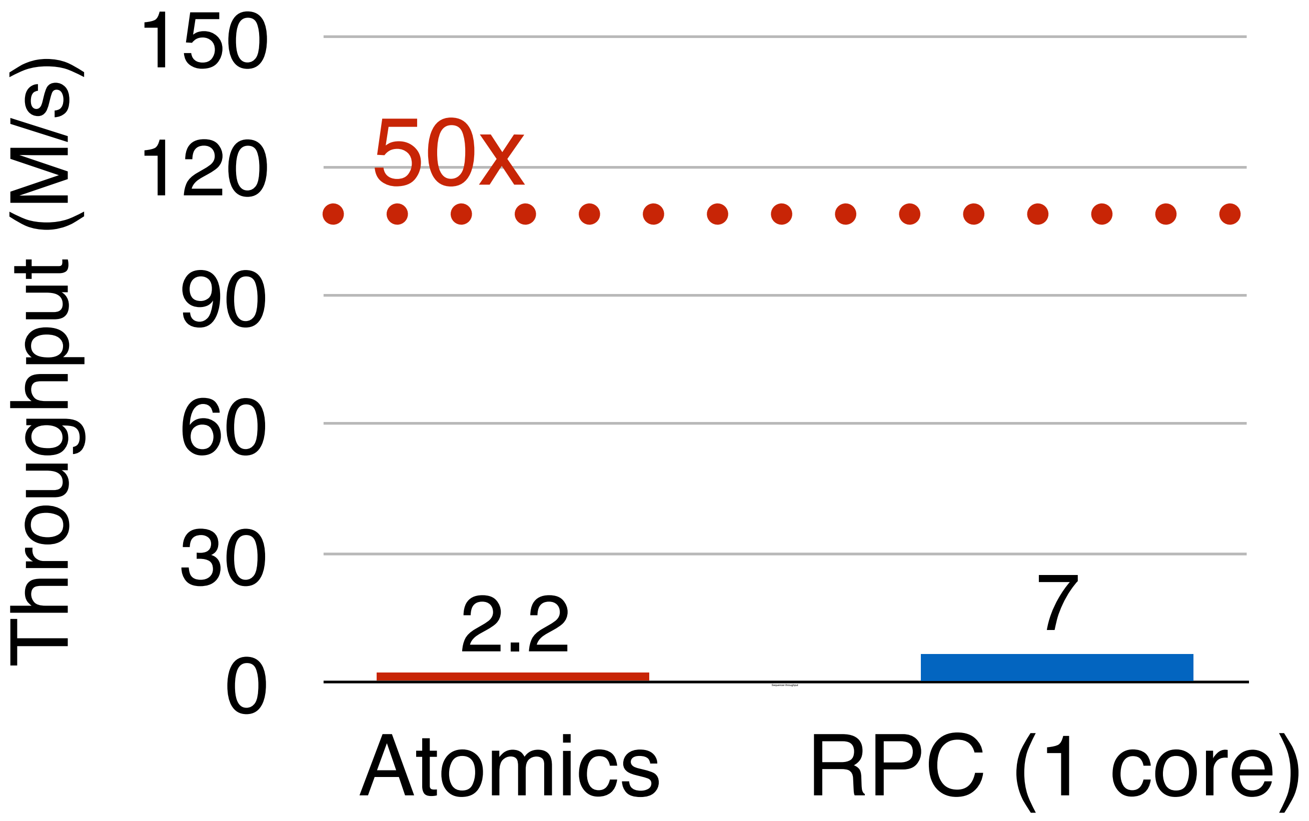
Latency ~500ns
Throughput ~2 M/s

Reduce contention: use CPU cores



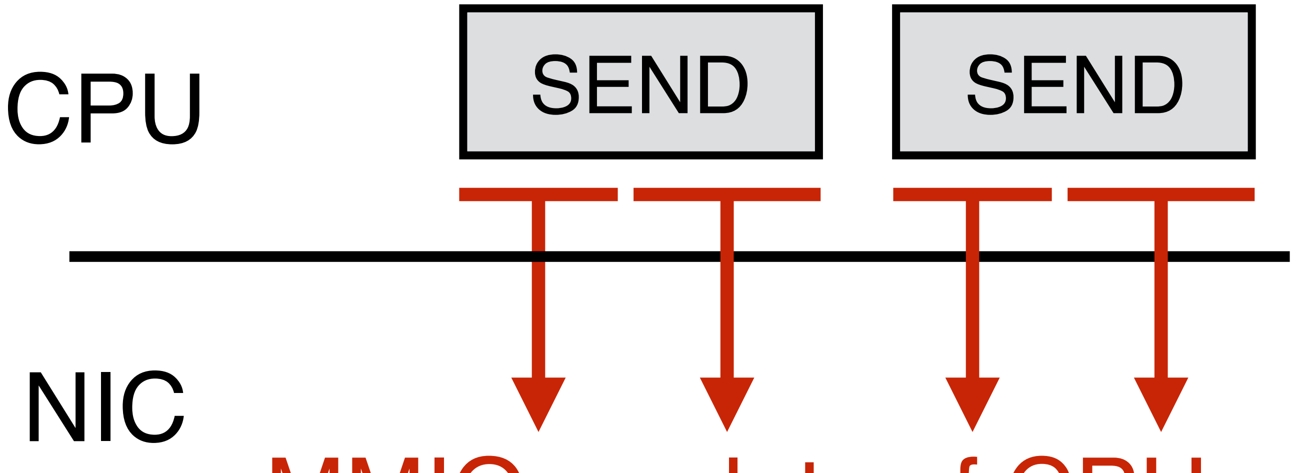
[HERD, SIGCOMM 14]

Sequencer throughput



Reduce MMIOs w/ Doorbell batching

Push



CPU

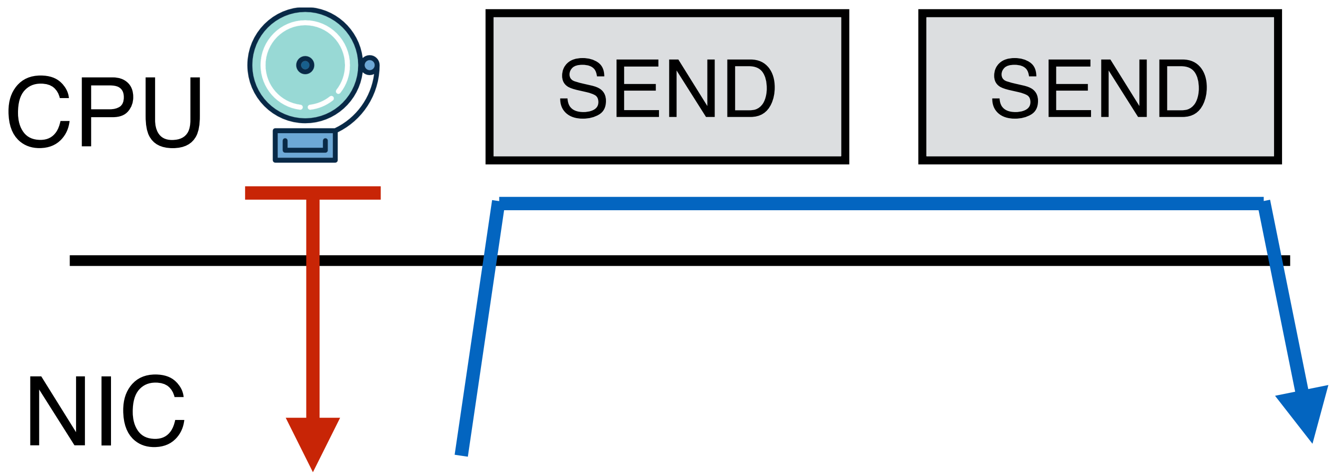
SEND

SEND

NIC

MMIOs ⇒ lots of CPU cycles

Pull



CPU

SEND

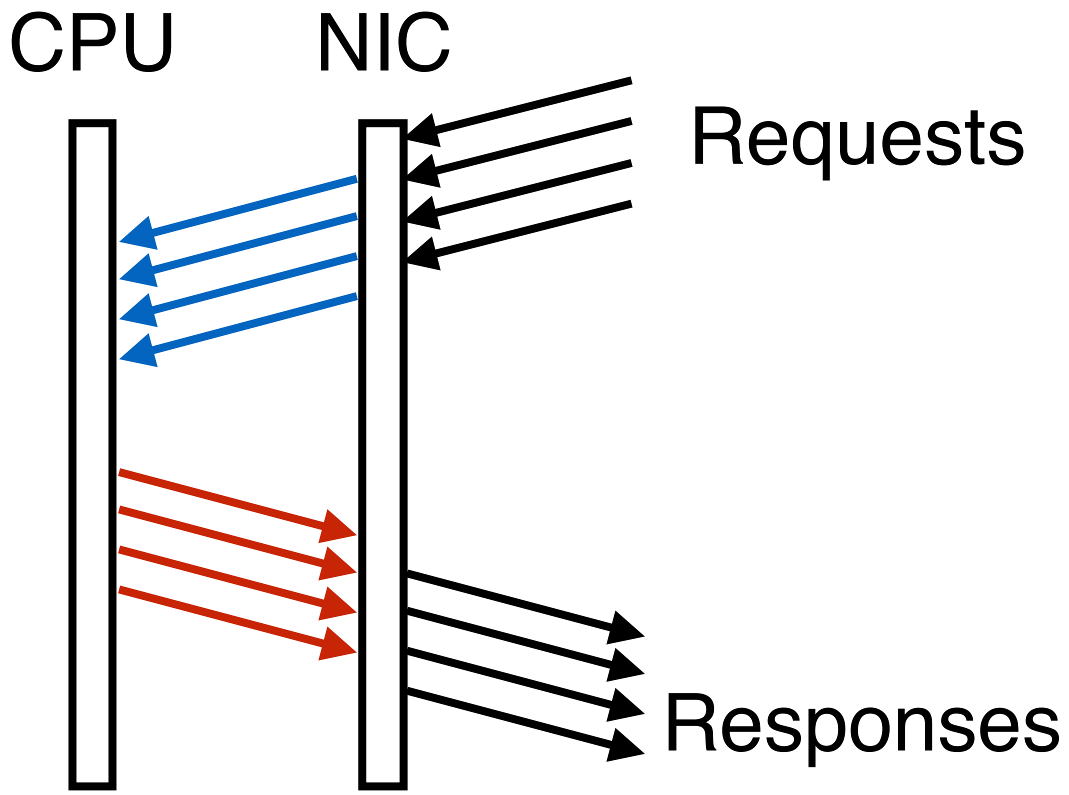
SEND

NIC

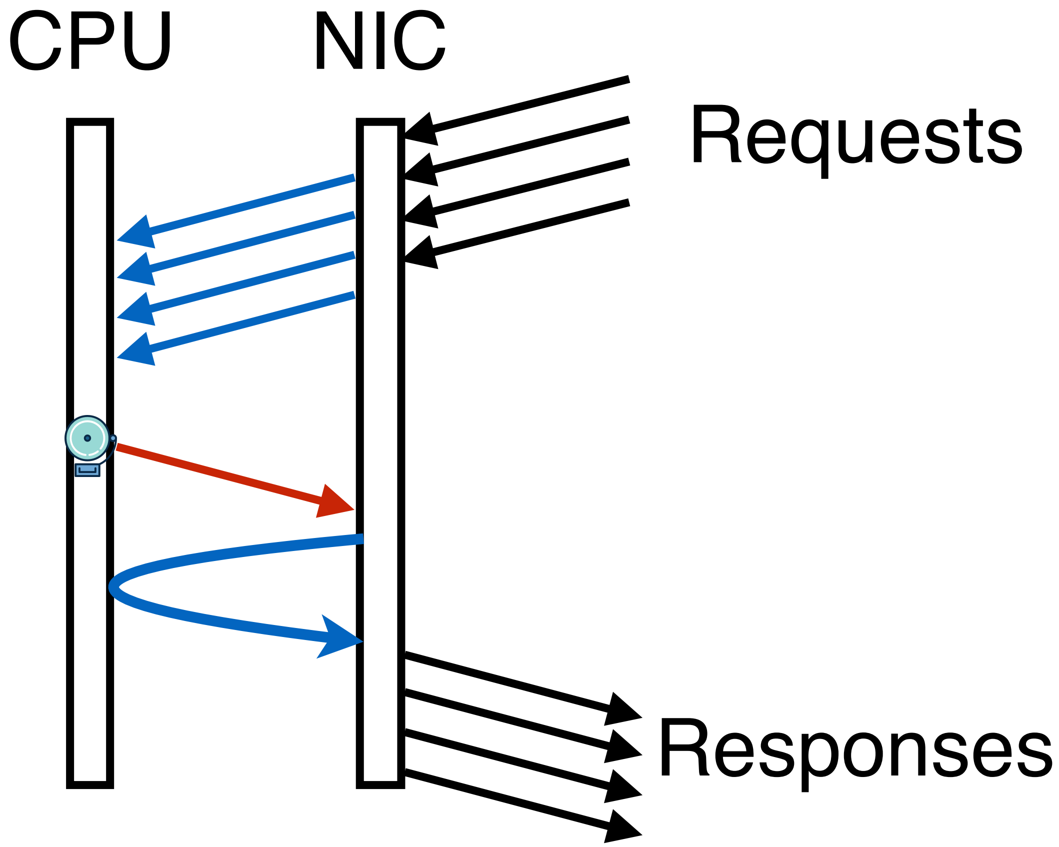
DMA

RPCs w/ Doorbell batching

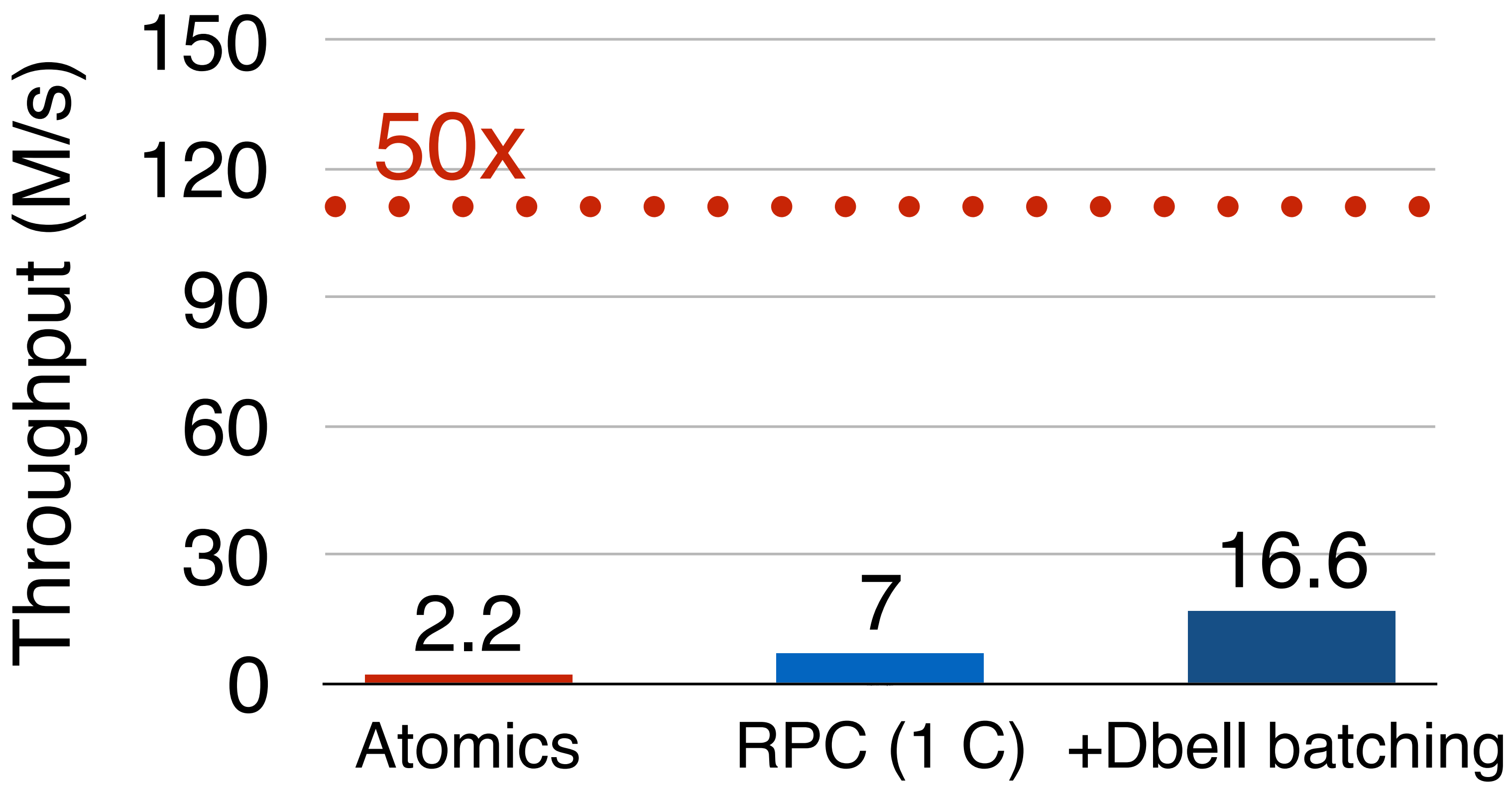
Push



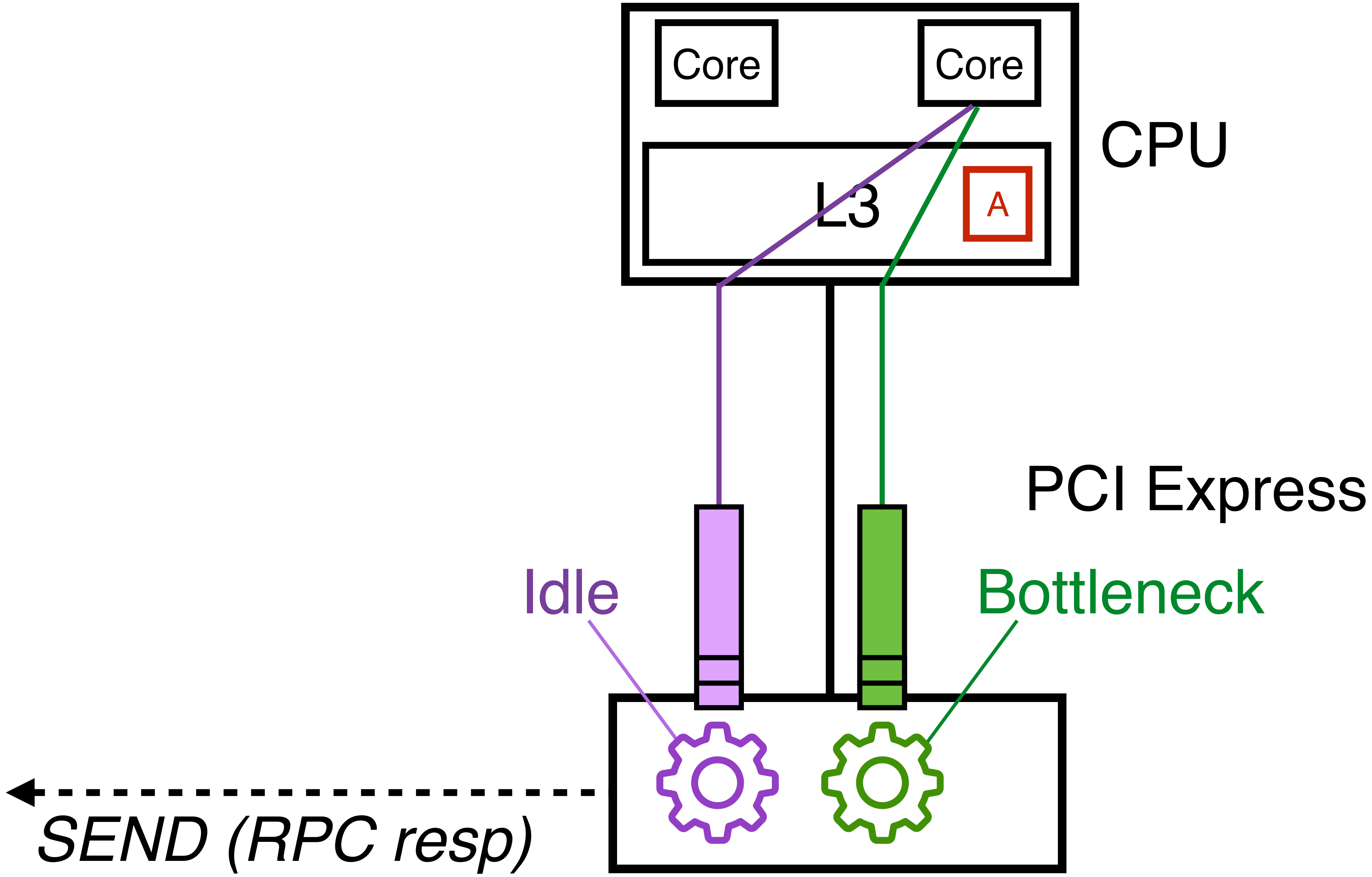
Pull (Doorbell batching)



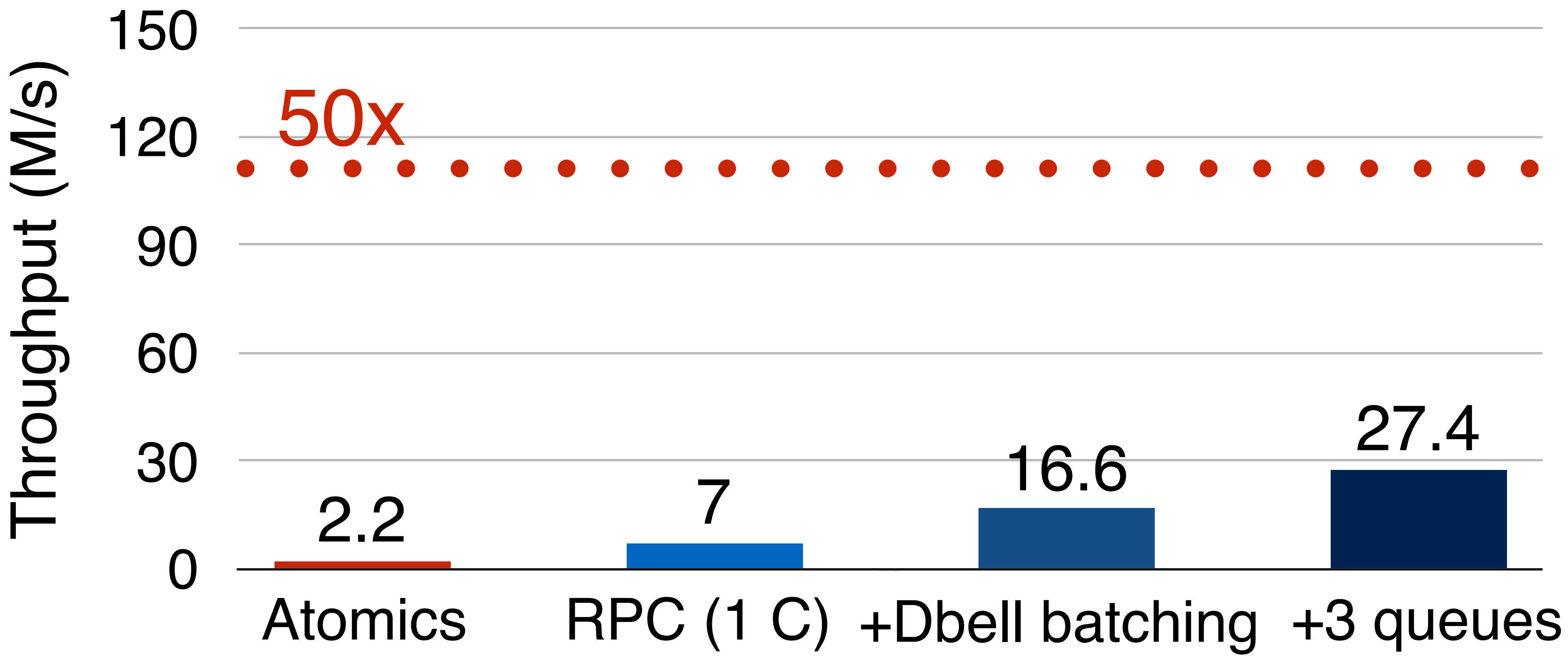
Sequencer throughput

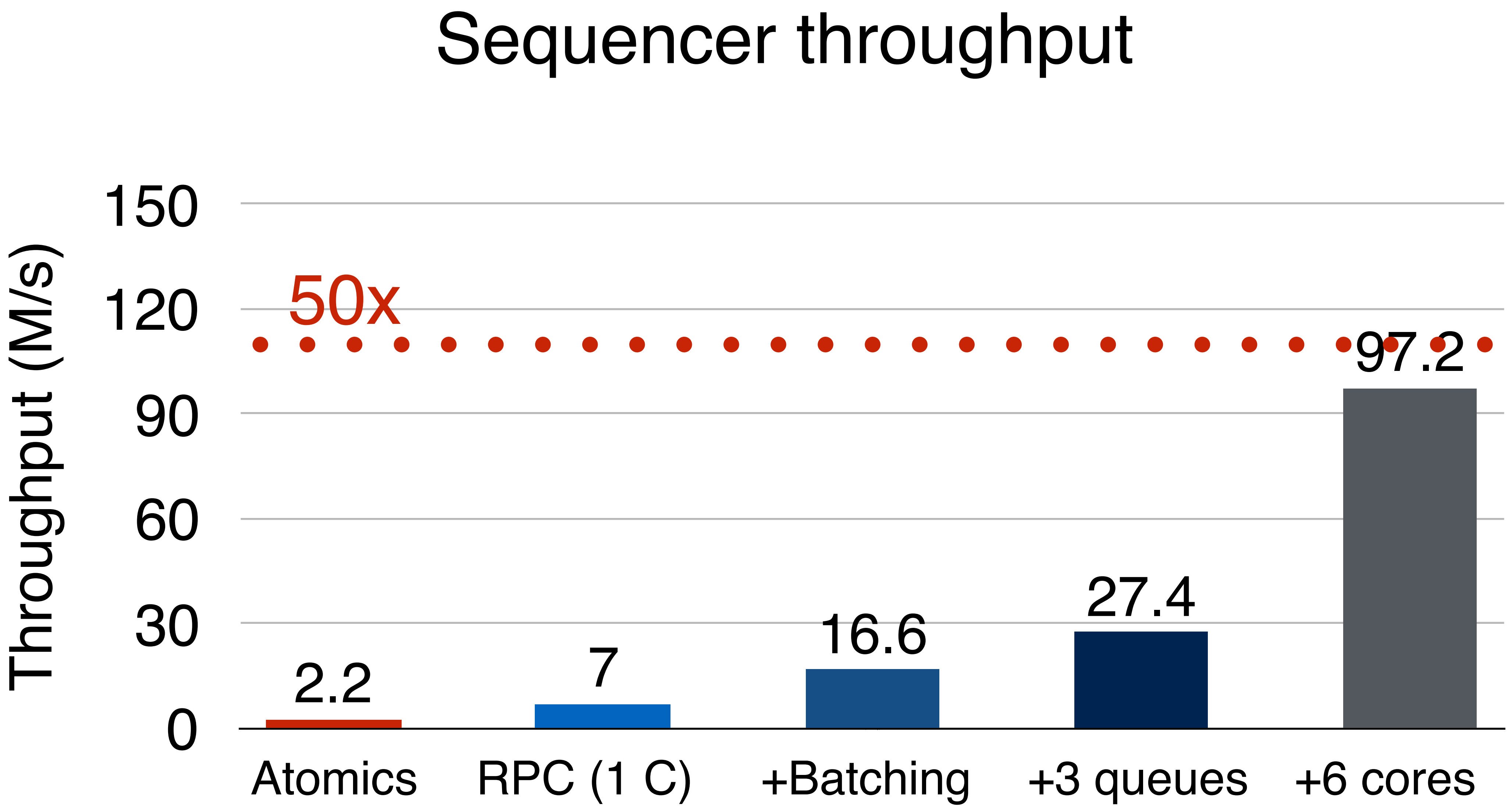


Exploit NIC parallelism w/ multiQ



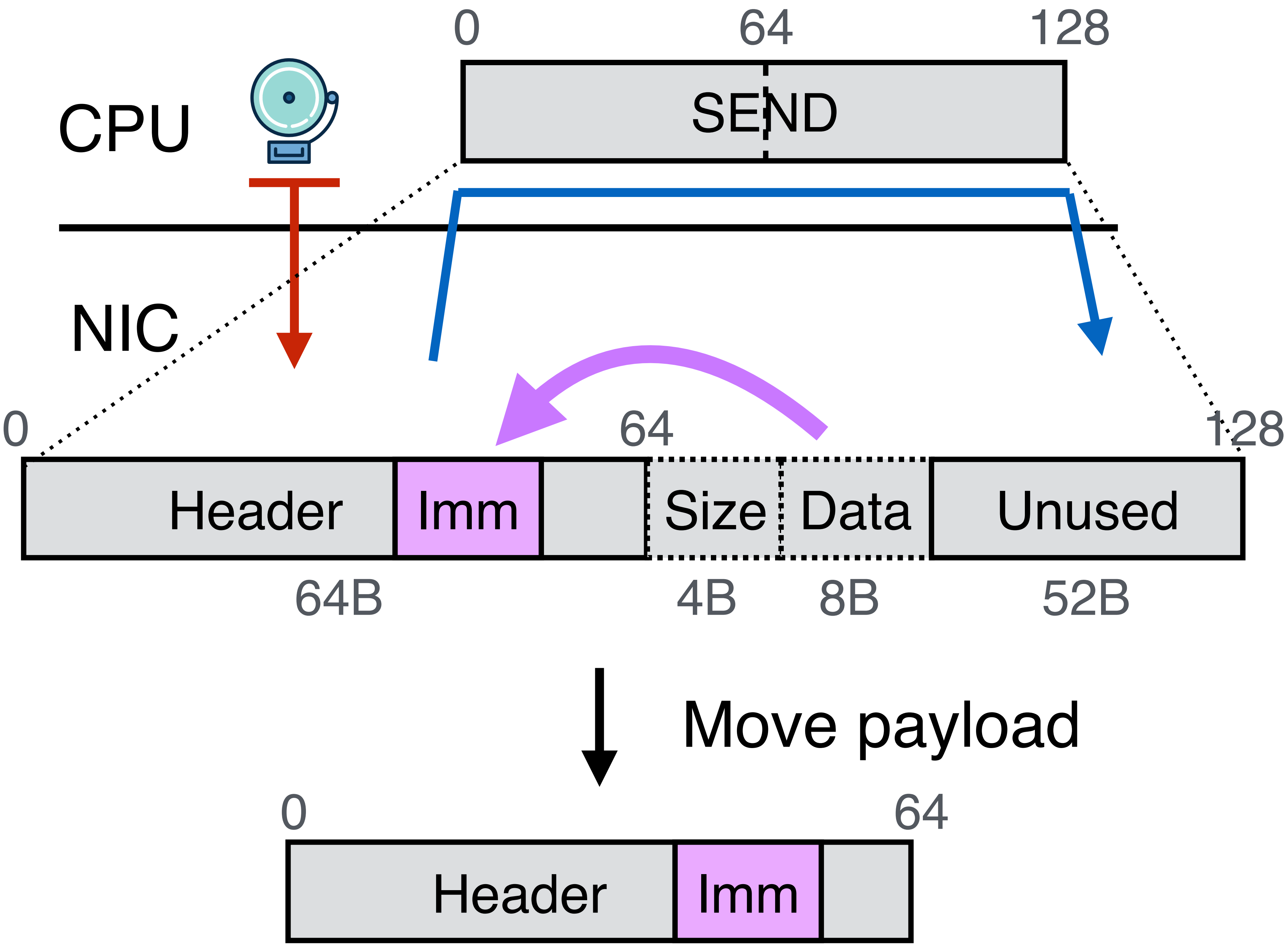
Sequencer throughput



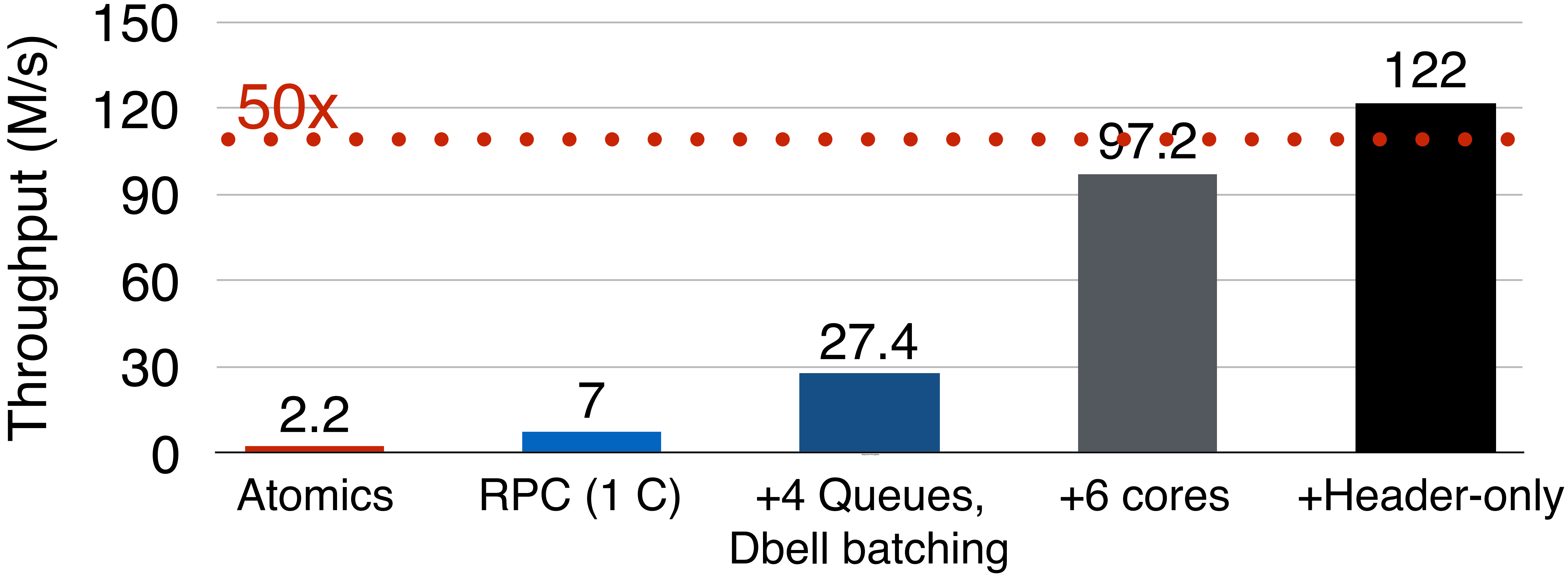


Bottleneck = PCIe DMA bandwidth (paper)

Reduce DMA size: Header-only



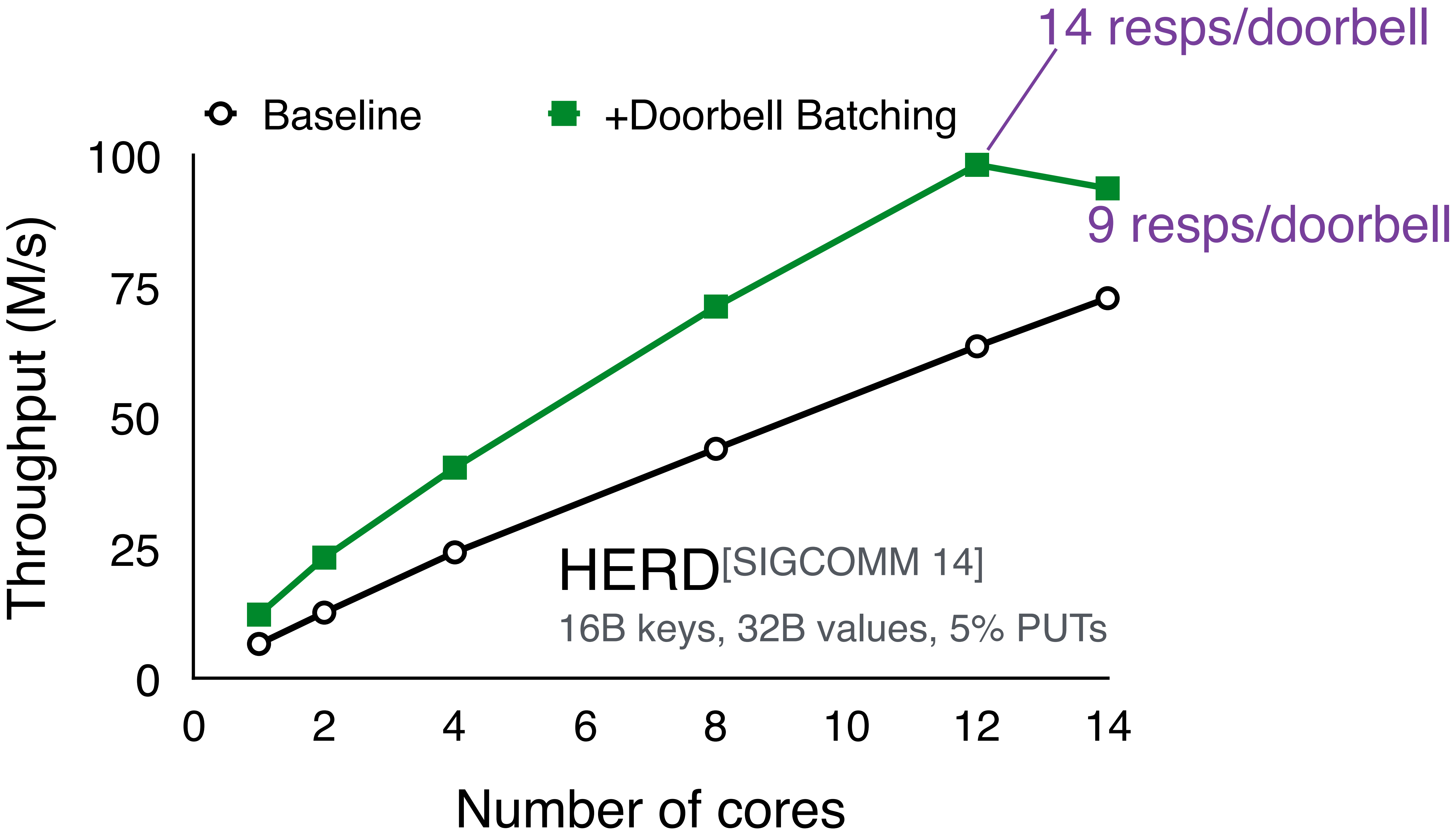
Sequencer throughput



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



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