Performance Isolation Anomalies in RDMA

Yiwen Zhang

with Juncheng Gu, Youngmoon Lee, Mosharaf Chowdhury, and Kang G. Shin

UNIVERSITY OF MICHIGAN

RDMA Is Being Deployed in Datacenters

Cloud operators are aggressively deploying RDMA in datacenters[1][2][3]

[1] Guo, Chuanxiong, et al."RDMA over Commodity Ethemet at Scale. " SIGCOMM'16
[2] Mittal, Radhika, et al. "TIMELY: RTT-based Congestion Control for the Datacenter." SIGCOMM'15
[3] Zhu, Yibo, et al. " Congestion control for large-scale RDMA deployments." SIGCOMM'15

RDMA Is Being Deployed in Datacenters

Cloud operators are aggressively deploying RDMA in datacenters[1][2][3]

Growing demands in ultra-low latency applications

- Key-value store & remote paging

High bandwidth applications

- Cloud storage & memory-intensive workloads

[1] Guo, Chuanxiong, et al."RDMA over Commodity Ethernet at Scale. " SIGCOMM'16
[2] Mittal, Radhika, et al. "TIMELY: RTT-based Congestion Control for the Datacenter." SIGCOMM'15
[3] Zhu, Yibo, et al. " Congestion control for large-scale RDMA deployments." SIGCOMM'15

RDMA Is Being Deployed in Datacenters

Cloud operators are aggressively deploying RDMA in datacenters

RDMA provides both low latency and high bandwidth

- Order-of-magnitude improvements in latency and throughput
- With minimal CPU overhead!

Mellanox TECHNOLOGIES

Google

Great! But There Are Limits …

At large-scale deployments, RDMA-enabled applications are unlikely to run in vacuum – the network must be shared

Great! But There Are Limits …

At large-scale deployments, RDMA-enabled applications are unlikely to run in vacuum – the network must be shared

HPC community uses static partitioning to minimize sharing[1]

Researches in RDMA over Ethernet-based datacenters focus on the vagaries of Priority-based Flow Control (PFC)[2][3]

[1] Ranadive, Adit, et al. "FaReS:Fairresource scheduling for VMM-bypass In Infiniband devices." CCGRID 2010
[2] Guo, Chuanxiong, et al. "RDMA over Commodity Ethernet at Scale. " SIGCOMM'16
[3] Zhu, Yibo, et al. " Congestion control for large-scale RDMA deployments." SIGCOMM'15

What Happens When Multiple RDMA-Enabled Applications Share The Network?

At A First Glance…

Scenarios	Fair?
10B vs. 10B	

At A First Glance…

Scenarios	Fair?
10B vs. 10B	
10B vs. 1MB	

At A First Glance…

Scenarios	Fair?
10B vs. 10B	
10B vs. 1MB	
1MB vs. 1MB	

At A First Glance…

Scenarios	Fair?
10B vs. 10B	
10B vs. 1MB	
1MB vs. 1MB	
1MB vs. 1GB	

Benchmarking Tool[1]

Modified based on *Mellanox Perftest* tool

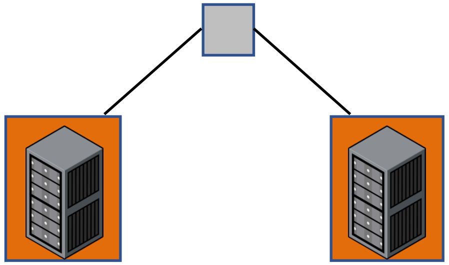
- Creates 2 flows to simultaneously transfer a stream of messages
- Single queue pair for each flow
- Measures bandwidth and latency characteristics only when both flows are active

[1] https://github.com/Infiniswap/frdma_benchmark

Benchmarking Tool[1]

Modified based on *Mellanox Perftest* tool

- Creates 2 flows to simultaneously transfer a stream of messages
- Single queue pair for each flow
- Measures bandwidth and latency characteristics only when both flows are active
- Both flows share the same link



[1] https://github.com/Infiniswap/frdma_benchmark

RDMA Design Parameters

RDMA Verbs

- WRITE, READ, WRITE WITH IMM (WIMM), and SEND/RECEIVE

Transport Type

- All experiments using Reliable-Connected (RC) Queue Pairs

INLINE Message

- Enabled INLINE message for 10 Byte and 100 Byte messages in the experiment

Application-Level Parameters

Request Pipelining

- Provide better performance, but hard to configure for fair comparison
- Disabled by default

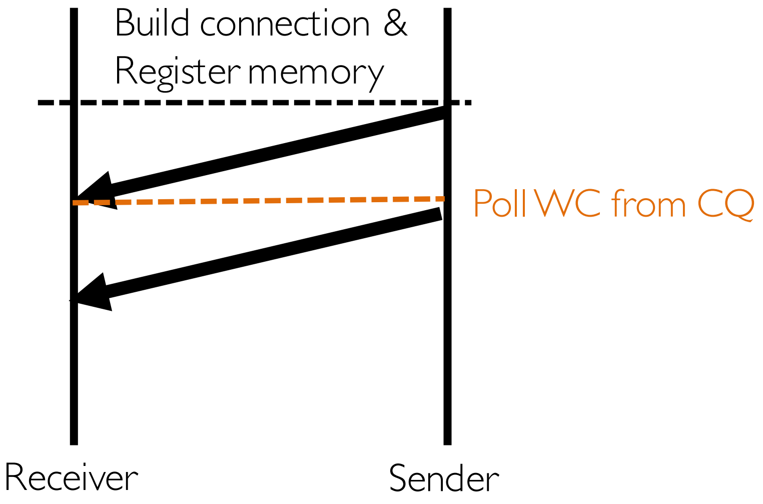
Polling mechanism

- Busy vs Event-triggered polling

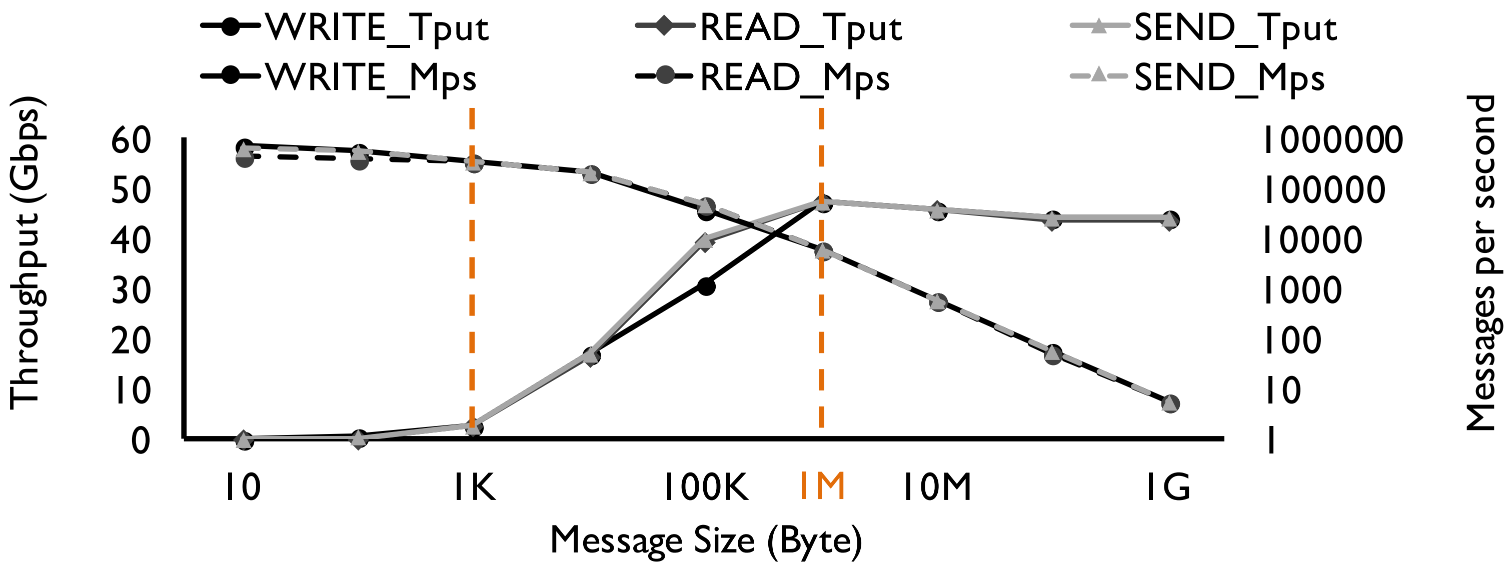
Application-Level Parameters

Message Acknowledgement

- Next work request is posted until the WC of the previous one is polled from CQ
- No other flow control acknowledgment is used



Define an Elephant and a Mouse



Mouse

Elephant

Elephant vs. Elephant

Compare two throughput-sensitive flows by varying verb types, message sizes, and polling mechanism.

- WRITE, READ, WIMM, & SEND verbs transferring 1MB & 1GB messages
- Total amount of data transferred fixed at 1TB
- Both flows using **event-triggered polling**
- Generated bandwidth ratio matrix

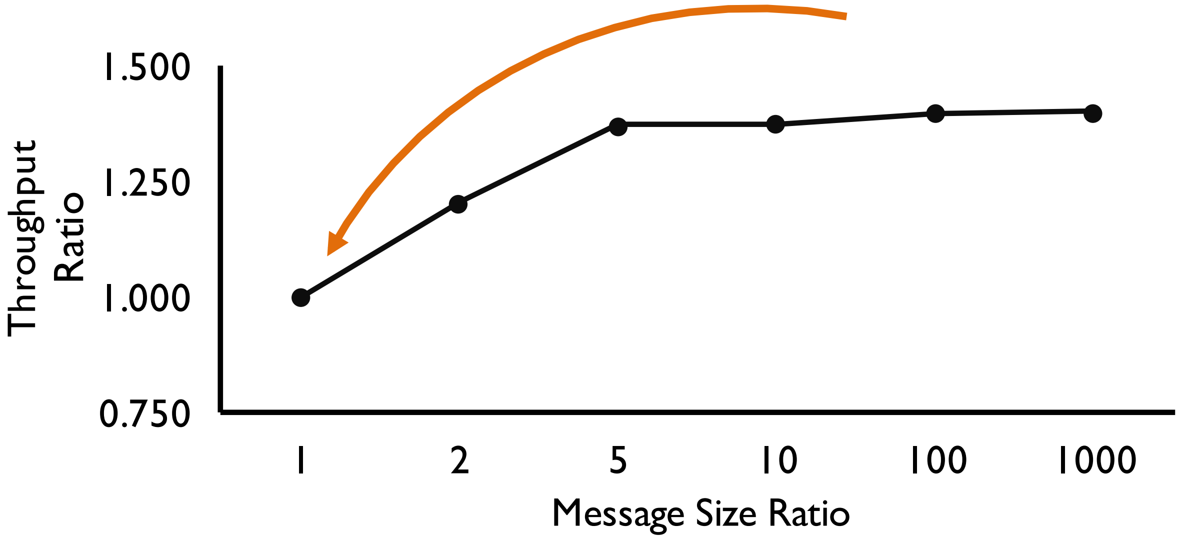
Elephant vs. Elephant: Larger Flows Win

		SEND		WIMM		READ		WRITE	
		1MB	1GB	1MB	1GB	1MB	1GB	1MB	1GB
WRITE	1GB	🟧	⬛	🟧	⬛	🟧	⬛	🟧	⬛
	1MB	⬛	🟧	⬛	🟧	⬛	🟧	⬛	
READ	1GB	🟧	⬛	🟧	⬛	🟧	⬛		
	1MB	⬛	🟧	⬛	🟧	⬛			
WIMM	1GB	🟧	⬛	🟧	⬛				
	1MB	⬛	🟧	⬛					
SEND	1GB	🟧	⬛						
	1MB	⬛							

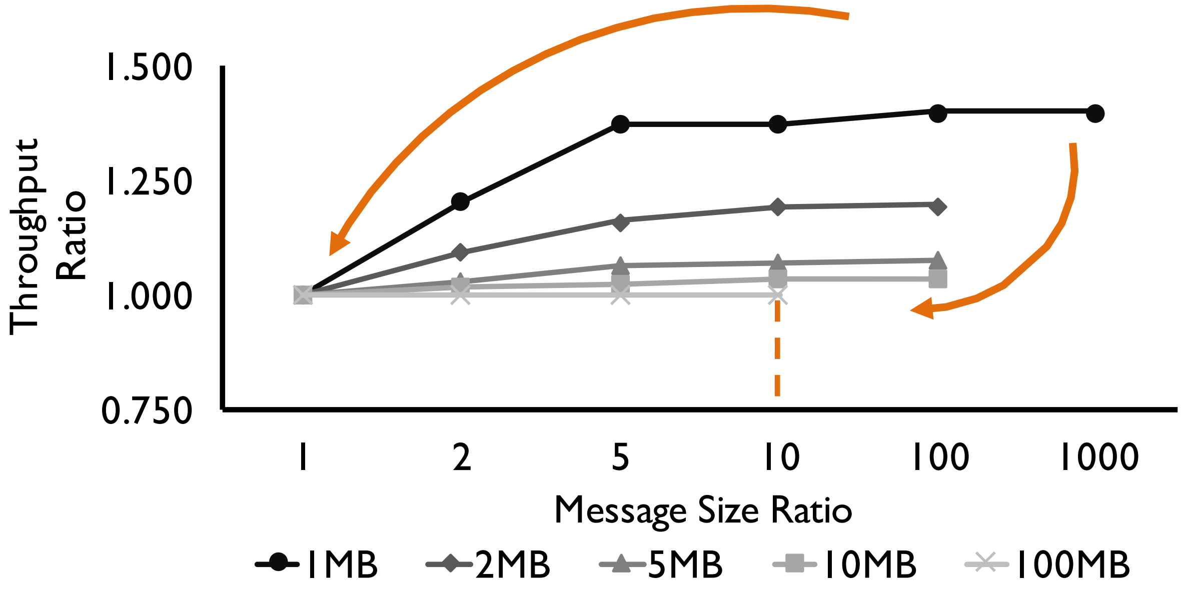
⬛ Fair

🟧 Unfair

Getting Better with Larger Base Flows

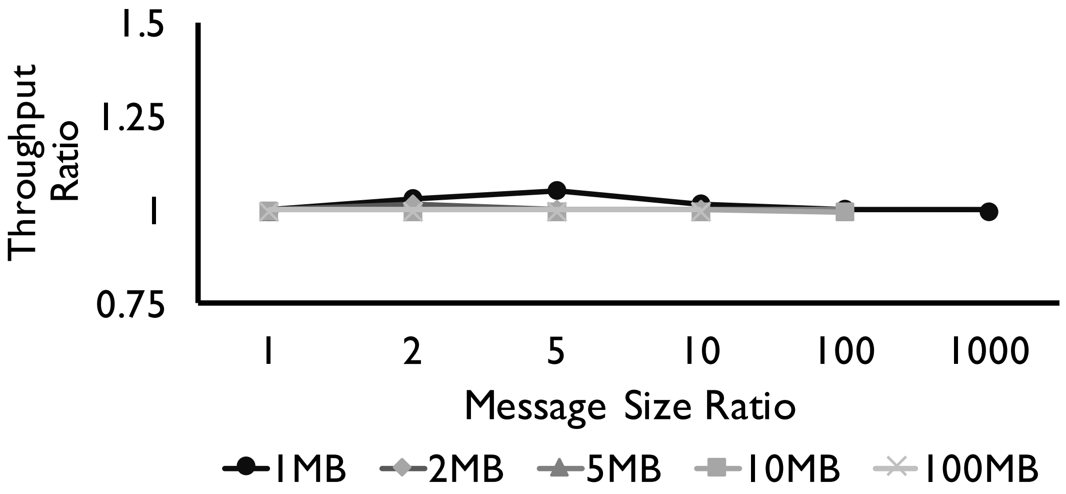


Getting Better with Larger Base Flows

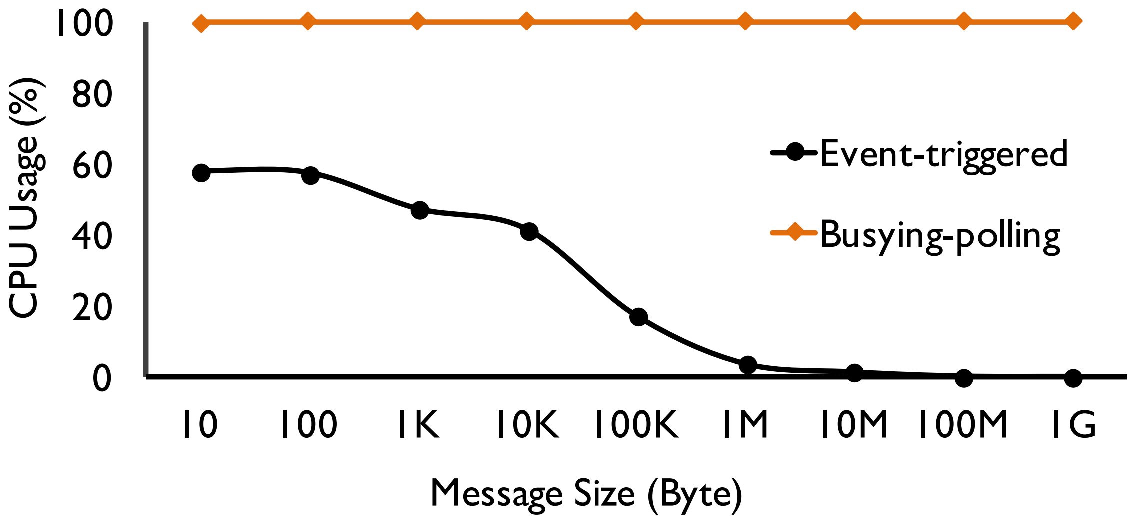


Polling Matters: Is Busy-polling Better?

Both flows use busy-polling.



But There Is a Tradeoff in CPU Usage



At A First Glance…

Scenarios	Fair?
10B vs. 10B	
10B vs. 1MB	
1MB vs. 1MB	
1MB vs. 1GB	

At A First Glance…

Scenarios	Fair?
10B vs. 10B	
10B vs. 1MB	
1MB vs. 1MB	Fair
1MB vs. 1GB	

At A First Glance…

Scenarios	Fair?
10B vs. 10B	
10B vs. 1MB	
1MB vs. 1MB	Fair
1MB vs. 1GB	Unfair

At A First Glance…

Scenarios	Fair?
10B vs. 10B	
10B vs. 1MB	
1MB vs. 1MB	Depends on CPU
1MB vs. 1GB	Unfair

At A First Glance…

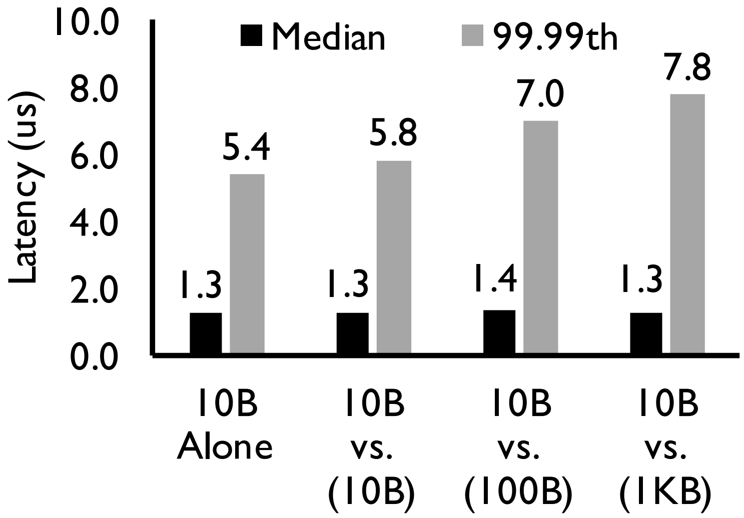
Scenarios	Fair?
10B vs. 10B	
10B vs. 1MB	
1MB vs. 1MB	Depends on CPU
1MB vs. 1GB	Depends on CPU

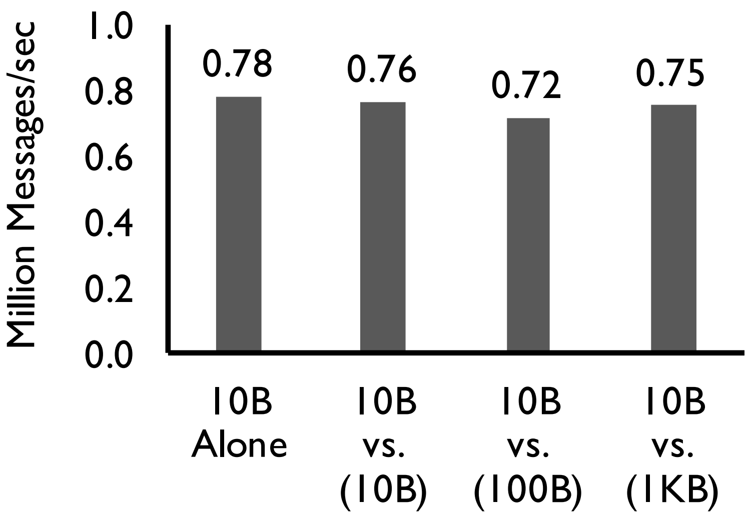
Mouse vs. Mouse: Pick a Base Flow

Compare two latency-sensitive flows with varying message sizes.

- All flows using WRITE operation with busy polling
- 10B, 100B and 1KB messages
- Pick 10B as base flow
- Measured latency and MPS of the base flow transferring 10 million messages at the presence of a competing flow

Mouse vs. Mouse: Worse Tails





At A First Glance…

Scenarios	Fair?
10B vs. 10B	
10B vs. 1MB	
1MB vs. 1MB	Depends on CPU
1MB vs. 1GB	Depends on CPU

At A First Glance…

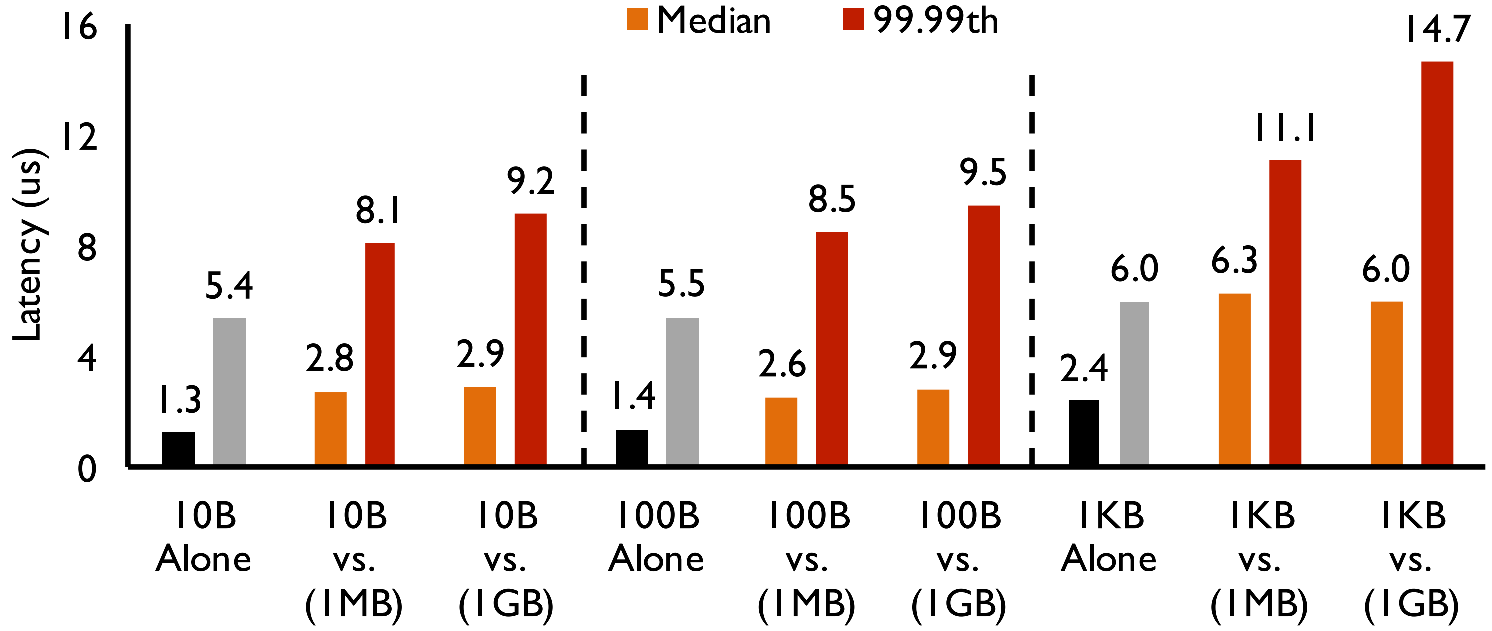
Scenarios	Fair?
10B vs. 10B	Good enough
10B vs. 1MB	
1MB vs. 1MB	Depends on CPU
1MB vs. 1GB	Depends on CPU

Mouse vs. Elephant

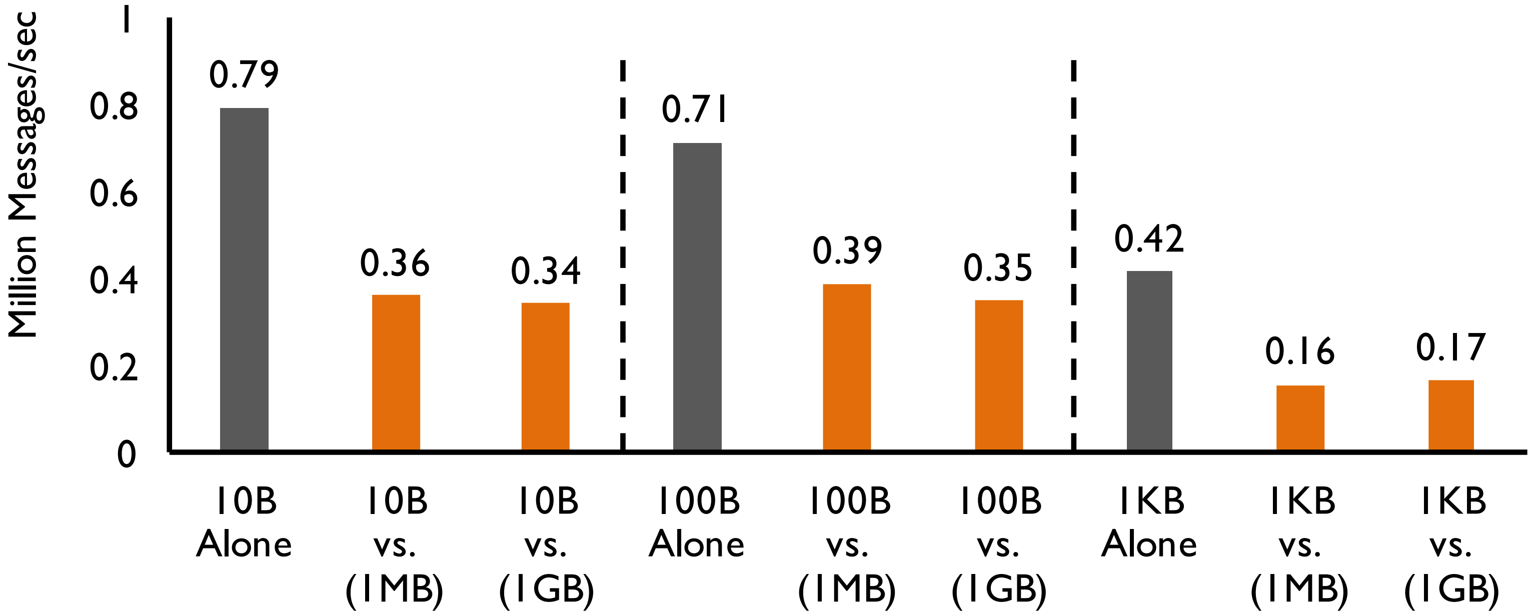
Study performance isolation of a mouse flow running under a background elephant flow.

- All flows using WRITE operation
- All mouse flows sending 10 millions messages
- Mouse flows using busy polling while background elephant flows using event-triggered polling
- Measured latency and MPS of mouse flows

Mouse vs. Elephant: Mouse Flows Suffer



Mouse vs. Elephant: Mouse Flows Suffer



At A First Glance…

Scenarios	Fair?
10B vs. 10B	Good enough
10B vs. 1MB	
1MB vs. 1MB	Depends on CPU
1MB vs. 1GB	Depends on CPU

At A First Glance…

Scenarios	Fair?
10B vs. 10B	Good enough
10B vs. 1MB	Unfair
1MB vs. 1MB	Depends on CPU
1MB vs. 1GB	Depends on CPU

Hardware is Not Enough for Isolation

So far we ran all experiments using Mellanox FDR ConnectX-3 (56 Gbps) NIC on CloudLab.

Switch to Mellanox EDR ConnectX-4 (100 Gbps) NIC on the Umich Conflux cluster.

- The isolation problem in the elephant vs. elephant case still exists with a throughput ratio of 1.32.
- In the mouse vs. mouse case the problem appears to be mitigated; we did not observe large tail-latency variations when two mouse flows compete.
- In the mouse vs. elephant scenario, mouse flows are still affected by large background flows, where the median latency increases by up to 5×.

What Happens to Isolation in More Sophisticated and Optimized Applications?

Performance Isolation in HERD[1]

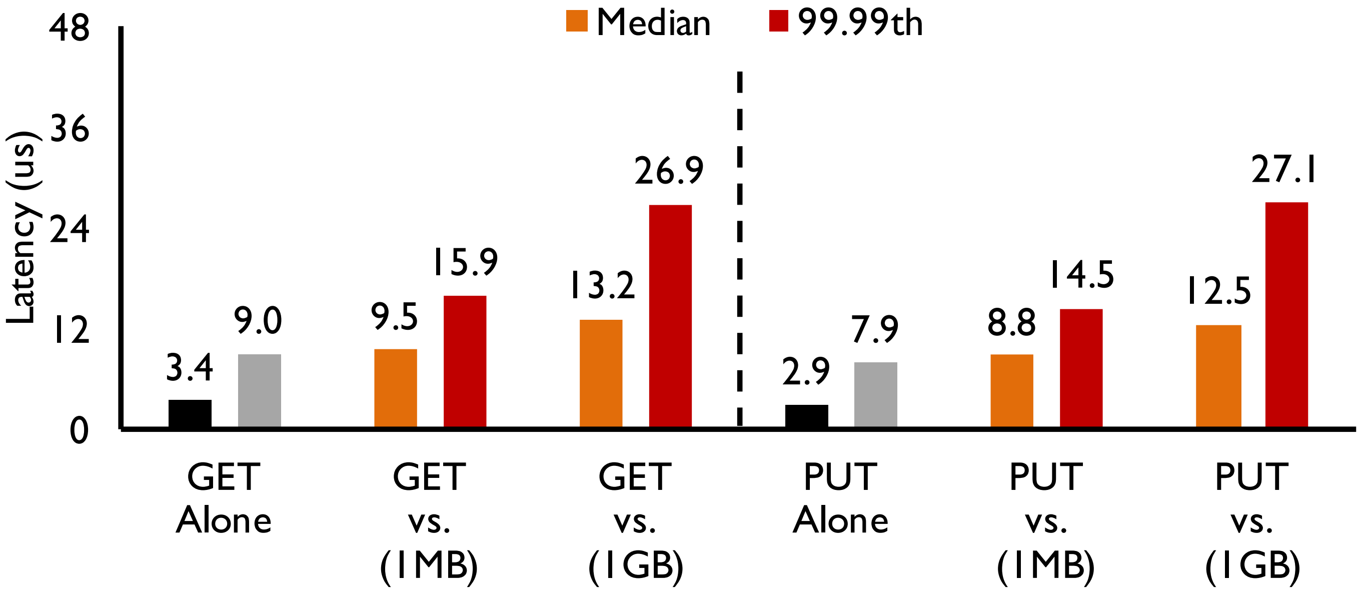
Interested to know how isolation is maintained in HERD **at a presence of a background elephant flow**.

Running HERD on the Umich Conflux cluster.

- 5 million PUT/GET requests.
- Background flows using 1MB or 1GB messages with event-triggered polling
- Measured median and tail latency of HERD requests with and without a background flow

[1] Kalia, Anuj, et al. "Using RDMA efficiently for key-value services" SIGCOMM 2014

HERD vs. Elephant: HERD Also Suffers



HERD vs. Elephant: Summary

HERD also has isolation issues when running with big background flows

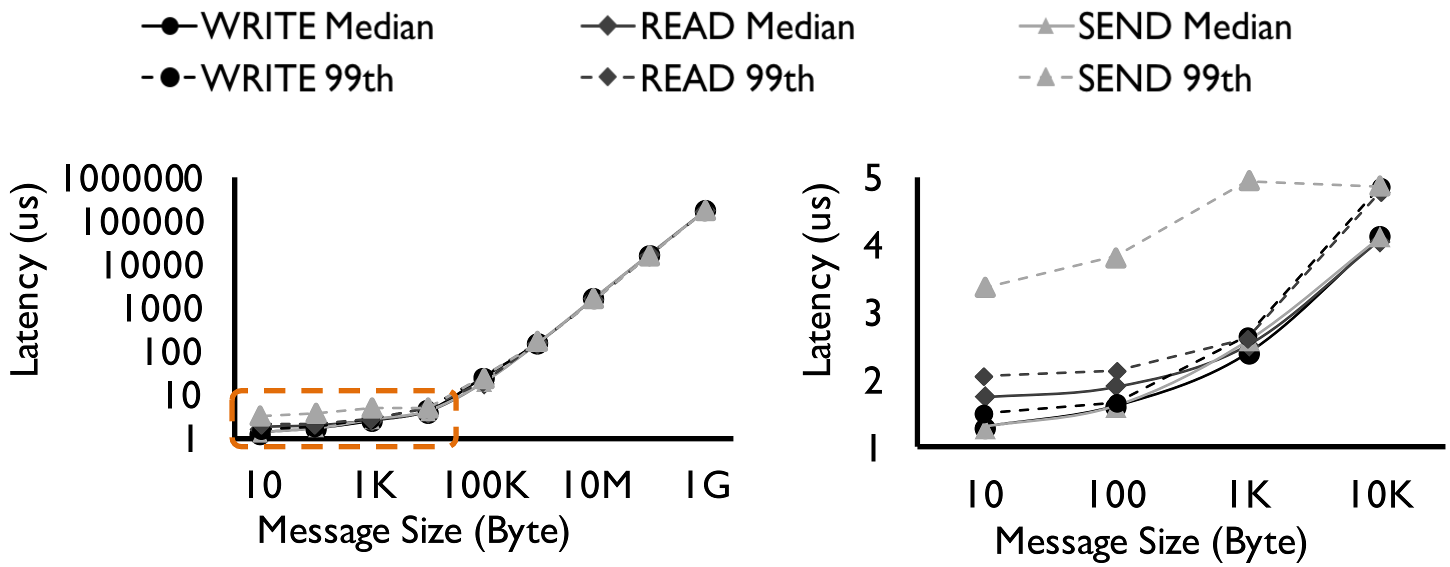
Currently, we are working on a solution to provide isolation in RDMA

Special thanks to Yue Tan's great help in generating isolation data on HERD

Summary

- When the size difference of two flows are small, no matter they are small flows or very big flows, the isolation appears to be good
- How fast an application can post RDMA requests onto the RNIC is the only thing that matters in a throughput-sensitive environment
- When the size difference of two flows are big, there is a performance degradation of the smaller flow
- Current hardware might not help to entirely resolve the issue

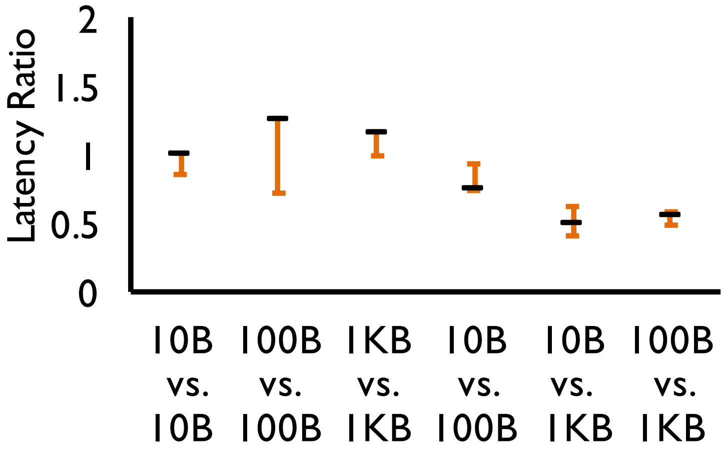
Mouse Flow Latency

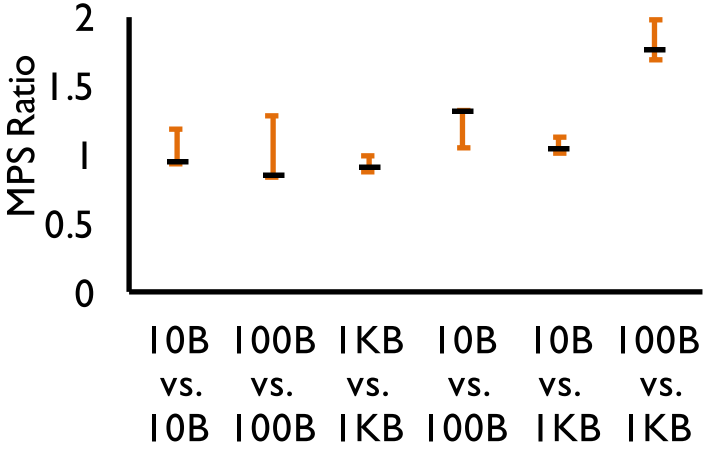


Elephant vs. Elephant: Matrix

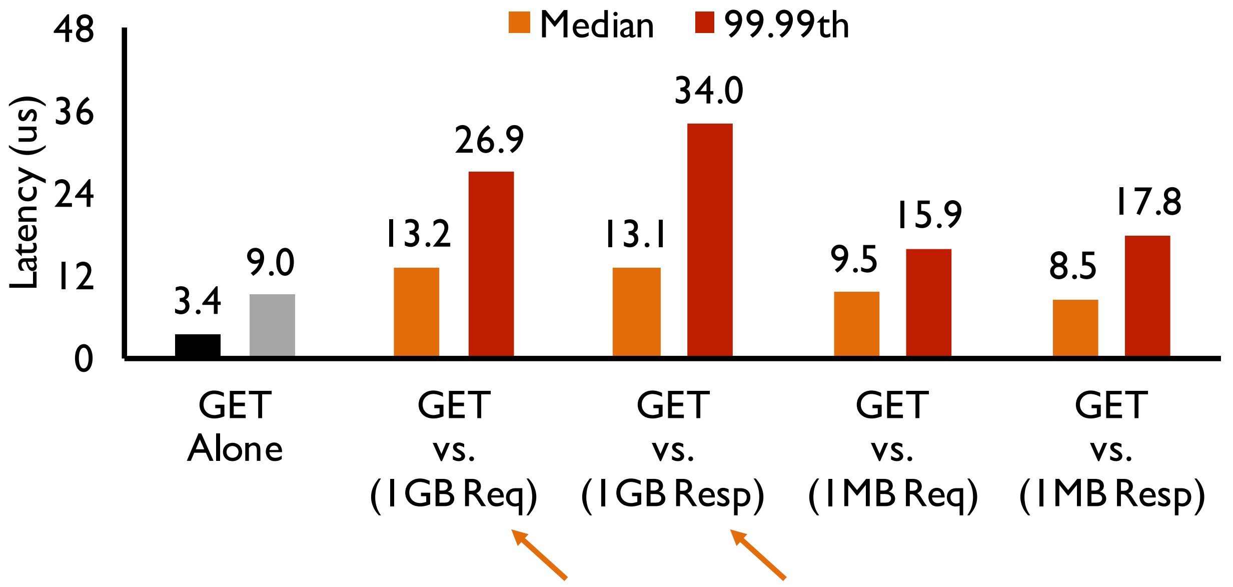
		SEND		WIMM		READ		WRITE	
		1MB	1GB	1MB	1GB	1MB	1GB	1MB	1GB
WRITE	1GB	1.41	1.00	1.44	1.00	1.39	1.00	1.40	1.00
	1MB	1.02	0.71	1.00	0.72	0.99	0.71	1.00	
READ	1GB	1.40	1.00	1.43	1.00	1.37	1.00		
	1MB	1.08	0.71	1.04	0.71	1.00			
WIMM	1GB	1.40	1.00	1.44	1.00				
	1MB	1.00	0.70	1.00					
SEND	1GB	1.41	1.00						
	1MB	1.00							

Mouse vs. Mouse: Unpredicted Behavior

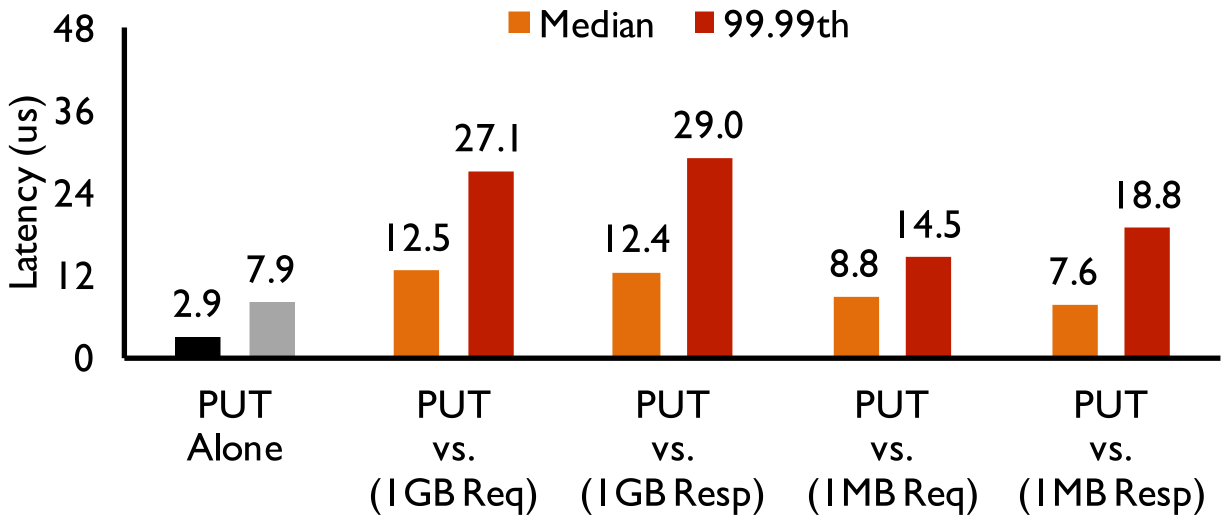




HERD vs. Elephant: HERD Also Suffers



HERD vs. Elephant: HERD Also Suffers



Summary

Elephant vs. Elephant:

- Polling mechanism dictates bandwidth allocation
- How fast an application can post RDMA requests onto the RNIC is the only thing that matters in a throughput-sensitive environment
- Tradeoff between CPU and Bandwidth

Mouse vs. Mouse:

- Little predictability between flows using equal-sized messages
- Increase in tail latency and decrease in MPS
- Isolation issue mitigated when switching to better hardware

Summary

Mouse vs. Elephant:

- In the presence of both types of flows, latency-sensitive flows suffer
- The requests posted by the mouse flows may queue up in RNIC's queue buffer while the RNIC is doing continuous DMA reads from the main memory due to the background flow

HERD vs. Elephant:

- Isolation issues remain when running with background elephant flows Up to 4x increase in the median latency