DSAIL Data Systems and AI Lab MIT CSAIL

BROWN

Fast Networks and the Next Generation of Transactional Database Systems

Tim Kraska <kraska@mit.edu>

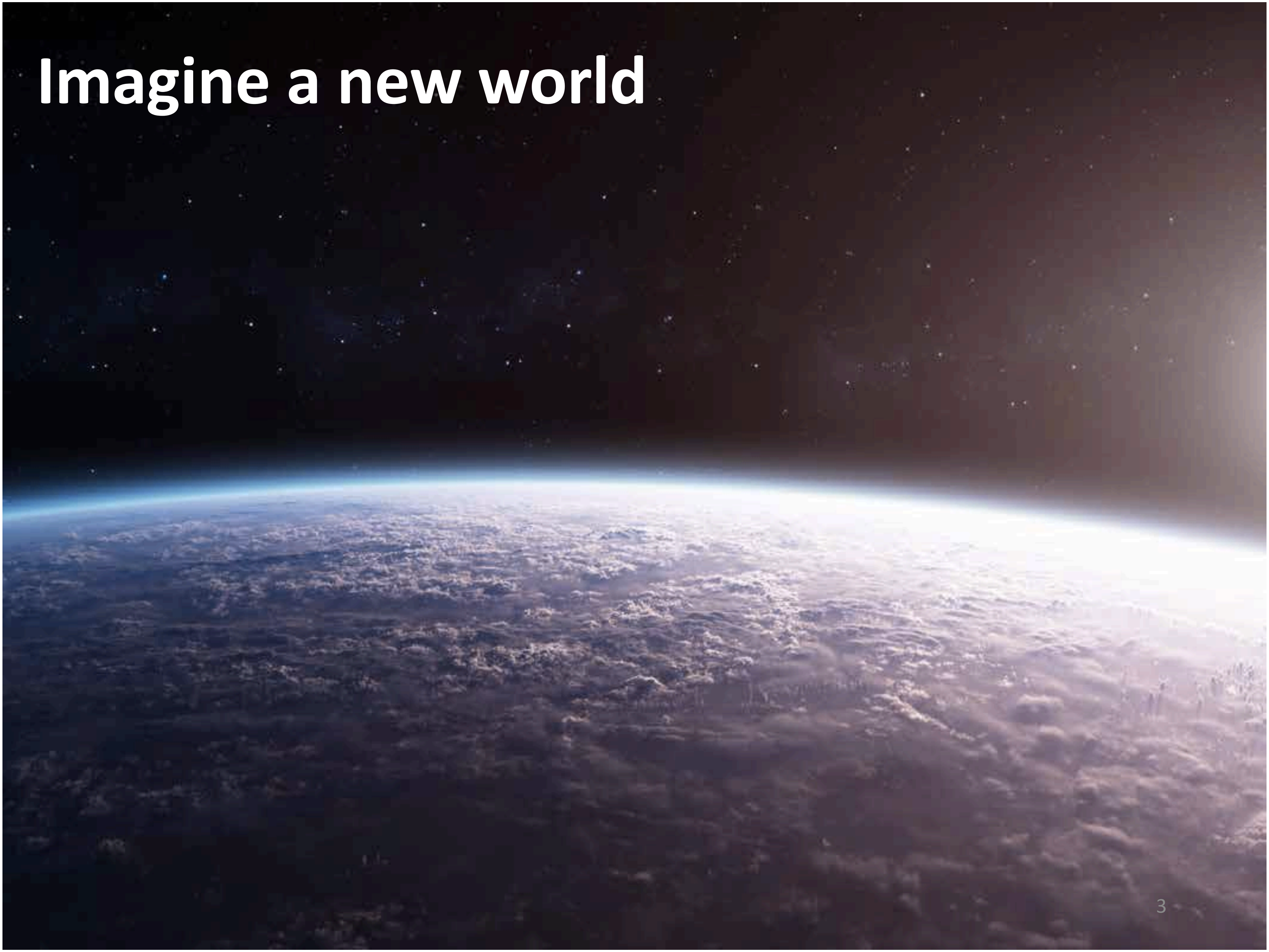
Microsoft Google intel NSF Mellanox Technologies ORACLE

Erfan Zamanian

<erfan_zamanian_dolati@brown.edu>

Is graduating very soon and is looking industry jobs

Imagine a new world



A better world


Merry Christmas

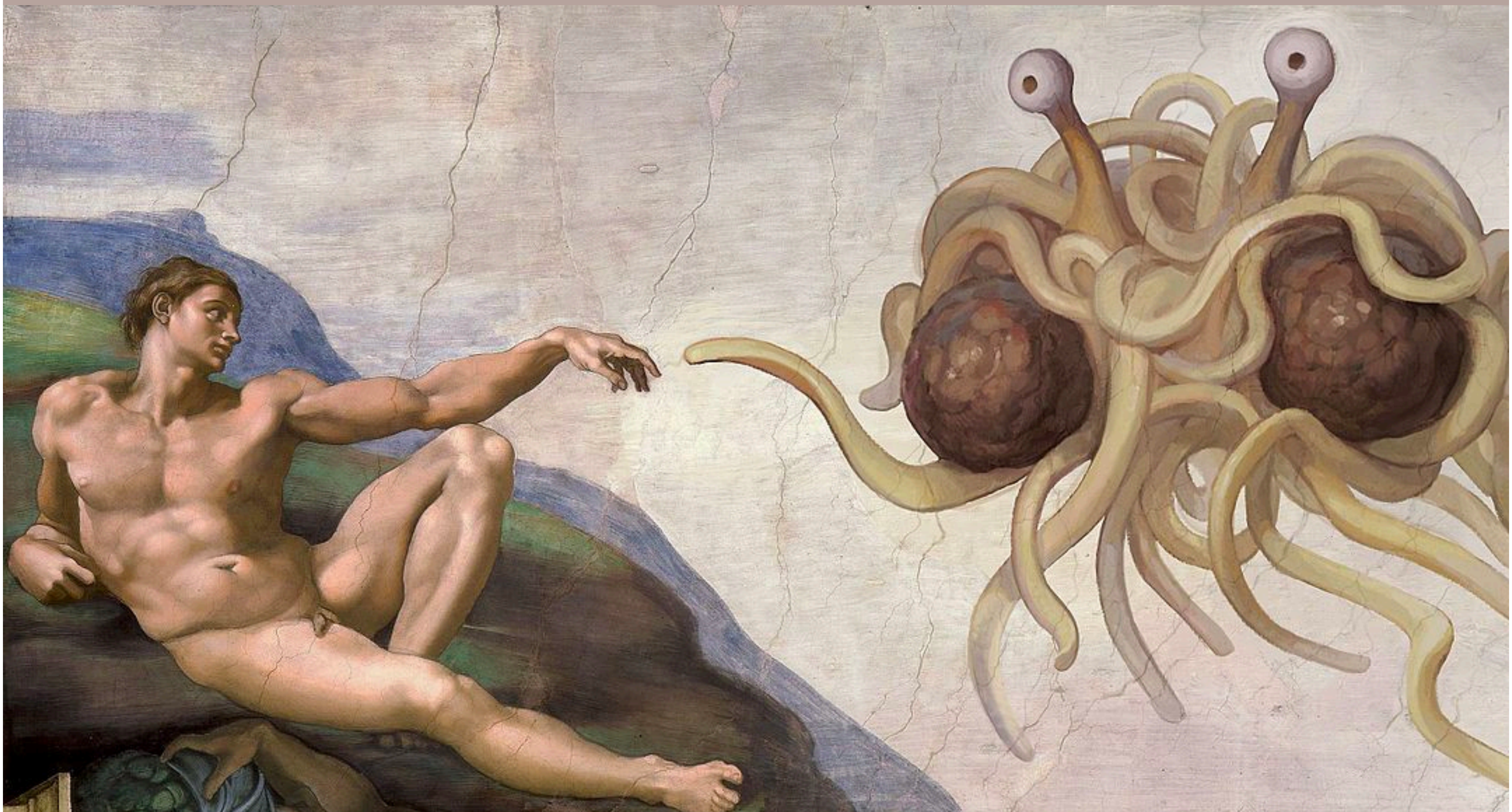
and with **Scalable Distributed Transactions**

For centuries app & system developers try to avoid distributed transactions with huge pain

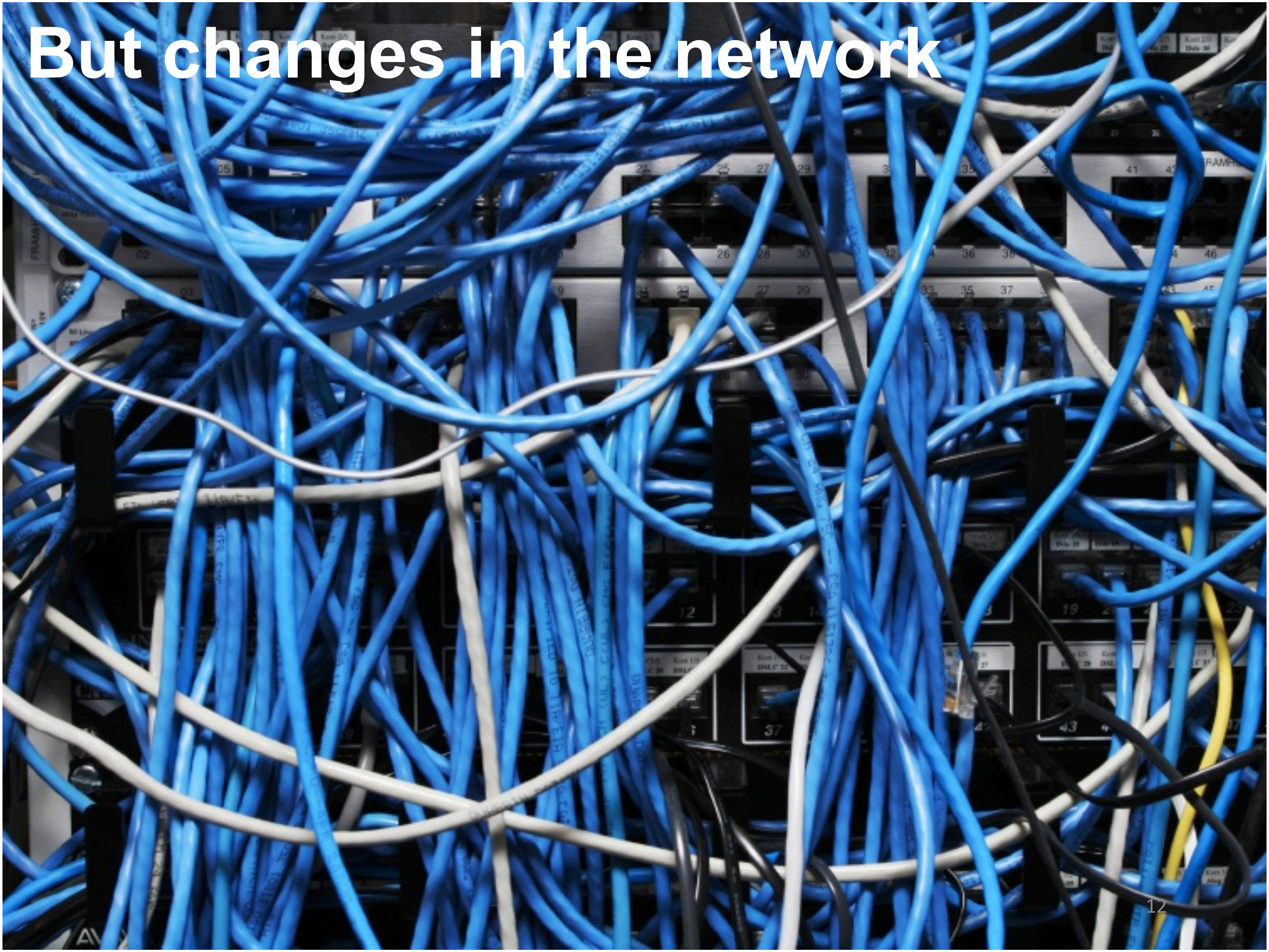
- Static- and dynamic data partitioning
- Data-aware scheduling
- Replication schemes
- ….

But if distributed transaction would scale, all this **pain would go away**

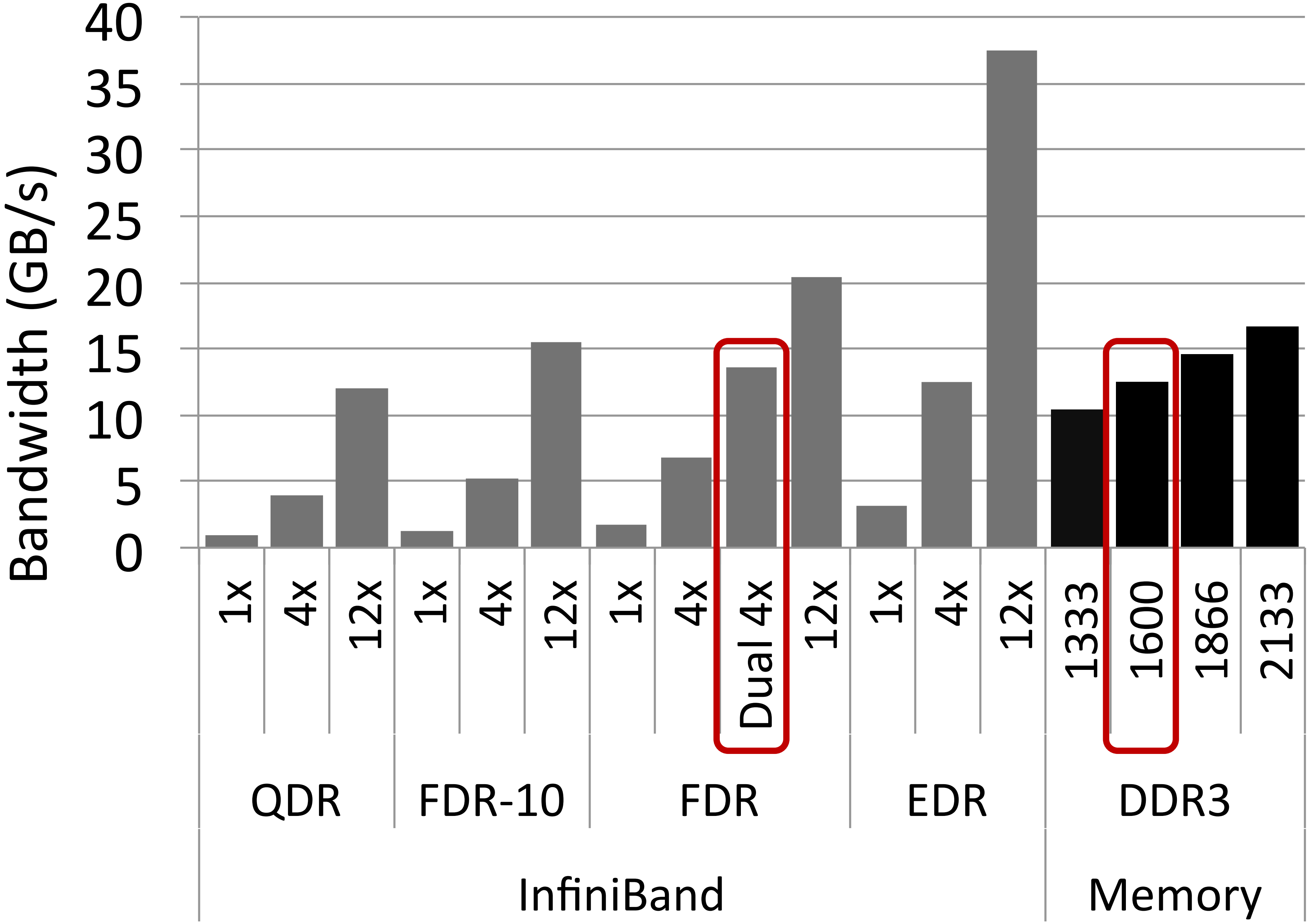
Yet, the relieve does not come from....



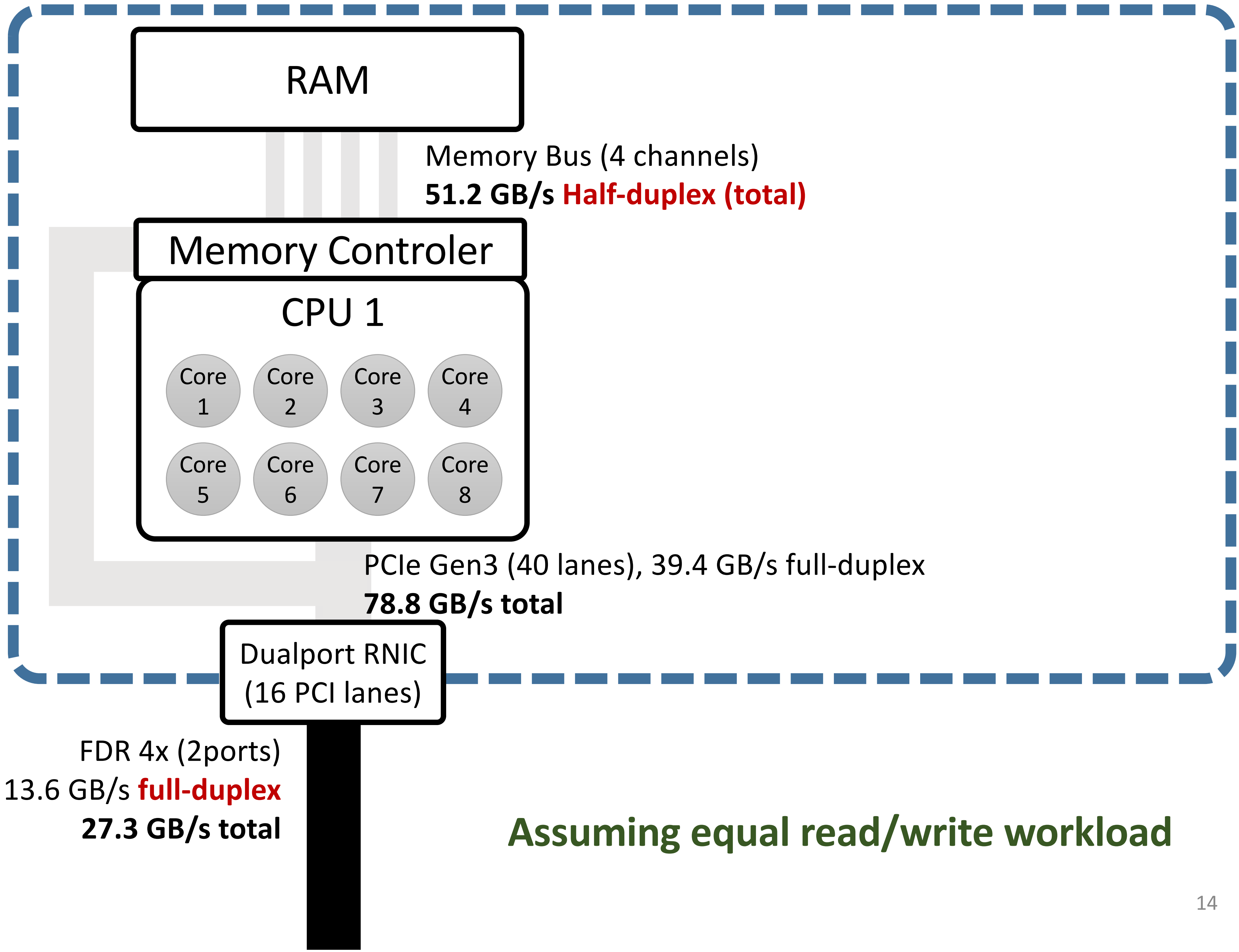
But changes in the network



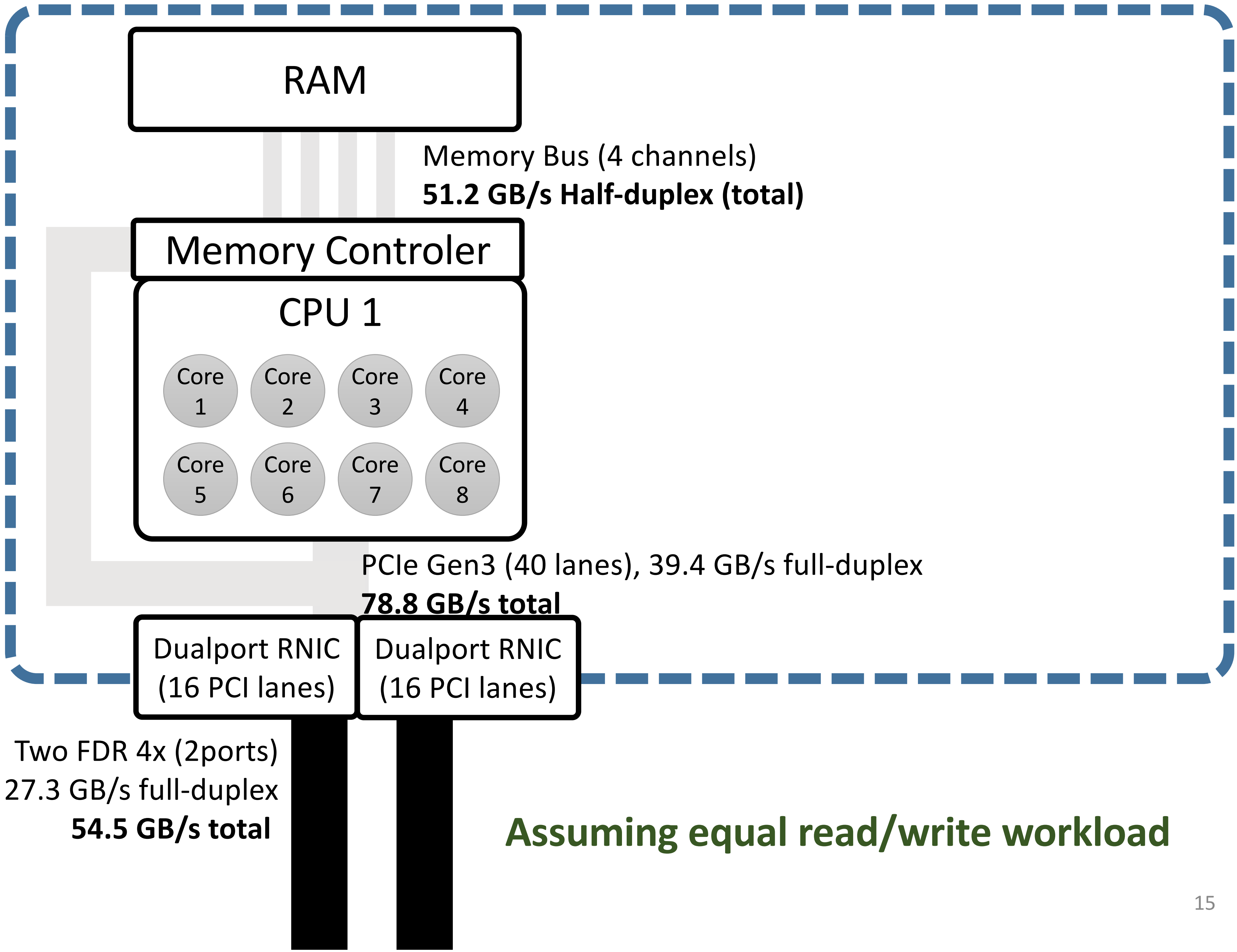
Specs



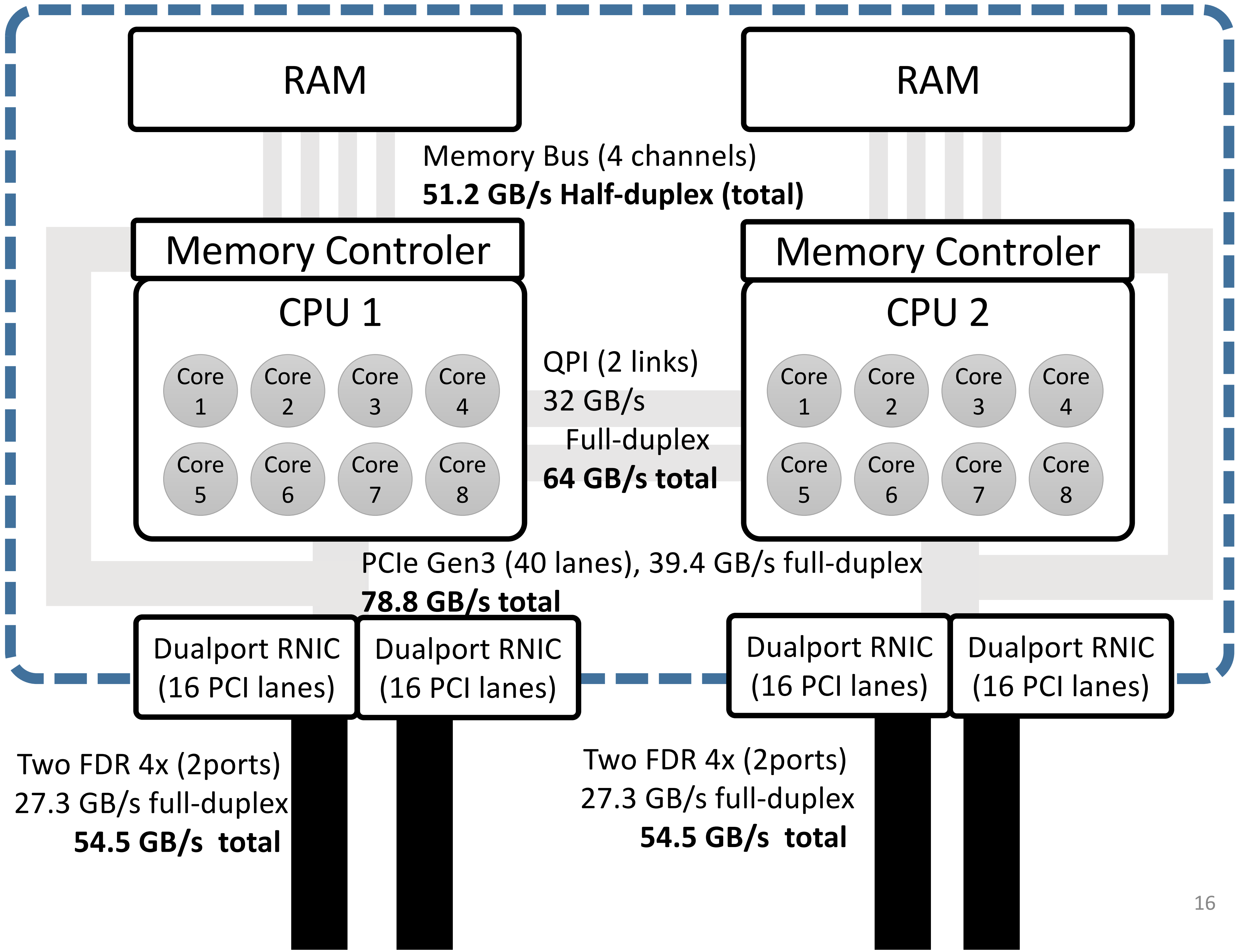
Bandwidth (GB/s)
40
35
30
25
20
15
10
5
0
1x
4x
12x
1x
4x
12x
1x
4x
Dual 4x
12x
1x
4x
12x
1333
1600
1866
2133
QDR
FDR-10
FDR
EDR
DDR3
InfiniBand
Memory



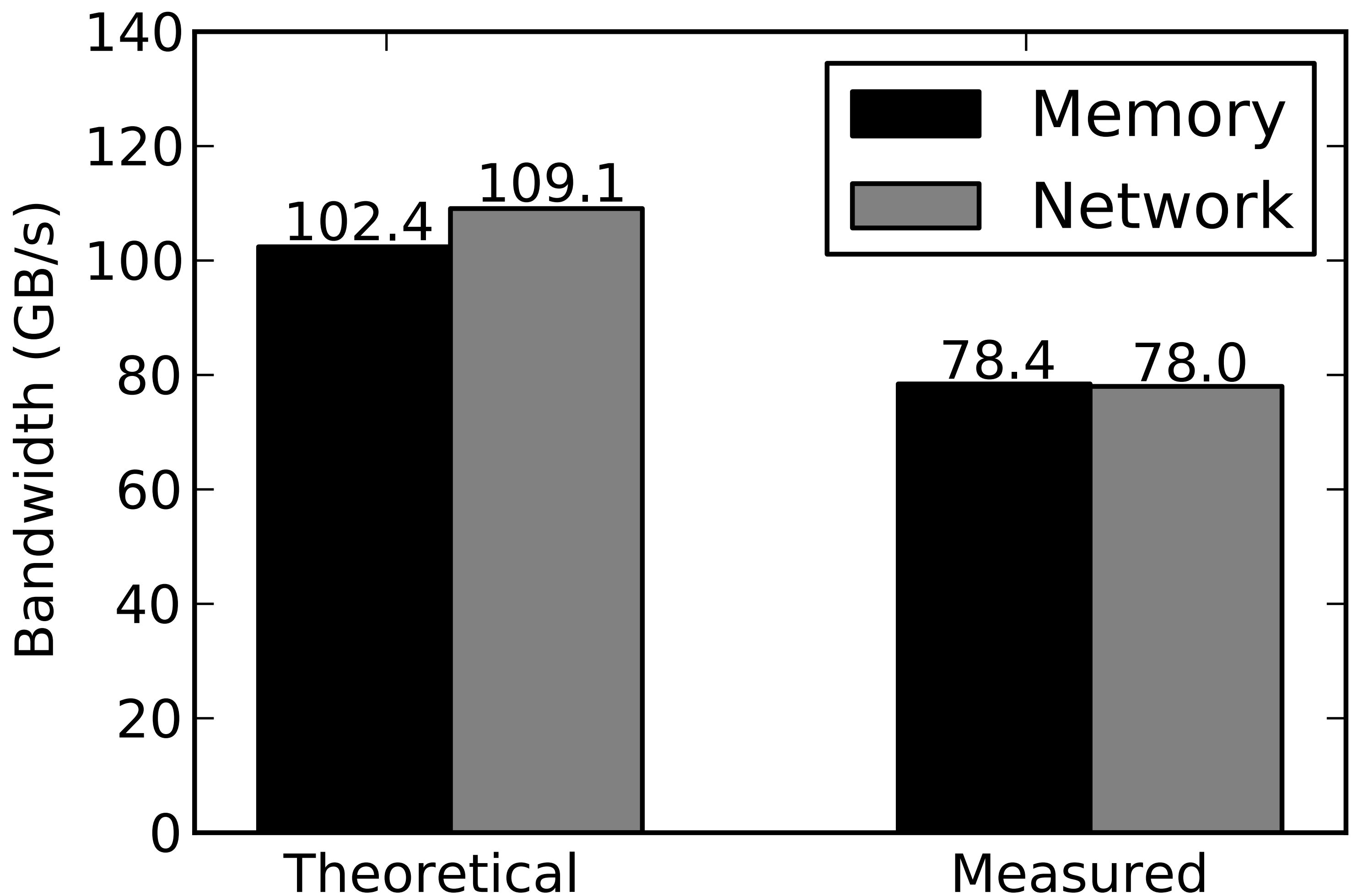
Assuming equal read/write workload



Assuming equal read/write workload


RAM
RAM
Memory Bus (4 channels)
51.2 GB/s Half-duplex (total)
Memory Controler
Memory Controler
CPU 1
Core 1
Core 2
Core 3
Core 4
Core 5
Core 6
Core 7
Core 8
QPI (2 links)
32 GB/s
Full-duplex
64 GB/s total
CPU 2
Core 1
Core 2
Core 3
Core 4
Core 5
Core 6
Core 7
Core 8
PCIe Gen3 (40 lanes), 39.4 GB/s full-duplex
78.8 GB/s total
Dualport RNIC (16 PCI lanes)
Dualport RNIC (16 PCI lanes)
Dualport RNIC (16 PCI lanes)
Dualport RNIC (16 PCI lanes)
Two FDR 4x (2ports)
27.3 GB/s full-duplex
54.5 GB/s total
Two FDR 4x (2ports)
27.3 GB/s full-duplex
54.5 GB/s total

Total Bandwidth over 2 Sockets with **Four** Dual-port FDR 4x NICs



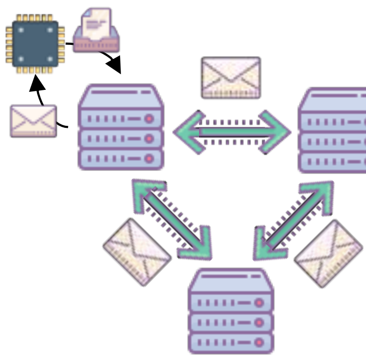
2 servers, each with: Xeon E5-2660 v2 CPUs, 256GB of DDR3-1600 RAM,
FOUR 2-port InfiniBand FDR 4x NIC per machine

Intel is working on 400Gbs NIC

"There is no shared nothing if you have 400Gbs NIC"

--- David Cohen, Intel

Why are distributed transactions considered not scalable

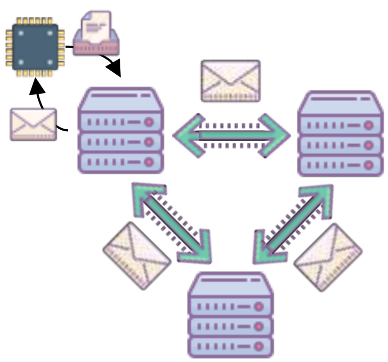


Message Overhead

Using up network BW

Higher latency of distributed trxs

But with the next generation of networks (e.g., Infiniband HDR)



Message Overhead

Low/zero CPU overhead

(2-sided RDMA: low overhead 1-sided RDMA: zero overhead.)

Using up network BW

High bandwidth

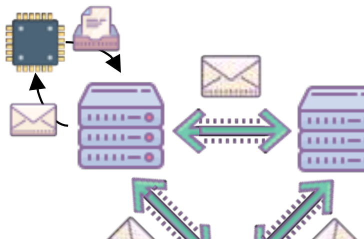
(is getting on par with memory BW)

Higher latency of distributed trxs

Low latency

(5x-30x better than 10Gb Ethernet)

But with the next generation of networks (e.g., Infiniband HDR)



Is **Higher Latency** a **problem**?

1. No contention point → No
2. Contention point → Kinda,

   but in this case the workload is per definition not scalable (more later)

Higher latency of distributed trxs

Low latency

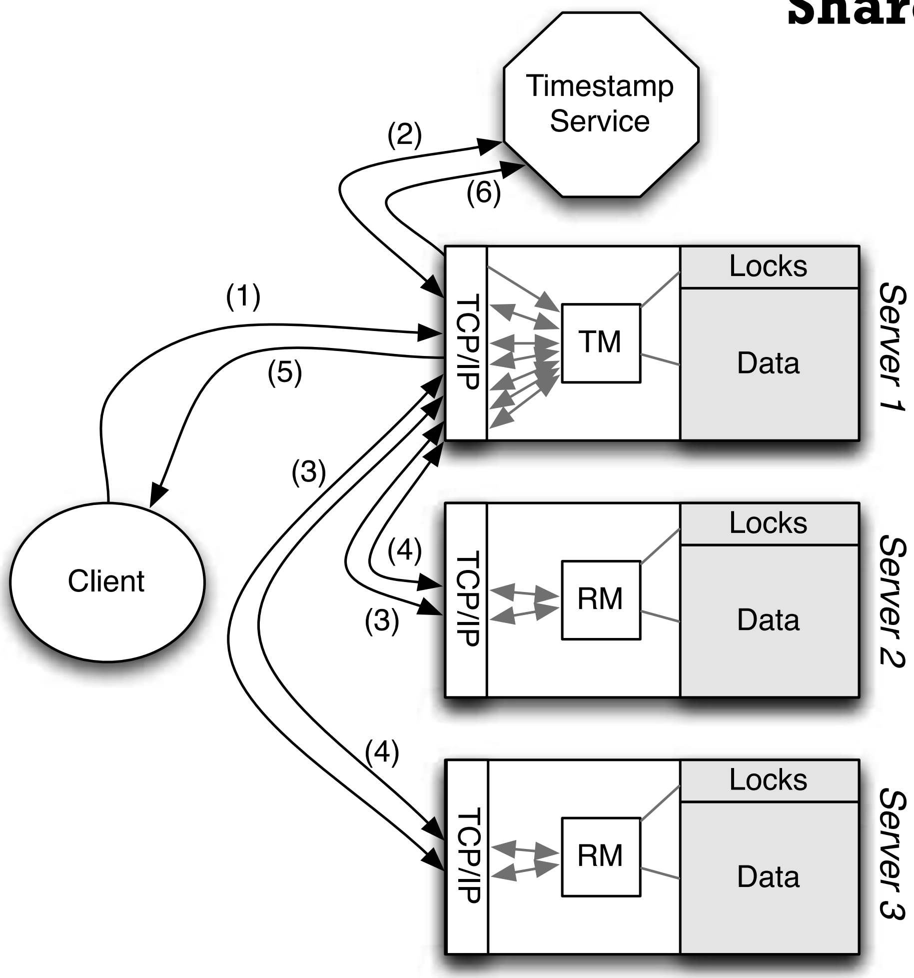
(5x-30x better than 10Gb Ethernet)

(2-

1-sided RDMA: zero overhead.) memory BW)

Can we simply **upgrade** the network with **IPoIB** and **we are done**???

Quick Experiment



Shared-nothing In-Memory Distributed DB

- Data partitioned horizontally on servers
- One server acts as the trx's coordinator
- Generalized Snapshot Isolation guarantees
- 100% distributed transactions

1 NIC (single-port) Ethernet

1.25GB/s

IP over IB

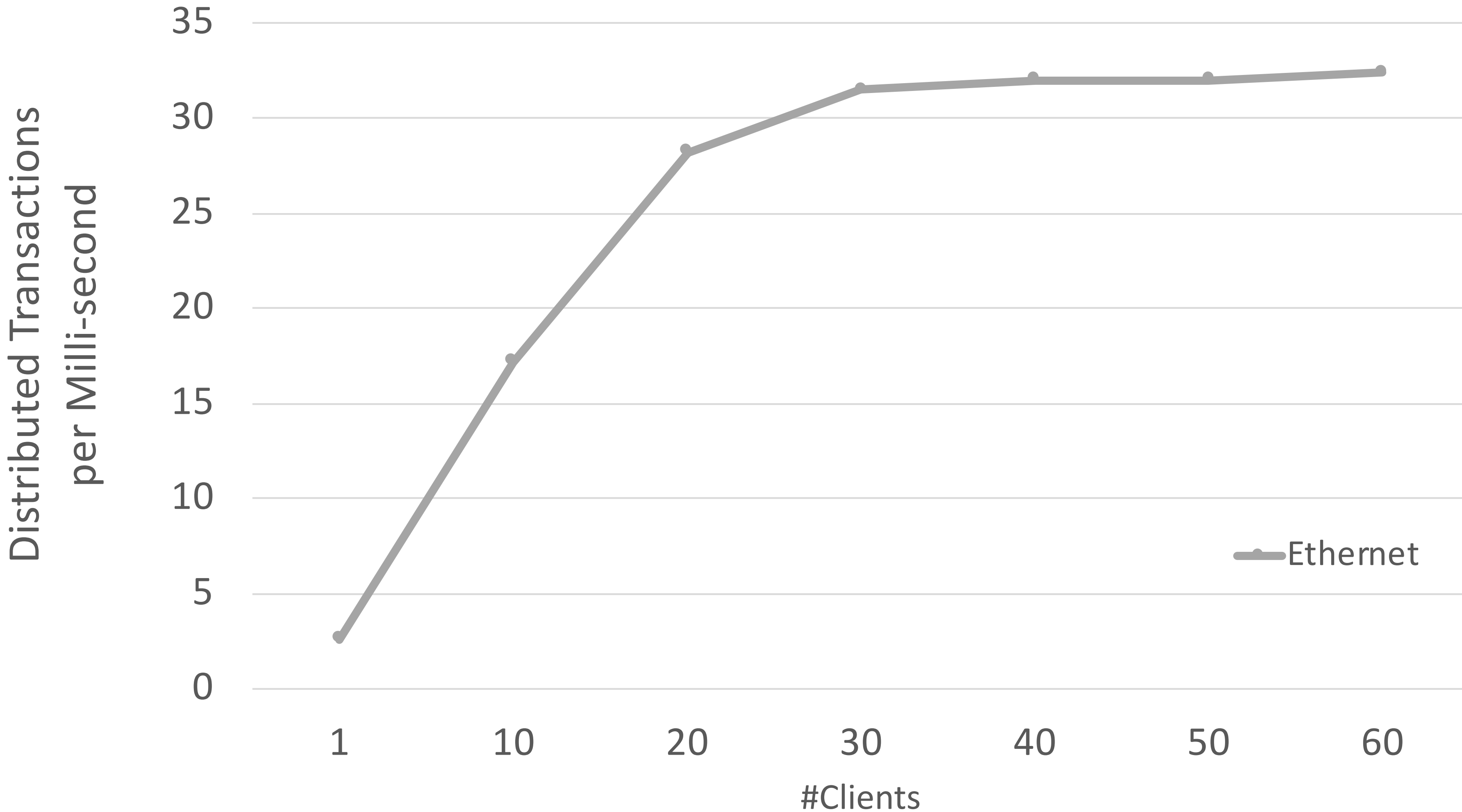
1x NIC (dual-port) IB FDR 4x

13.64 GB/s

S. Elnikety et al. Database replication using generalized snapshot isolation. In SRDS, 2005

Y. Lin et al. Middleware based data replication providing snapshot isolation. In SIGMOD, 2005

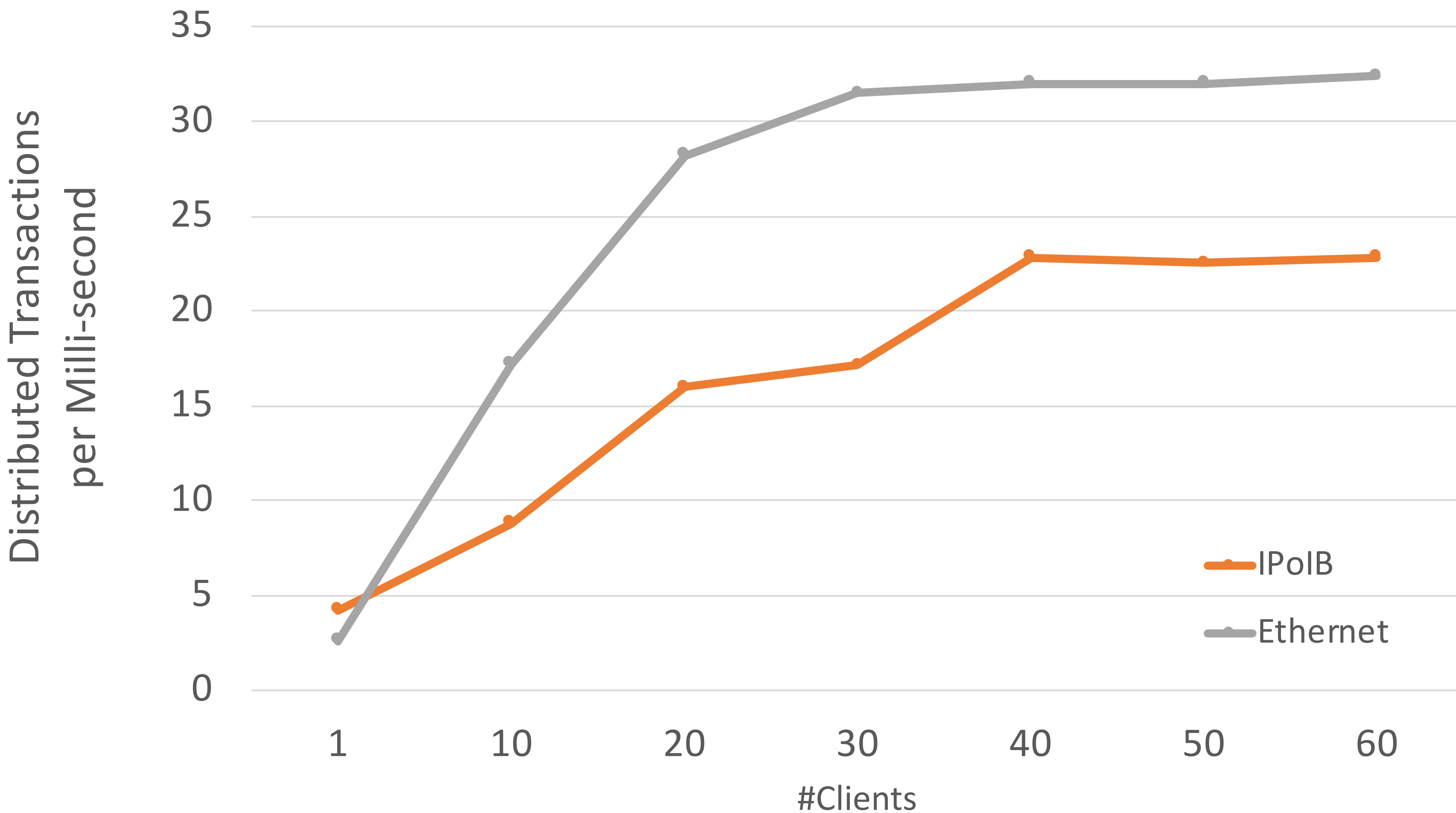
Ethernet vs IPoIB Transaction Throughput (in 1000 trx/s)



8 servers, each with: Xeon E5-2660 v2 CPUs, 256GB of DDR3-1600 RAM, one 2-port InfiniBand FDR 4x NIC

Workload: 100% distributed transactions, every transaction spans 3 servers

Ethernet vs IPoIB Transaction Throughput (in 1000 trx/s)



8 servers, each with: Xeon E5-2660 v2 CPUs, 256GB of DDR3-1600 RAM, one 2-port InfiniBand FDR 4x NIC
Workload: 100% distributed transactions, every transaction spans 3 servers

What are the key problems

- **Remote-Direct-Memory-Access (RDMA)**
  → Requires a new memory layout

- **New bottlenecks**
  (e.g., creating global timestamps for snapshot isolation)

New-Generation Network: RDMA

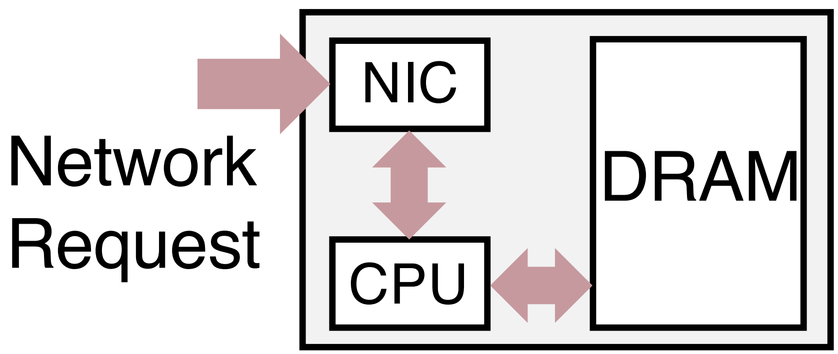
RDMA: **Remote Direct Memory Access**

Three modes: IPoIB, one-sided, two-sided

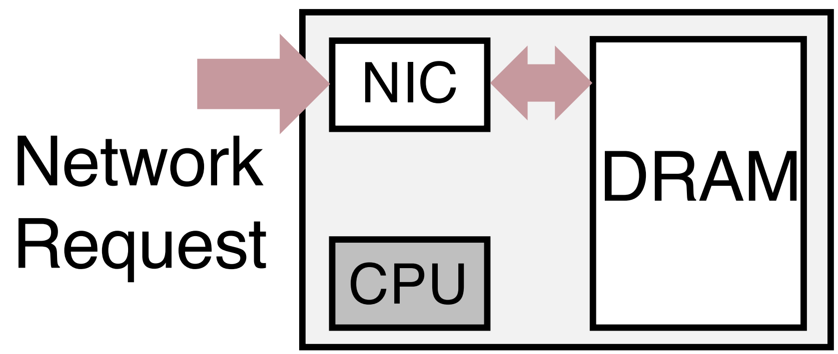
New-Generation Network: RDMA

RDMA: **Remote Direct Memory Access**

Three modes: IPoIB, **one-sided**, two-sided

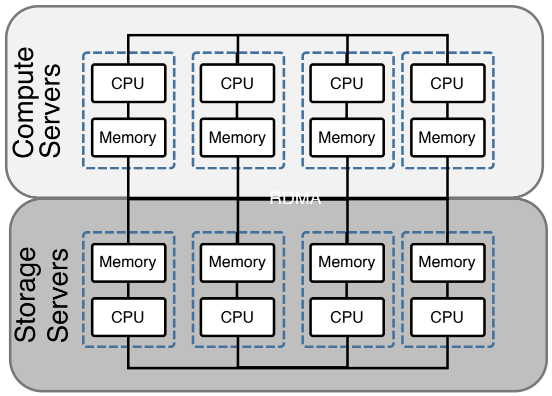


Conventional network



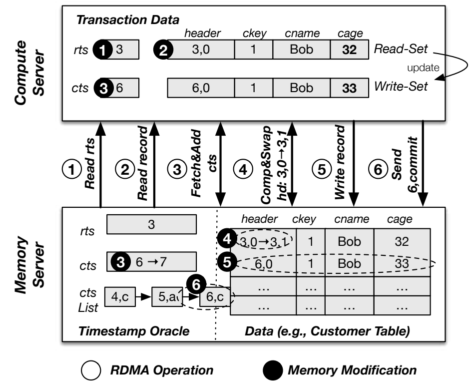
1-sided RDMA

4 Key Results from 4 Years



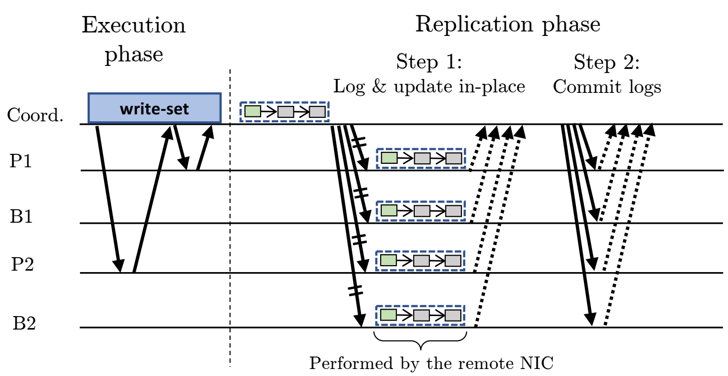
NAM-Architecture

Carsten Binnig, Andrew Crotty, Alex Galakatos, Tim Kraska, Erfan Zamanian: **The End of Slow Networks: It's Time for a Redesign.** PVLDB 9(7): 528-539 (2016)



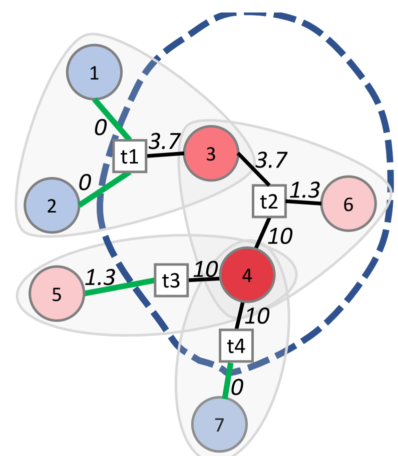
Scalable SI Protocol

Erfan Zamanian et al: **The End of a Myth: Distributed Transaction Can Scale.** PVLDB 10(6): 685-696 (2017)



Active Replication

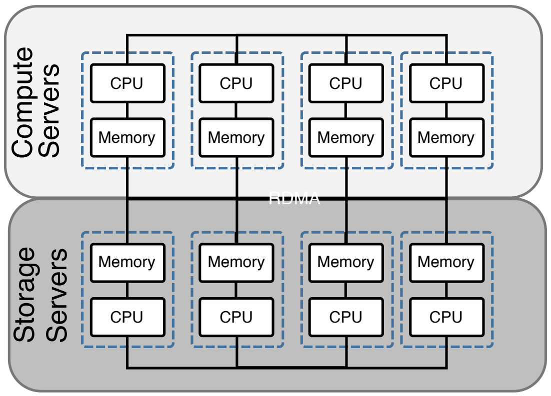
Erfan Zamanian et al: **Rethinking Database High Availability with RDMA Networks**. PVLDB 12(11): 1637-1650 (2019)



Contention-centric Clustering

Erfan Zamanian et al: Chiller: Contention-centric Transaction Execution and Data Partitioning for Fast Networks. CoRR abs/1811.12204 (2018)

4 Key Results from 4 Years



NAM-Architecture

Carsten Binnig, Andrew Crotty, Alex Galakatos, Tim Kraska, Erfan Zamanian: **The End of Slow Networks: It's Time for a Redesign.** PVLDB 9(7): 528-539 (2016)

Scalable SI Protocol

Erfan Zamanian et al: **The End of a Myth: Distributed Transaction Can Scale.** PVLDB 10(6): 685-696 (2017)

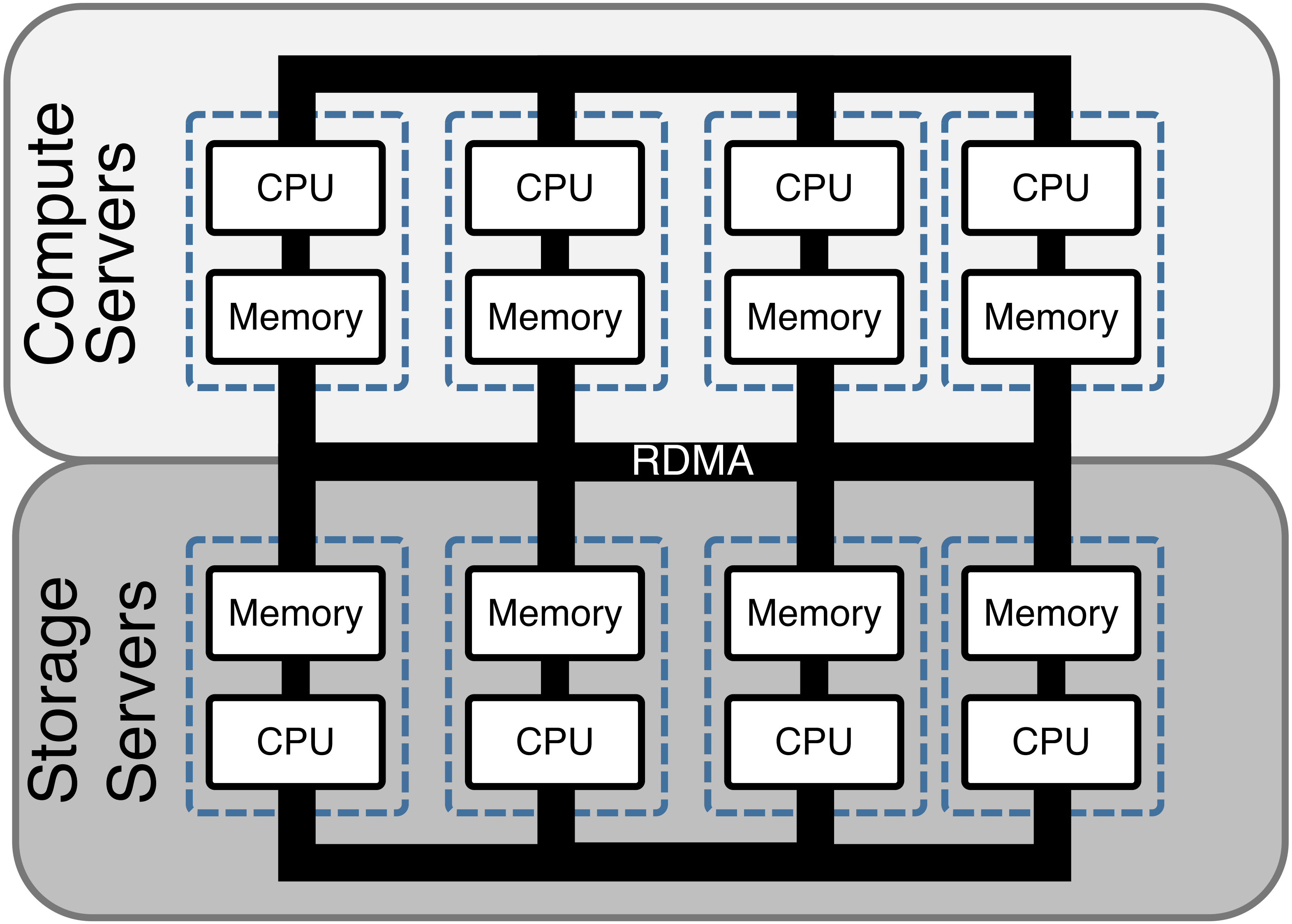
Active Replication

Erfan Zamanian et al: **Rethinking Database High Availability with RDMA Networks.** PVLDB 12(11): 1637-1650 (2019)

Contention-centric Clustering

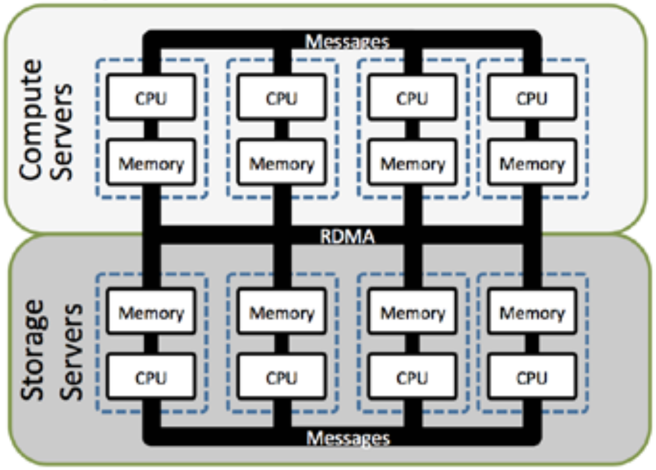
Erfan Zamanian et al: Chiller: Contention-centric Transaction Execution and Data Partitioning for Fast Networks. CoRR abs/1811.12204 (2018)

The Network-Attached-Memory (NAM) Architecture



Carsten Binnig, Andrew Crotty, Alex Galakatos, Tim Kraska, Erfan Zamanian: **The End of Slow Networks: It's Time for a Redesign.** PVLDB 9(7): 528-539 (2016)

NAM Architecture



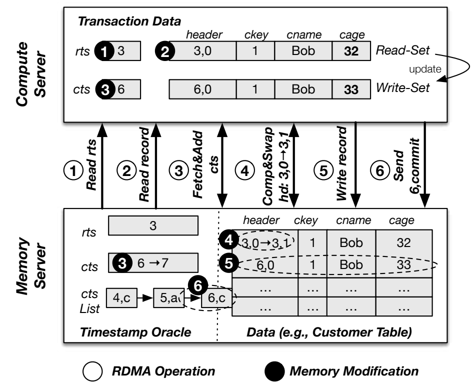
- **Memory Servers**
  - Expose their memory to compute servers
  - Store multiple versions of each record
- **All trxs are distributed by default**
- **A logical separation, not a physical**
  - Physical co-location to leverage locality as an optimization
- **Optimized for Remote-Direct-Memory-Access (RDMA)** (no messages) → avoids CPU overhead
- Benefits:
  - Compute and storage can **independently** scale
  - Can efficiently handle **data imbalance**
  - **Helps us to understand how far RDMA and fast networks can go and where messages are really needed.**

4 Key Results from 4 Years

skip

NAM-Architecture

Carsten Binnig, Andrew Crotty, Alex Galakatos, Tim Kraska, Erfan Zamanian: **The End of Slow Networks: It's Time for a Redesign.** PVLDB 9(7): 528-539 (2016)



Scalable SI Protocol

Erfan Zamanian et al: **The End of a Myth: Distributed Transaction Can Scale.** PVLDB 10(6): 685-696 (2017)

Active Replication

Erfan Zamanian et al: **Rethinking Database High Availability with RDMA Networks.** PVLDB 12(11): 1637-1650 (2019)

Contention-centric Clustering

Erfan Zamanian et al: Chiller: Contention-centric Transaction Execution and Data Partitioning for Fast Networks. CoRR abs/1811.12204 (2018)

Goal

Explore if we can design an entire SI protocol using one-sided RDMA read/write

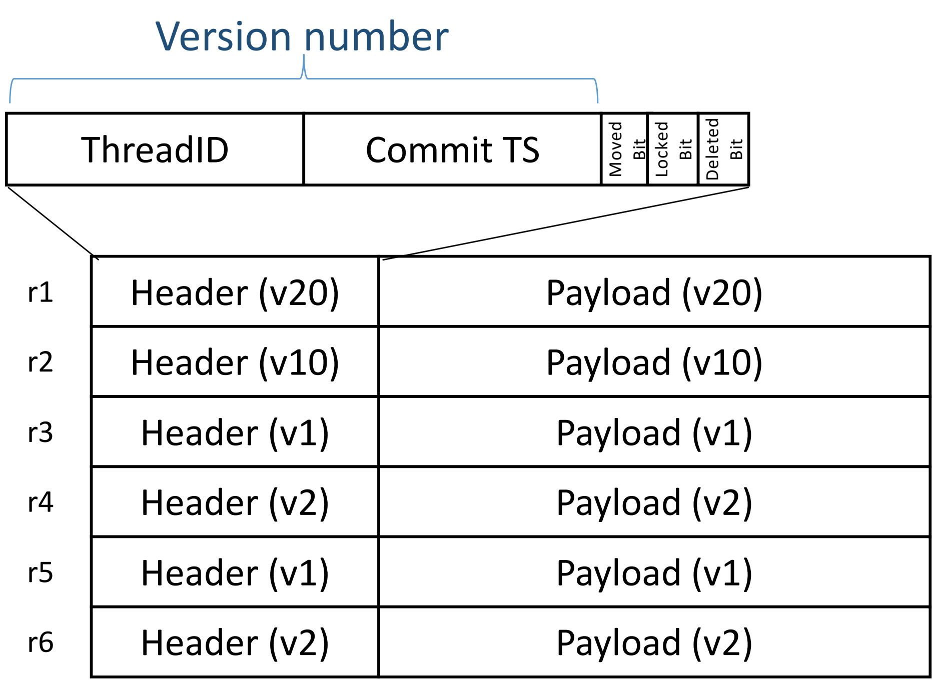
(RPC/2-sided is an optimization)

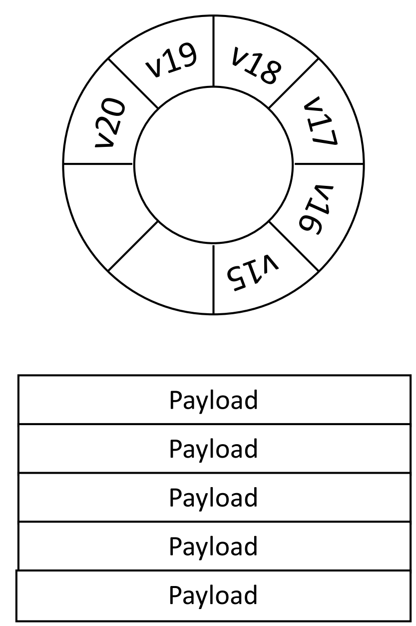
An RDMA Friendly Memory Layout

1) Fast scan and retrieval of single key for the most recent version

- Record offsets as identifiers (memory location is known from the record-id)
- Requires to re-organize data from time to time

2) Circular buffer containing version number and payload pointers for older versions





See paper for more details and other optimizations

Timestamp Oracle

Needed to determine the currently valid version-id

Global counter updated w/ atomic RDMA

- Doesn't scale with number of servers
- As slow as the slowest transaction
- Requires either
  - Several roundtrip of atomic RDMA
  - 2-sided send/receive

Solution: Timestamp Vector

- Similar to vector-clocks but with two important differences:
  1. **Read-TS is a vector, a version consist of a <single thread-id, commit-counter> pair**
     - Version numbers are small (2 integers)
     - Faster comparisons
  2. **Monotonically increasing:** no branching, no need for epochs
- Still SI-guarantees
- Avoids problems with long-running transactions and stale-reads

Distr. Shared Storage

Timestamp

Thread ID	Last commit TS
1	5
2	3
3	11

Data

(versions' headers & payloads)

* For monotonicity NIC has to ensure left-to-right write order and no DMA re-ordering

The Basic RSI Protocol

Client (Compute Server)

1) RDMA-READ Timestamp
2) for r ∈ read-set
    1) RDMA-READ current version
    2) Check Version, if needed RDMA-READ older version from buffer
3) Increment local thread TS
4) for w ∈ write-set:
    1) RDMA-Compare-And-Swap record header (set lock-bit).
    2) If compare-and-swap fails or latest version is new, abort and revert changes
    3) If not move, RDMA-WRITE record to circular buffer
    4) RDMA-WRITE version in place
5) RDMA-WRITE to update Thread-Counter

Distr. Shared Storage

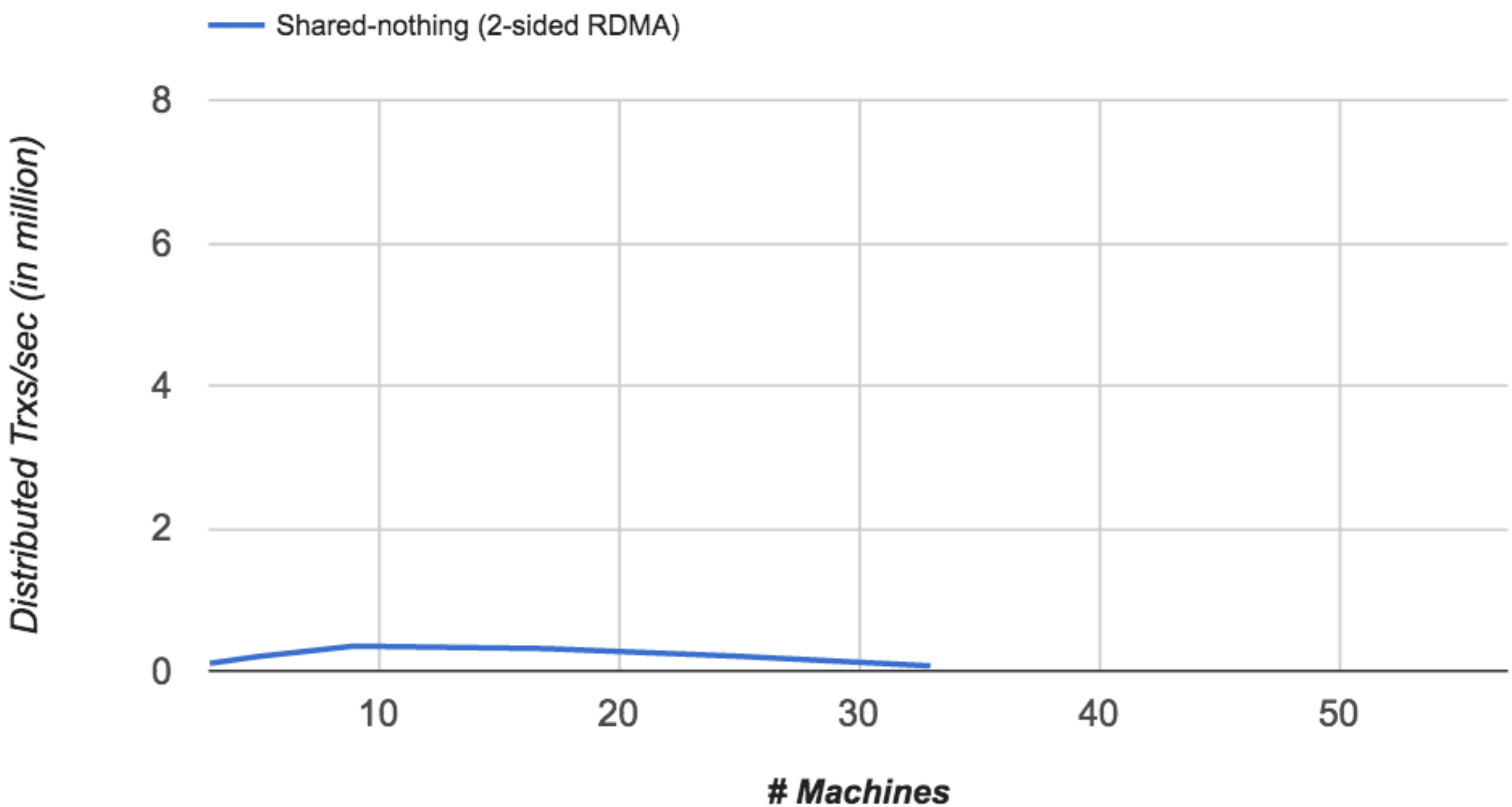
Timestamp

Thread ID	Last commit TS
1	5
2	2
3	11

Data

(versions' headers & payloads)

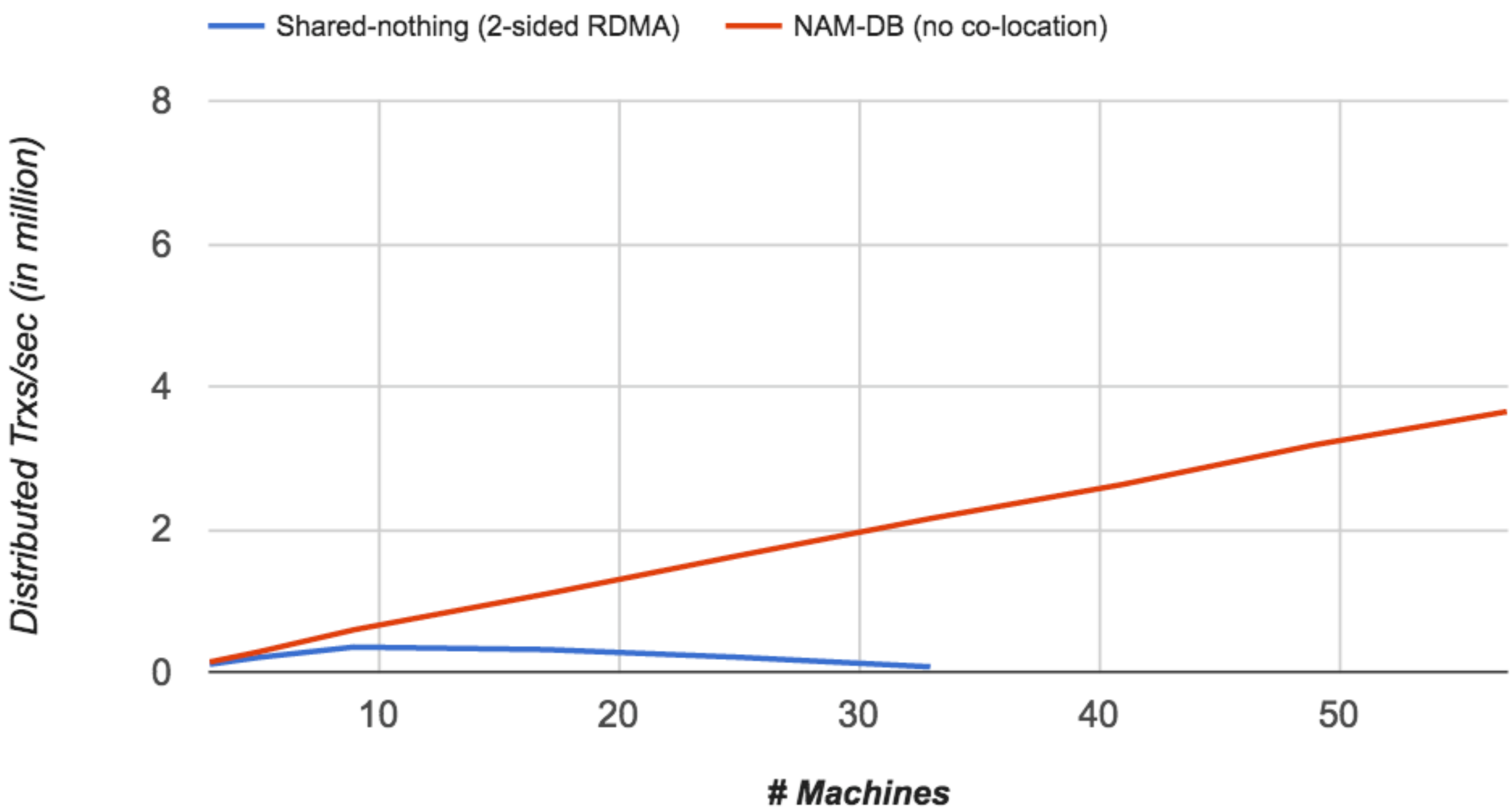
Scale-Out Experiment on TPC-C



Workload: standard TPC-C, with 50 warehouses per server.
27 machines of type: Two Xeon E7-4820 processors (each with 8 cores), 128 GB RAM
28 machines of type: Two Xeon E5-2660 processors (each with 8 cores), 256 GB RAM

Scale-Out Experiment on TPC-C

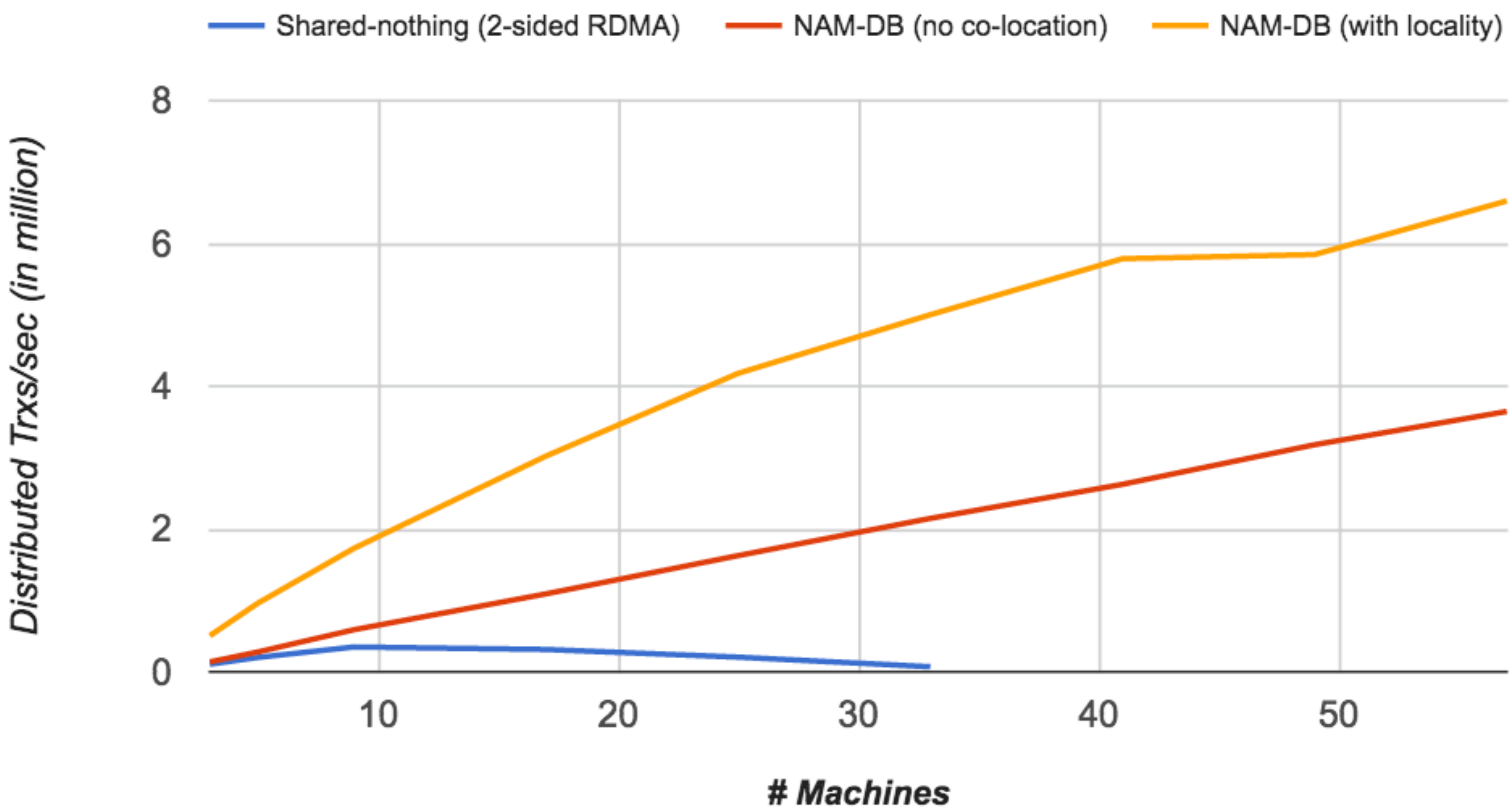
All Distributed transactions



Workload: standard TPC-C, with 50 warehouses per server.
27 machines of type: Two Xeon E7-4820 processors (each with 8 cores), 128 GB RAM
28 machines of type: Two Xeon E5-2660 processors (each with 8 cores), 256 GB RAM

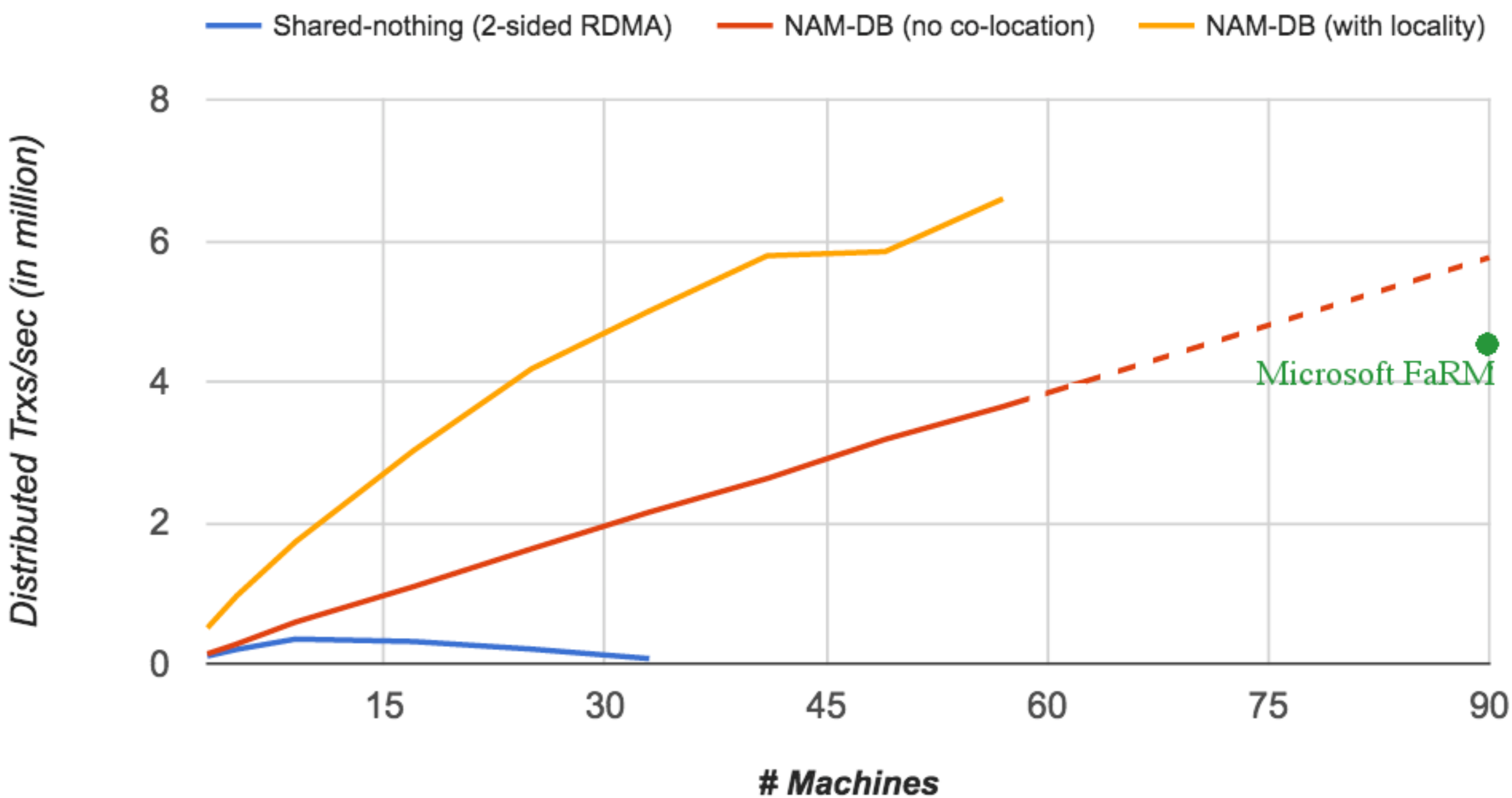
Scale-Out Experiment on TPC-C

90% local transactions, 10% distributed



Workload: standard TPC-C, with 50 warehouses per server.
27 machines of type: Two Xeon E7-4820 processors (each with 8 cores), 128 GB RAM
28 machines of type: Two Xeon E5-2660 processors (each with 8 cores), 256 GB RAM

Scale-Out Experiment on TPC-C



Workload: standard TPC-C, with 50 warehouses per server.
27 machines of type: Two Xeon E7-4820 processors (each with 8 cores), 128 GB RAM
28 machines of type: Two Xeon E5-2660 processors (each with 8 cores), 256 GB RAM
FaRM: From the paper “No compromises: distributed transactions with consistency, availability, and performance”

4 Key Results from 4 Years

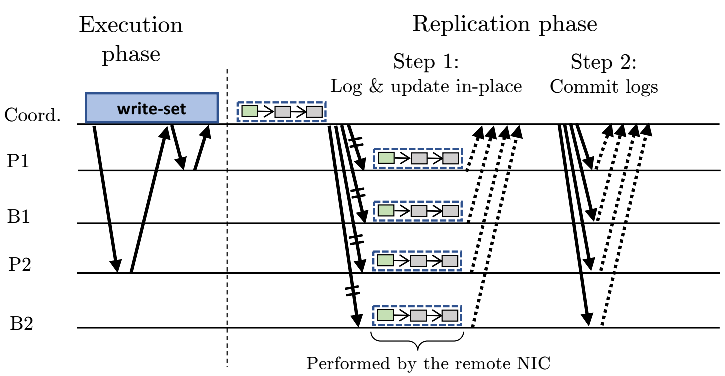
skip

NAM-Architecture

Carsten Binnig, Andrew Crotty, Alex Galakatos, Tim Kraska, Erfan Zamanian: **The End of Slow Networks: It's Time for a Redesign.** PVLDB 9(7): 528-539 (2016)

Scalable SI Protocol

Erfan Zamanian et al: **The End of a Myth: Distributed Transaction Can Scale.** PVLDB 10(6): 685-696 (2017)



Active Replication

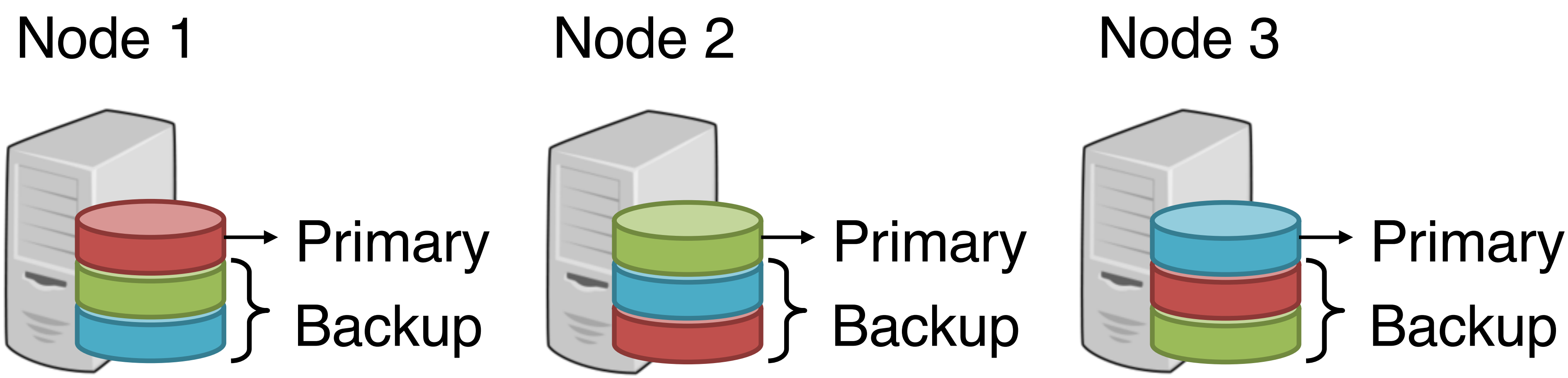
Erfan Zamanian et al: **Rethinking Database High Availability with RDMA Networks.** PVLDB 12(11): 1637-1650 (2019)

Contention-centric Clustering

Erfan Zamanian et al: Chiller: Contention-centric Transaction Execution and Data Partitioning for Fast Networks. CoRR abs/1811.12204 (2018)

Context of this Work

- Cluster over Local area network (LAN)
- Striped master model

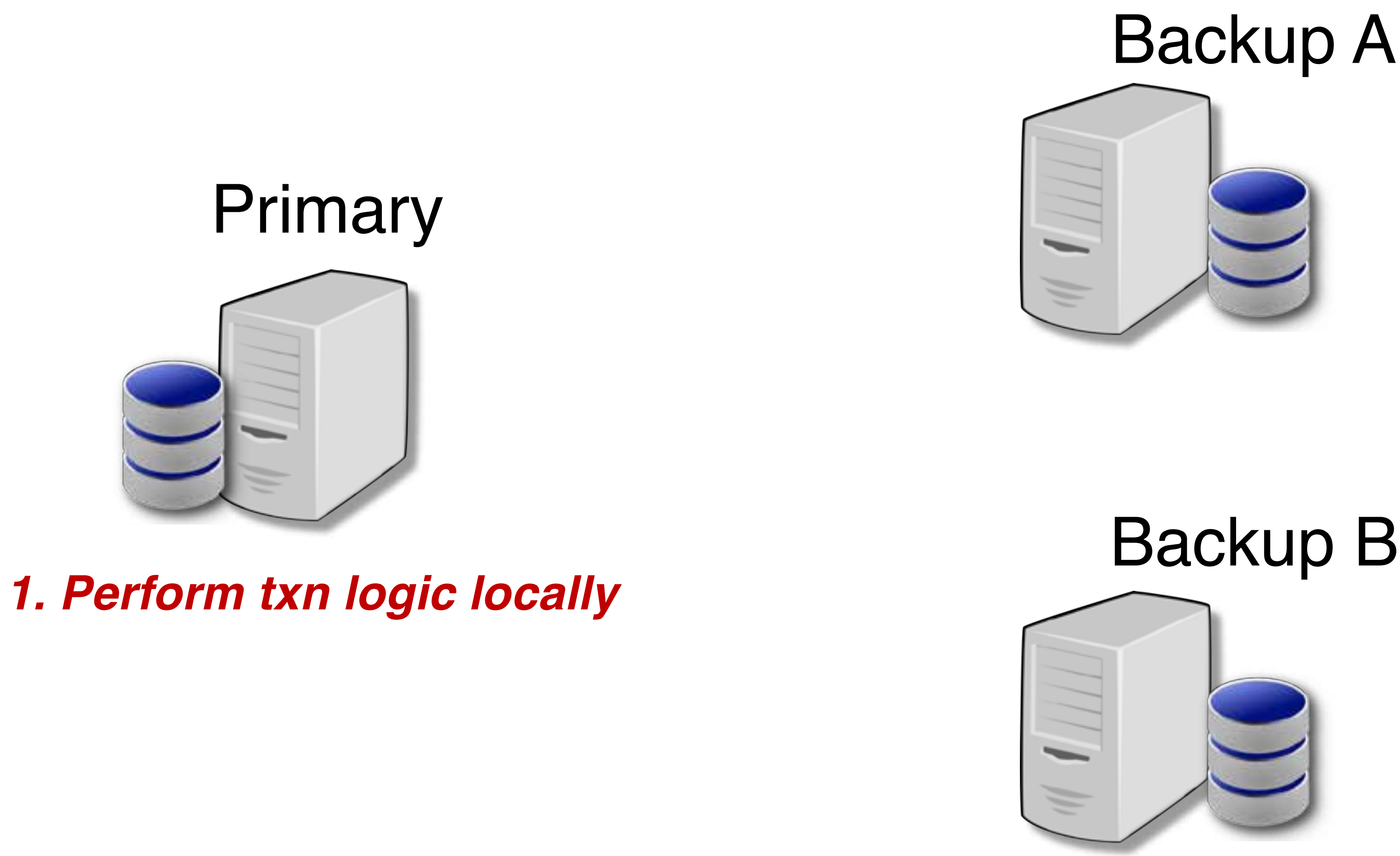


All Existing Replication Protocols Trade CPU for Network Bandwidth

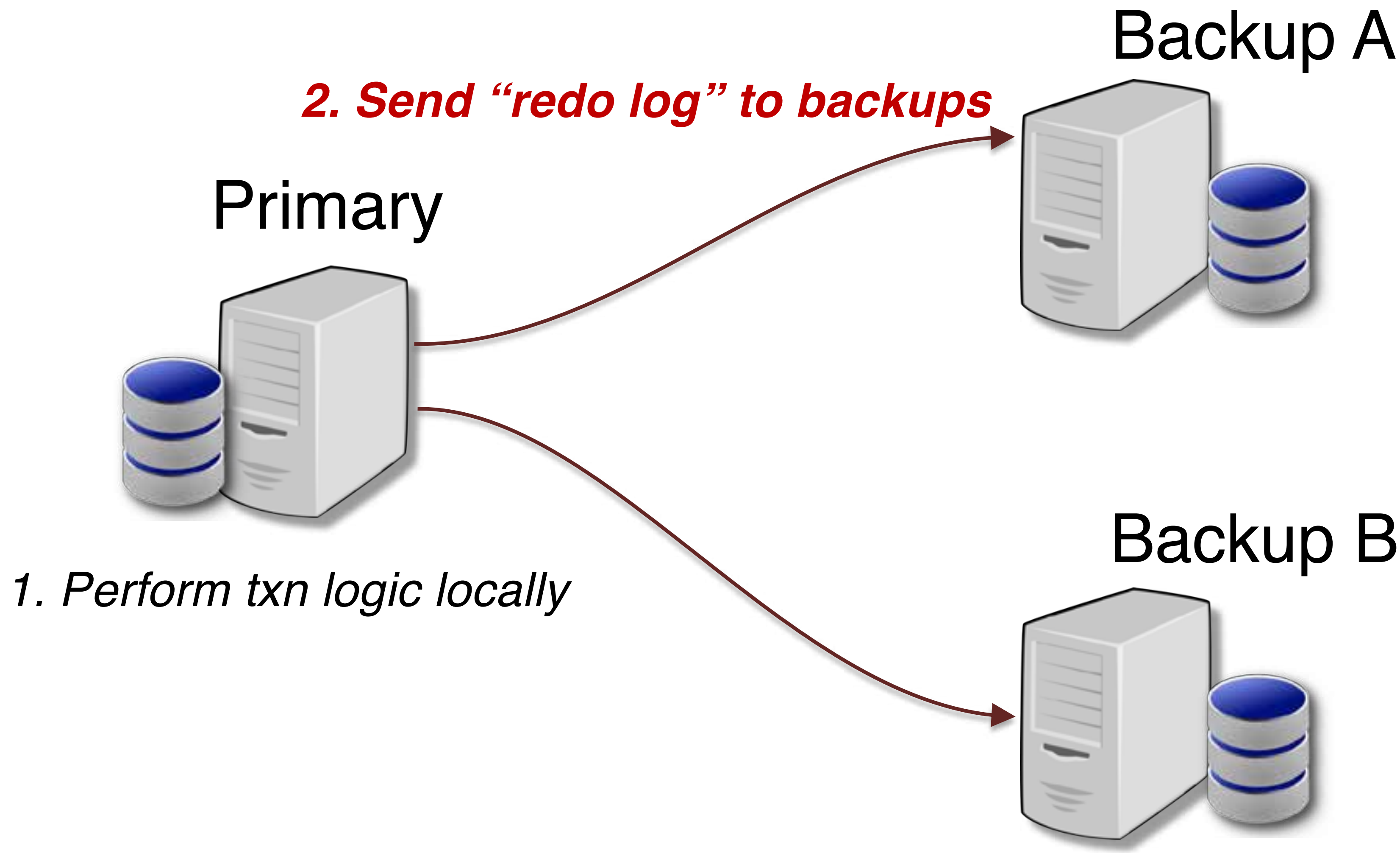
CPU demand

Network demand

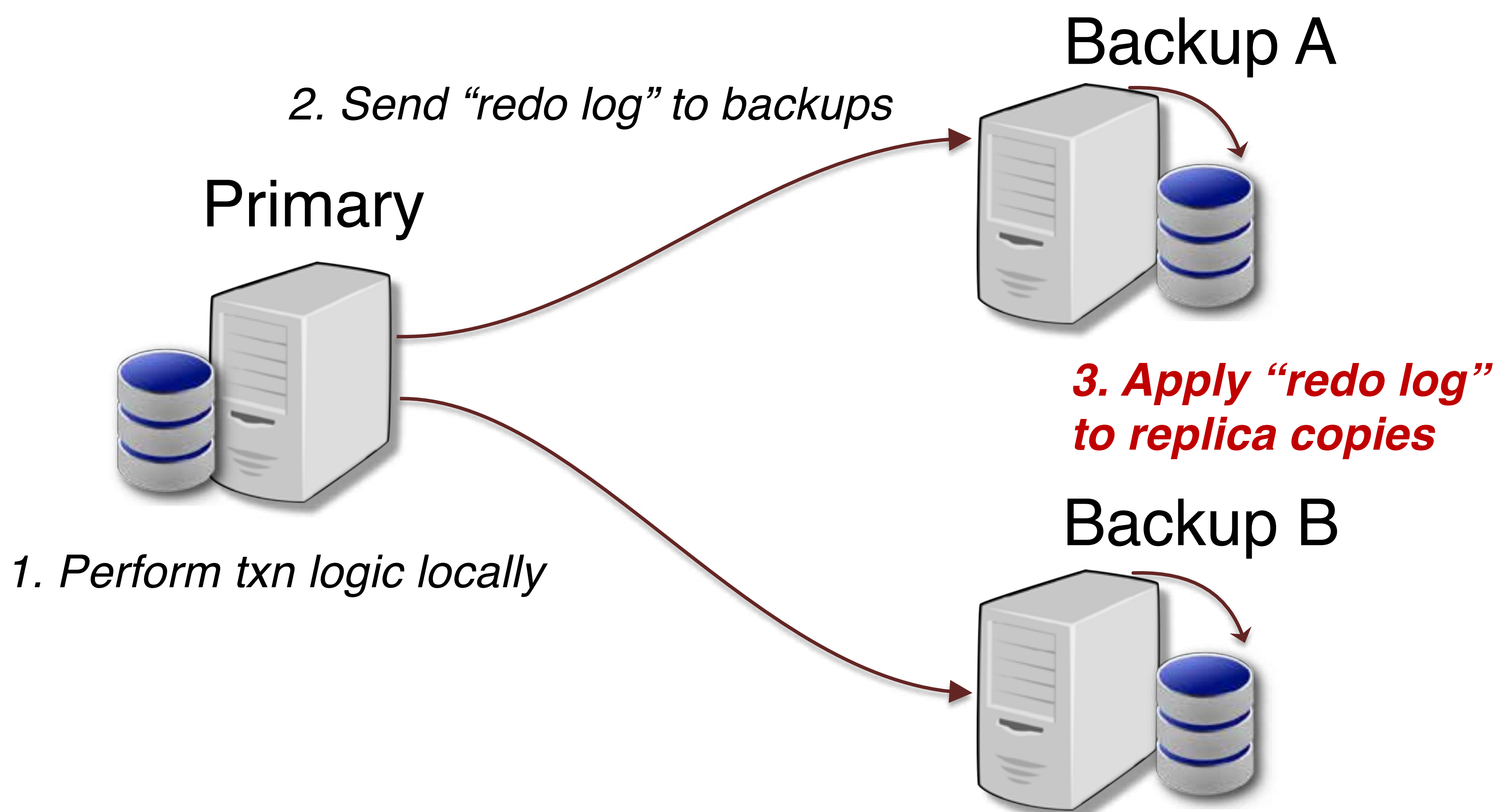
Active-Passive Replication
(i.e. log shipping)



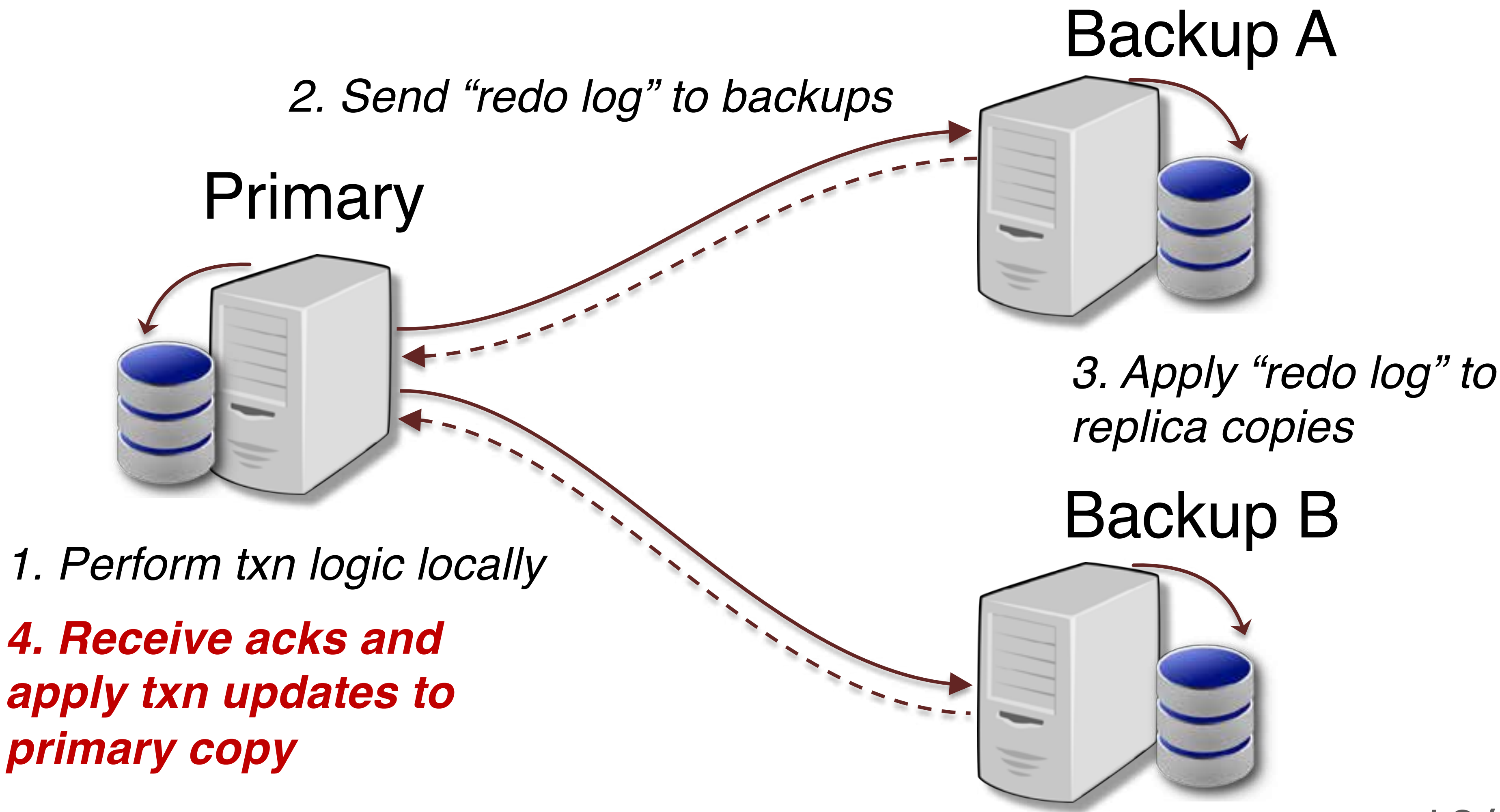
Active-Passive Replication (i.e. log shipping)



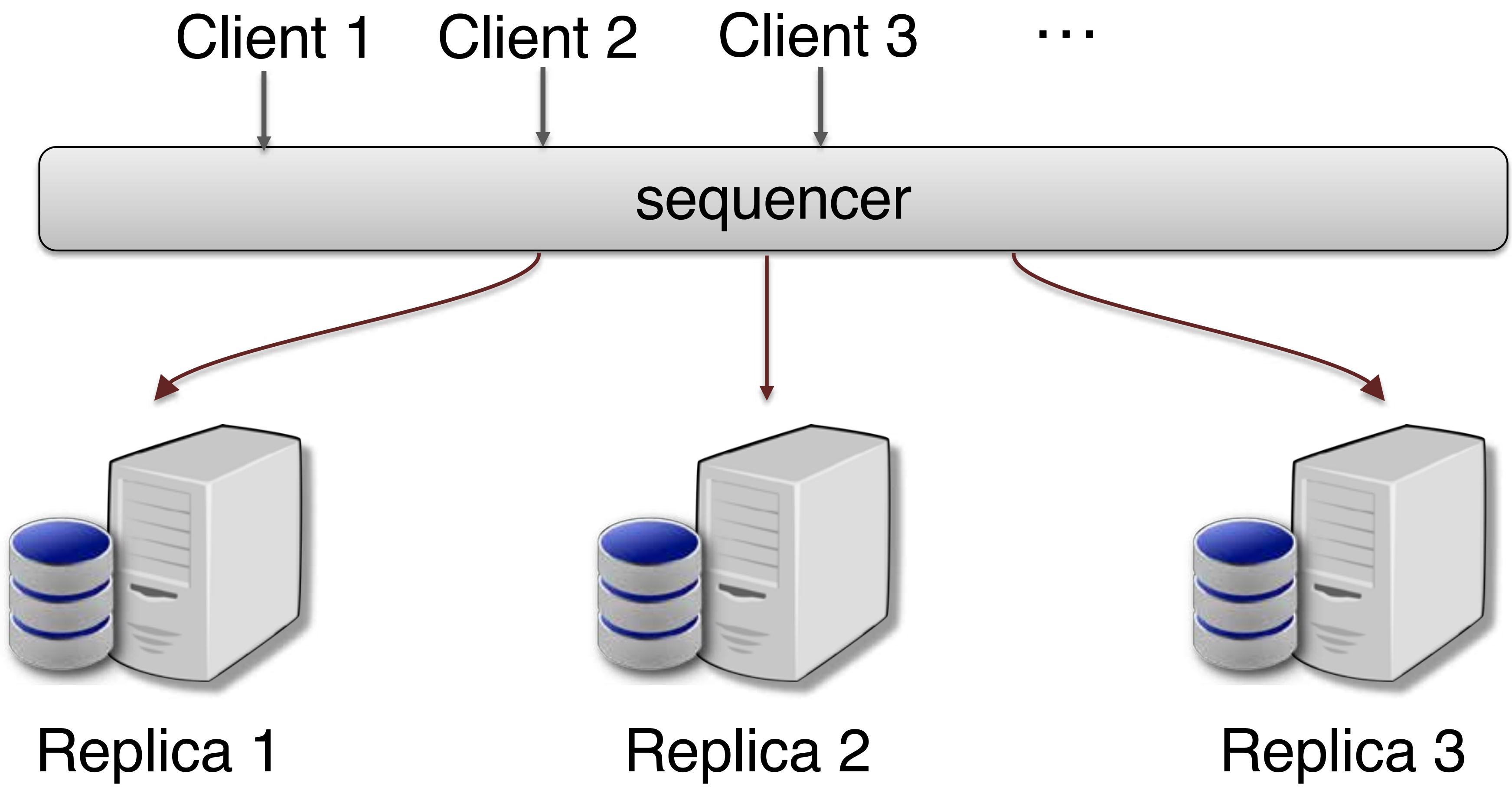
Active-Passive Replication (i.e. log shipping)



Active-Passive Replication (i.e. log shipping)



Active-Active Replication



- Sequencer sends **inputs of transactions** to replicas
- Each replica runs transactions **deterministically**

CPU vs. Network Tradeoff

	Active-Passive	Active-Active

CPU
demand

Network
demand

CPU vs. Network Tradeoff

	Active-Passive		Active-Active
CPU Demand	**Replay logs**	**<**	**Duplicate execution**

CPU demand

Network demand

CPU vs. Network Tradeoff

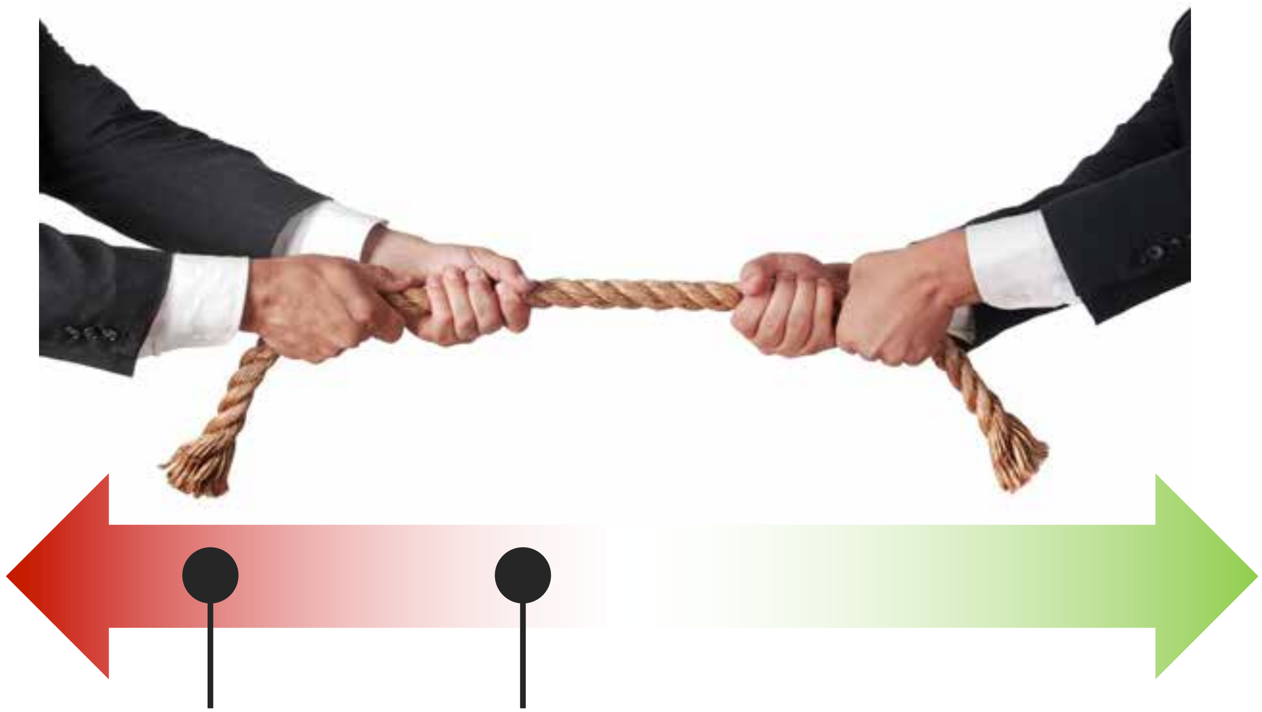
	Active-Passive		Active-Active
CPU Demand	**Replay logs**	**<**	**Duplicate execution**
Network Demand	**Send logs**	**>**	**Send input**

CPU demand

Network demand

CPU vs. Network Tradeoff

	Active-Passive		Active-Active
CPU Demand	**Replay logs**	**<**	**Duplicate execution**
Network Demand	**Send logs**	**>**	**Send input**



New Idea: **Active Memory** Replication

Key idea: Coordinator **directly updates** memory states of backup nodes using **one-sided RDMA**

CPU demand

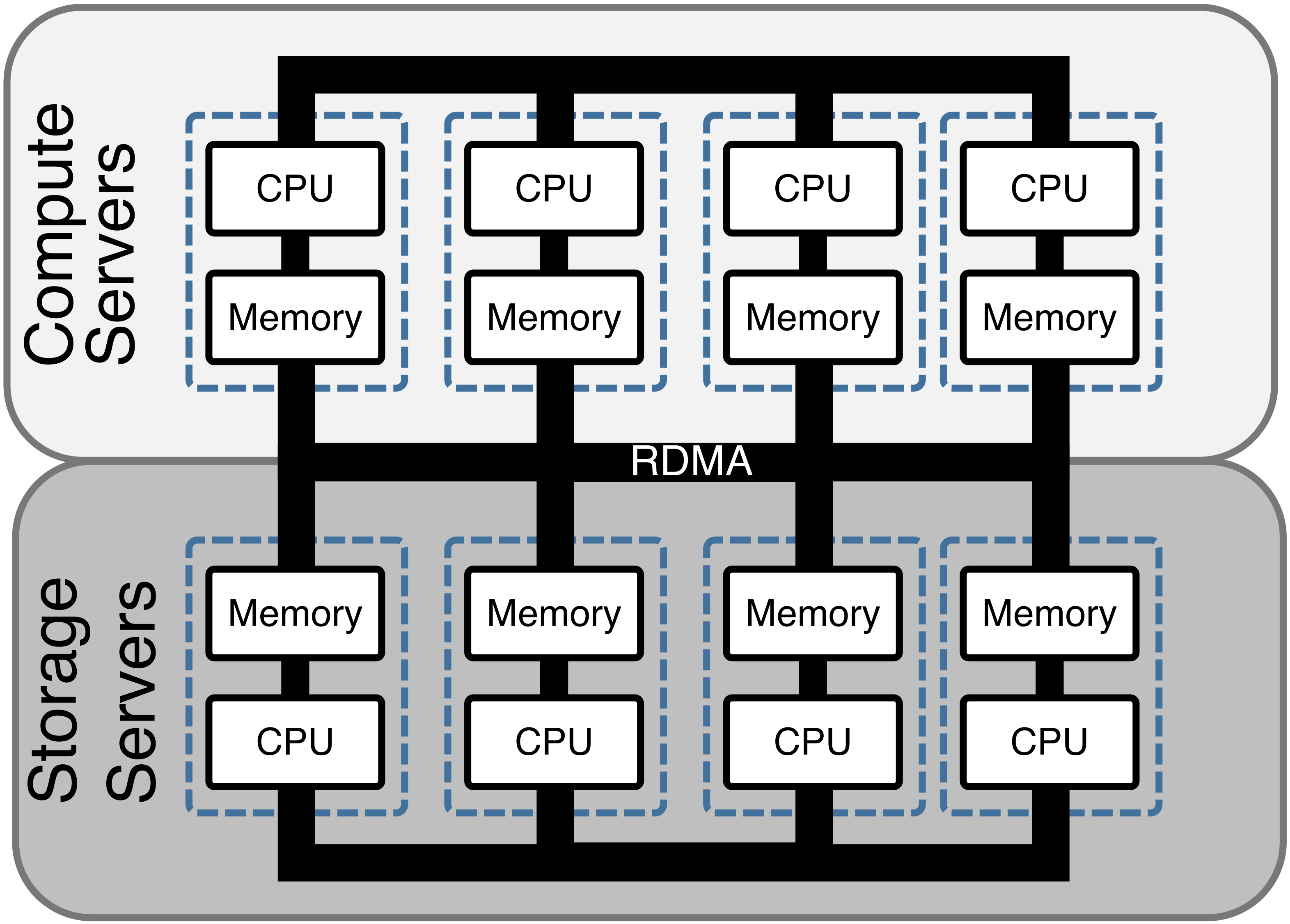
Network demand

Active/Active

Active/Passive (Log Shipping)

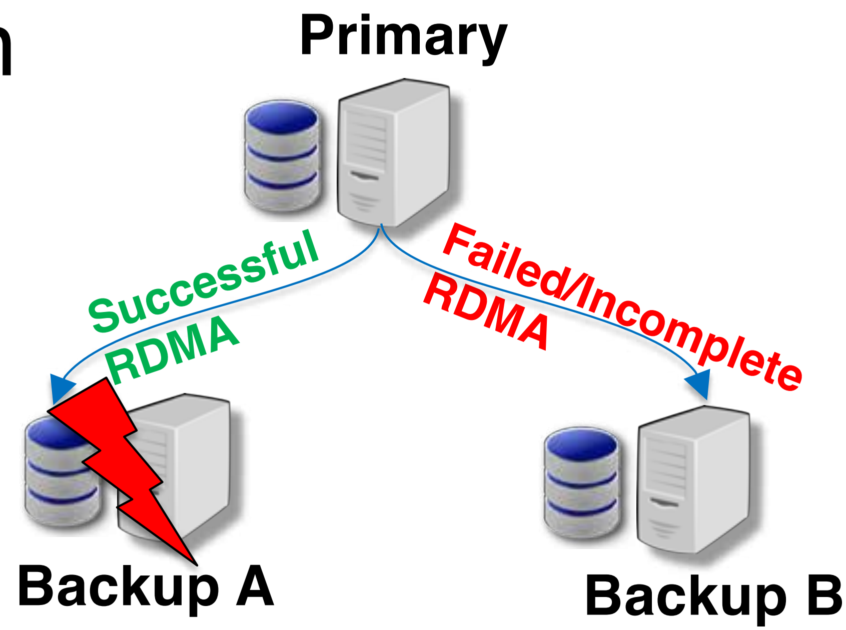
Active Memory

The Network-Attached-Memory (NAM) Architecture



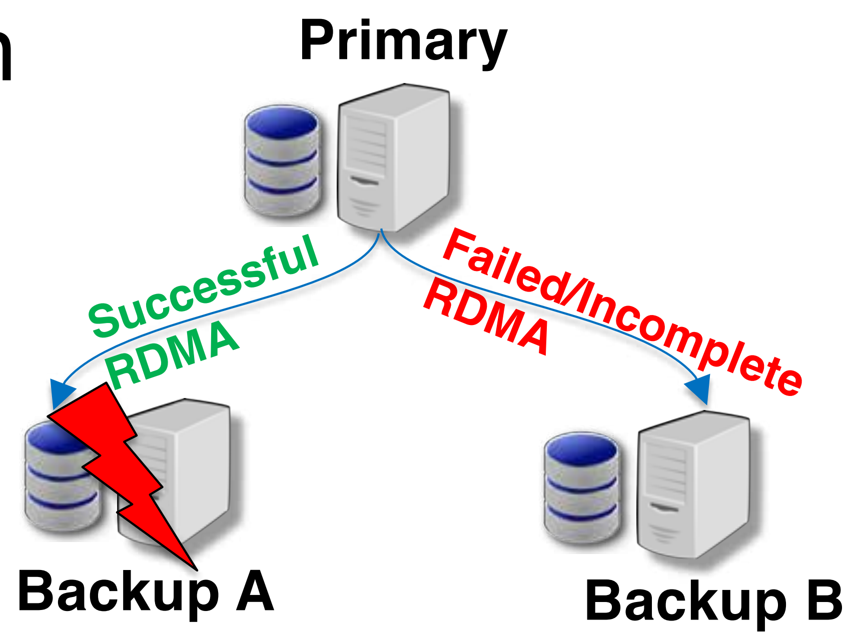
Challenge: Fault Tolerance

All or nothing replication



Challenge: Fault Tolerance

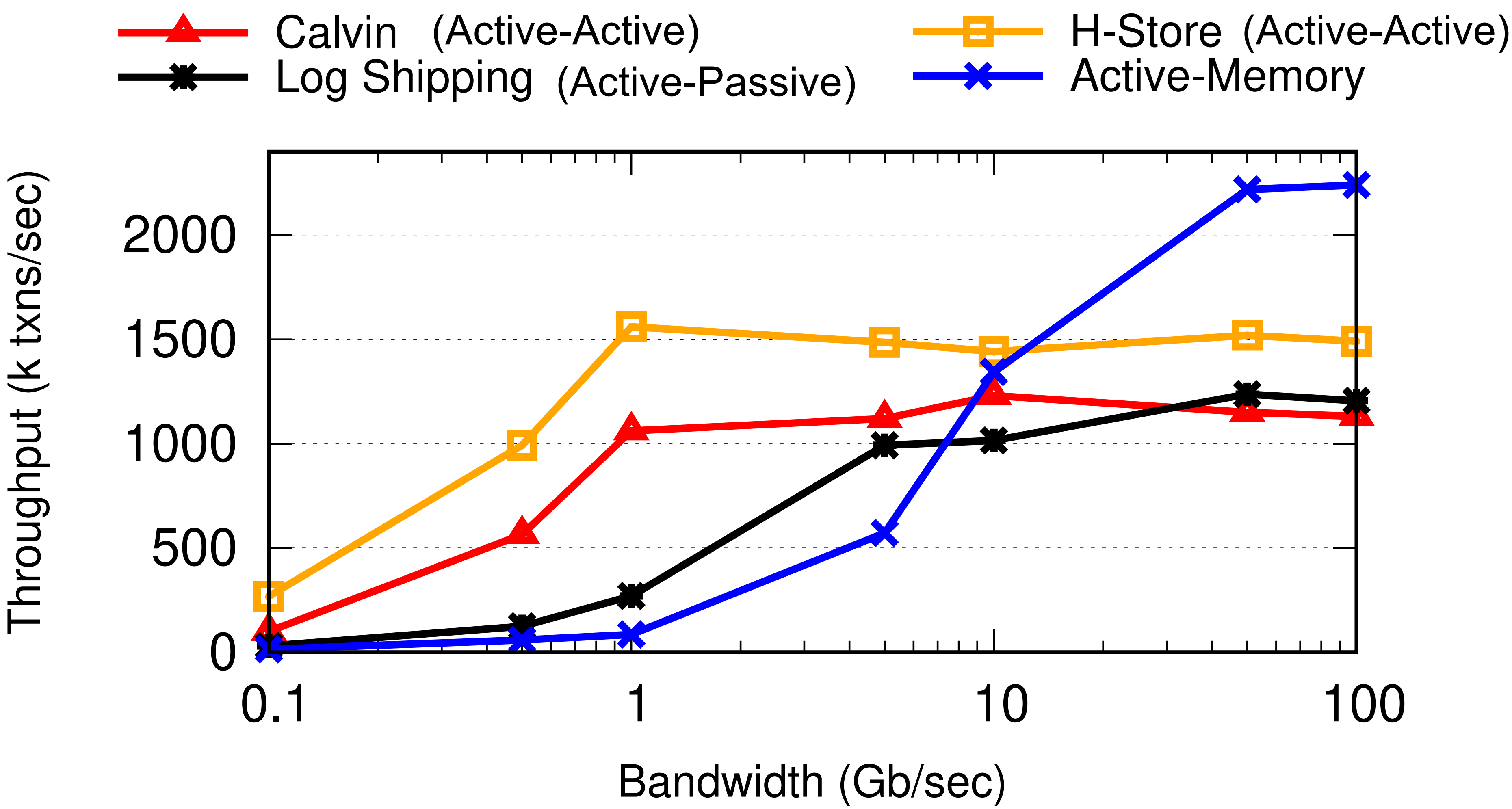
All or nothing replication



Solution: **UNDO logging**

Write UNDO log to a backup **before** writing the actual data

Evaluation: Network Bandwidth

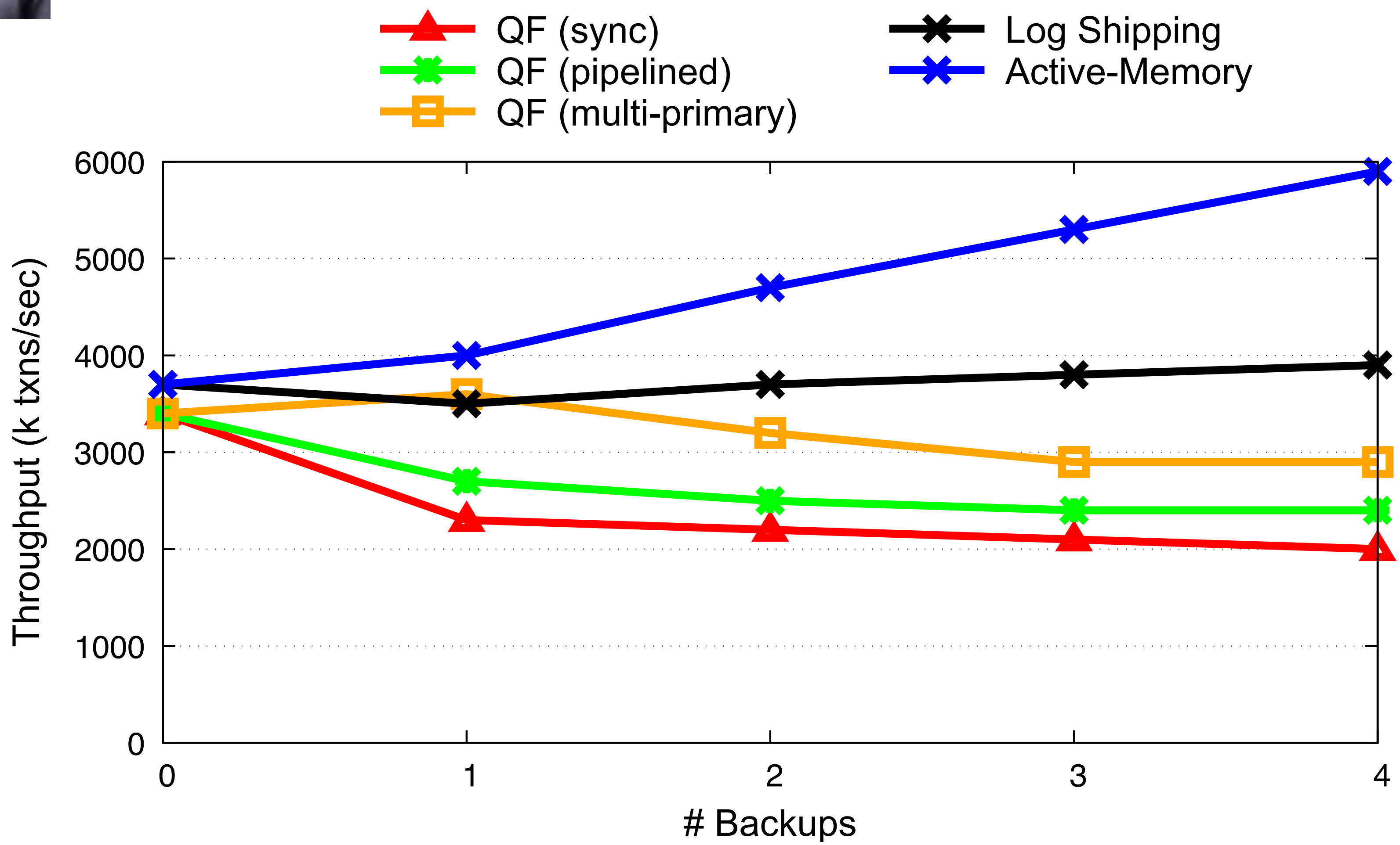


2-phase locking not SI

Workload: YCSB, 1KB records w/ 10 columns, 5M records / server. Each txn has 10 RMW ops

8 nodes: 2 Xeon E7-4820 processors (each with 2x10 cores), 256 GB RAM, 3 way-replication

Just for Ippokratis: Comparison Against Query Fresh



T. Wang, R. Johnson, and I. Pandis. **Query fresh: Log shipping on steroids**. PVLDB, 11(4):406–419, 2017

4 Key Results from 4 Years

skip

NAM-Architecture

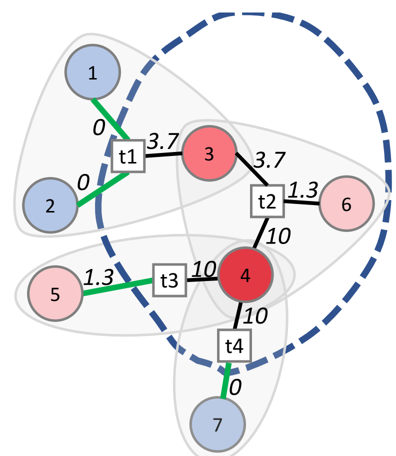
Carsten Binnig, Andrew Crotty, Alex Galakatos, Tim Kraska, Erfan Zamanian: **The End of Slow Networks: It's Time for a Redesign.** PVLDB 9(7): 528-539 (2016)

Scalable SI Protocol

Erfan Zamanian et al: **The End of a Myth: Distributed Transaction Can Scale.** PVLDB 10(6): 685-696 (2017)

Active Replication

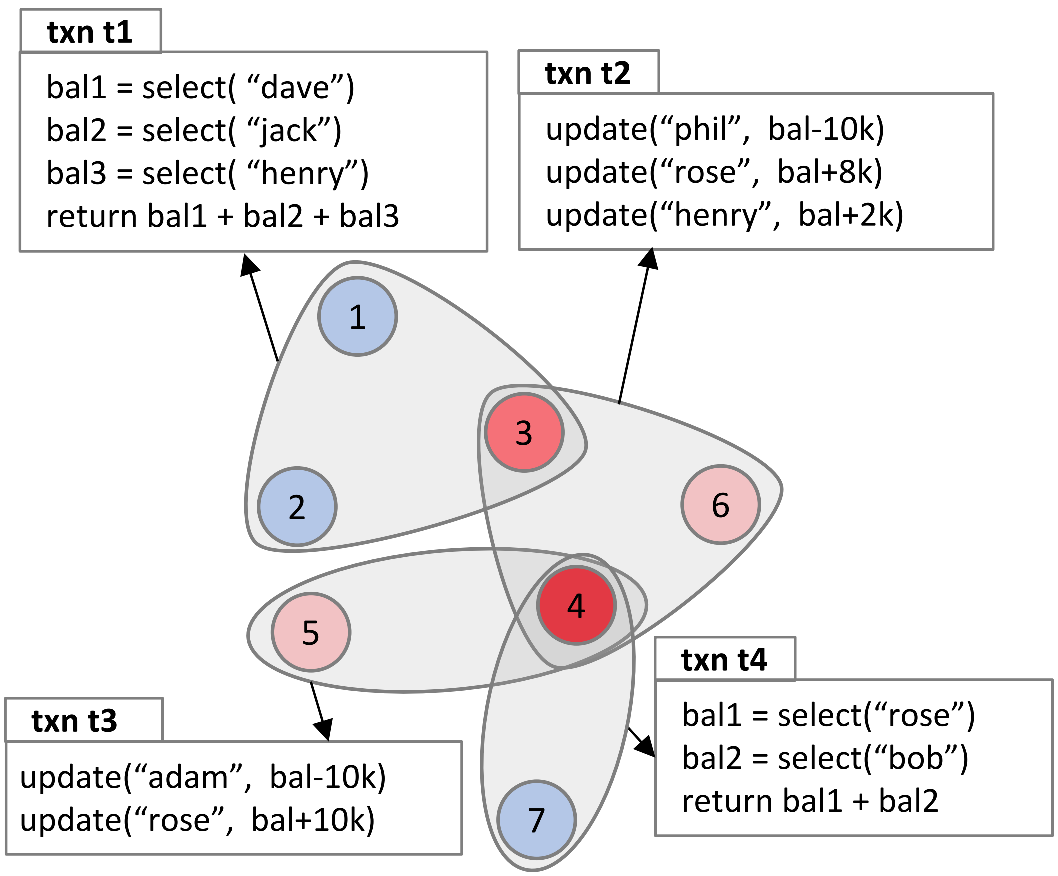
Erfan Zamanian et al: **Rethinking Database High Availability with RDMA Networks.** PVLDB 12(11): 1637-1650 (2019)

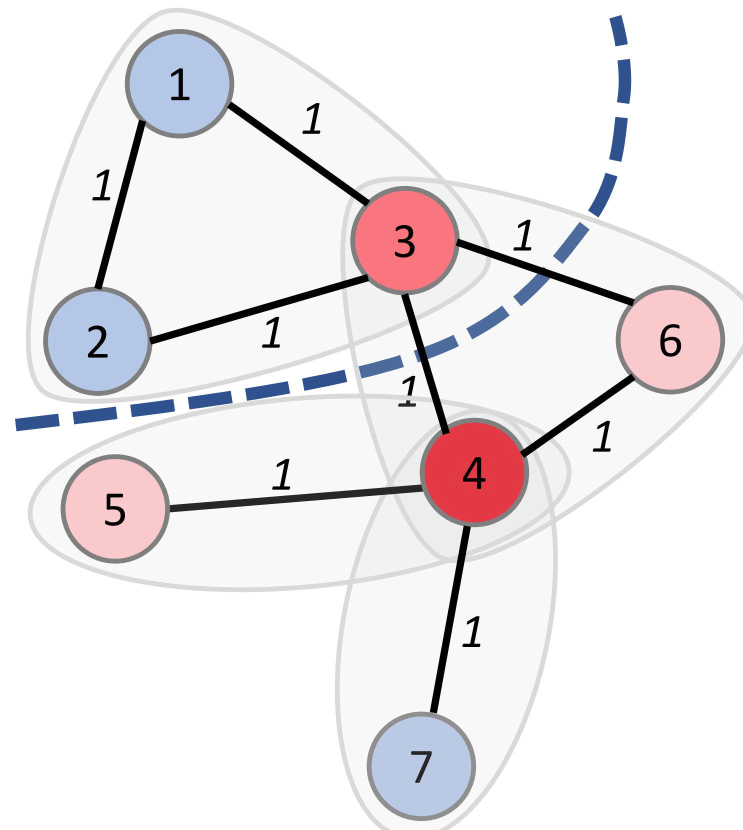


Contention-centric Clustering

Erfan Zamanian et al: Chiller: Contention-centric Transaction Execution and Data Partitioning for Fast Networks. CoRR abs/1811.12204 (2018)

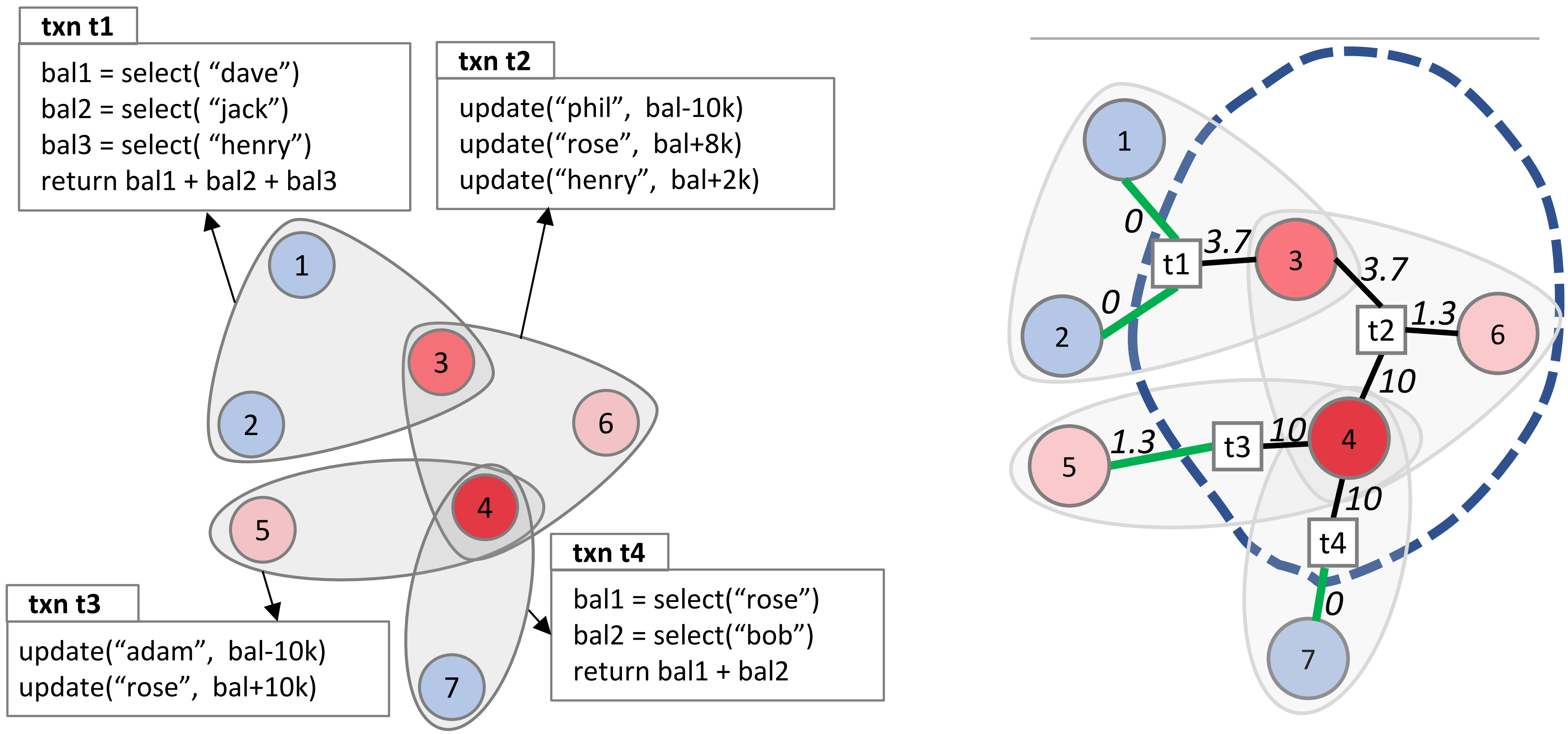
Traditional Partitioning





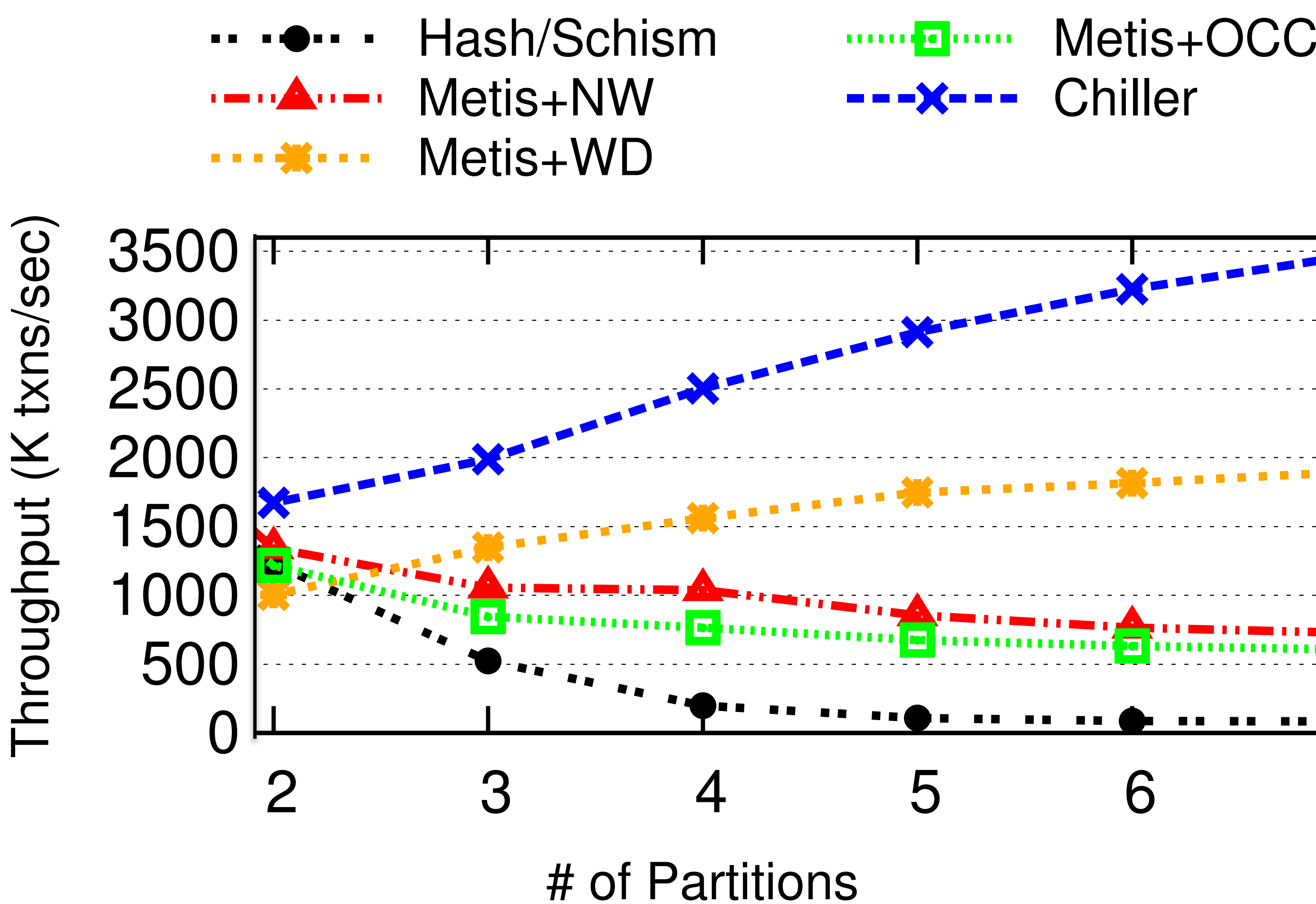
Erfan Zamanian et al: Chiller: Contention-centric Transaction Execution and Data Partitioning for Fast Networks. CoRR abs/1811.12204 (2018)

Chiller Partitioning



Erfan Zamanian et al: Chiller: Contention-centric Transaction Execution and Data Partitioning for Fast Networks. CoRR abs/1811.12204 (2018)

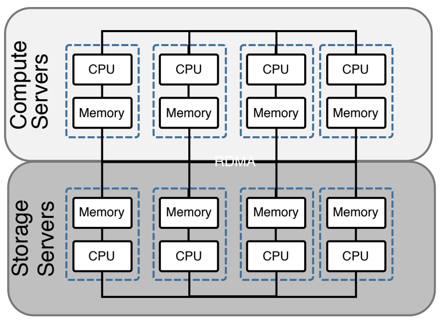
Results



Erfan Zamanian et al: Chiller: Contention-centric Transaction Execution and Data Partitioning for Fast Networks. CoRR abs/1811.12204 (2018)

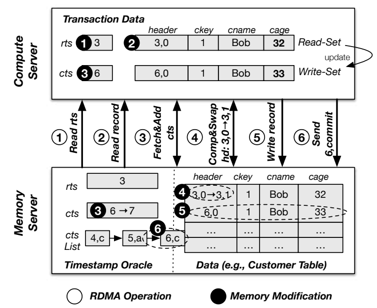
Conclusions

"There is no shared nothing, if you have 400Gbs NIC"



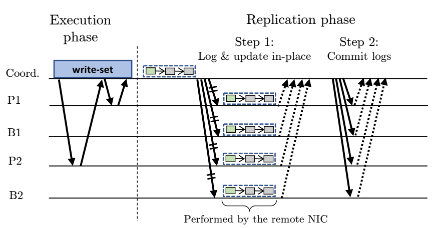
NAM-Architecture

Carsten Binnig, Andrew Crotty, Alex Galakatos, Tim Kraska, Erfan Zamanian: **The End of Slow Networks: It's Time for a Redesign.** PVLDB 9(7): 528-539 (2016)



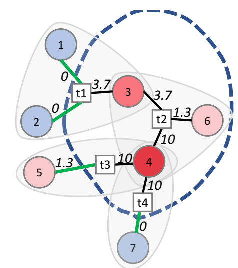
Scalable SI Protocol

Erfan Zamanian et al: **The End of a Myth: Distributed Transaction Can Scale.** PVLDB 10(6): 685-696 (2017)



Active Replication

Erfan Zamanian et al: **Rethinking Database High Availability with RDMA Networks**. PVLDB 12(11): 1637-1650 (2019)

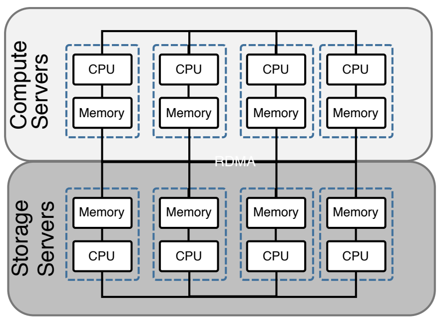


Contention-centric Clustering

Erfan Zamanian et al: Chiller: Contention-centric Transaction Execution and Data Partitioning for Fast Networks. CoRR abs/1811.12204 (2018)

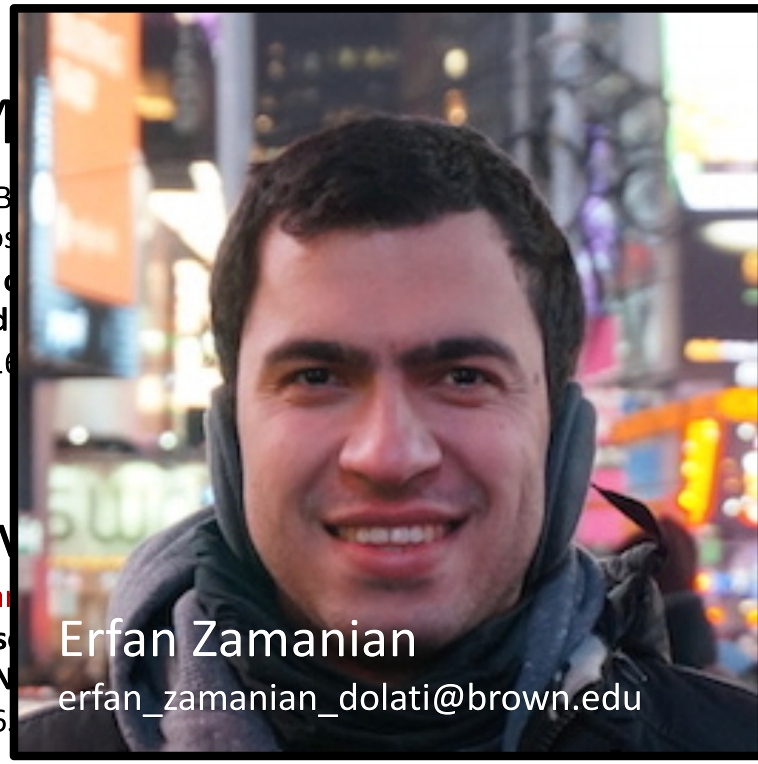
Conclusions

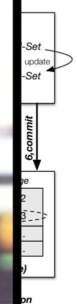
"There is no shared nothing, if you have 400Gbs NIC"



NAM

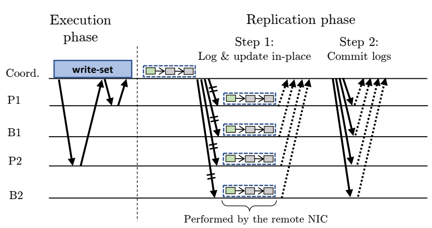
Carsten B
Galakatos
The End o
for a Red
539 (201





Scalable SI Protocol

Erfan Zamanian et al: **The End of a Myth: Distributed Transaction Can Scale.** PVLDB 10(6): 685-696 (2017)



Activ

Erfan Za
Databas
RDMA N
1637-16

Contention-centric Clustering

Erfan Zamanian et al: Chiller: Contention-centric Transaction Execution and Data Partitioning for Fast Networks. CoRR abs/1811.12204 (2018)