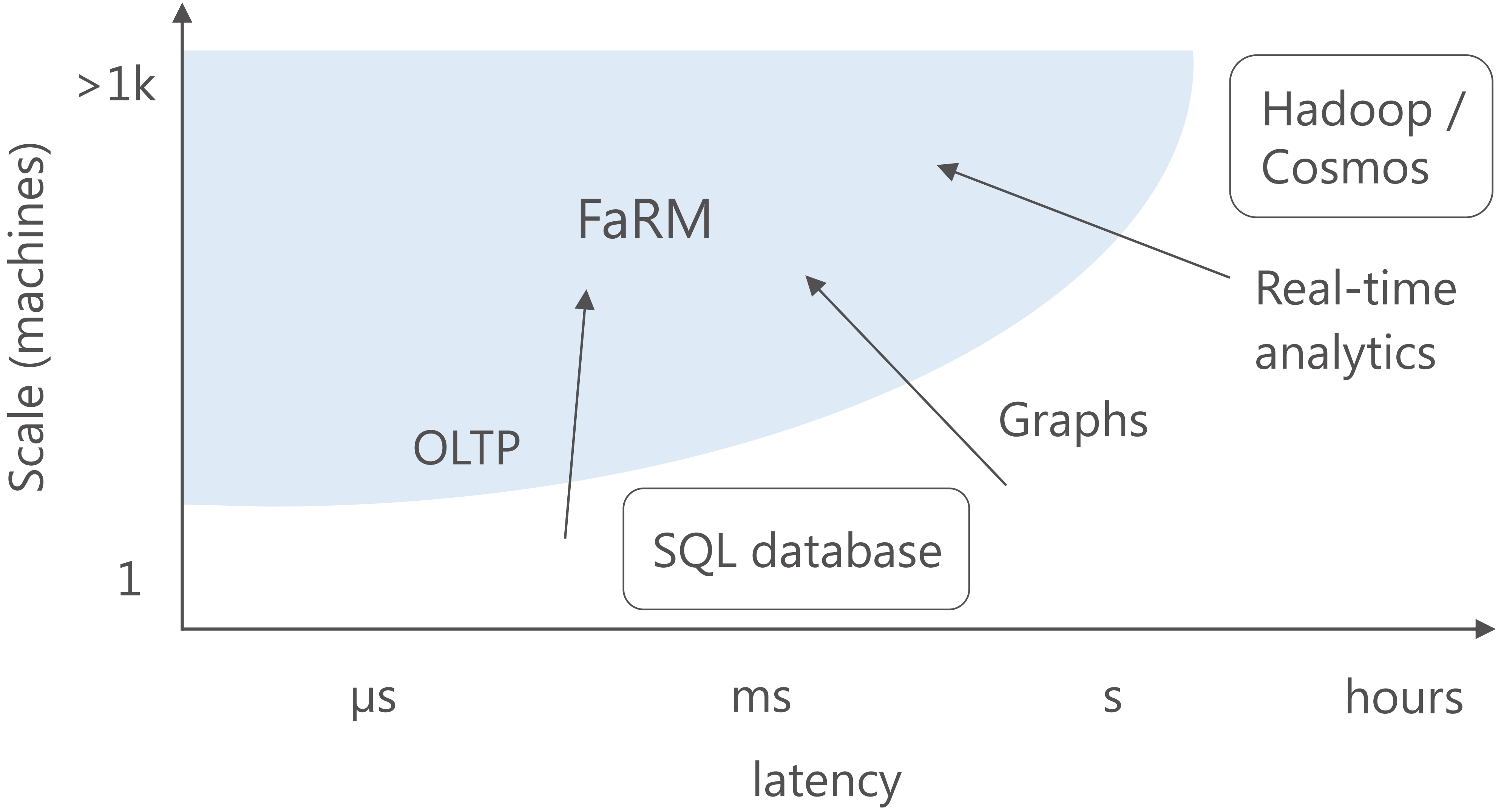
FaRM: A Platform for Low-latency Computing

Aleksandar Dragojević, Dushyanth Narayanan, Greg O'Shea, Miguel Castro, Chiranjeeb Buragohain, Richie Khanna, Matt Renzelmann, Knut Magne Risvik, Sudipta Sengupta, Alex Shamis, Bin Shi, Tim Tan, John Zheng

Microsoft Research

Distributed computing



Why low latency matters

More work within latency budget (<100 ms)

10-100 dependent accesses if latency is in ms range

1k-10k if it is in μs range

Freshness

Denormalized data for low latency

Services process data offline and bulk load into online component

Low latency allows to keep only one representation

Easier development

Less effort on tuning, more on user experience

Should not be underestimated

Enabled by hardware trends

Large amounts of DRAM

256 GB DRAM per commodity machine

New memory technology with higher density soon

Non-volatile memory

Use battery to flush DRAM to SSD on a power failure

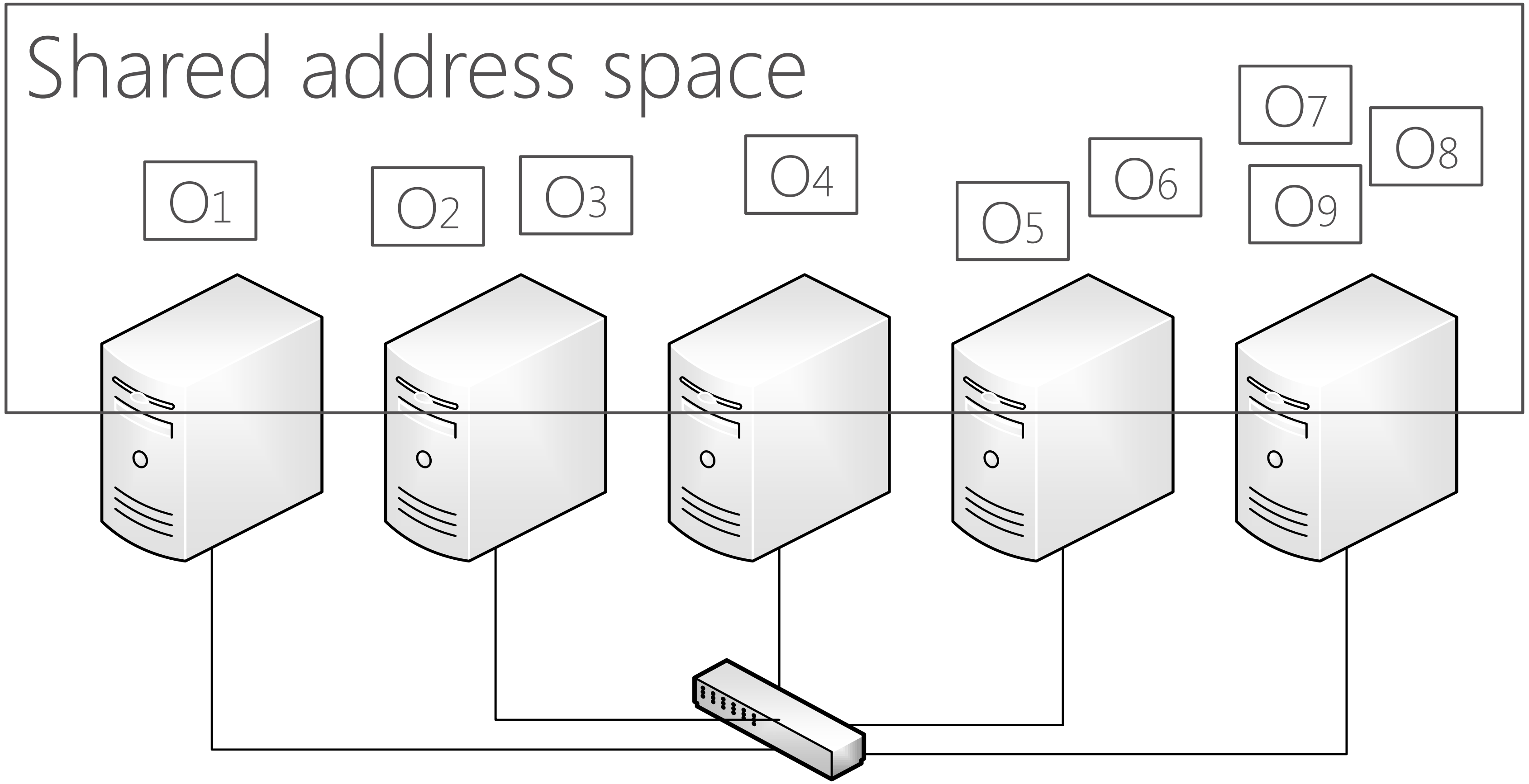
Non-volatile memory technology

Fast networks with RDMA

100 Gbps of bandwidth,100 M ops/s, 1-3 μs latency

RDMA reads and writes

FaRM



General platform

Key-value, graph, relational

Transactions

Read, write, alloc, free

Replicated in memory

Performance

High throughput

Low latency

Outline

Design

Performance

Future work

CPU is the bottleneck

Design the system from first principles to use the hardware effectively

Use one-sided RDMA operations	Reduce message counts	Effectively use parallelism

RDMA in FaRM

Read objects with RDMA

NIC performs DMA (CPU not involved)

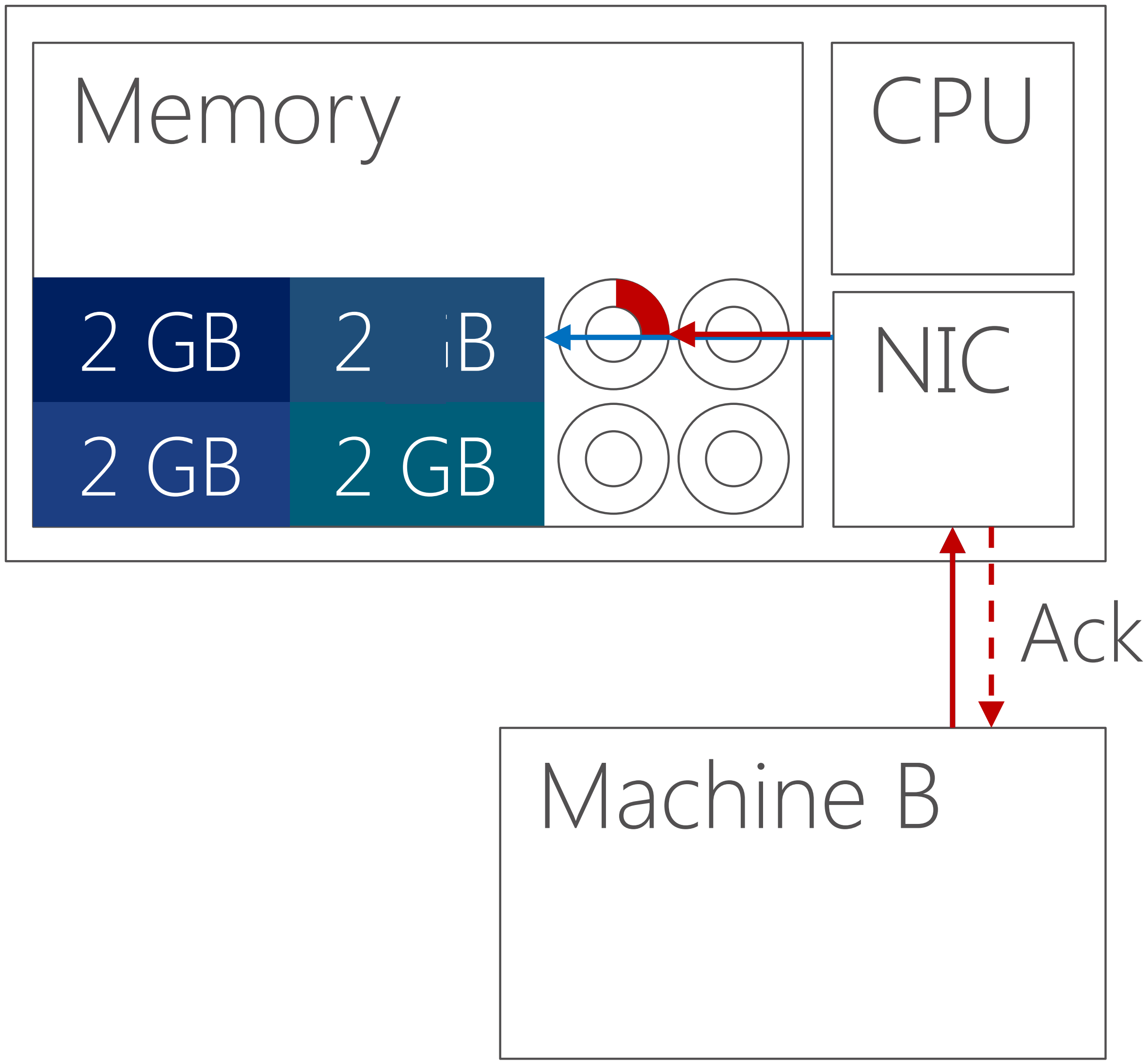
FaRM ensures reads are consistent

Write messages to buffers

Receiver's CPU polls

Hardware acks the write

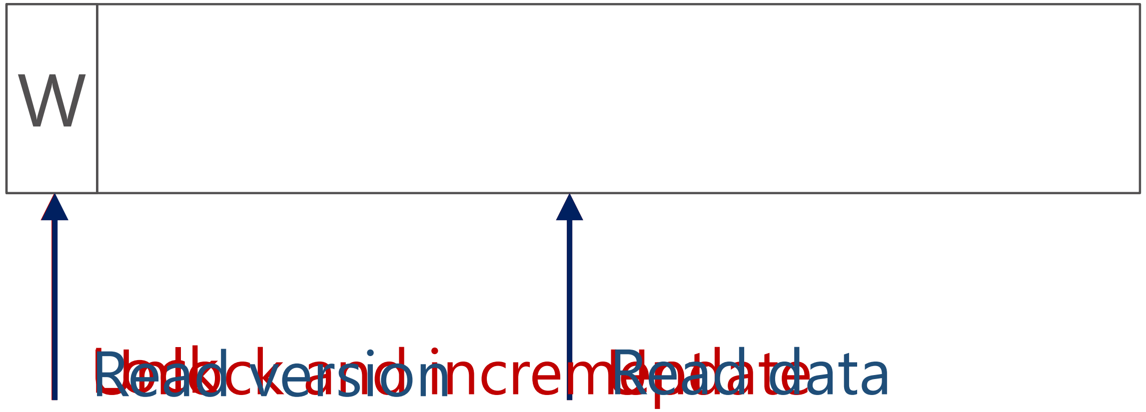
Also used as persistent logs



Lockless reads

Header version

64-bit to avoid overflow



Consistent if versions match and object is not locked

Read requires three network accesses

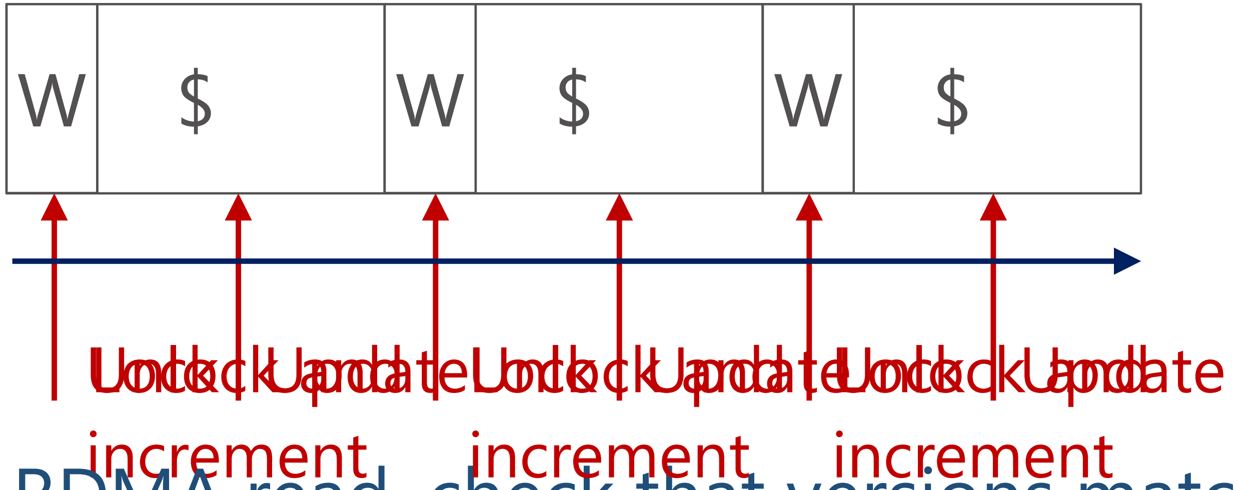
Read

Update

Lockless reads

Header version

Cache line versions



Space efficiency: 16-bit cache-line versions

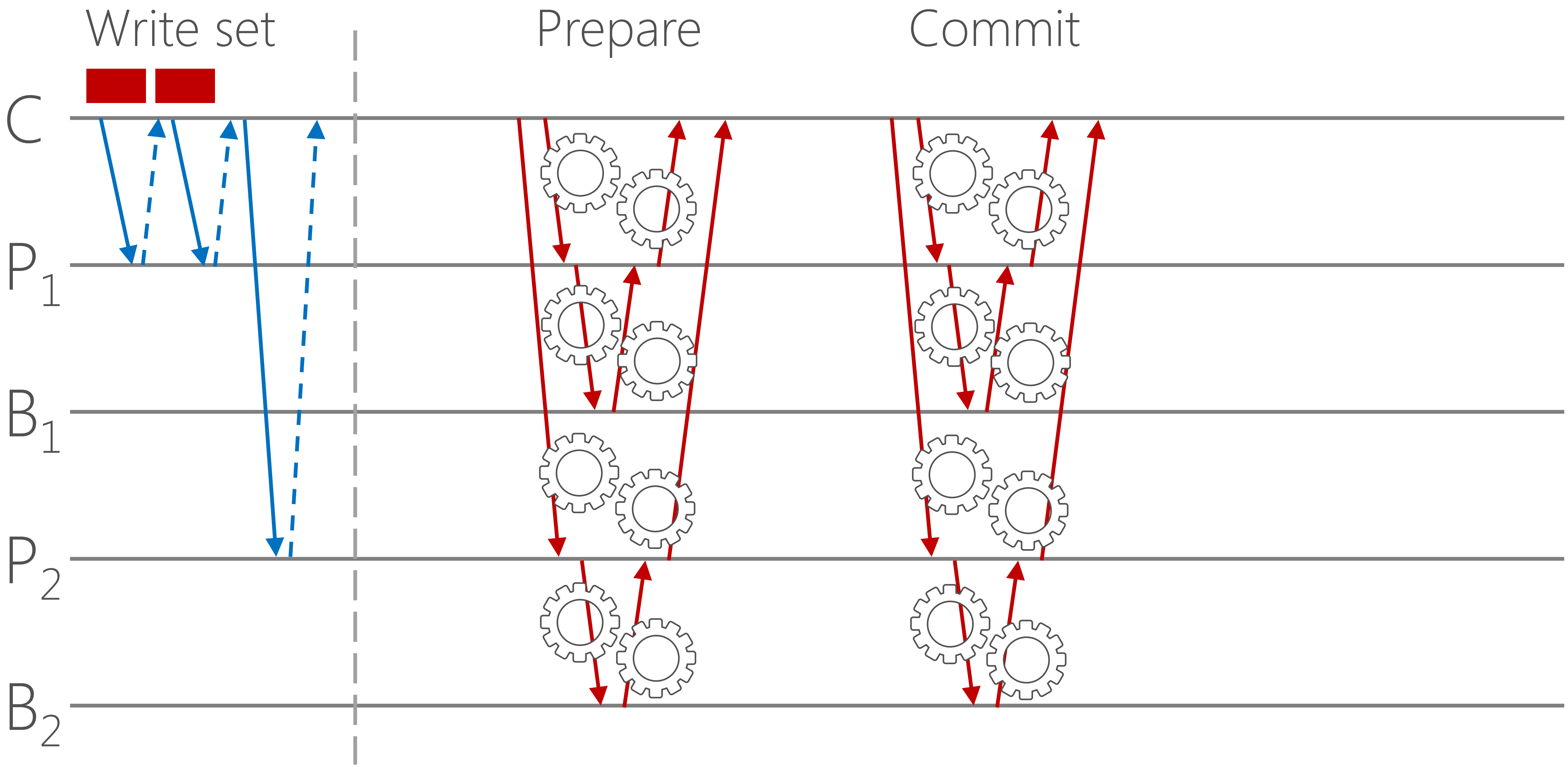
RDMA read, check that versions match and that read does not take too long

$t_{update\_min} = 40$ ns

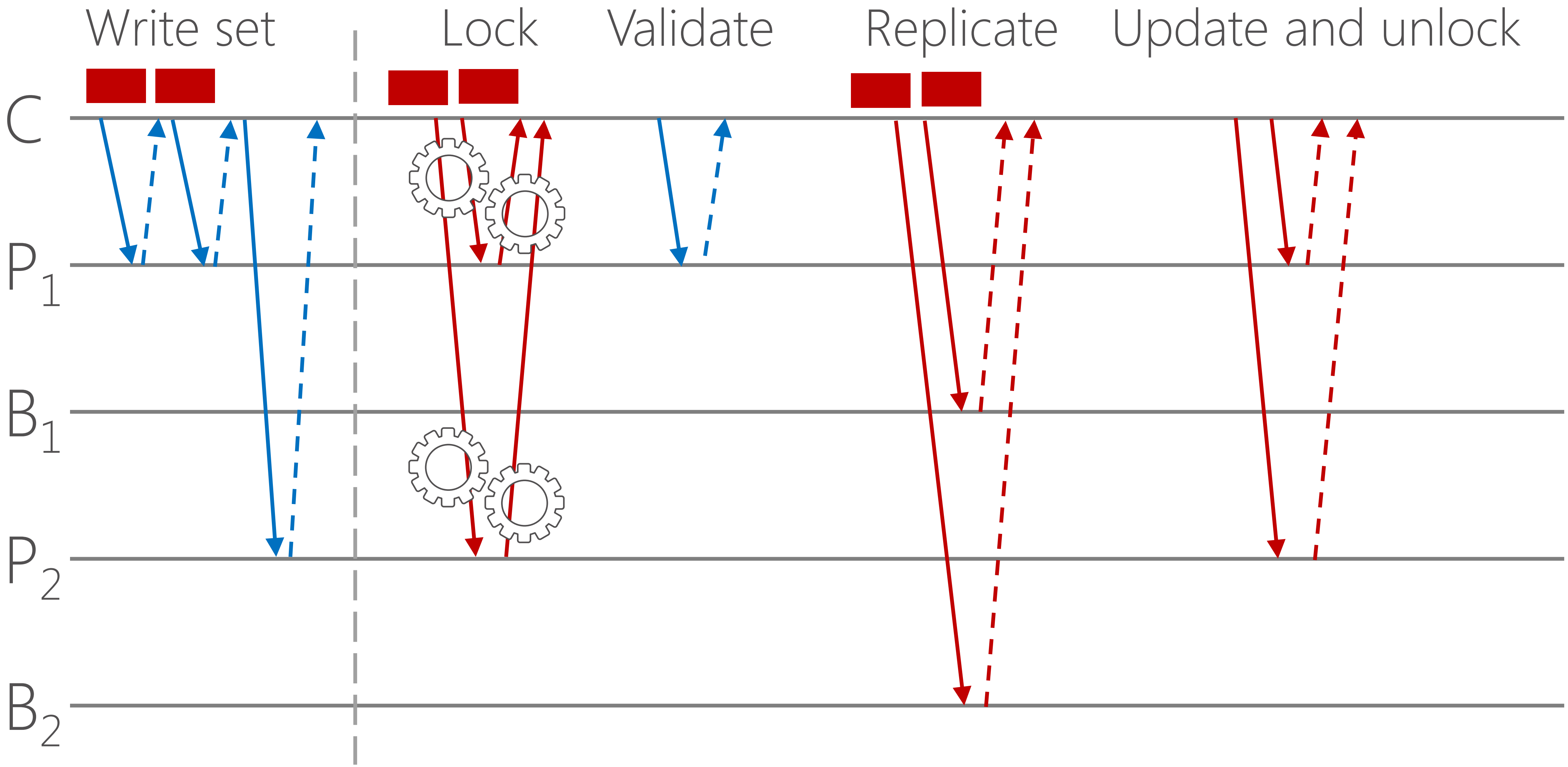
$t_{read\_max} = 40 \text{ ns} * 2^{16} * (1 - \varepsilon) = 2 \text{ ms}$

Transaction execution

Two phase commit

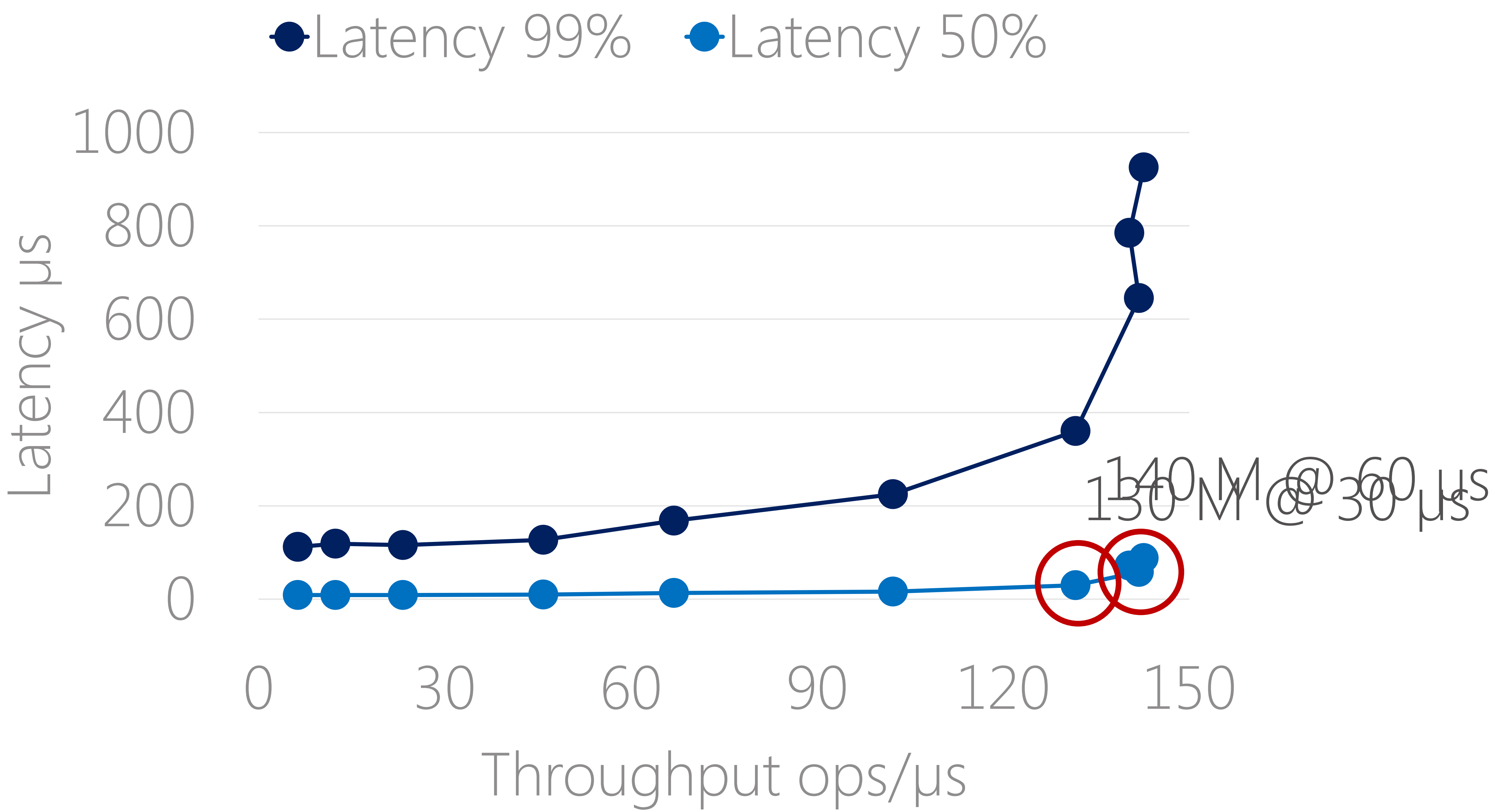


FaRM commit

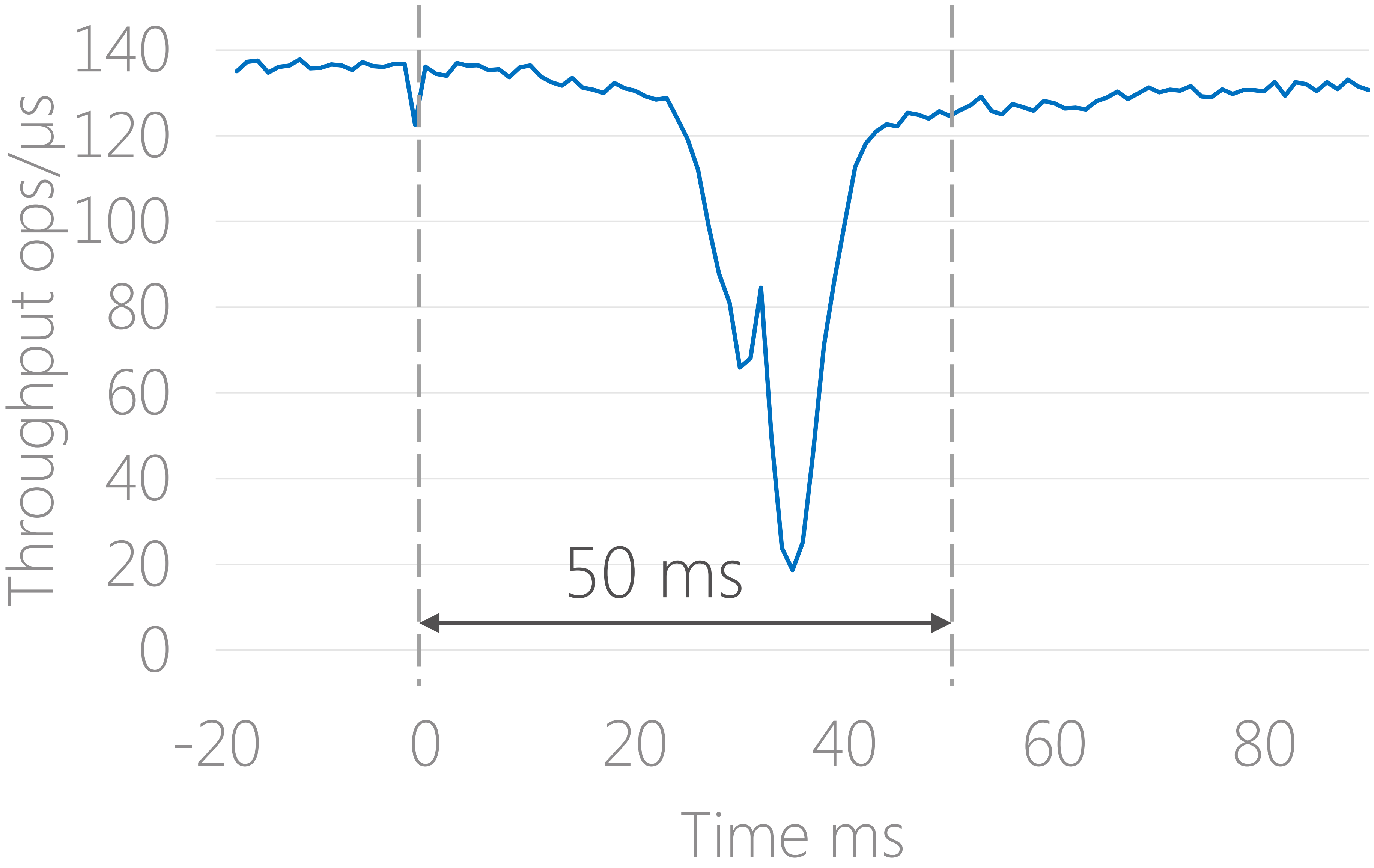

Write set
Lock
Validate
Replicate
Update and unlock
C
P1
B1
P2
B2

Performance

TATP performance



TATP recovery



Future work

Data stores

Graphs, scale-out OLTP, support analytics on fresh data

Hardware acceleration

Custom hardware primitives for low latency and high performance

Cold data

Keep cold data on storage without losing performance for hot data

Disaster recovery

Geo-replication without sacrificing too much latency

Security

RDMA does not have strong security

Extra slides

Settings

90 machines

2x Infiniband Mellanox ConnectX-3 56 Gbps

32 hardware threads

256 GB DRAM

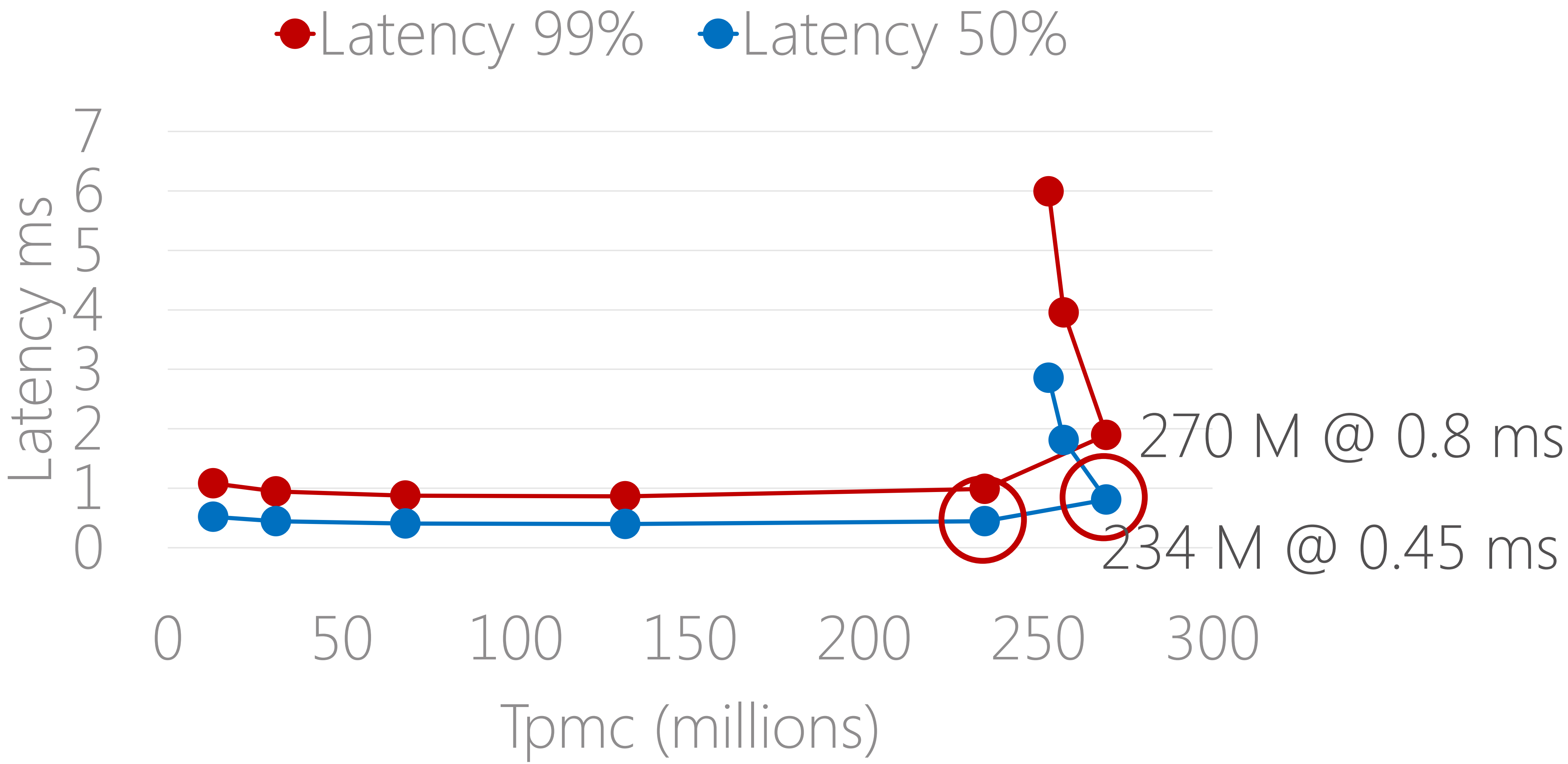
Standard OLTP benchmarks

TATP, TPCC

Performance, speed of recovery

3-way replication

TPCC performance



18 machines failing