Storm: a fast transactional dataplane for remote data structures

Stanko Novakovic   Yizhou Shan   Aasheesh Kolli   Michael Cui
Yiying Zhang   Haggai Eran   Boris Pismenny   Liran Liss   Michael Wei
Dan Tsafrir   Marcos Aguilera

12th ACM International Systems and Storage Conference (SYSTOR)

What is Remote Direct Memory Access (RDMA)?

- Initiate transfer, hardware executes, async. poll for completions

- Infiniband (IB): specialized network stack for RDMA
  - Fully implemented in hardware (PCIe-based adapters) →
  - Also: IB transport on top of IP and lossless Ethernet

- Key benefits:
  1. one-sided access
  2. user-level w/ minimal instr. footprint

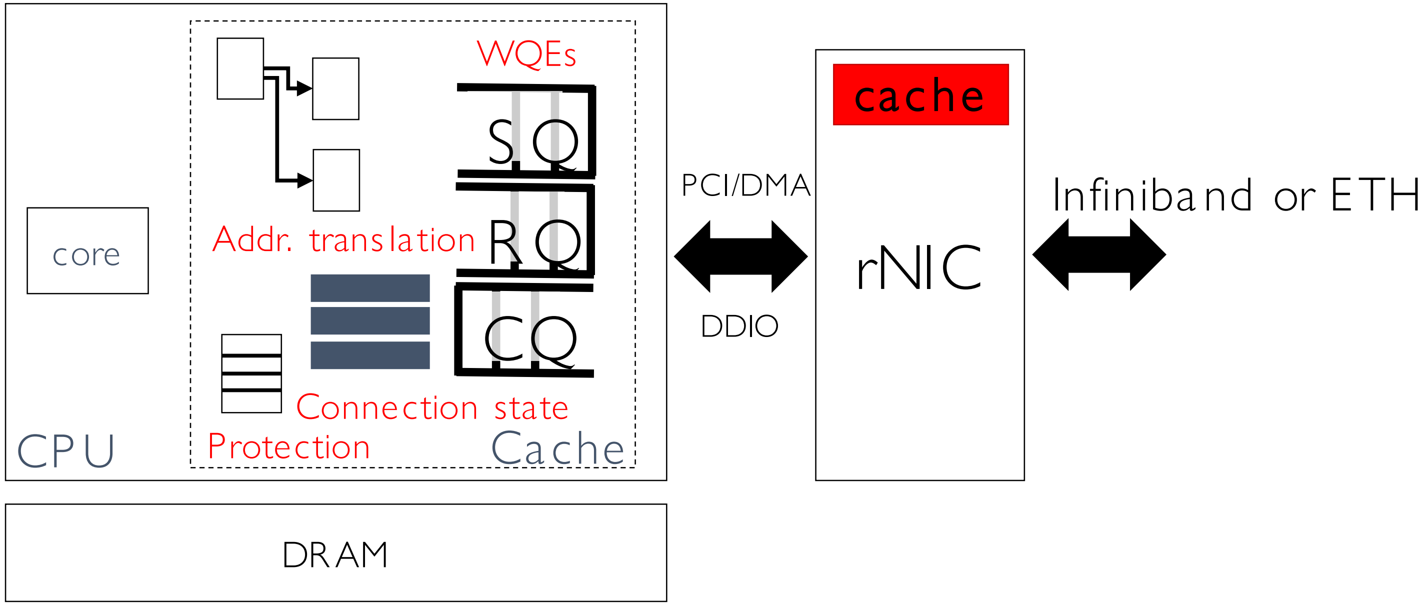
Remote data structures

- Hash tables, graphs, trees, queues, etc
  - Fine-grain accesses
  - High fan-out
  - Pointer-linked
  - Transactional access
  - Throughput (IOPS) bound
  - Latency Service Level Objective (SLO)

- Other (perhaps less interesting) use cases: analytics, VM migration
  - Bulk transfers, bandwidth-bound

What are common concerns?

1. Scalability: network state kept in limited hardware resources



2. Round-trips: pointer-linked data structures

What are common concerns?

1. Scalability: network state kept in limited hardware resources
    - FARM: Use locks to share QP connections (Dragojevic’14)
    - FaSST/eRPC: Don’t use connections (Kalia’19)
    - LITE: Enforce protection in kernel (Tsai’17)

2. Round-trips: pointer-linked data structures
    - FARM: Use Hopscotch algorithm, one RTT common case
    - FaSST/eRPC: Leverage RPCs rather than one-sided reads

Outline

- Problem statement
- Key insights
- Storm design
- Results

Key insights (1/2)

- Hardware has gotten much better!!!
  - ConnectX-4/5 (CX4/5) vs. ConnectX-3 (CX3)
    - 40M IOPS on CX4 → 4x higher than CX3
    - Scales up to 64 machines → on CX3 IOPS collapses for >10 machines
    - CX4 achieves 10M IOPS when zero cache hits → max IOPS for uncontended CX3
    - Break-even point with datagram send/recv currently at ~4k connections
- Possible further improvements with ConnectX-6
- How is HW getting better?
  - More concurrency, better prefetching, larger caches, etc

Key insights (2/2)

- FARM:
  - Locks degrade throughput unnecessarily
  - Large buckets (due to larger keys) wastes throughput

- FaSST/eRPC:
  - Two-sided doesn’t allow for maximum full-duplex throughput
    - Especially for requests larger than a cache line (no inlining)
  - Onloaded congestion control adds overhead

- LITE:
  - Kernel adds overhead (fine-grain accesses)
  - No support for async. operations

Our approach / Storm design principles

1. Use connections but minimal count
    - Lock-free QP sharing if really necessary
    - Offloaded congestion control and retransmissions

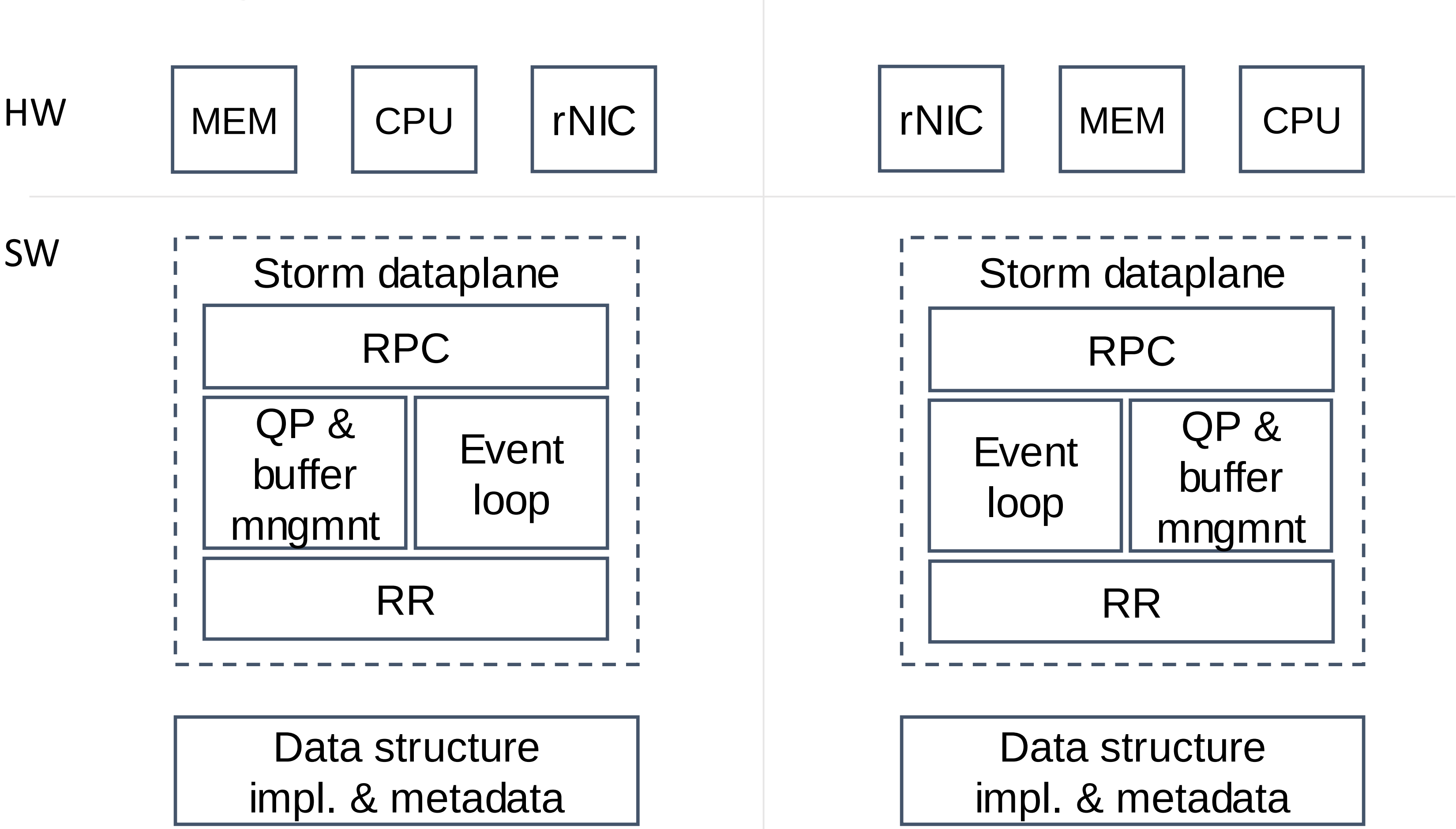
2. Use one-sided reads whenever possible
    - First one-sided, then RPC (*one-two-sided*)
    - RPC also implemented using one-sided writes

3. Leverage abundant memory
    - Cache metadata and/or reduce collisions in hash tables

4. Minimize translation & protection state
    - Use contiguous physical allocation

5. And don't forget to deploy on new hardware!!!

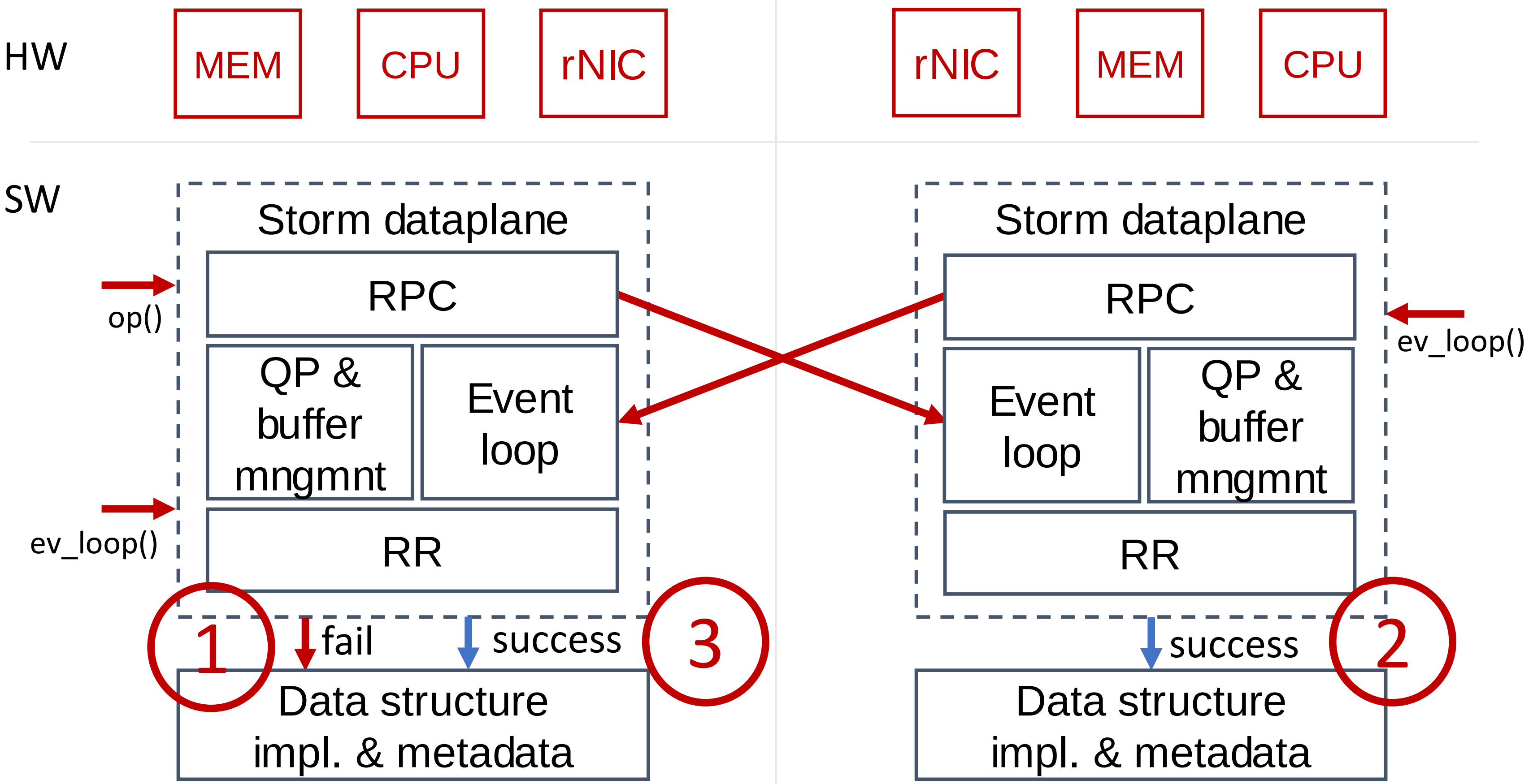
Storm design



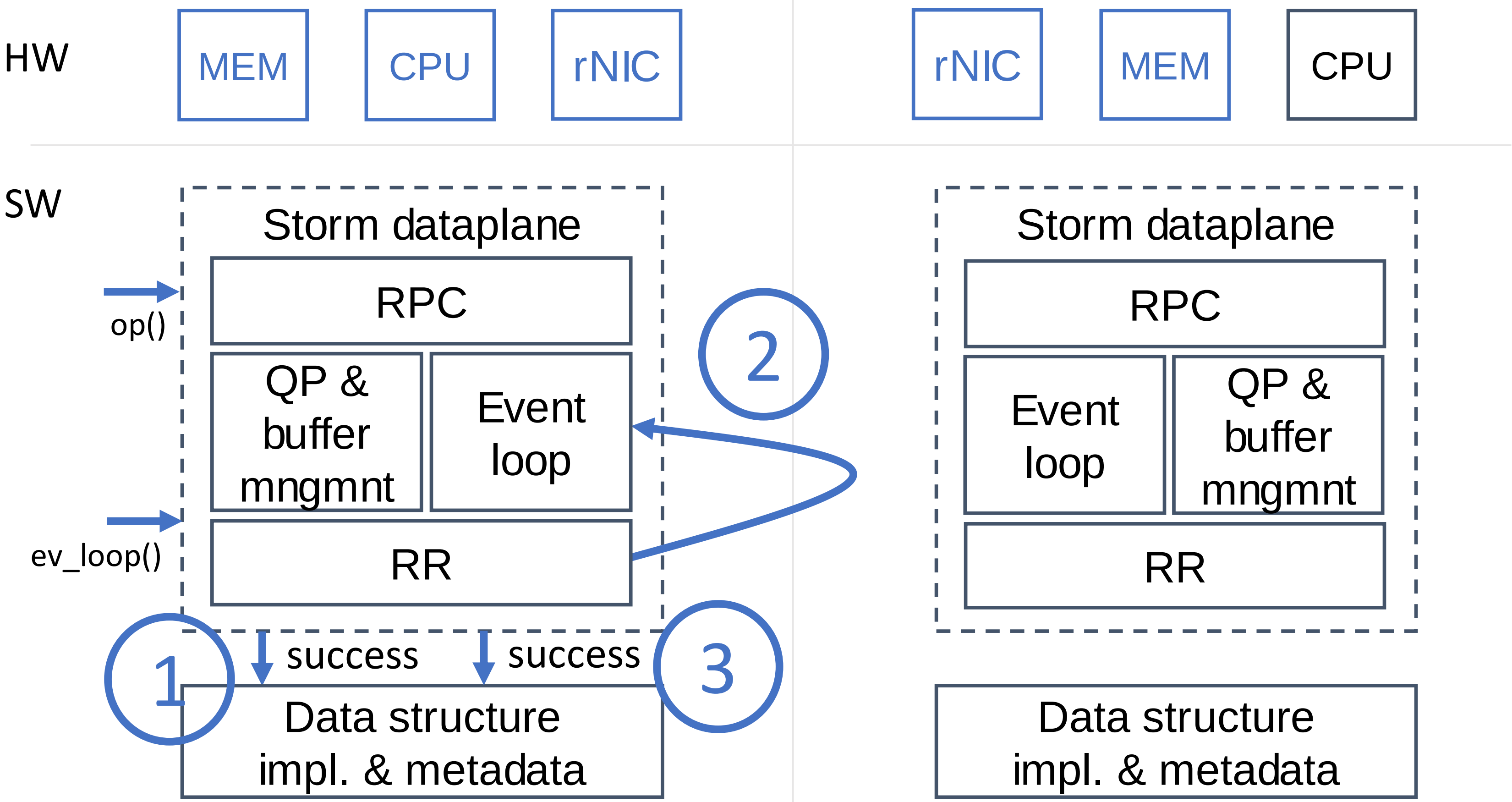
Division of responsibilities:

- **Storm DP only understands RDMA connections and memory regions**
- **Data structure understands data layout and implements metadata caching**

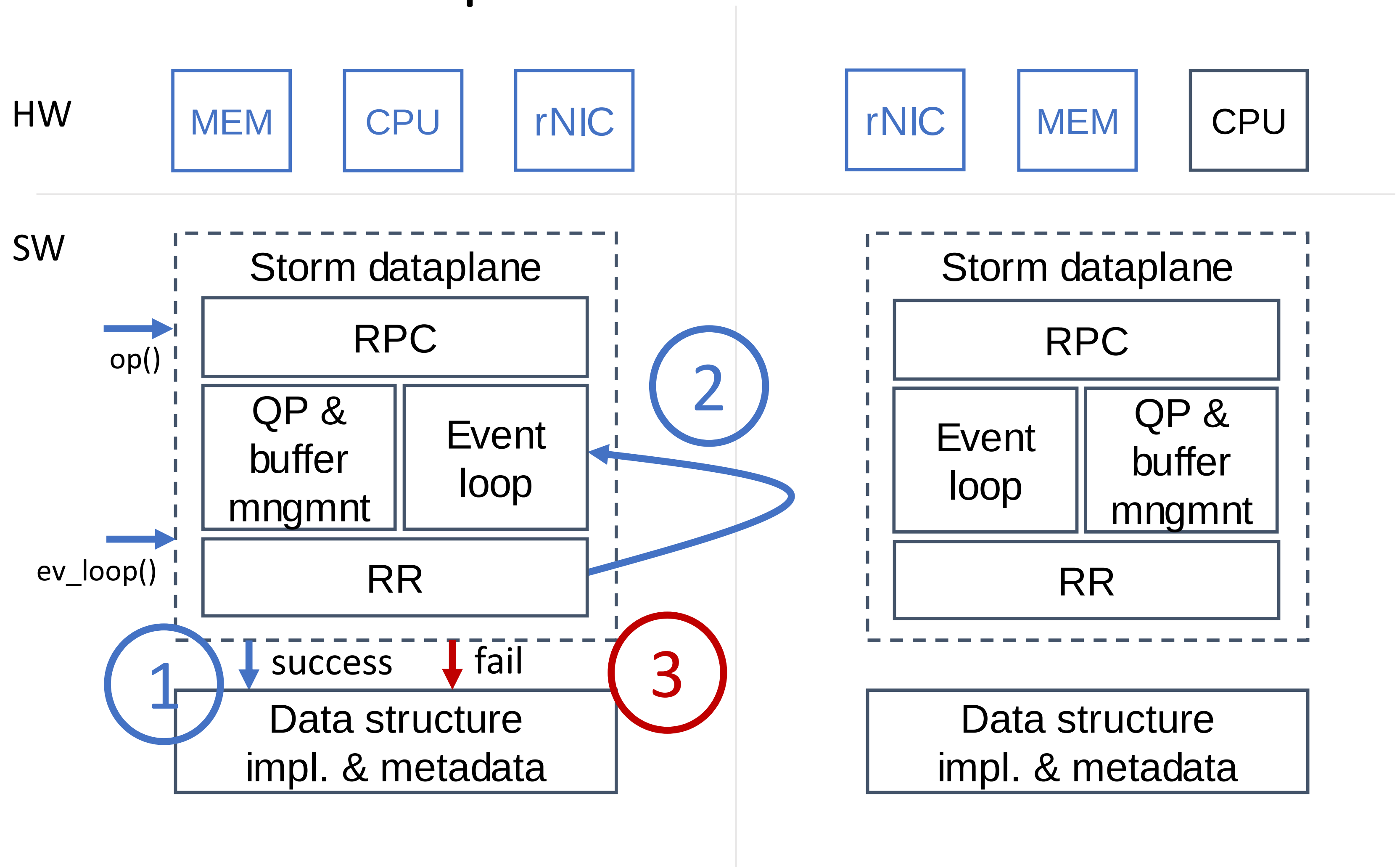
Two-sided operations



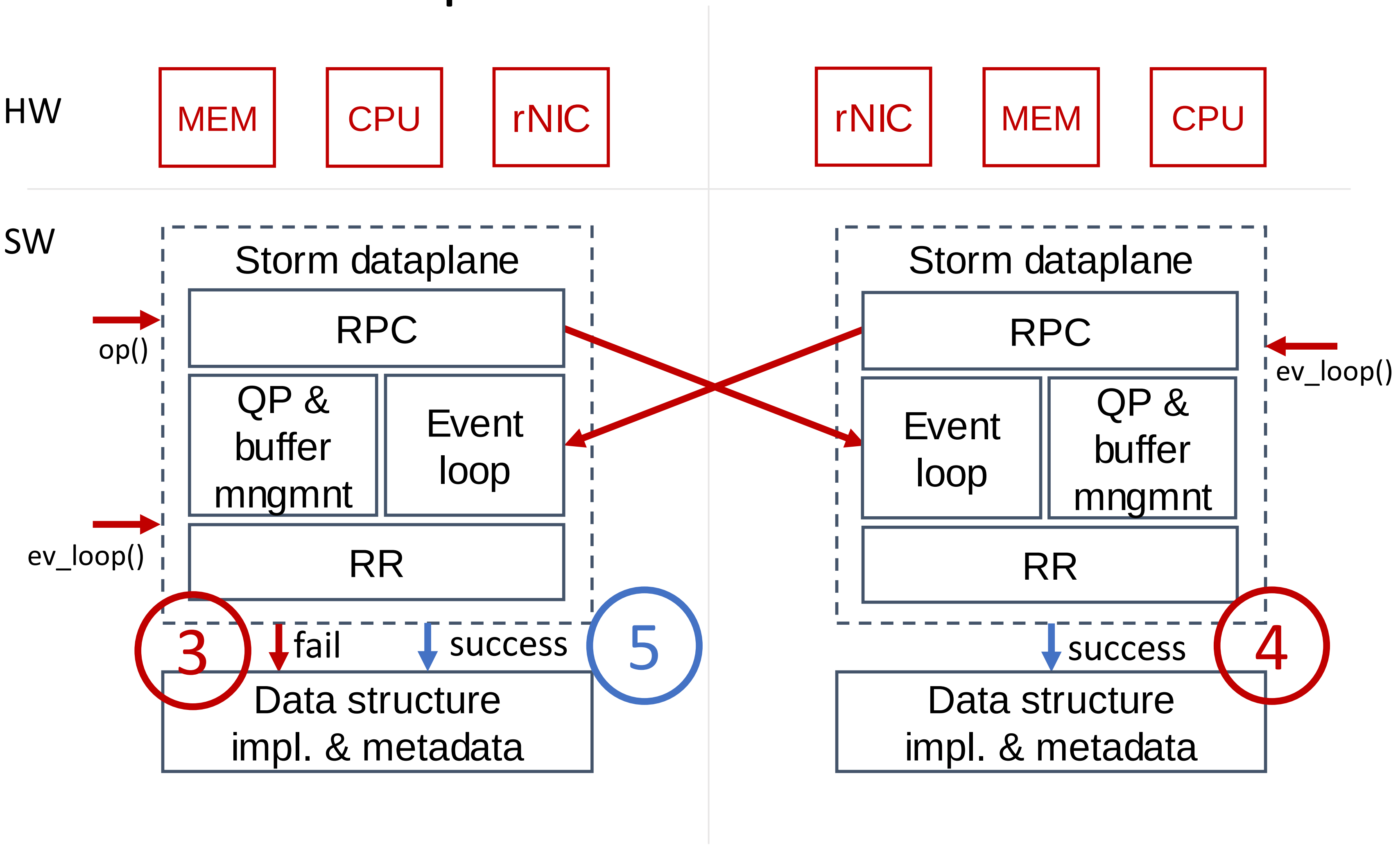
One-sided operations



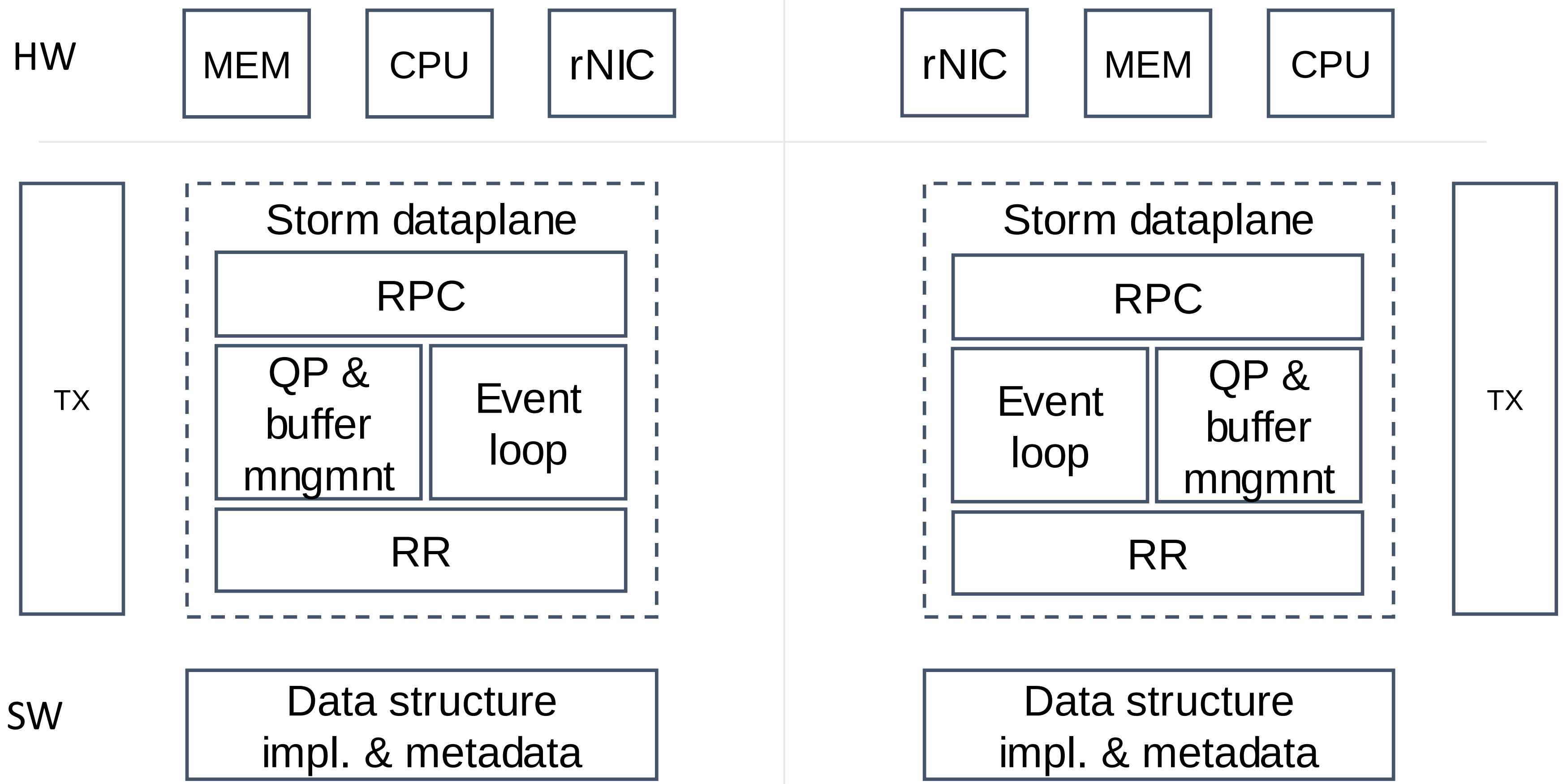
One-two-sided operations



One-two-sided operations



Distributed transactions



Support for concurrent data structures using transactions

Data structure API (three callbacks)

- RPC handler
  - Processing two-sided communication
  - Implements complex paths, such as acquiring locks and commits
- Lookup start
  - Check if address is known (cached) or we can guess
  - If yes, leverage RDMA read
- Lookup end
  - Check if data is valid and cache for future use

Storm implementation & exp. setup

- 13k LOC of C++, w/o MICA modifications [Lim'14]
- HPC cluster w/ 32 Dell machines
  - High-speed Infiniband network (100Gbps)
  - Mellanox ConnectX-4 – similar in perf to CX5
  - Emulation of 3-4x larger clusters possible on Storm
- Benchmarks:
  - Key-value transactional micro-benchmark
  - Telecommunication Application Transaction Processing (TATP)

Outline

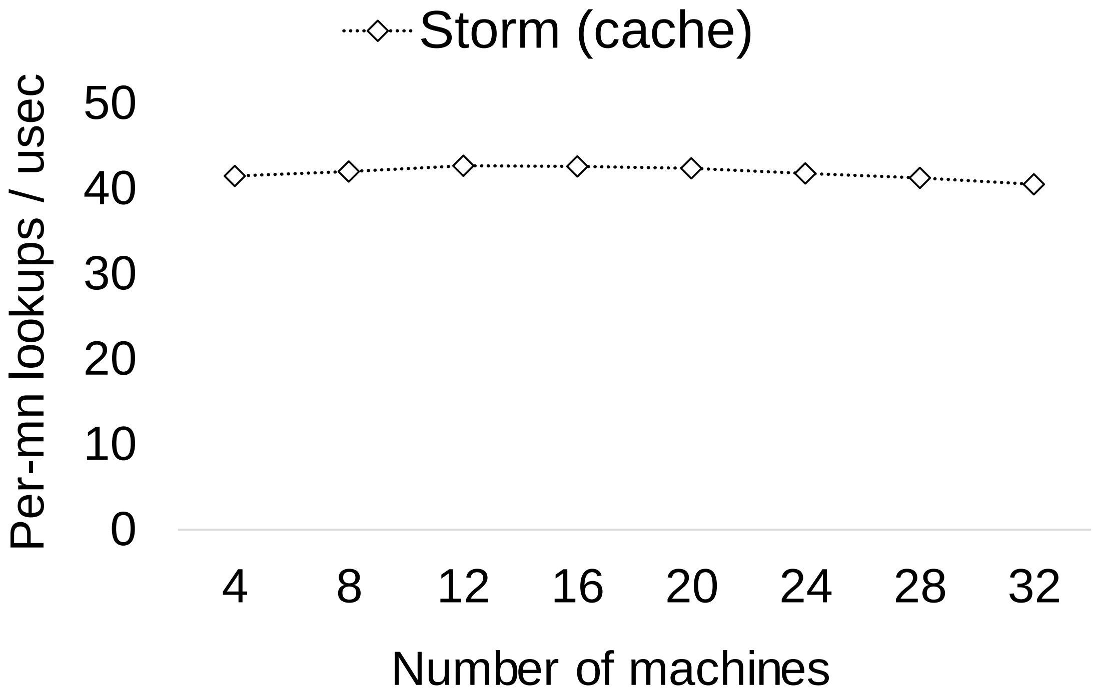
- Problem statement
- Key insights
- Storm design
- Results

Baselines

- Emulated FARM (modified: Lock-free_FaRM)
  - No connection sharing, 1KB “neighborhoods”
- eRPC
  - With and without active congestion control
- LITE (modified: Async_LITE)
  - Added support for asynchronous operations

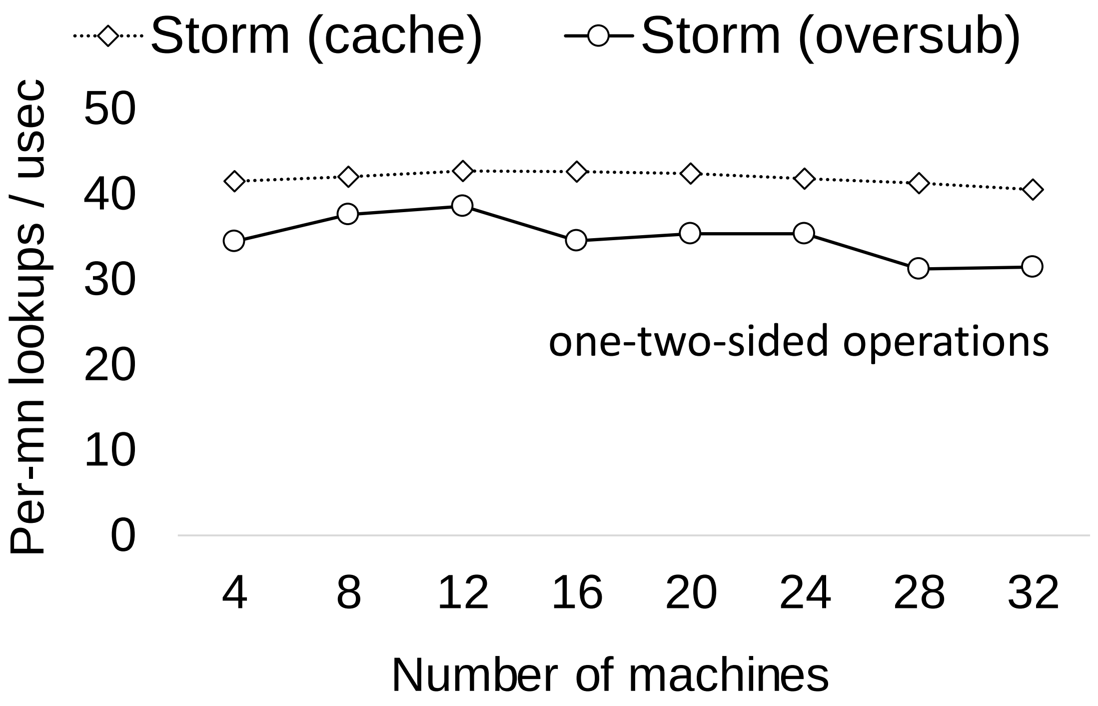
Storm results

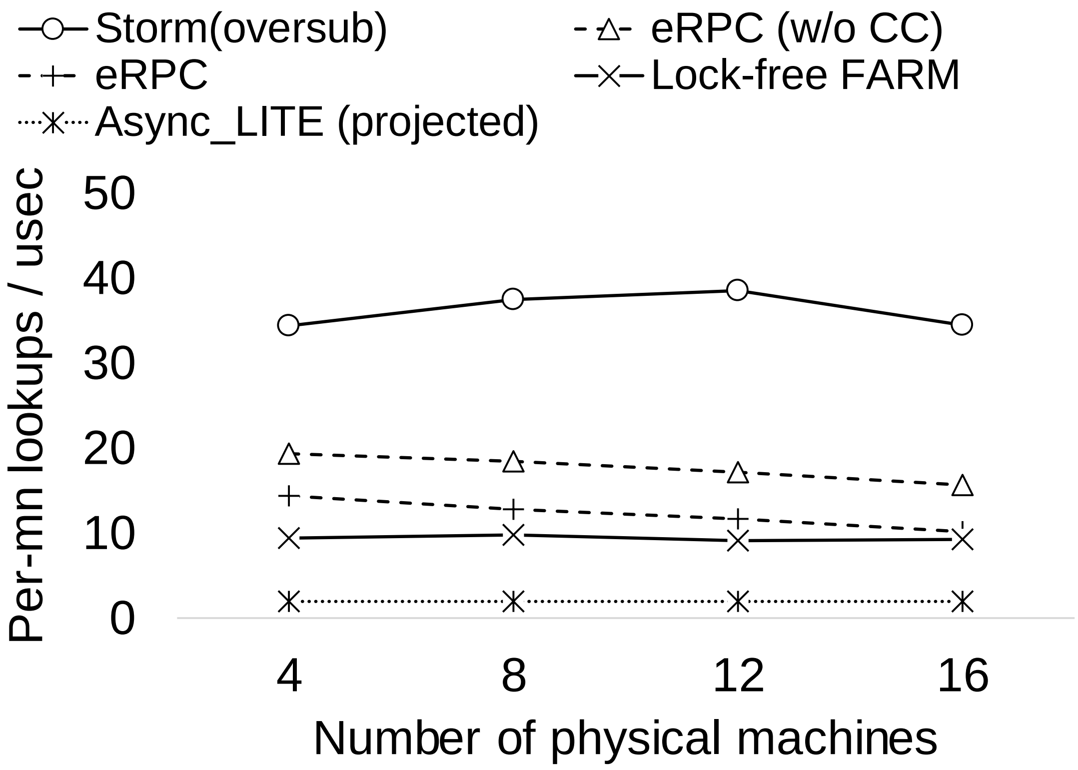
- Single-lookup workload
  - 128B KV pairs, 100M items, 20 threads per mn



Storm results

- Single-lookup workload
  - 128B KV pairs, 100M items, 20 threads per mn





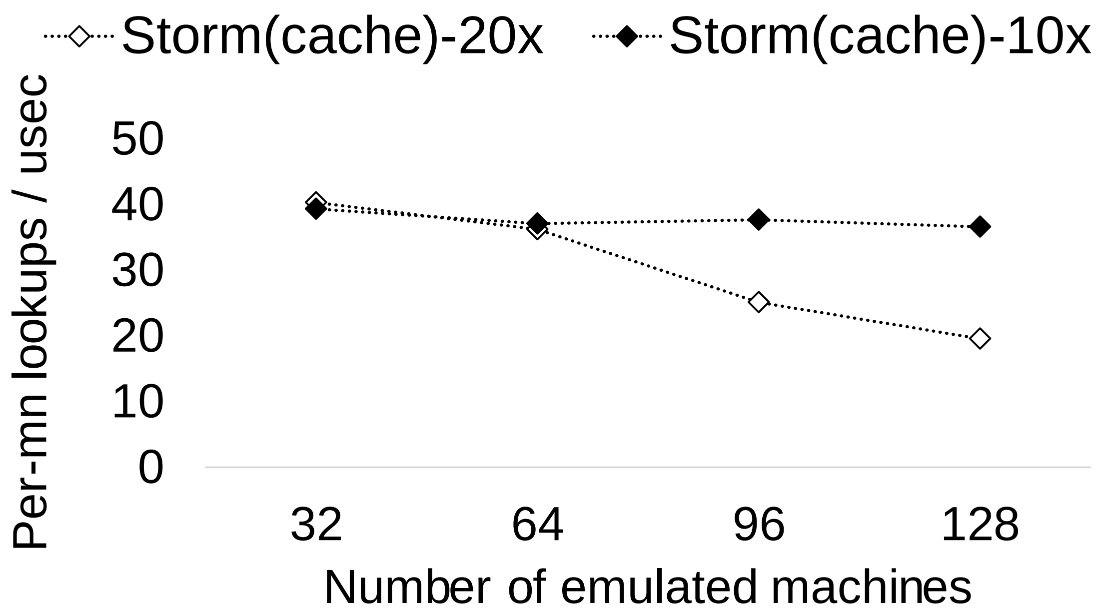
- TATP: 11.8 million per node with Storm (oversub)

Does Storm scale well?

- Storm scales well up to 64mn

- Reduce thread count by 2x
  - 2x fewer threads → 2x fewer QPs

- Do we need more than 10 threads?
  - Lock-free QP sharing



Conclusion & future work

- RDMA datacenter users should get a hardware upgrade
  - More scalable hardware available
  - Take advantage of one-sided primitives

- Leverage caching and oversubscription (in hash tables)
  - One-sided read in the common case

- Ongoing research threads:
  - Designing “far” memory data structures (HotOS’19)
  - Memory allocator for repurposing unused memory
  - Lock-free mechanisms for QP sharing