Silent Data Access Protocol for NVRAM+RDMA Distributed Storage

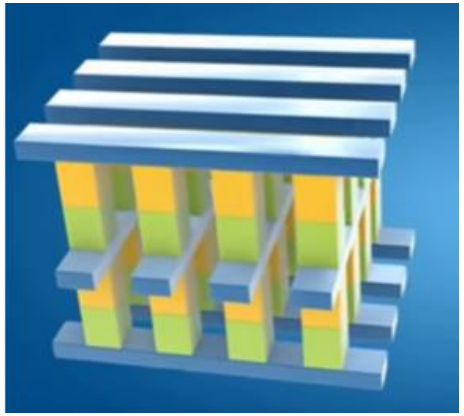
Qingyue Liu
ql9@rice.edu
ECE Department, Rice University

Peter Varman
pjv@rice.edu
ECE Department, Rice University

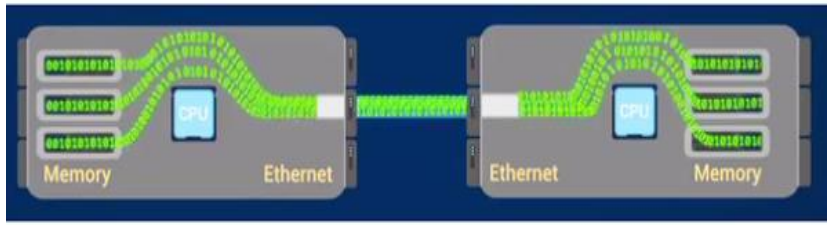
May 7, 2020

Background: NVRAM+RDMA Architecture

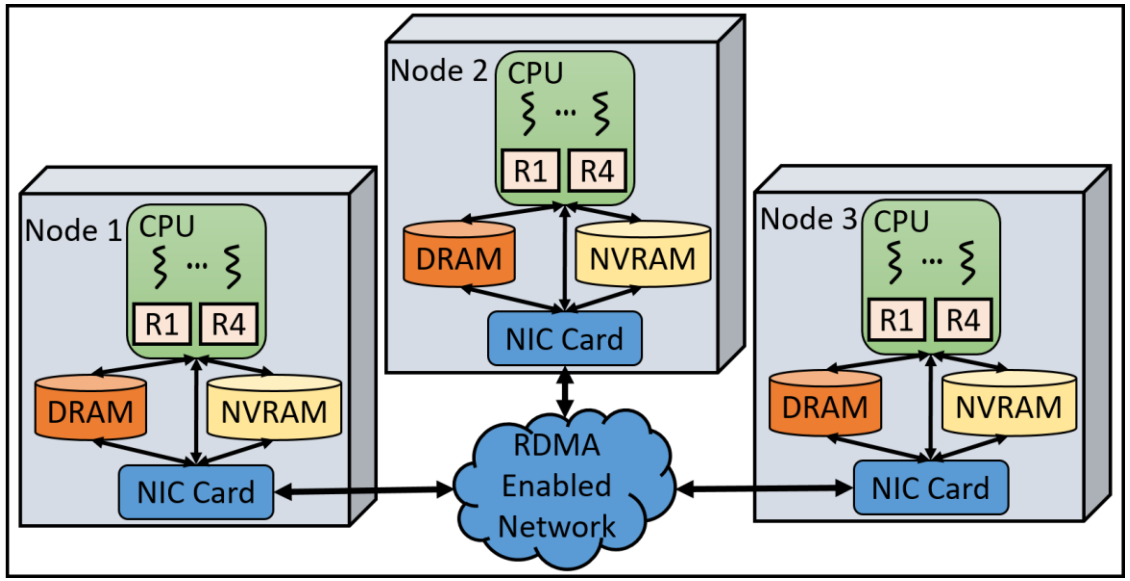
- Future Distributed Storage Systems: NVRAM + RDMA
  - ➢ NVRAM is used directly as persistent database or persistent cache
    - Cache-line access
    - Persistent
  - ➢ Communication between storage nodes using RDMA protocols
    - Bypass TCP/IP stack
    - Microsecond level I/O latency
  - ➢ NVRAM+RDMA
    - Bypass CPU



+



=



Outline

Background

Previous Work

Telepathy

- RDMA-based Management Structure
- Telepathy Data Access Protocol

Experiments and Analysis

Conclusion

Discussion: Data Replication Protocols

Asynchronous (e.g. MongoDB [1])	• Read from primary/secondary; Write initiated at primary • Strong or Eventual consistency depending on the read protocol
Two-phase Commit (e.g. Ceph [2])	• Read from primary; Write initiated at primary • Strong consistency
Paxos/Raft (e.g. Cockroach [3], Spanner [4], Kudu [5])	• Read from primary or contact primary; Write initiated at primary • External or Snapshot consistency
Quorum (e.g. Dynamo [6], Cassandra [7])	• Read from any node; Write initiated at any node (quorum rule) • Eventual consistency
Pipeline (e.g. HDFS [8])	• Read and write need to contact name node • Strong consistency
Telepathy	• Read from any node; Write initiated at any node • Strong Consistency

Previous Work: RDMA in Distributed Storage Systems

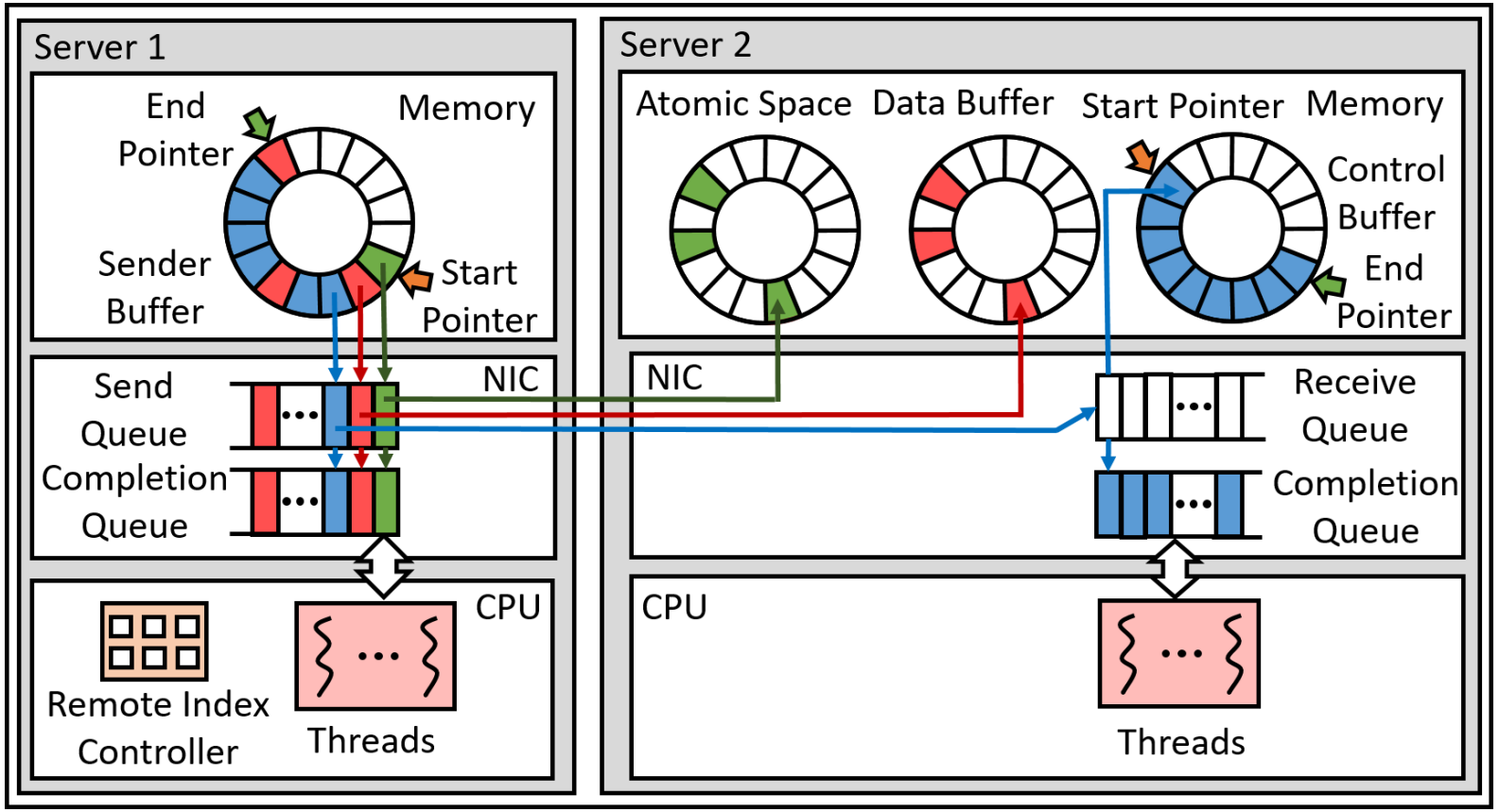
- Replace traditional socket-based channel with two-sided RDMA operations
  - Examples: Ceph [2], RDMA-based memcached [9], RDMA-based HDFS [10], FaSST [11] and Hotpot [12]
- Modify the lower-level communication mechanisms and related APIs
  - Examples: FaRM [13], Octopus [14], Derecho [15]
  - Redesign communication channels
    - Use one-sided RDMA pull for reads
    - Use one-sided RDMA push for writes
  - RDMC: An RDMA multicast pattern
- Common issue
  - Data access protocol itself is not changed
  - Benefits only from faster transmission speeds

Overview of Telepathy

- Data access protocol for distributed key-value storage systems in an NVRAM + RDMA cluster
- High-performance read/write protocol
  - Read from <u>any replica</u>
  - Write initiated at <u>any node</u>
- Strong consistency
  - Reads of an object to any replica return the value of the latest write
- Leverage RDMA features for data and control
  - RDMA Atomics for serializing read and write accesses to an object
  - 1-sided silent RDMA Writes and Reads
- Low CPU utilization

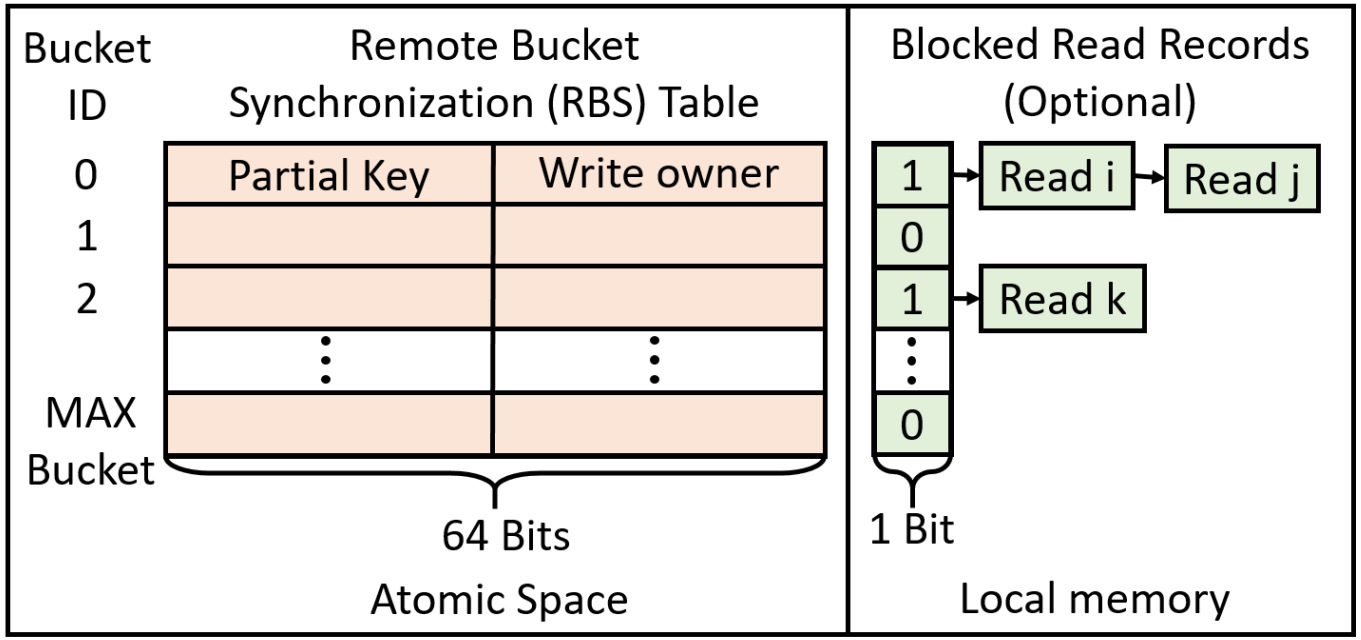
Decoupled Communication Channel (DCC)

- DCC is a novel communication channel for use in Telepathy
- NIC card automatically splits different message types at the hardware level
- Control messages use RDMA two-sided protocol and are consumed in FCFS order from the receiver's Control Buffer
- Data blocks use RDMA one-sided protocol and are consumed from the receiver's Data Buffer in an order specified by the sender application
- Atomic space is the registered memory region used to arbitrate concurrent updates from remote writers



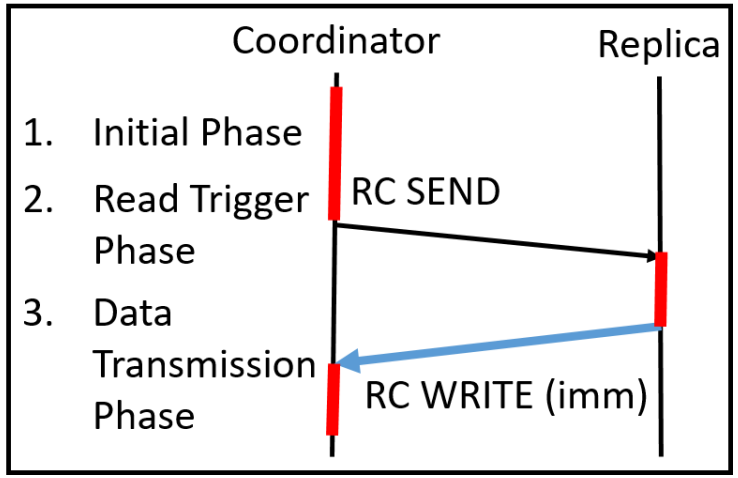
Remote Bucket Synchronization (RBS) Table

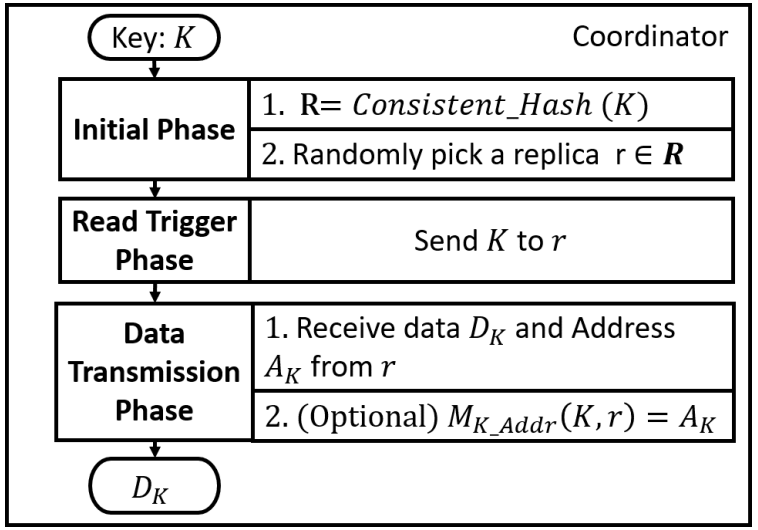
- Write serialization and read consistency are realized using a Remote Bucket Synchronization Table (RBS Table) in the atomic space region of Telepathy's registered memory
- RDMA atomic operation CAS is used to silently lock the bucket entry of the inflight update key
  - The low order bits of each entry hold the coordinator id of the update key
  - The high-order bits hold some bits of the update key and act as a Bloom Filter for detecting conflicting reads
- Blocked Read Records structure is used when livelock is detected in the default silent-reads fast path *i.e.* if the replica-based read protocol is triggered

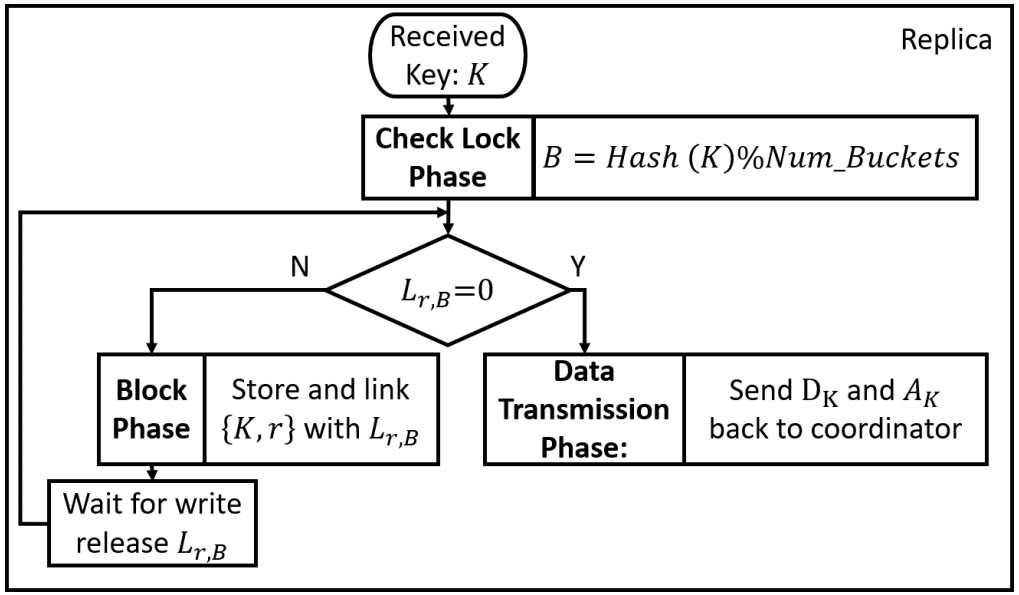


Read Protocol: Replica-based Read

- 3-Step Read Protocol
- Uses RDMA two-sided operations
- Replica nodes wake up for handling the read
- Two situations to use Replica-based Read
  - When the remote address of the data is not cached in the coordinator
  - A fallback path when livelock is detected in the Silent-Read protocol

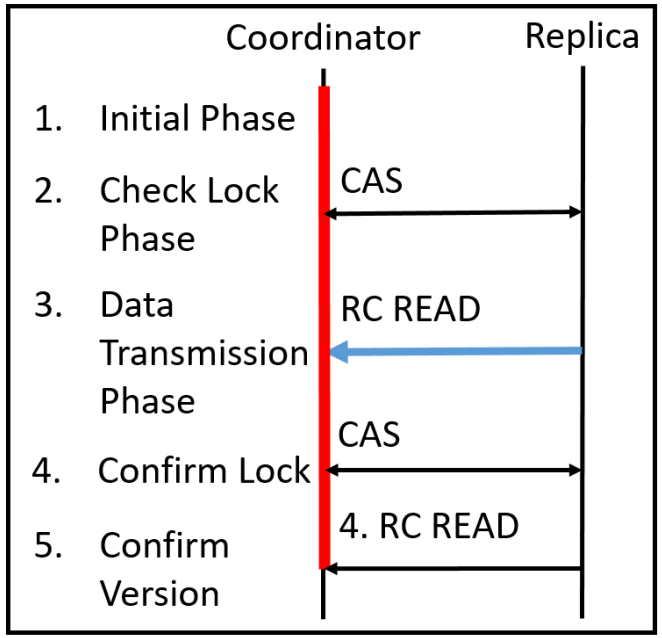


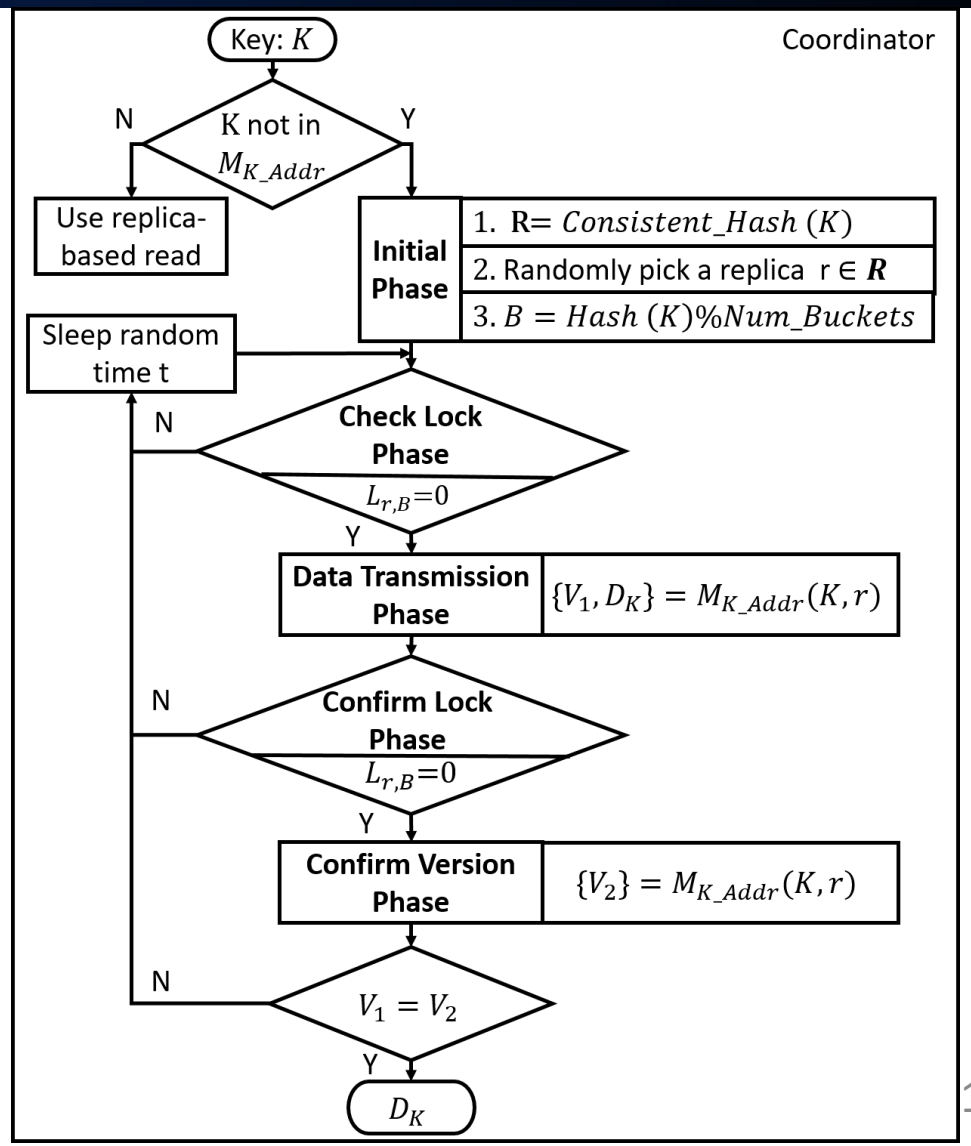




Read Protocol: Silent Read

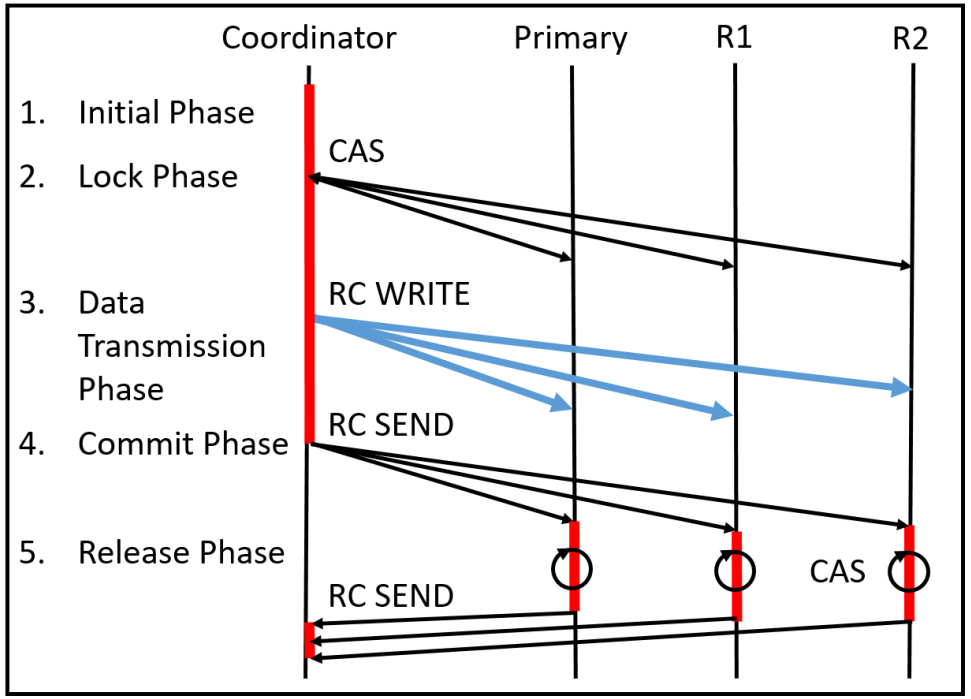
- 5-Step Silent Read Protocol
- Only RDMA one-sided semantics are used
- Replica nodes are not interrupted by read
- If strong consistency is not needed, reads can ignore the last version check to get snapshot isolation

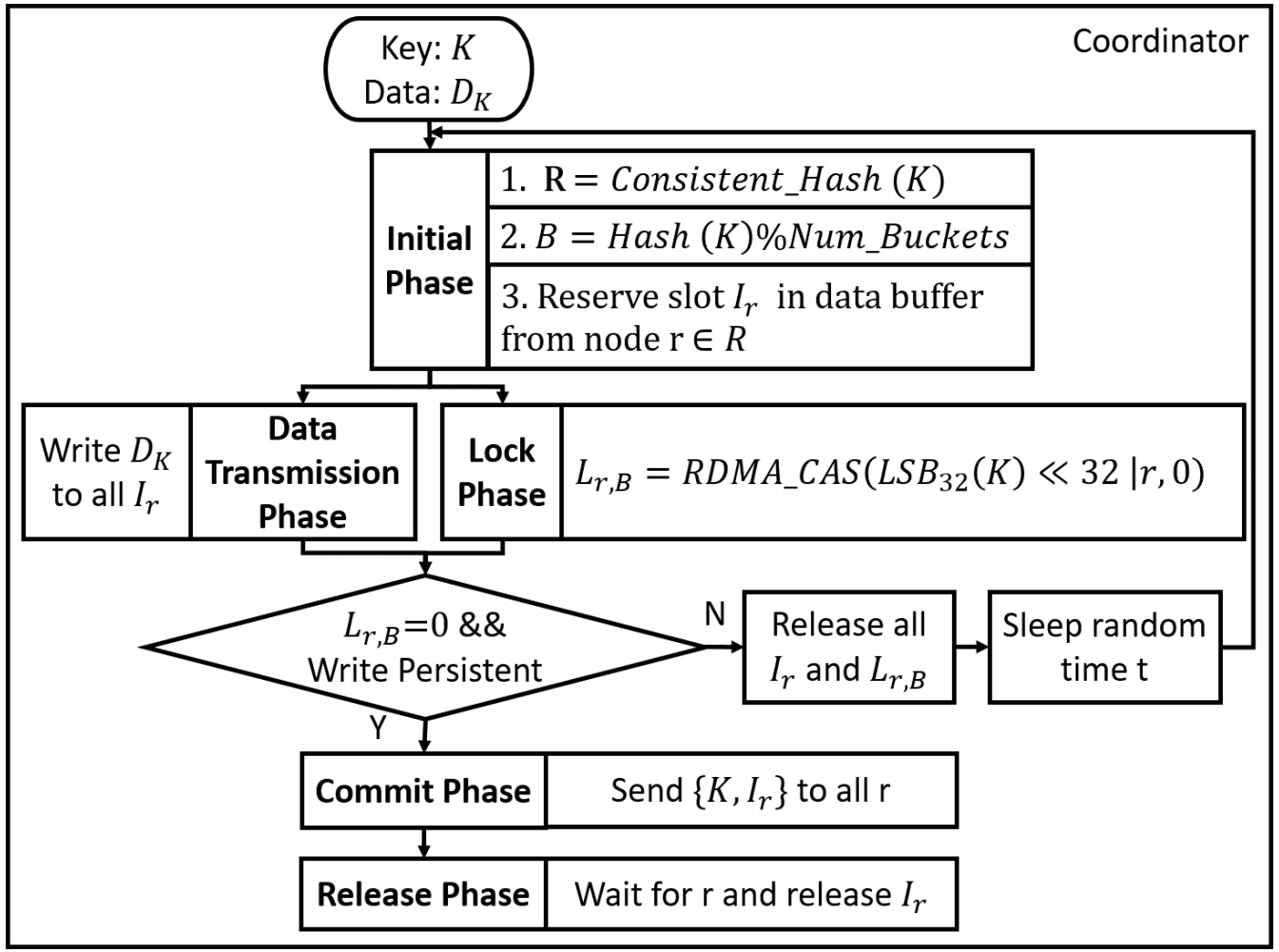




Write Protocol: Coordinator Side

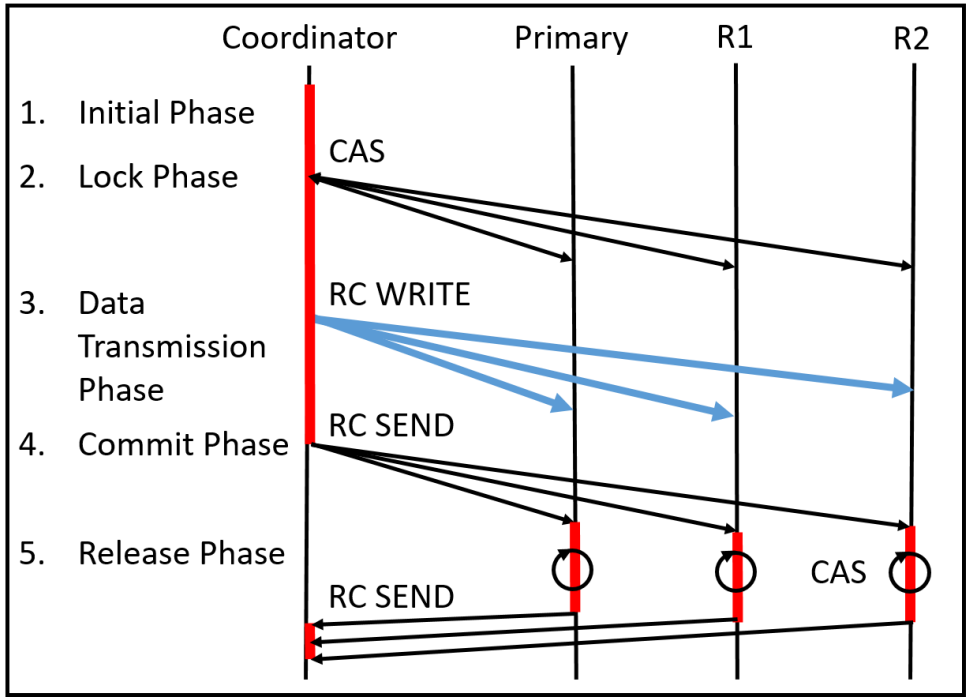
- At Coordinator side:
  - RDMA Atomics are used to silently resolve write conflicts among multiple coordinators
  - Silent data transmission is separated from control flow

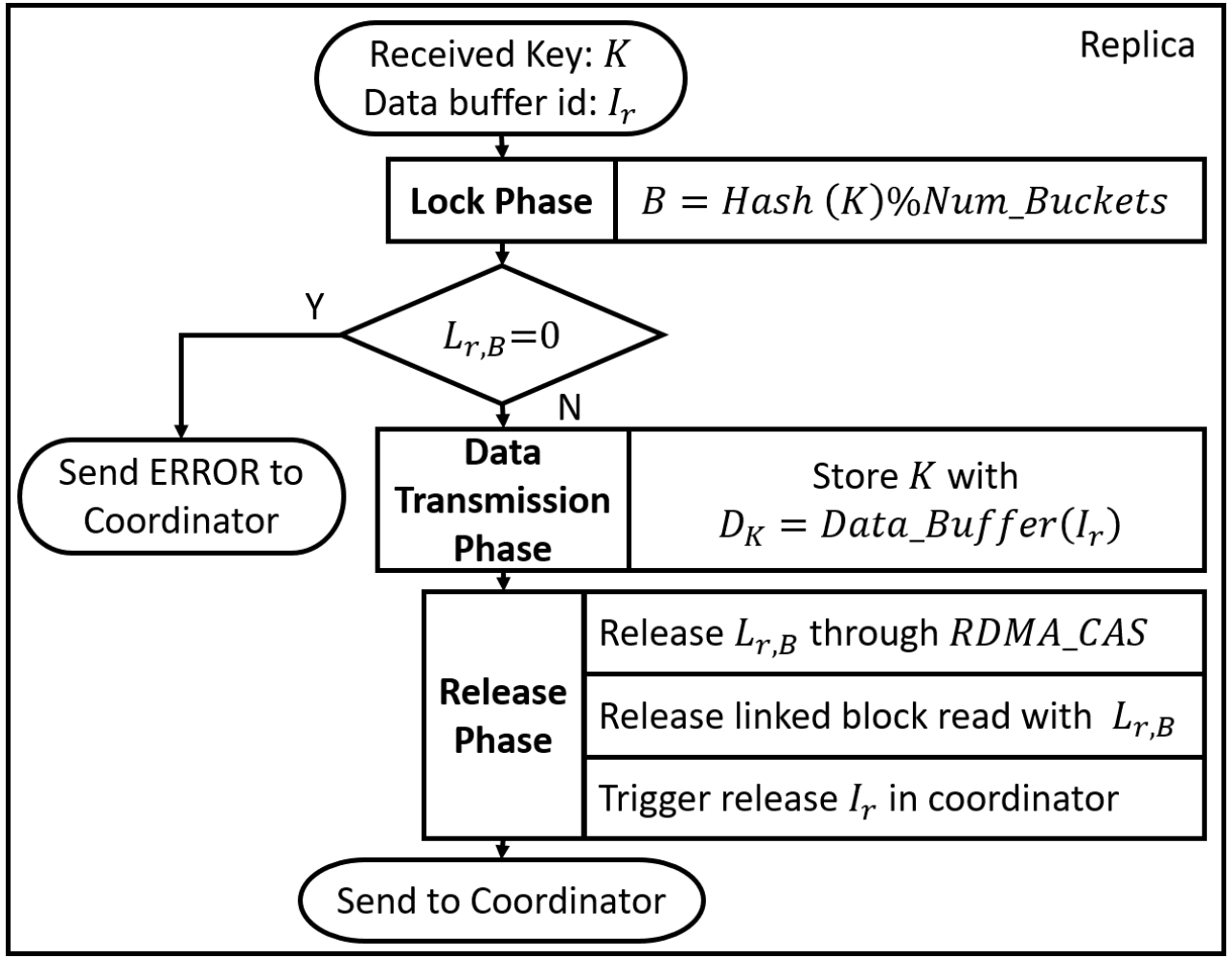




Write Protocol: Replica Side

- At Replica side:
  - ➢ CPU side will not be interrupted until the commit phase





Experimental Setup

- Telepathy is implemented on a cluster of servers connected through an Infiniband (IB) network
- The system is deployed on 12 servers in the Chameleon cluster infrastructure [16]
- The configuration of each server is shown as follows

CPU information	Intel(R) Xeon(R) CPU E5-2670 v3
Processor speed	2.30 GHz
Number of CPUs	48
Threads/Core	2
Memory size	128 GB
NIC Card	MT27500 Family [ConnectX-3]
Network	Infiniband

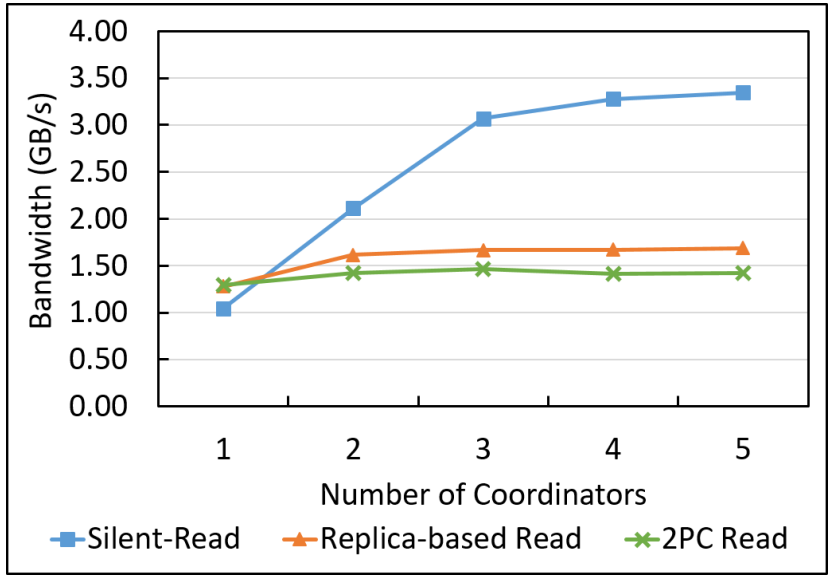
- DRAM is used as our storage backend, due to limitations of our testbed
- YCSB benchmark is used to evaluate our designs

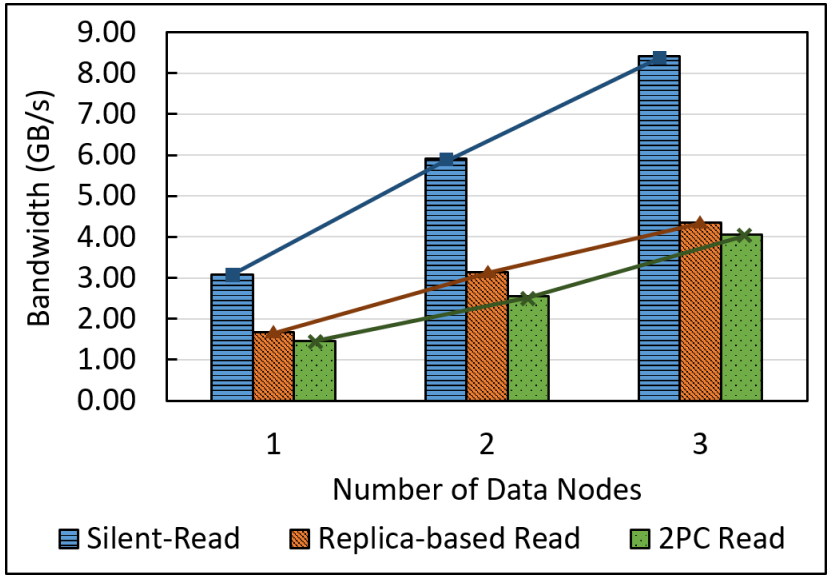
Comparison Protocol: 2 Phase Commit (2PC)

- RDMA two-sided operations are used to optimize the conventional 2PC protocol
- 2PC Read:
  - The coordinator directly sends the key to the primary to obtain the data
  - Primary send back data
- 2PC Write:
  - The coordinator sends the key-data pair together with the write command to the primary
  - Phase 1: Primary forwards the key-data pair to all replicas
  - Phase 2: After the primary receive replies from all replicas, it sends them commit messages

Bandwidth: Read Protocol

- Replay trace of 1 million pure reads from different coordinators
- Bandwidths of three different read protocols (Silent Read, Replica-based Read, 2PC) are compared

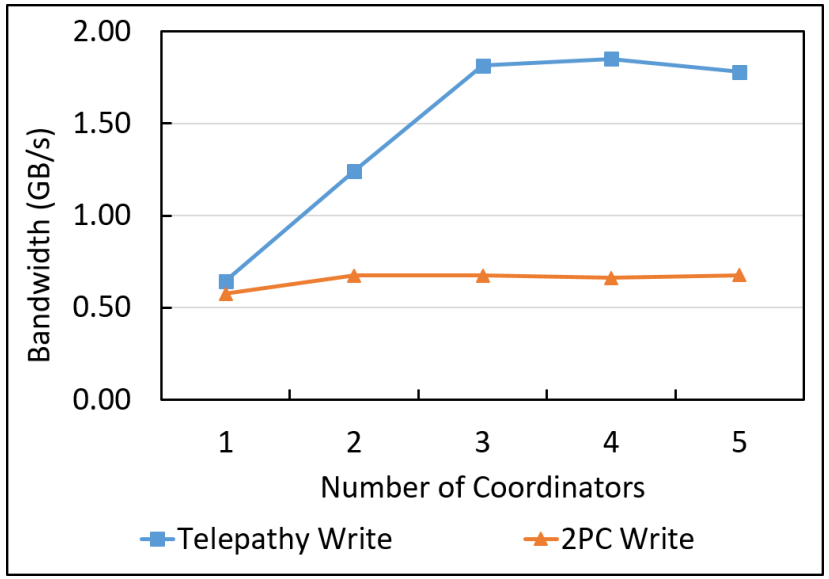


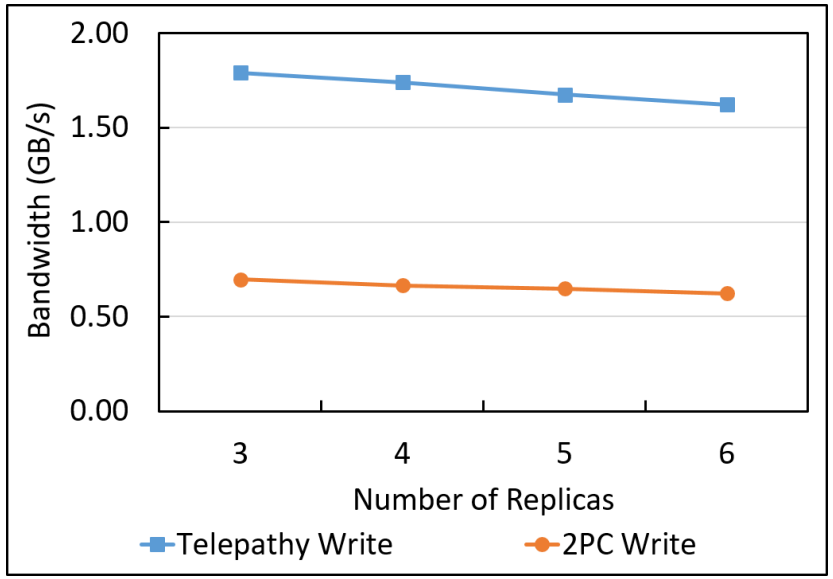


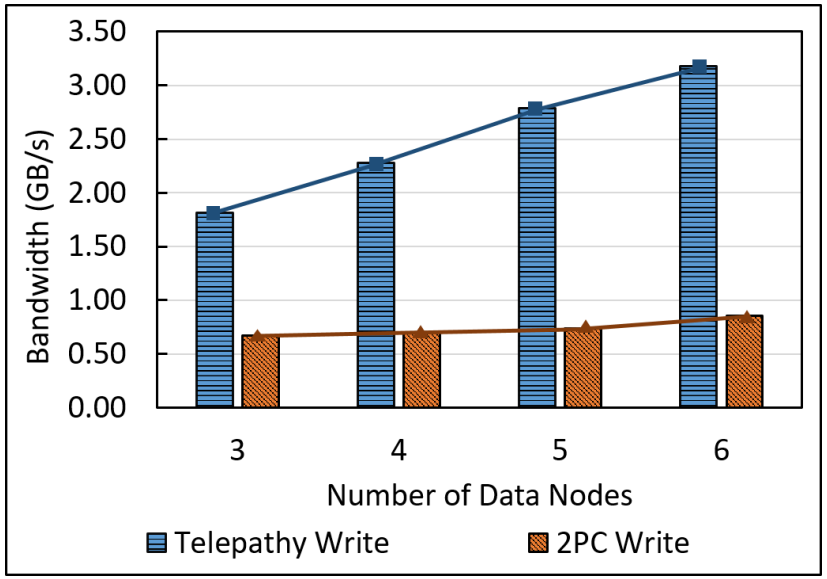
- Experiment 1:
  - Data nodes: 1
  - Replicas: 1
  - Coordinators: 1~5
- Experiment 2:
  - Data nodes: 3
  - Replicas: 3
  - Coordinators: 9

Bandwidth: Write Protocol

- Replay trace of 1 million pure writes from different coordinators
- Bandwidths of two different write protocols (Telepathy Write, 2PC Write) are compared



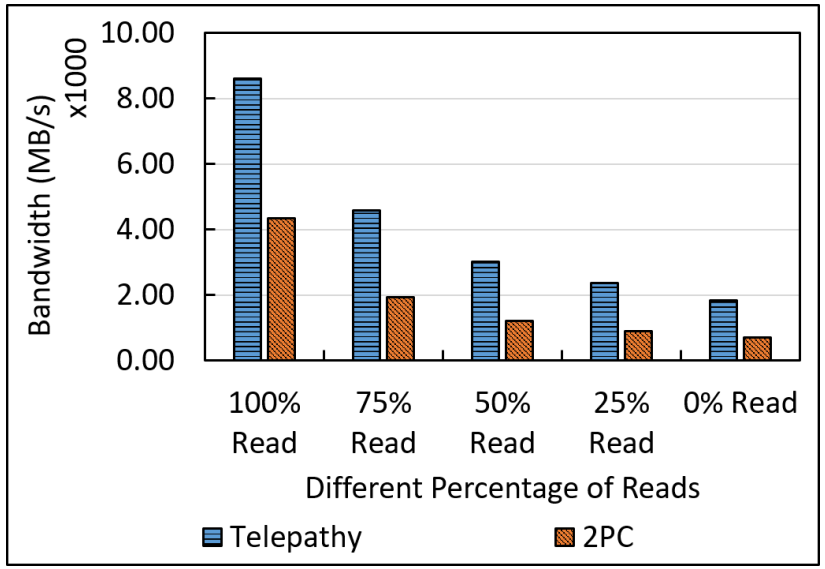




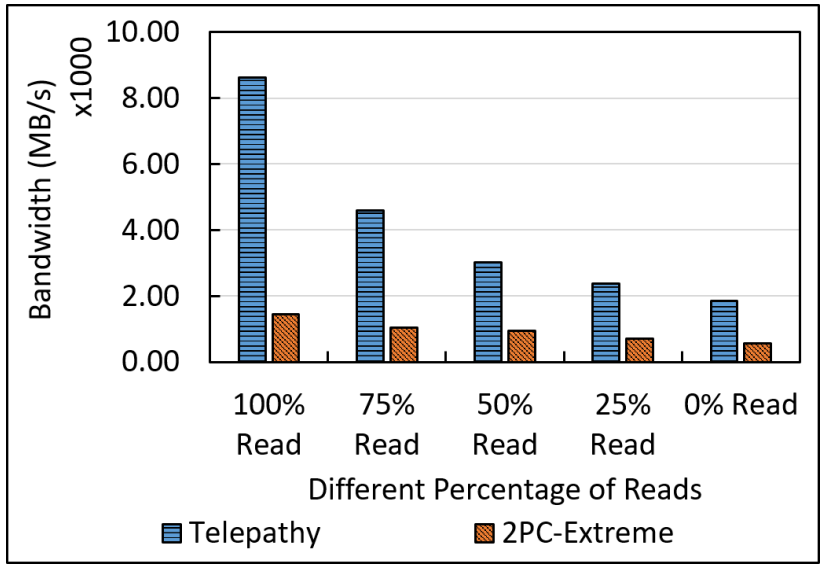
- Experiment 1:
  - Data nodes: 3
  - Replicas: 3
  - Coordinators: 1~5
- Experiment 2:
  - Data nodes: 6
  - Replicas: 3~6
  - Coordinators: 6
- Experiment 3:
  - Data nodes: 3~6
  - Replicas: 3
  - Coordinators: 9

Bandwidth: Uniform vs. Skewed Node Access

- Uniform Distribution
  - Data nodes have equal probability of being a primary node
- Skewed Distribution
  - Primary node are Zipf distributed with exponent set to 4
  - For three data nodes the probabilities for being a primary are 93%, 5.8% and 1.2%
- Experiments
  - Data nodes: 3
  - Replicas: 3
  - Coordinators: 9
  - Percentage of reads :0%, 25%, 50%, 75%, 100%



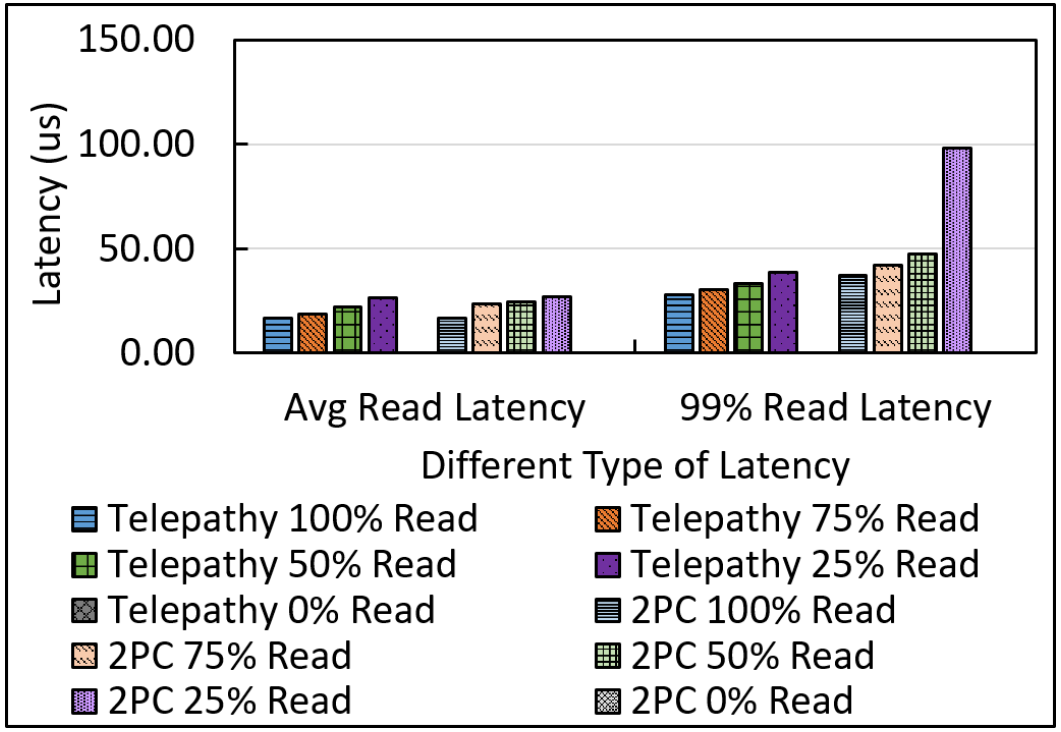
Uniform Case

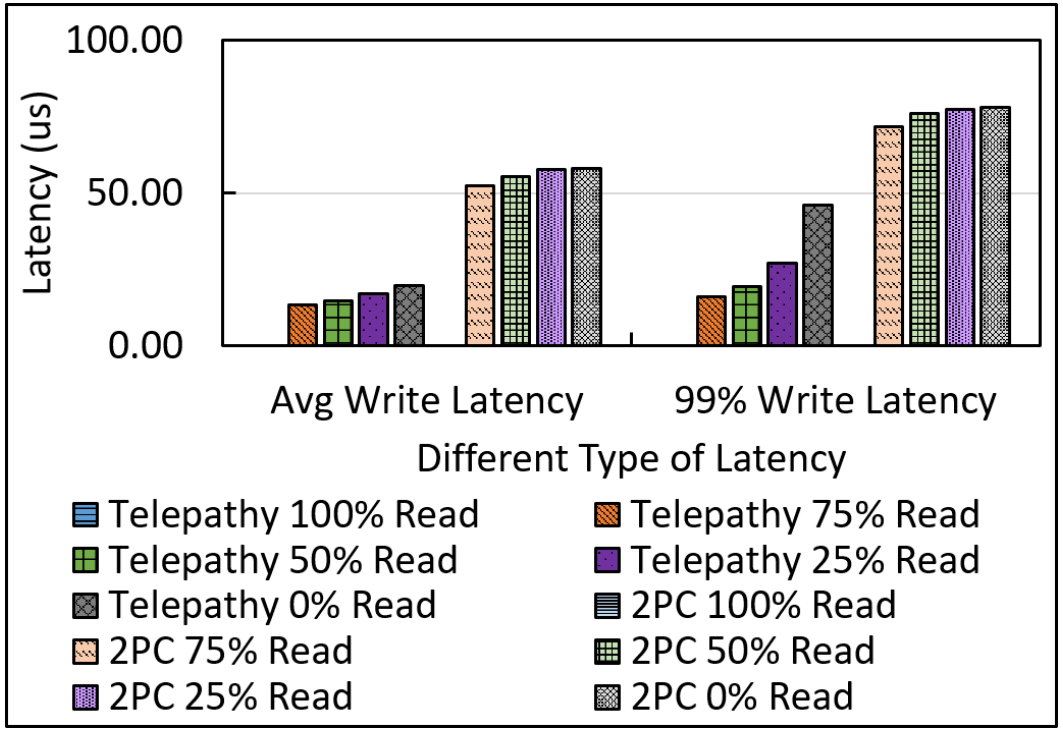


Skewed Case

Latency: Read & Write

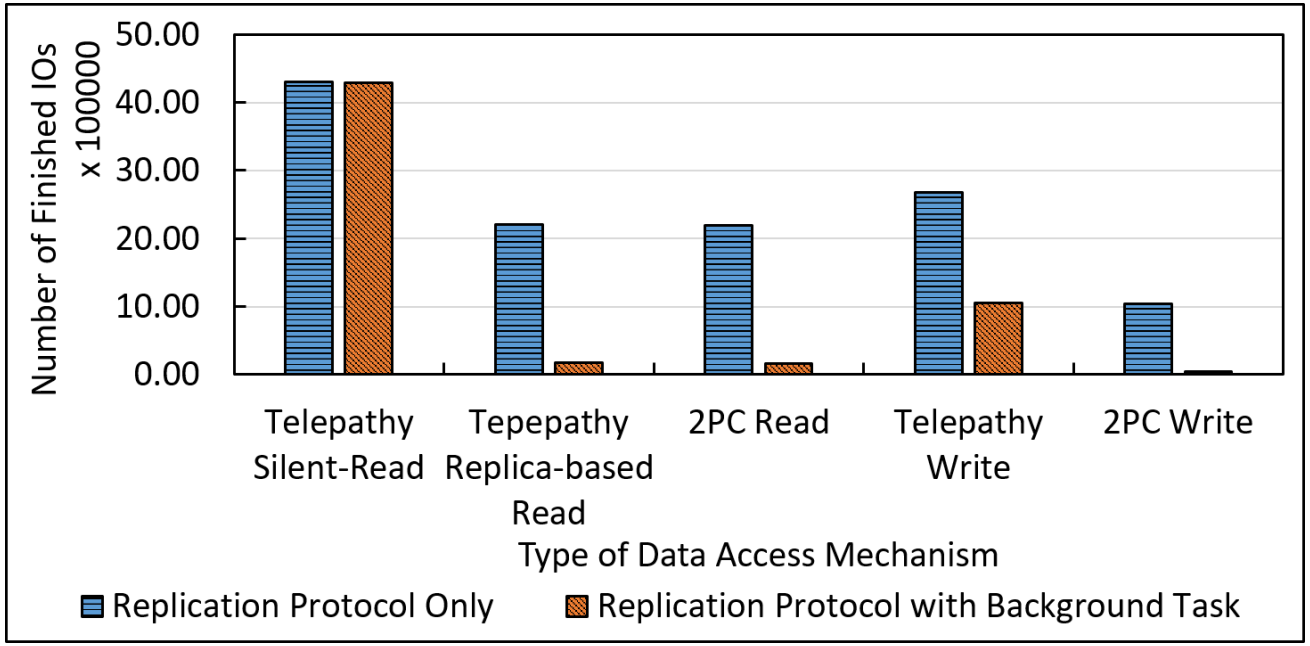
- Experiments
  - Data nodes: 3
  - Replicas: 3
  - Coordinators: 9
  - Percentage of reads :0%, 25%, 50%, 75%, 100%





CPU Efficiency Improved by Telepathy

- Experiments
  - Data nodes: 3
  - Replicas: 3
  - Coordinators: 9
  - Run a CPU-intensive background task on each core of all servers
  - Number of IOs completed with and without the background task for 100% reads and 100% writes



Conclusion

- Telepathy is a novel data replication and access mechanism for RDMA-based distributed KV stores
- Telepathy is a fully distributed mechanism
  - IO writes are handled by any server
  - IO reads are served by any of the replicas
- Strong consistency is guaranteed while providing high IO concurrency
- Hybrid RDMA semantics are used to directly and efficiently transmit data to target servers
- Telepathy can achieve low IO latency and high throughput, with extremely low CPU utilization

Reference

[1] K. Chodorow, MongoDB: The Definitive Guide: Powerful and Scalable Data Storage. ” O’Reilly Media, Inc.”, 2013.

[2] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. Long, and C. Maltzahn, “Ceph: A scalable, high-performance distributed file system,” in Proceedings of the 7th symposium on Operating systems design and implementation. USENIX Association, 2006, pp. 307–320. . 205–220, 2007.

[3] C. Labs, “CockroachDB: Ultra-resilient SQL for global business,” in https://www.cockroachlabs.com/, 2018.

[4] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild et al., “Spanner: Googles globally distributed database,” ACM Transactions on Computer Systems (TOCS), vol. 31, no. 3, p. 8, 2013.

[5] T. Lipcon, D. Alves, D. Burkert, J.-D. Cryans, A. Dembo, M. Percy, S. Rus, D. Wang, M. Bertozzi, C. P. McCabe et al., “Kudu: Storage for fast analytics on fast data,” 2015.

[6] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, “Dynamo: amazon’s highly available key-value store,” ACM SIGOPS operating systems review, vol. 41, no. 6, pp. 205–220, 2007.

[7] A. Lakshman and P. Malik, “Cassandra: a decentralized structured storagesystem,”ACMSIGOPSOperatingSystemsReview,vol.44,no.2, pp. 35–40, 2010.

[8] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on. IEEE, 2010, pp. 1–10.

[9] J. Jose, H. Subramoni, M. Luo, M. Zhang, J. Huang, M. Wasi-ur Rahman, N. S. Islam, X. Ouyang, H. Wang, S. Sur et al., “Memcached design on high performance rdma capable interconnects,” in 2011 International Conference on Parallel Processing. IEEE, 2011, pp. 743– 752.

[10] N. S. Islam, M. W. Rahman, J. Jose, R. Rajachandrasekar, H. Wang, H. Subramoni, C. Murthy, and D. K. Panda, “High performance RDMA-based design of HDFS over InfiniBand,” in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE Computer Society Press, 2012, p. 35.

[11] A. Kalia, M. Kaminsky, and D. G. Andersen, “FaSST: Fast, Scalable and Simple Distributed Transactions with Two-Sided RDMA Datagram RPCs,” in 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 185–201.

[12] Y. Shan, S.-Y. Tsai, and Y. Zhang, “Distributed shared persistent memory,” in Proceedings of the 2017 Symposium on Cloud Computing, 2017, pp. 323–337.

[13] A. Dragojevi´c, D. Narayanan, M. Castro, and O. Hodson, “FaRM: Fast remote memory,” in 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14), 2014, pp. 401–414.

[14]Y. Lu, J. Shu, Y. Chen, and T. Li, “Octopus: an rdma-enabled distributed persistent memory file system,” in 2017 USENIX Annual Technical Conference (USENIX ATC 17), 2017, pp. 773–785.

[15] S. Jha, J. Behrens, T. Gkountouvas, M. Milano, W. Song, E. Tremel, R. V. Renesse, S. Zink, and K. P. Birman, “Derecho: Fast state machine replication for cloud services,” ACM Transactions on Computer Systems (TOCS), vol. 36, no. 2, pp. 1–49, 2019.

[16] N. S. Foundation, “A configurable experimental environment for largescale cloud research,” in https://www.chameleoncloud.org/, 2019.