Key-value-stores from the Lens of Shared Memory

URCS Systems Seminar Discussion

Alan Beadle and Haosen Wen

May 29, 2020

Key-value-stores from the Lens of Shared Memory

ACM SIGARCH article published May 12, 2020

Vijay Nagarajan and Boris Grot are Associate Professors; Vasilis Gavrielatos and Antonis Katsarakis are senior PhD students, all from the University of Edinburgh.

Two approaches

- Shared nothing vs shared memory
- Either way, authors argue that architects can add value using insights from similar problems in hardware

Coherence-inspired Replication (1/4)

- Failures in datacenter are infrequent (2.5 hours for 2000 servers at Google)
- Optimization tradeoffs involving normal fast path and slow path for handling failures
- Again, perhaps insight from hardware design is useful

Coherence-inspired Replication (2/4)

- Chain replication (CRAQ[1], Chain Replication with Apportioned Queries) tries to balance failure tolerance with good fast-path performance

[1] Object Storage on CRAQ. Jeff Terrace and Michael J. Freedman, Princeton University (ATC '09)

Coherence-inspired Replication (3/4)

- Distributed object-storage system
- A write propagates to replicas in predetermined chain order
- Permits local reads but has high write latency

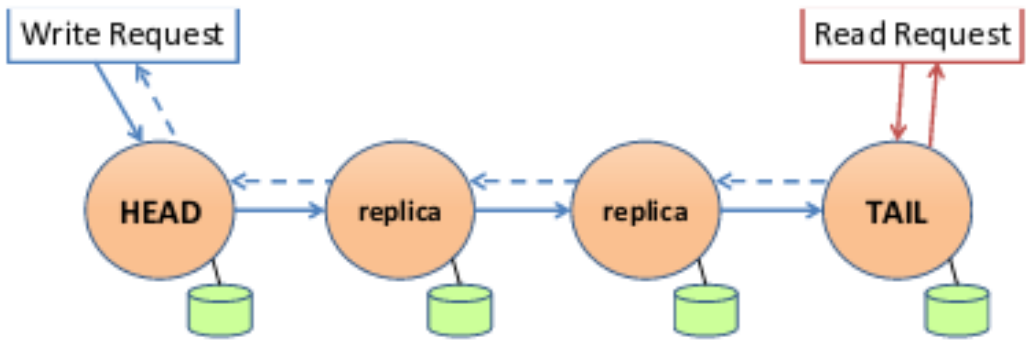

Figure 1: **All reads in Chain Replication must be handled by the tail node, while all writes propagate down the chain from the head.**

Coherence-inspired Replication (4/4)

- Hermes[1] uses a coherence-inspired timestamp approach; invalidations propagate to all replicas concurrently

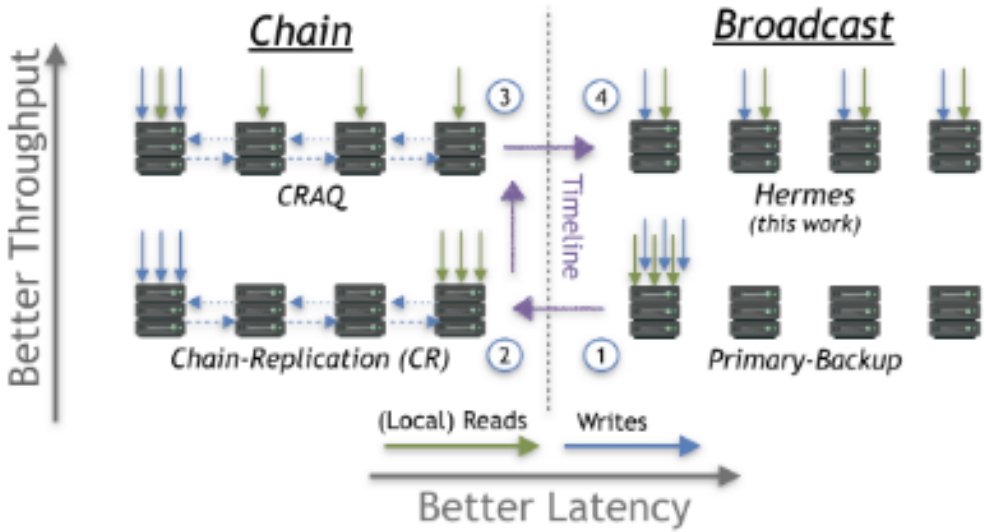

Figure 1. *Comparison of reliable membership-based protocols in terms of throughput and latency.*

[1] Hermes: a Fast, Fault-Tolerant and Linearizable Replication Protocol. Antonios Katsarakis, Vasilis Gavrielatos, M. R. Siavash Katebzadeh,Arpit Joshi (Intel), Aleksandar Dragojevic (Microsoft Research), Boris Grot, Vijay Nagarajan. University of Edinburgh. (ASPLOS '20)

Network Ordering (1/3)

- Some coherence protocols use ordering guarantees provided by the interconnect
- Similar to bus-based snooping

Network Ordering (2/3)

- Network-ordered Paxos (NOPaxos)[1]is a protocol for state machine replication based on ordered unreliable multicast (OUM) primitive
- "exploits network ordering to provide strongly consistent replication without coordination"

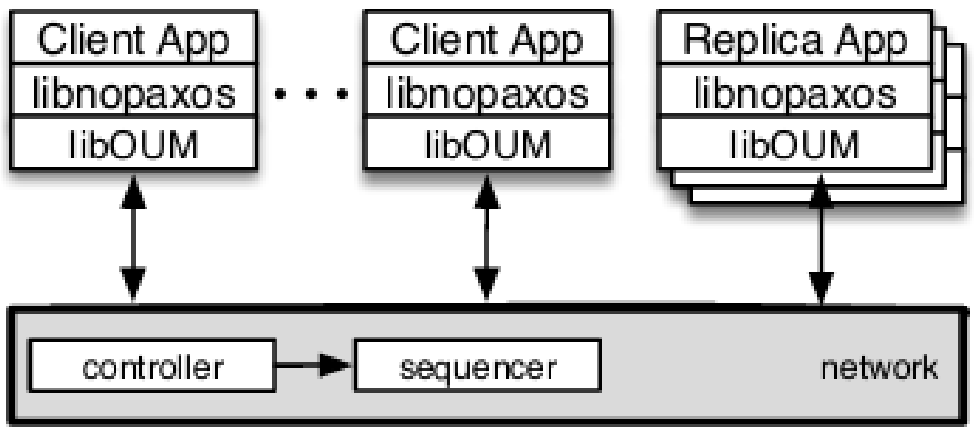

Figure 1: Architecture of NOPaxos.

[1]Just say no to paxos overhead: replacing consensus with network ordering. Jialin Li, Ellis Michael, Naveen Kr. Sharma, Adriana Szekeres, and Dan R. K. Ports,University of Washington (OSDI '16)

Network Ordering (3/3)

- Network-imposed ordering always limits concurrency
- Maybe architectural support for high-throughput ordering can help

Shared Memory-Inspired Consistency (1/4)

- Borrowing shared memory's consistency model to distributed storage
  - Synchronizations are much more expensive in a distributed world
  - Replications in each part (node or cluster) of the system are the ground of inconsistency
  - Trade-offs between programmability and potential efficiency

Shared Memory-Inspired Consistency (2/4)

- Design space of memory consistency model: strength and scope
- Strength: Lin $>$ SC $>$ Causal(+) $>$ Release $>$ Eventual
  - Stronger consistency model offers better programmability, but expensive
- Scope: HRF(heterogeneous race-free)-direct VS HRF-indirect

Shared Memory-Inspired Consistency (3/4)

- Causal/Causal+[1]: Serializations respect both program orders and causality orders, which is determined by write-into order
- Weak Ordering[2]: Hardware appears SC if program obey some synchronization model, e.g. data-race-free
  - Release Consistency[3]: Sessions sandwiched by acquires and releases appear SC

---

[1] Don't settle for eventual: scalable causal consistency for wide-area storage with COPS. Wyatt Lloyd and Michael J Freedman, Princeton University; Michael Kaminsky, Intel Research; David G. Andersen, CMU (SOSP'11)

[2] Weak ordering — a new definition. Sarita V. Adve and Mark D. Hill, University of Wisconsin, Madison (ISCA'90)

[3] Kite: efficient and available release consistency for the datacenter. Vasilis Gavrielatos, Antonios Katsarakis, Vijay Nagarajan, and Boris Grot, the University of Edinburgh; Arpit Joshi, Intel (PPoPP'20)

Shared Memory-Inspired Consistency (4/4)

- Scoped consistency model: sequential consistency for heterogeneous race-free (SC for HRF)[1]:
  - Direct: accesses to the same key from different scopes are always considered races
  - Indirect: accesses to the same key are races if they are unordered considering both synchronization orders in the same scope and program orders.

[1] Heterogeneous-race-free Memory Models. Derek R. Hower, Blake A. Hechtman, Bradford M. Beckmann, Benedict R. Gaster, Mark D. Hill, Steven K. Reinhardt, and David A. Wood, AMD Research (ASPLOS'14)

Protocol Offloading (1/2)

- In shared-memory microprocessors, coherence protocols use dedicated controllers instead of taking up CPU cycles
- In data centers, CPUs might become a bottleneck for coherence protocol actions

Protocol Offloading (2/2)

- "Consensus in a box"[1] offloads this to an FPGA
- Architects can help identify consistency primitives for datacenters

[1] Consensus in a Box: Inexpensive Coordination in Hardware. Zsolt István, David Sidler, and Gustavo Alonso, ETH Zürich; Marko Vukolić, IBM Research—Zürich (NSDI '16)

Do I need backup slides?