Accelerating Kubernetes with In-network Caching

Stefanos Sagkriotis[1], Dimitrios Pezaros[2]

School of Computing Science, University of Glasgow, UK

[1]s.sagkriotis.1@research.gla.ac.uk, [2]dimitrios.pezaros@glasgow.ac.uk

Background

- Kubernetes relies on etcd to obtain a consistent Key-Value Store (KVS) used to coordinate services and control-plane components. Etcd is based on Raft, a quorum-based platform that **lacks horizontal scalability** [1][3]. For example, write latency in etcd can go from 80ms for a cluster of 9 nodes to 160ms for a cluster of 21 nodes. [2]
- **However, a** KVS deployed in Tofino ASIC using P4 can provide **sub-Round Trip Time responses** in μs latency, as shown by NetChain [3]. We attempt to leverage the performance improvements to accelerate Kubernetes operations and benefit the platform's scalability.
- Kubernetes generates read-mostly workloads during pod scaling. Only a third of the generated queries are writes.

Contributions

1. Identification of shortcoming in previous state-of-the-art (SotA) in-network replication frameworks.
2. Implementation of a replication mechanism that promotes scalability and reduced latency while maintaining consistency.
3. Proposal of a Kubernetes design that utilises in-network caching to enhance scalability, provide higher throughput and reduced latency.

Design

Programmable Data Plane (PDP)

- Scalability
- Is previous SotA scalable? – Due to the Chain Replication method, scalability is bottlenecked to the reply rate of the reference node (tail).
- Can we do better? – CRAQ allows each node to generate replies if a Key-Value (KV) pair is clean, i.e., no pending commits.

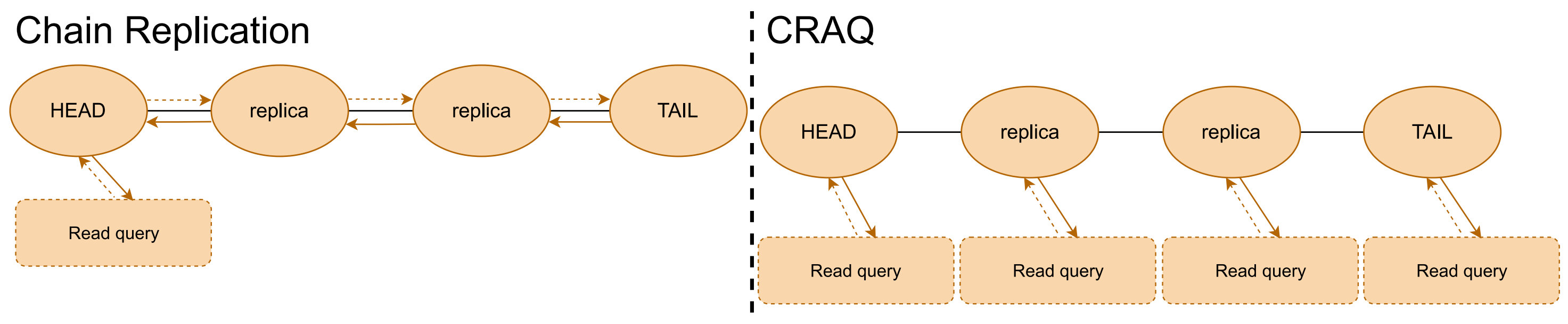

Fig. 1: Read queries in Chain Replication vs CRAQ

- Packet parsing
- Can be extensive due to forwarding to reference node (tail). CRAQ minimises this by forwarding only dirty queries.
- A minimal packet header is utilised with few overhead fields: 20 bytes (16 bytes for value) vs 72 bytes for NetChain.
- Control plane
- Manages all chain-related information: forwarding rules, participating nodes, roles (head, tail, replica). The client can be chain-agnostic.
- Removes information from packet headers, shortening parsing times.

Kubernetes

- API requests monitoring
- Etcd monitoring by scraping Prometheus.
- Data plane measurements
- P4Runtime used to identify hot KV pairs through P4 counters.
- Decision making based on traffic statistics
- Most popular KV pairs to be placed in PDP provided there is enough space

Overview

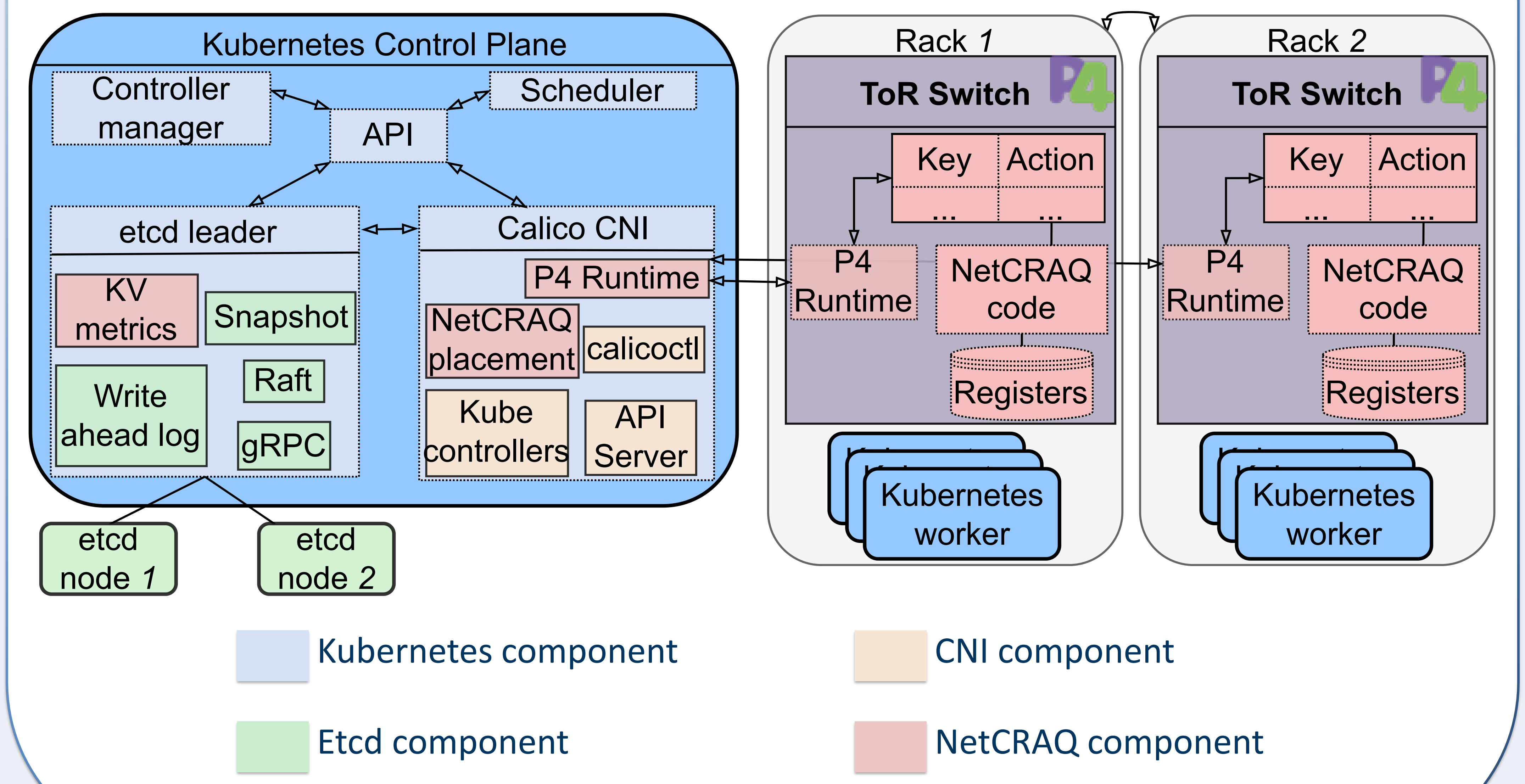

Fig. 2: Overview of proposed design

Evaluation

NetCRAQ implemented using P4. Tested using Mininet, Bmv2, P4utils.

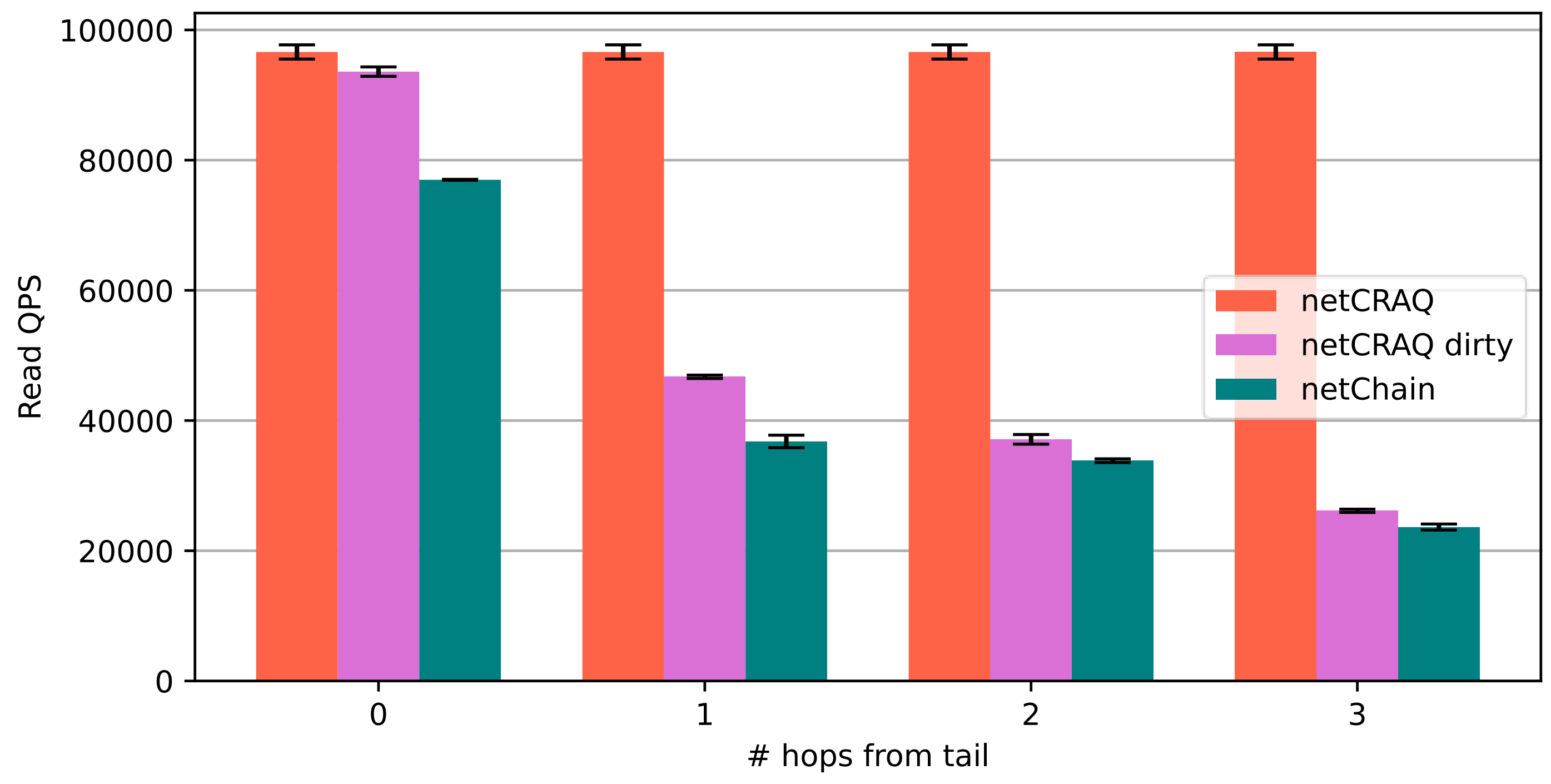

Fig. 3: Read throughput vs Distance from tail

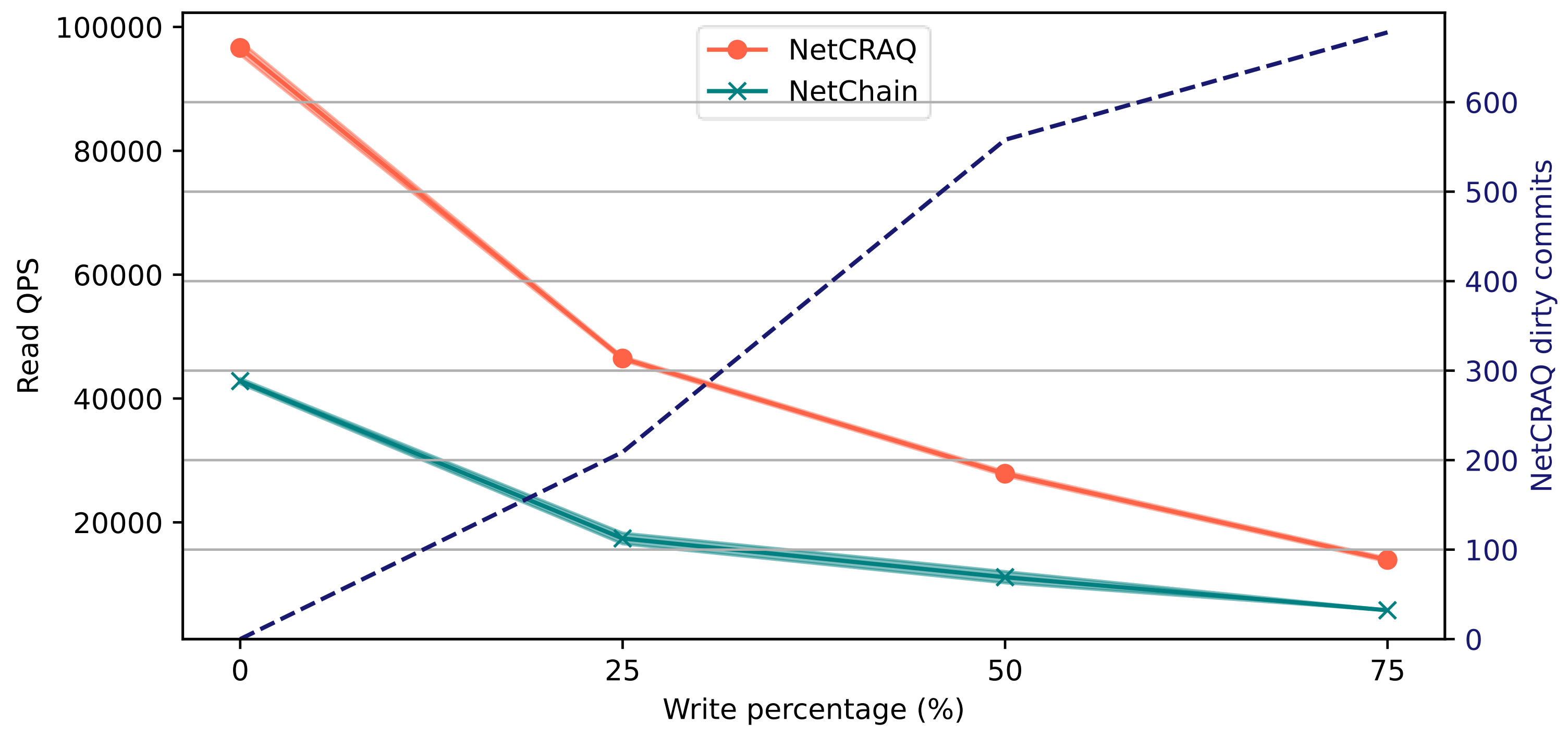

Fig. 4: Read throughput with mixed read/writes

Future Steps

1. Transfer implementation to the Tofino ASIC and evaluate performance differences in hardware. Optimisations stem from reduced number of hops and parsing therefore we expect emulation results to be indicative.
2. Conclude work on Kubernetes components that redirect queries to PDP and evaluate the performance difference between the default setup and a setup that utilises in-network replication.
3. Examine alternative use cases and evaluate their performance.

References

[1] Ricardo Jiménez-Peris, M. Patiño Martínez, Gustavo Alonso, and Bettina Kemme. 2003. Are Quorums an Alternative for Data Replication? ACM Trans. Database Syst. 28, 3 (sep 2003), 257–294. https://doi.org/10.1145/937598.937601

[2] Andrew Jeffery, Heidi Howard, and Richard Mortier. 2021. Rearchitecting Kubernetes for the Edge. In Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking (Online, United Kingdom) (EdgeSys '21). Association for Computing Machinery, New York, NY, USA, 7–12. https://doi.org/10.1145/3434770.3459730

[3] Xin Jin, Xiaozhou Li, Haoyu Zhang, Nate Foster, Jeongkeun Lee, Robert Soulé, Changhoon Kim, and Ion Stoica. 2018. NetChain: Scale-Free Sub-RTT Coordination. In 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18). USENIX Association, Renton, WA, 35–49. https://www.usenix.org/conference/nsdi18/presentation/jin

[4] Robbert van Renesse and B. Schneider. 2004. Chain Replication for Supporting High Throughput and Availability. In 6th Symposium on Operating System Design and Implementation (OSDI 2004), San Francisco, California, USA, December 6-8, 2004, Eric A. Brewer and Peter Chen (Eds.). USENIX Association, 91–104. http://www.usenix.org/events/osdi04/tech/renesse.html