Migrating Key-Value Data Stores to the Edge
Orchestration platforms and their Key-Value stores

MESOS

Kubernetes

HashiCorp
Nomad

Apache ZooKeeper™
etcd

HashiCorp
Consul
Motivation

K8s
The cloud

K3s
Single-site
Isolated

KubeEdge
Cloud-centric
Blast radius

K8s
The edge

Control plane + etcd

Worker
etcd is the problem

3 nodes, leader partitioned between $t=5$ and $t=10$, 10ms link delay, successful requests, 10,000 rps
Avoiding coordination with *dismerge*

- All nodes are leaders
- Leaders are now local
- Replication is lazy

Bringing datastore nodes closer to clients with *dismerge*
**dismerge** is a solution

3 nodes, leader partitioned between $t=5$ and $t=10$, 10ms link delay, successful requests, 10,000 rps
Handling conflicts with custom datatypes

Coarse grained
Using raw values

Fine grained
Introspecting values
Linking back to orchestrators

\[ \text{etcd} \rightarrow \text{dismerge} \]

- Local operation
- Availability
- Resiliency

Some modifications needed to orchestrators e.g. StatefulSets
Conclusion

Orchestration platforms are being *pushed to*, not *designed for* the edge

*dismerge* places new edge-focused orchestration platforms on strong foundations, providing local operation, availability and resiliency.