Distributed Key-Value Stores: Performance and Scalability for Flash Media

Richard Elling

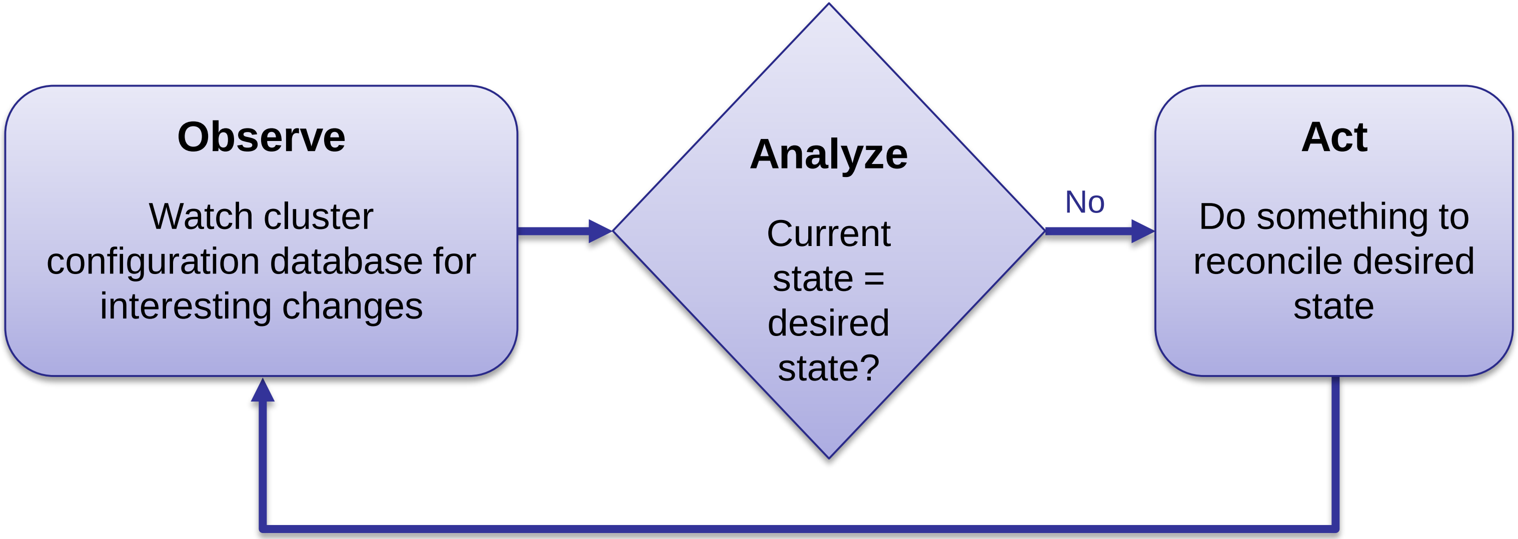
Key-Value (KV) Databases Overview

- Simple is important
  - Fast, easy to use
- Consistency is a differentiator
  - Strong consistency
    - Important for control planes
    - Examples: etcd, consul, zookeeper
  - Weak consistency (aka eventually consistent, NoSQL)
    - Popular for large-scale applications
    - Examples: dynamo, cosmos db

Kubernetes and KV

- Distributed system scheduler for containers
- How it works
  - Declare desired state in key-value database
  - Controllers (aka operators) watch the state and act to change current state to desired state
- End result
  - Billions of applications served reliably
  - Clusters up to 5,000 nodes each

Kubernetes Controllers/Operators



Observe

Watch cluster configuration database for interesting changes

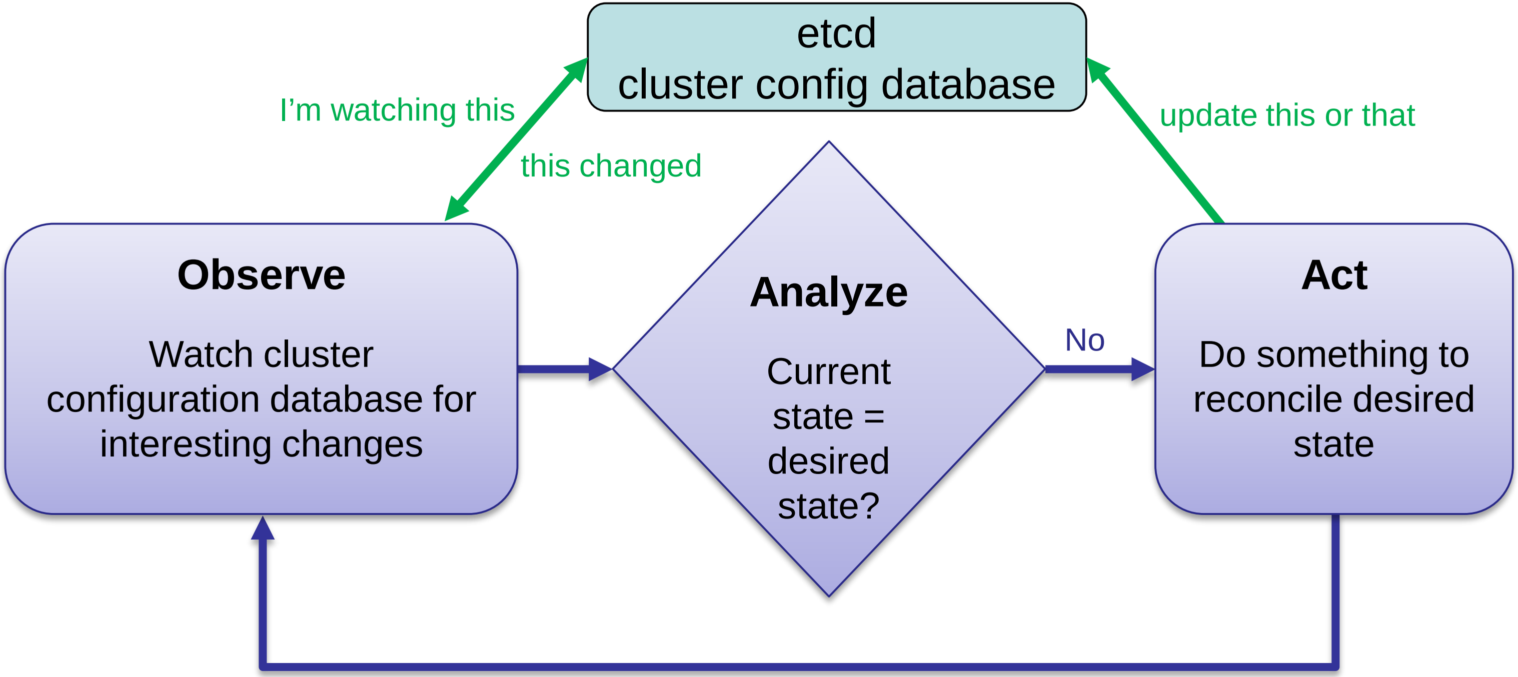
Analyze

Current state = desired state?

Act

Do something to reconcile desired state

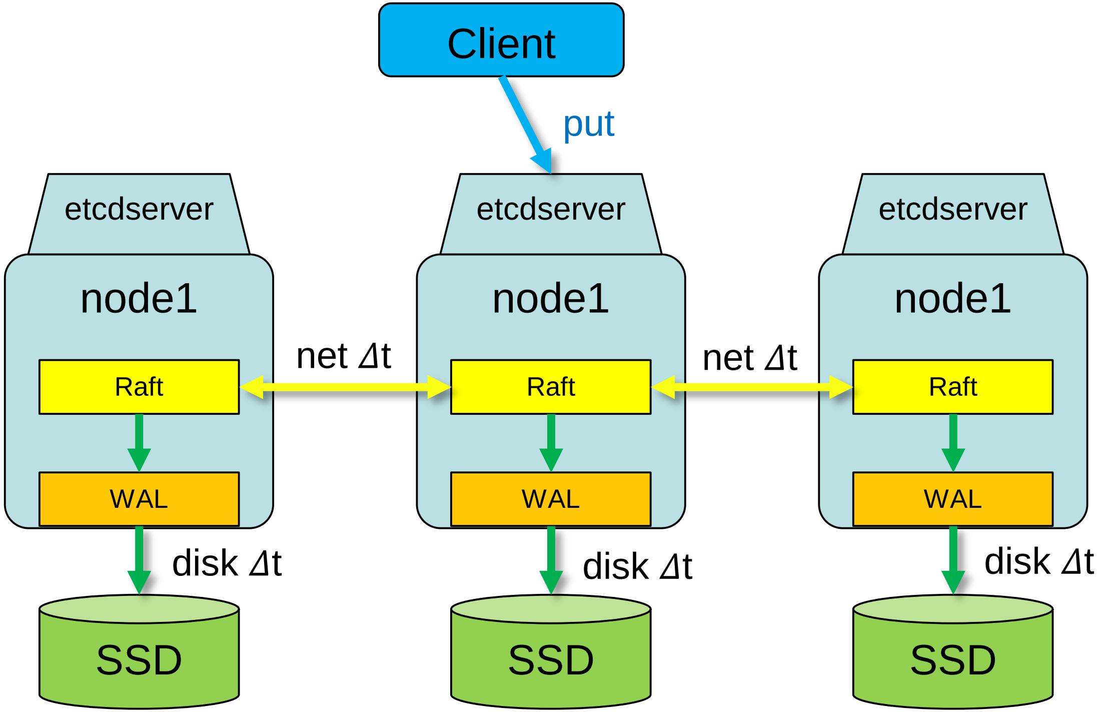
Kubernetes Controllers/Operators


etcd
cluster config database
I'm watching this
this changed
update this or that
Observe
Watch cluster configuration database for interesting changes
Analyze
Current state = desired state?
No
Act
Do something to reconcile desired state

Etcd Cluster Configuration Database

Requirements	Etcd in Kubernetes
Consistency	Single-writer Updates acknowledged by quorum of masters
Availability	Multiple-masters
Partition tolerance	Raft protocol ensures all masters are consistent
Performance	Writes committed to persistent storage log Reads satisfied by any master

Etcd Persistent Storage

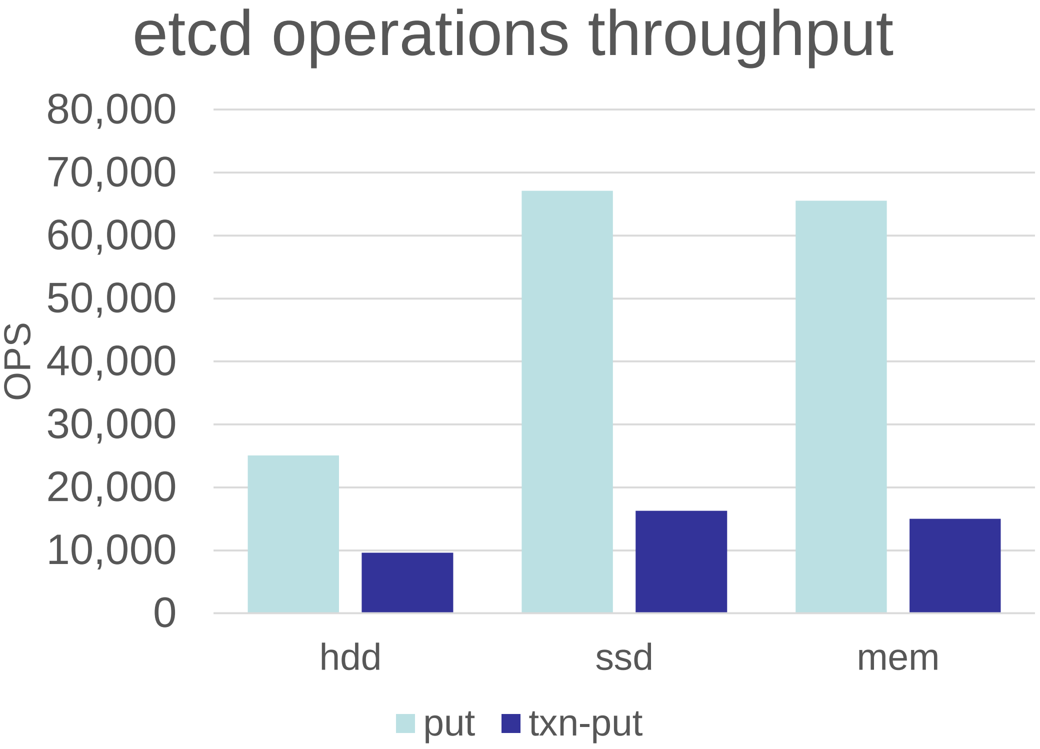


Kubernetes Practical Implementations

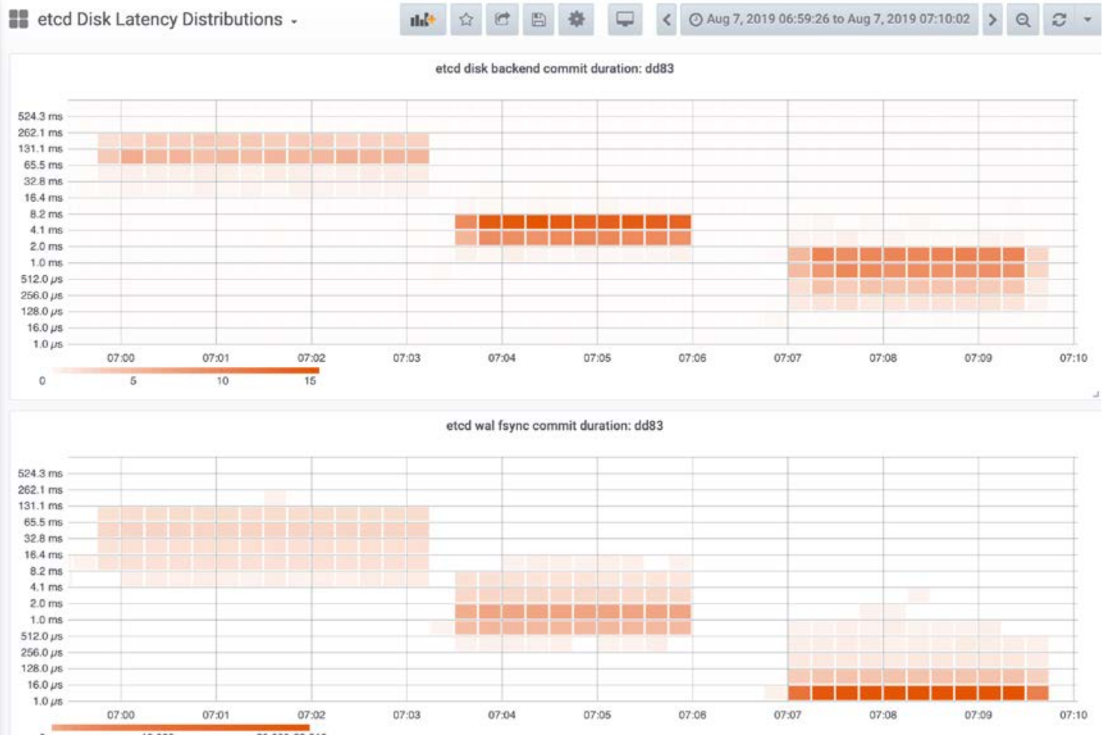
- Many nodes + fast SSD storage =
  - Many events to watch
  - When things break, many changes to process

- Can we improve etcd write workload scalability by adding faster storage?

etcd Benchmark



Transaction Latency Analysis



etcd Disk Latency Distributions

Aug 7, 2019 06:59:26 to Aug 7, 2019 07:10:02

etcd disk backend commit duration: dd83

etcd wal fsync commit duration: dd83

Thank You

Richard.Elling@VikingEnterprise.com