Acto: Automatic End-to-End Testing for Kubernetes Operators

https://github.com/xlab-uiuc/acto

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

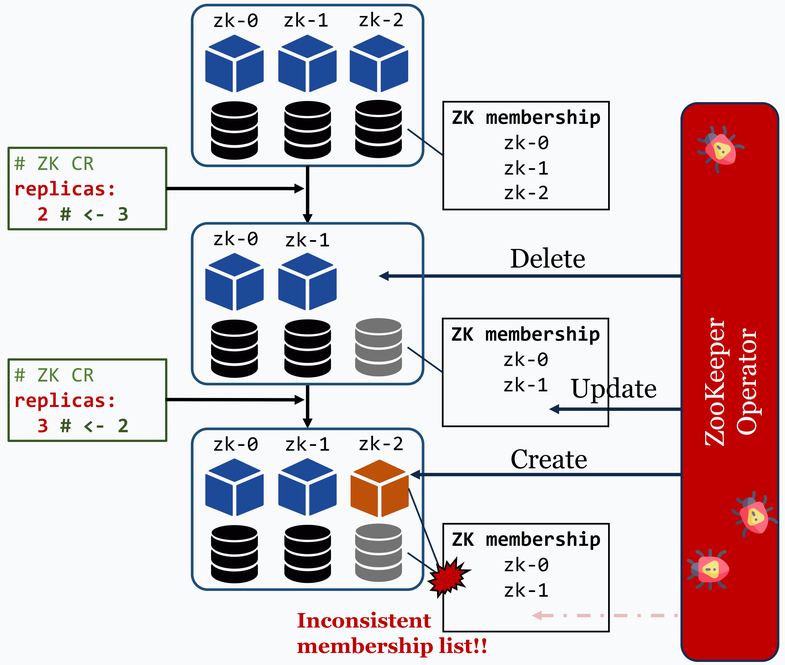
IBM Research

KubeCon | CloudNativeCon

Overview

- Acto is a **push-button E2E testing** tool that finds critical bugs in Kubernetes operators
- Acto checks three correctness properties
  - *Always* reconciling the system to the desired state
  - *Always* recovering the system from bad states
  - *Always* being resilient to operation errors
- Acto has been applied to 39 operators and found hundreds of bugs in them.

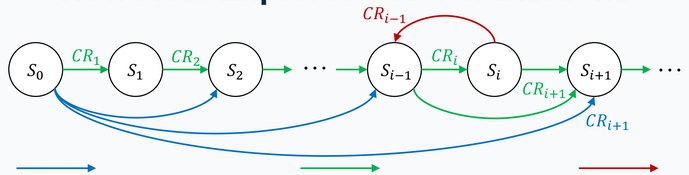
An operator bug example



Pravega's ZooKeeper operator fails to manage the ZooKeeper membership list correctly, causing the scaling operations to fail

Key Ideas[1]

How Acto Explores State Transitions

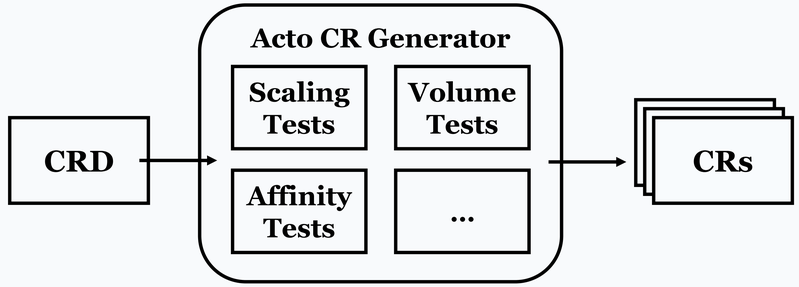


Single operation if the operator follows level-triggering property

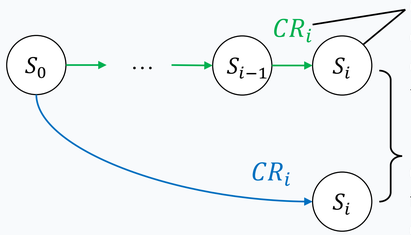
Operation sequence testing operations from existing states

Error-state recovery if the operator can restore the system from error states

Automatic Custom Resource Generation



Automatic Oracles



Consistency Oracle checks if the operator view is consistent with the Kubernetes view

Differential Oracle checks if the system is reconciled to the **same** end state from different start states

[1] "Acto: Automatic End-to-End Testing for Operation Correctness of Cloud System Management", SOSP'23

Main Results

- Acto is applied to 39 open-source operators
- Acto tested each operator with a nightly run

cassandra, redis, CockroachDB, PyTorch, RabbitMQ, PERCONA XtraDB Cluster, APACHE ZooKeeper, kafka, mongoDB, ClickHouse, Flink, Solr

Acto has found hundreds of bugs in these popular Kubernetes operators so far!

From Research to Open Source

- We'd like to turn Acto into an open-source project
- We are here to learn from the community
- Do you want to use Acto to test your operator?

Give us a star as encouragement.
Your feedback means a lot!