Jepsen

Distributed Systems Safety Research

Andi Skrgat

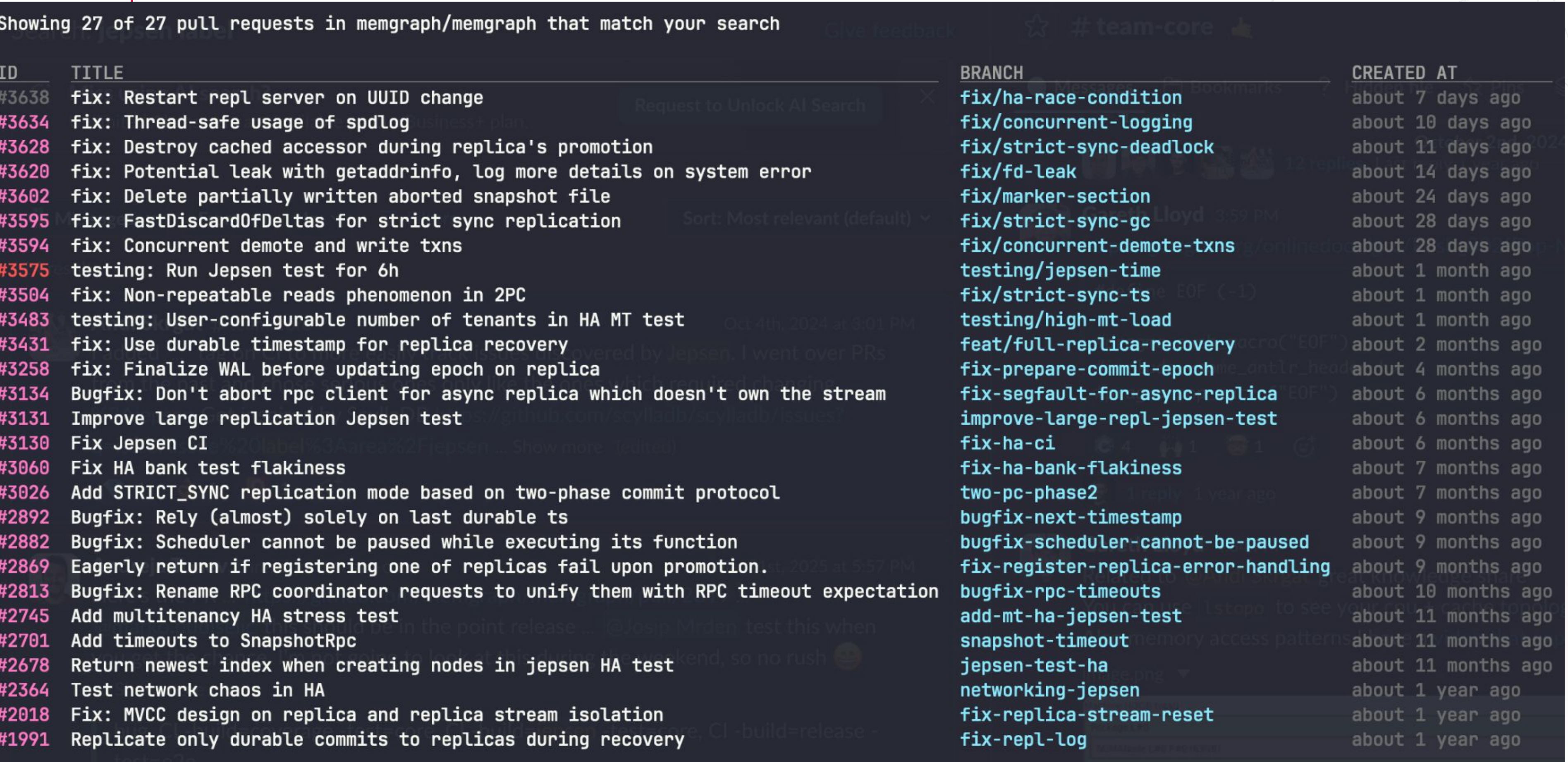
Content of this deck

1. Motivation for this talk
2. What is Jepsen?
3. Tests in Memgraph
4. Test in detail
5. Jepsen & CI
6. How to debug Jepsen?

01

Motivation

```
gh pr list --label “Jepsen” --state all
```

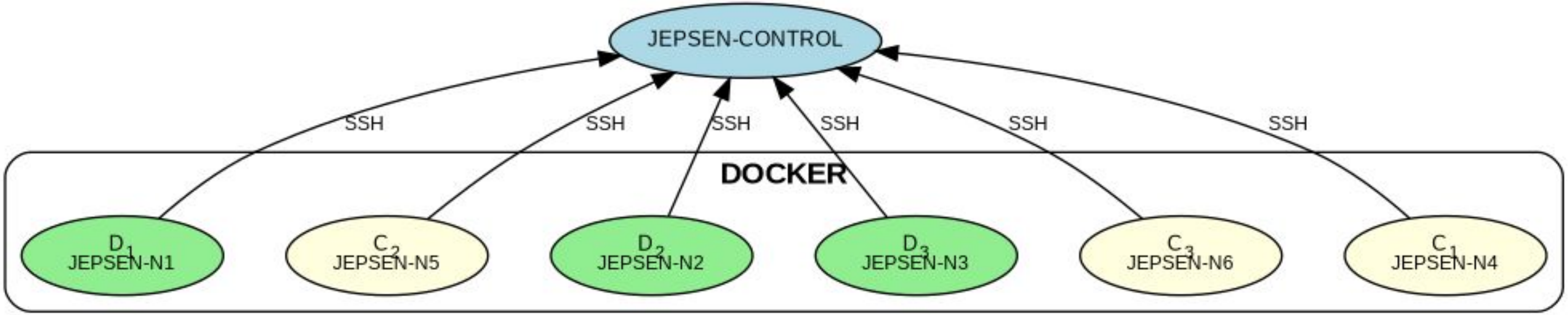


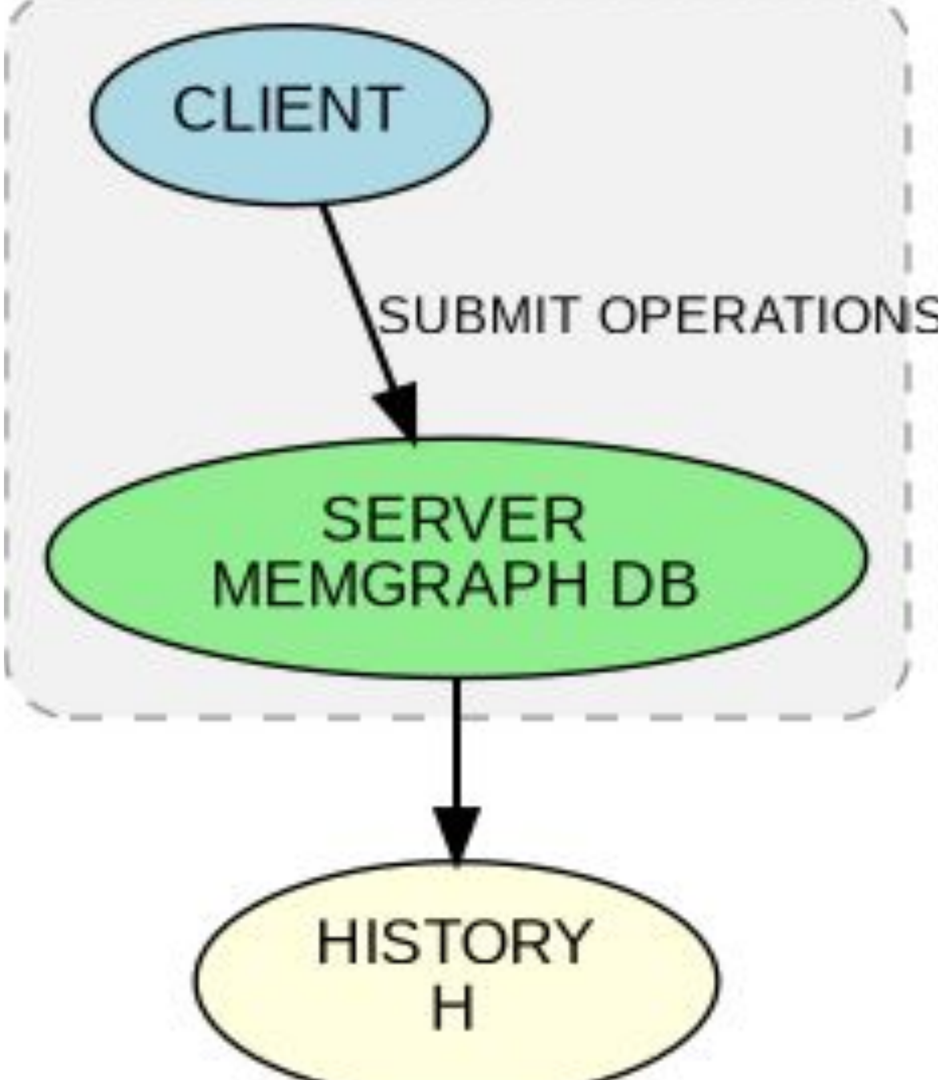
```
Showing 27 of 27 pull requests in memgraph/memgraph that match your search

ID     TITLE                                                                               BRANCH                               CREATED AT
#3638  fix: Restart repl server on UUID change                                             fix/ha-race-condition                about 7 days ago
#3634  fix: Thread-safe usage of spdlog                                                    fix/concurrent-logging               about 10 days ago
#3628  fix: Destroy cached accessor during replica's promotion                             fix/strict-sync-deadlock             about 11 days ago
#3620  fix: Potential leak with getaddrinfo, log more details on system error              fix/fd-leak                          about 14 days ago
#3602  fix: Delete partially written aborted snapshot file                                 fix/marker-section                   about 24 days ago
#3595  fix: FastDiscardOfDeltas for strict sync replication                                fix/strict-sync-gc                   about 28 days ago
#3594  fix: Concurrent demote and write txns                                               fix/concurrent-demote-txns           about 28 days ago
#3575  testing: Run Jepsen test for 6h                                                     testing/jepsen-time                  about 1 month ago
#3504  fix: Non-repeatable reads phenomenon in 2PC                                         fix/strict-sync-ts                   about 1 month ago
#3483  testing: User-configurable number of tenants in HA MT test                          testing/high-mt-load                 about 1 month ago
#3431  fix: Use durable timestamp for replica recovery                                     feat/full-replica-recovery           about 2 months ago
#3258  fix: Finalize WAL before updating epoch on replica                                  fix-prepare-commit-epoch             about 4 months ago
#3134  Bugfix: Don't abort rpc client for async replica which doesn't own the stream       fix-segfault-for-async-replica       about 6 months ago
#3131  Improve large replication Jepsen test                                               improve-large-repl-jepsen-test       about 6 months ago
#3130  Fix Jepsen CI                                                                       fix-ha-ci                            about 6 months ago
#3060  Fix HA bank test flakiness                                                          fix-ha-bank-flakiness                about 7 months ago
#3026  Add STRICT_SYNC replication mode based on two-phase commit protocol                 two-pc-phase2                        about 7 months ago
#2892  Bugfix: Rely (almost) solely on last durable ts                                     bugfix-next-timestamp                about 9 months ago
#2882  Bugfix: Scheduler cannot be paused while executing its function                     bugfix-scheduler-cannot-be-paused    about 9 months ago
#2869  Eagerly return if registering one of replicas fail upon promotion.                  fix-register-replica-error-handling  about 9 months ago
#2813  Bugfix: Rename RPC coordinator requests to unify them with RPC timeout expectation  bugfix-rpc-timeouts                  about 10 months ago
#2745  Add multitenancy HA stress test                                                     add-mt-ha-jepsen-test                about 11 months ago
#2701  Add timeouts to SnapshotRpc                                                         snapshot-timeout                     about 11 months ago
#2678  Return newest index when creating nodes in jepsen HA test                           jepsen-test-ha                       about 11 months ago
#2364  Test network chaos in HA                                                            networking-jepsen                    about 1 year ago
#2018  Fix: MVCC design on replica and replica stream isolation                            fix-replica-stream-reset             about 1 year ago
#1991  Replicate only durable commits to replicas during recovery                          fix-repl-log                         about 1 year ago
```

02

What is Jepsen?

- A testing tool for distributed systems
- A collection of libraries
- Written in Clojure
- Working on binaries, not on source code ⇒ bugs in production
- Works with Docker containers, MG binary compiled for Debian 12
- Consistency checker
- **Injecting faults**
- Cannot prove correctness, only failures


JEPSEN-CONTROL
SSH
SSH
SSH
SSH
SSH
SSH
DOCKER
D1
JEPSEN-N1
C2
JEPSEN-N5
D2
JEPSEN-N2
D3
JEPSEN-N3
C3
JEPSEN-N6
C1
JEPSEN-N4

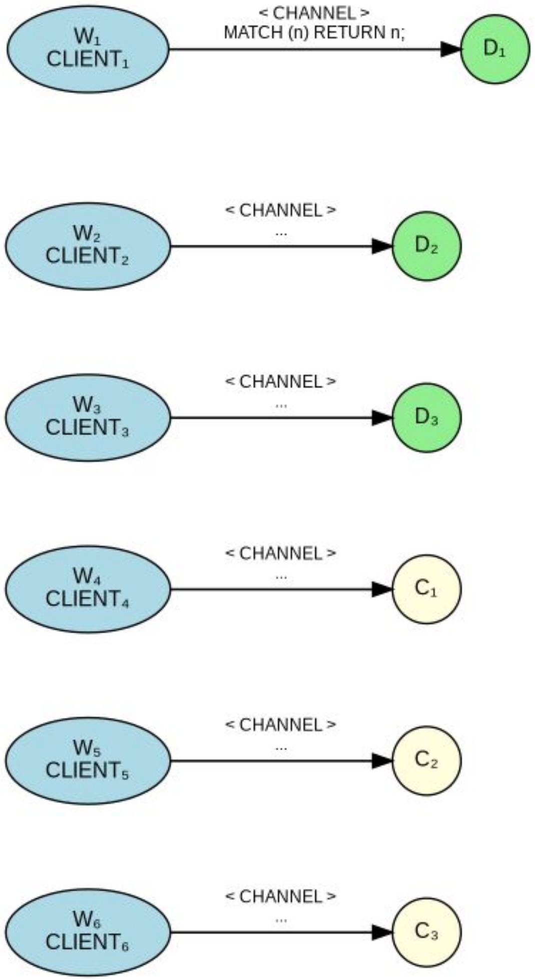

CLIENT
SUBMIT OPERATIONS
SERVER
MEMGRAPH DB
HISTORY
H

Legend

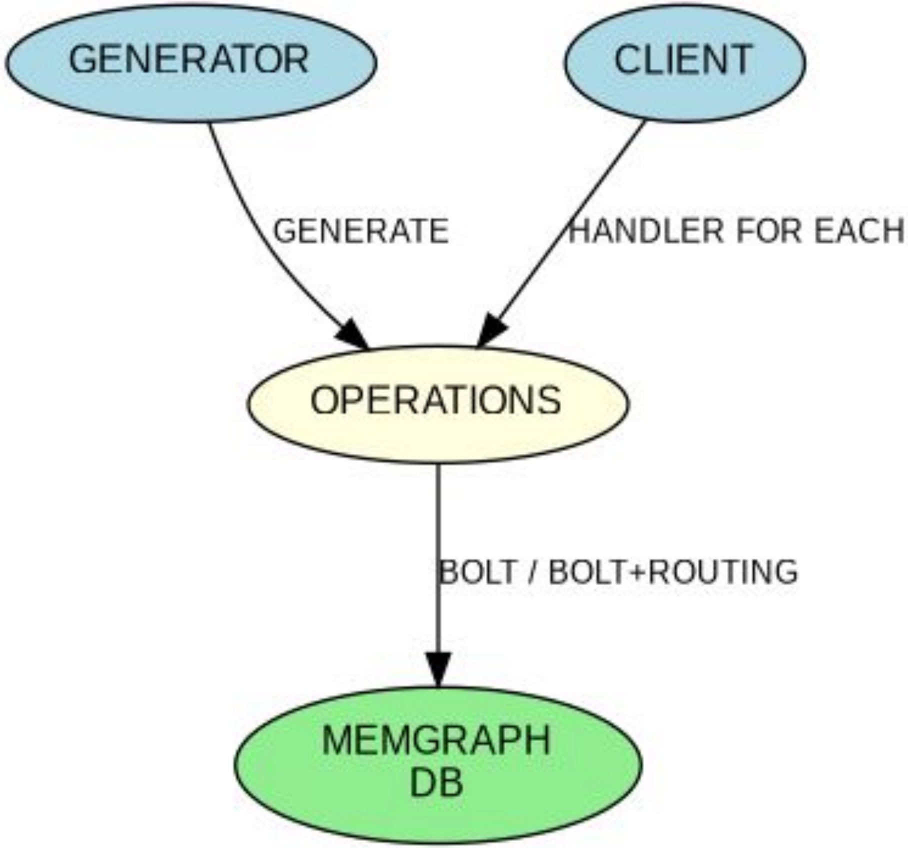
W = WORKER

D = DATA INSTANCE = MEMGRAPH

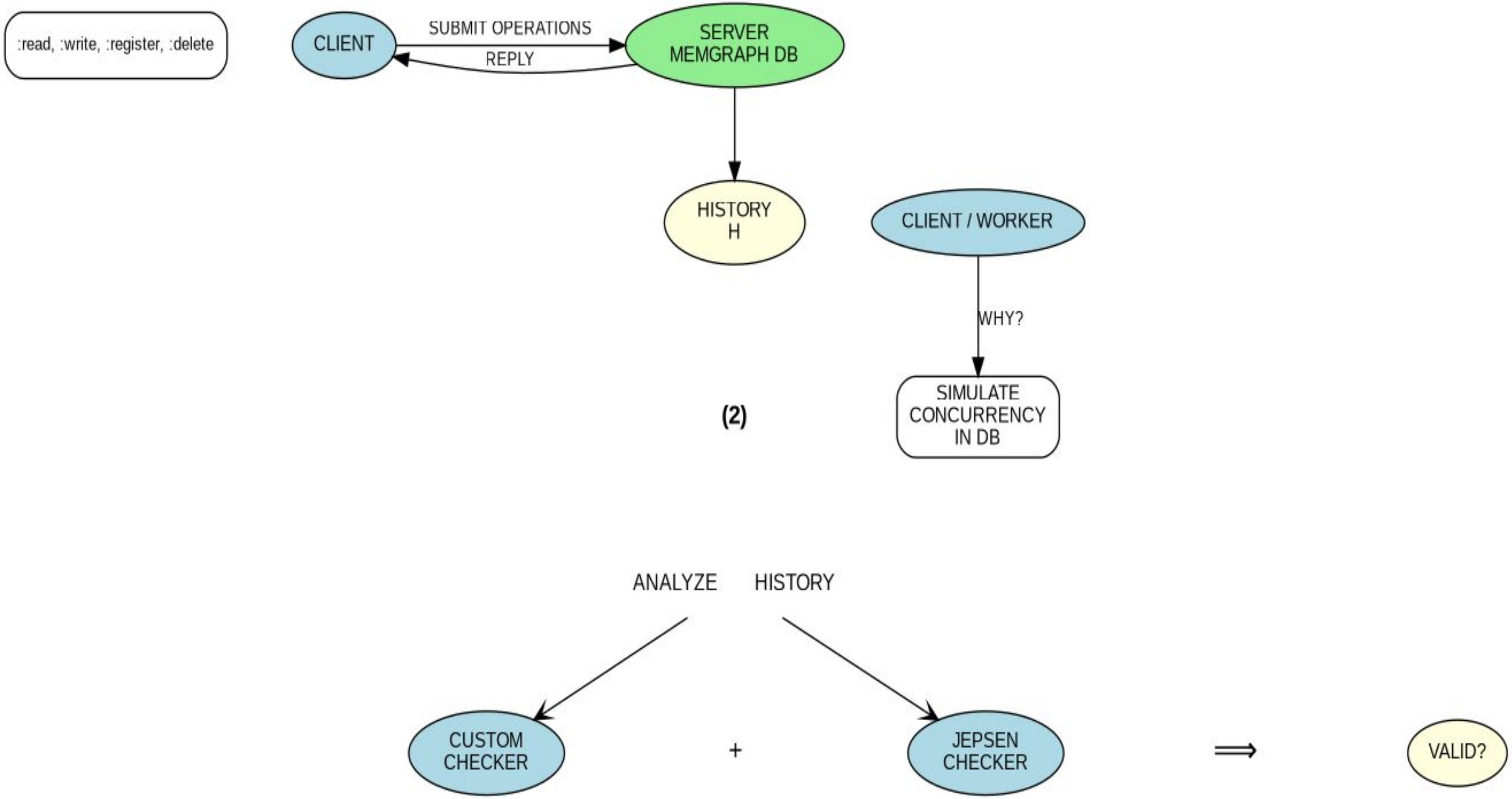
C = COORDINATOR = MEMGRAPH



(1) SUBMITTING OPERATIONS



(1)



:read, :write, :register, :delete

CLIENT

SUBMIT OPERATIONS

REPLY

SERVER
MEMGRAPH DB

HISTORY
H

CLIENT / WORKER

WHY?

(2)

SIMULATE
CONCURRENCY
IN DB

ANALYZE HISTORY

CUSTOM
CHECKER

+

JEPSEN
CHECKER

⟹

VALID?

Summary

- Num of workers = num of instances
- Clients connect to instances
- Generator generates sequences of operations
- Client has a handler for each operation
- Cypher query is sent to Memgraph DB using Bolt/Bolt+routing connection
- There is a handler for each of the requested operation
- Result is written to history
- History analyzed at the end

03

Tests in MG

Replication tests

- tests/jepsen/src/memgraph/replication/bank.clj
- tests/jepsen/src/memgraph/replication/large.clj
- Run on diff
- Bank test tests that our transactional guarantees hold
  - Add money to account 1
  - Remove money from account 2
- Doesn’t use HA
- Large test adds nodes and checks all the time whether constant node creation gets replicated

HA tests

- tests/jepsen/src/memgraph/high_availability/bank/test.clj
- tests/jepsen/src/memgraph/high_availability/create/test.clj
- HA bank test tests that our transactional guarantees hold
  - Add money to account 1
  - Remove money from account 2
  - Run on diff

HA create test

- tests/jepsen/src/memgraph/high_availability/create/test.clj
- Runs every night
  - Day 1: tests sync replication
  - Day 2: tests sync + async replication
  - Day 3: tests strict sync replication
  - Day 4: tests strict sync + async replication
- You can run it manually with Jepsen HA Stress Tests workflow on CI

MT test

- tests/jepsen/src/memgraph/mtenancy/test.clj
- Runs every night
  - Day 1: tests sync replication
  - Day 2: tests sync + async replication
  - Day 3: tests strict sync replication
  - Day 4: tests strict sync + async replication

- You can run it manually with Jepsen HA Stress Tests workflow on CI
- specifically tests TTL operations (was needed for some customer)

Types of failures

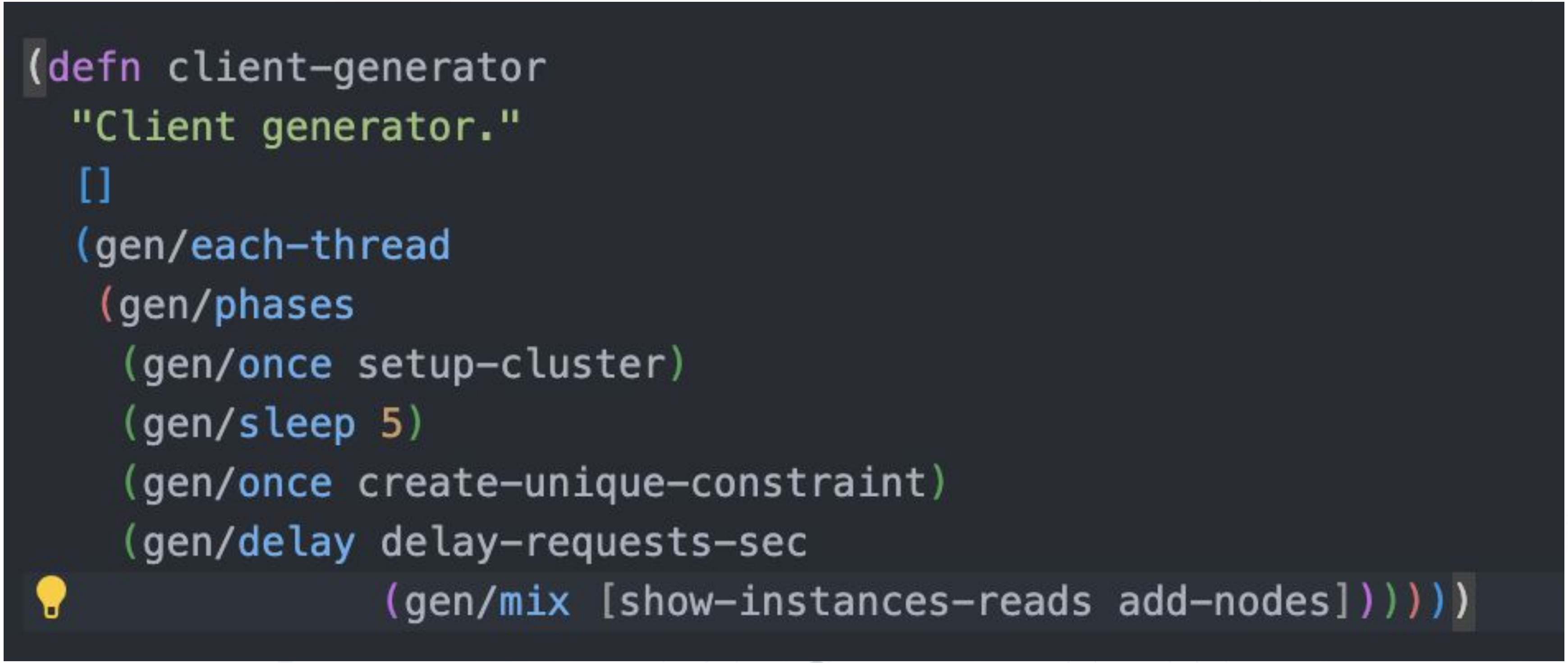
- Kill a node (either the current main or the current leader)
- Network partition random halves
- Isolate a single node
- Ring network communication
- Disruption of TCP packets, e.g
  - Delay 35% of TCP packets, drop 20% of them ...

- TODO: Disk corruption

04

Test in details

Create a generator for operations



```clojure
(defn client-generator
  "Client generator."
  []
  (gen/each-thread
   (gen/phases
    (gen/once setup-cluster)
    (gen/sleep 5)
    (gen/once create-unique-constraint)
    (gen/delay delay-requests-sec
               (gen/mix [show-instances-reads add-nodes])))))
```

INITIALIZATION

(1) CHOOSE 1ST LEADER

(2) CHOOSE 1ST MAIN

(3) SETUP CLUSTER

Unique constraints

- Unique constraints are used because there could be multiple clients connecting to the coordinator and writing the same batch of nodes (with the same IDs) ⇒ we want to test at the end that we have a sorted monotonically increasing set of nodes on each instance

Writes

- Create a batch of nodes with random size using bolt+routing
  - Random because we want to test what happens with different “replication speed” and “replication size”

SHOW INSTANCES

- Collect the cluster state throughout the test execution

When is the test valid?

- Client managed to connect at the end of the test and collect all nodes
- All nodes' ids are monotonically increasing (a range) and same on all nodes
- There are no duplicated nodes
- Hamming and Jaccard consistency measures are taken
- Setup cluster didn't fail
- SHOW INSTANCES never failed in any way except if we unable to connect to the coordinator
- SHOW INSTANCES always returns the same number of instances (Raft stability)

And

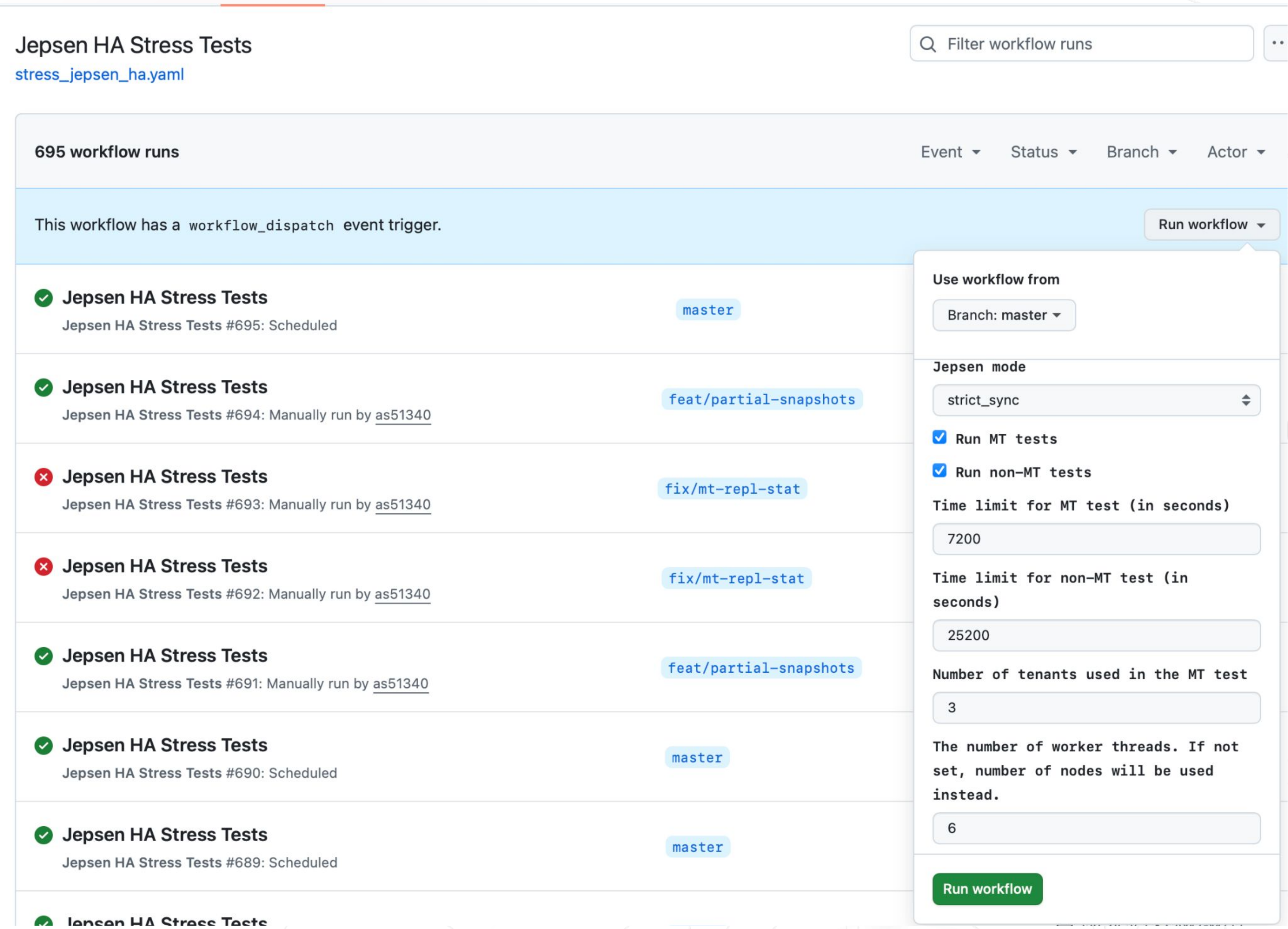
- There is never more than one main in the Raft state as observed through SHOW INSTANCES
- There is never more or less than 3 coordinators observed through SHOW INSTANCES
- Creating nodes never fails except
  - Txn timeout occurred
  - No write server available
  - Sync replica is down
  - Strict sync replica is down
  - Instance is not main anymore

And

- Creating nodes never fails except
    - Main is unwriteable
    - Unique constraint was violated
    - Cannot get shared access to the storage

05

Usage



Jepsen HA Stress Tests

stress_jepsen_ha.yaml

695 workflow runs

This workflow has a `workflow_dispatch` event trigger.

✓ Jepsen HA Stress Tests — master
Jepsen HA Stress Tests #695: Scheduled

✓ Jepsen HA Stress Tests — feat/partial-snapshots
Jepsen HA Stress Tests #694: Manually run by as51340

✗ Jepsen HA Stress Tests — fix/mt-repl-stat
Jepsen HA Stress Tests #693: Manually run by as51340

✗ Jepsen HA Stress Tests — fix/mt-repl-stat
Jepsen HA Stress Tests #692: Manually run by as51340

✓ Jepsen HA Stress Tests — feat/partial-snapshots
Jepsen HA Stress Tests #691: Manually run by as51340

✓ Jepsen HA Stress Tests — master
Jepsen HA Stress Tests #690: Scheduled

✓ Jepsen HA Stress Tests — master
Jepsen HA Stress Tests #689: Scheduled

Jepsen HA Stress Tests

Use workflow from

Branch: master

Jepsen mode: strict_sync

☑ Run MT tests

☑ Run non-MT tests

Time limit for MT test (in seconds): 7200

Time limit for non-MT test (in seconds): 25200

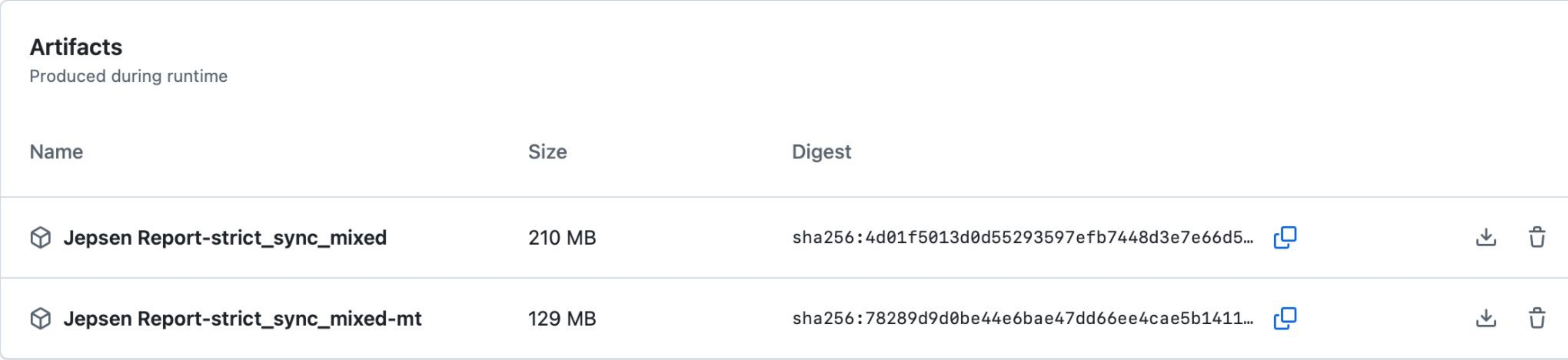
Number of tenants used in the MT test: 3

The number of worker threads. If not set, number of nodes will be used instead.: 6

Run workflow

06
Debugging

After the test has finished



Artifacts

Produced during runtime

Name	Size	Digest
Jepsen Report-strict_sync_mixed	210 MB	sha256:4d01f5013d0d55293597efb7448d3e7e66d5…
Jepsen Report-strict_sync_mixed-mt	129 MB	sha256:78289d9d0be44e6bae47dd66ee4cae5b1411…

During the test run

- ssh to the instance on which Jepsen is run
- docker exec -it jepsen-n1 bash
- cat /opt/memgraph/memgraph.log

- gdb -p PID # if you suspect there is some kind of a deadlock

- Disk space could be a problem, you will see it on GitHub if that was the case in running logs

Thank you for your time!

MEM GRAPH

www.memgraph.com