Weekly Research Progress Report

Student: Sungho Hong　　　　　**Date:**8/10/2020　　　　　**# of hrs worked this week:** 30

Problems discussed in last week's report

1. **Test Results using Jepsen I/O**
    a. Results from Redis
    b. Replicate the test cases of Redis
    c. Apply the test case for etcd

2. **Integration of Jepsen I/O with RamCloud**
    a. Roadmap of integrating RamCloud client

Test Results from Jepsen I/O

Results from Redis

Workload
-   Jepsen/elle generates transactions made up of reads and appends operations
-   **Configurable Arguments**
    -   **key-dist**
        -   controls probability distribution for keys used by transactions
        -   uniform means every key has an equal probability of appearing
        -   exponential means that key i in the current key pool is k^i times more likely than the first key to be chosen.
    -   **key-count**
        -   number of distinct keys
    -   **txn-length (min - max)**
        -   number of operations per txn
    -   **writes-per-key (min - max)**
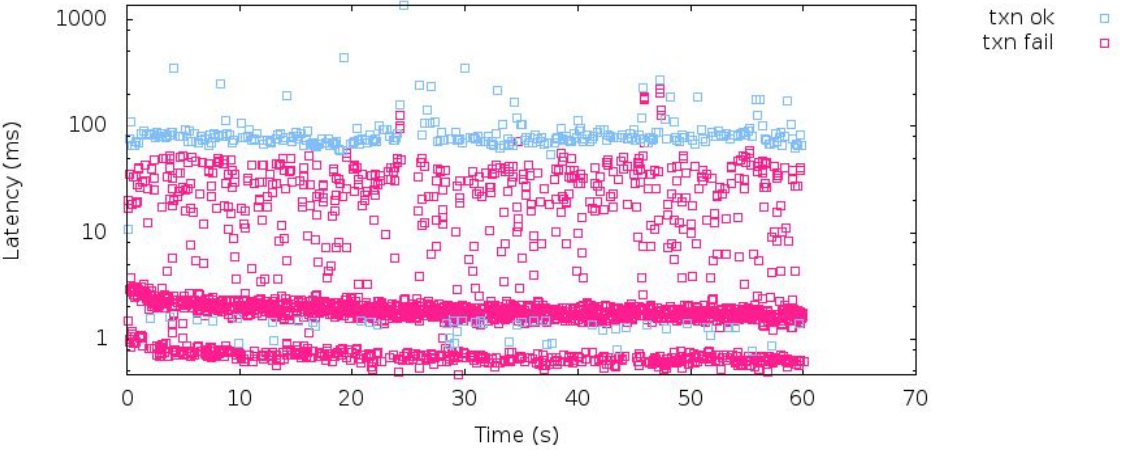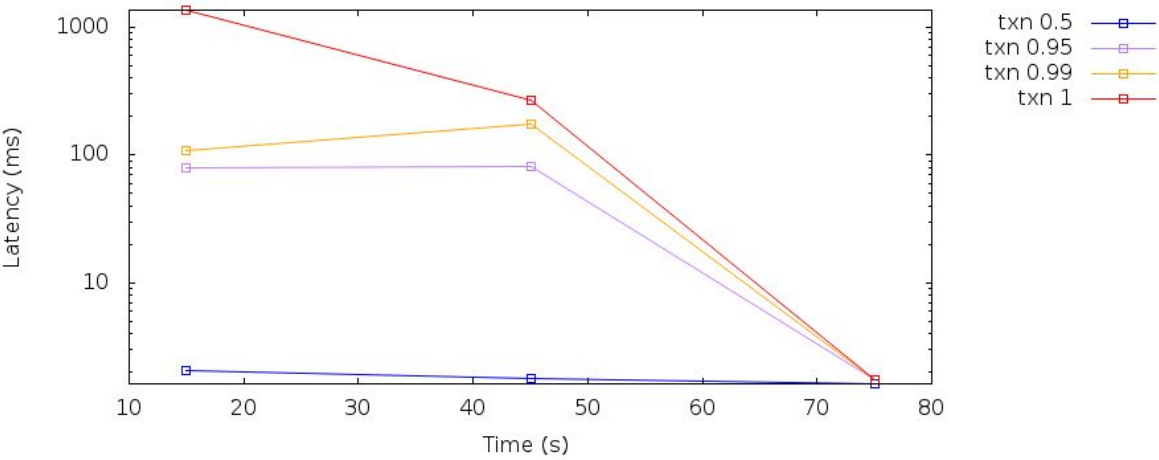   -   number of operations per key

Nemesis
- **Kill**
    - Kill and restore db processes
- **Join, Leave**
    - Attach or Detach a node from the db
- **Pause**
    - Pause db processes
- **Partition**
    - Kill and heal network partitions
- **Island (Customized for Redis)**
    - Uses partition
    - A generator which picks a primary, isolates it from all other nodes, then issues leave requests to the isolated primary asking every other node to leave the cluster.
- **Mystery (Customized for Redis)**
    - Uses kill and join processes
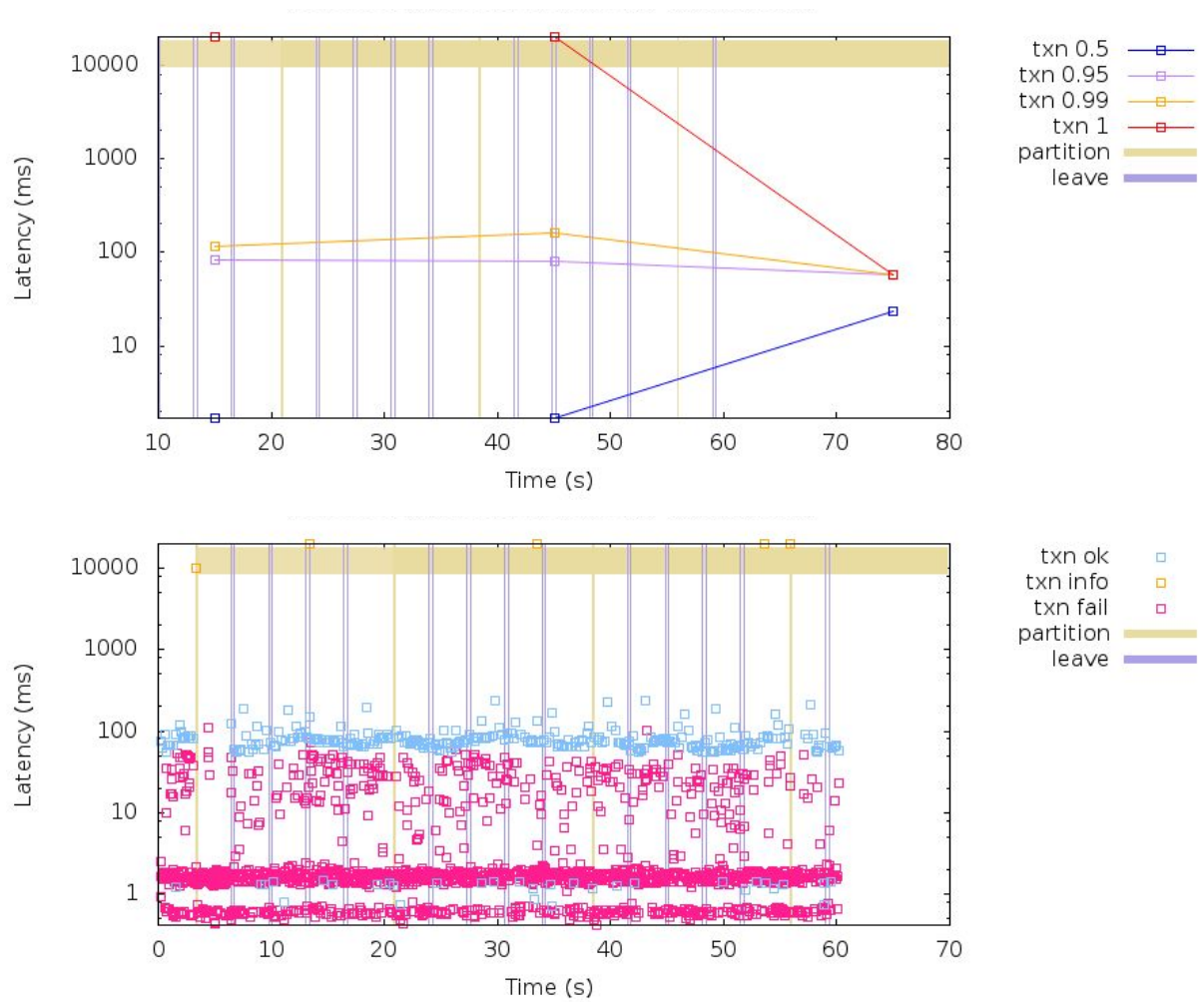   - Hard coded with involving crashes and membership changes

Useful tweaks
- **Test duration time**
    - The duration of a single test
- **Iteration of tests**
    - The number of conducted test
- **Nemesis duration**
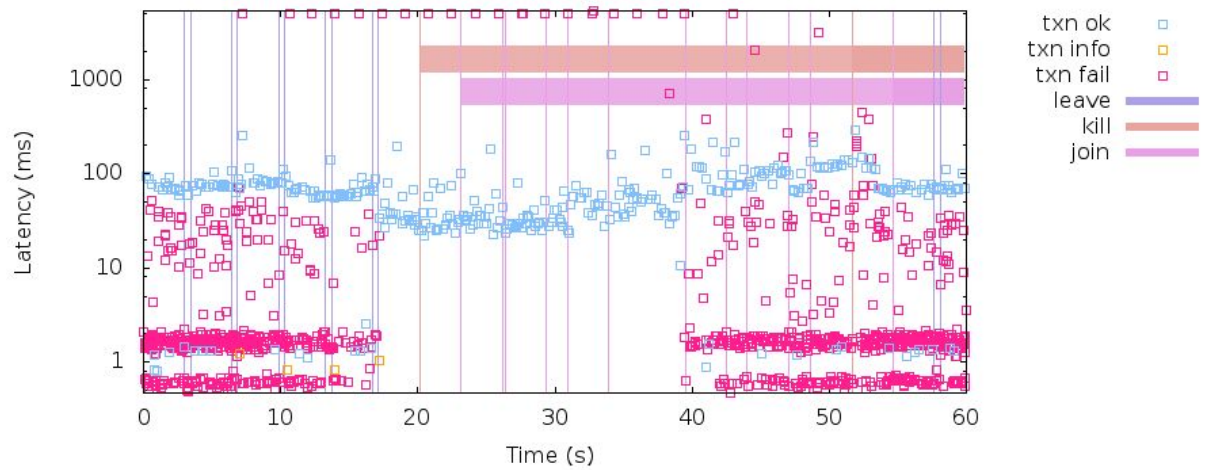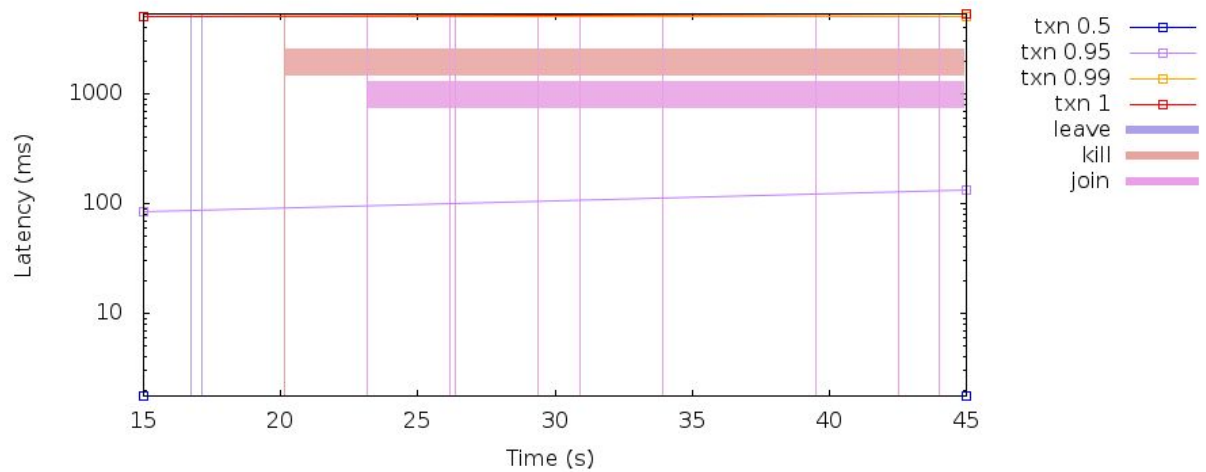   - The frequency of the nemesis

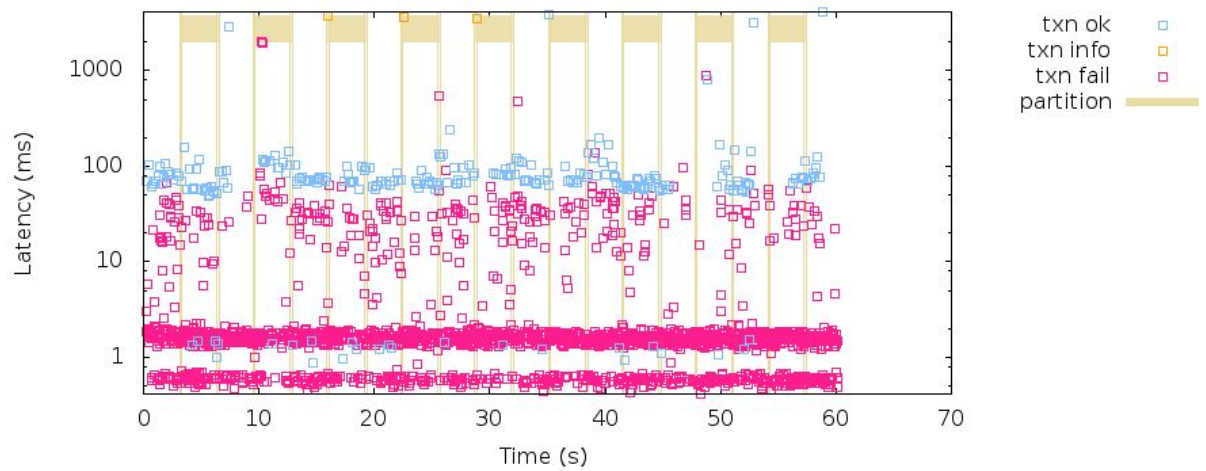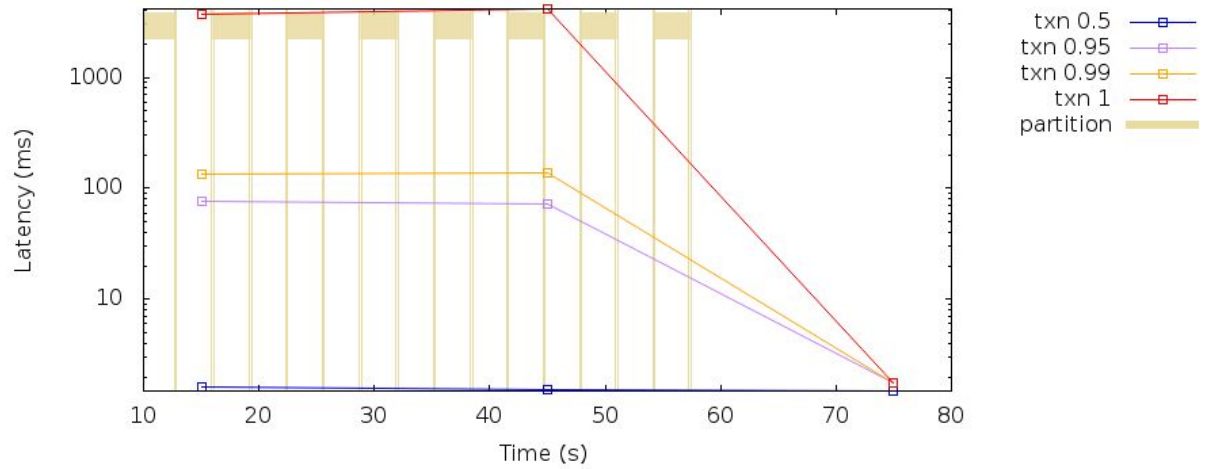**Test case:** Latest version of redis ran with no nemesis. No anomalies are found.

Test case: Latest version of redis ran with island nemesis that is applied only once. No anomalies are found.
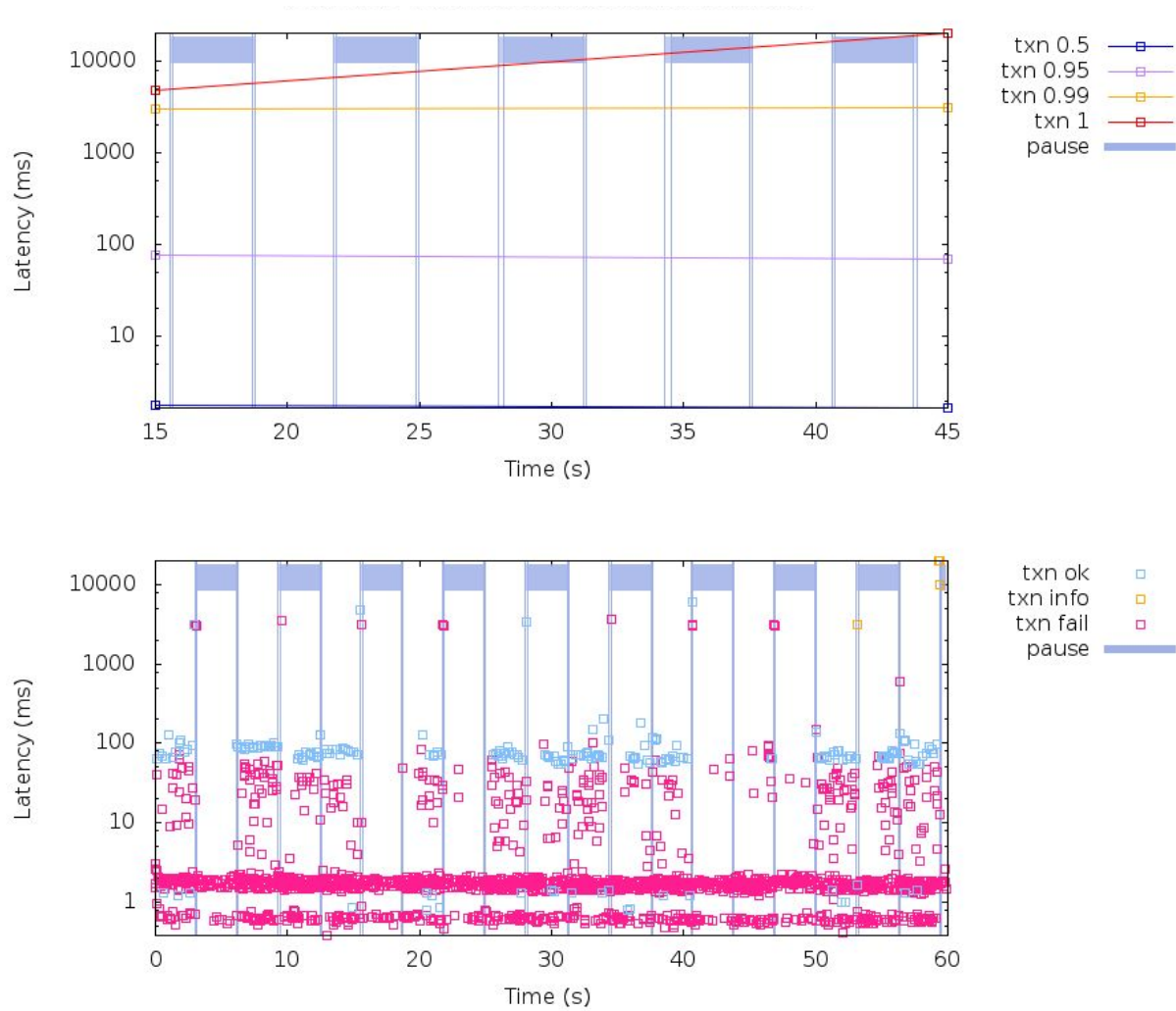
Test case: Latest version of redis ran with customized nemesis (kill, join) that is applied only once. No anomalies are found.
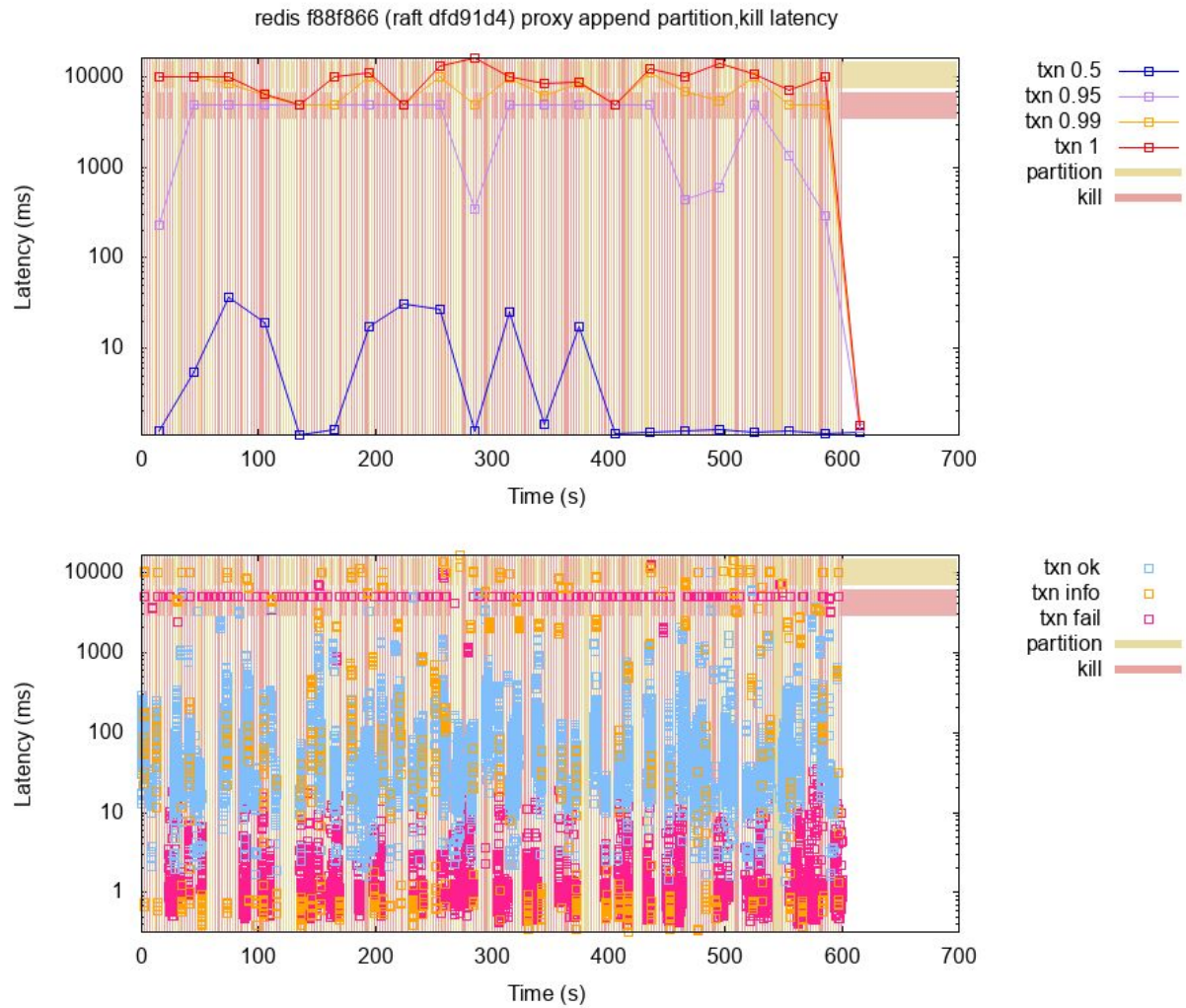
Test case: Latest version of redis ran with partition nemesis that is applied frequently
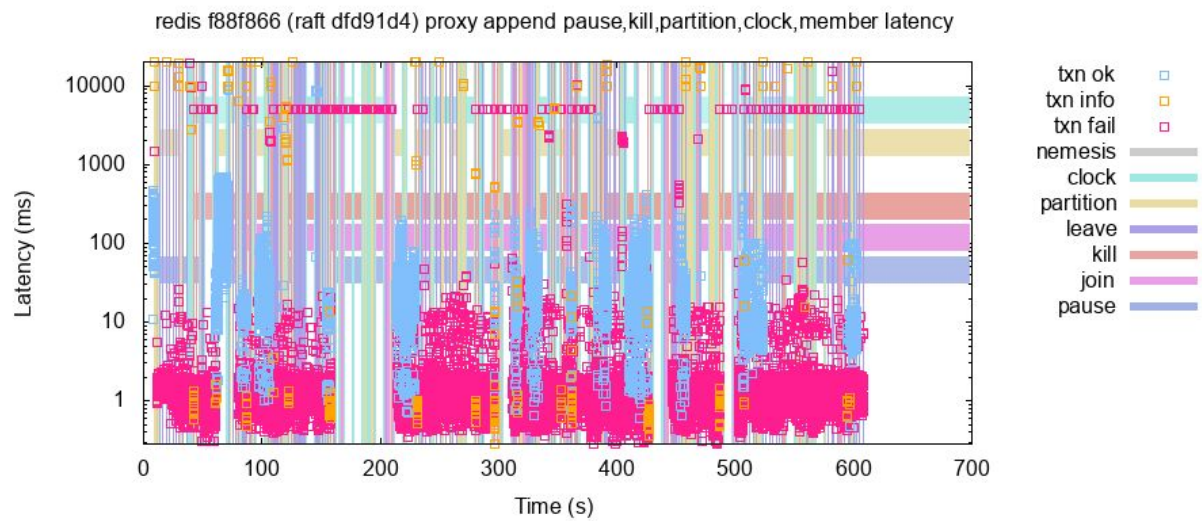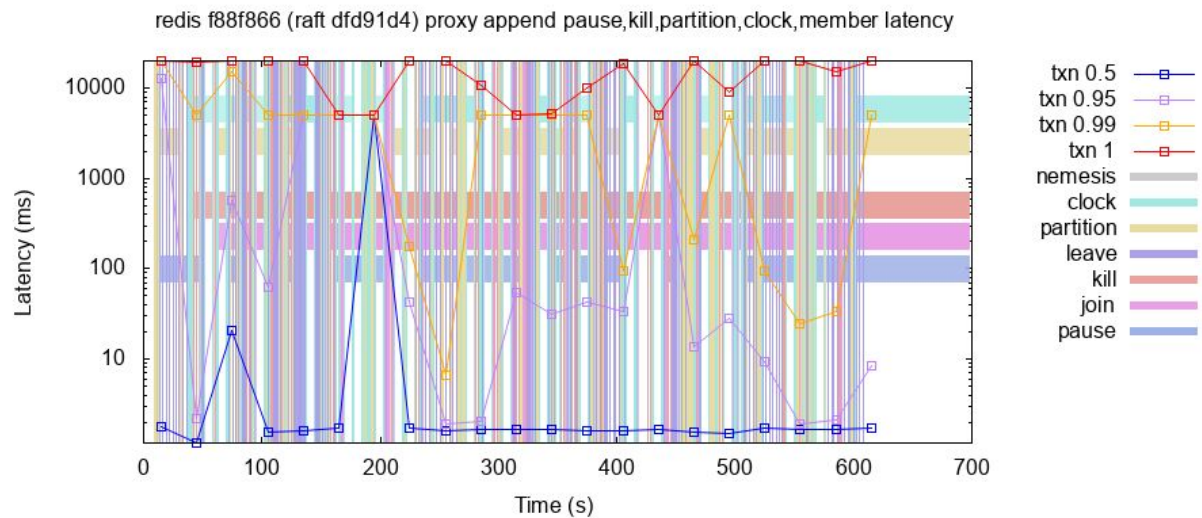No anomalies are found.

Test case: Latest version of redis ran with pause nemesis that is applied frequently
No anomalies are found

Test case: Previous version of redis partitioned and killed frequently (2ms)
G1a (Aborted Read: T2 sees T1's failed write) was detected

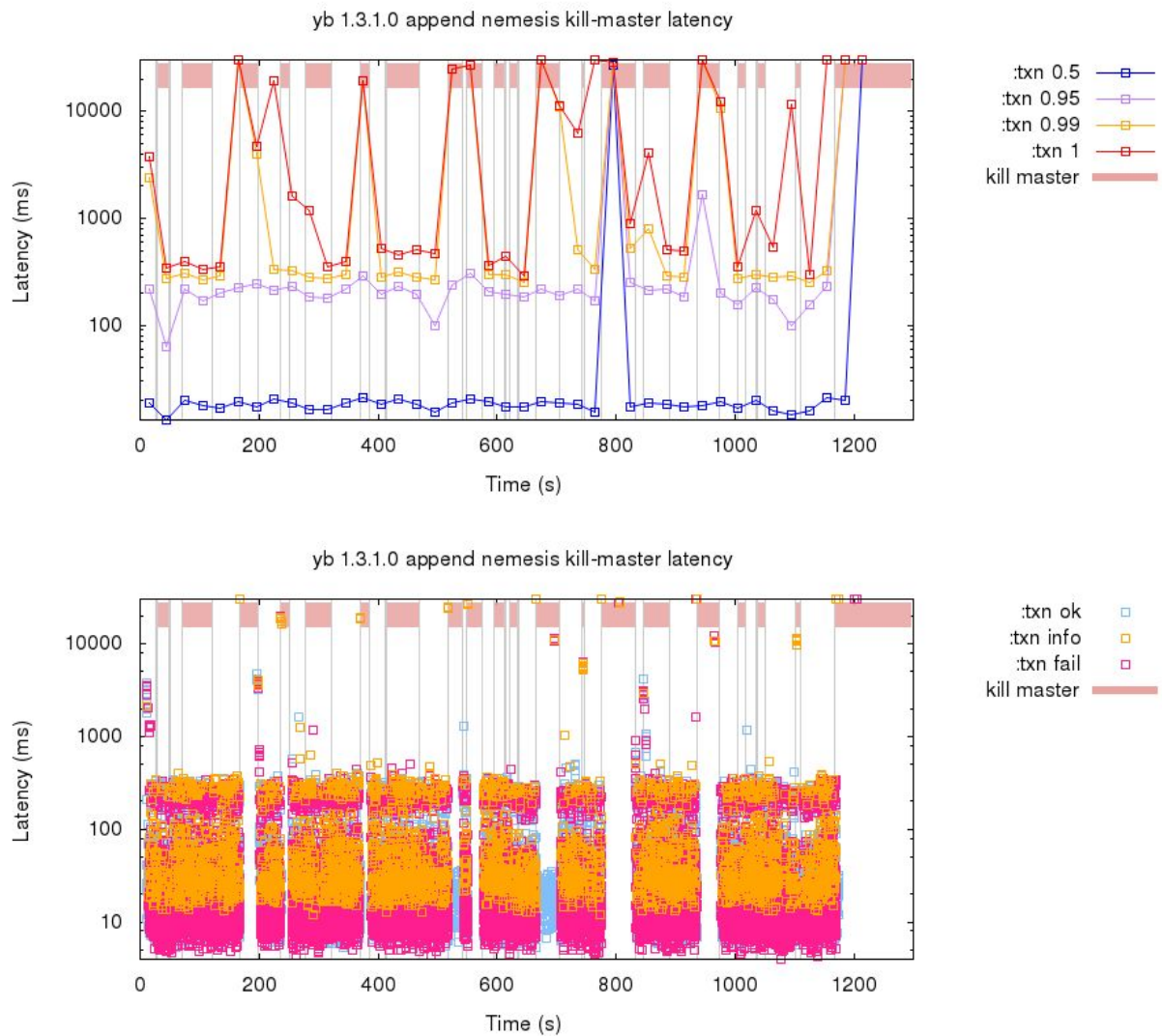redis f88f866 (raft dfd91d4) proxy append partition,kill latency

Test case: Previous version of redis partitioned and killed frequently (2ms)
G1a (Aborted Read) and Incompatible-order (Lost Update) was detected

redis f88f866 (raft dfd91d4) proxy append pause,kill,partition,clock,member latency

redis f88f866 (raft dfd91d4) proxy append pause,kill,partition,clock,member latency

Results from Yugabyte

Test case: Previous version of Yugabyte, master is killed frequently
G2 (Anti dependency) was detected

yb 1.3.1.0 append nemesis kill-master latency

yb 1.3.1.0 append nemesis kill-master latency

Test case: Previous version of Yugabyte, master is paused frequently
G2 (Anti dependency) was detected

yb 1.3.1.0 append nemesis pause-master latency

yb 1.3.1.0 append nemesis pause-master latency