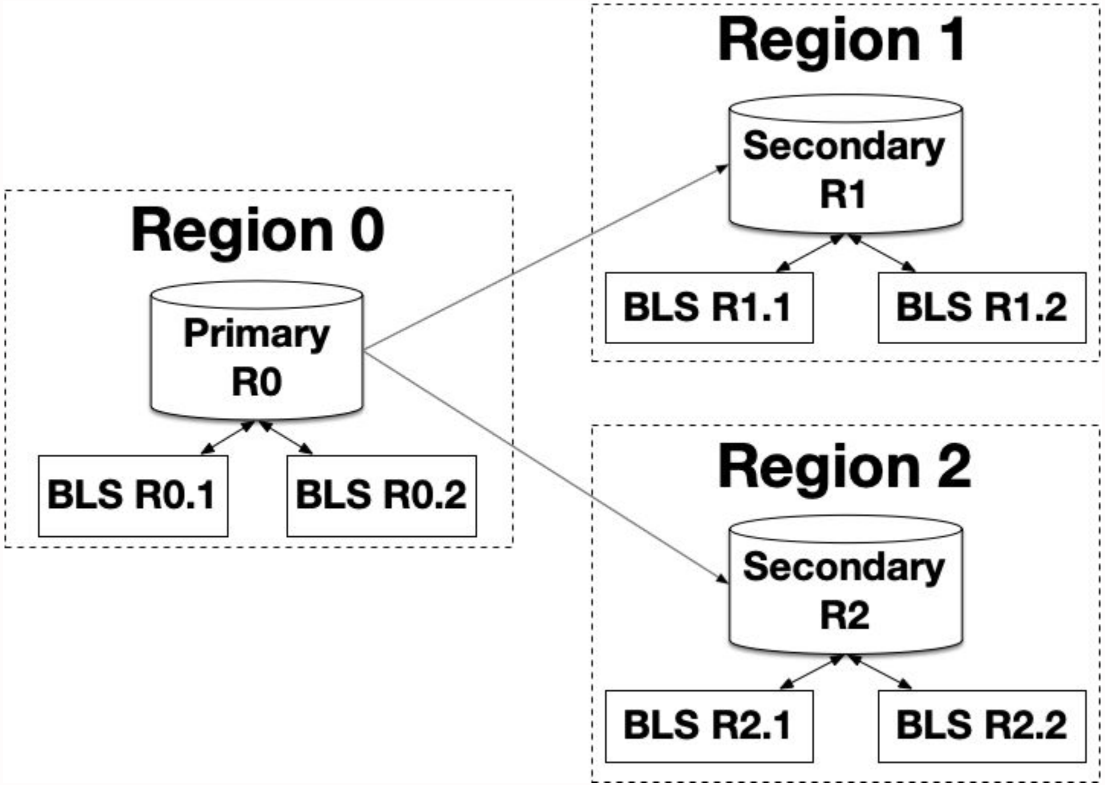
FlexiRaft: Flexible Quorums with Raft

CIDR 2023

Authors

Ritwik Yadav (ritwikyadav@meta.com),
Anirban Rahut (arahut@meta.com)

MySQL Semisynchronous Setup



Problems

- **Code changes were error prone**
    - Logic spread across multiple bespoke automation tools
- **Hard to reason about consistency guarantees and correctness**
    - Crash recovery, leader election and disaster recovery exercises coordinated externally
- **Significant manual effort required during outages**

Why Raft?

- Strong leader semantics
- Designed for understandability
- Complete specification for practical systems
- State space reduction and clearly defined phases
- Only servers with the most recent data can become leaders
- Several well-tested open source implementations

Modifications to Raft + Impact

- **Quorums made configurable**
  - End users pick tradeoffs between latency, throughput and fault tolerance
- **Restricting data commit quorums to regionally local servers**
  - Lower latency
  - Higher throughput
- **Tail latencies independent of number of replicas**
- **Automation tools were simplified**
- **Smarter fault tolerance**

Configurable Modes

- **What is a *group*?** Members of a replica set organized into disjoint sets based on physical proximity

Static Mode

- **Disjunction**
  - ***Data commit quorum***
    - Majority in 2 out of 5 groups: {G1, G2, ..., G5}
    - OR
    - Majority in 2 out of 3 groups: {G6, G7, G8}
  - G1 to G5 could be in the US. G6 to G8 could be in Europe.
  - ***Leader election quorum***
    - Majority in 4 out of 5 groups: {G1, G2, ..., G5}
    - AND
    - Majority in 2 out of 3 groups: {G6, G7, G8}

Configurable Modes

Static Mode

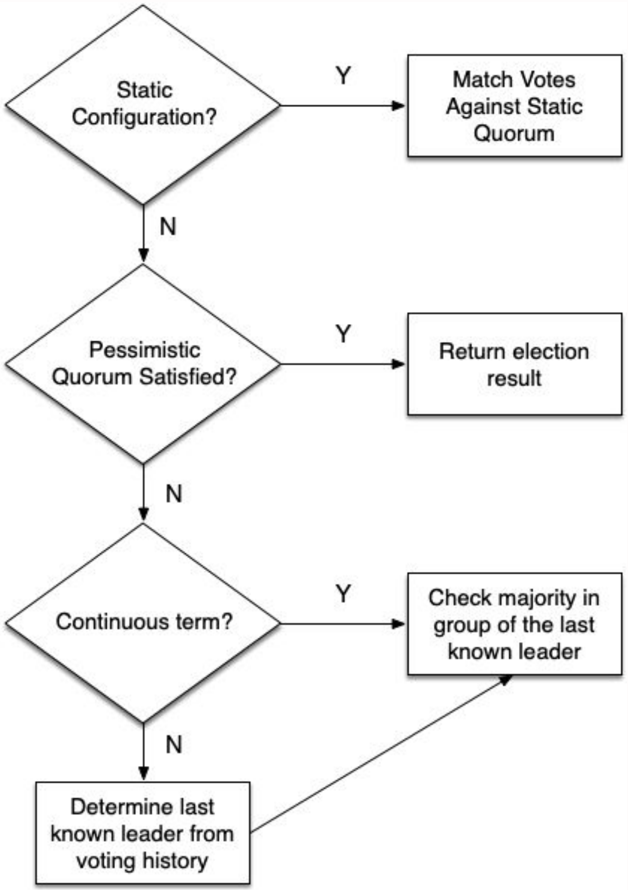
- **Conjunction**
    - ***Data commit quorum***
        - Majority in 2 out of 5 groups: {G1, G2, ..., G5}
        - AND
        - Majority in 2 out of 3 groups: {G6, G7, G8}
    - G1 to G5 could be on the East coast of US. G6 to G8 could be on West coast.
    - ***Leader election quorum***
        - Majority in 4 out of 5 groups: {G1, G2, ..., G5}
        - AND
        - Majority in 2 out of 3 groups: {G6, G7, G8}

Configurable Modes

Dynamic Mode

- Data commit and leader election quorums reconfigured with each election
- *Data commit quorum*
    - Always limited to one group
    - Majority in leader’s group
- *Leader election quorum*
    - Also majority in leader’s group
    - If implemented using static mode, majority in all groups would be needed

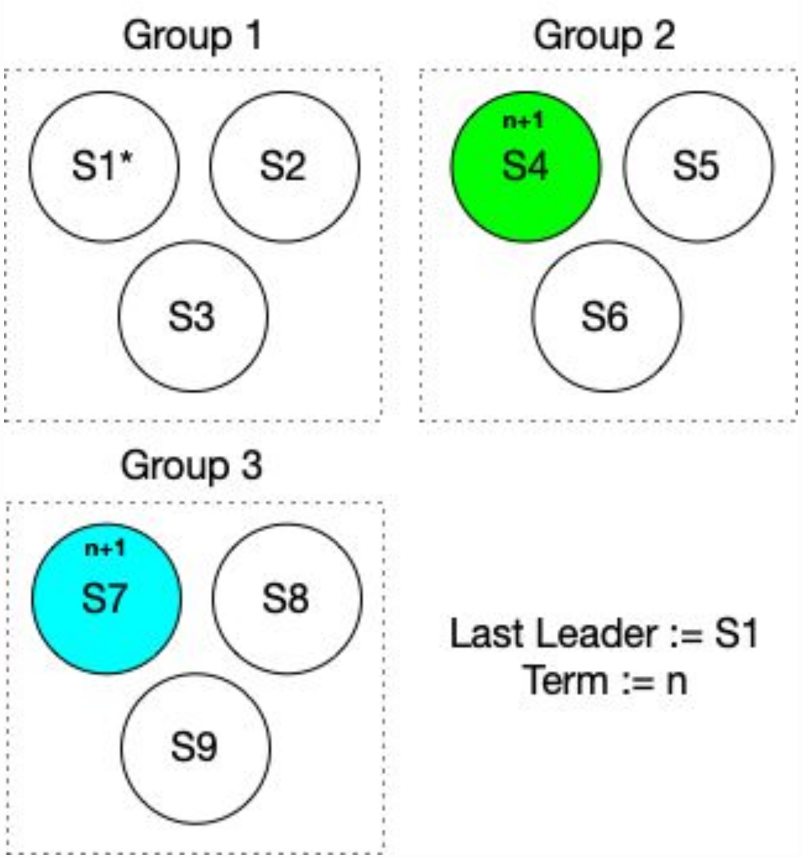
Simplified Algorithm


Static Configuration?
Y
Match Votes Against Static Quorum
N
Pessimistic Quorum Satisfied?
Y
Return election result
N
Continuous term?
Y
Check majority in group of the last known leader
N
Determine last known leader from voting history

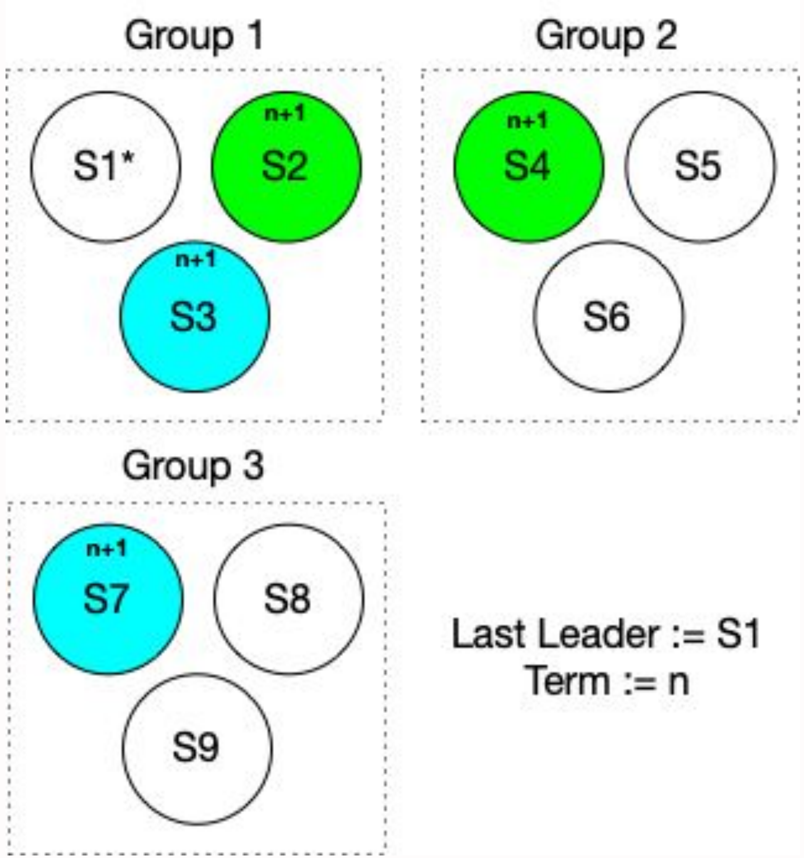
Fault tolerance

- **Static mode**
  - Can survive failure of one group
- **Dynamic Mode**
  - Failure of the leader group will disrupt both the leader election and data commit quorums
  - Certain coordinated failures may cause availability loss even when majority in leader group is functioning

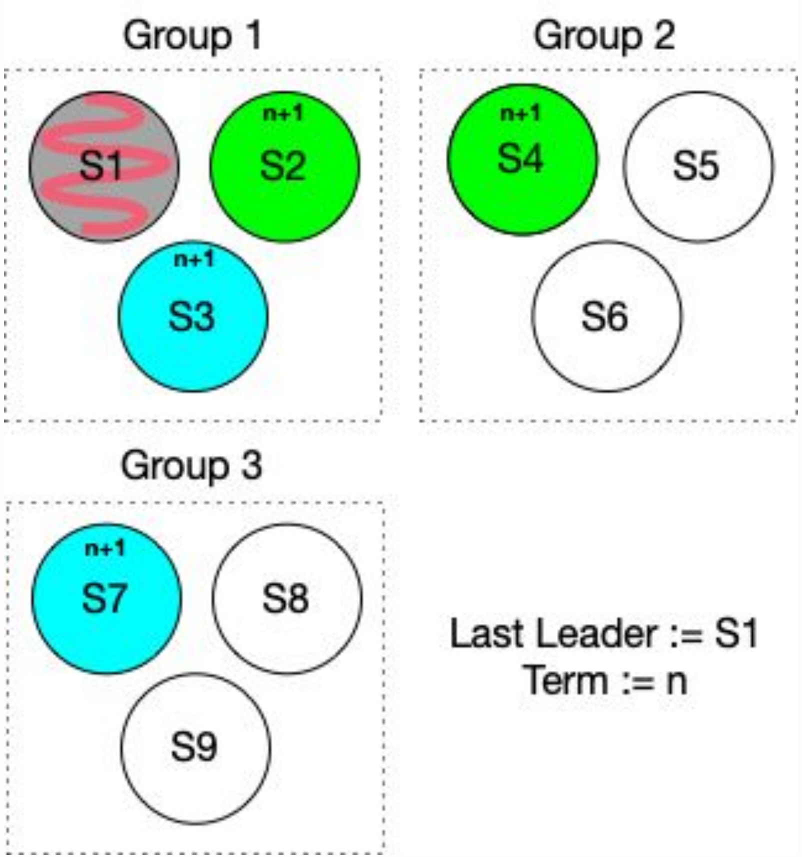
Fault tolerance



Fault tolerance

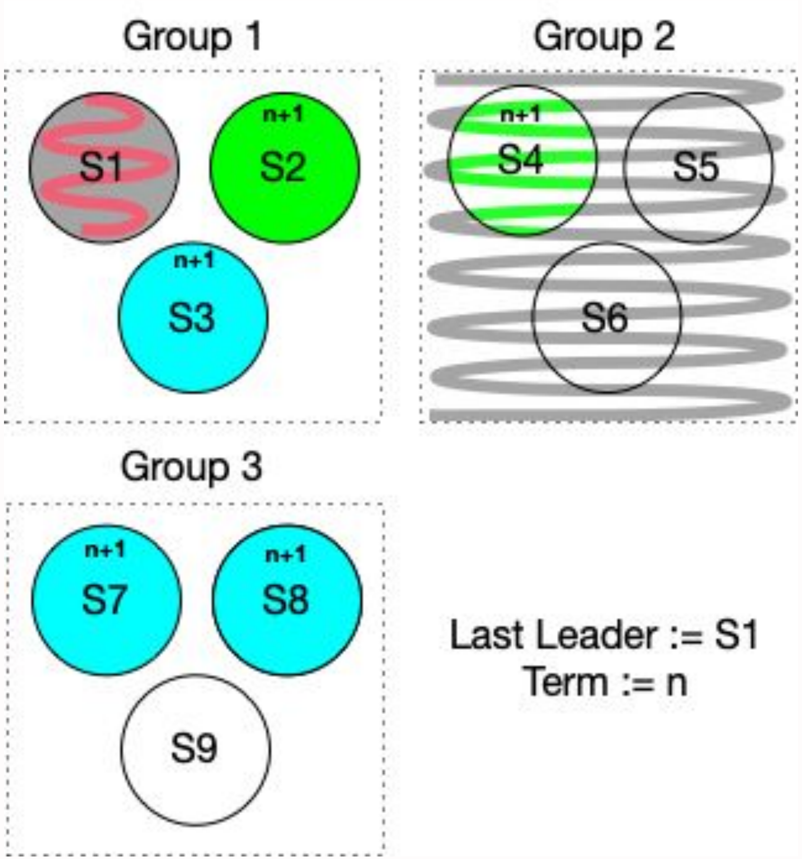


Fault tolerance

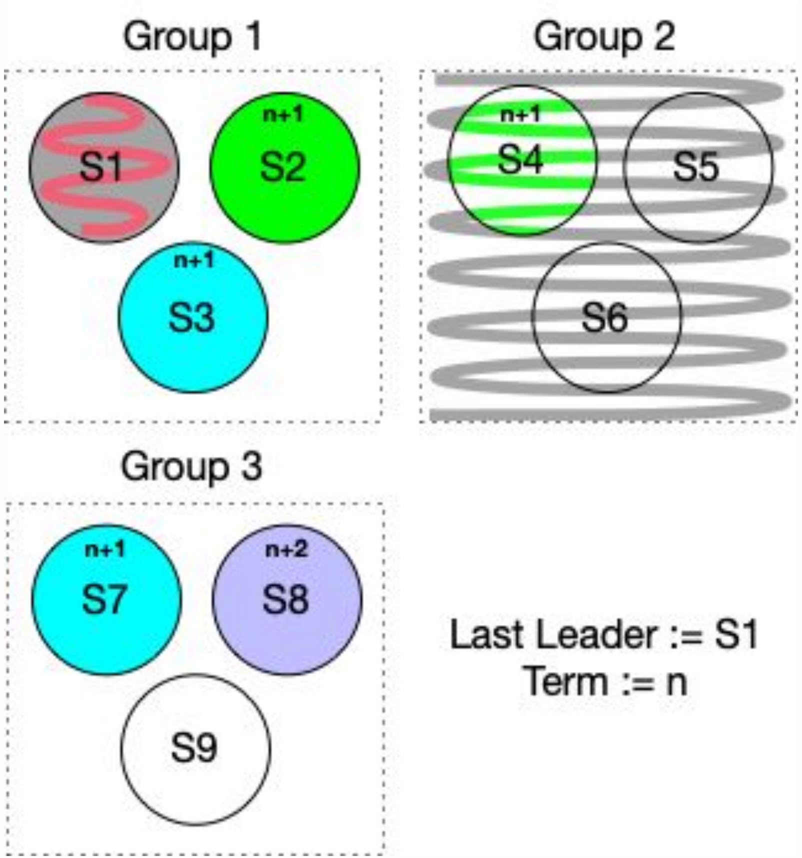


Last Leader := S1
Term := n

Fault tolerance



Fault tolerance



Experimental Results

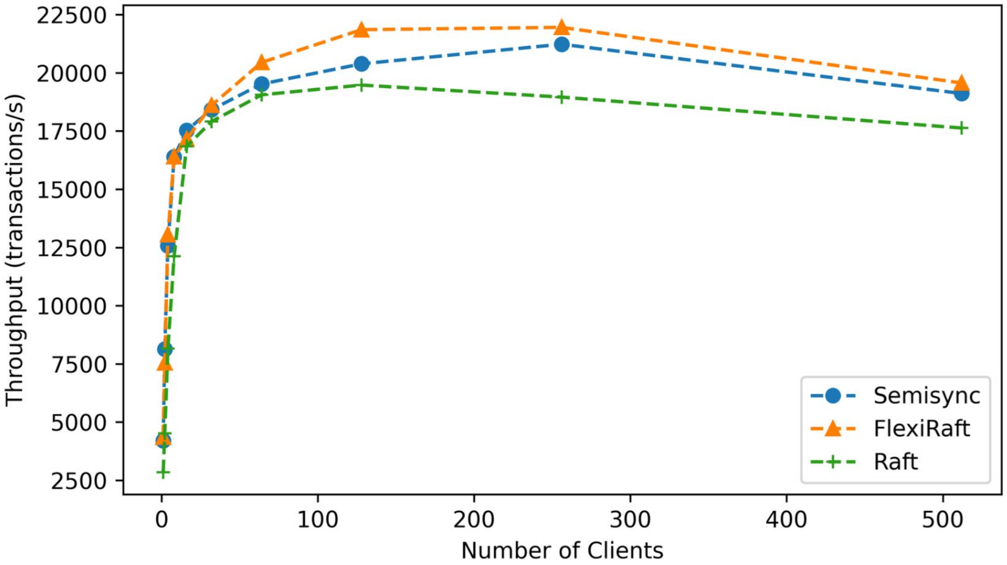


Figure 3. Comparison of throughput observed. Throughput is expressed as transactions per second.

Experimental Results

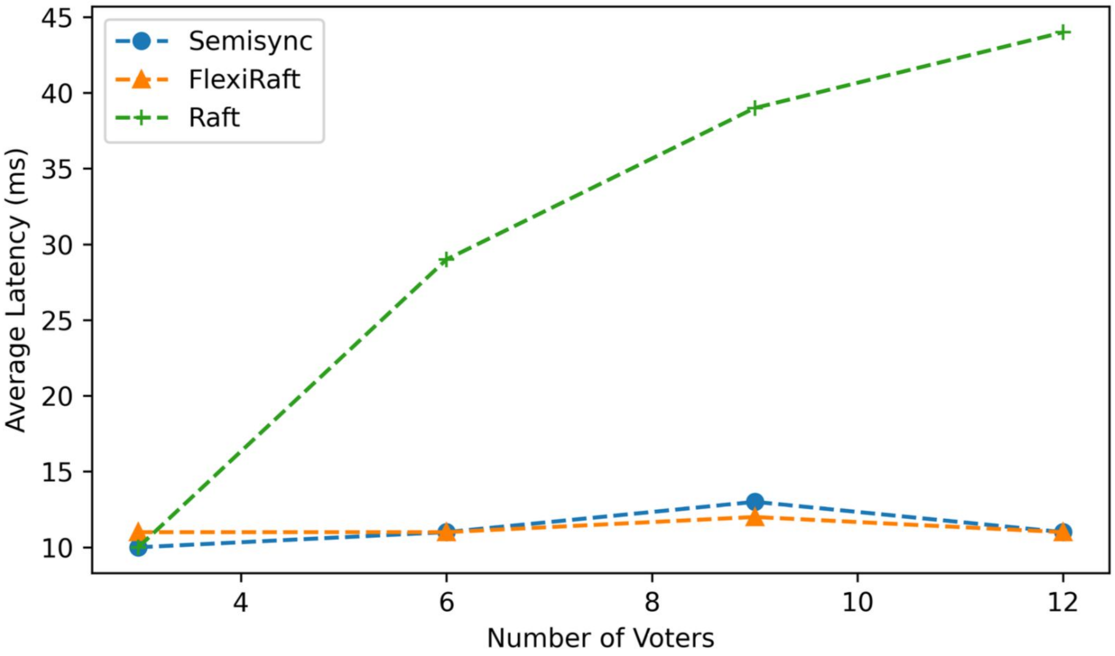


Figure 4. Effect on latency with increasing replica set size

Takeaways

- **Guardrails are essential** when offering the choice of quorum selection to end users
- **Implementation details matter**
  - Quorum aware optimizations to advance commit mark
  - Asynchrony in local vote counting
- Some **optional add-ons to Raft are critical for performance at scale**
  - Pre-voting
  - Joint consensus

Questions / Discussion