Dynamic Multi-Raft

Dongxu Huang

PingCAP

CNUTCon 2017
全球运维技术大会

上海·光大会展中心大酒店 | 2017.9.10-11

智能时代的新运维

大数据运维

DevOps

安全

SRE

Kubernetes

Serverless

游戏运维

AIOps

智能化运维

基础架构

监控

互联网金融

主办方 Geekbang> 极客邦科技 InfoQ

SPEAKER INTRODUCE

黄东旭 CTO & Cofounder, PingCAP

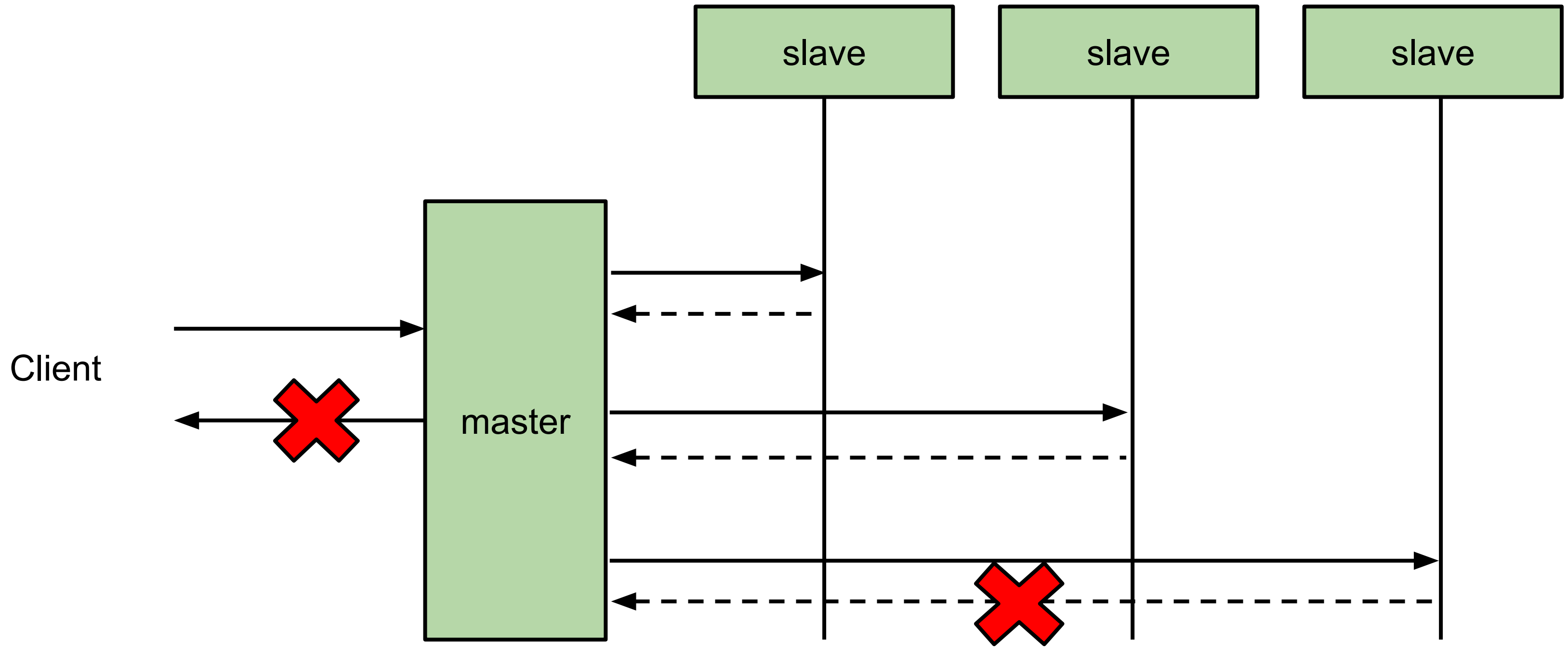
- MSRA / Netease / WandouLabs / PingCAP
- Hacker / Infrastructure software engineer
- Distributed system / Database / PL / …
- Codis / TiDB / TiKV
- Golang / Rust / Python

Consensus

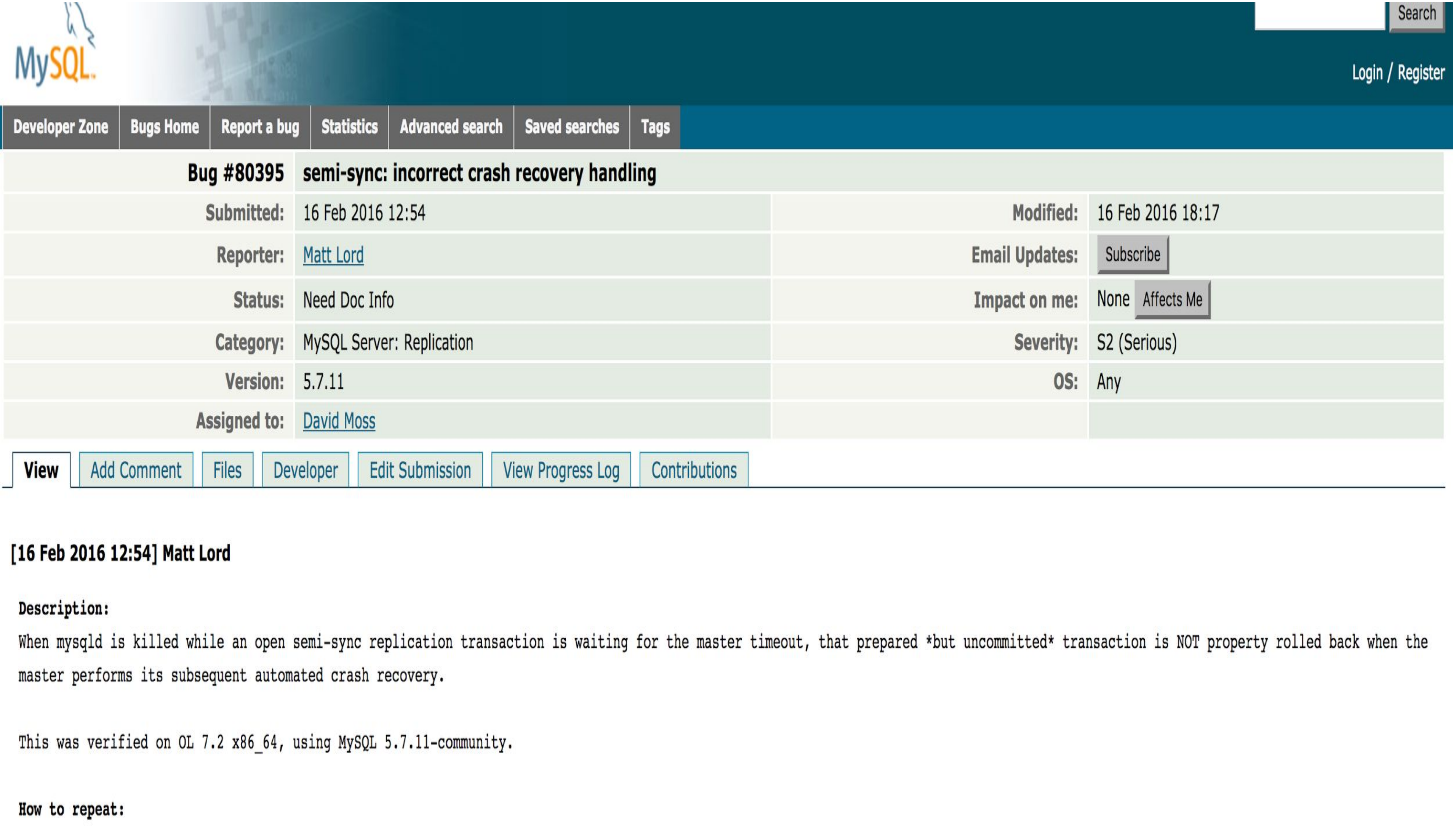
is the only problem in distributed system...

Modern HA

- Master-slave is not an option, why?



MySQL MHA + Semi-Sync?



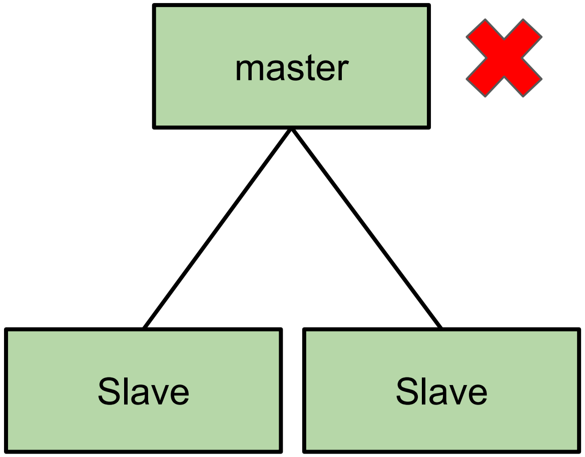
Bug #80395 **semi-sync: incorrect crash recovery handling**

Submitted:	16 Feb 2016 12:54	Modified:	16 Feb 2016 18:17
Reporter:	Matt Lord	Email Updates:	Subscribe
Status:	Need Doc Info	Impact on me:	None Affects Me
Category:	MySQL Server: Replication	Severity:	S2 (Serious)
Version:	5.7.11	OS:	Any
Assigned to:	David Moss		

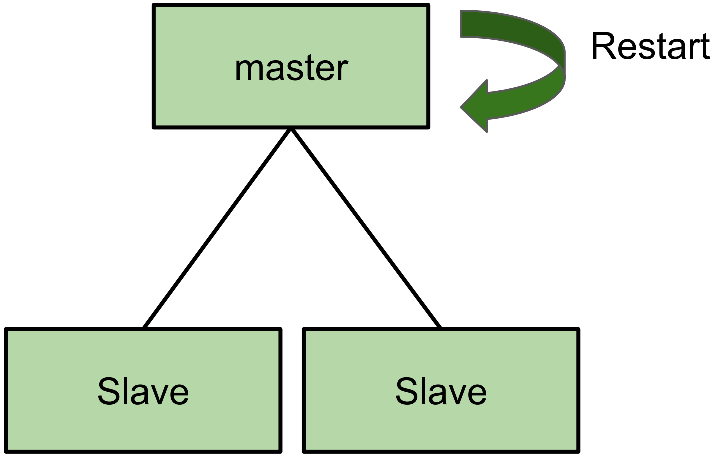
[16 Feb 2016 12:54] Matt Lord

```
Description:
When mysqld is killed while an open semi-sync replication transaction is waiting for the master timeout, that prepared *but uncommitted* transaction is NOT property rolled back when the
master performs its subsequent automated crash recovery.

This was verified on OL 7.2 x86_64, using MySQL 5.7.11-community.

How to repeat:
```

(1) Insert A, but master crash when doing semi-sync

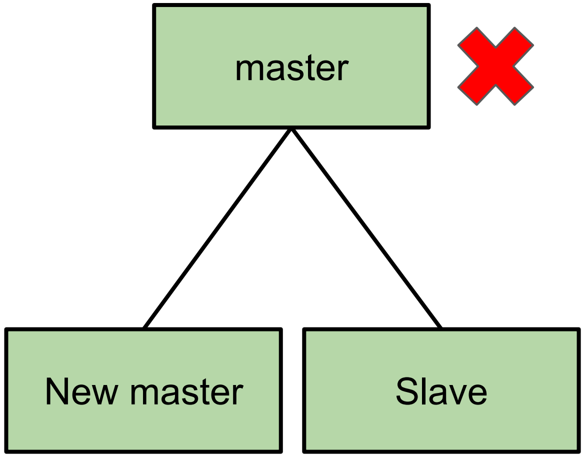


(2) Read A, OK; But degrade to async-replication



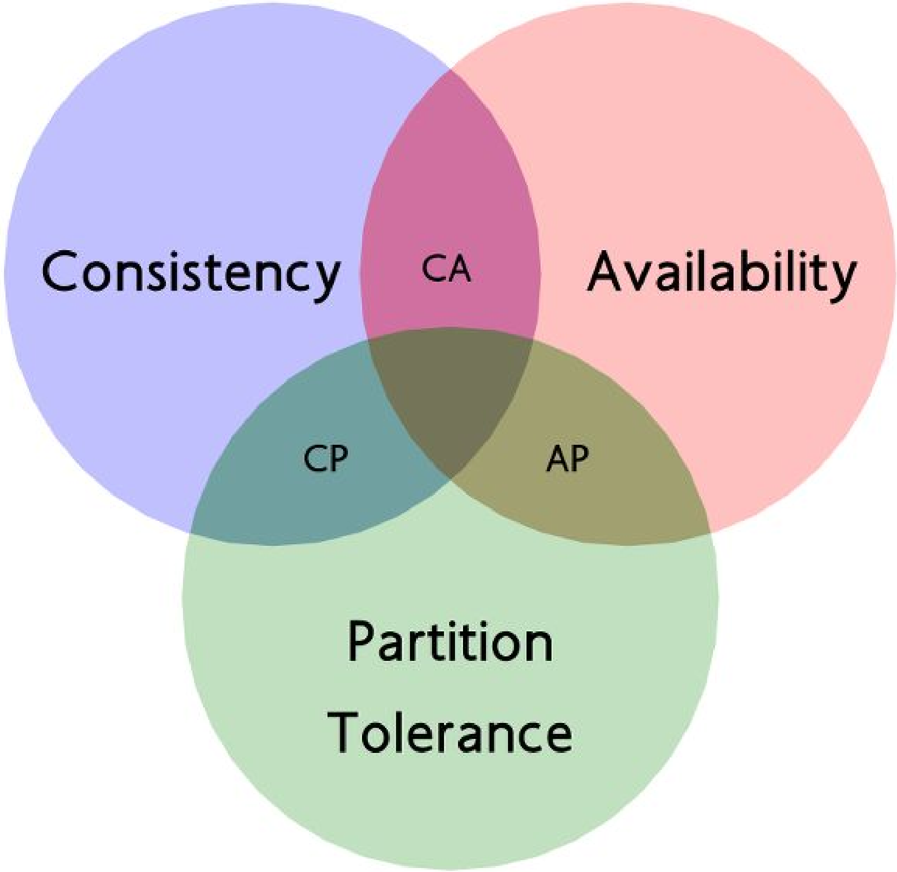
(3) Master crash again...then new master is elected

Read A, fail, oops...

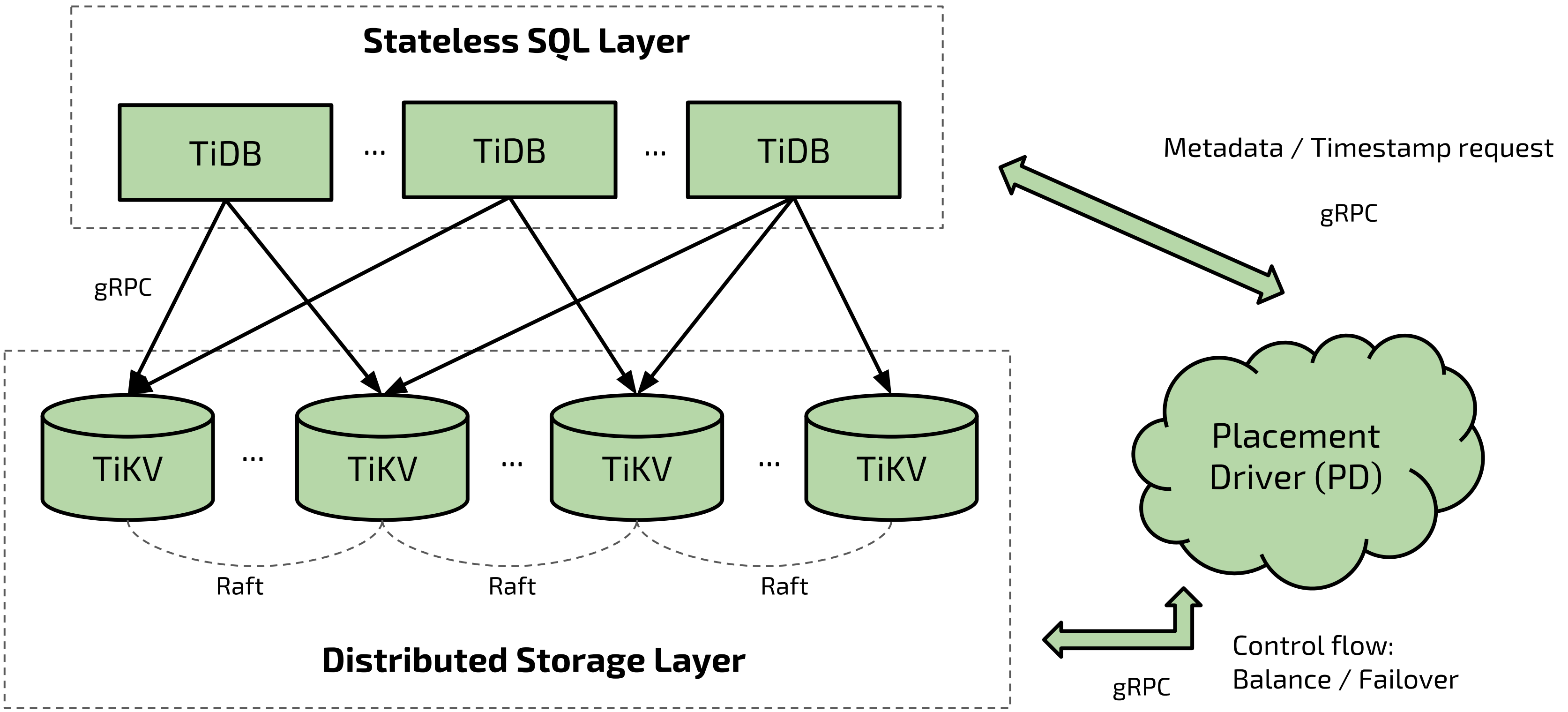


TiDB Project (Requirement)

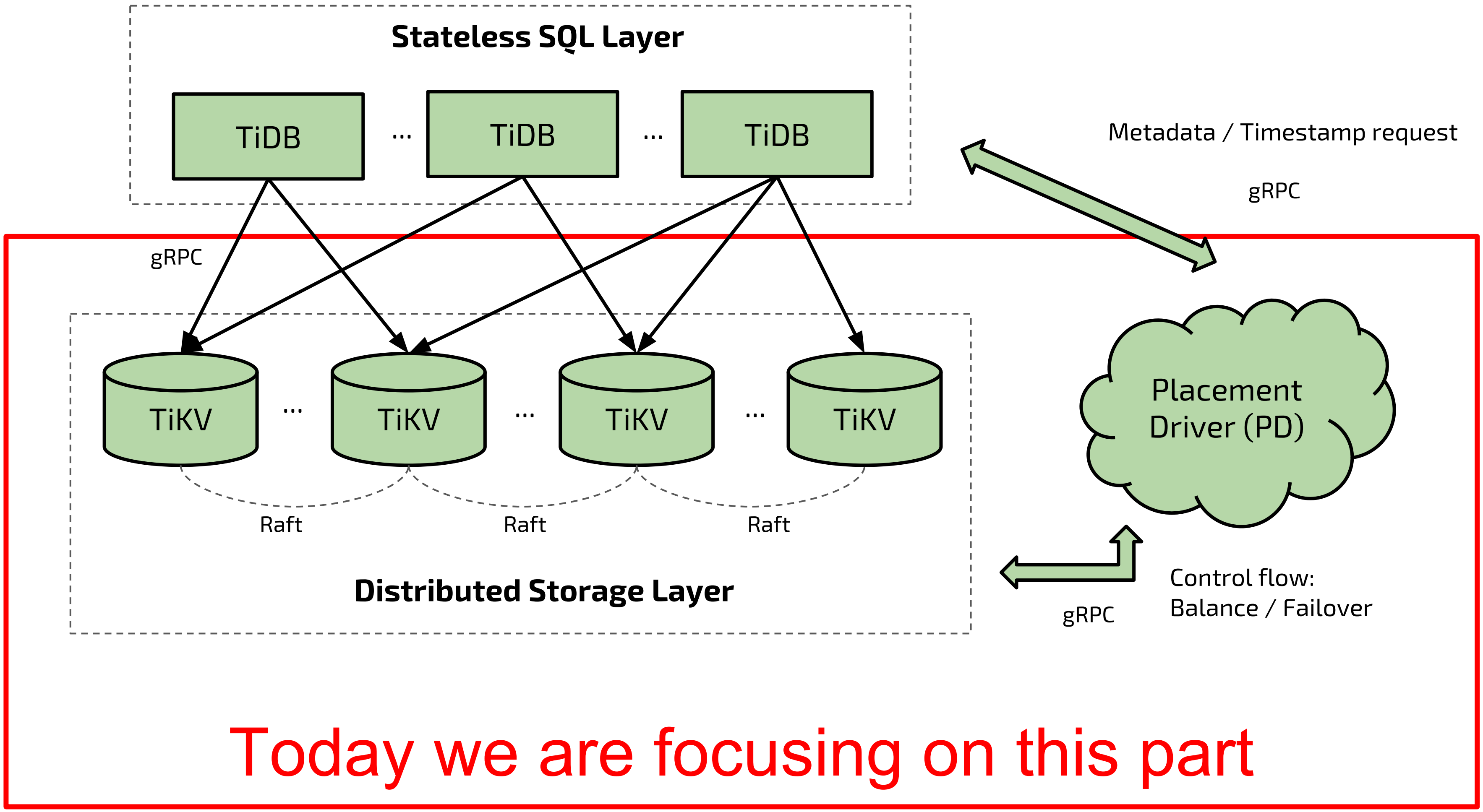
- Strong consistency
- Scalability
- High availability



TiDB Project Overview



TiDB Project Overview



Today we are focusing on this part

Replicated State Machines

- All servers execute same commands in same order
- System makes progress as long as any majority of servers up
- Agreement on shared state (single system image)
- Recovers from server failures autonomously
    - Minority of servers fail: no problem
    - Majority fail: lose availability, retain strong consistency
- IMHO, there are only two RSM implementations:
    - Multi-Paxos / Raft

The problem in Paxos (Multi Paxos)

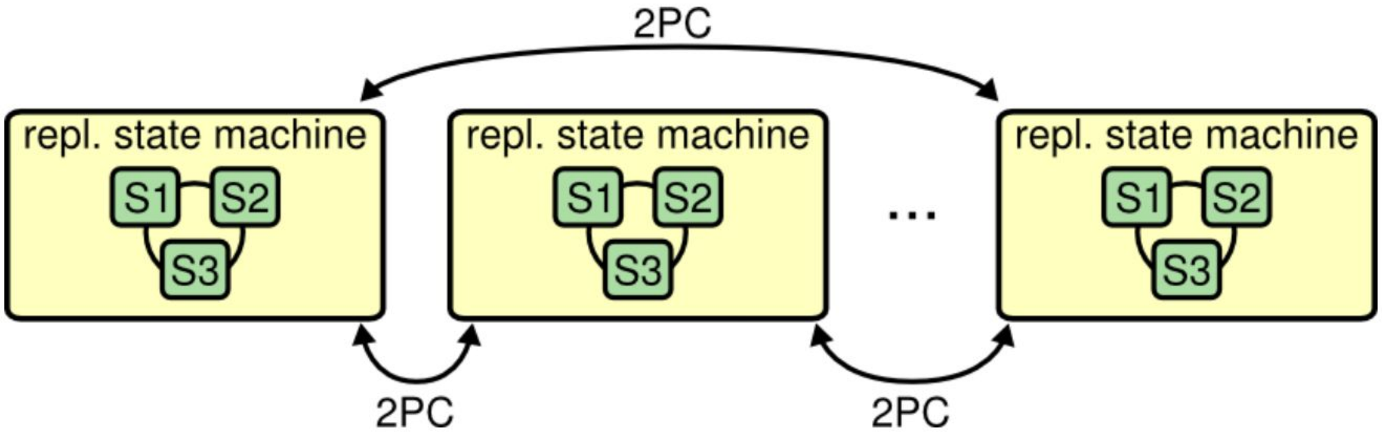
> *“The dirty little secret of the NSDI community is that at most five people really, truly understand every part of Paxos;-).”*
>
> *—NSDI reviewer*

Raft saves the day

- **Leader election**
  - Select one of the servers to act as cluster leader
  - Detect crashes, choose new leader
- **Log replication**
  - Leader takes commands from clients, appends to its log
  - Leader replicates its log to other servers (overwriting inconsistencies)
- **Safety**
  - Only a server with an up-to-date log can become leader

Use Raft in database

- Single RSM is **NOT** gonna work.
- You need 2PC to retain strong consistency across different RSMs.

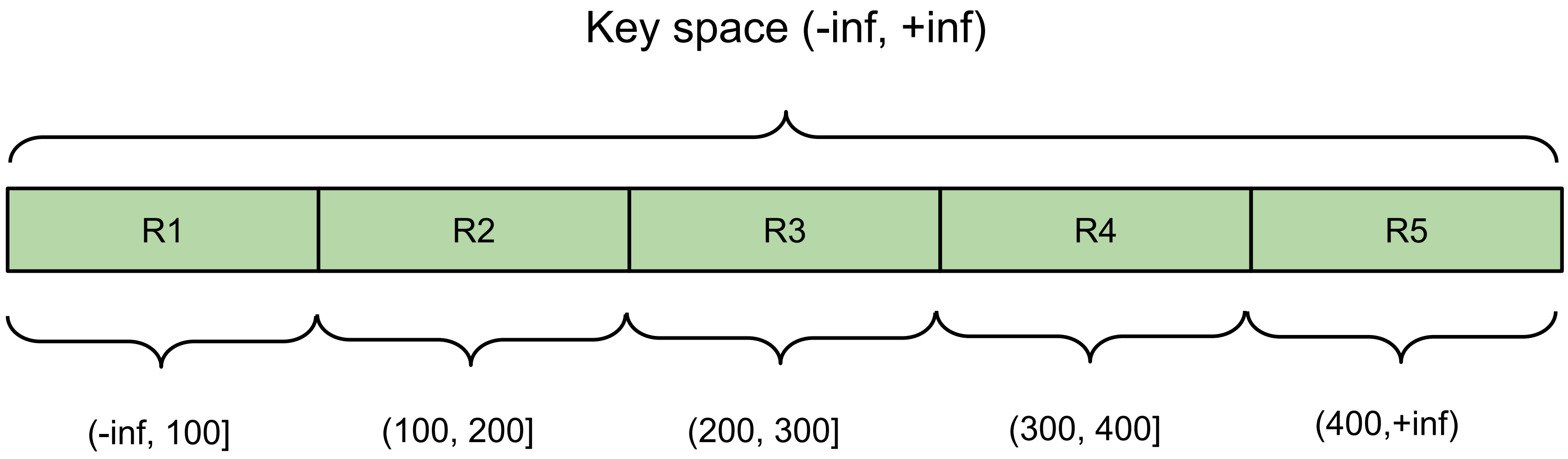


Raft in database

- How to shard?
- How to split / merge dynamically?
- How to balance the workload?
- How to improve the throughput?

Sharding Raft in TiKV

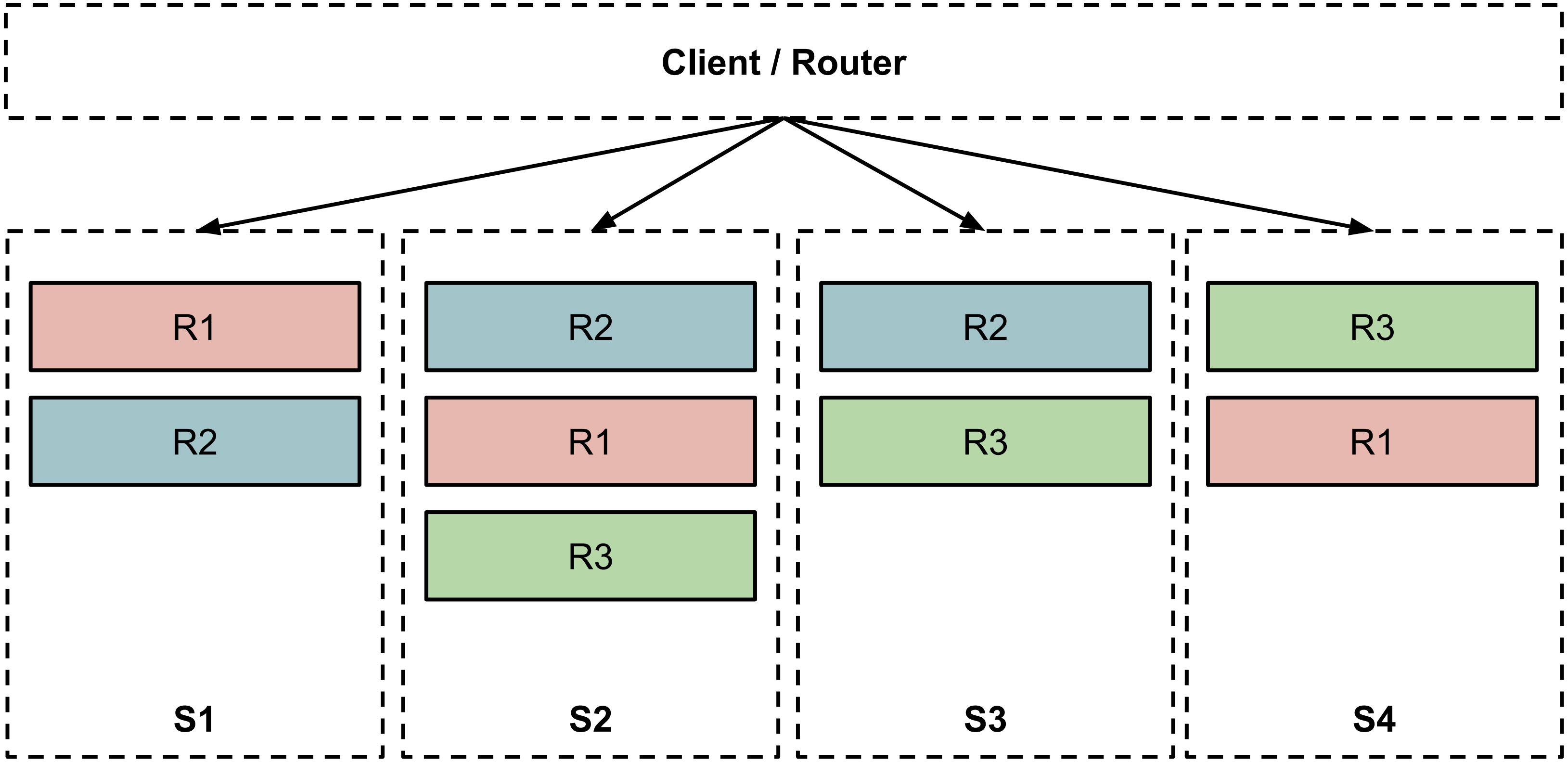
- Split key space into **Regions (normally in byte-order) logically**
- Each region is a raft group
  - Default size: 96 ~ 128 MB
  - Why?



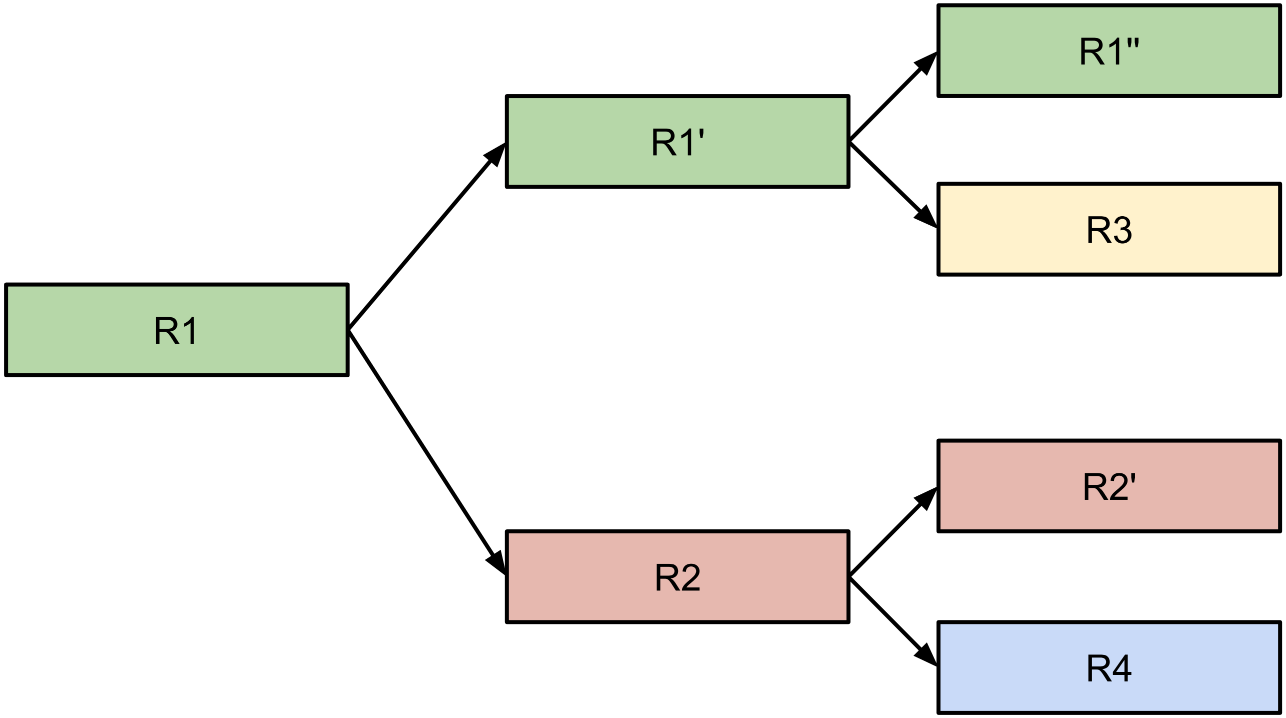
Meta data storage

- We stores region meta in an in-memory B-Tree (in PD)
  - Sorted by the start key of region
  - We can find the right region which contains specific key in O(log N)
- PD is not ‘***the source of truth’***, data server is. Why?
  - Split is always happening
  - The metadata stored in PD may be out-of-date
  - Retry is important

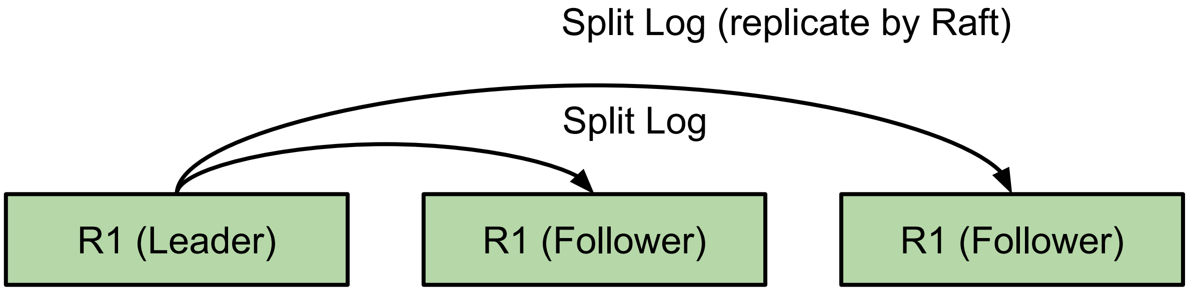
Sharding Raft in TiKV



Sharding Raft in TiKV

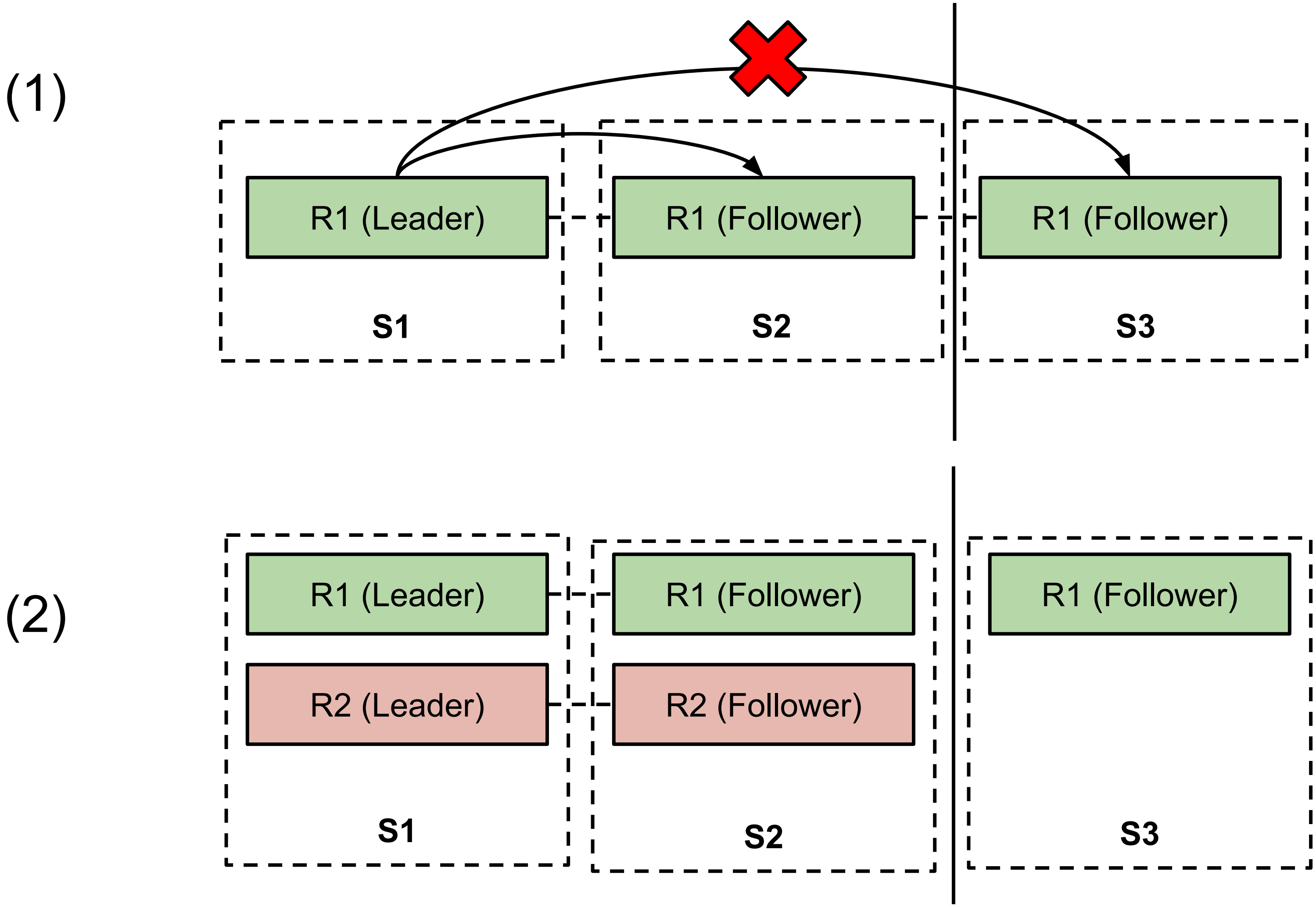


Dynamic split / merge



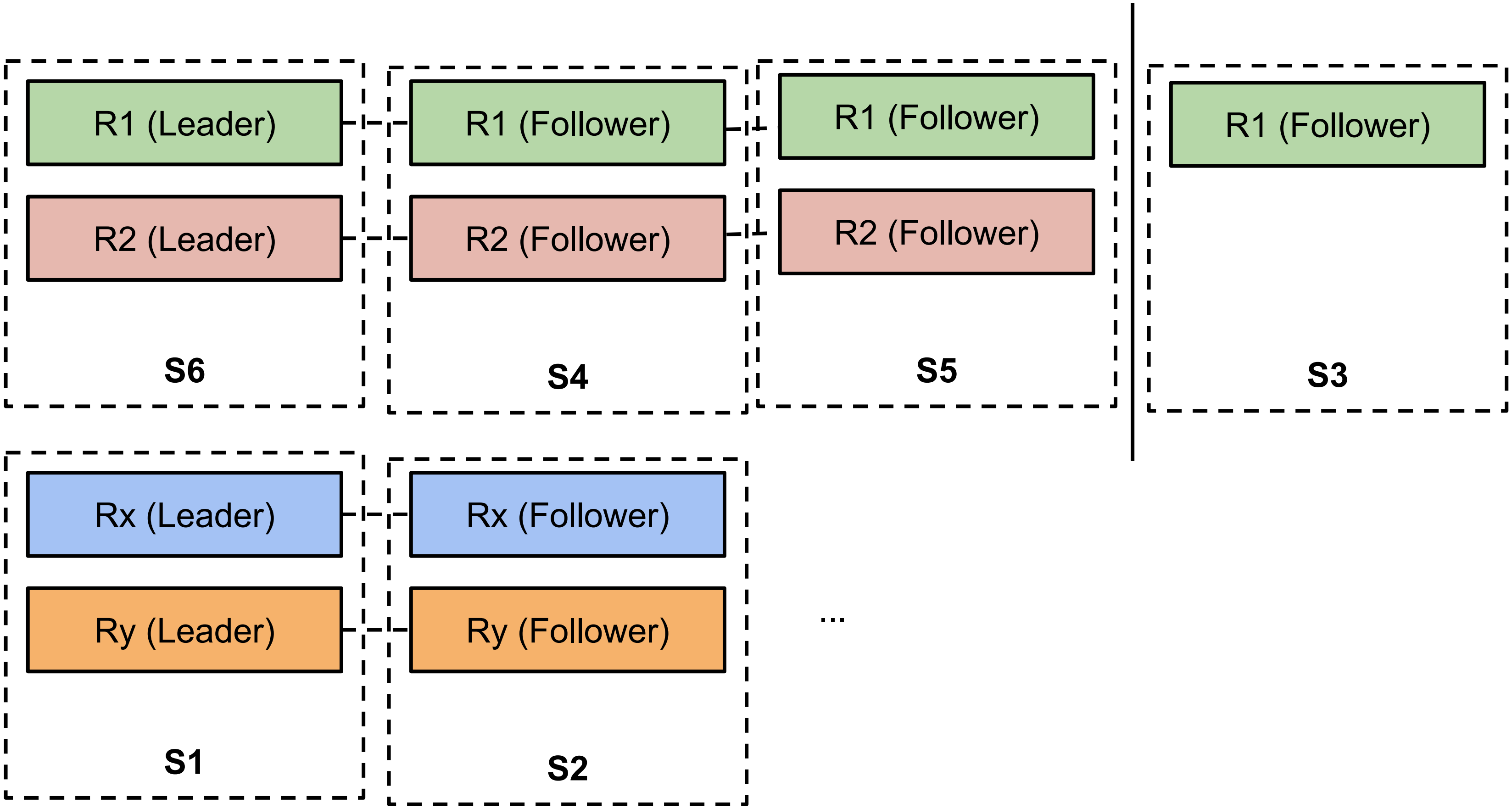
Simple...Huh?

An abnormal situation...



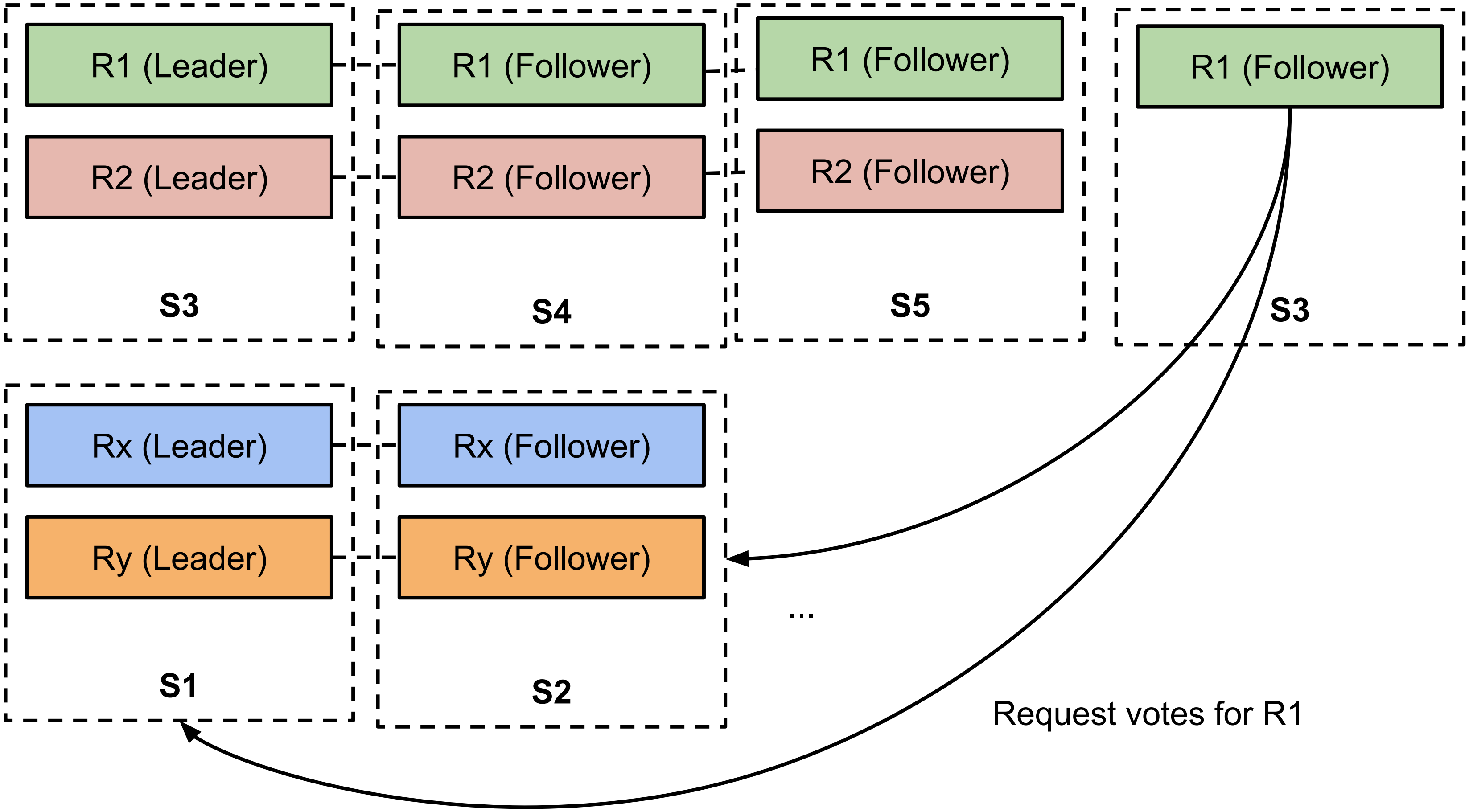
An abnormal situation...

(3) After N rounds of split or membership changes...

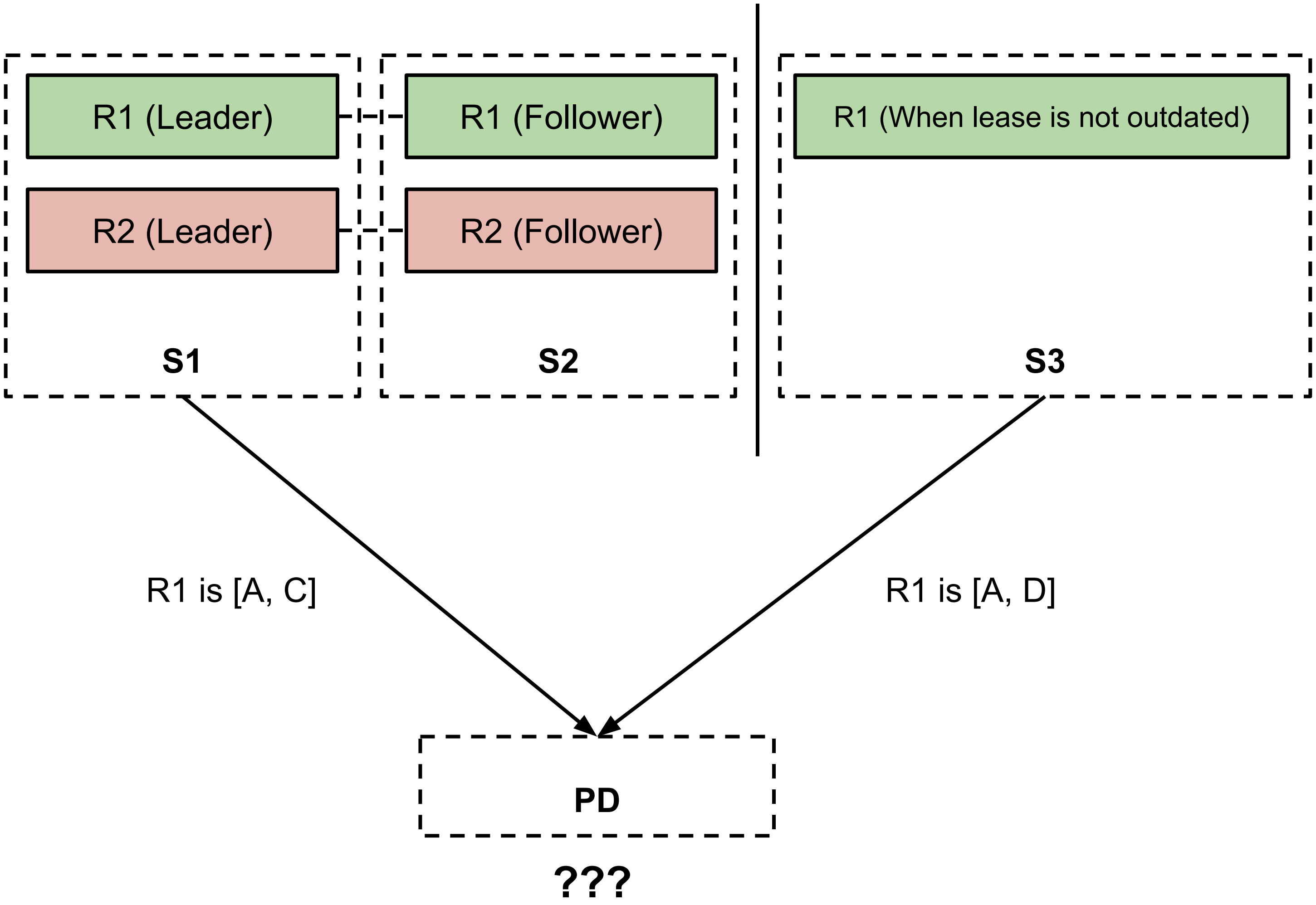


An abnormal situation...

(4)



Another abnormal situation

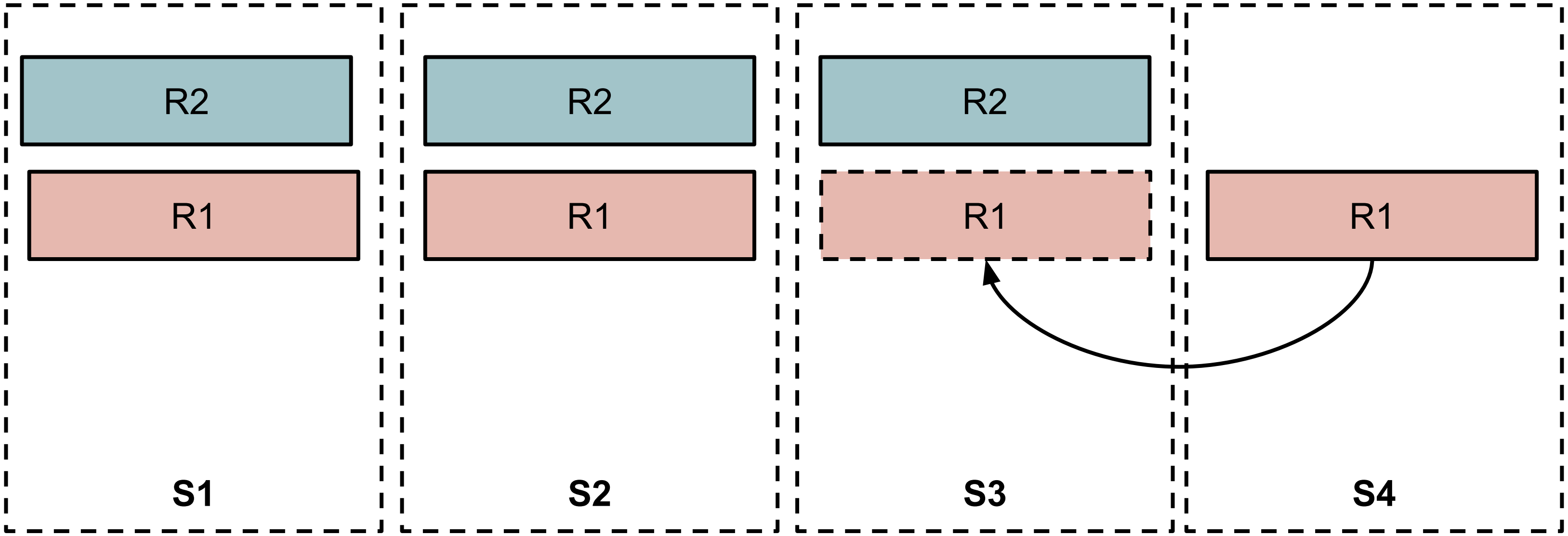


Introduce Region Epoch

- Epoch(Region X) := {ConfVer, SplitVer}
- Every configuration change in Region X will increase the ConfVer
- Every split occurs in Region X will increase the SplitVer
- Let’s say Epoch(R1) >= Epoch(R2), if and only if:
  - ConfVer(R1) >= ConfVer(R2) and SplitVer(R1) >= SplitVer(R2)
- Larger epoch always win

What about merge?

- Make sure all replica for these two regions are in same nodes
- And no more rebalance for these two regions

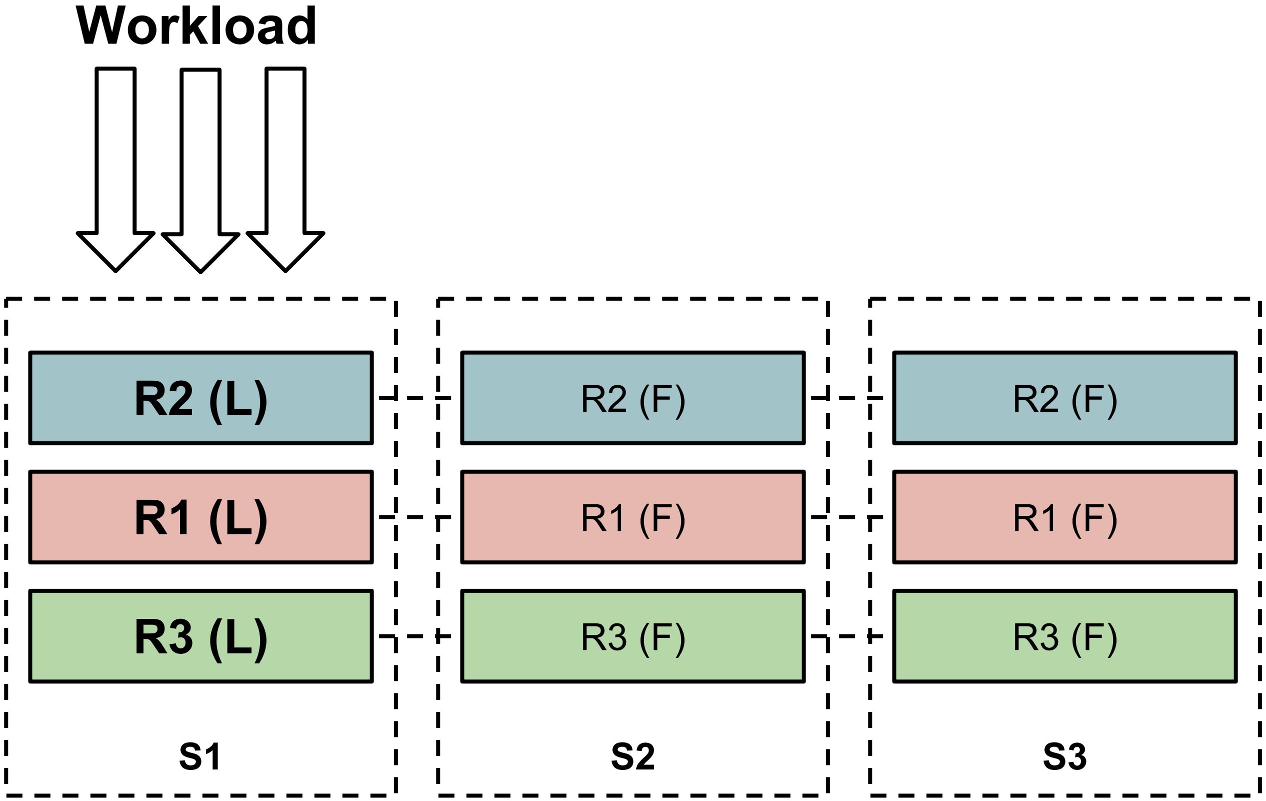


Storage

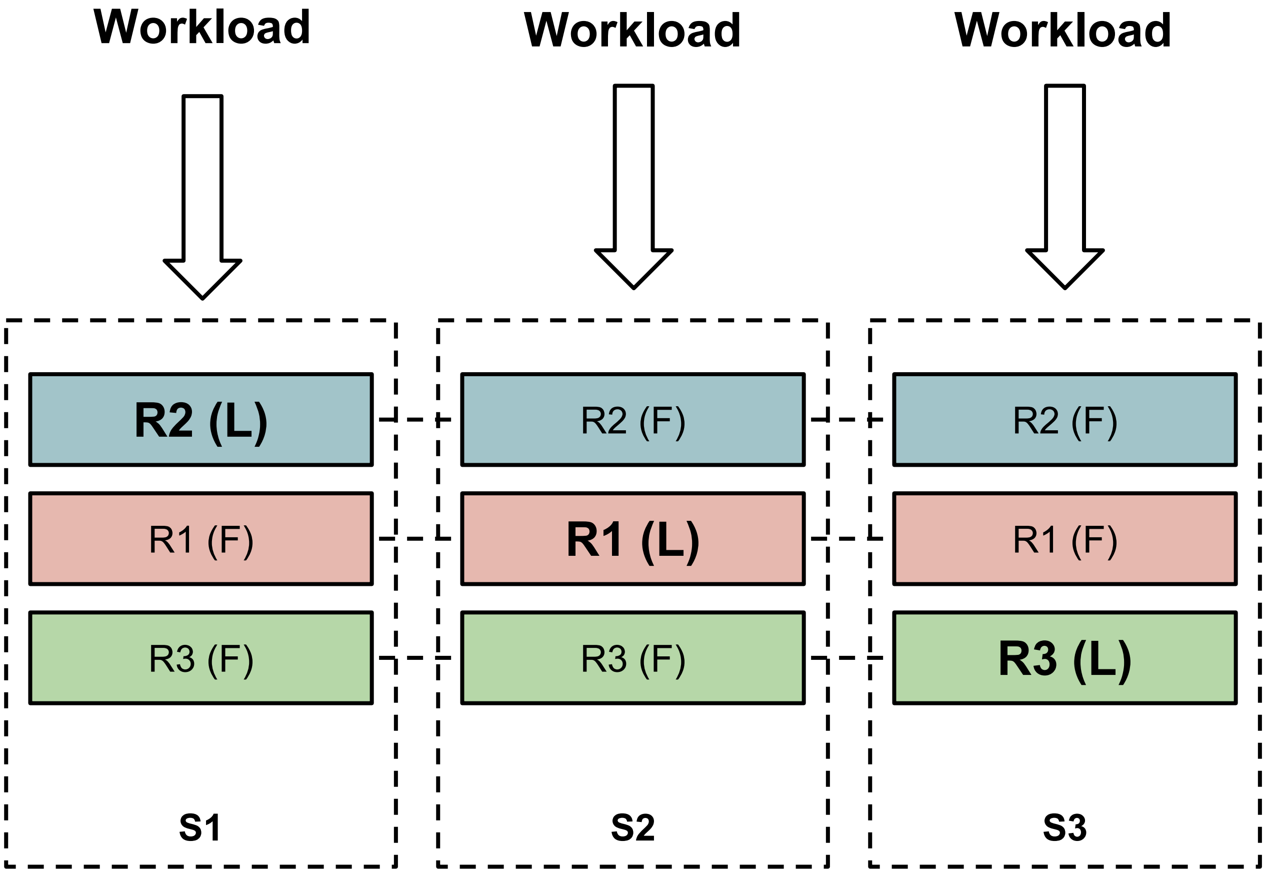
- RSM storage
  - All regions in the same physical node share one RocksDB instance
- Log storage
  - Journal-like storage
  - Share with region storage

Leadership transfer

- For fast rebalance, since Raft is a randomized algorithm, there is a certain probability that one node has too many leaders.

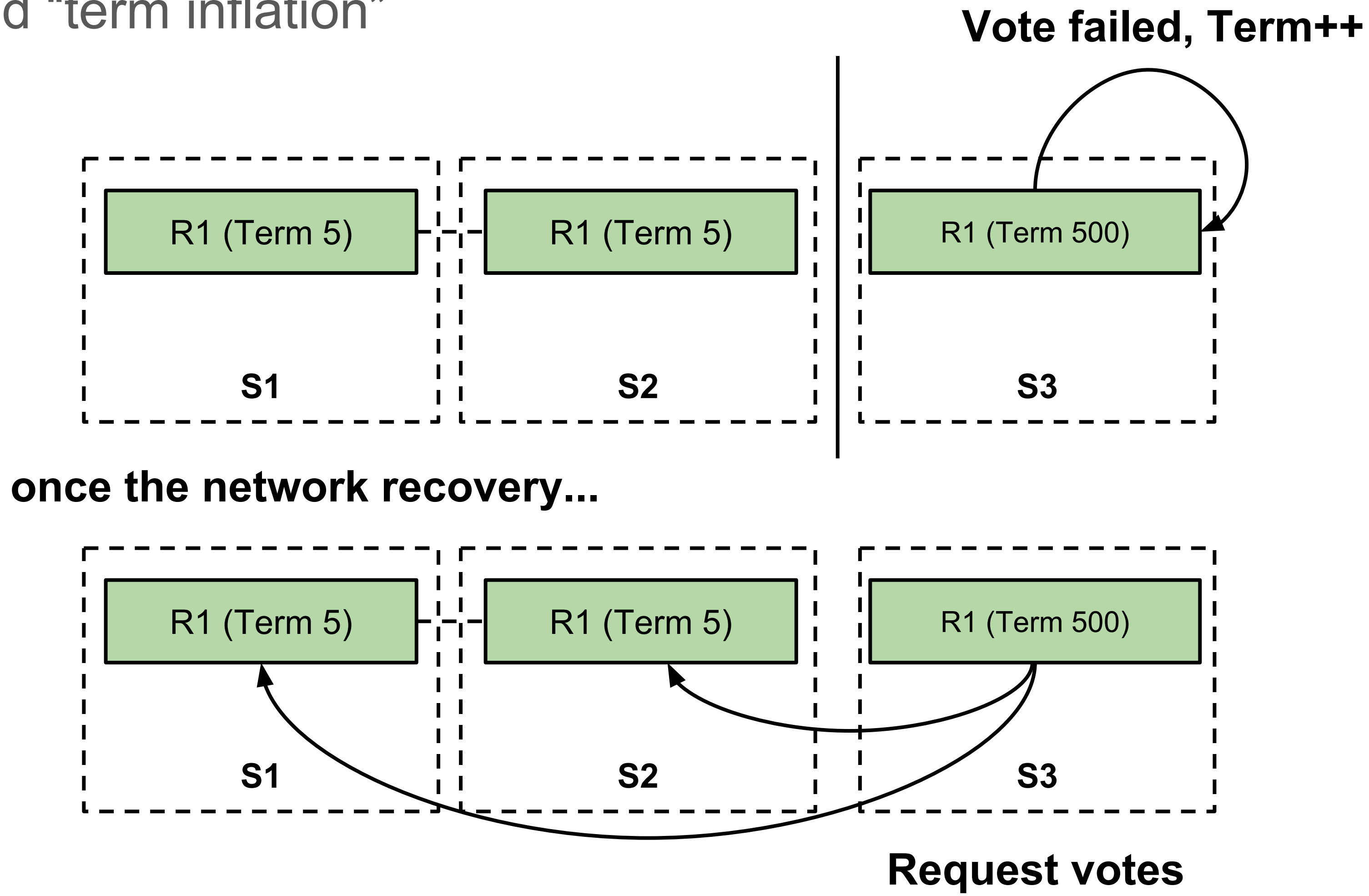


Leadership transfer

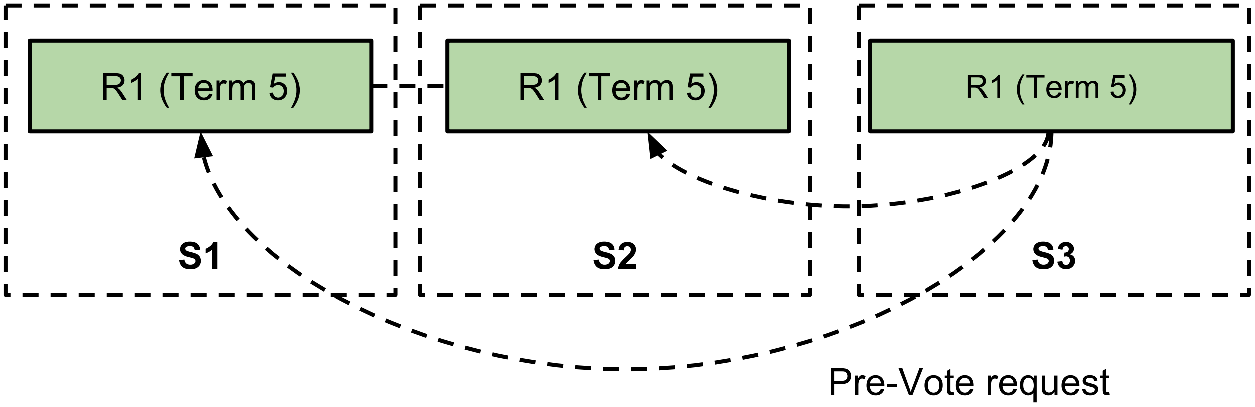


Pre-vote algorithm

- Avoid “term inflation”



Pre-vote algorithm



S3 sends Pre-vote request to S1 and S2 to make sure S3’s log is up-to-date, when S3 receives responses from a majority of the cluster, S3 will increase its term and start a normal election

How to test

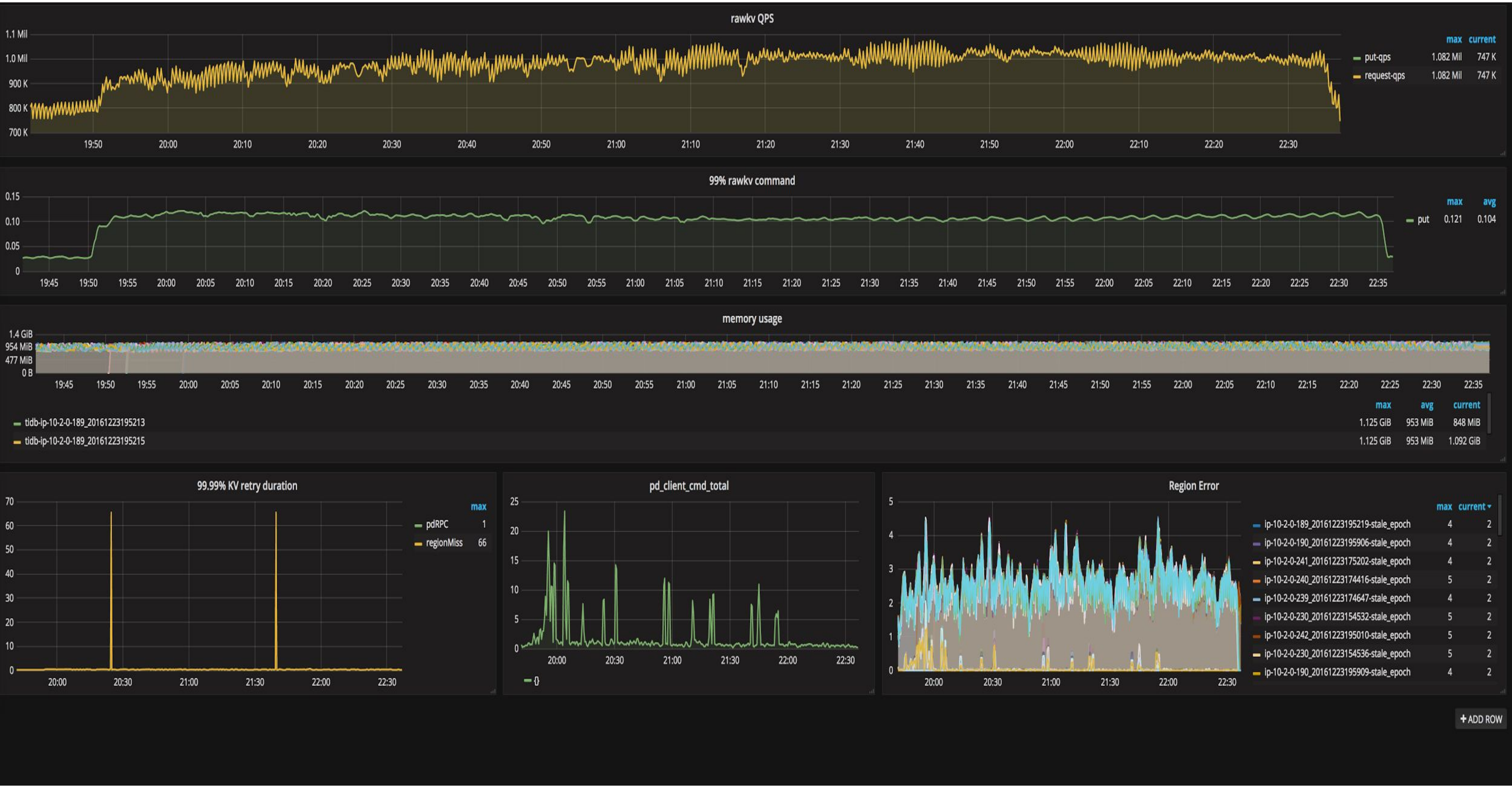
- Testing in distributed system is really hard
- Test-Driven Development
- Test cases from community
    - Lots of tests in MySQL drivers/connectors
    - Lots of ORMs
    - Lots of applications (Record---replay)
- Fault injection
    - Hardware: disk error, network card, cpu, clock
    - Software: file system, network and protocol
- Simulate everything：Network
- Distribute testing
    - Jepsen
    - Namazu

Benchmark

- 46 Physical nodes
- 460 TiKV instances (1 tikv instance for 1 HDD)
- TiKV Raw API Put (Raft)

Put(key, value)
key size: 21 bytes
value size: random (1~100 bytes)

Benchmark



rawkv QPS

	max	current
put-qps	1.082 Mil	747 K
request-qps	1.082 Mil	747 K

99% rawkv command

	max	avg
put	0.121	0.104

memory usage

	max	avg	current
tidb-ip-10-2-0-189_20161223195213	1.125 GiB	953 MiB	848 MiB
tidb-ip-10-2-0-189_20161223195215	1.125 GiB	953 MiB	1.092 GiB

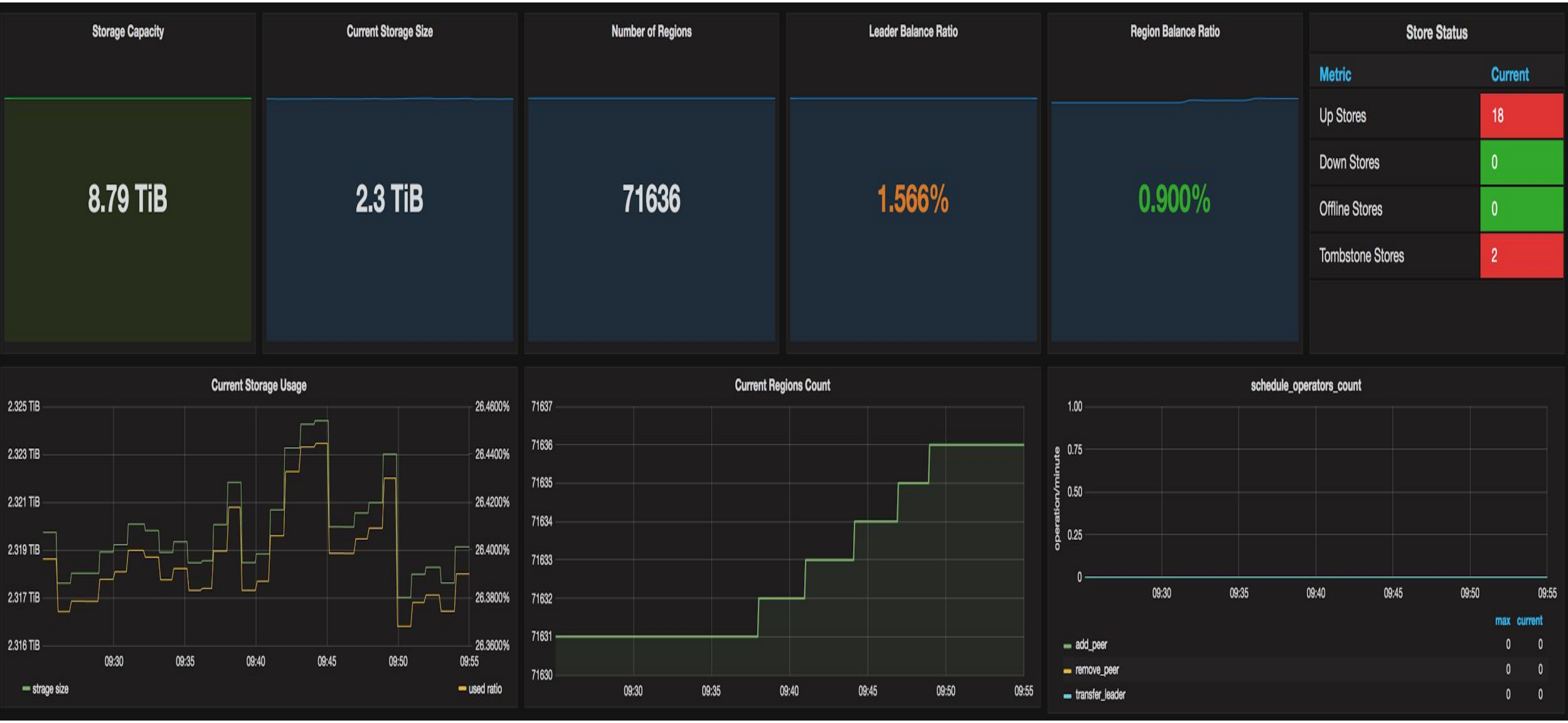
99.99% KV retry duration

	max
pdRPC	1
regionMiss	66

Region Error

	max	current ▾
ip-10-2-0-189_20161223195219-stale_epoch	4	2
ip-10-2-0-190_20161223195906-stale_epoch	4	2
ip-10-2-0-241_20161223175202-stale_epoch	4	2
ip-10-2-0-240_20161223174416-stale_epoch	5	2
ip-10-2-0-239_20161223174647-stale_epoch	4	2
ip-10-2-0-230_20161223154532-stale_epoch	5	2
ip-10-2-0-242_20161223195010-stale_epoch	5	2
ip-10-2-0-230_20161223154536-stale_epoch	5	2
ip-10-2-0-190_20161223195909-stale_epoch	4	2

Benchmark



Storage Capacity

8.79 TiB

Current Storage Size

2.3 TiB

Number of Regions

71636

Leader Balance Ratio

1.566%

Region Balance Ratio

0.900%

Store Status

Metric	Current
Up Stores	18
Down Stores	0
Offline Stores	0
Tombstone Stores	2

Current Storage Usage

Current Regions Count

schedule_operators_count

	max	current
add_peer	0	0
remove_peer	0	0
transfer_leader	0	0

THANKS!

2017 ArchSummit 深圳
TiDB 讨论

让创新技术推动社会进步

HELP TO BUILD A BETTER SOCIETY WITH
INNOVATIVE TECHNOLOGIES

Geekbang.

极客邦科技

InfoQ queue

专注中高端技术人员的技术媒体

EGO EXTRA GEEKS' ORGANIZATION NETWORKS

高端技术人员学习型社交平台

StuQ queue

斯达克学院

实践驱动的 IT 教育平台

地址：北京市朝阳区洛娃大厦C座8层1801室

网址：www.geekbang.org