PingCAP

# Go 搭建大型开源分布式数据库技术内幕

shenli@PingCAP

PingCAP

# 关于我

- 申砾 (Shen Li)
- TiDB 技术负责人
- 网易有道 / 360搜索 / PingCAP
- Infrastructure software engineer

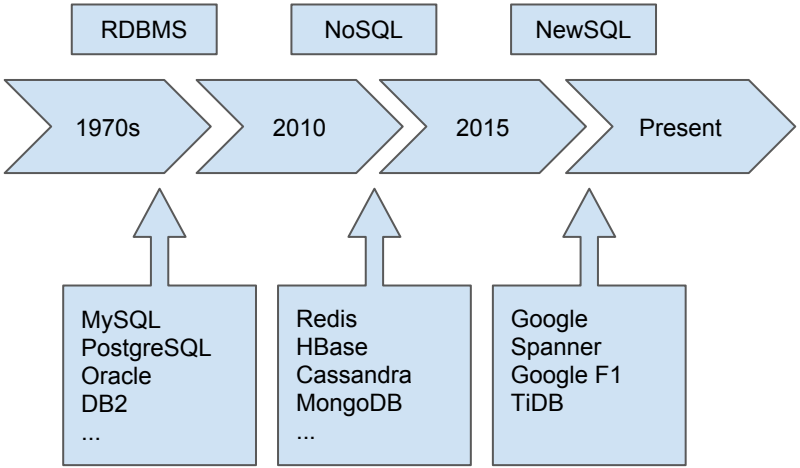
PingCAP

# 为什么需要一个新的数据库？

PingCAP

# 从单机数据库到 NewSQL

- 关系型数据库
- NoSQL
- 中间件
- NewSQL



PingCAP

# NewSQL 是什么

- 水平扩展
- 事务
- 高可用 & 自动故障恢复
- SQL

# TiDB

- Scalability as the first class feature
- SQL is necessary
- Compatible with MySQL, in most cases
- OLTP + OLAP = HTAP (Hybrid Transactional/Analytical Processing)
- 24/7 availability, even in case of datacenter outages
- Open source, of course

TiDB

A Distributed SQL Database

PingCAP

# 如何构建分布式数据库？

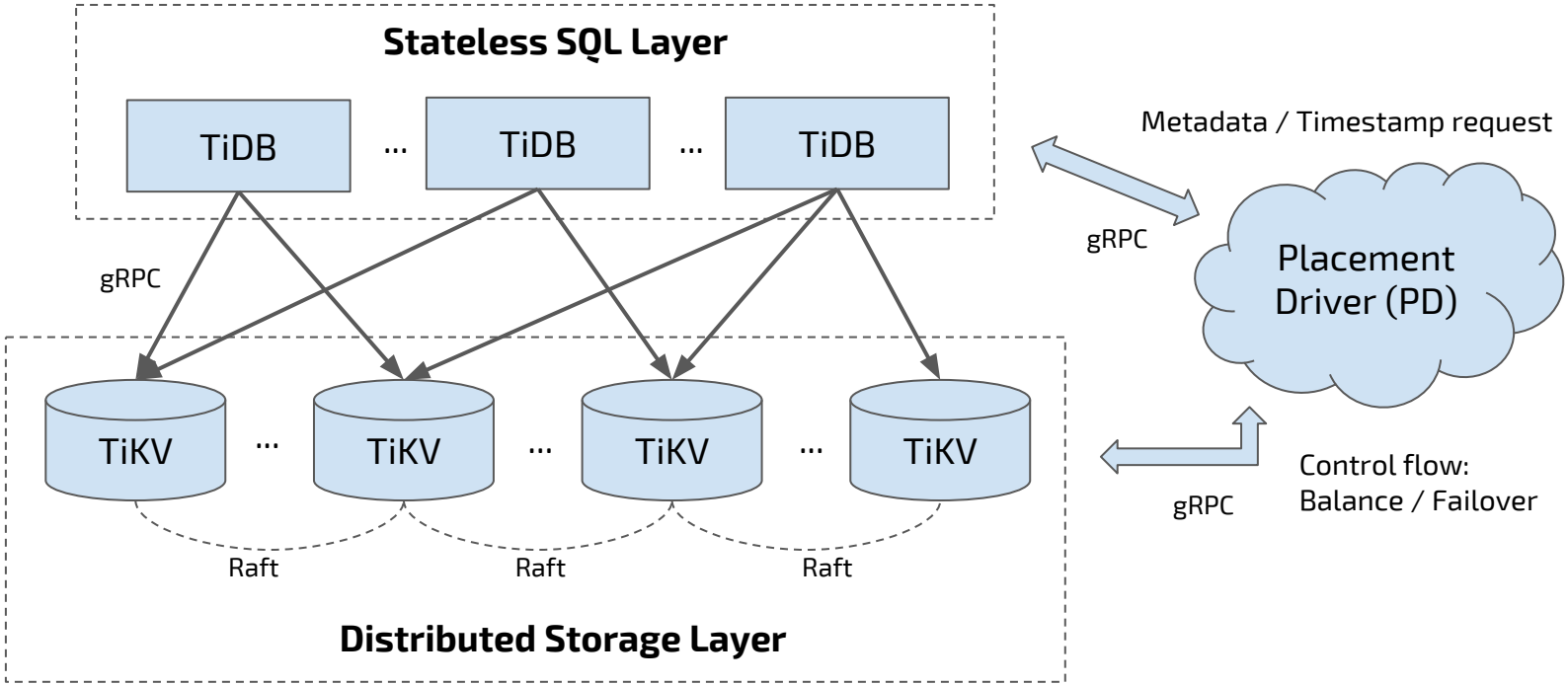
PingCAP

# 原则

- 分层
- **Make it right and make it fast.**
- 测试很重要
- 简单易用
- 和社区结合

PingCAP

# 架构


Stateless SQL Layer
TiDB
...
TiDB
...
TiDB
Metadata / Timestamp request
gRPC
gRPC
Placement Driver (PD)
TiKV
...
TiKV
...
TiKV
...
TiKV
gRPC
Control flow: Balance / Failover
Raft
Raft
Raft
Distributed Storage Layer

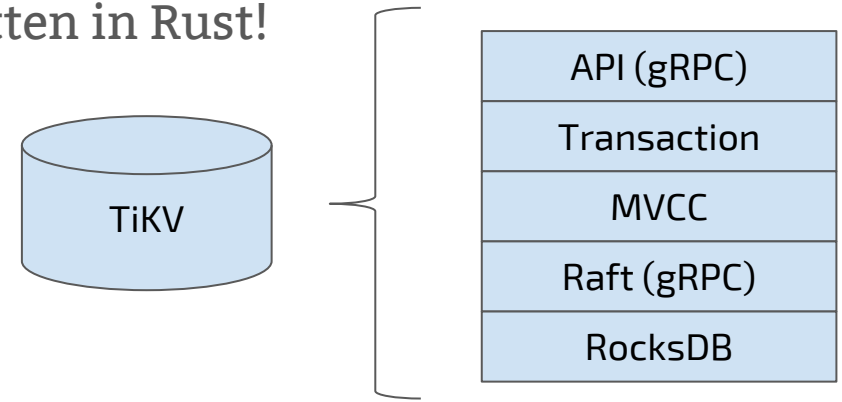
PingCAP

# 数据分片

- Hash Based Partition
  - Redis
  - 不利于范围 Scan
- Range Based Partition
  - Hbase
  - Range 需要足够大且足够小

# Storage stack 1/2

- TiKV is the underlying storage layer
- Physically, data is stored in RocksDB
- We build a Raft layer on top of RocksDB
  - What is Raft?
- Written in Rust!

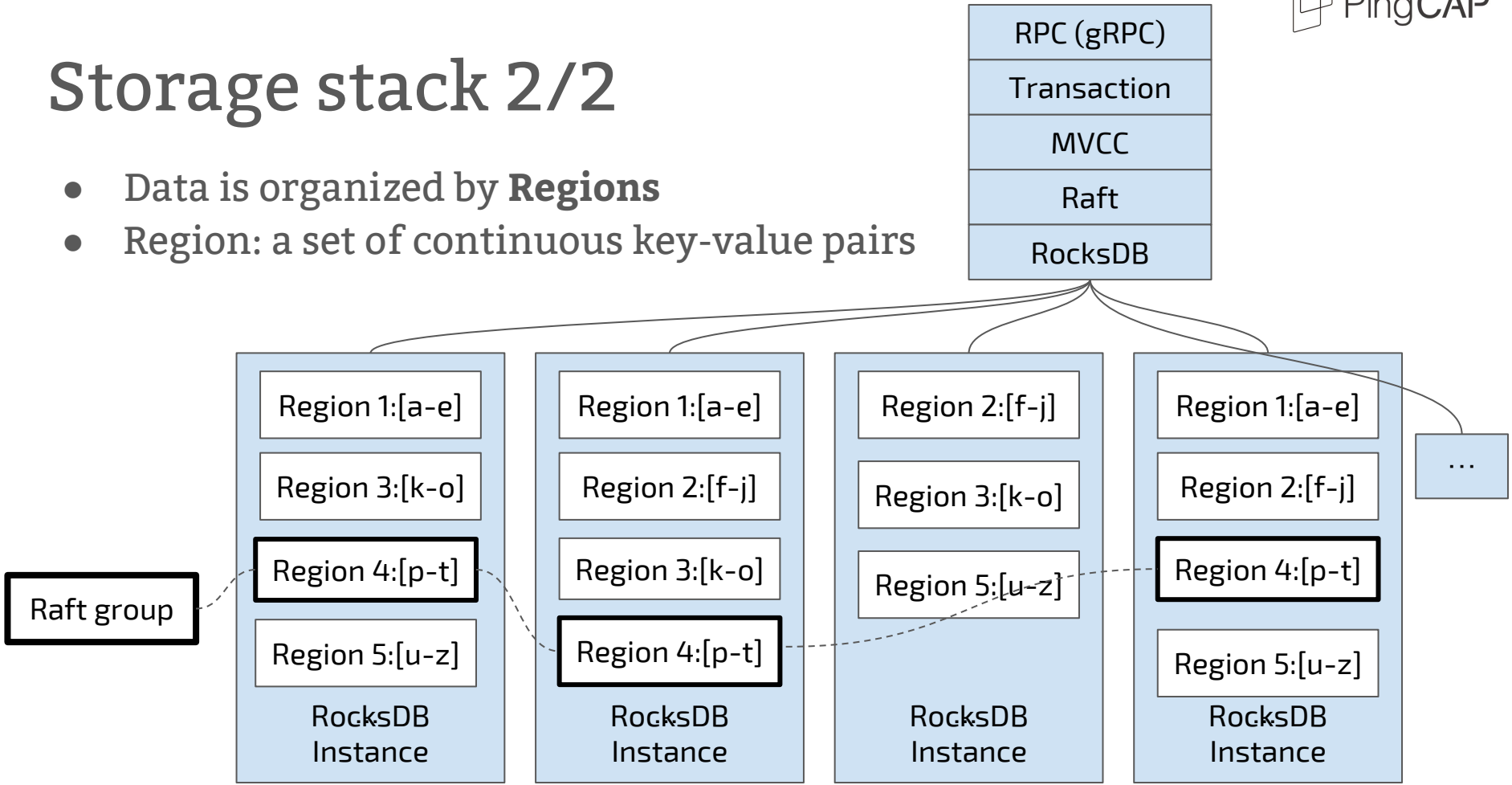



**Transactional KV API**
(https://github.com/pingcap/tidb/blob/master/cmd/benchkv/main.go)

**Raw KV API**
(https://github.com/pingcap/tidb/blob/master/cmd/benchraw/main.go)

# Storage stack 2/2

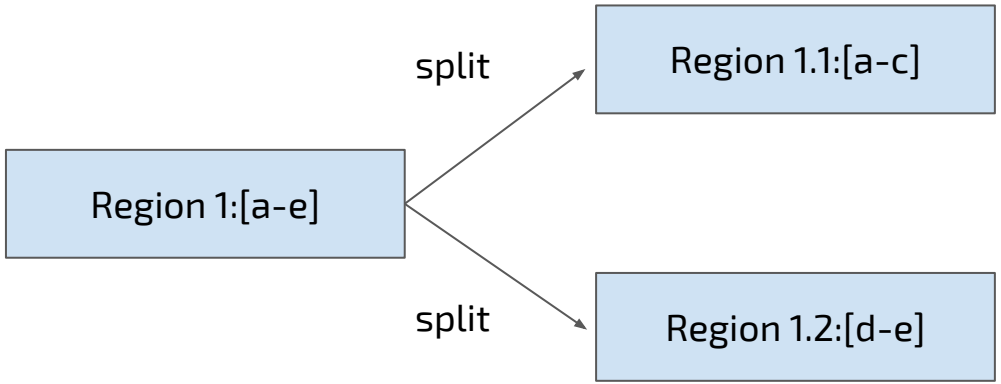
- Data is organized by **Regions**
- Region: a set of continuous key-value pairs



PingCAP

# Raft

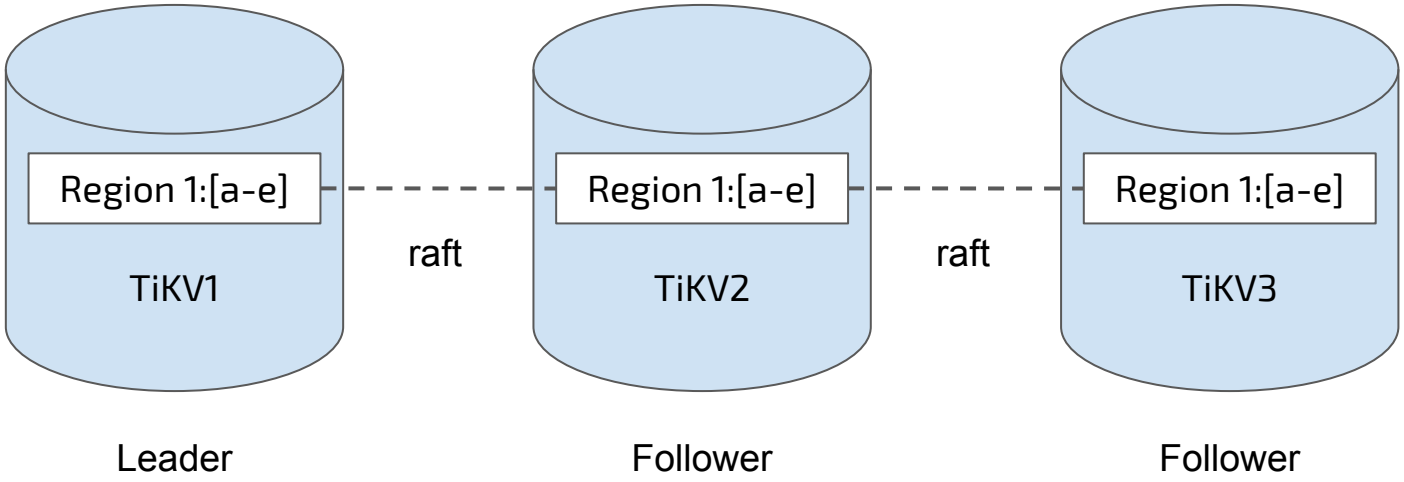
- 复制/分裂/负载均衡



PingCAP

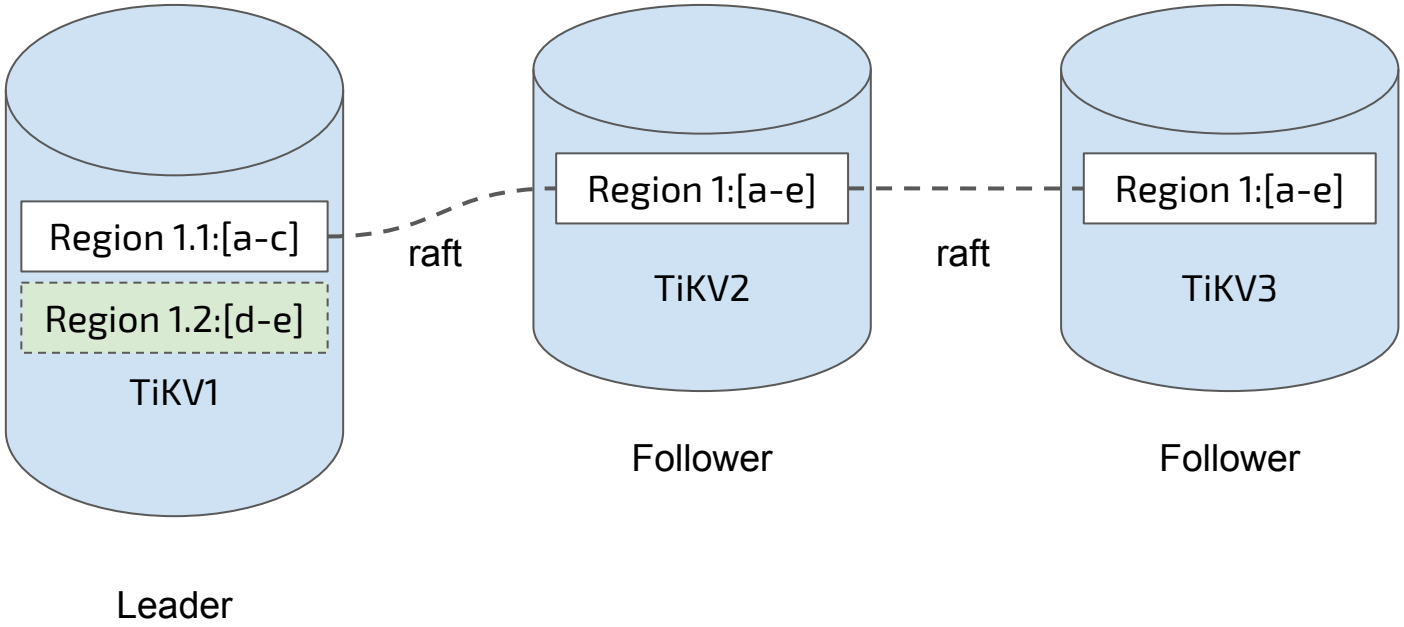
# 分裂: 1/4

Raft group



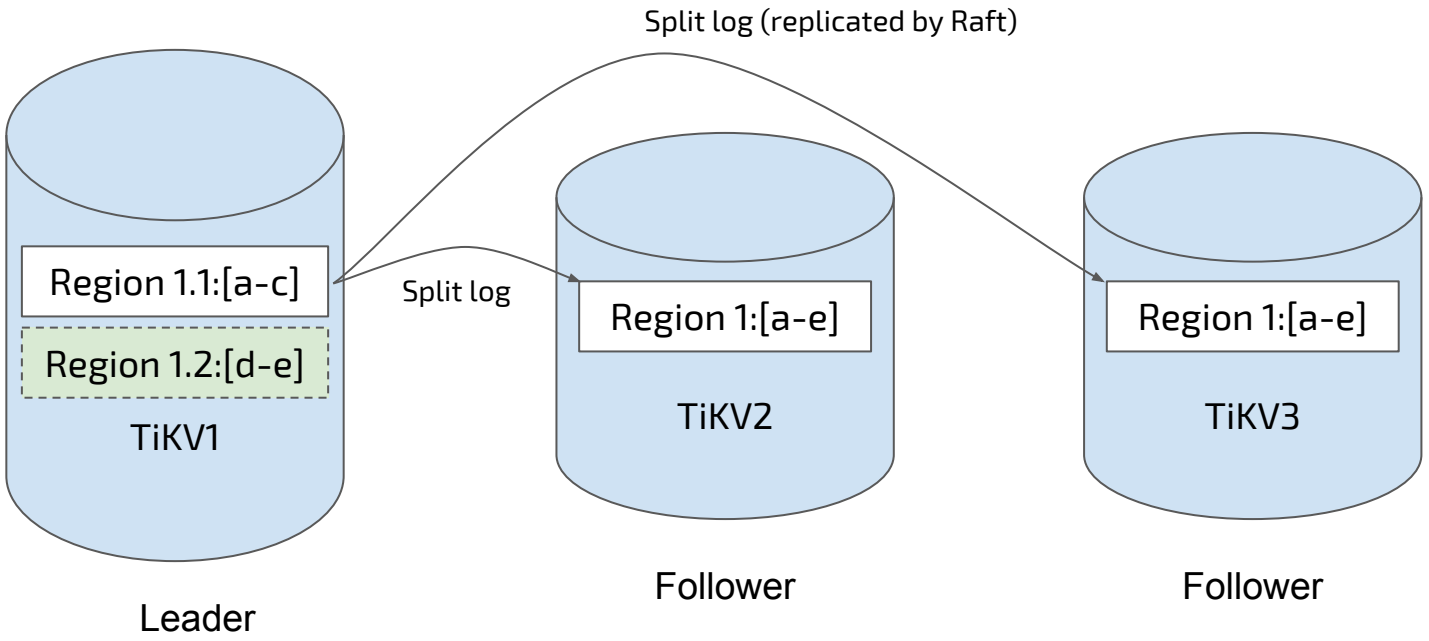
PingCAP

# 分裂: 2/4

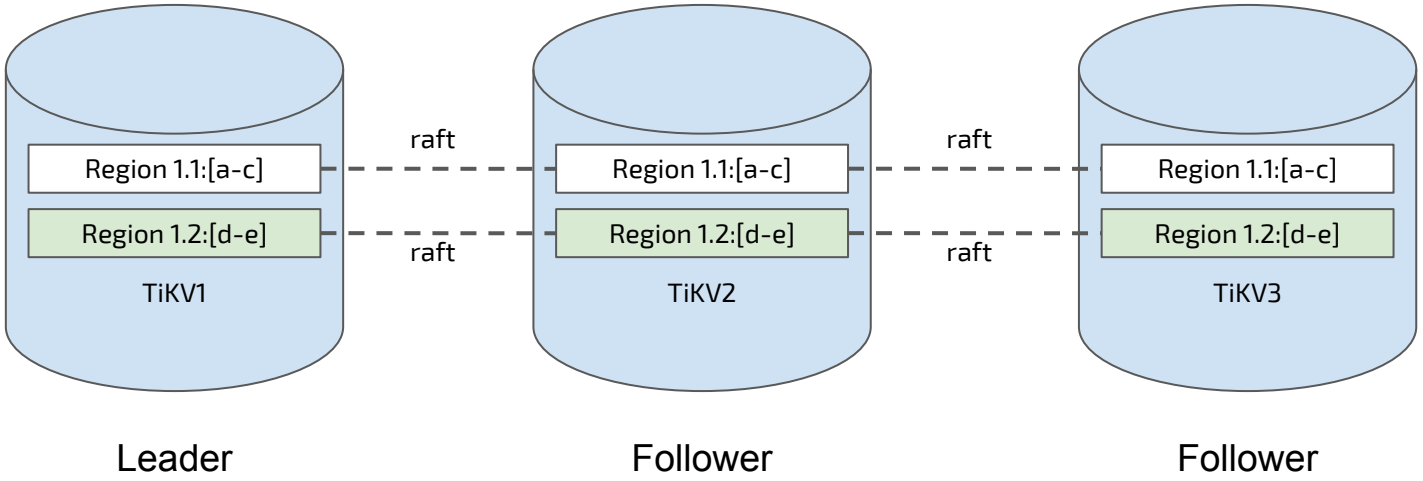


PingCAP

# 分裂: 3/4

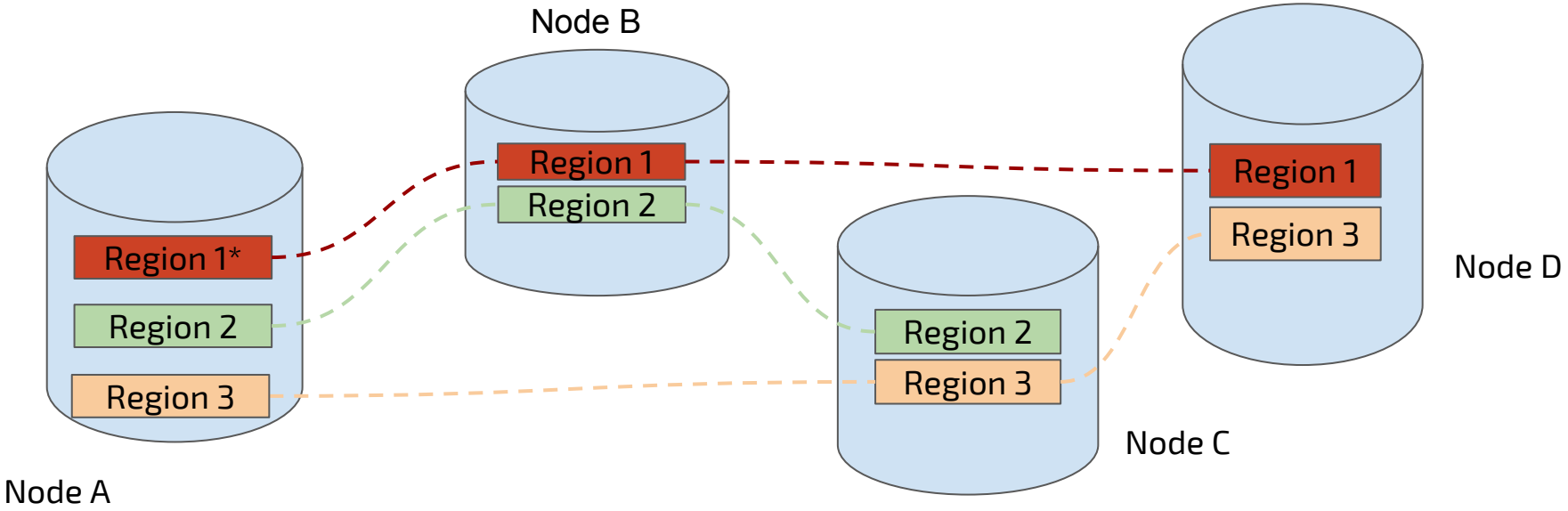


# 分裂: 4/4


Region 1.1:[a-c]
Region 1.2:[d-e]
TiKV1
raft
raft
Region 1.1:[a-c]
Region 1.2:[d-e]
TiKV2
raft
raft
Region 1.1:[a-c]
Region 1.2:[d-e]
TiKV3
Leader
Follower
Follower

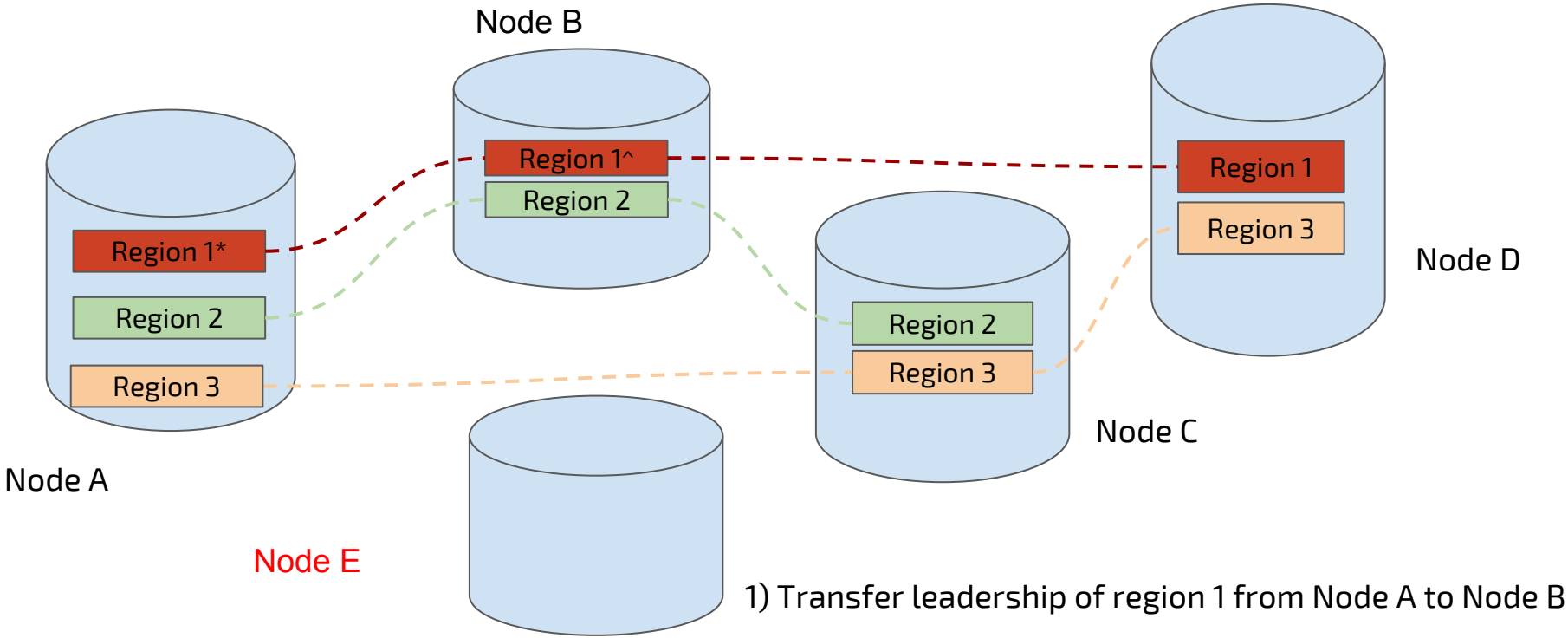
PingCAP

# Scale-out (initial state)



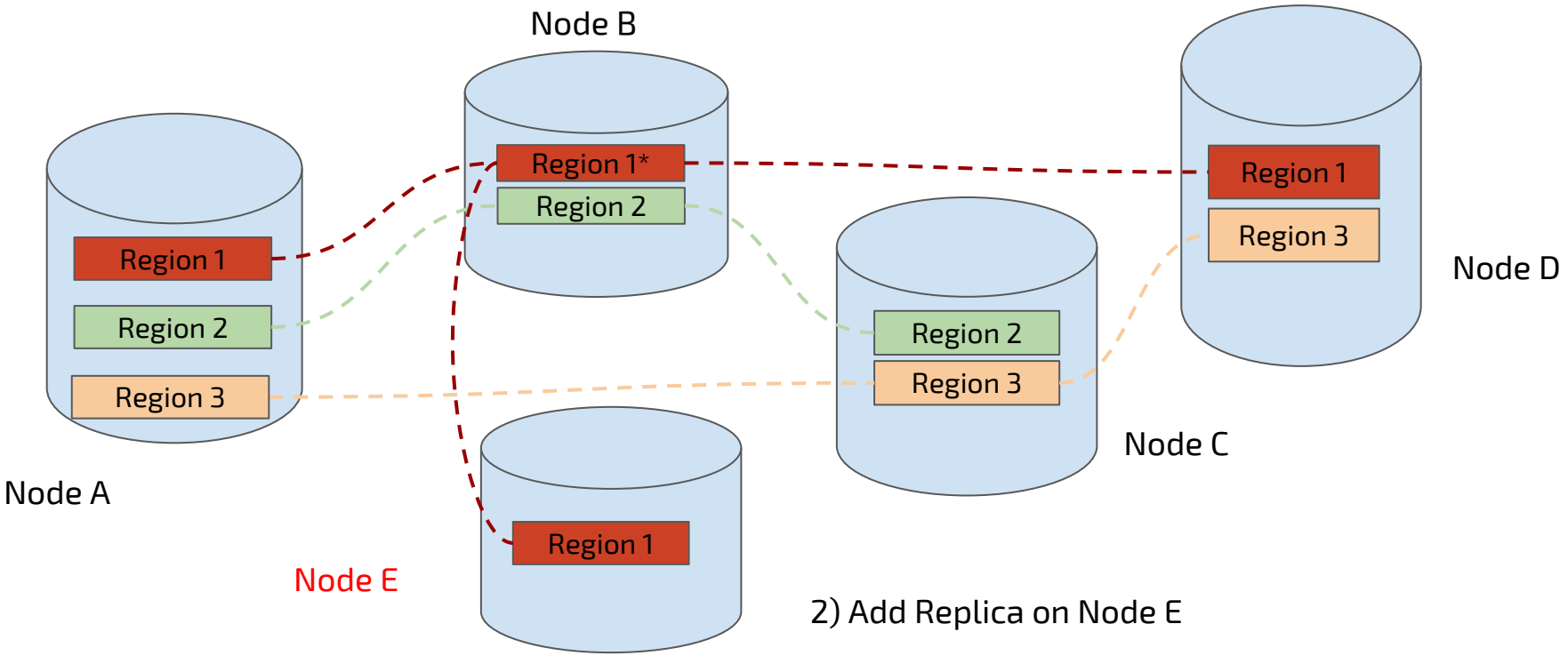
PingCAP

# Scale-out (add new node)



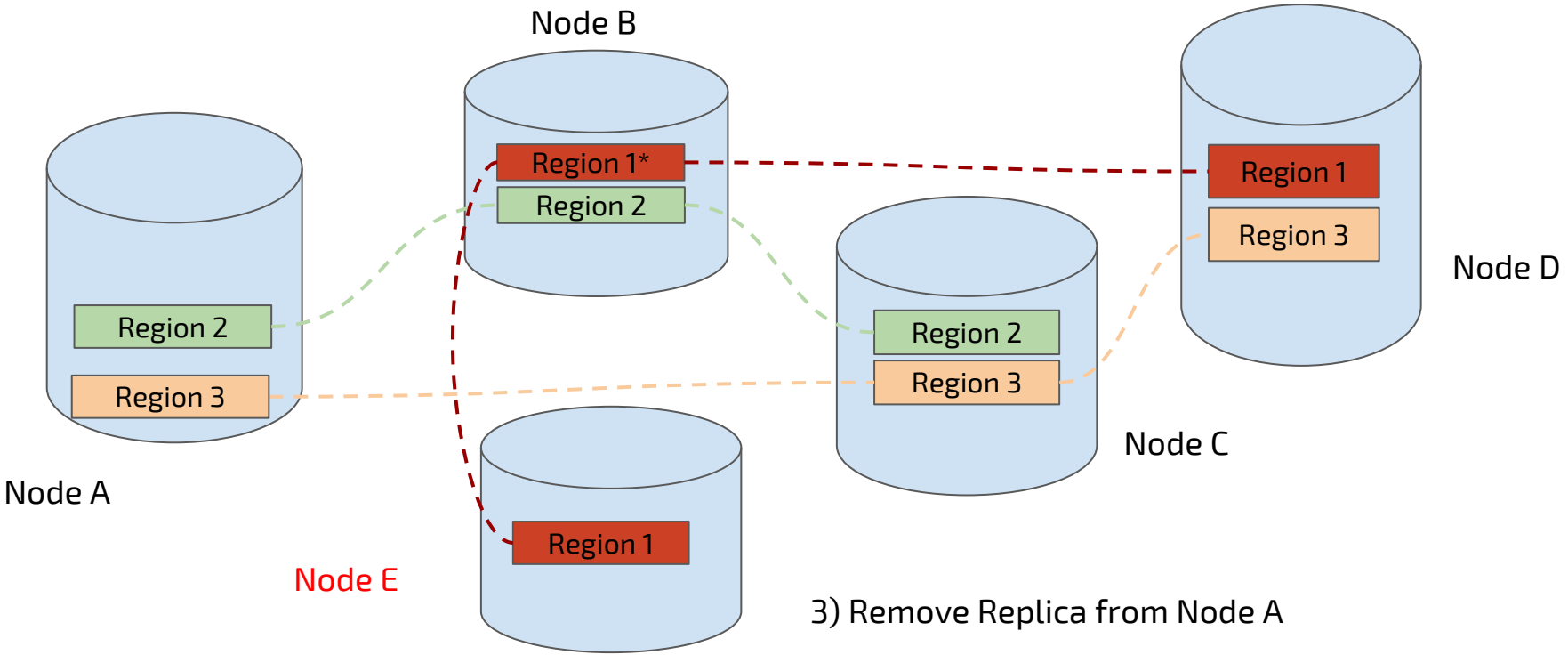
PingCAP

# Scale-out (balance)



PingCAP

# Scale-out (balance)



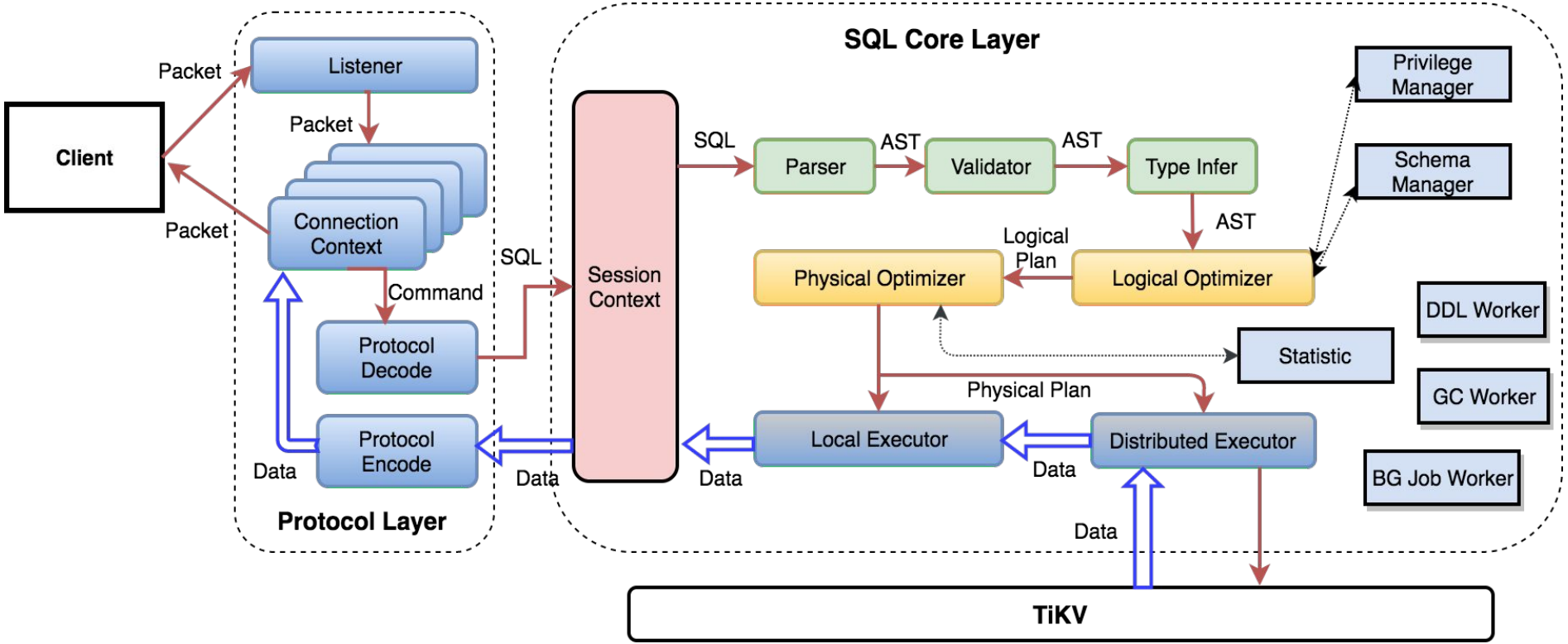
PingCAP

# 事务

- Percolator
- 去中心化的两阶段提交
  - Timestamp Allocator
- 优化的事务流程
- Repeatable Read，RU

PingCAP

# TiDB SQL 层架构



PingCAP

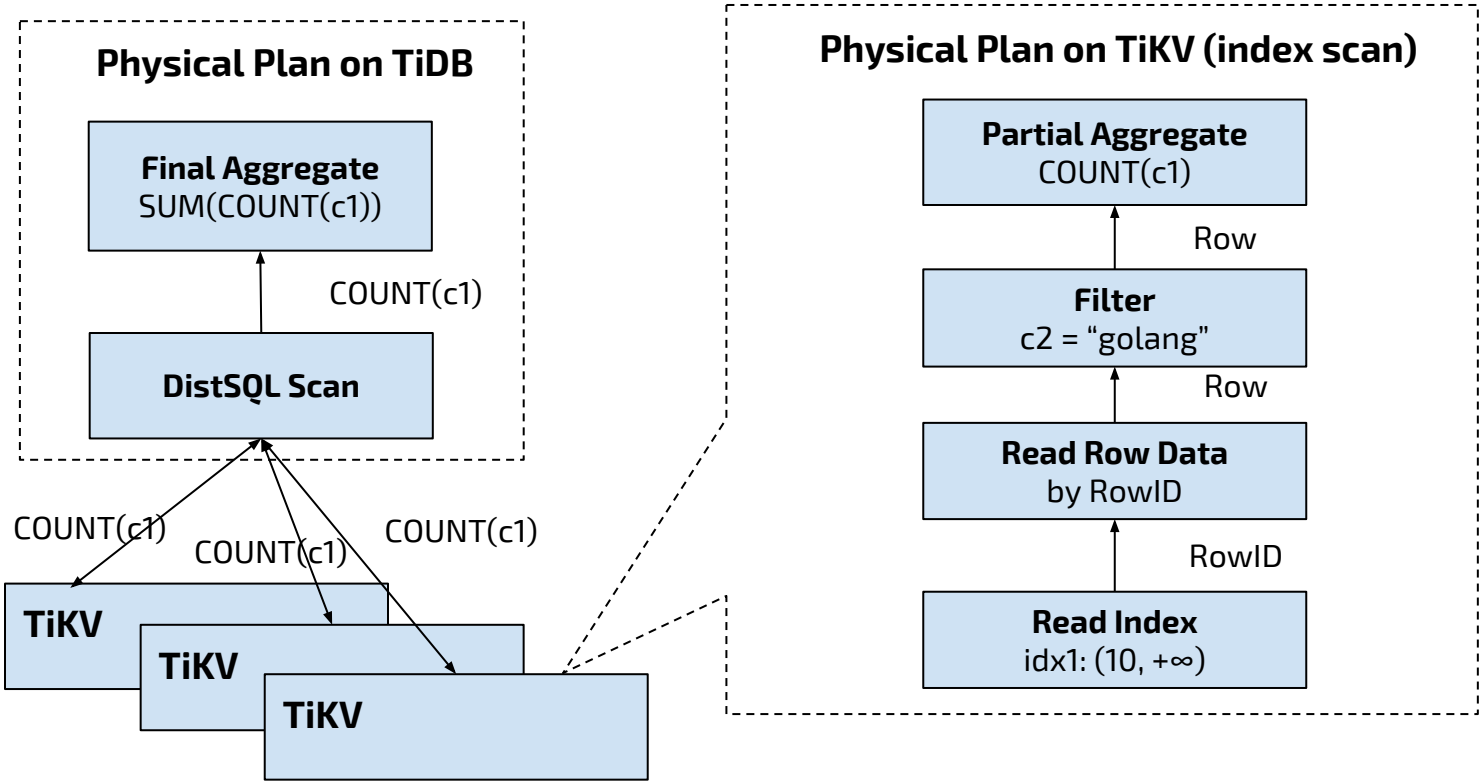
# 例子

```
CREATE TABLE t (c1 INT, c2 TEXT, KEY idx_c1(c1));

SELECT COUNT(c1) FROM t WHERE c1 > 10 AND c2 = ‘golang’;
```

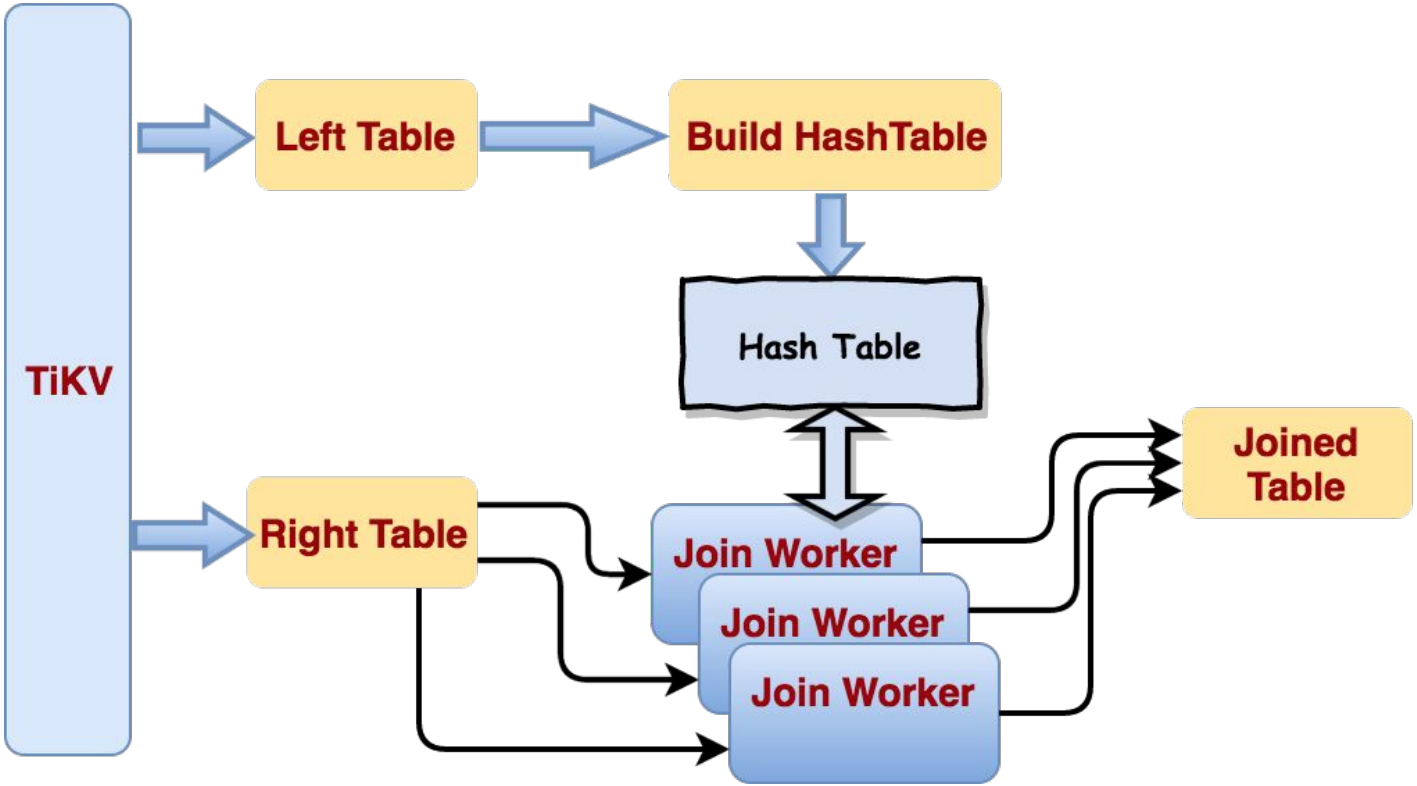
PingCAP

# 查询计划



PingCAP

# 并行 Hash Join



PingCAP

# 还有一些你看不到的东西：调度

PingCAP

# 调度的目标

- CPU
- IO
- 内存
- 磁盘使用量
- 网络流量
- Location Awareness

# 调度的方法

- PD 周期性根据 Cache 中的集群信息，生成调度计划（Operator）
- Operator 是作用于一个 Region 的一系列操作
  - Transfer Leader：将 raft group 的 leader 转让给某个 Peer
  - Add Peer：向 raft group 添加一个副本
  - Remove Peer：移除 raft group 中的一个副本
- PD 收到 Region 心跳时，将 Operator 下发
- 下次心跳时，通过新的状态判断 Operator 是否完成
- Operator 只是 PD 提供给 tikv 的建议，具体是否被执行以 tikv 为准

# 调度的策略

- LeaderBalance
  - 统计不同 Store 上的 Leader 数量
    - 从 Leader 最多的 Store 上找出一个 Leader Peer，将 leadership 移走
    - 从 Leader 最少的 Store 上找出一个 Follower Peer，将 leadership 移入
- RegionBalance
  - 统计不同 Store 上的 Peer 数量
    - 从 Peer 最多（磁盘空间最紧张）的 Store 上找出一个 Region
    - 找到 Peer 最少（磁盘空间最富余）的 Store
    - 生成 [AddPeer, RemovePeer] 或 [AddPeer, TransferLeader, RemovePeer]
- HotRegionBalance
  - 统计一段时间内的 Region 流量排行榜
  - 统计排行榜 TopN 在 Store 的分布情况
  - 生成 Operator 使之均衡

# 调度的难点

- 难以评判什么样的数据分布情况是最优解
  - 机器配置不同
  - CPU、内存、磁盘、网络多种因素相互制约
  - 用户场景多变
- 调度所依赖的集群状态不一定是最新的
- 调度本身也会带来系统负担

# 多副本管理策略

- 使用多副本保证数据安全（Data safety）
- 维持数据副本数
  - 副本数不足：AddPeer
  - 副本数过多：RemovePeer
- 优化数据的地理位置分布
  - tikv-server 按照拓扑结构打上多级 labels
  - PD 根据拓扑结构移动数据（AddPeer+RemovePeer），使得多个副本尽可能隔离

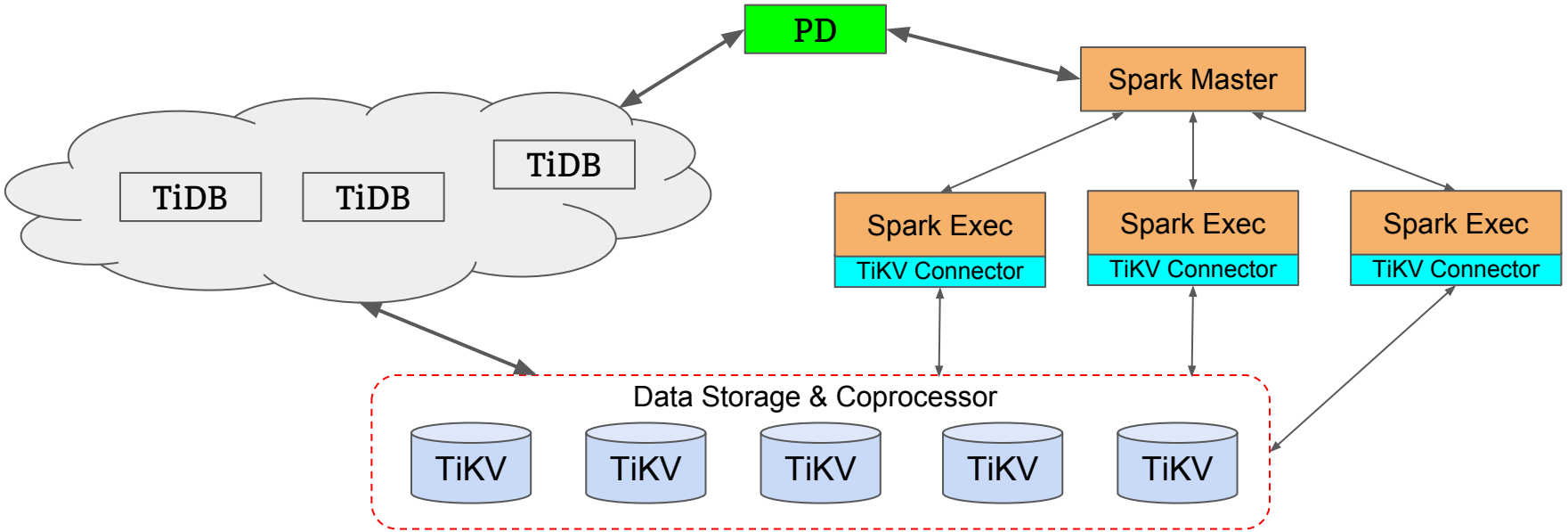
# 介绍两个有趣的项目

# Spark on TiDB

PingCAP

# TiSpark

TiDB + SparkSQL = TiSpark



PingCAP

# Features Beyond Raw Spark

- Index support
- Complex Calculation Pushdown
- CBO
  - Pick up right Access Path
  - Join Reorder

# Use Case

- Analytical / Transactional support all on one platform
  - No need for ETL
  - Real-time query with Spark
  - Possiblility for get rid of Hadoop
- Embrace Spark echo-system
  - Support of complex transformation and analytics with Scala / Python and R
  - Machine Learning Libraries
  - Spark Streaming

PingCAP

# TiDB on K8S

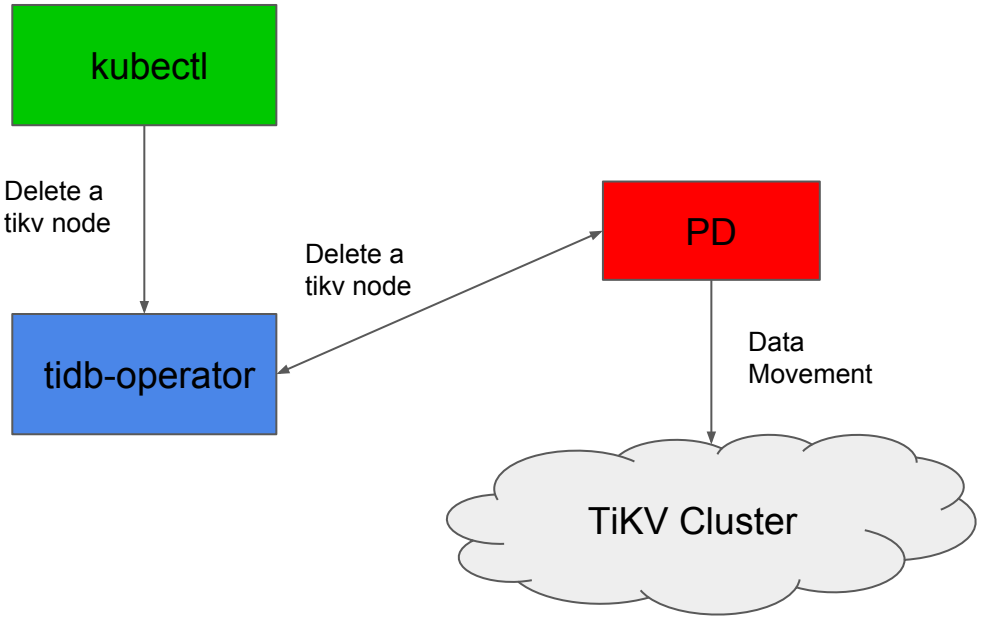
PingCAP

# TiDB with Kubernetes 1/3

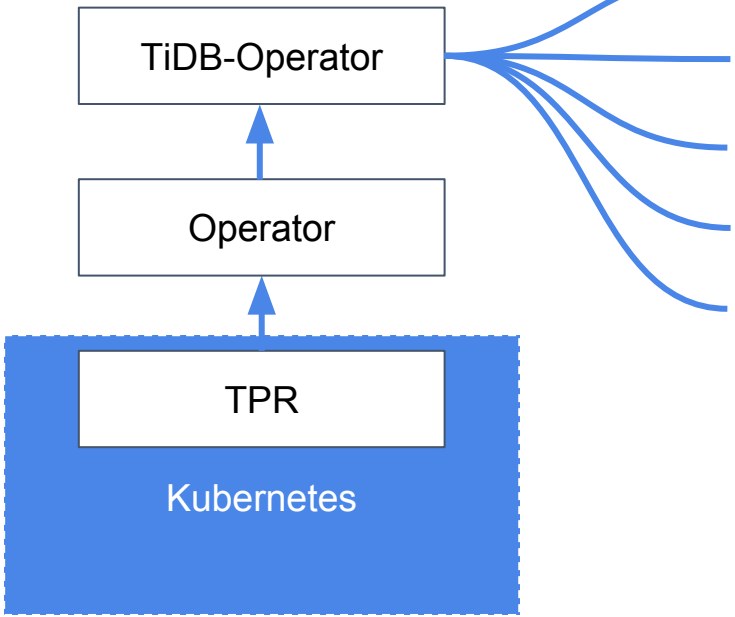
- Kubernetes 是容器编排的最佳方案
- 难点
  - Stateless is Easy, Stateful is Hard
  - Application domain knowledge
  - IO Isolation
- tidb-operator (Inspired by etcd-operator)

PingCAP

# TiDB with Kubernetes 2/3



# TiDB with Kubernetes 3/3




- Create
- Rolling update
- Scale out
- Failover
- Backup

# One more thing … …

PingCAP

# Document Store is coming

- Document store for TiDB
  - MySQL 5.7.12 Document Store
  - Json Type
  - Index for Json
  - X-Protocol
  - Mongodb Interface?

PingCAP

# 感谢

https://github.com/pingcap/tidb

https://github.com/pingcap/tikv

Contact me:

shenli@pingcap.com