

# Go 在分布式数据库中的应用

liuqi@pingcap.com

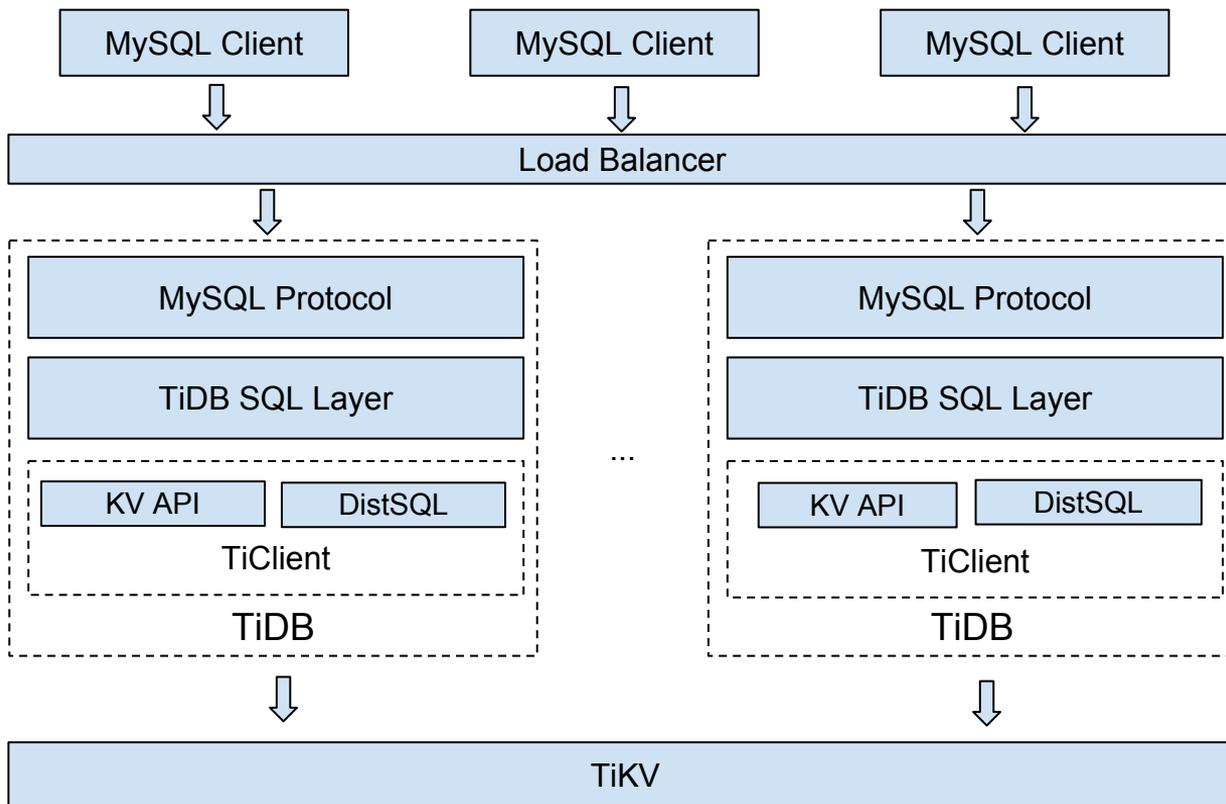
# 我与 go 的那些事

- 2011 实现了一个 bt 客户端
- 2013 用 go 京东的消息推送系统
- 2014 创建 Codis
- 2015
  - 创立 PingCAP
  - 创建 TiDB 项目
  - 开始被人黑 😂
- 2016 创建 TiKV 项目

# TiDB 简介

- 水平伸缩
  - 按需扩容/缩容
- 异步/在线 Schema 变更
  - 无论多大的表, 所有的 Schema 变更都不会阻塞现有的操作
- 分布式事务
  - 分裂, 合并, Rebalance 仍然保持一致性
- MySQL 协议兼容
  - 无缝业务迁移, 不改代码立刻获得分布式的能力
- 跨数据中心复制, 异地多活
  - Raft
  - 通常是三个数据中心, 容忍一个挂掉, 五个数据中心, 容忍两个挂掉
- Go 语言实现
  - 开发速度快

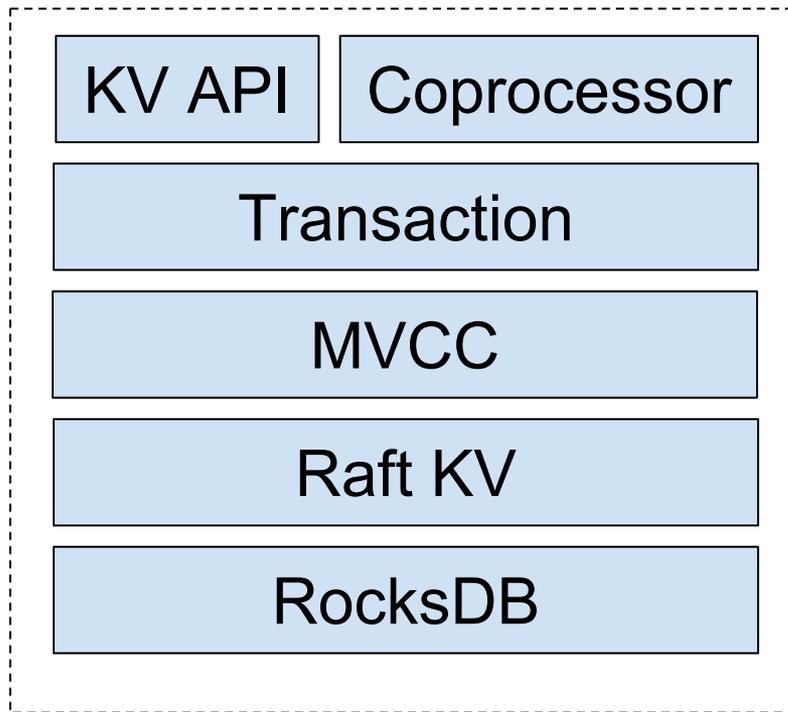
# TiDB 架构图



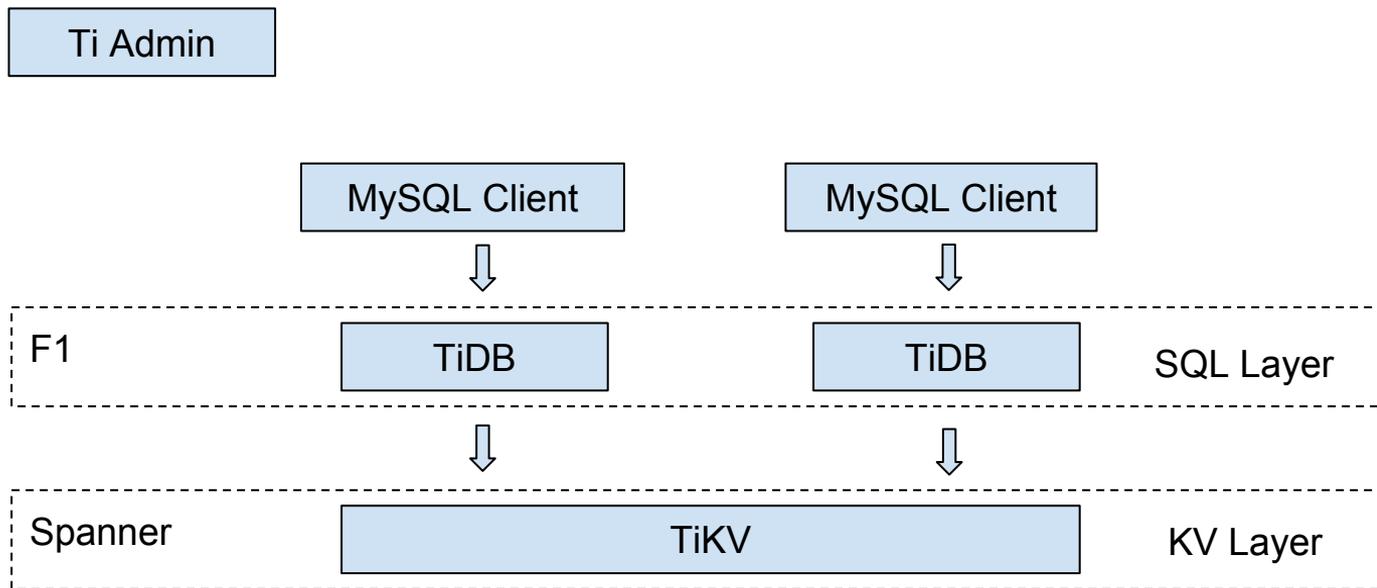
# TiKV 简介

- 高性能分布式 Key/Value
  - 提供最小化的 API (get/set/del/scan...)
- 内置 MVCC 支持
  - 通过指定版本可以访问一段时间内的任意数据
- 分布式事务
  - 基于 Google Percolator 模型
- 跨数据中心复制, 异地多活
  - Raft (Port from etcd)
  - 感谢 Diego Ongaro 创造了 Raft, 感谢 etcd 高质量的实现
  - 为什么选择 Raft
    - 相对 multi-paxos 简单, 成熟, 稳定
- Rust 语言实现
  - 运行速度极快
  - 编译巨慢

# TiKV 架构图



# Ti 全景图



# Go 如何高效跨语言通讯

- Go 和 Rust 的结合
  - 使用协议沟通
  - 高性能的序列化和反序列化
  - TCP / HTTP
  - Protobuf
    - Go 的默认实现性能太差(大量的内存分配)
    - 推荐使用 gogo 作为替代
- Why not Cgo?
  - Cgo call 开销问题
  - 写起来容易出现泄漏
  - 最重要的是太丑了!!!

# Go in TiDB

- SQL Layer
- MySQL Protocol Server
- Placement Driver (PD)
- Control Panel (TiAdmin)

# TiDB SQL Layer

- goyacc / golex
  - Go 社区的语言应用开发工具
- SQL Optimizer
  - 执行计划
- MPP framework
  - 条件下推
  - Coprocessor
  - Distributed SQL

# MySQL Protocol Server

- Session 维护
- MySQL 网络协议解析
- 海量并发连接
  - 无状态的 Server
  - Go 擅长的领域

# Placement Driver (PD)

- 数据分布信息存储
  - Region Split / Merge
  - 与 RPC Client 配合
- Auto Rebalance
- TSO
  - TimeStamp Oracle
- 上帝视角

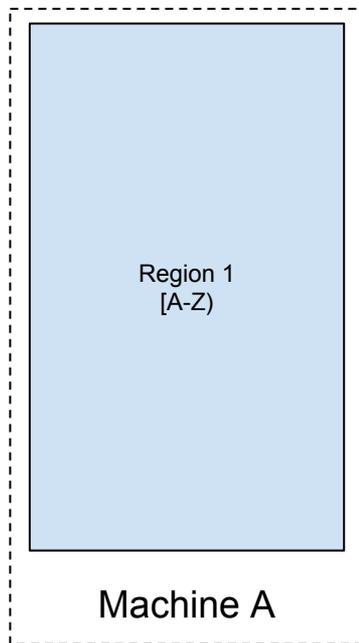
# Control Panel (TiAdmin)

- WebUI
  - Beego Rocks!
- Metric 收集 / 展示
  - TPS / QPS / IOPS ...
- 进程 / 服务管理监控
  - 组件比较多, 尽量少的人工介入
  - 进程状态监控
- etcd 维护进程和服务状态信息
  - 去单点
  - 持久化

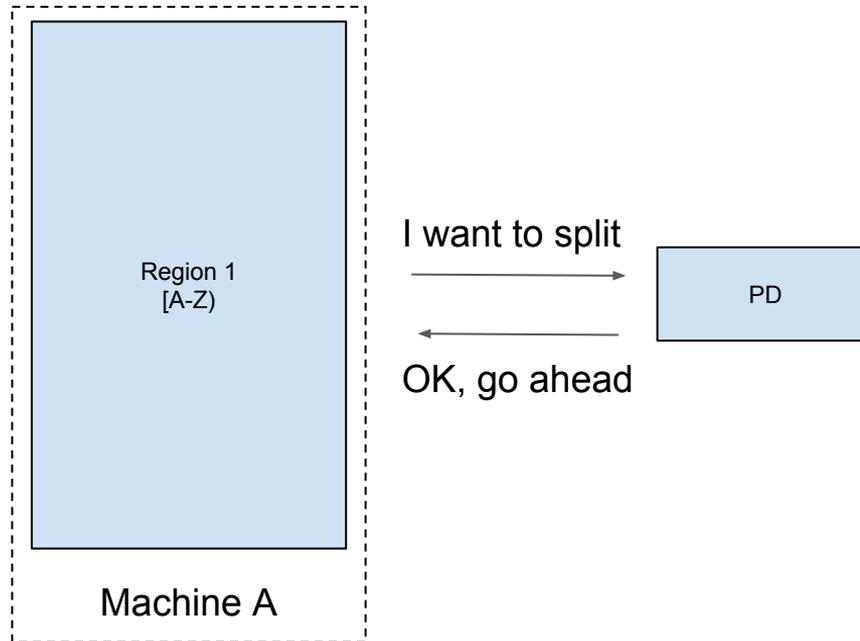
# 如何实现动态伸缩

1. 弹性扩展
  - a. Split
  - b. Reblance
2. 弹性收缩
  - a. Reblance
  - b. Merge

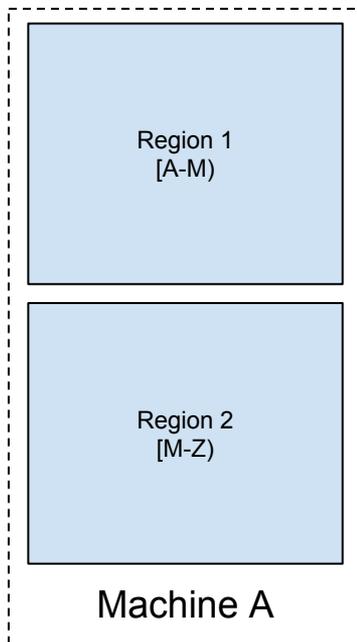
# Region Split



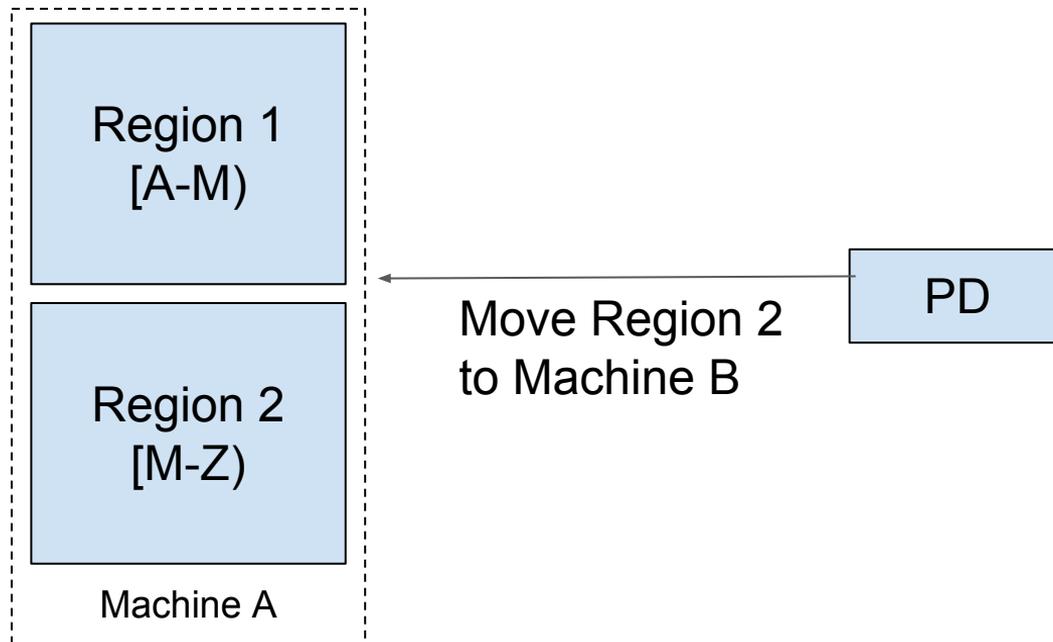
# Region Split



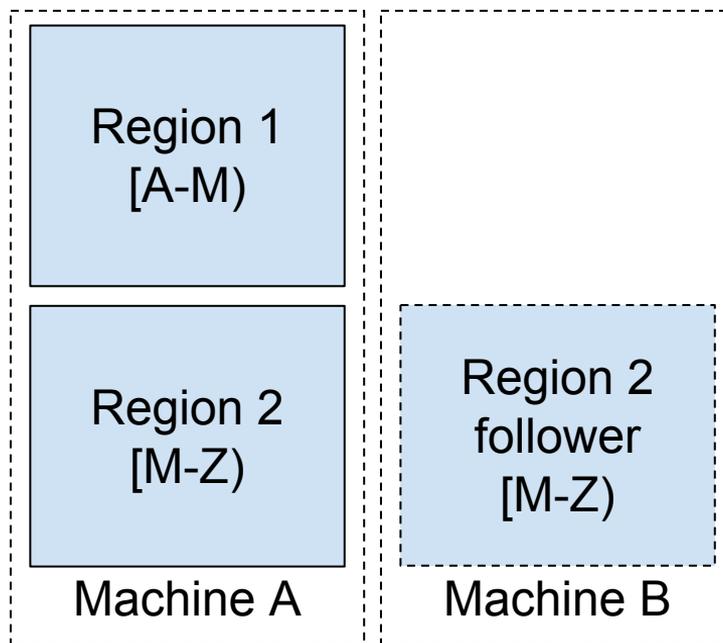
# Region Split



# Region Move

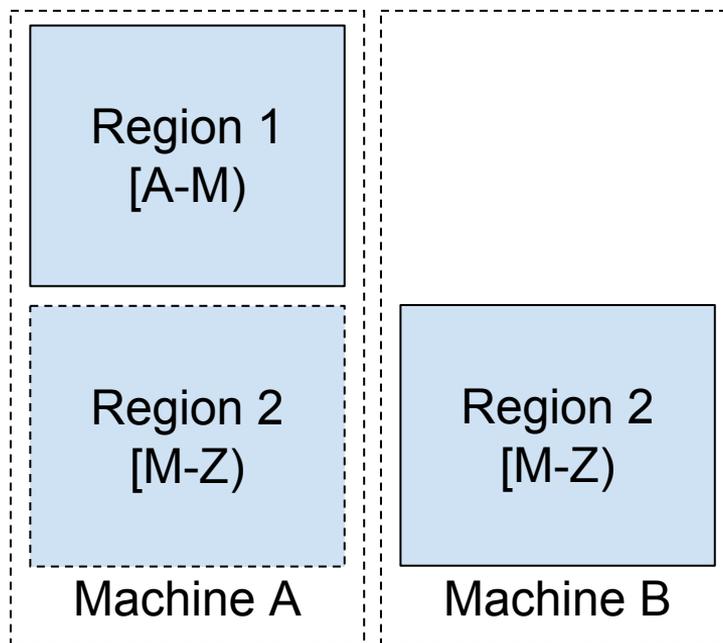


## Region Move (Balance)



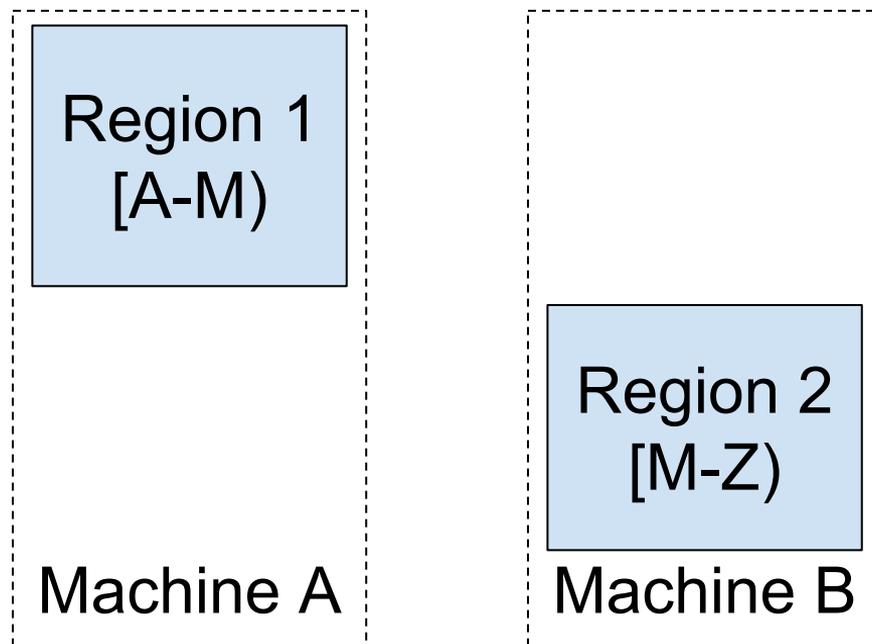
Using Raft Configuration Change

## Region Move (Balance)



Using Raft Configuration Change

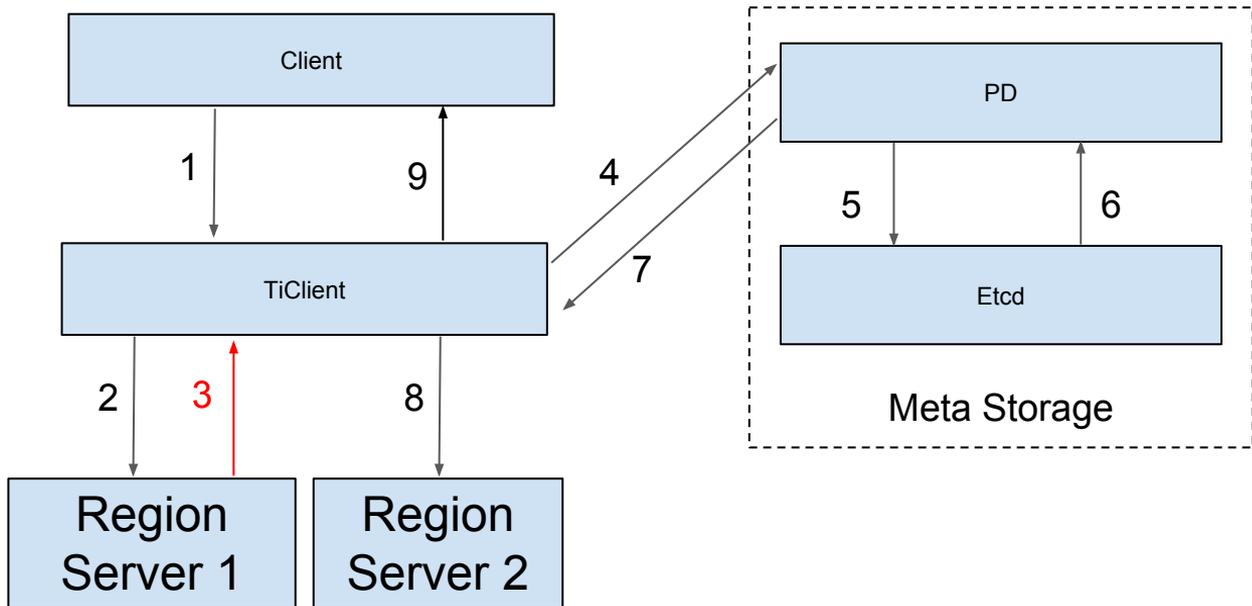
## Region Move (Balance)



# 如何进行数据路由

- TiClient
- Placement Driver
- etcd

# 如何进行数据路由



→  
err: not in my region



## 总结 go 语言一些心得

- go 是一门工程性，实践性的语言，无需特殊技巧
- 编译速度快，至少 1.4 以前很快 😂
- 很容易做并发，streaming, worker pool, connection pool
- 超爱 pprof
- 无依赖部署
- 社区强大，各种库比较齐备
- 和生态易于整合 (etcd, k8s)
- 再也没有争论了 (gofmt, golint...)

谢谢大家

<https://github.com/pingcap/tidb>

We're hiring.