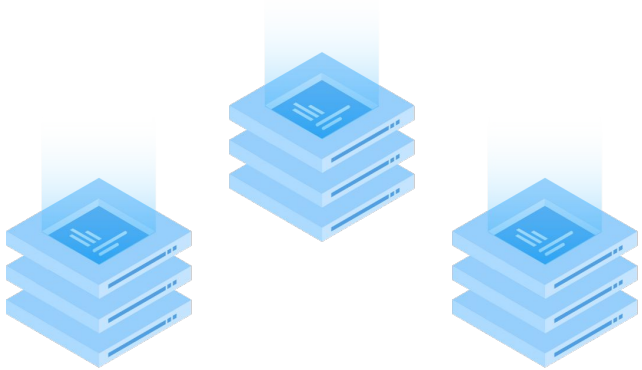
High Performance TiDB-Lesson 1

PingCAP TiDB



Before we begin

- Context: (speak it out)
- Goal (speak it out)
- Outline:
  - Database evolution
  - TiDB introduction
  - TiDB architecture
  - ...
- Lab requirements (if needed):

PingCAP

High Performance TiDB

课程目标

课程受众：

- 具备代码能力、热爱开源、热爱源码的一线研发人员

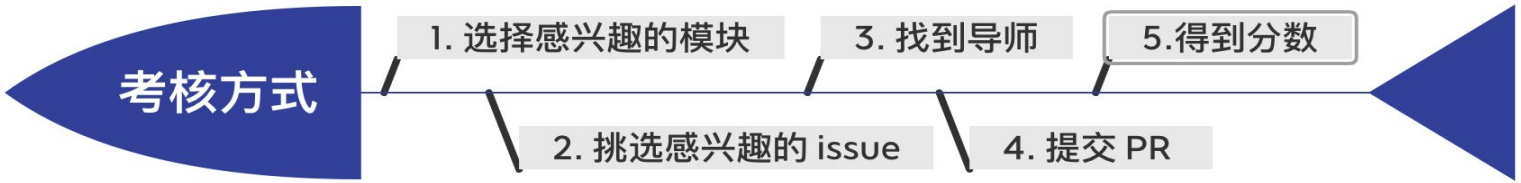
课程目标：

- 掌握 TiDB 内核实现原理
- 精通一个或多个模块
- 能够对一个或者多个模块给出源码级别的调优
- 具备成为 TiDB 社区 committer 的潜力

课程考核

考核目标：深度参与 TiDB 社区源码级别的性能调优，在导师指导下，协作或独立完成一定难度与数量的任务，对 TiDB 的性能作出实质性的贡献。

考核方式（总分积累至 3000 分即完成课程考核）：



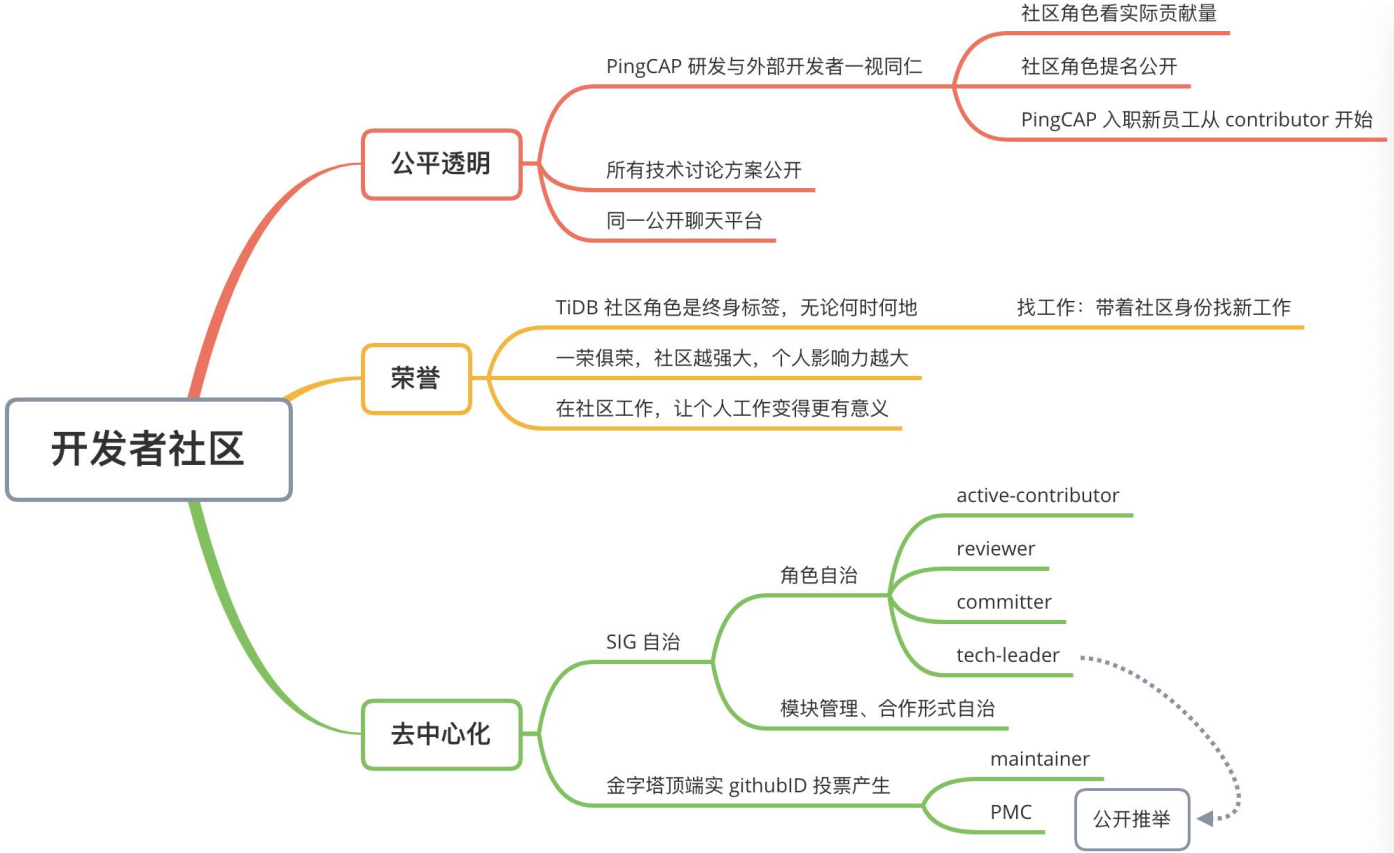
课程组织方式

- 前一周周末：主办方课程预习材料发放
- 周三以前：预习当周基础知识
- 周三晚 6:00：主办方上线课程视频，视频包括：
  - 本周课程重点知识
  - 本周课程相关 issue 或 文档类作业发放
- 周四以后：
  - 文档类作业当周周日之前需要提交，选择认领，人人可以认领。
  - issue 可以根据自己的兴趣点，选择认领，完成 PR 开发，合并到 master 分支，获取分数

PingCAP

TiDB community

Grow with TiDB community



PingCAP

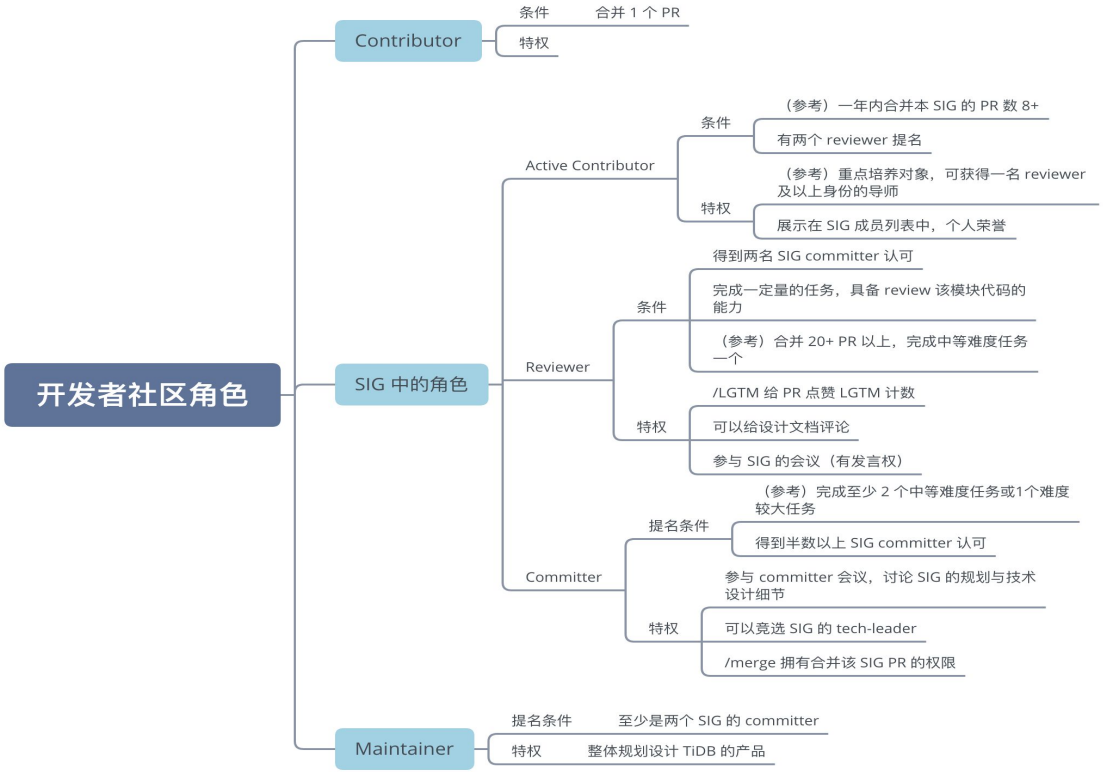
专项兴趣小组 (SIG : Special Interest Groups)

https://github.com/pingcap/community/tree/master/special-interest-groups

- TiDB 社区根据技术模块主要由 SIG（专项兴趣小组）组织而成。
- SIG 成员由多个公司和组织的成员组成，这些人共同维护某个特定模块（如 transaction），推进该模块调研与开发。
- 对于 TiDB 项目来说，任意模块都应该有其对应的 SIG。

PingCAP

Grow in TiDB community



PingCAP

PingCAP

TiDB Architecture Overview

Before we begin

- Context: TiDB Architecure
- Goal : Introduce the architecure of TiDB
- Outline:
  - Overview
  - TiKV - Storage Engine
  - TiDB - SQL Engine
  - Tools

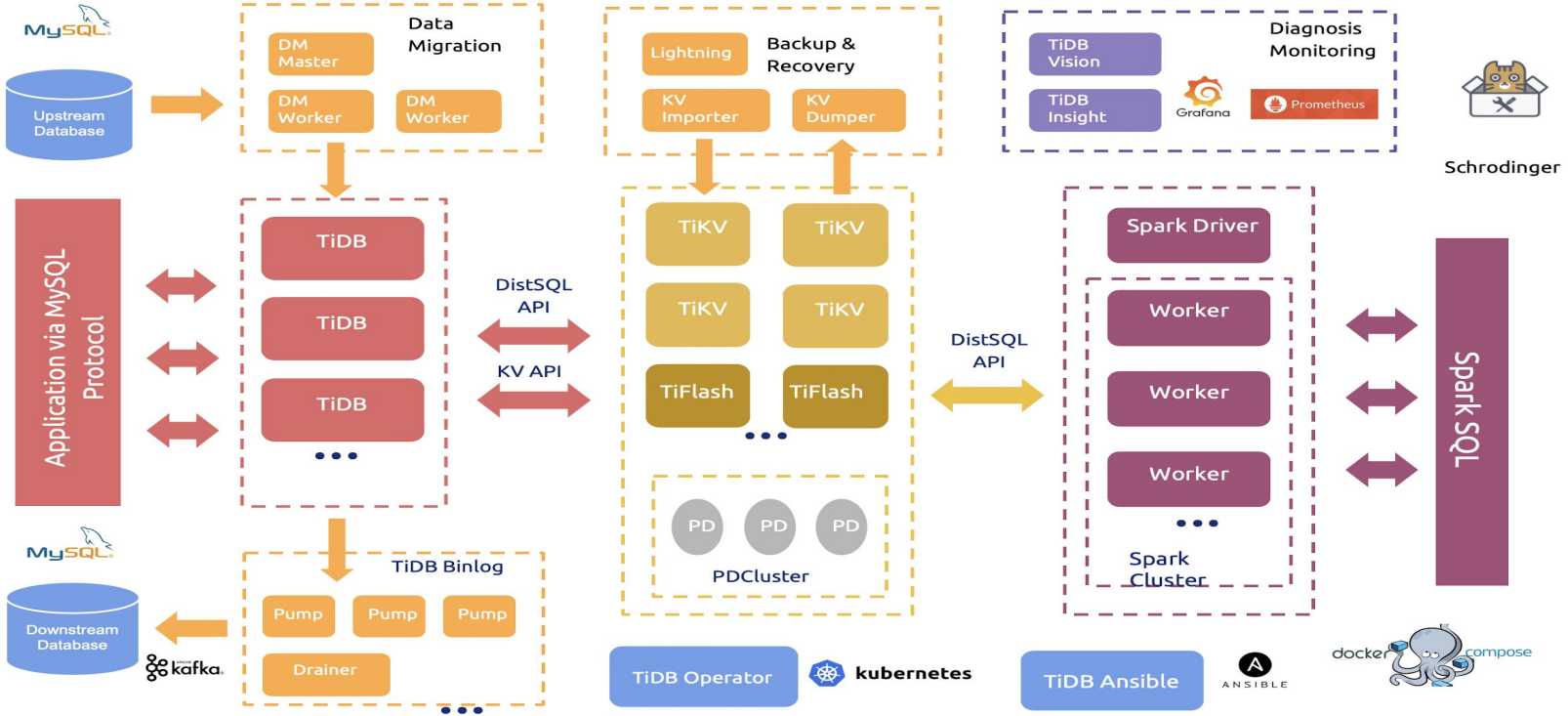
What is TiDB?

- An **open-source** distributed **NewSQL** database for hybrid transactional and analytical processing (**HTAP**) which speaks MySQL procotol
- Key Features
  - **Horizontal Scalability**
    - Scale-out transparent to the application layer
  - **High Availability**
    - Self-healing thanks to modern consensus algorithm like Raft/Multi-Paxos
  - **ACID** Compliance
    - Distributed transaction
  - **MySQL** Protocol & Dialect
    - MySQL syntax & ecosystem tools

TiDB
A Distributed SQL Database

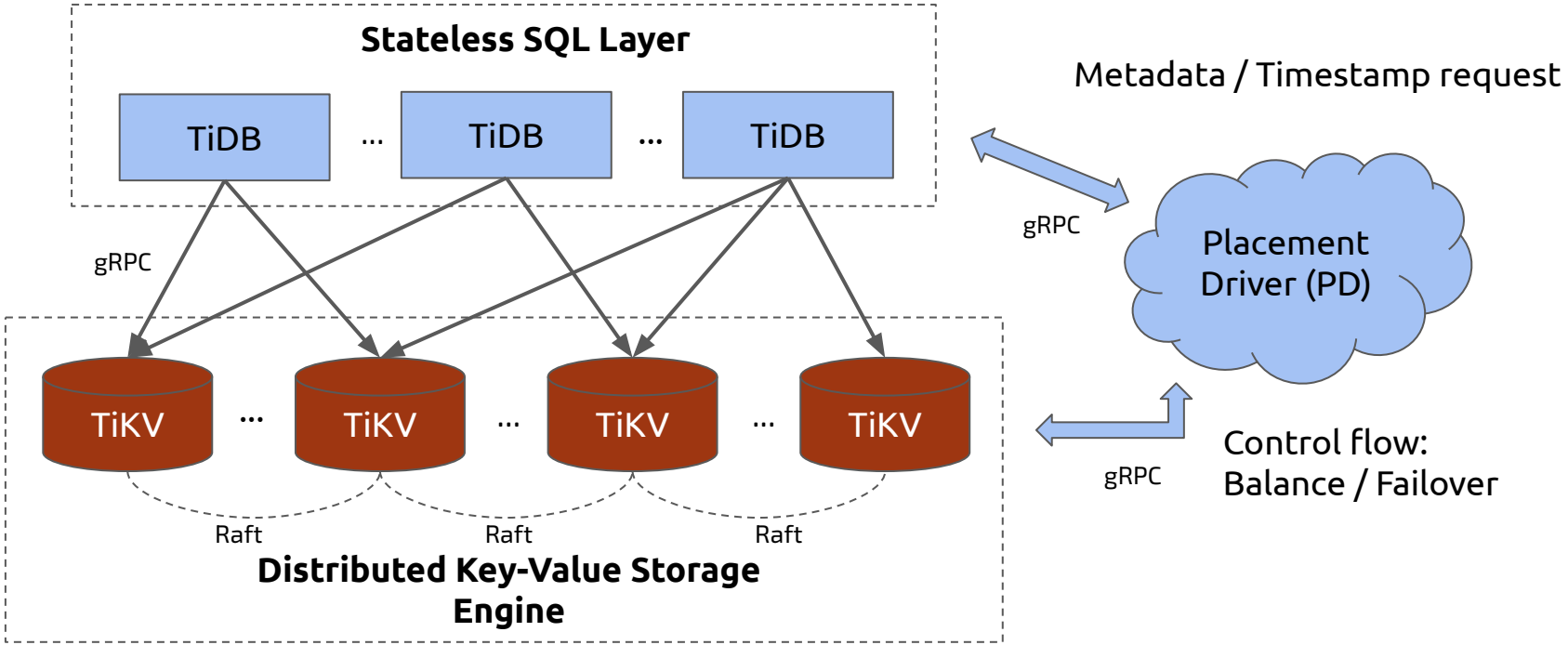
TiDB is **NOT** a MySQL middleware or a MySQL fork

TiDB Architecture(1/2)

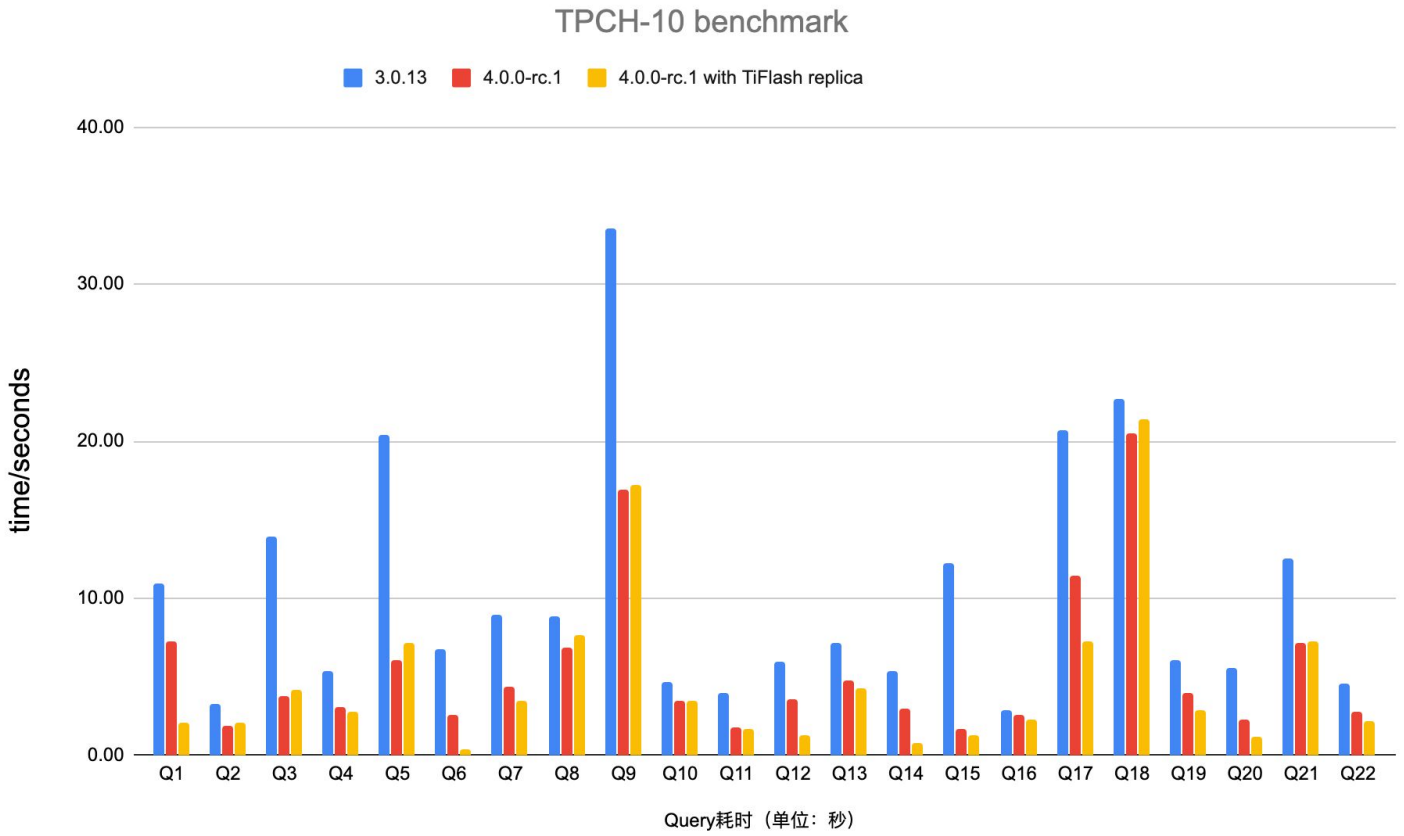


PingCAP

TiDB Architecture(2/2)



Performance-TPCH



PingCAP

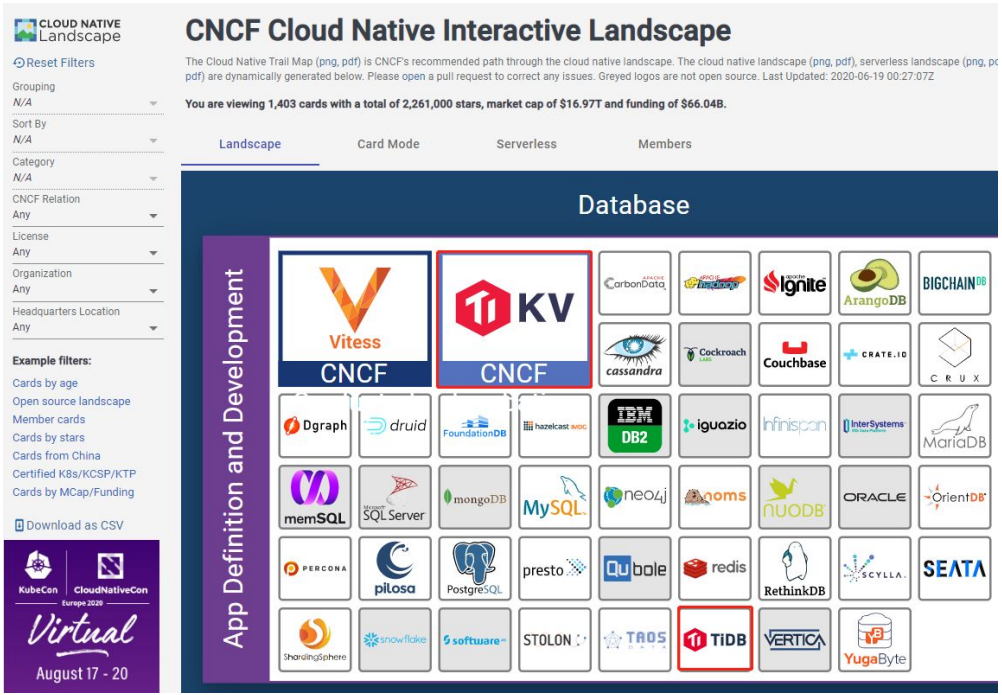
Open Source Community

TiDB

- **24K+** Stars
- **430+** Contributors

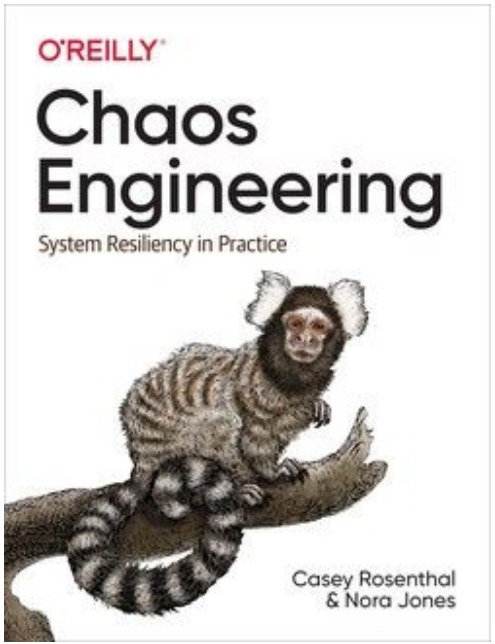
TiKV CLOUD NATIVE COMPUTING FOUNDATION

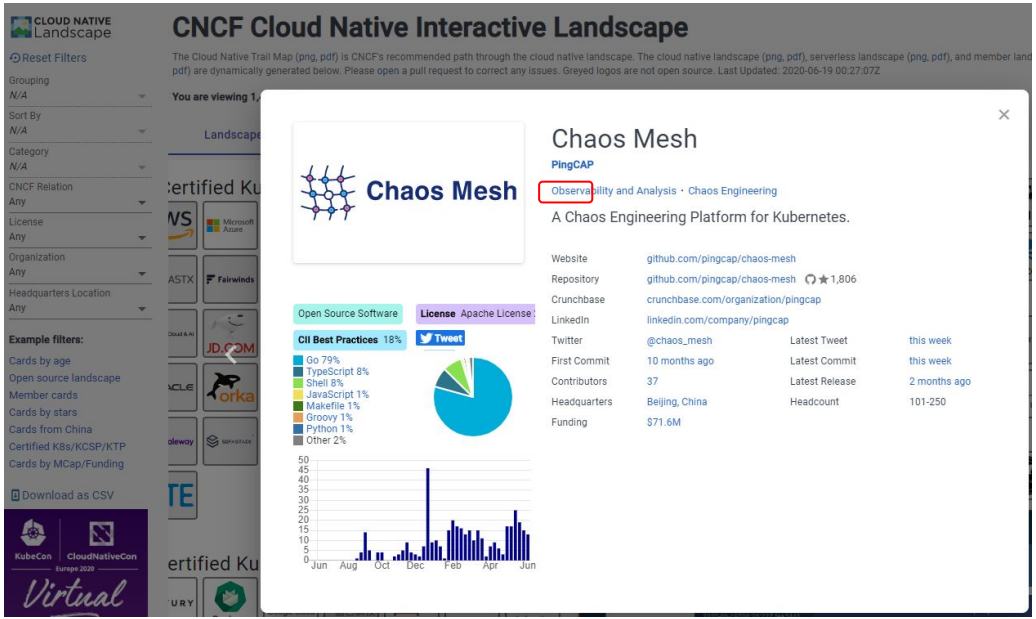
- **7K+** Stars
- **240+** Contributors



PingCAP

Chaos Engineering Practice in TiDB





PingCAP engineer contributed the *Chaos Engineering on a Database* chapter to O'REILLY's Book on Chaos Engineering

PingCAP open sourced the Chaos Mesh project on Dec. 31 2019 and is donating it to CNCF Sandbox to be the cloud-native Chaos Engineering platform on Kubernetes

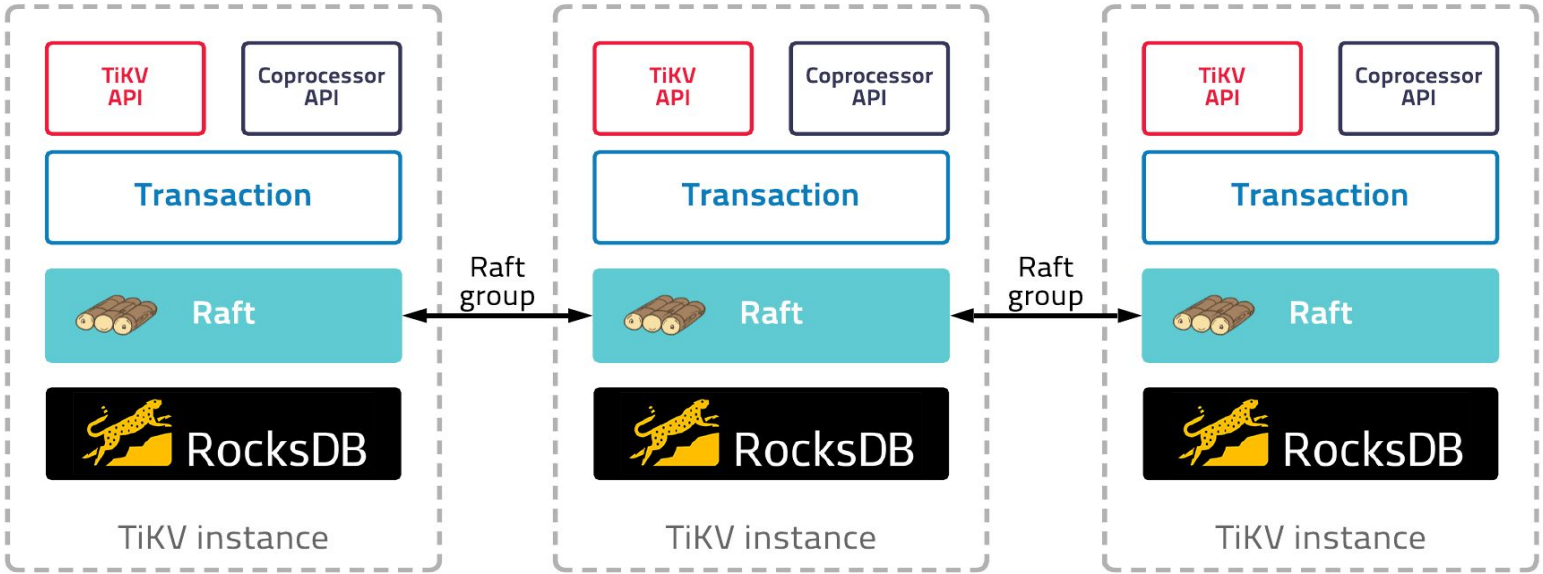
PingCAP

TiKV: the storage engine

What is TiKV

- A **distributed transactional key-value** database originally created by PingCAP as the underlying storage engine for TiDB
- based on the design of **Google Spanner** and **HBase**, but simpler to manage and without dependencies on any distributed file system
- a **CNCF** incubating project with 7.6 K GitHub Stars and 249 Contributors

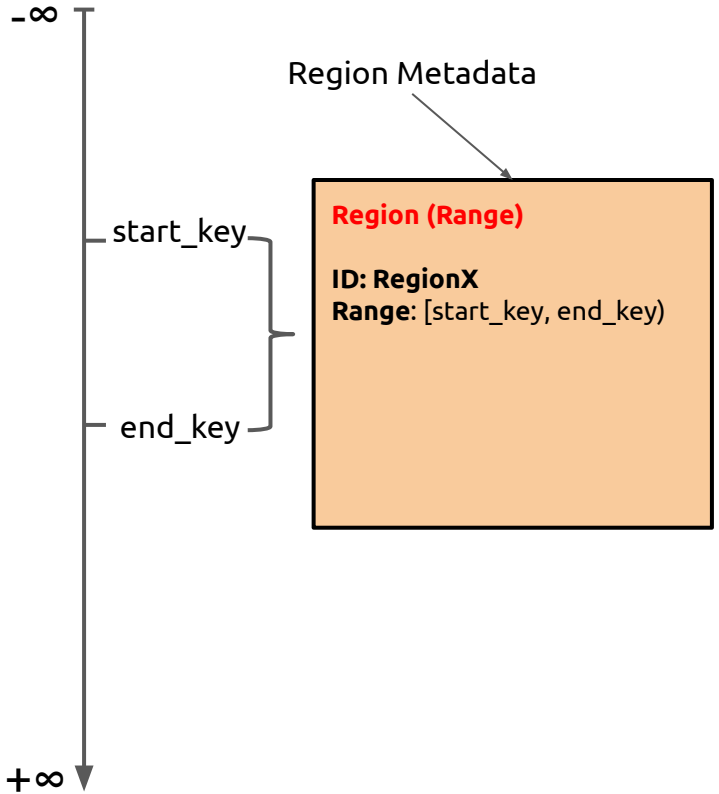
TiKV Architecture



Logical View of TiKV

- A giant ***Map***
  - Sorted Key-Value Map
  - Both keys and values are byte arrays
  - Keys are sorted by byte order
- Key space is divided into pieces
  - Data divided into chunks called "**Regions**"

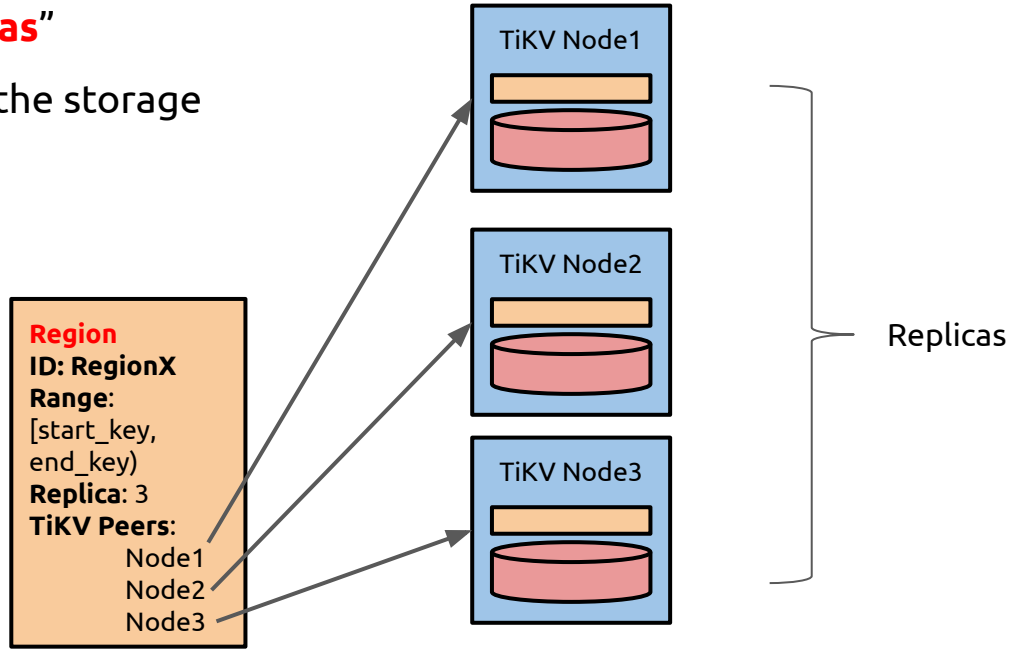
Terminology:
- **Region**



Physical View of TiKV

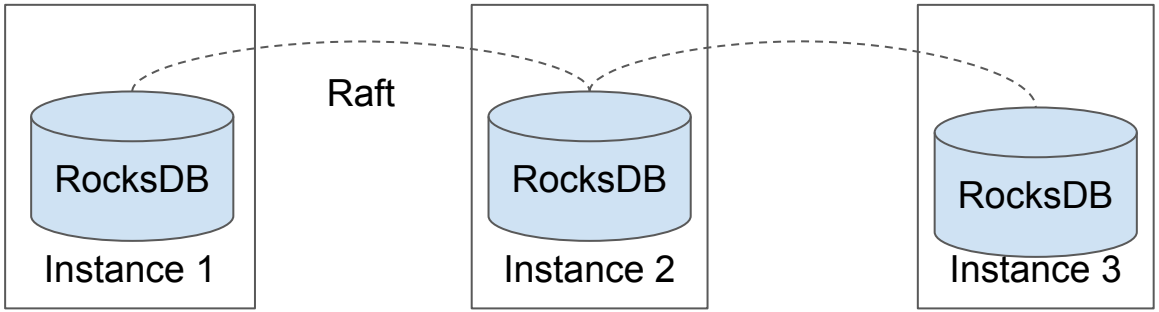
- A distributed storage engine
  - Each ***region*** has multiple "**replicas**"
  - Replicas are distributed among the storage instances via **Raft algorithm**

Terminology:
- **Replica**

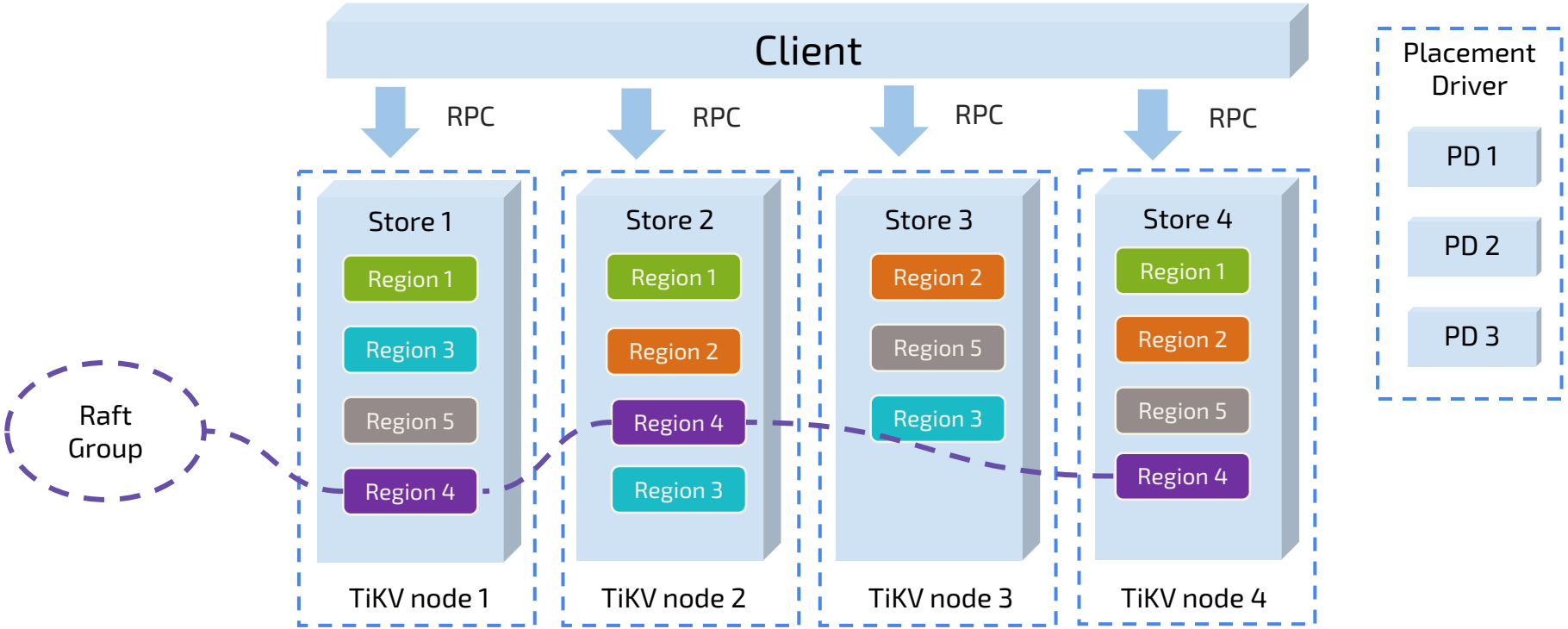


Raft

- Provide **strong consistency** guarantees over replicas
- Roles: **Leader,** Candidate, **Follower** and Learner
- Read/Write only go for **leader** and replicated to followers and learners
- Leader Election
- Membership Change

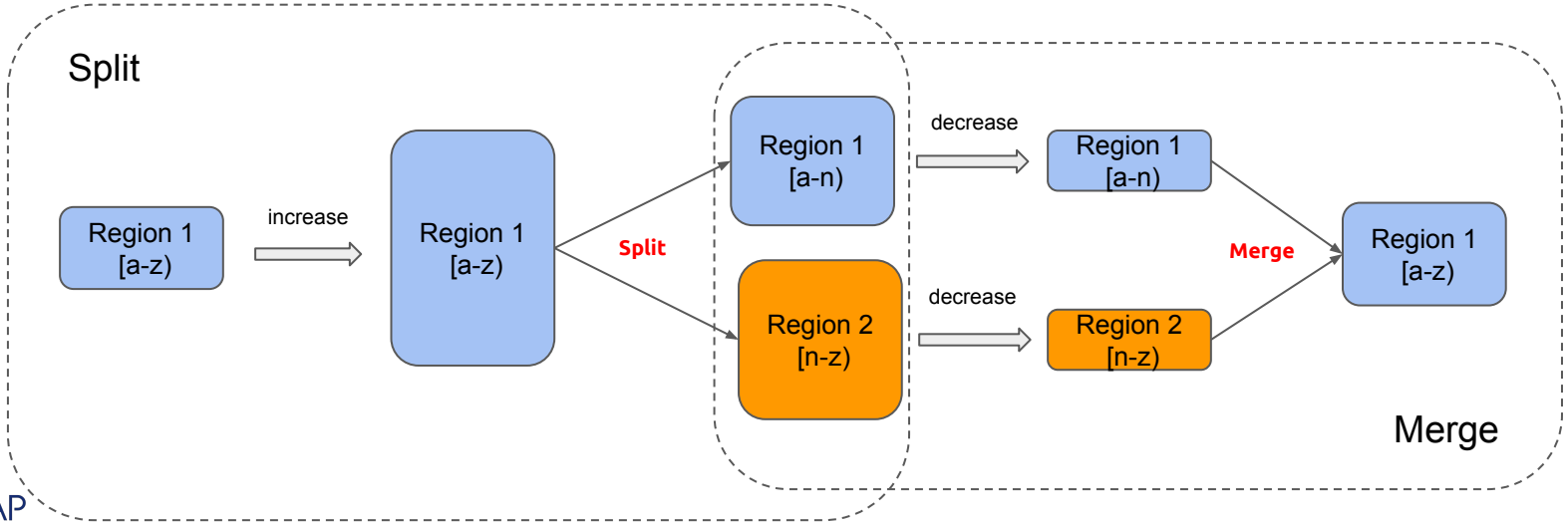


Raft Group

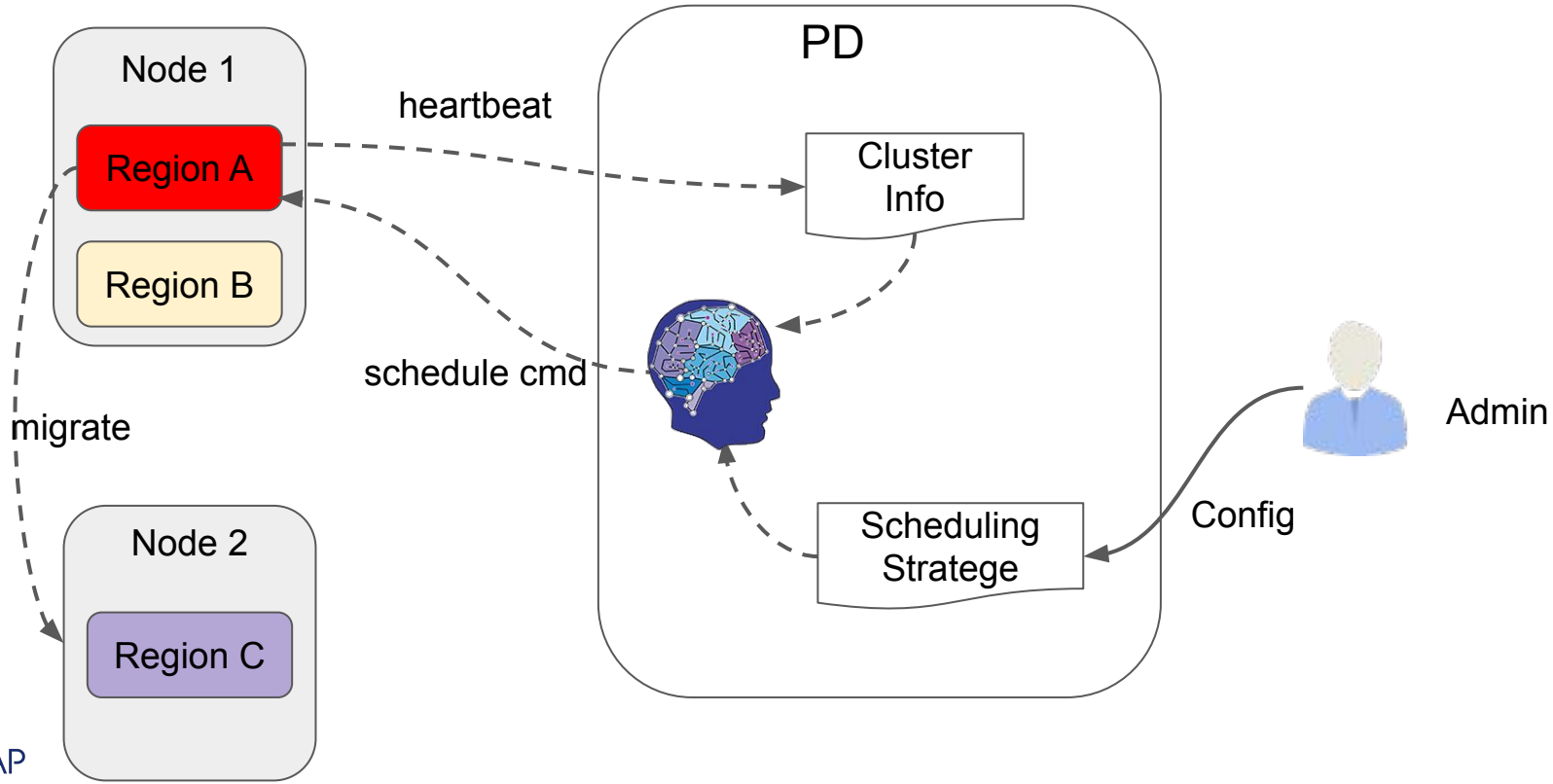

Client
RPC
RPC
RPC
RPC
Placement Driver
PD 1
PD 2
PD 3
Store 1
Region 1
Region 3
Region 5
Region 4
Store 2
Region 1
Region 2
Region 4
Region 3
Store 3
Region 2
Region 5
Region 3
Store 4
Region 1
Region 2
Region 5
Region 4
Raft Group
TiKV node 1
TiKV node 2
TiKV node 3
TiKV node 4

Dynamic Split/Merge

- Split/Merge based on **data size**
  - 96 MB by default to split
  - 20 MB by default to merge



PD : The manager of the raft group



Distributed Transactions

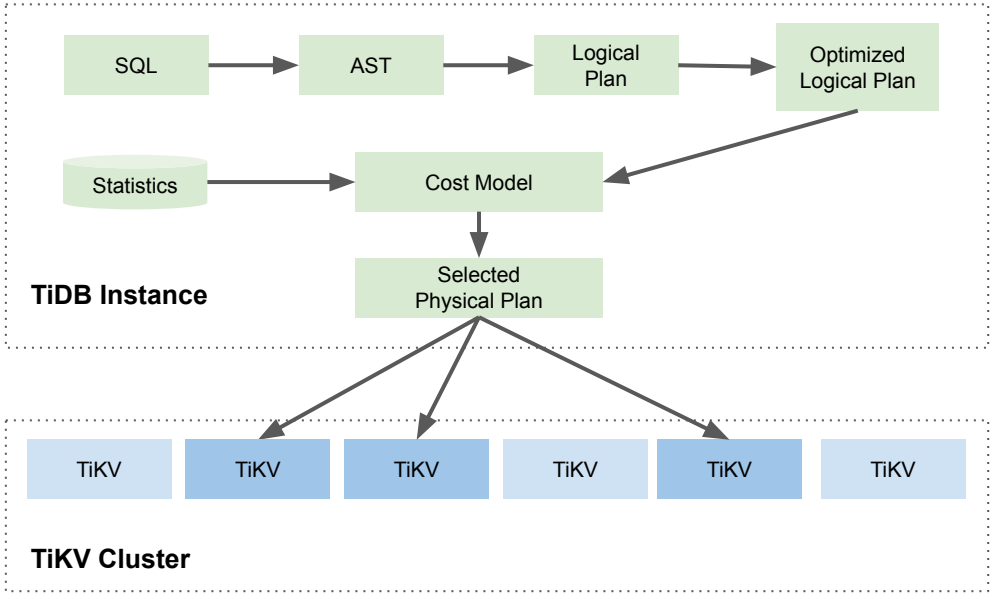
- TiKV exposes a full transactional K/V API
- Based on **Google Percolator** with optimizations
- An ‘almost’ decentralized 2-phase commit algorithm
  - Timestamp Allocator (PD)
  - ~4M timestamps allocations per seconds
- **Pessimistic Lock**(>= 3.0.8version, default), Optimistic Lock
- Isolation Level
  - **Snapshot Isolation**
  - **Read Committed**

PingCAP

TiDB: the SQL engine

SQL Layer (tidb-server)

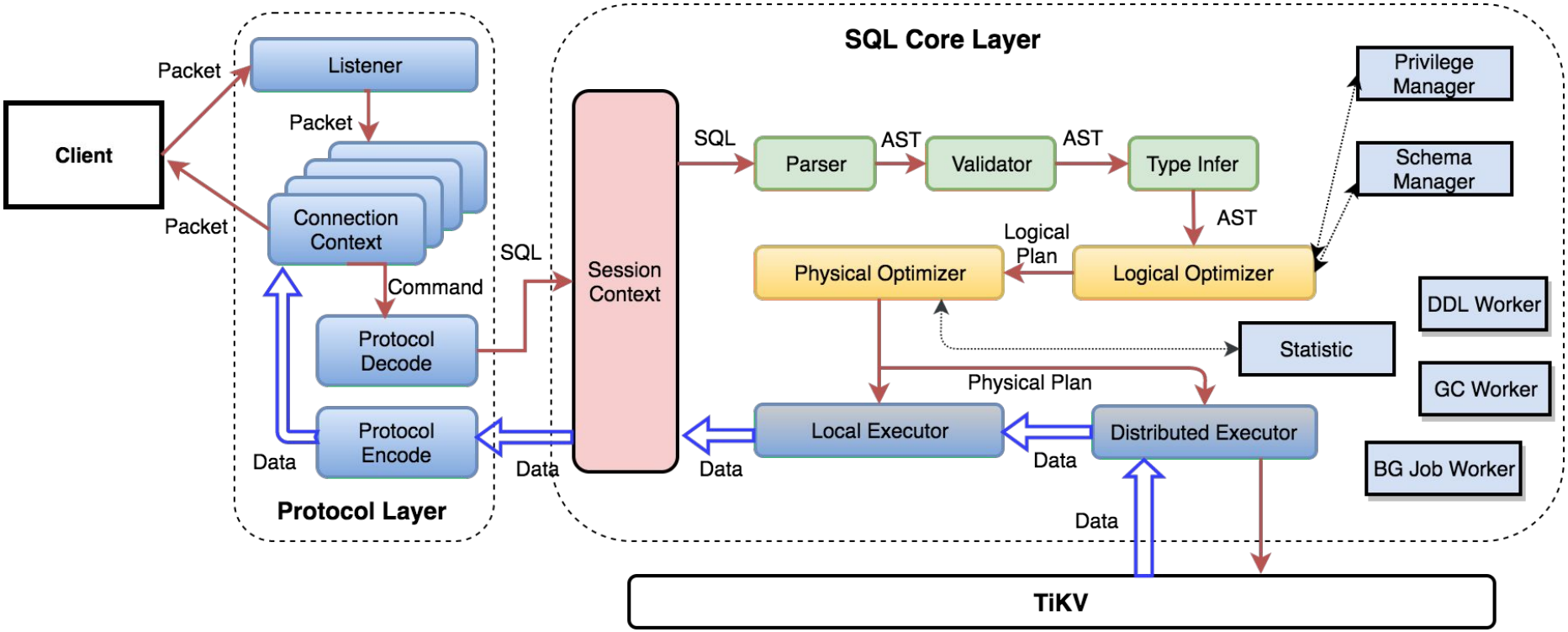
- Stateless SQL layer
  - Clients can connect to any existing tidb-server instance
- Full-featured SQL Layer
  - Speaks **MySQL** wire protocol
  - **CBO**
  - Secondary index support
  - **Online DDL**



https://github.com/pingcap/tidb

PingCAP

Architecture



Mapping relational model to key-value pairs

id (primary)	*name(unique)*	*age(non-unique)*	*score*
1	Bob	12	99

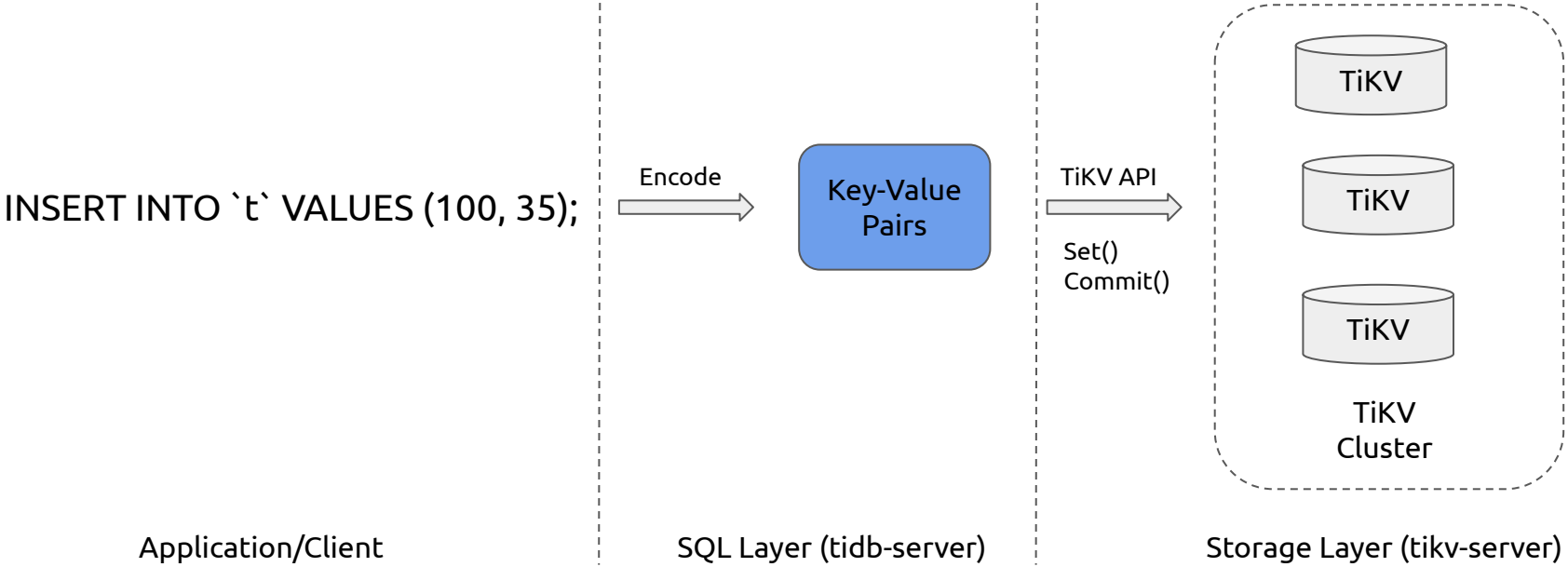
SQL Model

tidb-server

index_type	*key*	*value*
primary_index	t_{tableID}_r_1	(Bob, 12, 99)
name(unique)	t_{tableID}_i_Bob	1
age(non-unique)	t_{tableID}_i_(12,1)	null

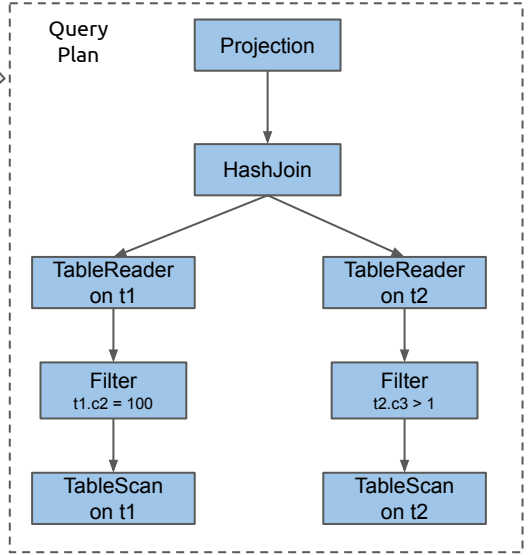
Key-Value Model in TiKV

Write Data into a TiDB Cluster

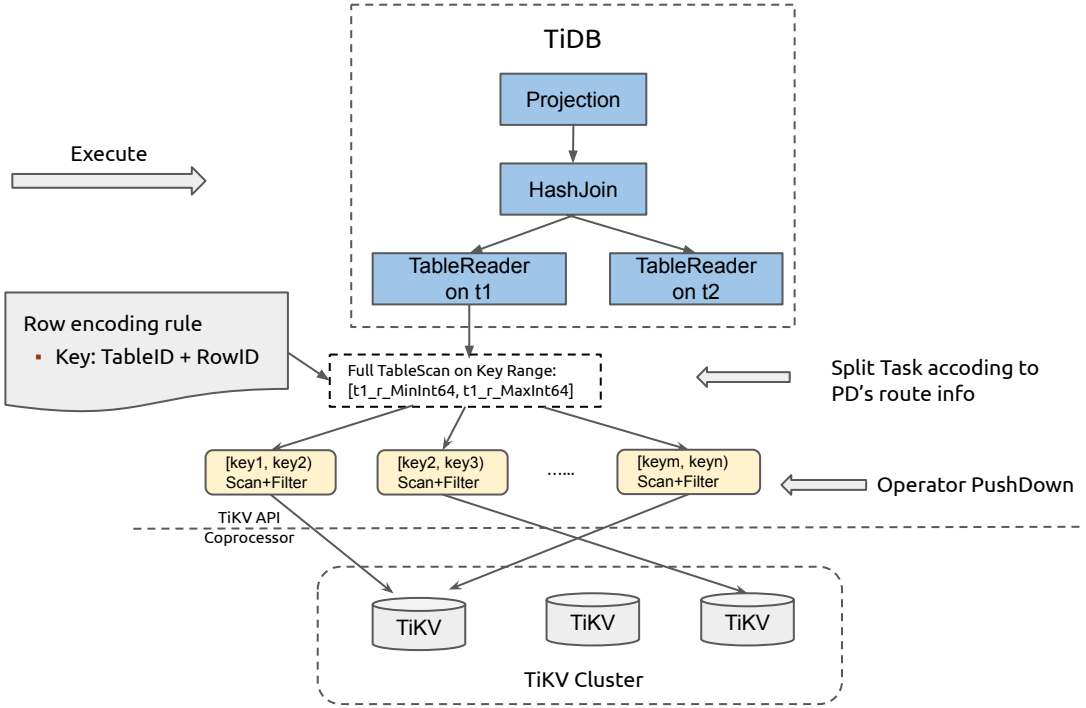


Read Data from a TiDB Cluster

select t1.c1, t2.c3
from t1 join t2 on t1.c1 = t2.c1 where
t1.c2 = 100 and t2.c3 > 1;



Execute



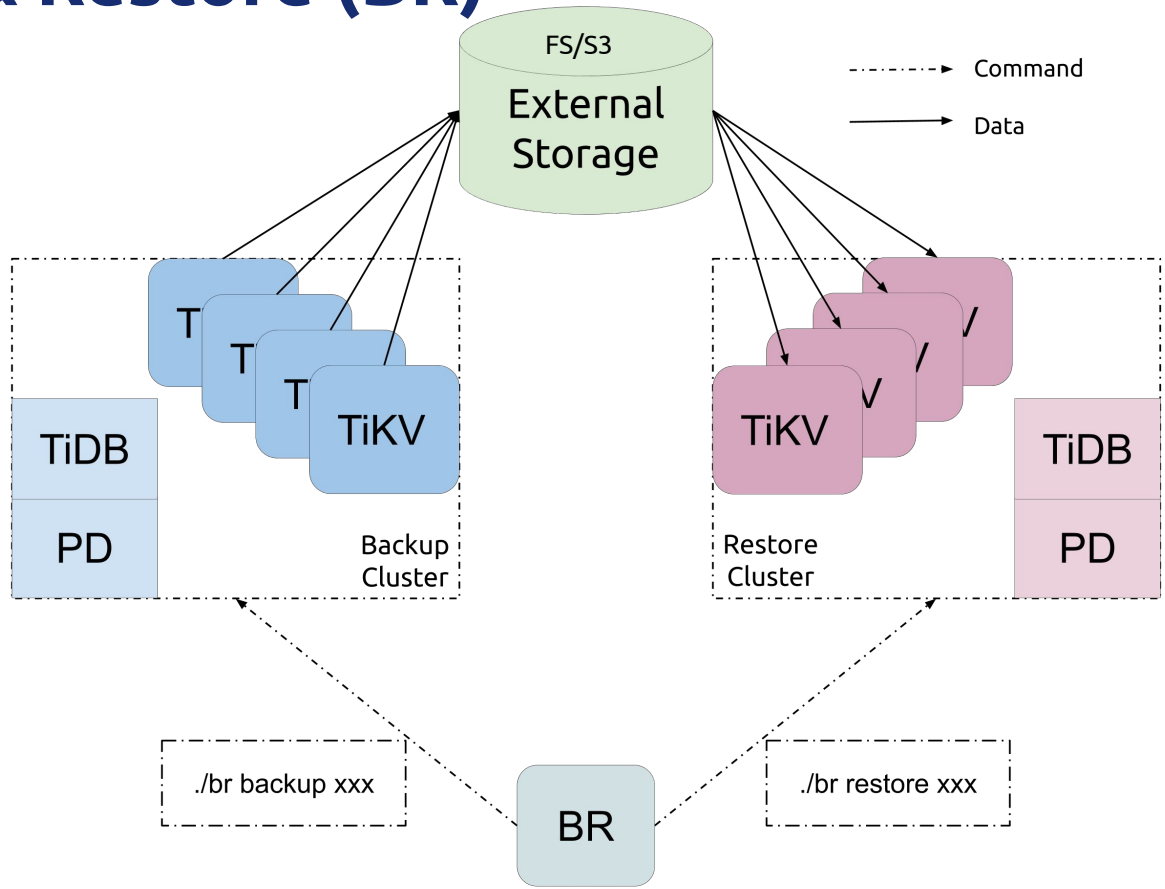
PingCAP

PingCAP

Tools

Fast Backup & Restore (BR)

- For large cluster
- Distributed backup/restore
- ~150MB/s per TiKV instance for backup/restore (with tunable backup speed)
- Support External storage，such as NFS、S3
- Multiple backup methods：Full/DB/Table
- Support CA / SSL security certification

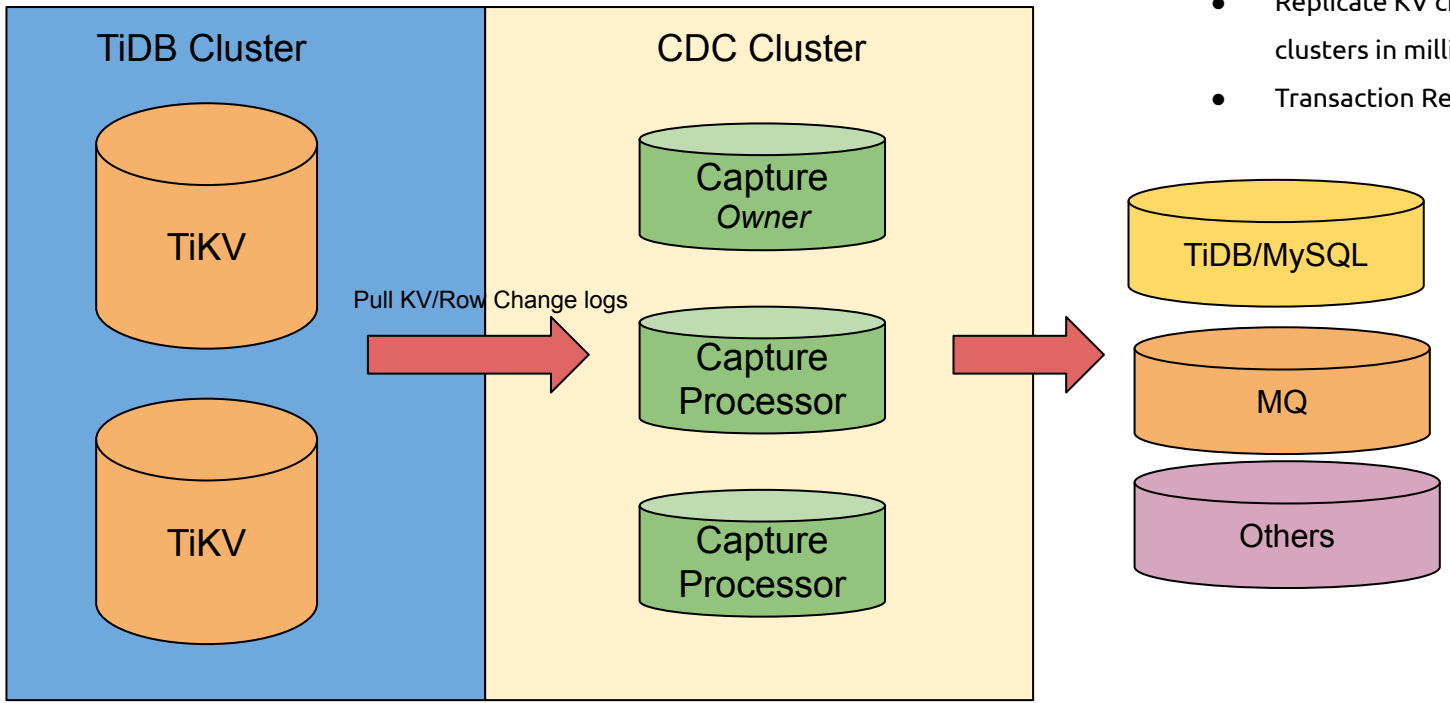


PingCAP

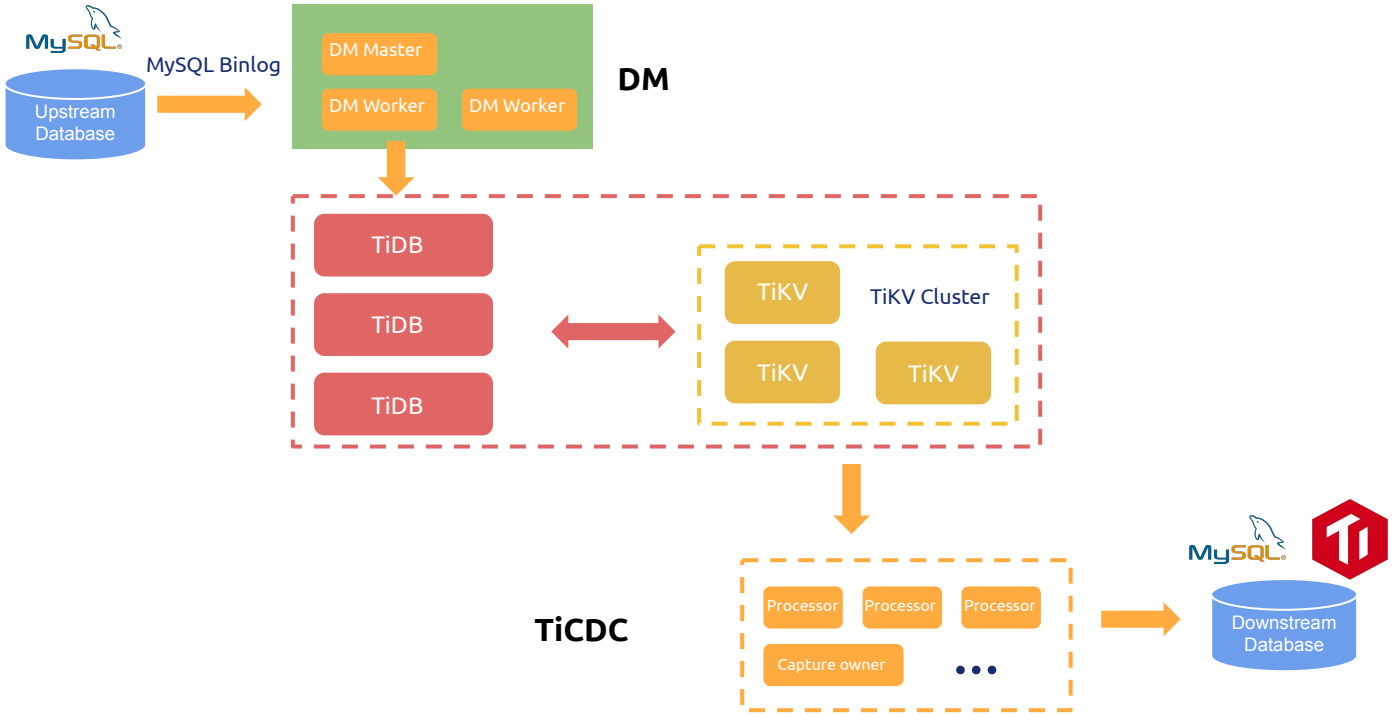
TiCDC

Change Data Capture Tool

- Highly Available
- Distributed, scale to any TiKV cluster size
- Replicate KV changes between TiKV clusters in milliseconds
- Transaction Restore



Data Migration (DM)



PingCAP

Homework

Homework

分值：200

题目描述：

本地下载 TiDB，TiKV，PD 源代码，改写源码并编译部署以下环境：

- 1 TiDB
- 1 PD
- 3 TiKV

改写后

- 使得 TiDB 启动事务时，会打一个 “hello transaction” 的日志

输出：一篇文章介绍以上过程

截止时间：本周日 24:00:00（逾期不给分）

作业提交方式：提交至个人 github，将链接发送给 talent-plan@tidb.io

课程答疑与学习反馈

扫描左侧二维码填写报名信息，加入课程学习交流群，课程讲师在线答疑，学习效果 up up！

PingCAP

PingCAP

Thank You !