

TiDB 的 HTAP 之路

过去, 现在和将来



About Me

- 分析产品负责人@PingCAP
- 曾就职于网易杭研，担任 BigData Infra Team Lead
- 主要关注大数据，分布式数据库，SQL on Hadoop 等领域



TiDB 有很多故事

- 每个故事都可以有多个视角
- 这是一个从 AP 视角讲 HTAP 故事的分享, 当然还有技术讨论

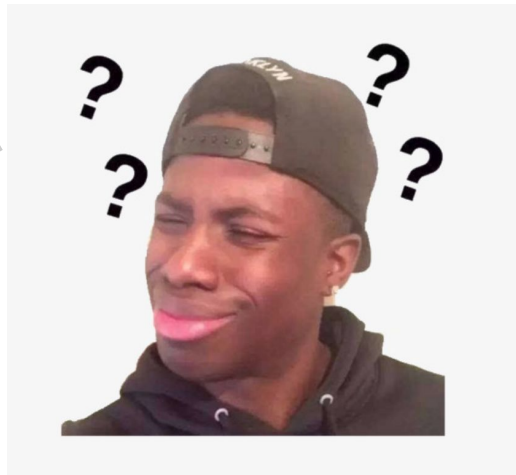
ITPUB.NET



TiDB for HTAP

100% TP 和 80% AP

用户: 为什么是 80% 不是 75%, 也不是 85% ?



TiDB for HTAP

TiDB 是一款 *HTAP* 数据库

所以, 后来我们改用比较精确(时髦)的说法...



TiDB for HTAP

It's a long long journey

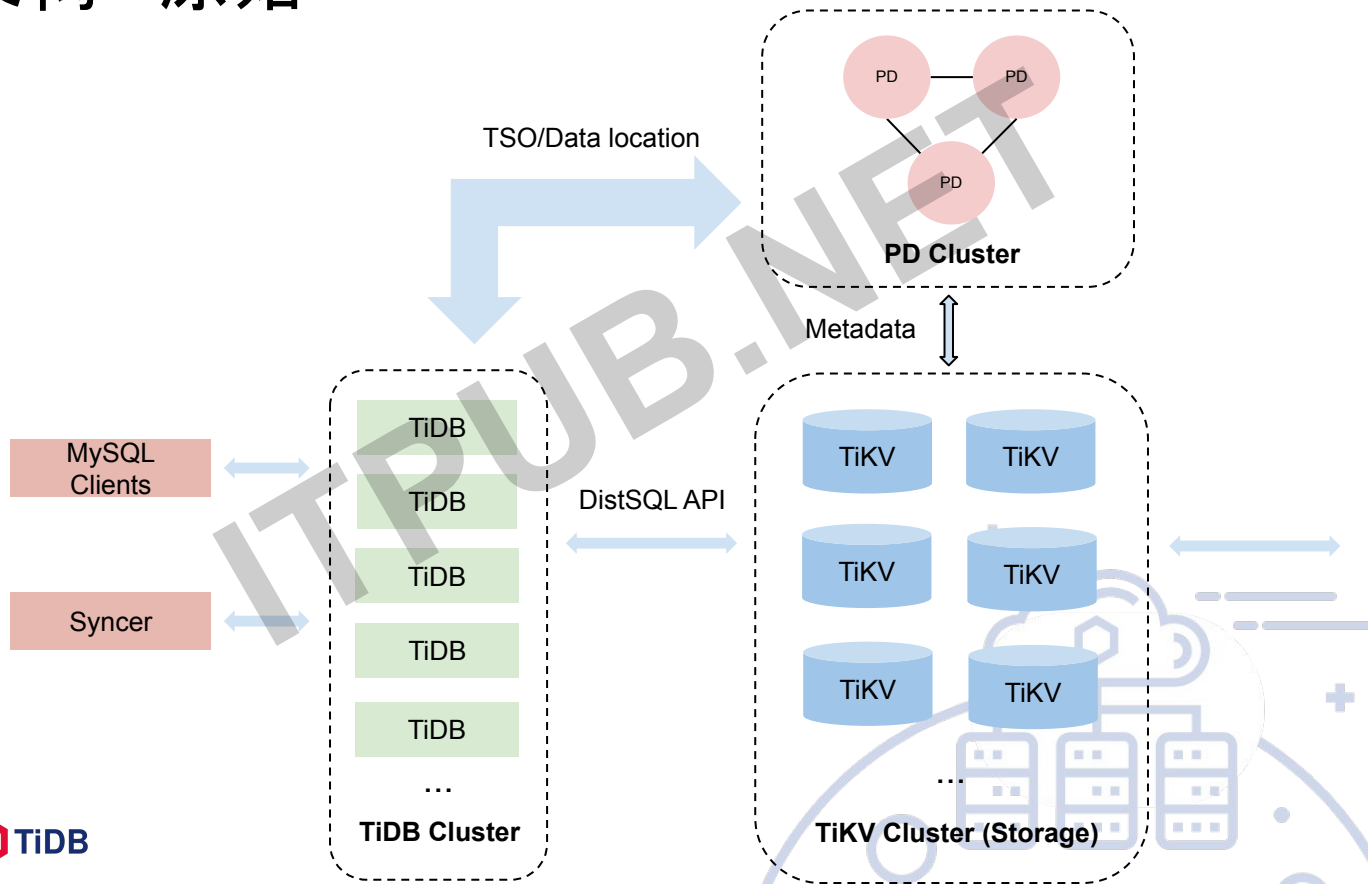


从 TiDB 的上古时代说起

- 受到 Google Spanner 启发，我们做了 TiDB
- 在 Pre GA 版本，TiDB 是
 - 一个可自由扩容(算力，存储)的数据库
 - 兼容支持 MySQL 语法和协议
 - 透明的数据分片策略 - Range 分片
 - 强一致，无视分片的分布式事务支持



TiDB 架构 - 原始



简单说：同款不同尺寸



S



XXXXXXXXXXXXXL

TP 处女秀

- 我们:TiDB 很好用的啦, 可以替换分库分表 MySQL 做 TP 业务。
- 客户:我咋知道你们够稳定呢? 我们先把生产库同步到 TiDB 集群测测看吧。



FP 处女秀

- 我们:用的咋样？
- 客户:同步数据之后做实时分析真的挺方便的...
- 我们:...

ITPUB.NET



中台 AP 能力



- TiDB 非常适合中台场景
- 协议兼容, 轻松同步 MySQL 生产库
- 透明无阻碍的跨分片查询
- 数据实时落地
- 海量存储允许多数据源汇聚
- 备库 - 中台分析二合一

Everyone Happy Now?



一年以后

- TP 场景
 - 客户:虽然还有各种问题...真香！
- AP 场景
 - 客户1:年度报表算的好慢！
 - 客户2:老是 OOM！
 - 客户3:没法和大数据平台结合！



不匹配的算力



PingCAP



TiDB



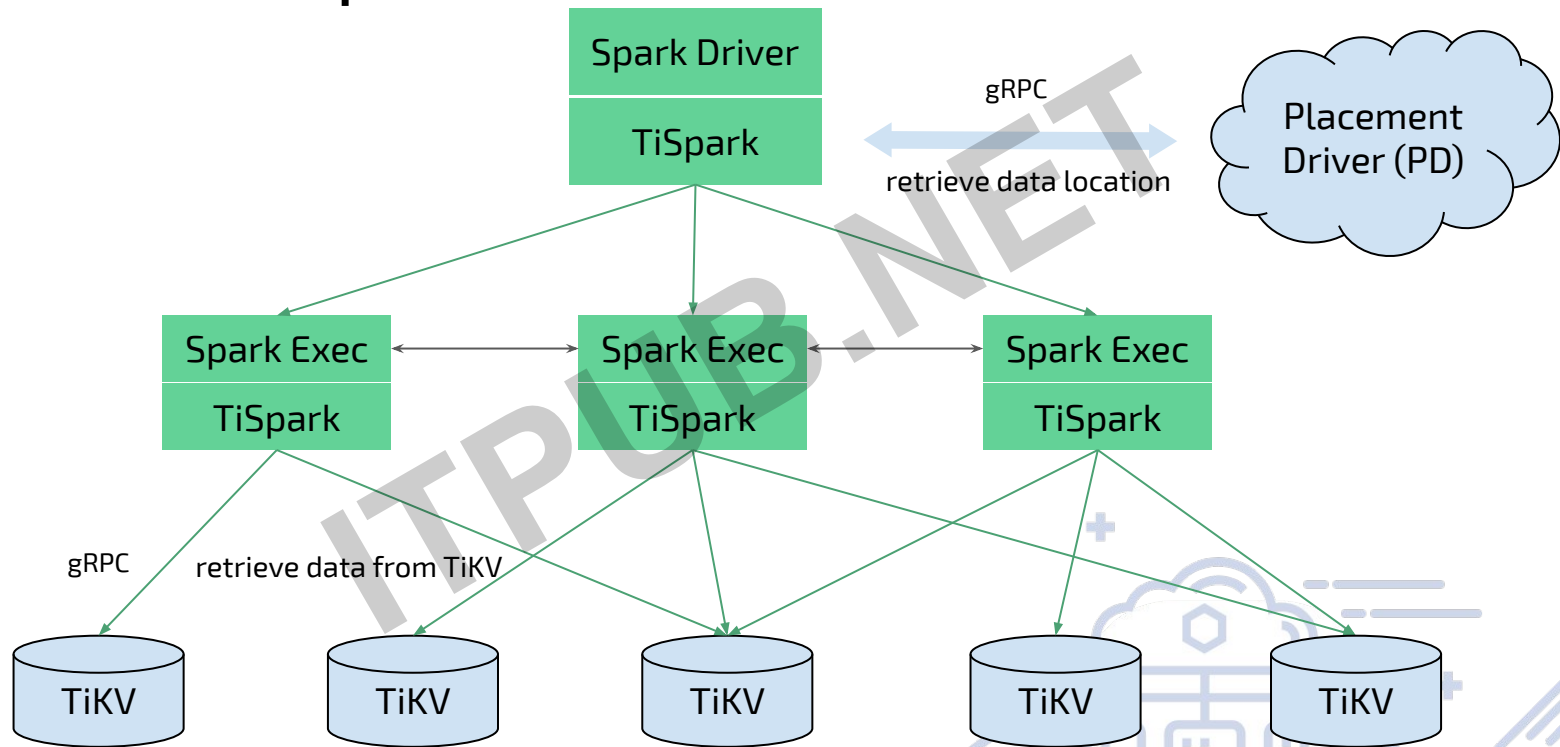
<https://tinyheadedkingdom.com/>

不匹配的算力

- TiDB 之间无法直接交换数据
- TiKV 之间也无法在计算过程中交换数据
- 海量存储(TiKV), 半单机计算(TiDB)
 - 只能通过 TiDB 服务器 Scale-Up 改善
- Coprocessor 无法处理需要数据交换的算子
 - Join, Full Aggregation, Distinct



借助外力 - TiSpark



TiSpark

- Spark 帮我们做分布式计算
 - 成熟的分布式计算平台
 - 更快(?), 更多, 更稳(?)
- 完整继承 Apache Spark 生态圈
 - 无痛衔接大数据生态圈
 - 脚本, JDBC, Python, R, Apache Zeppelin, 衔接 Hadoop 数仓...

Everyone Happy Now?



TiSpark

- Apache Spark 只能提供低并发的粗暴计算
 - 计算模型重, 资源消耗高
 - 更合适报表和重量级 Adhoc 查询
- 用户在很多场合下仍需要高并发中小规模 AP 能力
 - 低消耗低延迟的复杂查询能力
 - TiDB 运维远比 Spark 集群简单



ITPUB.NET



**BACK
TO
START**

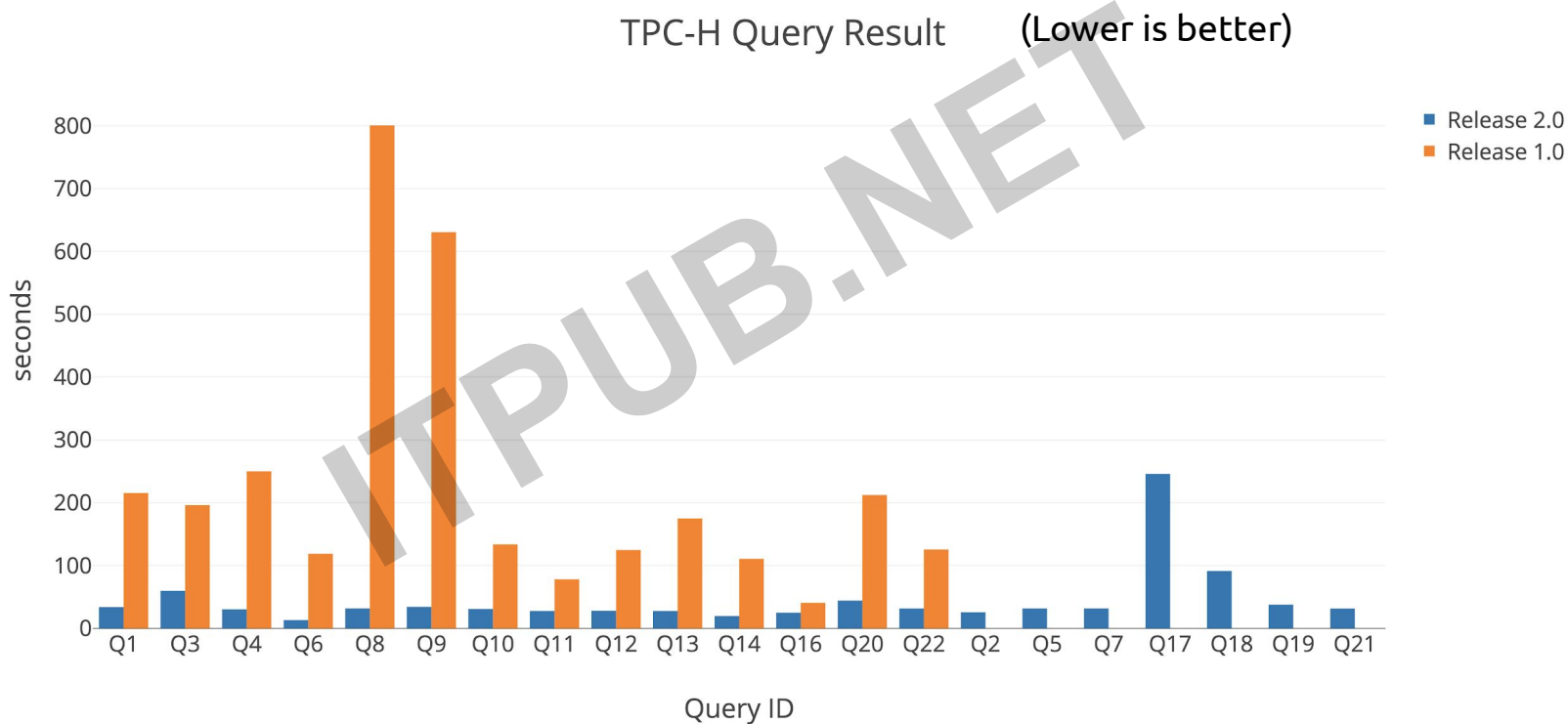


与此同时...

- 我们也在围绕单机 TiDB 进行各种优化
 - 在中小规模场景更聪明, 更高效, 更迅速
- 优化器
 - 你叫它优化器? → RBO + CBO 优化器 → Cascades 优化器(WIP)
- 执行器
 - 经典火山模型 → ~~HT~~ Batch Execution → Vectorized Execution(WIP)
 - 更好的并发与 Pipeline
- 分区表, Index Merge 等等



TiDB 1.0 vs 2.0



Everyone Happy Now?



核心矛盾

- 至此，我们仍然有 2 个核心矛盾：
 - 行存对于分析场景不友好
 - 『没有列存，你们也敢说自己是 HTAP？』
 - 无法做到 Workload 隔离
 - 『我跑跑查询 CPU 就 1000% 辣！』
 - TiSpark 场景下会更糟糕



行存 vs 列存

行存

id	name	age
0962	Jane	30
7658	John	45
3589	Jim	20
5523	Susan	52

SELECT avg(age) from emp;

列存

id	name	age
0962	Jane	30
7658	John	45
3589	Jim	20
5523	Susan	52

TP / AP 干扰



无法兼顾？

『如果你妈和你老婆都掉进河里，你要救哪个？』

『为什么不能都救？』

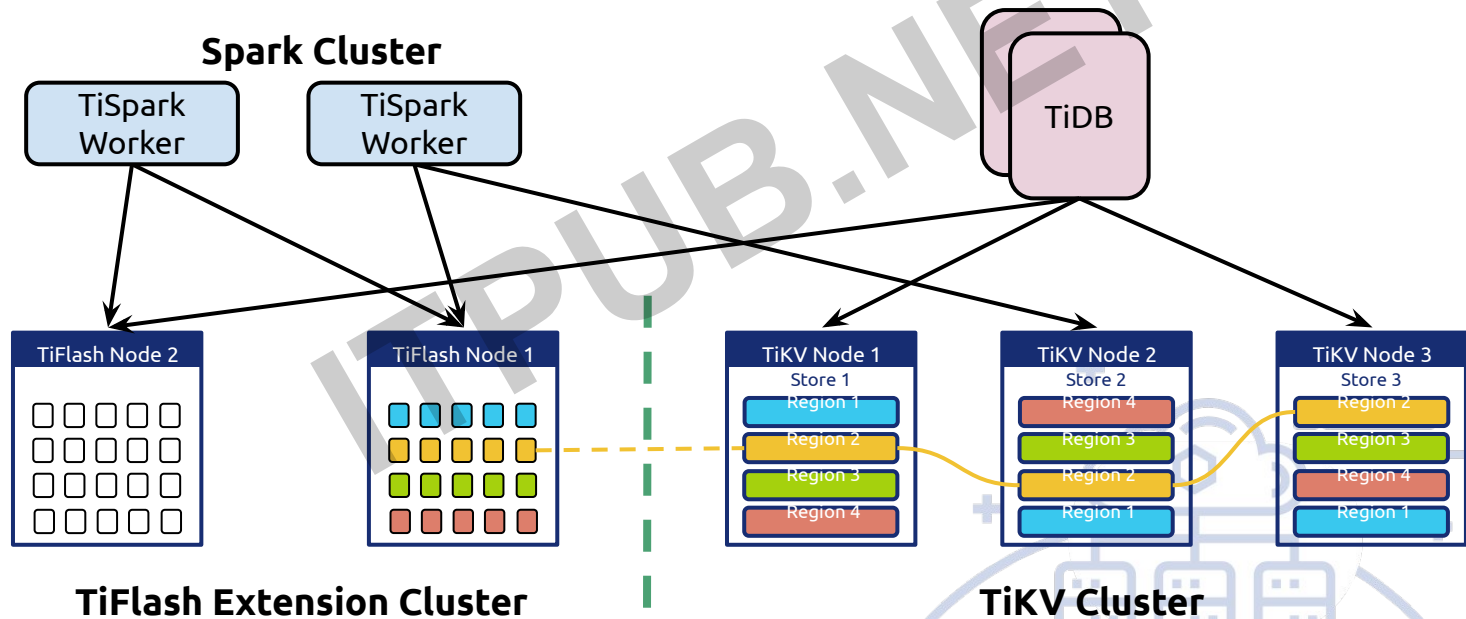


TiFlash Extension - 2019年

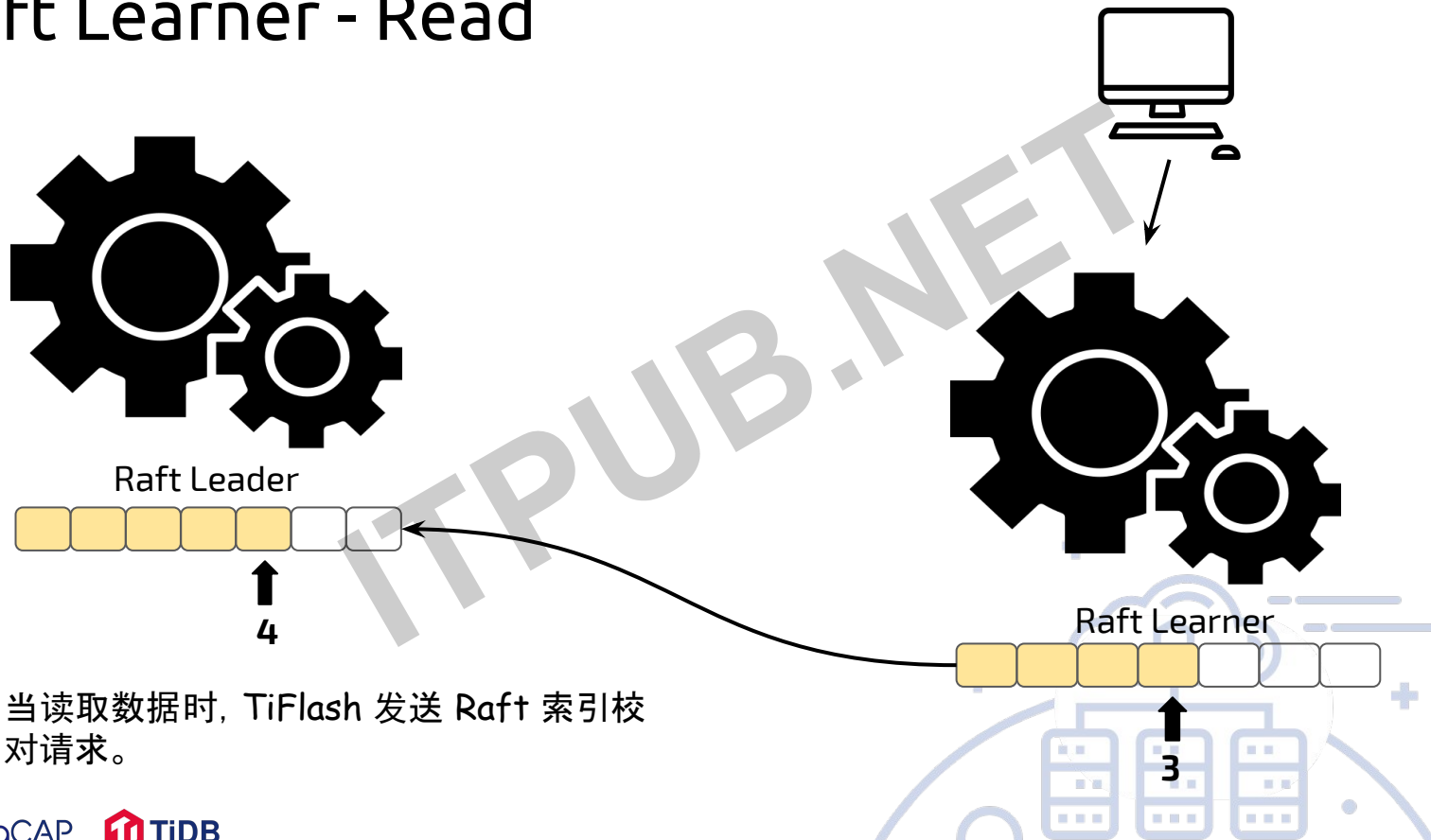
- 通过 Raft Learner 独立同步一套列存
 - Raft Learner 提供极低消耗的副本同步
 - Raft Learner 读取协议配合 MVCC 提供**强一致**的读取
- 通过 Label 进行物理隔离
 - AP / TP 作业互相无影响
 - 但在用户希望的时候, 他们也能互相融合



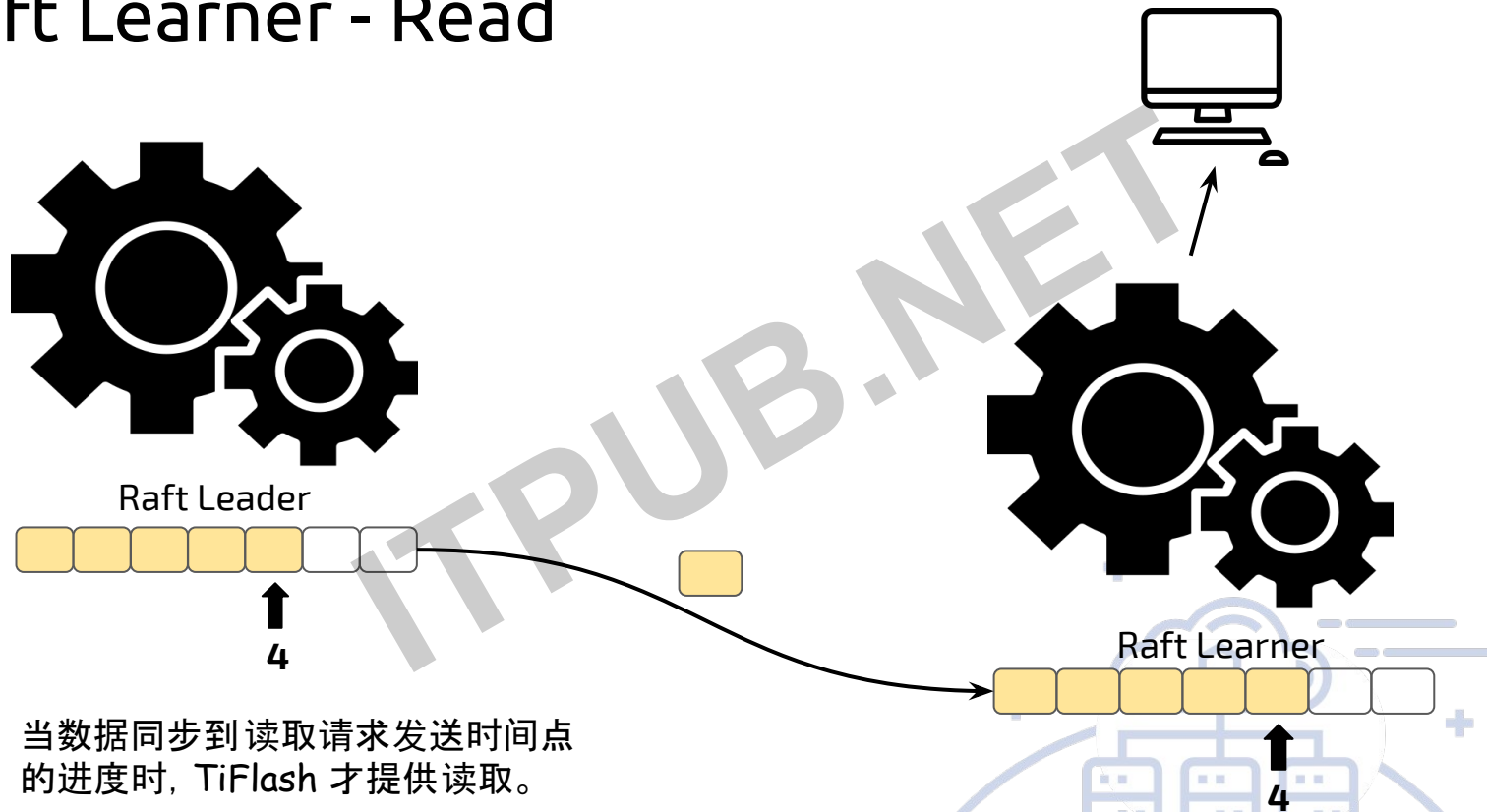
TiFlash 架构



Raft Learner - Read



Raft Learner - Read



当数据同步到读取请求发送时间点的进度时, TiFlash 才提供读取。

TiFlash 项目状态

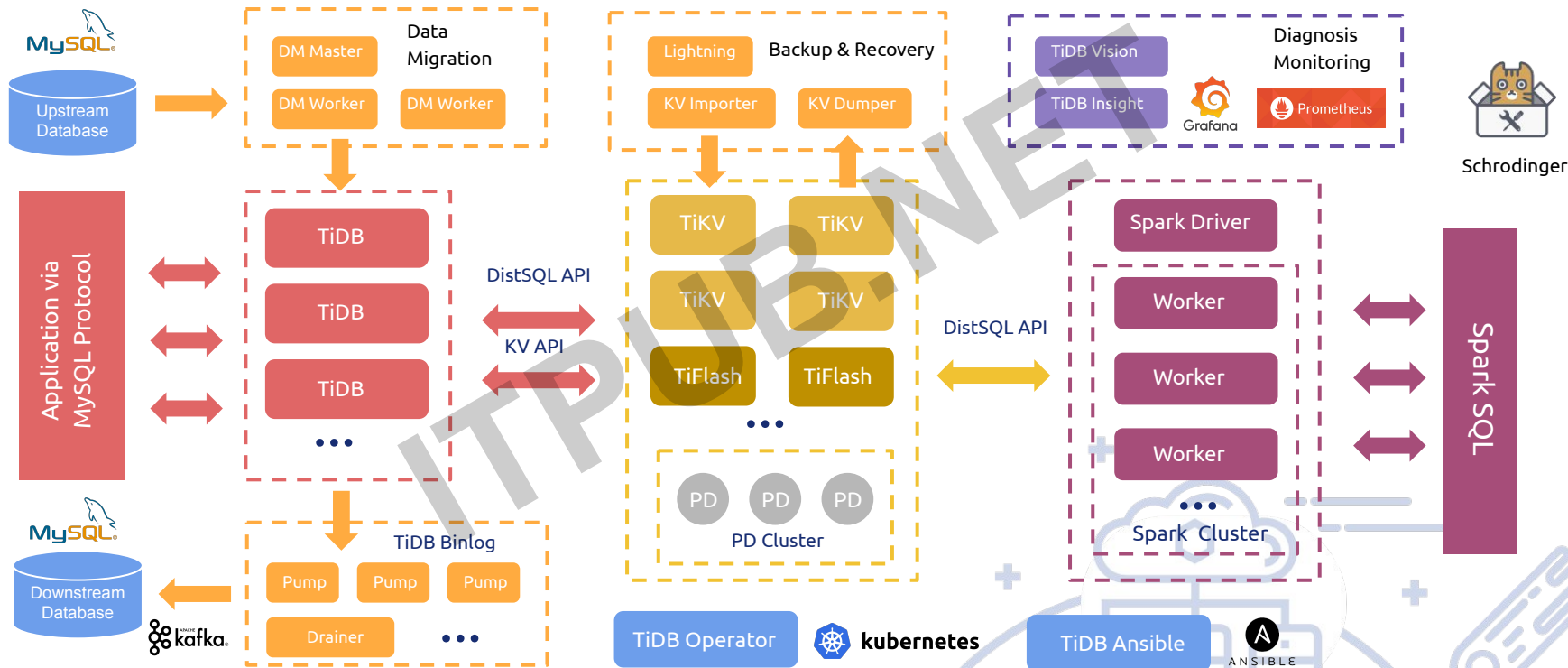
- Ready for POC now
 - Spark 入口
- 2019年内 GA
 - 包含 TiSpark 以及 TiDB 双入口



至此为止

- $\text{TiDB} = \cancel{X\% \text{ TP}} + \cancel{Y\% \text{ AP}} = \text{HTAP}$
 - TiDB 并不需要你选择 TP 还是 AP, 它就是 HTAP
- 一套平台, 兼容行存列存
 - 无痛数据同步
- 当主 TiDB 集群承担 TP 服务时, 方便地在列存上进行分析
- 或者, 将列存当做索引, 和行存共同提供混合服务

TiDB Today



Everyone Happy Now?



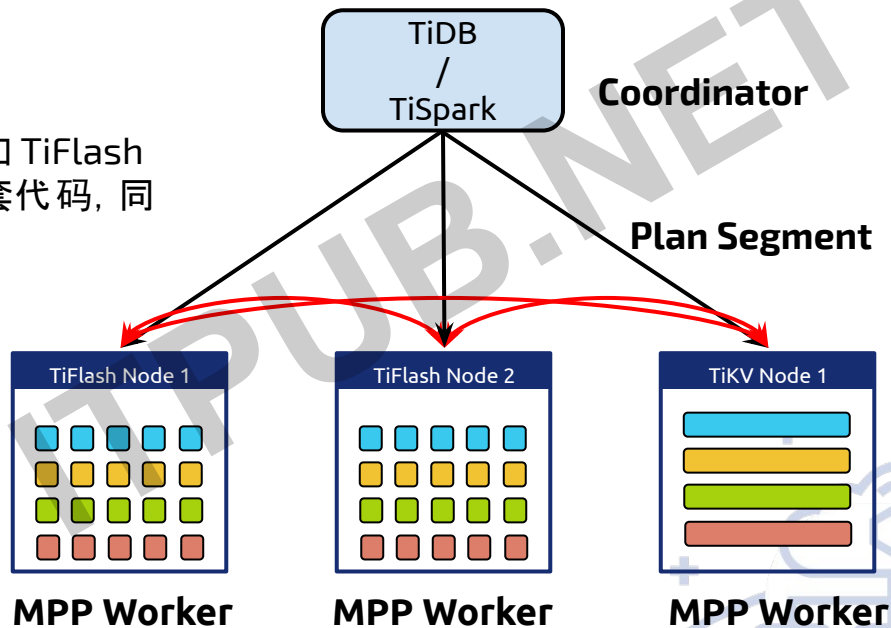
我们还缺什么

- TiSpark 作为唯一分布式计算引擎
 - 缺少中规模快速查询的解决方案
 - 略重的模型(MR 模型)- 仍需要 MPP 引擎
- 写入需要通过 TiKV
 - 大批量写入速度吞吐不够
 - 副本必须先以行存方式写入再同步为列存



MPP 引擎

统一协处理器层，让 TiKV 和 TiFlash 都能组成 MPP 集群。同一套代码，同一套引擎。



写入加强

- TiSpark 批量写入
 - 直接写入 TiKV (WIP, 2019 五月内)
 - TiSpark 终于不是只读系统了
 - 直接写入 TiFlash
 - TiFlash 需要能承担 Raft Leader 角色
 - TiFlash 侧完整的 Multi-Raft 协议对接



其他

- 可拔插的存储引擎, 除了行存列存之外
- Follower + Learner Read, 配合标签隔离: 更精细的分层 Workload 分担
- 预计算类优化
- 更优化的存储模型
- 更大的集群规模
- 以及等等: 还有很多其他工作...



Everyone Happy Then?



努力思考，努力改进，然后？
We will see.



Thank You !

