

UCloud + TiDB: NewSQL 的云化之路

黄东旭

PingCAP 联合创始人兼 CTO





关于我

黄东旭

MSRA / Netease / Wandoulabs / PingCAP
Infrastructure engineer / Open-source hacker
Codis / TiDB / TiKV

数据库怎么了？

业务形态
多种多样

接入终端
五花八门

存储成本
持续降低



海量数据

数据库怎么了？

对开发效率的无止境渴求
关系型数据库仍然是业务的核心
扩展性是新时代基础软件的第一要素
Everything is WEB-SCALE!

数据库怎么了？

「什么叫分库分表？」

「没办法，业务已经用了 MySQL，没法改代码」

「这个要带 Sharding key！」

「这个 JOIN 跨分片没法做」

「没法支持事务」



TiDB Project

- Google Spanner / F1 开源实现
- Google 的新一代分布式关系型数据库
- 目前唯一经过海量数据验证的分布式 OLTP 数据库模型
- MySQL 协议以及语法兼容

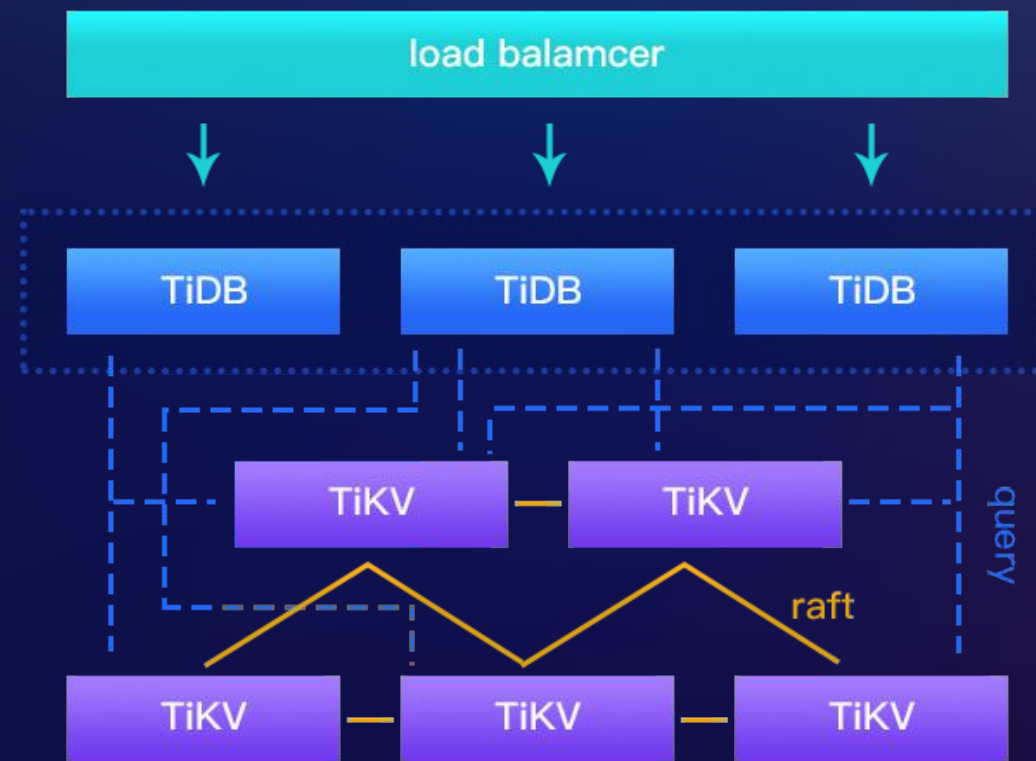


TiDB Project

	传统 MySQL 中间件 / 分库分表	TiDB
强一致的分布式事务	✗	✓
水平扩展	✗	✓
复杂查询 (JOIN/ GROUP BY/...)	✗	✓
无人工介入的高可用	✗	✓
业务兼容性	Low	High
容量	< 2TB	200 TB+



TiDB Project



unlimited
MySQL

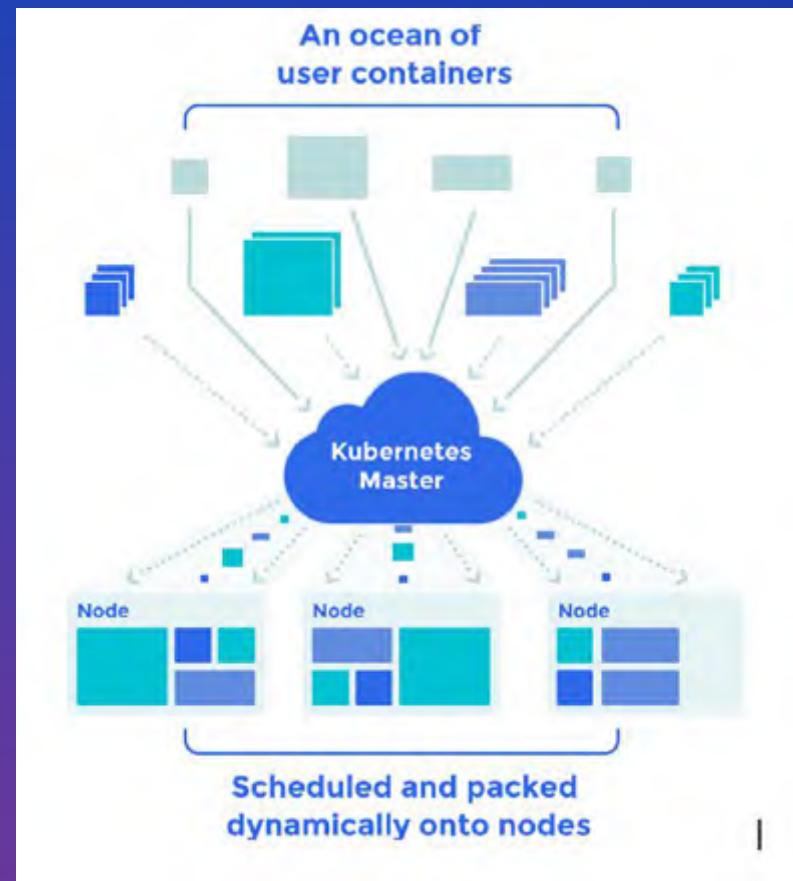


云时代的数据库？Cloud-Native

Scale 作为一等公民
Micro-service 友好
面向容器的部署
自管理

云时代的操作系统：Kubernetes

- Google 的大规模集群调度系统 Borg 的后继
- 自动集群调度
- 服务编排
- 自动化运维
- DCOS ?



最大的问题：状态

- 整个应用层分裂成 4 个的阵营：
 - Stateless applications
 - Single point stateful applications
 - Static distributed applications
 - Clustered applications



Single point stateful application

- 单点带状态服务
 - MySQL
 - PostgreSQL
 - Redis
- 使用 Static configuration or StatefulSets



Static distributed application

- 分布式静态带状态服务
 - ZooKeeper
 - Etcd
- StatefulSets



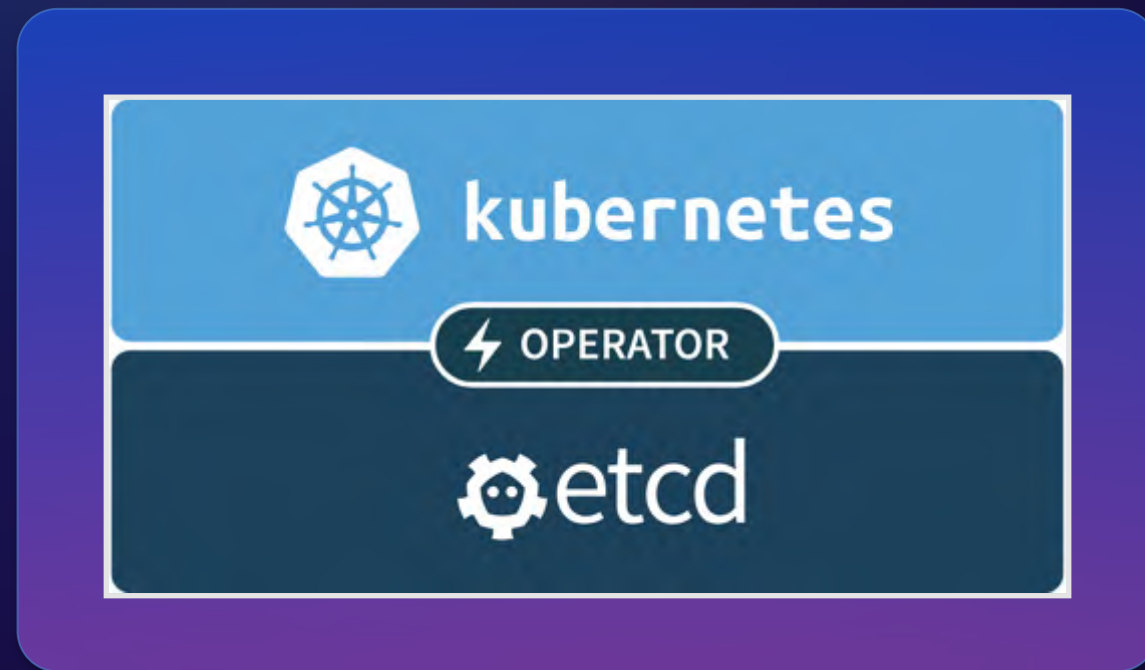
Clustered application

- 集群存储服务
 - Redis Cluster
 - Cassandra
 - TiDB
- 带有领域知识
- CoreOS Operator

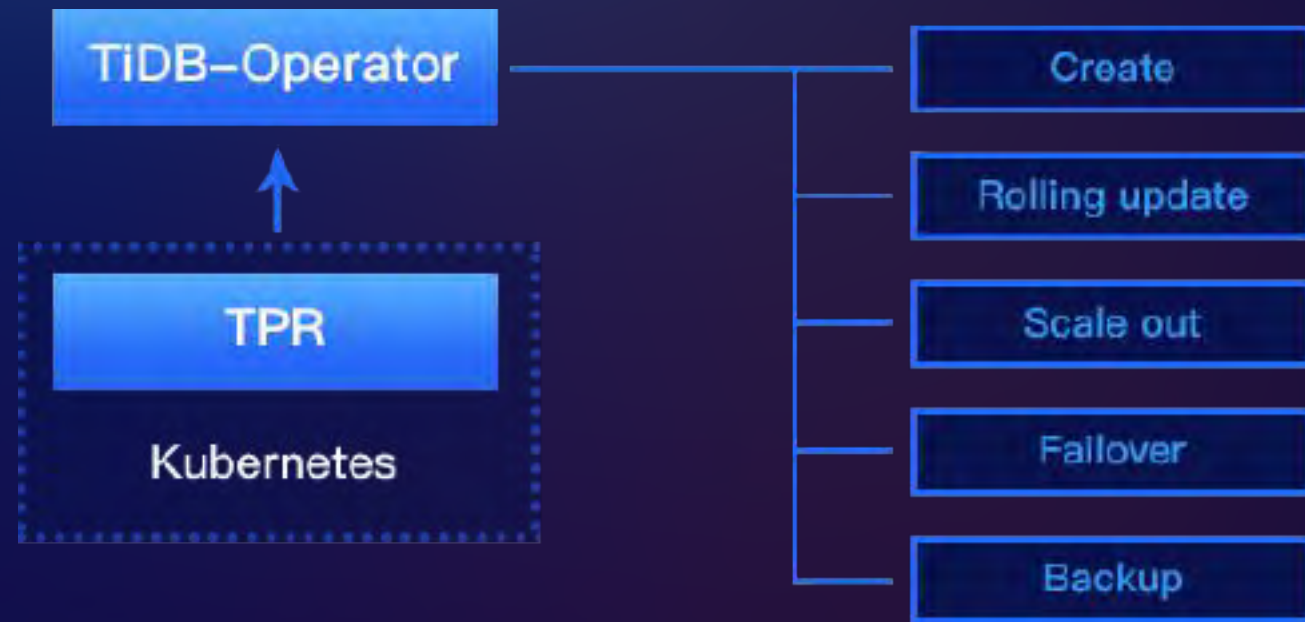


Operator

- Putting Operational Knowledge into Software
- K8S 不知道该怎么调度我，所以我来指导
- CoreOS 出品



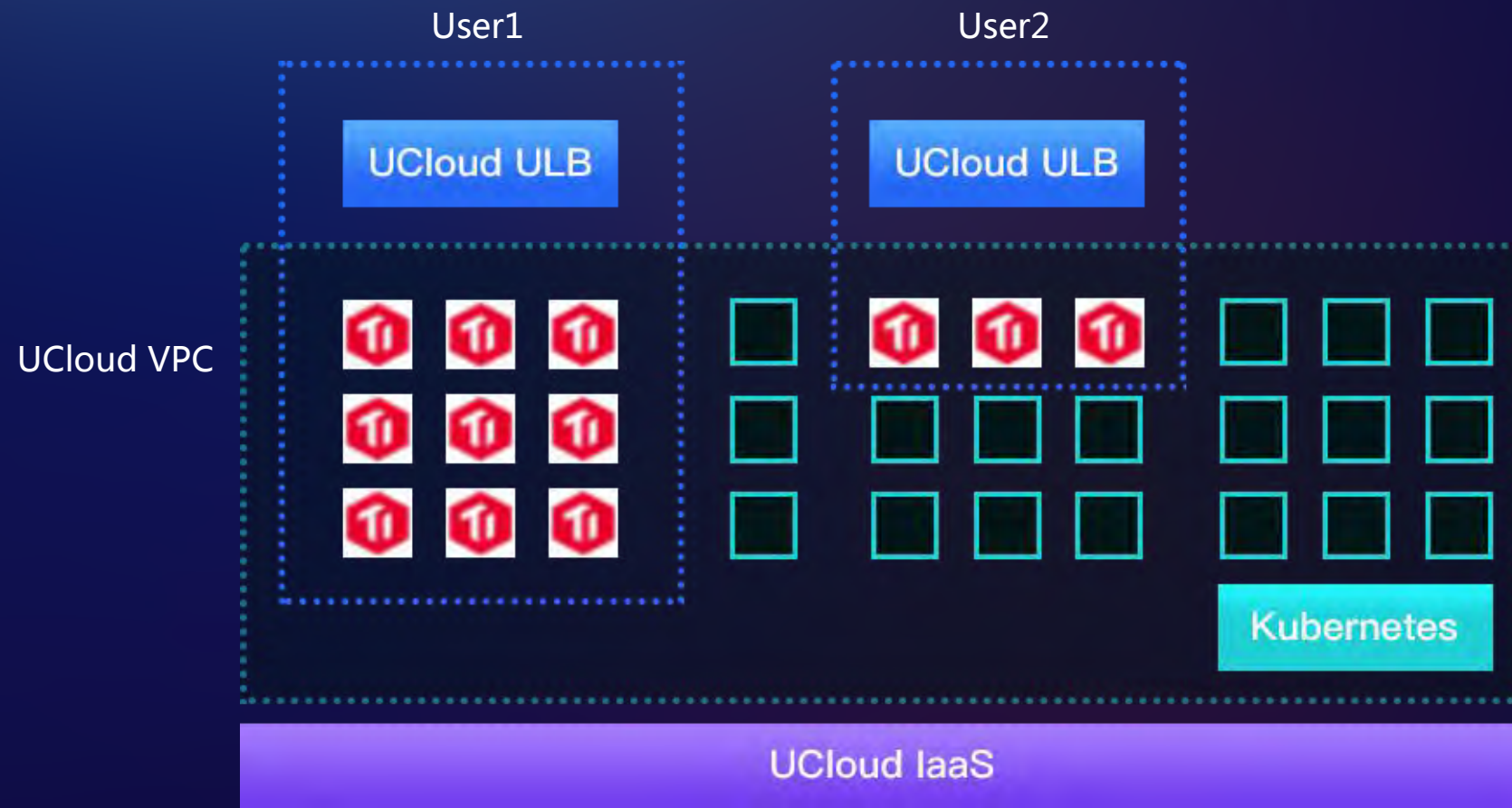
TiDB Operator



Why Operator

- 对 Kubernetes 的侵入性小
- PV / StatefulSet 并没有解决问题或者还不够

When TiDB meets UCloud





When TiDB meets UCloud

- 快速创建 / 扩展 / 收缩集群
- 实时 TiDB <=> MySQL 主从同步
- MySQL 备份格式兼容

Benchmark

- sysbench standard select test

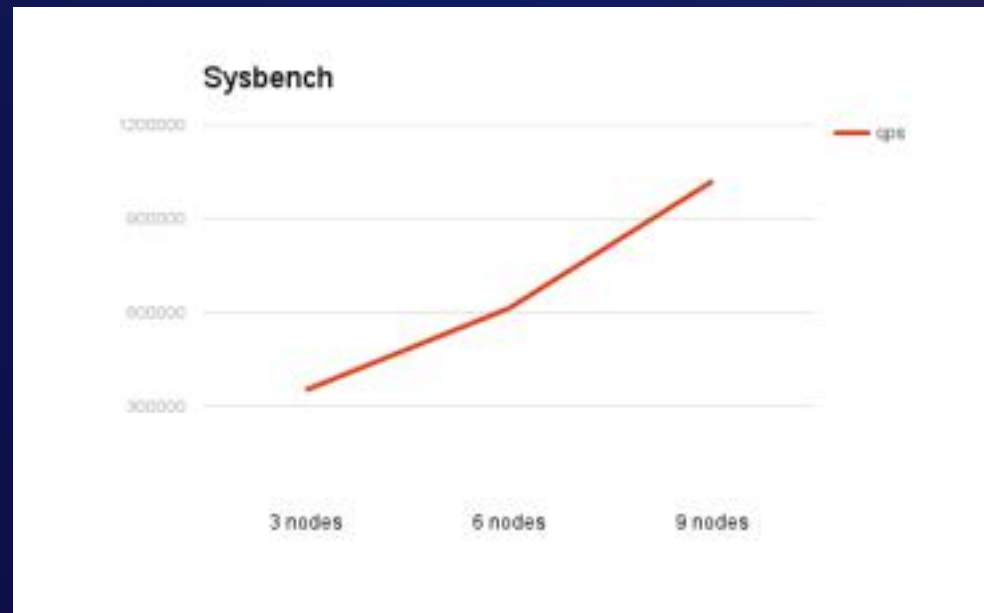
	table count	table size	sysbench threads	qps	latency(avg/.95)
TiDB	16	1M rows	128	26439.02	4.84ms / 11.25ms
TiDB	16	1M rows	256	27683.97	4.84ms / 11.25ms
MySQL	16	1M rows	64	90143.66	0.71ms / 1.83ms
MySQL	16	1M rows	128	94638.56	1.53ms / 2.18ms

Benchmark

- sysbench standard insert test

TiDB	16	1M rows	128	8086.33	15.82ms / 29.98ms
TiDB	16	1M rows	256	9796.68	26.13ms / 44.21ms
MySQL	16	1M rows	64	3373.91	35.9ms / 0.90ms
MySQL	16	1M rows	128	3150.73	40.61ms / 1.65ms

SQL that SCALES!



DEMO Time

没有银弹



- 自增 ID
- Time stamp ordering
- 业务自身存在冲突和热点，如秒杀
- 业务压力集中在小表



- MySQL 容量达到单机瓶颈
- 读写分布相对平均
- 并发事务支持，但是大多数时间冲突不高
- 有实时的复杂查询需求

THANKS!

