



Cloud + TSDB for ClickHouse

张健

关于我

- ▶ 目前就职于青云数据库团队
- ▶ GayHub: <https://github.com/zhang2014>
- ▶ Housepower: ClickHouse周边工具、文档翻译

目录

- ▶ 背景
- ▶ 我们的实践



背景



我们的诉求

- ▶ 自身云平台存在大量的监控数据需要处理
- ▶ 面对正处于数字化转型进程中的传统客户的需求
- ▶ 未来产品化的可能

为什么选择ClickHouse

- ▶ ClickHouse. Just makes you think faster.
- ▶ 我们存在一定的技术积累
- ▶ ClickHouse vs InfluxDB vs Green plum



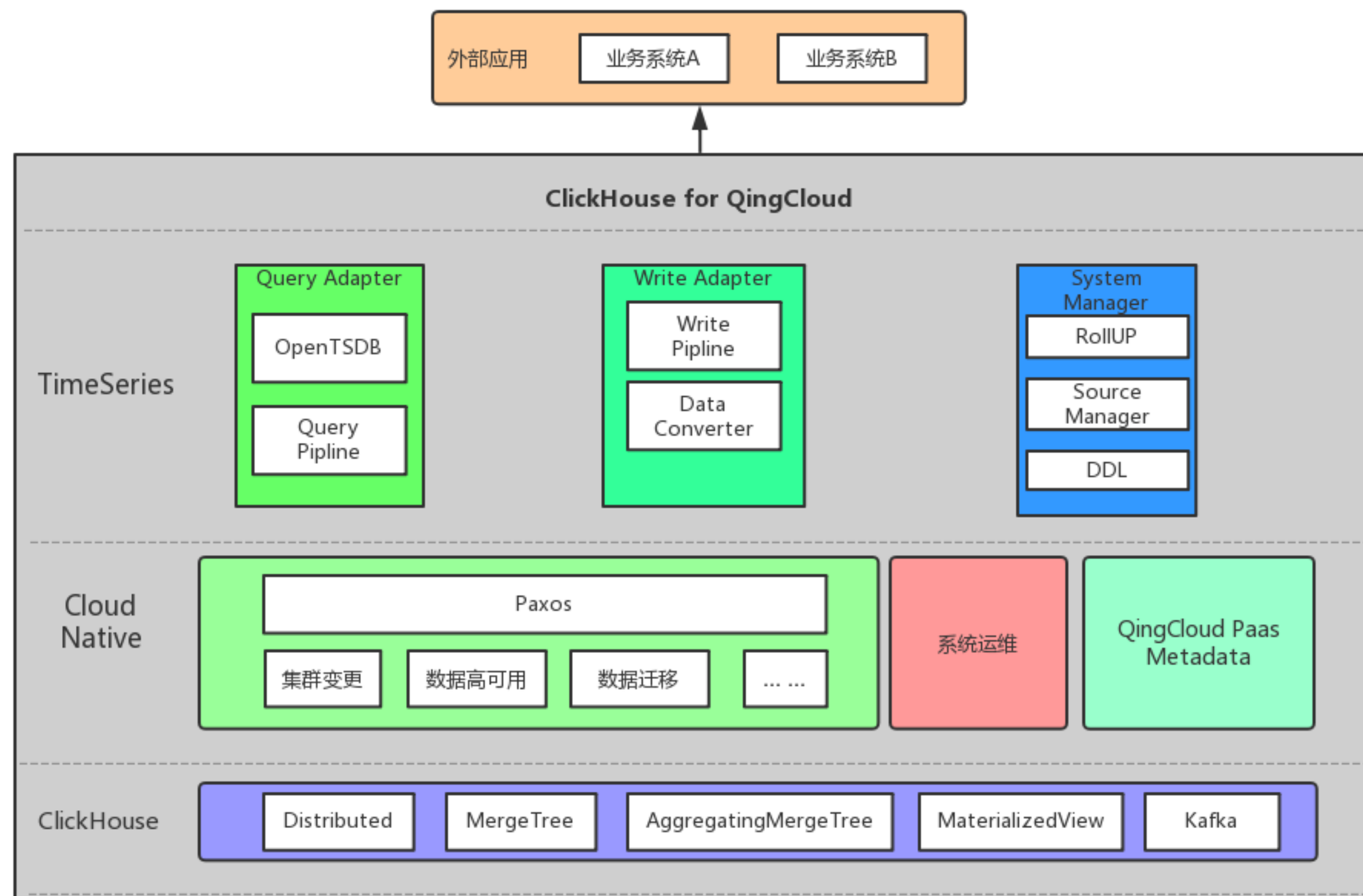
我们的实践



我们的实践

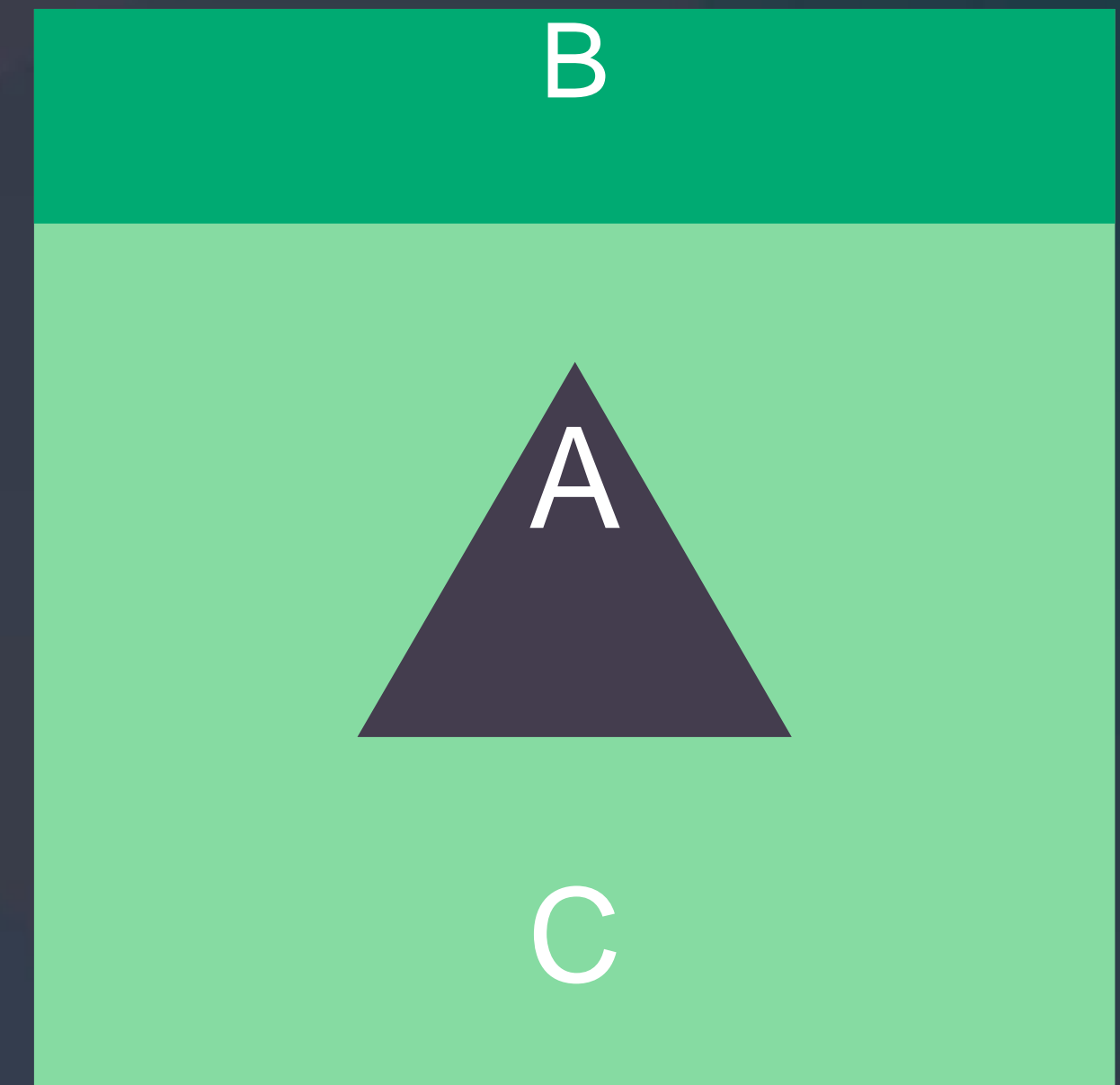
- ▶ Architecture
- ▶ Cloud Native
- ▶ TimeSeries
- ▶ TODO

Architecture



Cloud Native

- ▶ 逻辑概念
- ▶ 重写转发查询的方式实现
- ▶ 通过配置将节点组合成集群
- ▶ 不同集群间可以存在相同节点

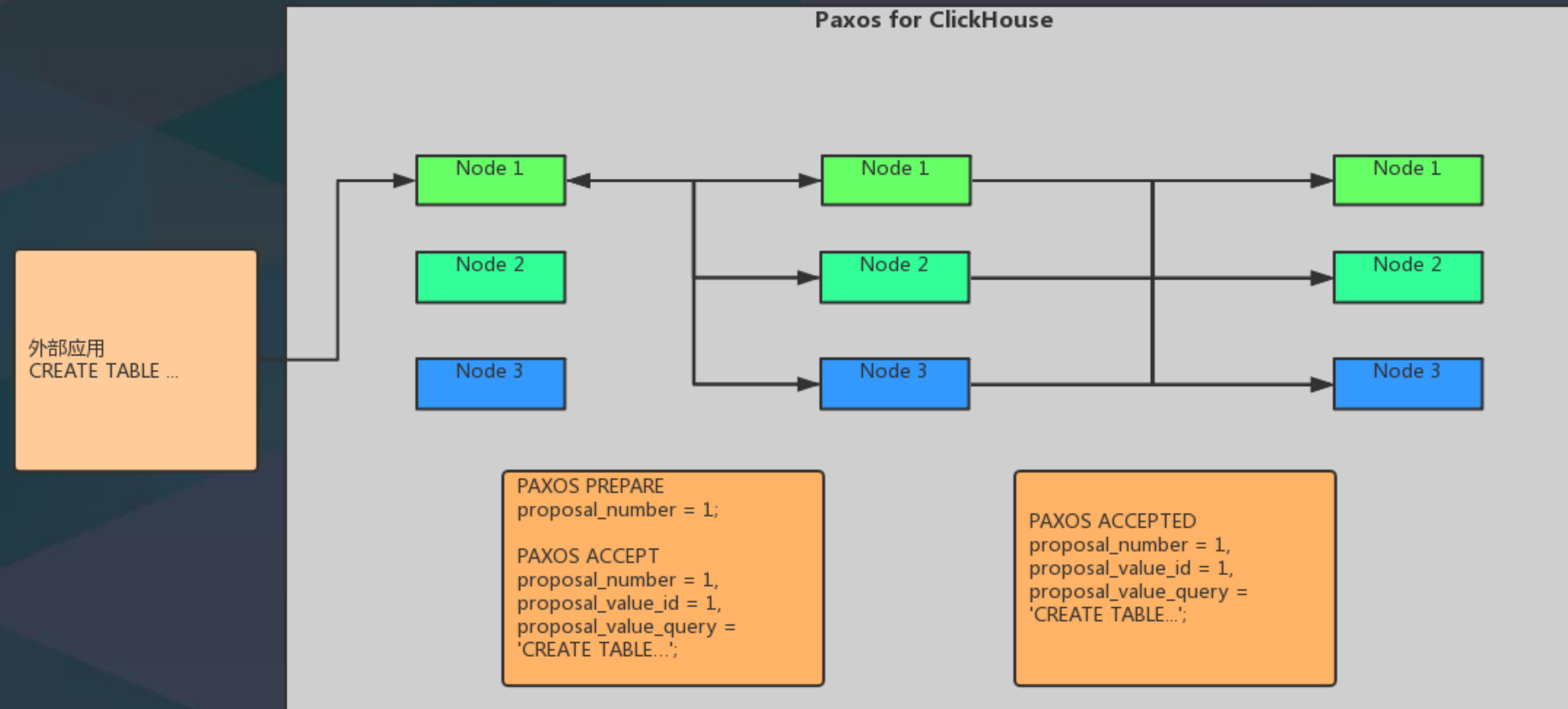


Cloud Native

- ▶ DDL状态一致性维护
- ▶ 集群结构可变更
- ▶ 集群数据可迁移

Cloud Native

► 使用Paxos保持节点间的DDL状态



Cloud Native

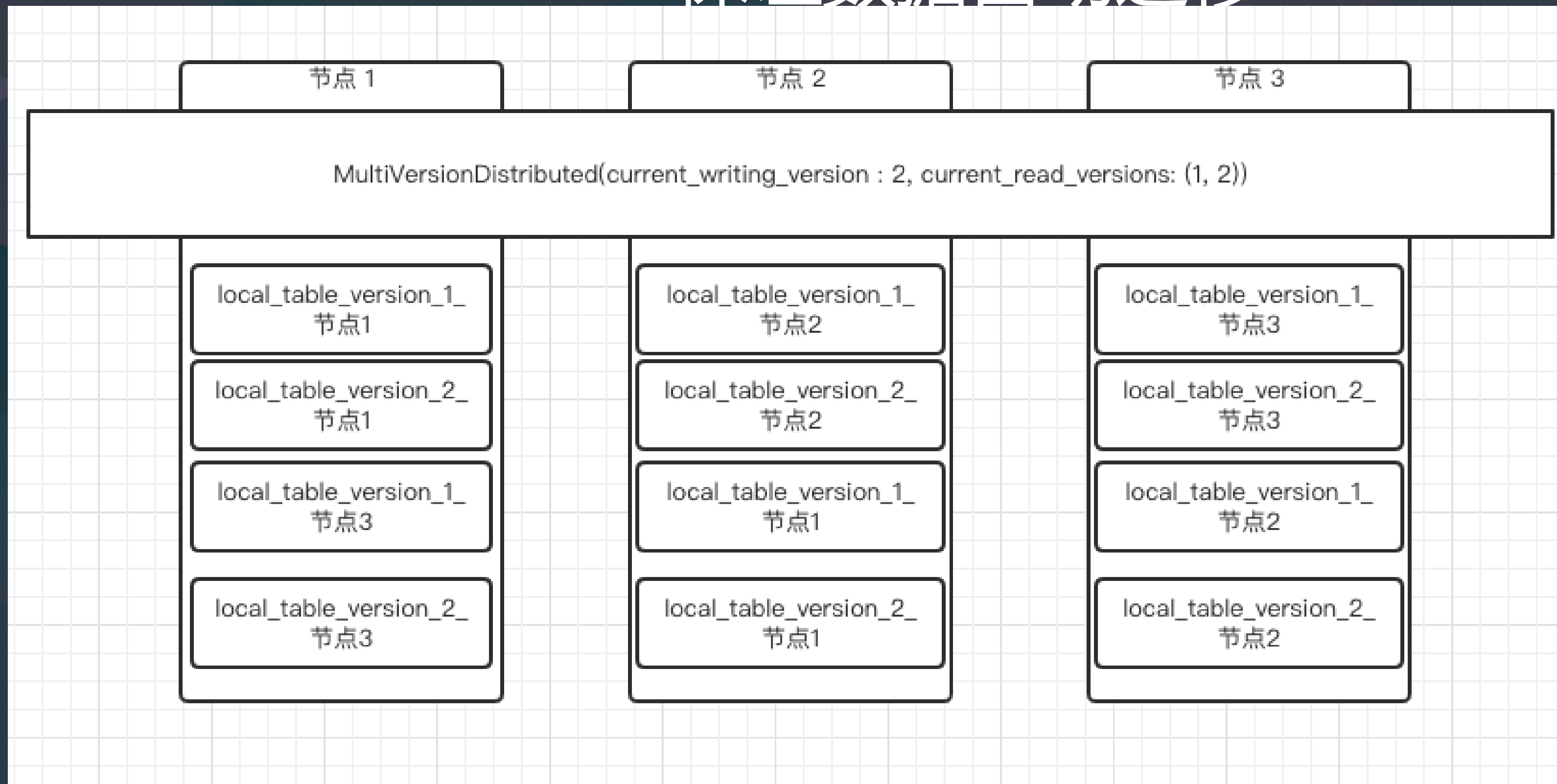
- ▶ 使用Paxos保持节点间的DDL状态
 - ▶ PAXOS PREPARE proposal_number = 1;
 - ▶ PAXOS ACCEPT proposal_number = 1, proposal_value_id = 1, proposal_value_query = 'CREATE TABLE...';
 - ▶ PAXOS ACCEPTED proposal_number = 1, proposal_value_id = 1, proposal_value_query = 'CREATE TABLE...';

Cloud Native

- ▶ 使用QingCloud PaaS Metadata 保持集群配置感知

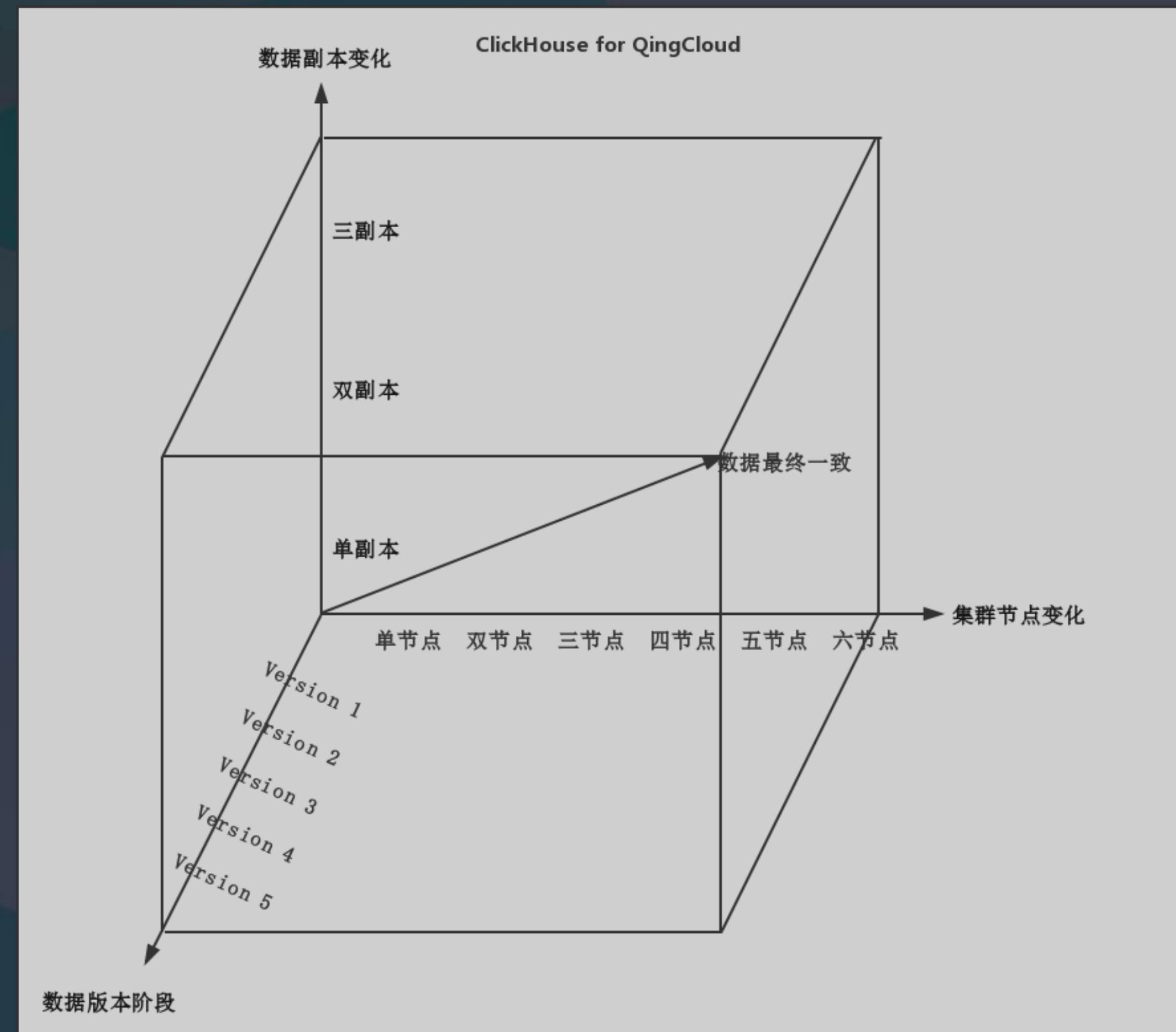
Cloud Native

► 使用MultiVersionDistributed保证数据自动迁移



Cloud Native

► 使用MultiVersionDistributed保证数据自动迁移



Cloud Native

- ▶ 使用MultiVersionDistributed保证数据自动迁移
 - ▶ `VERSION_1 = (READABLE NODE1), VERSION_2 = (READABLE, WRITEABLE, NODE1, NODE2), VERSION_3 = (NODE1, NODE2)`
 - ▶ `SET query_version = VERSION_1; SET writing_version = VERSION_3;`
 - ▶ `INSERT INTO table_A SELECT * FROM table_A;`
 - ▶ `VERSION_1 = (), VERSION_2 = (READABLE, NODE1, NODE2), VERSION_3 = (READABLE, WRITEABLE, NODE1, NODE2)`
 - ▶ `MERGE VERSIONS VERSION_2, VERSION_3 TO VERSION_3;`

Cloud Native

- ▶ 使用MultiVersionDistributed保证数据自动迁移
 - ▶ VERSION_1 = (READABLE NODE1), VERSION_2 = (READABLE, WRITEABLE, NODE1, NODE2), VERSION_3 = (NODE1, NODE2)
 - ▶ SET query_version = VERSION_1; SET writing_version = VERSION_3;
 - ▶ ~~INSERT INTO table_A SELECT * FROM table_A;~~
 - ▶ VERSION_1 = (READABLE, NODE1), VERSION_2 = (READABLE, NODE1, NODE2), VERSION_3 = (READABLE, WRITEABLE, NODE1, NODE2)
 - ▶ MERGE VERSIONS VERSION_1, VERSION_2, VERSION_3 TO VERSION_3;

TimeSeries

- ▶ 按列压缩
- ▶ RollUp : AggregatingMergeTree

TODO

- ▶ 动态Schema实现
- ▶ 提供system.kafka对kafka引擎的监控
- ▶ LowCardinality在高基数场景下的表现

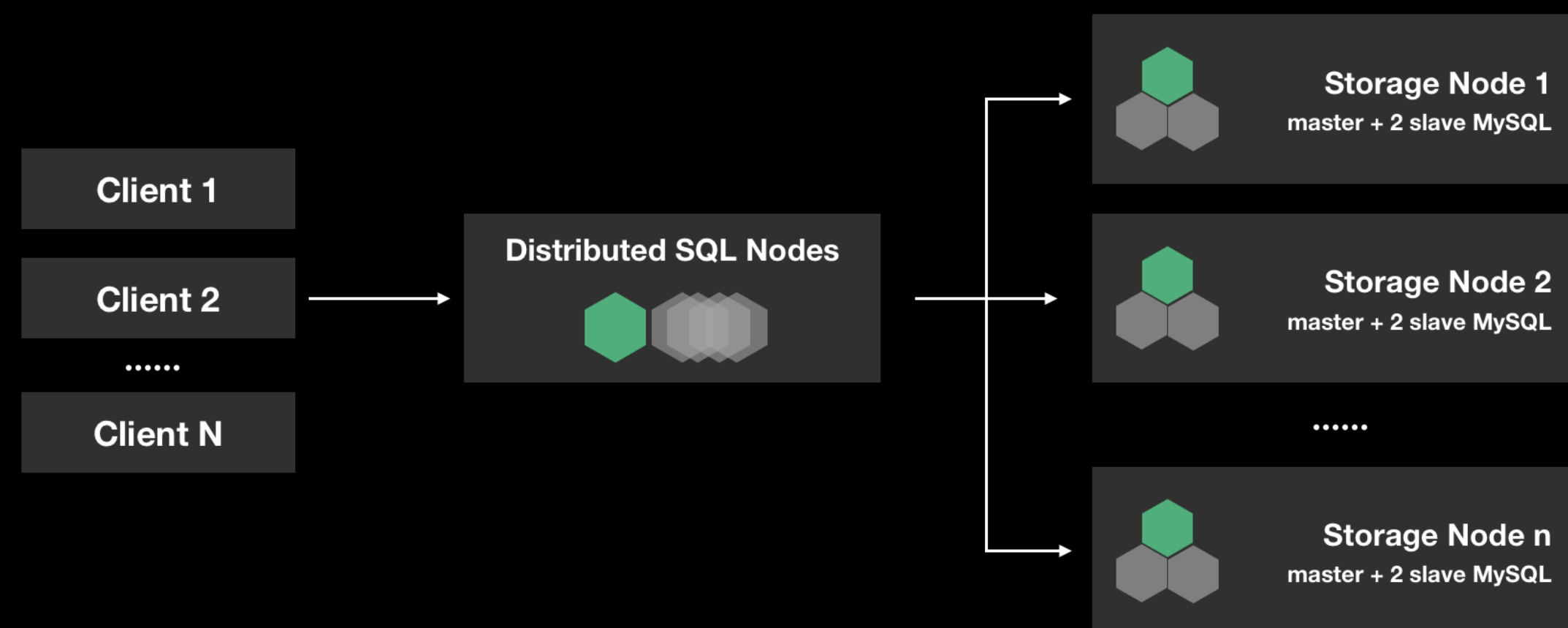
数据库



云原生分布式关系型数据库



- 金融级数据强一致
- 基于 MySQL 存储引擎
- 水平扩展
- 自动分库分表
- 支持 HTAP
- 智能扩容



- 关系型数据库: MySQL Plus、PostgreSQL
- 非关系型数据库: MongoDB
- 缓存: Redis/Redis Clusters
- Coming Soon: 时序数据库、分析数据库、键值数据库





Thank you.
winterzhang@yunify.com

