Alibaba Group
阿里巴巴集团

AliSQL 引领开源技术变革之路

数据库事业部-数据库内核-何登成

个人简介

- **何登成，资深技术专家@Alibaba**
  - ✓ 从04年开始从事数据库内核研发达10+年以上。先后参与并主导过国产神舟Oscar数据库，网易自研存储引擎NTSE/TNT等数据库产品的研发。同时也作为数据库总负责人参与了多年阿里巴巴双11购物狂欢节，蚂蚁新春红包的备战保障工作。有着丰富的数据库内核研发经验和数据库应用架构经验。目前负责阿里巴巴数据库内核研发团队，主导AliSQL的产品研发（AliSQL：开源MySQL的阿里分支）以及下一代数据库系统的规划和研发工作。

- **联系方式**
  - ✓ 微博：@何_登成

Agenda

AliSQL发展历史简要回顾

X-KV：高性能K-V接口

X-Cluster：AliSQL集群解决方案

AliSQL?

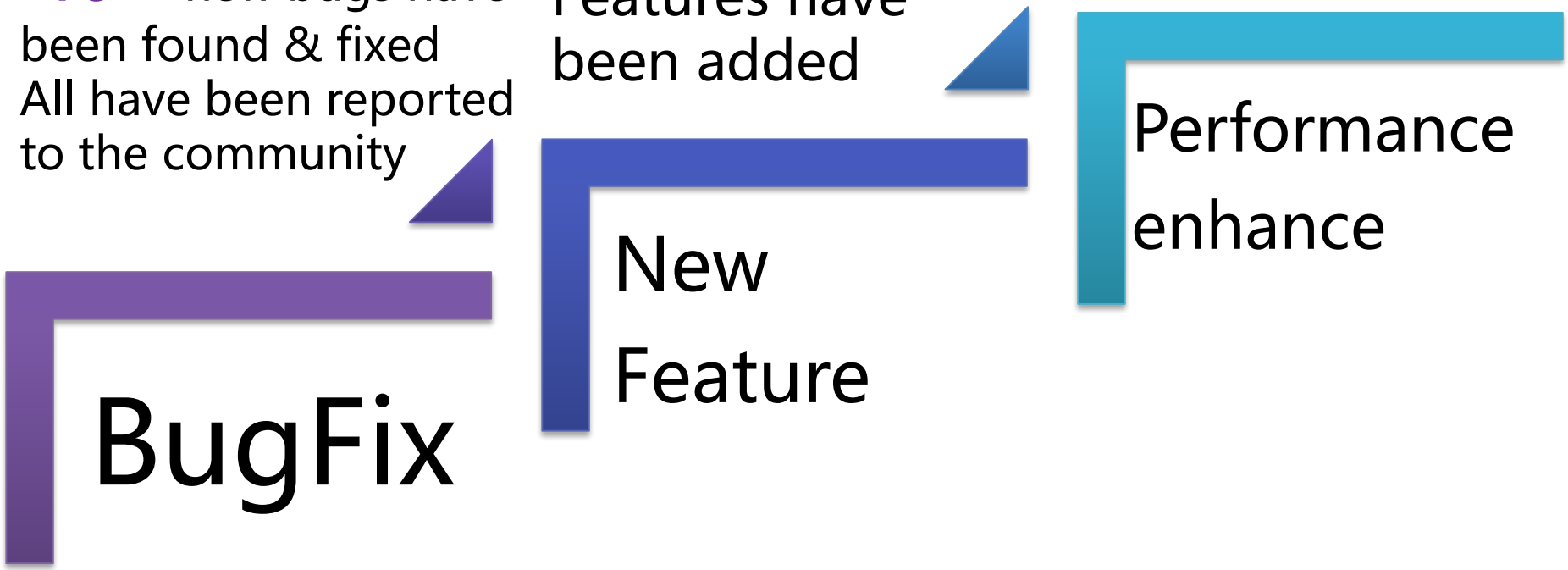
- Alibaba的MySQL分支
  - ✓ Since 2010
- 我们为什么发展一个MySQL分支?
  - ✓ 性能
  - ✓ 功能
  - ✓ 可运维性

AliSQL：成果展示

- 40+ new bugs have been found & fixed
- All have been reported to the community

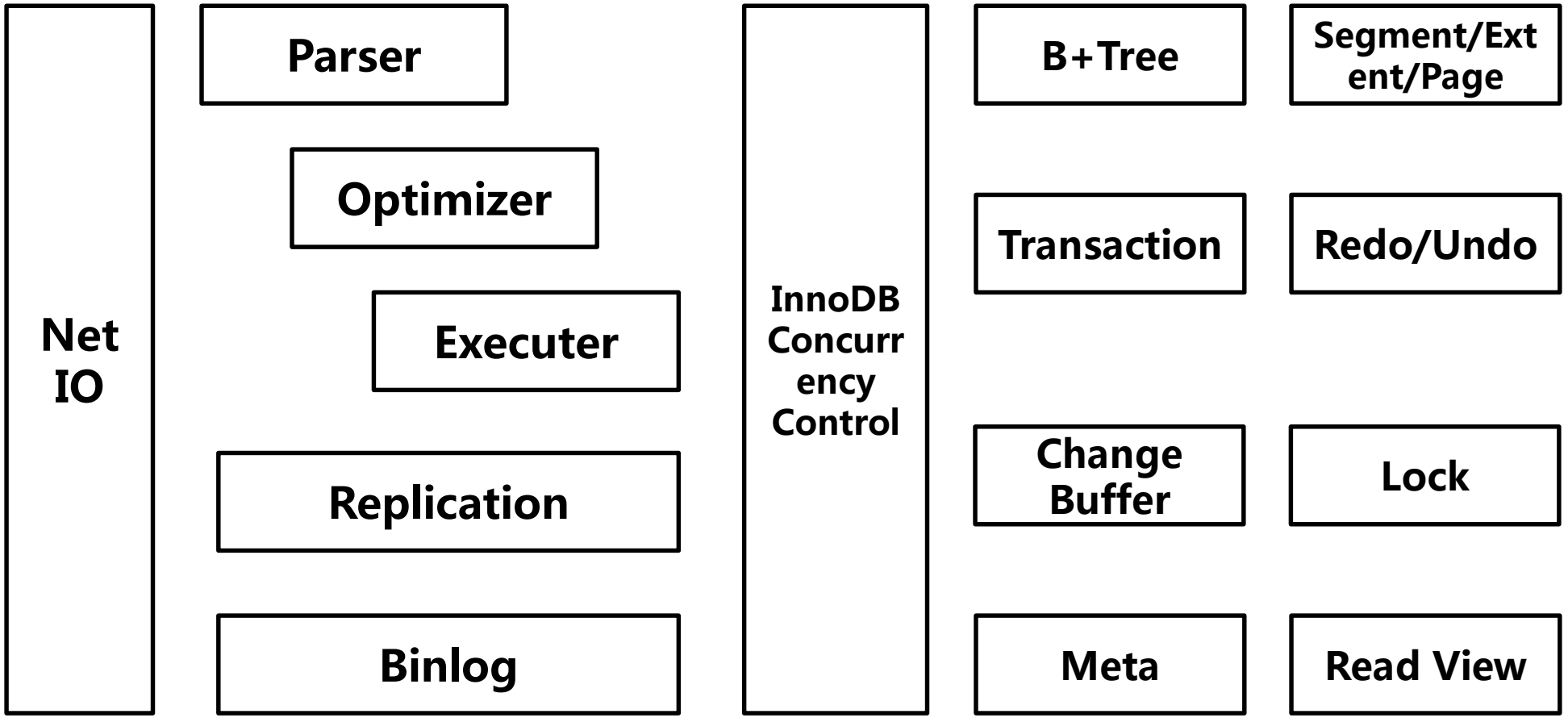
- 40+ new Features have been added

- 30+ bottlenecks have been optimized

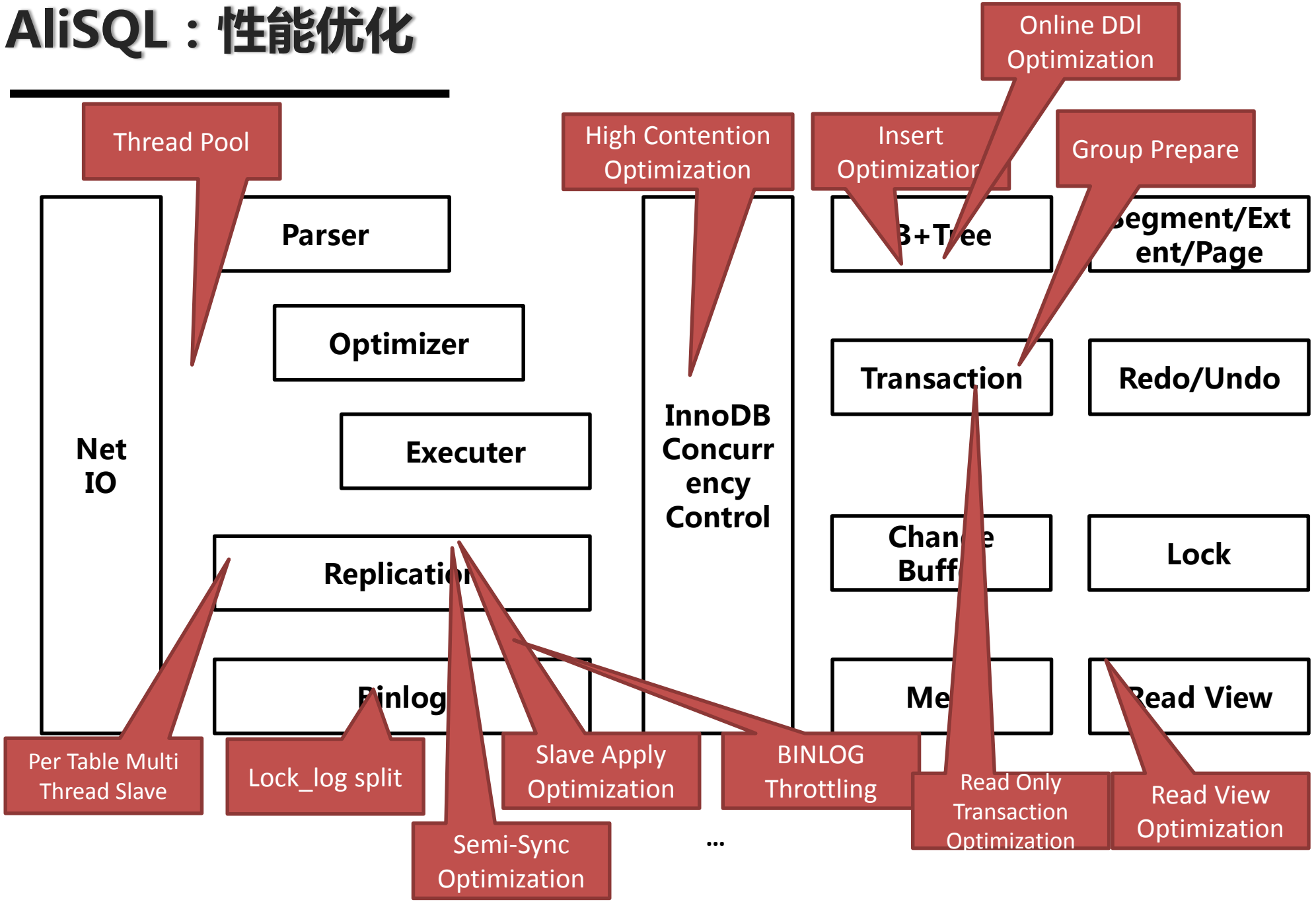


- Colin Charles Charles (2016). AliSQL and some features that have made it into MariaDB Server

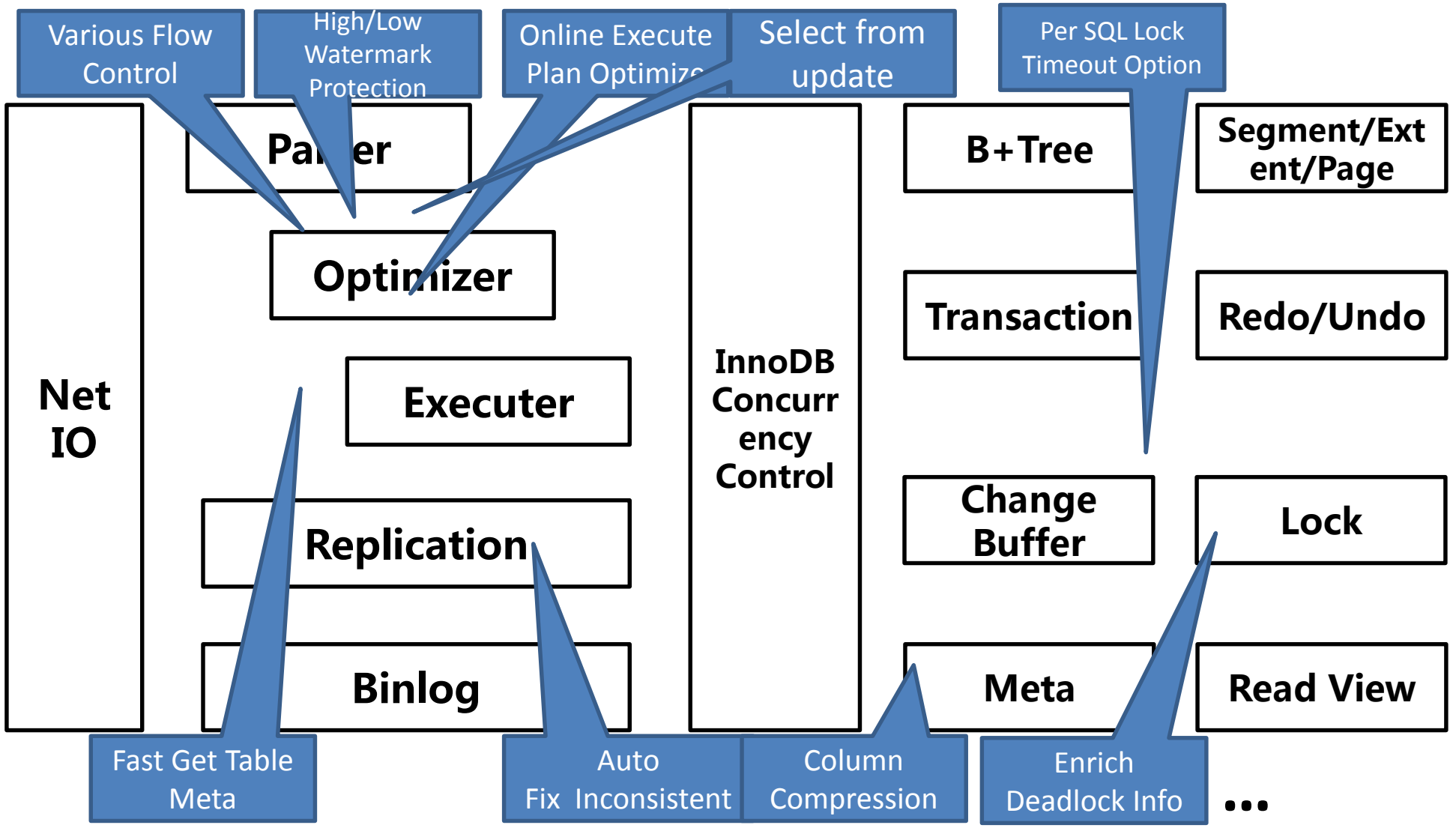
AliSQL：精简架构


Net IO
Parser
Optimizer
Executer
Replication
Binlog
InnoDB Concurrency Control
B+Tree
Segment/Extent/Page
Transaction
Redo/Undo
Change Buffer
Lock
Meta
Read View

AliSQL：性能优化


Online DDl Optimization
Thread Pool
High Contention Optimization
Insert Optimization
Group Prepare
Parser
Optimizer
Executer
Replication
Binlog
Net IO
InnoDB Concurrency Control
B+Tree
Segment/Extent/Page
Transaction
Redo/Undo
Change Buff
Lock
Me
Read View
Per Table Multi Thread Slave
Lock_log split
Semi-Sync Optimization
Slave Apply Optimization
BINLOG Throttling
...
Read Only Transaction Optimization
Read View Optimization

AliSQL：功能 & 可运维性增强



Agenda

AliSQL发展历史简要回顾

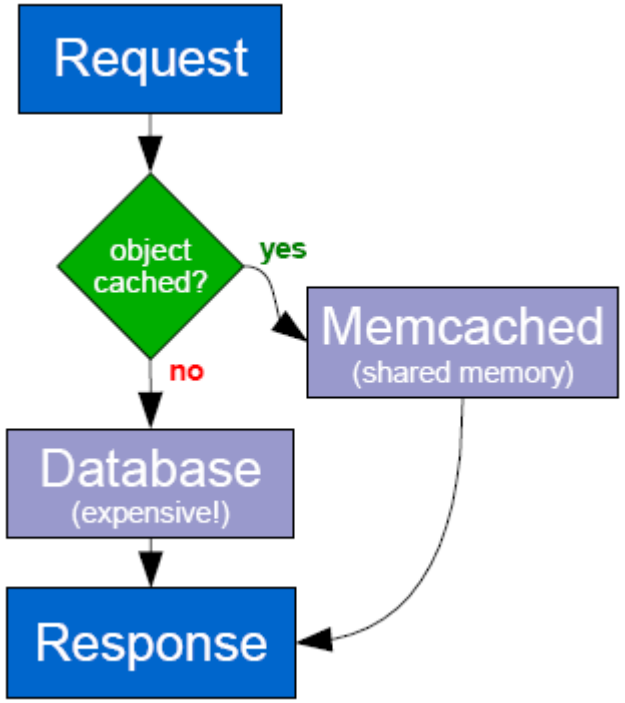
X-KV：高性能K-V接口

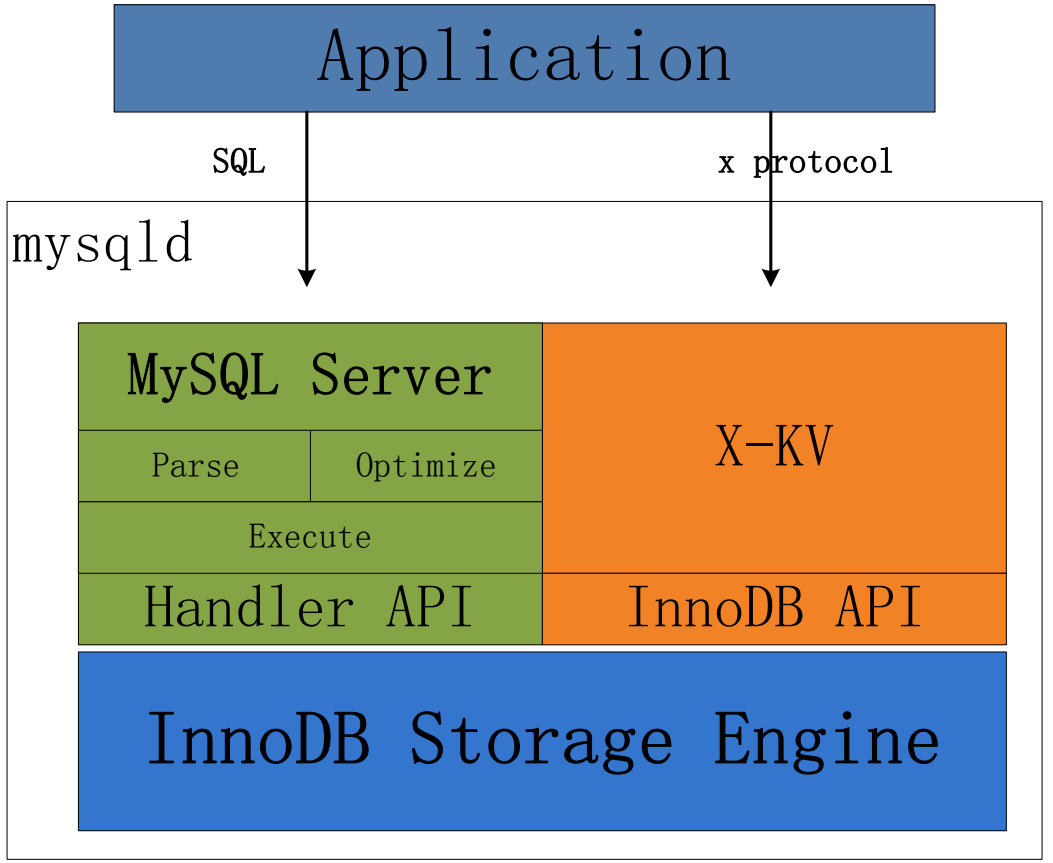
X-Cluster：AliSQL集群解决方案

什么是X-KV？

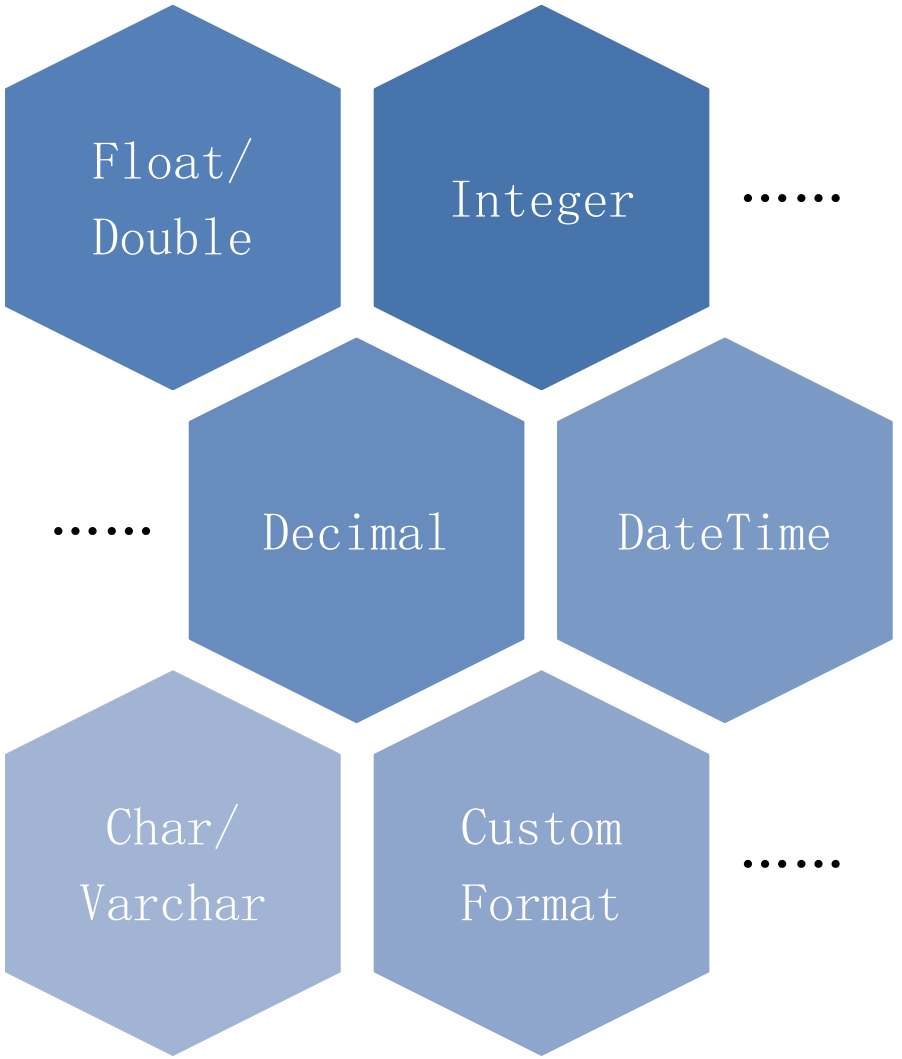
- **X-KV**
  - ✓ AliSQL高性能K-V接口，InnoDB Memcached Plugin的扩展

- **为什么需要X-KV？**
  - ✓ Query Performance
  - ✓ Data Consistency

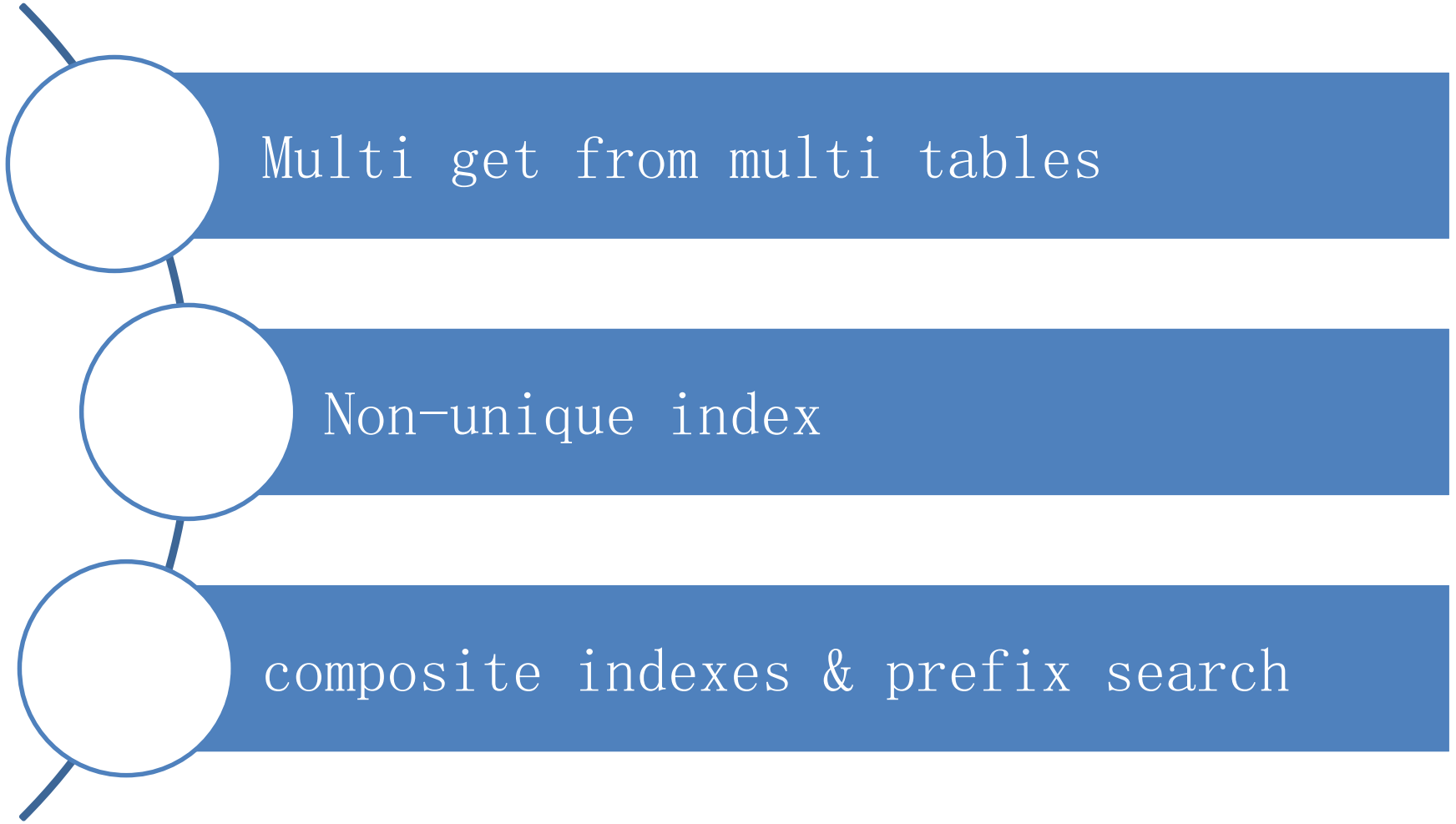




X-KV：Data Type支持增强



X-KV：功能增强


Multi get from multi tables
Non-unique index
composite indexes & prefix search

X-KV：新协议

- **InnoDB Memcached Plugin存在的问题**
  - ✓ 通过指定delimiter来区分每一列： delimiter如何选择？
  - ✓ NULL和空值无法区分：NULL = 空

PK		NAME		WORKING PLACE	
1	|	He dengcheng	|	阿里巴巴西溪园区8号楼	|

- **X-KV：新协议**
  - ✓ Field = Meta Info + Data
  - ✓ Meta Info
    - Version|Count|Length1|Length2|Length3…

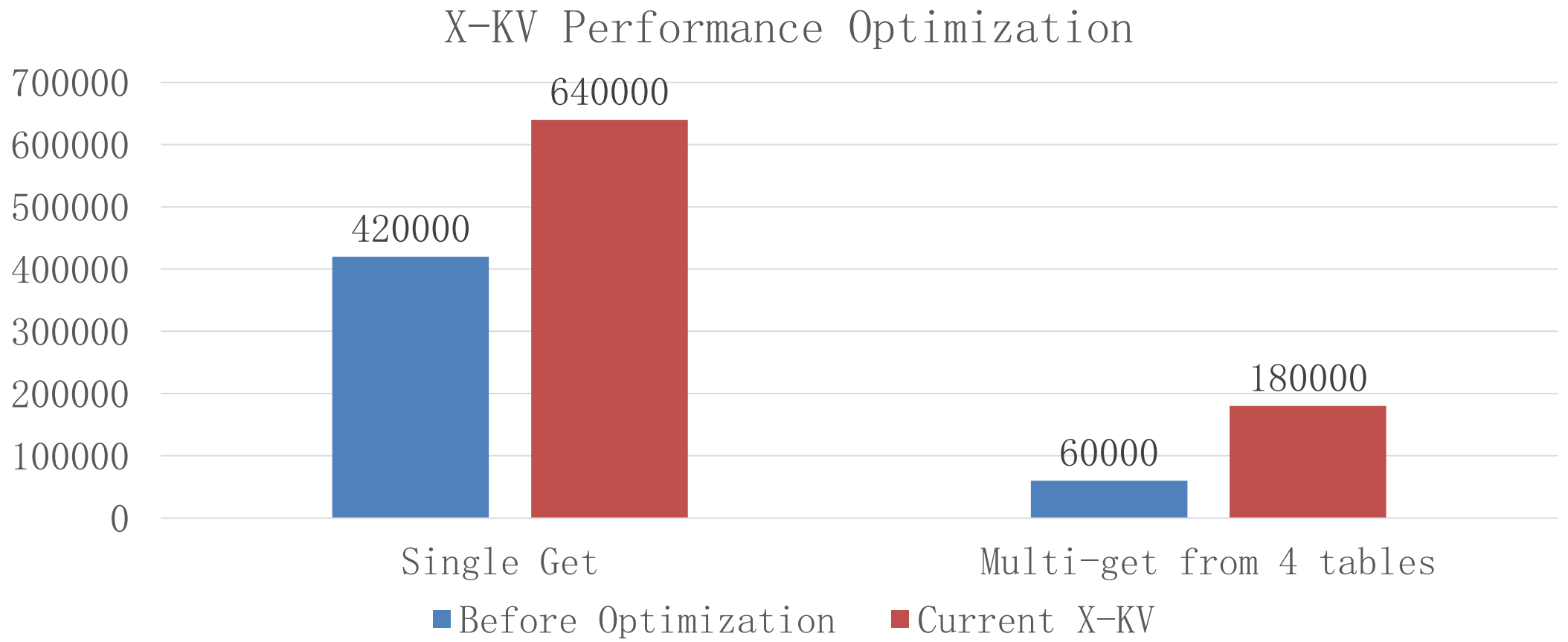
					PK	NAME	WORKING PLACE
0	3	1	12	31	1	He dengcheng	阿里巴巴西溪园区8号楼

X-KV：可运维性优化

- **InnoDB Memcached Plugin存在的问题**
  - ✓ 修改Container表：Uninstall -> Install，生效
  - ✓ 运维操作对业务有较大影响

- **X-KV：运维优化**
  - ✓ Container表新增K-V读取配置：直接生效
  - ✓ Container表修改原有配置：通过新协议的Version来生效
  - ✓ DDL，自动重新加载

X-KV：性能优化



- **测试场景**
  - ✓ 模拟阿里的交易数据库上的Query请求（优化后：网卡和CPU瓶颈）

Agenda

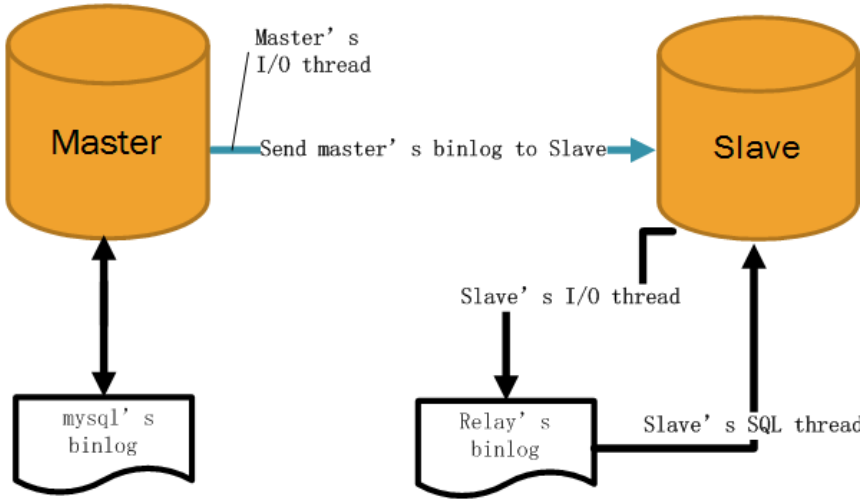
AliSQL发展历史简要回顾

X-KV：高性能K-V接口

X-Cluster：AliSQL集群解决方案

AliSQL/MySQL：Drawback

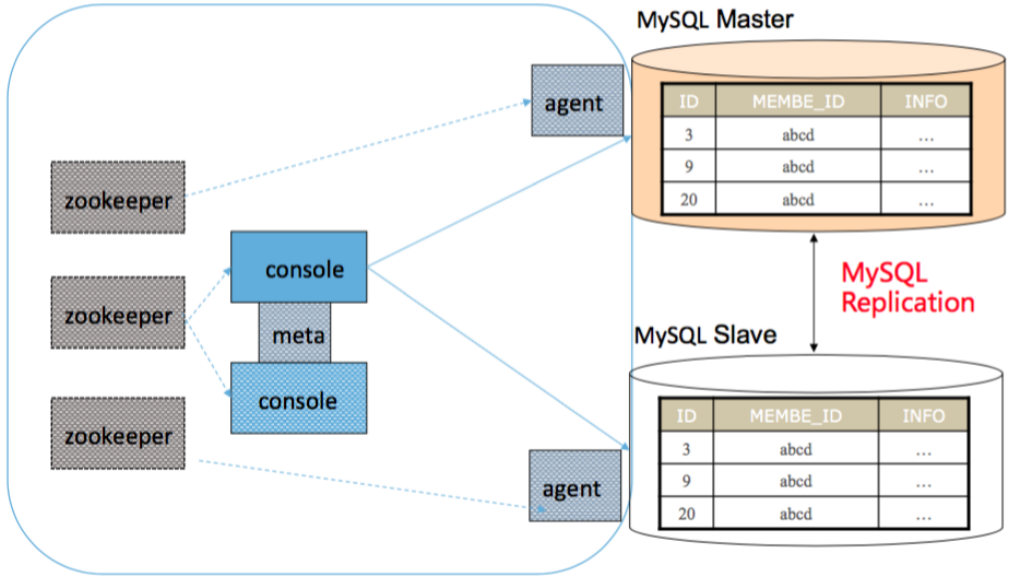
- **Why X-Cluster?**
  - ✓ 数据一致性
    - 异步复制
    - 半同步复制



  - ✓ 持续高可用

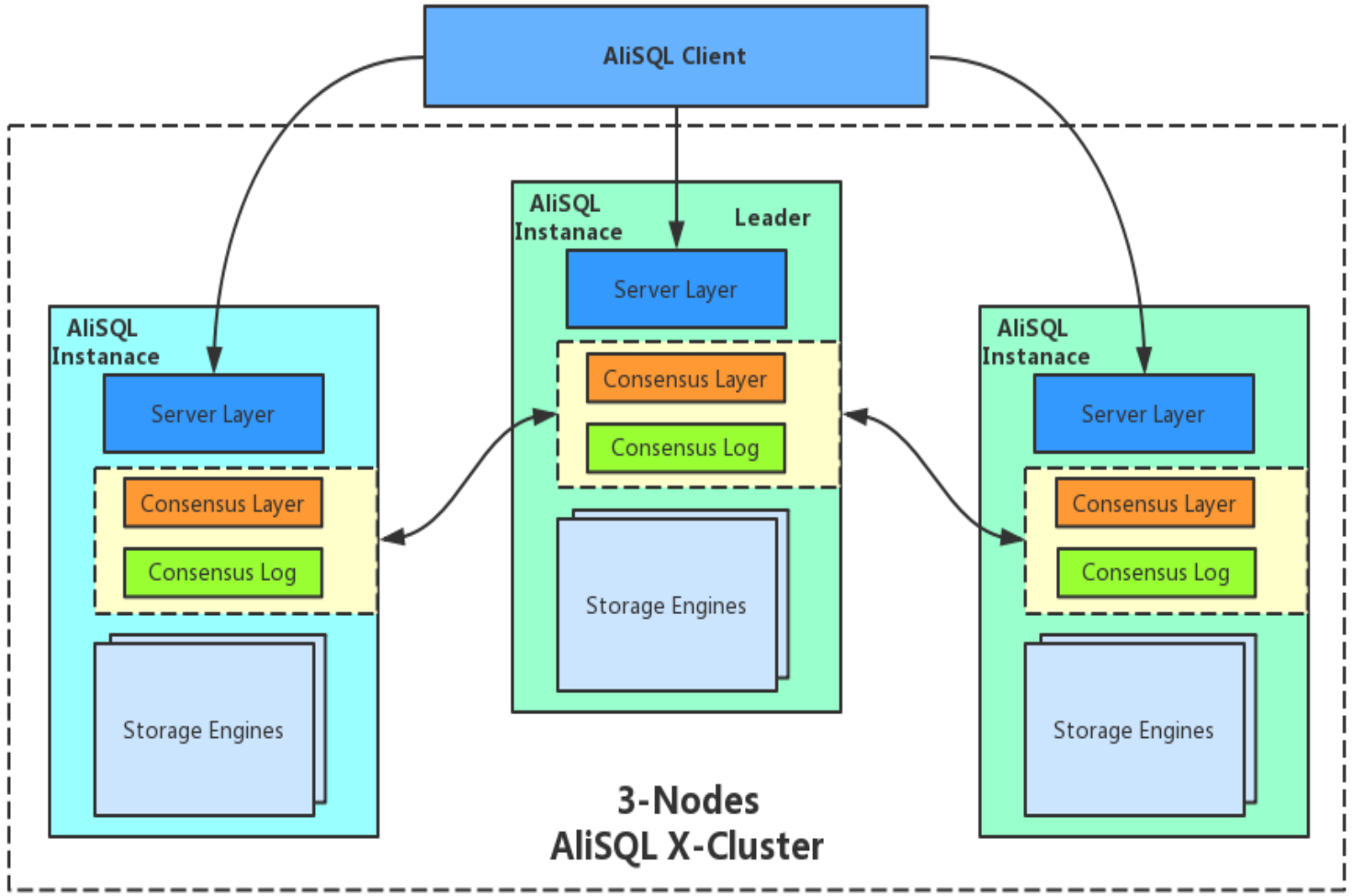
  - ✓ 区域化/全球化部署

  - ✓ 上下游生态联动



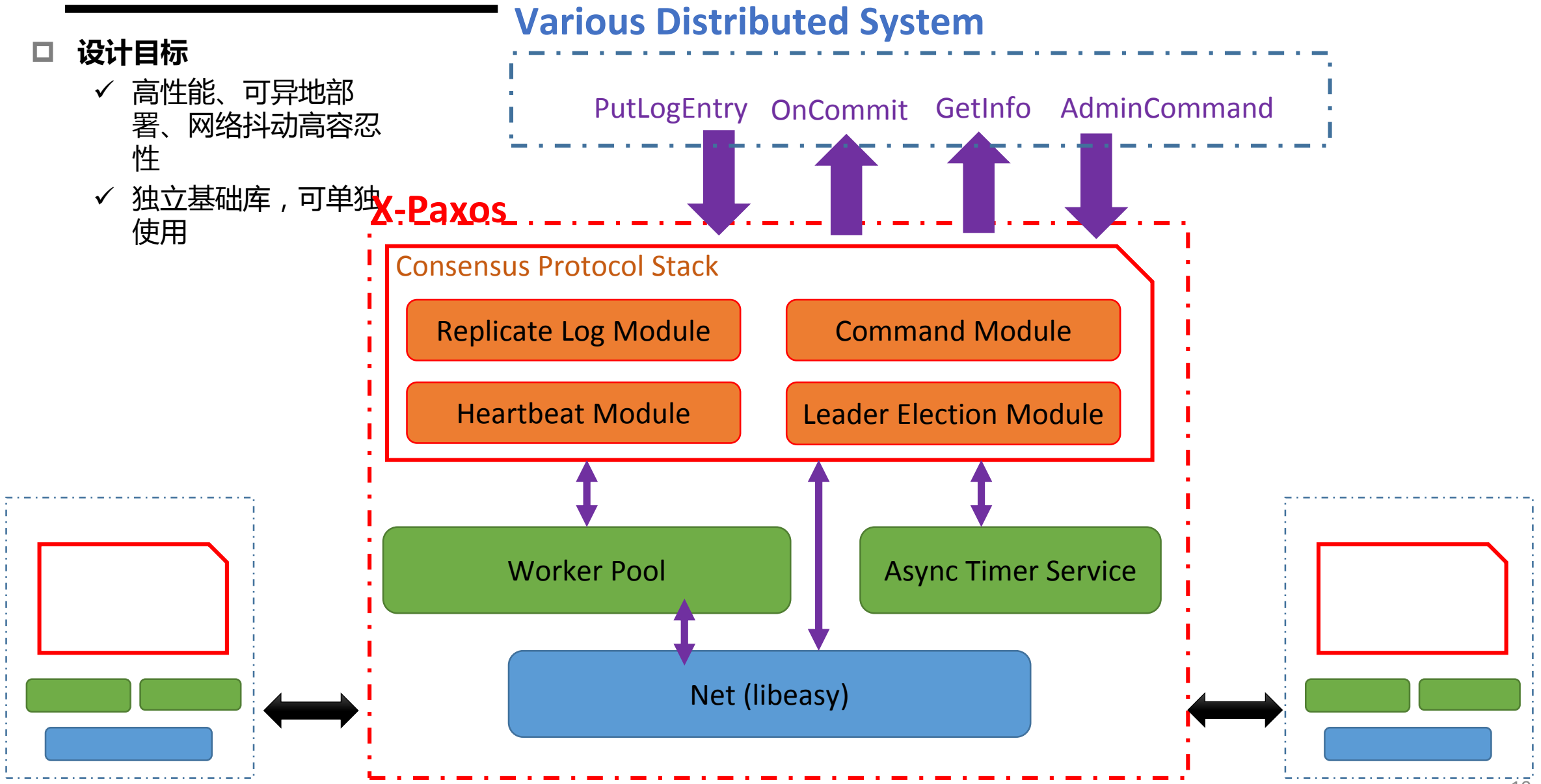
X-Cluster：Based on AliSQL

- 设计目标
  - ✓ **一体化架构**：运维友好
  - ✓ **极致性能**：同城三副本相对于单机性能下降在10%以内
  - ✓ **可异地部署**：异地部署，延时增加，但是保持高吞吐
  - ✓ **稳定性**：网络抖动高容忍性
  - ✓ **兼容性**：对原有生态100%兼容

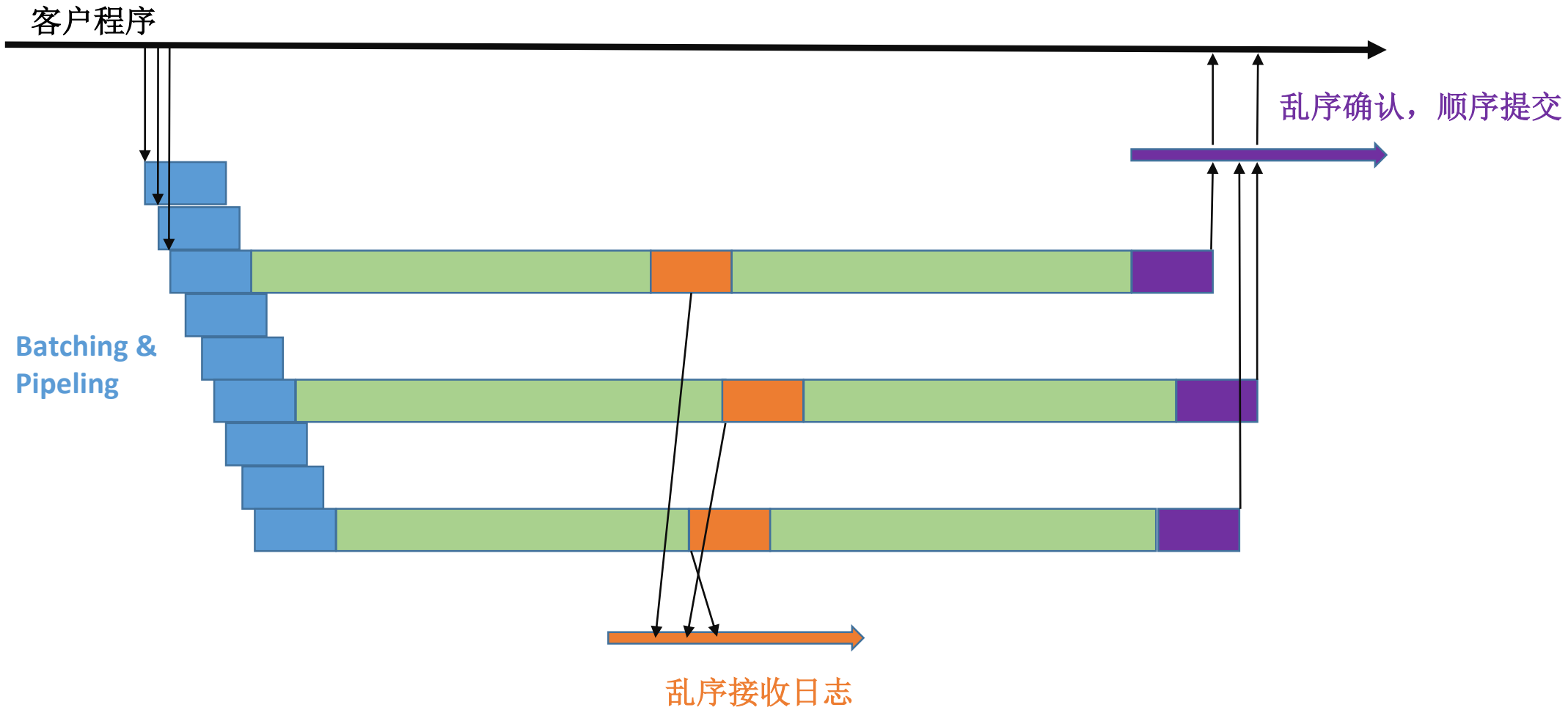


X-Cluster核心组件：X-Paxos

- **设计目标**
  - ✓ 高性能、可异地部署、网络抖动高容忍性
  - ✓ 独立基础库，可单独使用



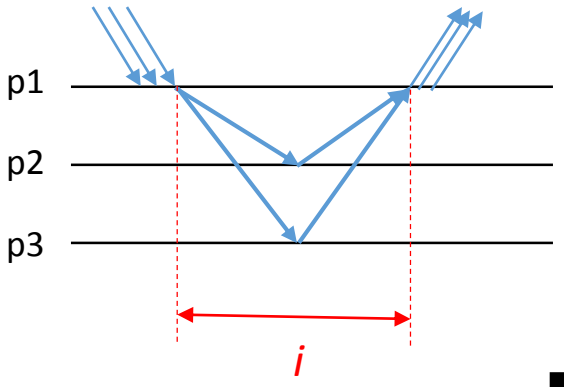
X-Cluster核心技术：Batching & Pipelining



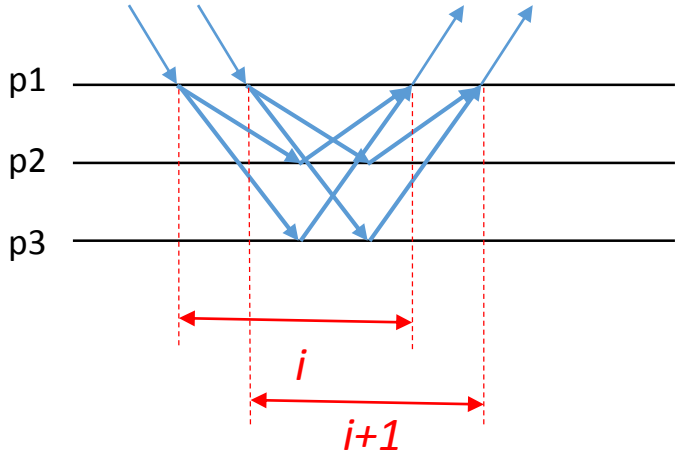
- Tuning paxos for high-throughput with batching and pipelining (ICDCN12)

X-Cluster核心技术：Batching & Pipelining（续）

Batching

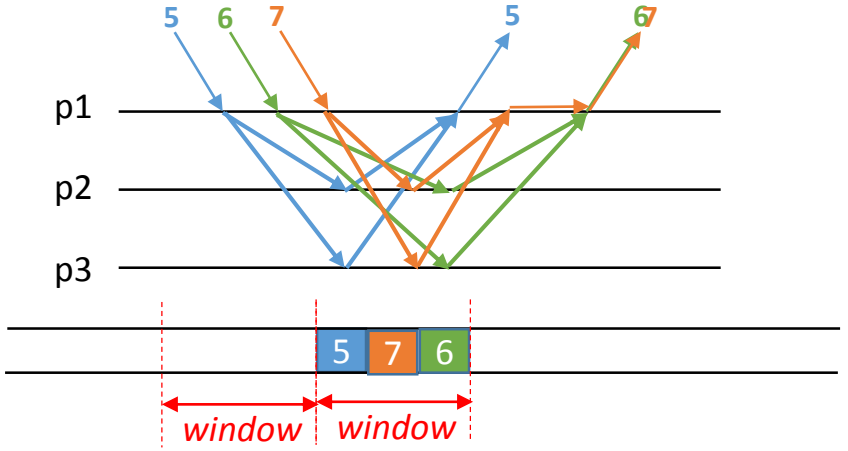


Pipelining



Log Reorder

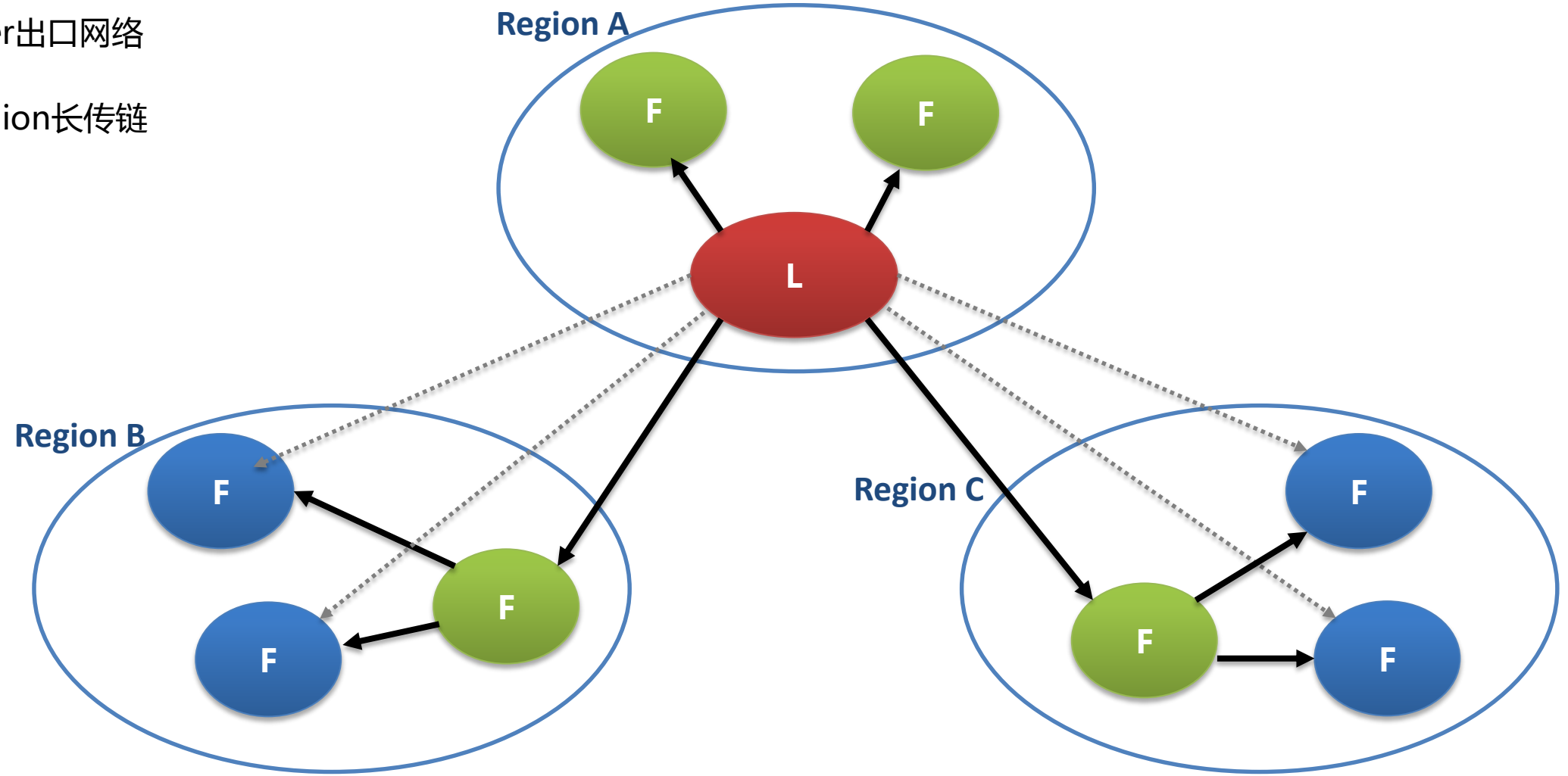
乱序确认顺序提交



乱序接收日志

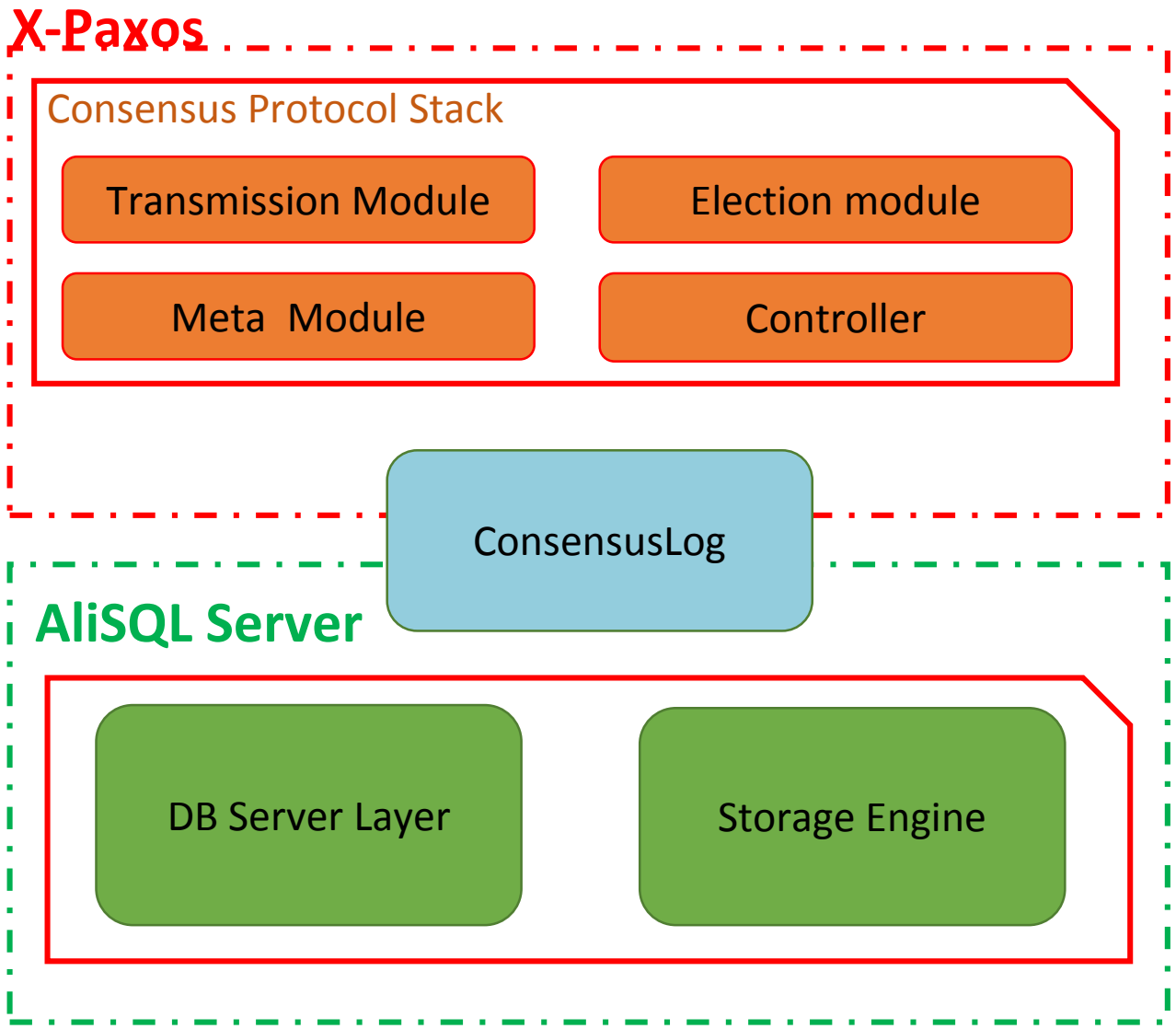
X-Cluster核心技术： Locality Aware Content Distribution

- **核心优势**
  - ✓ 解决Leader出口网络瓶颈
  - ✓ 降低跨Region长传链路带宽

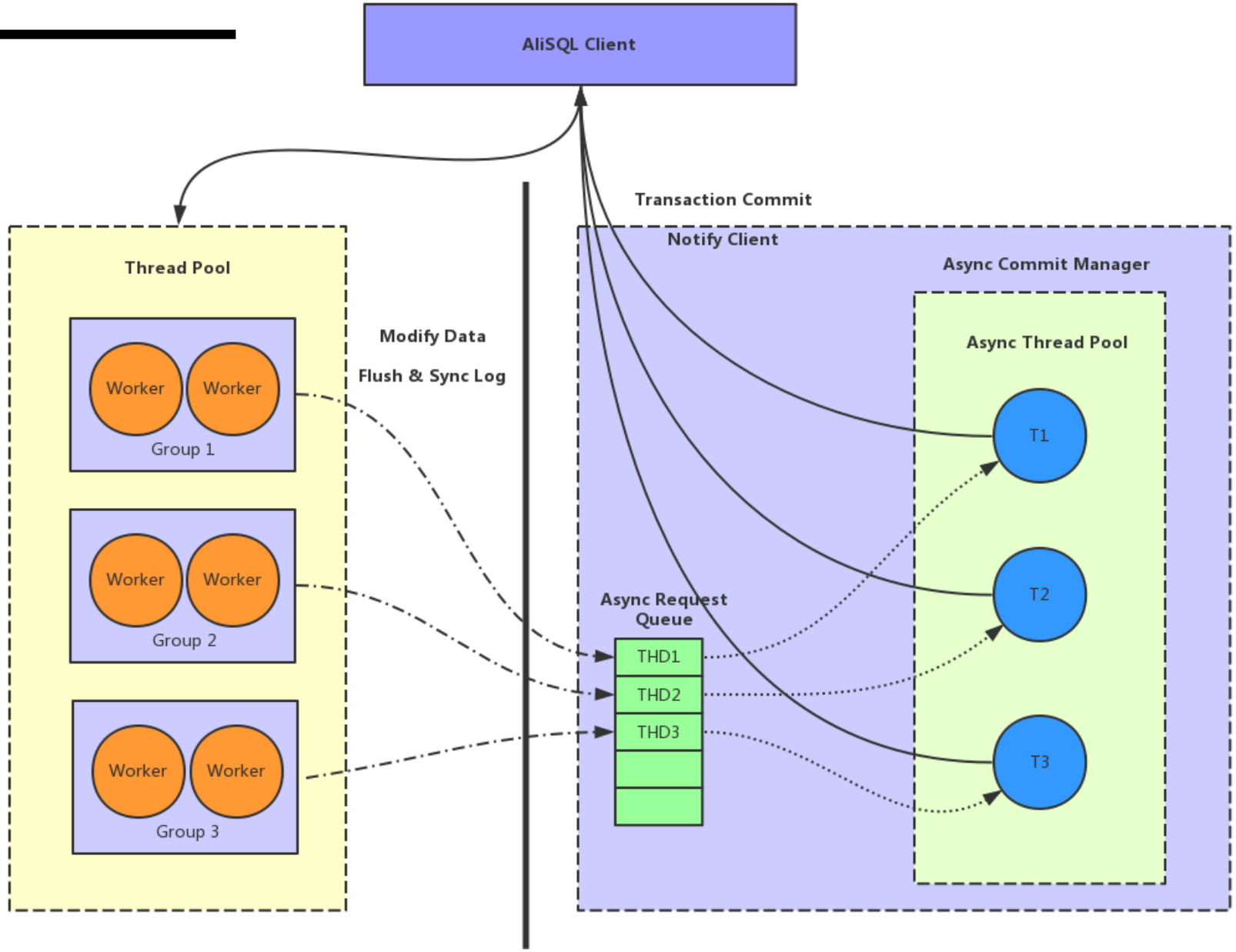


X-Cluster核心技术：日志实现

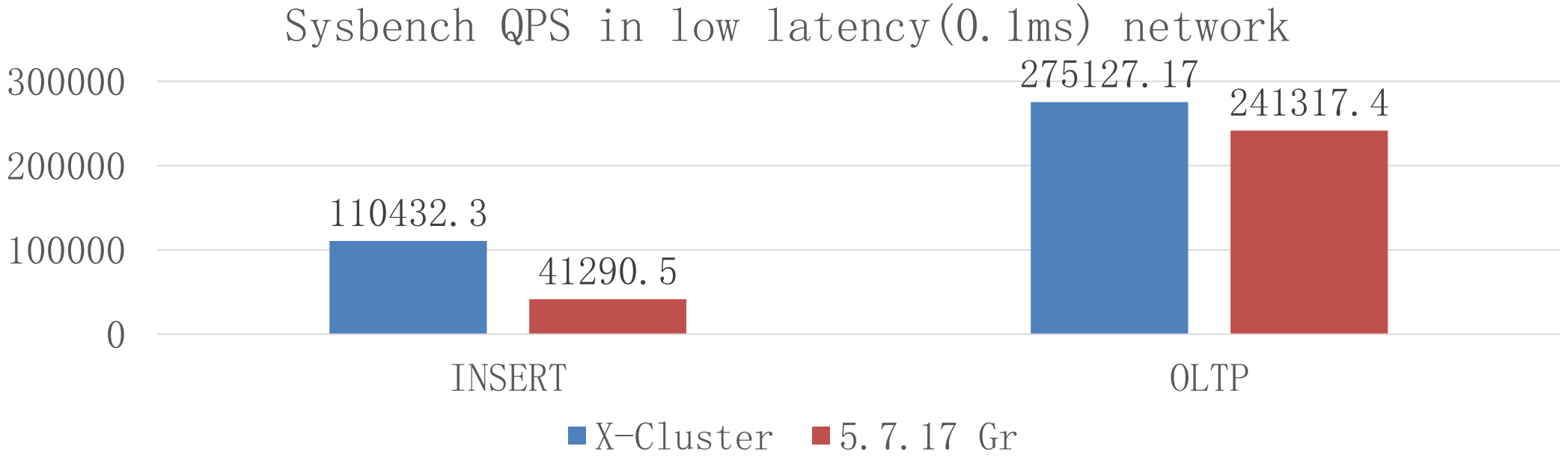
- **核心技术**
  - ✓ 插件式X-Paxos的日志
  - ✓ 归一化的ConsensusLog代替Binlog和RelayLog
  - ✓ 全局统一的Log Index

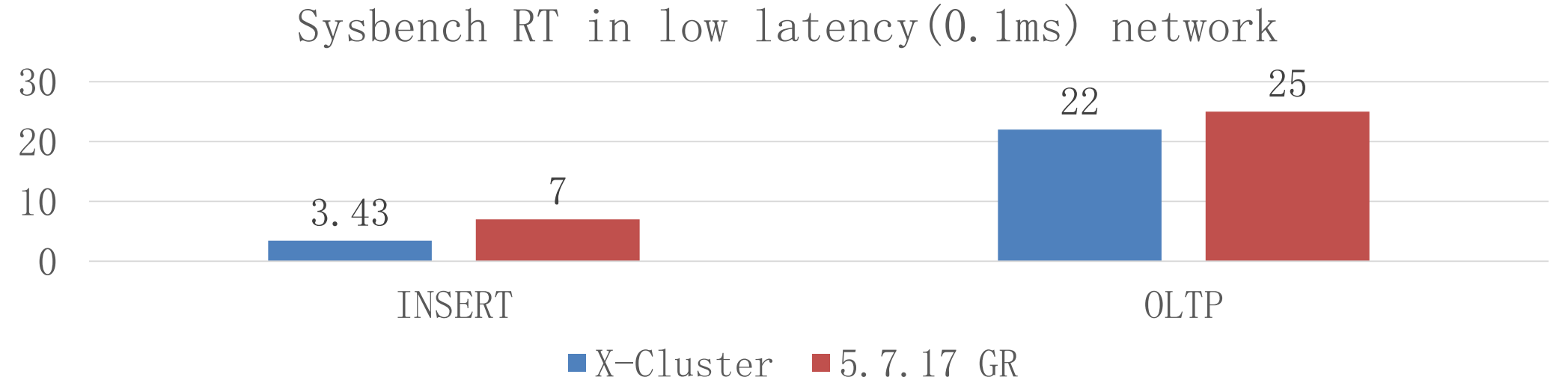


X-Cluster核心技术: Asynchronously Commit

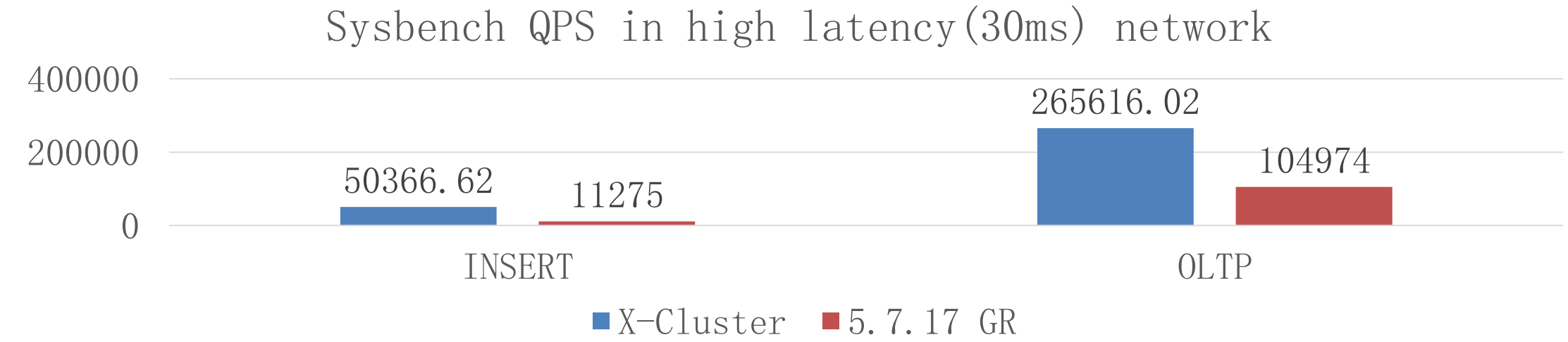


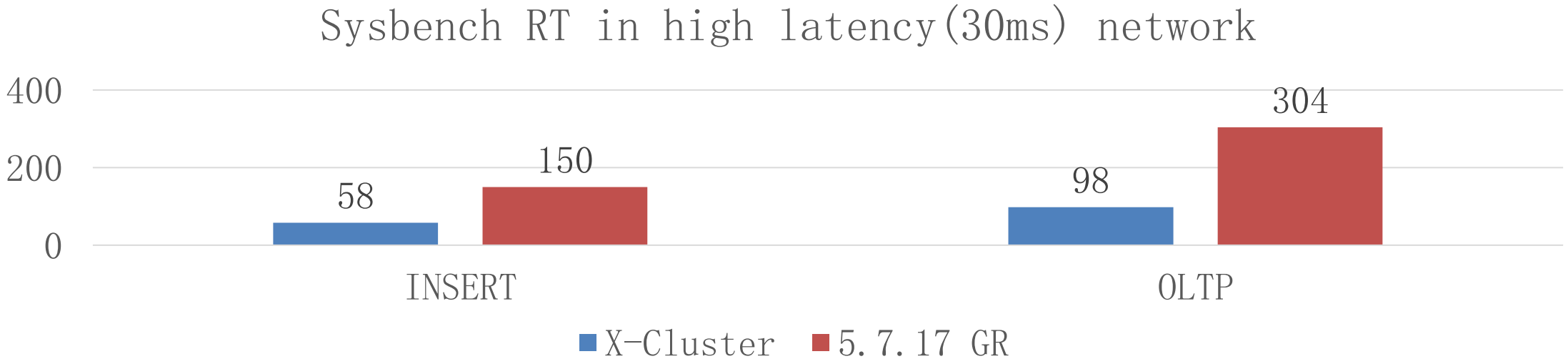
X-Cluster: vs MySQL Group Replication



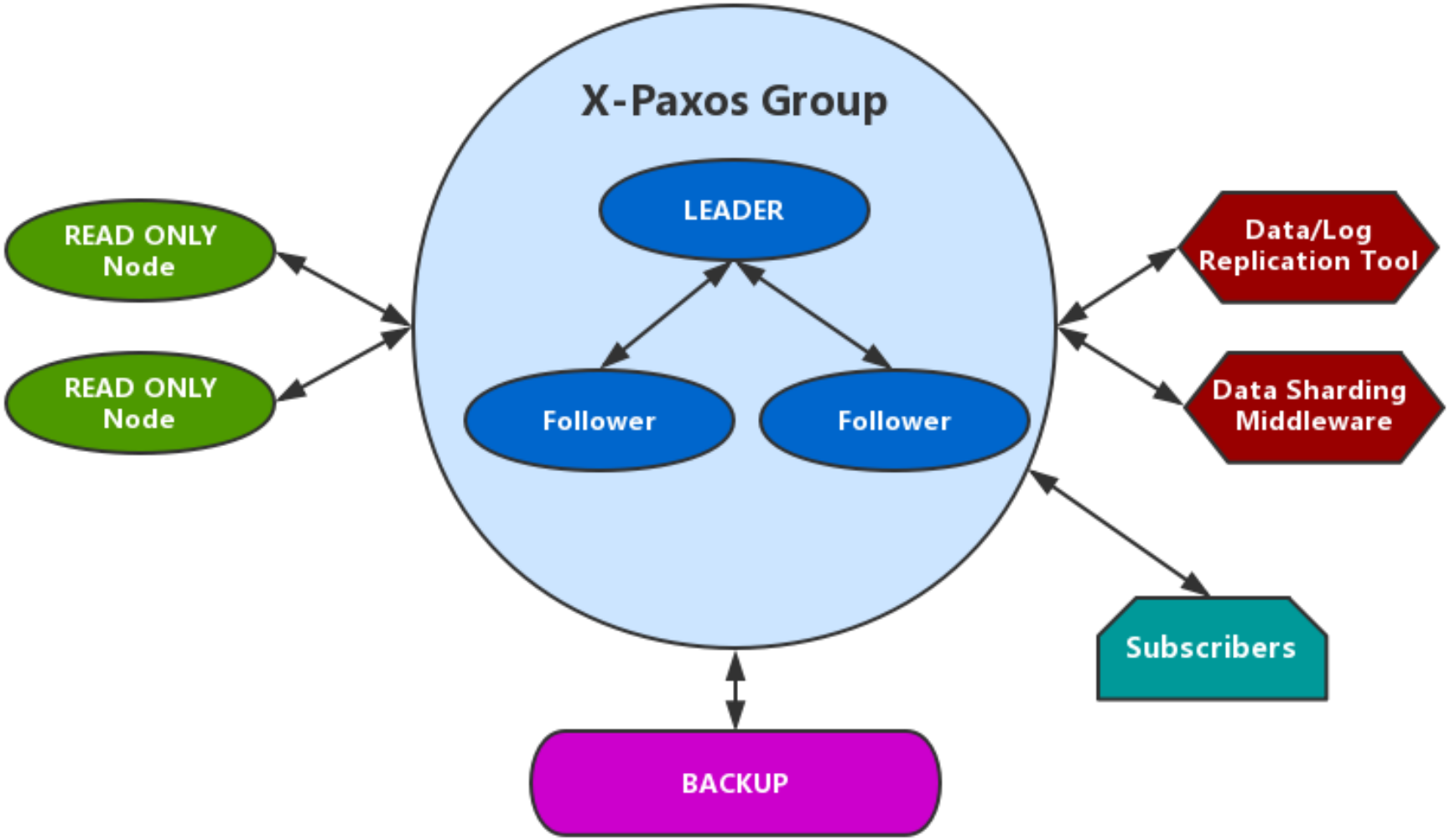


X-Cluster: vs MySQL Group Replication





X-Cluster生态：超越数据一致性和持续可用



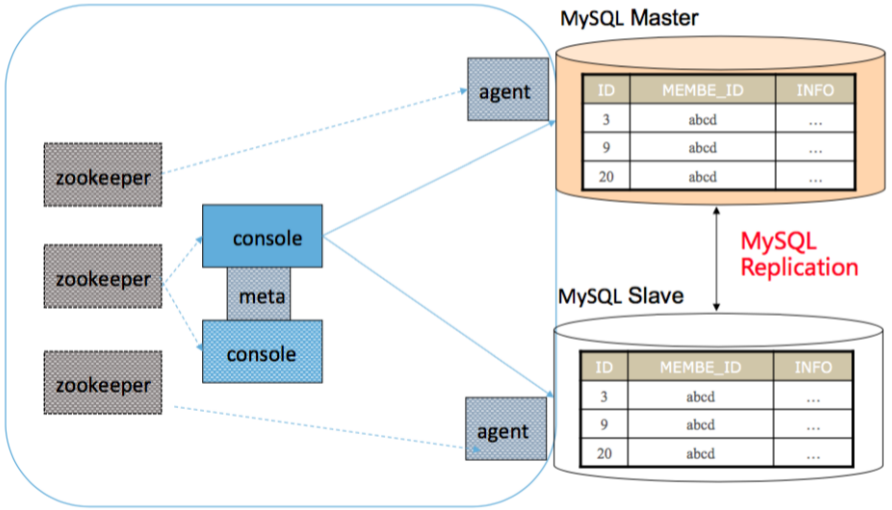
X-Cluster生态：持续备份

- **原有MySQL备份逻辑**
  - ✓ 定期备份Binlog文件
  - ✓ RPO一般比较大，例如：大于5分钟
  - ✓ 备份跟MySQL的数据一致性保障困难

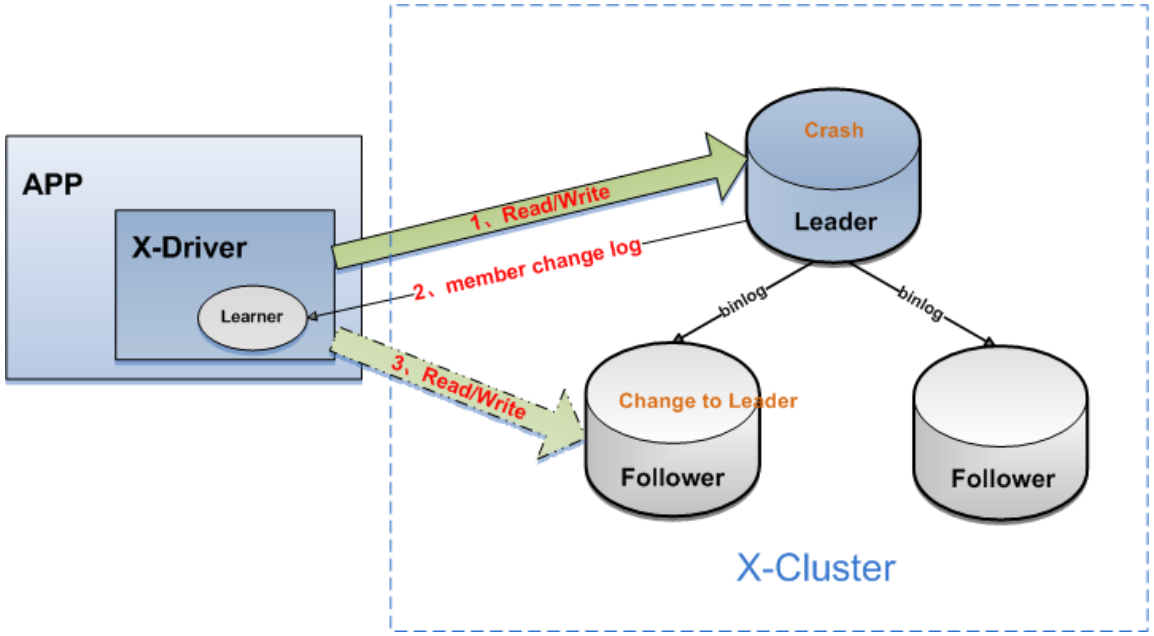
- **X-Cluster：持续备份**
  - ✓ 备份节点，作为X-Cluster的一个Learner节点。实时推送X-Cluster上达成多数派的日志
  - ✓ **RPO < 1秒**
  - ✓ 由于备份的一定是达成多数派的日志，因此无数据一致性问题

X-Cluster生态：自动化高可用

- **原有MySQL高可用方案**
  - ✓ 外部组件依赖：ADHA、ZK



- **X-Cluster：自动化高可用**
  - ✓ Client、Server一体化，No 外部组件依赖



X-Cluster生态：自动化增量日志消费

- **原有MySQL下游日志消费**
  - ✓ 准实时消费MySQL产生的日志
  - ✓ 问题之一：数据一致性
  - ✓ 问题之二：数据库主备切换与下游消费端的联动

- **X-Cluster：自动化增量日志消费**
  - ✓ 日志消费节点作为X-Cluster的Learner节点
  - ✓ 只消费达成多数派的日志：数据一致性
  - ✓ X-Cluster自动选主，新Leader自动向日志消费节点推送新日志：彻底解决联动问题

X-Cluster生态：区域化/全球化部署

- **按需增加Learner节点**
  - ✓ 增加读能力，但是不会带来强同步开销
- **按需增加Loger节点**
  - ✓ Loger节点只有日志，没有数据。
  - ✓ Loger节点可参与选主，但是没有新增存储开销。低成本节点。
  - ✓ 3节点X-Cluster = 2节点MySQL主备
- **权重化体系**
  - ✓ 可以指定节点选主权重，控制每个节点的选主优先级

X-Cluster实战：实战中踩过的坑，总结

- **异常处理**
  - ✓ 硬件异常
  - ✓ 网络异常：Leader Stickiness

- **Batching & Pipelining**
  - ✓ 极大事务
  - ✓ 极小事务
  - ✓ 不同网络时延下的Batching/Pipelining策略
  - ✓ 网络异常情况下的Batching/Pipelining策略

- **全球化部署下的优化**
  - ✓ 权重体系
  - ✓ 热点带来的影响

写在最后

联系方式

- ✓ 微博： 何_登成
- ✓ Linkedin： he dengcheng
- ✓ 邮箱： dengcheng.hedc@gmail.com

欢迎大家的骚扰和交流☺

谢谢大家