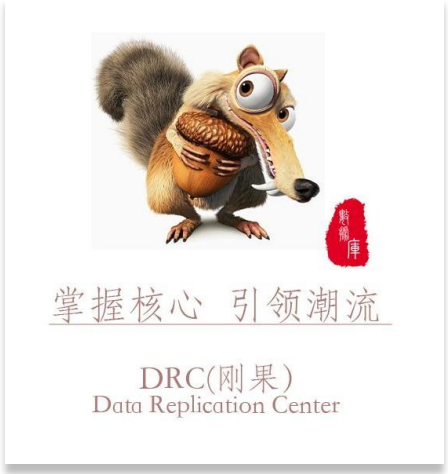
数据复制中心DRC



@tb杰睿
阿里集团DBA数据方案

2013年8月8日

摘要

- 简介和应用场景
- 架构和性能
- 技术难点和挑战
- 总结

简介

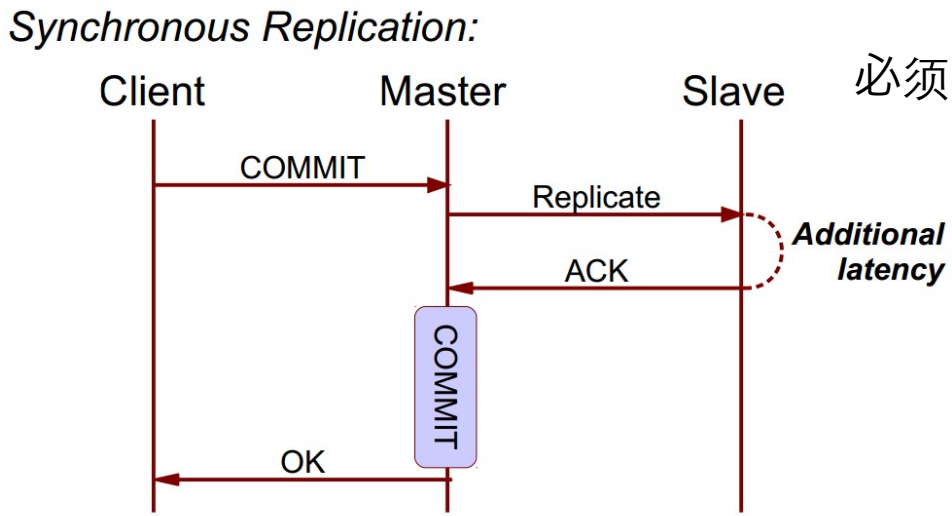
- @tb杰睿
- 数据方案组
- 数据复制中心Data Replication Center
  - 数据：OLTP数据源， 例如MySQL、OceanBase
  - 复制：高性能、 实时、事务一致
  - 中心：服务、集群、平台
- 应用场景
  - 多地域间数据库同步
  - 数据库增量消息分发

什么样的数据

- 数据库的增量数据
  - 数据变更
    - DML INSERT/UPDATE/DELETE
    - DDL CREATE/DROP/ALERT
    - DCL GRANT/REVOKE
  - 数据来源
    - MySQL Binlog
    - OceanBase Oblog

为什么需要数据复制

- 分布式集群
  - MegaStore
  - Galera
  - …
- 写入模型
  - 异步写
  - 多数写
  - 同步写



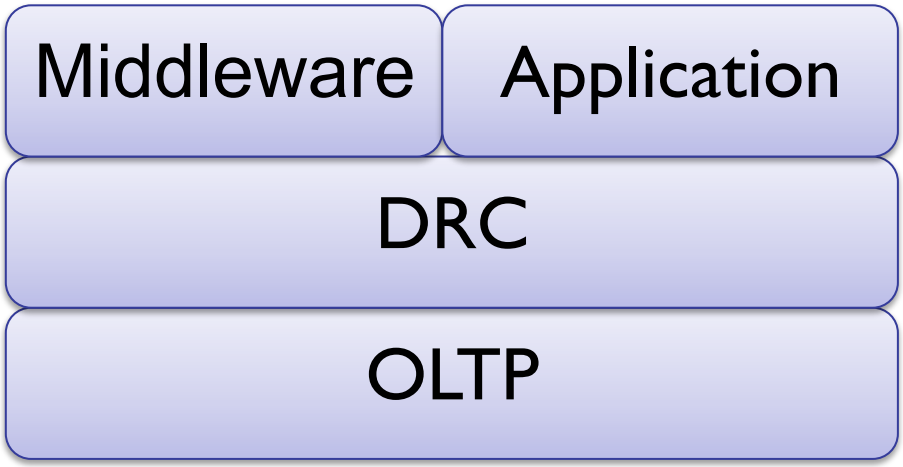
在线存储要求：
事务ACID
并发度高
响应时间低

测试结论：
在使用多数写模型的情况下性能将降低30%

必须要：异步写+**数据复制**

为什么是中心

- 工具？
- 函数库？



规模大
- 成千的数据库实例
- 数百的下游应用

配置复杂
- 双向复制
- 过滤规则

系统容灾
- 集群容灾
- 主备切换

数据安全
- 权限控制
- 安全审计

业务场景

- 跨地域的实时在线数据库同步服务
  - 南北机房
  - 机房内主备延迟
- 数据库的增量数据一对多分发服务
  - 商品、交易、评价
  - 广告、结算
  - 大数据计算
  - 缓存

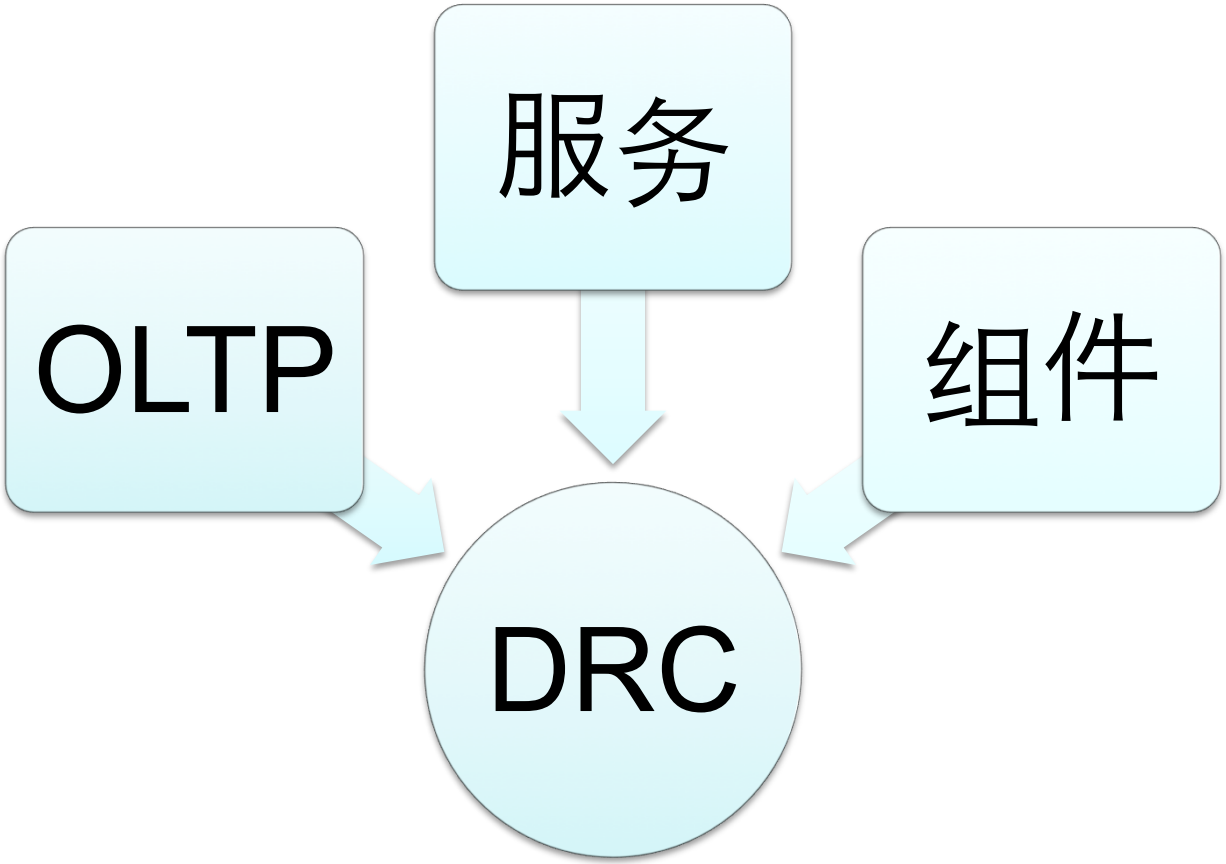
谁在用

- 搜索
- 广告
- 大数据计算
- 缓存失效

阿里集团-数据库技术团队

DRC的定位

- 阿里数据库的非常重要的底层基础组件



关键词

- 提供在线的数据复制服务
- 数据库的底层基础组件

服务

组件

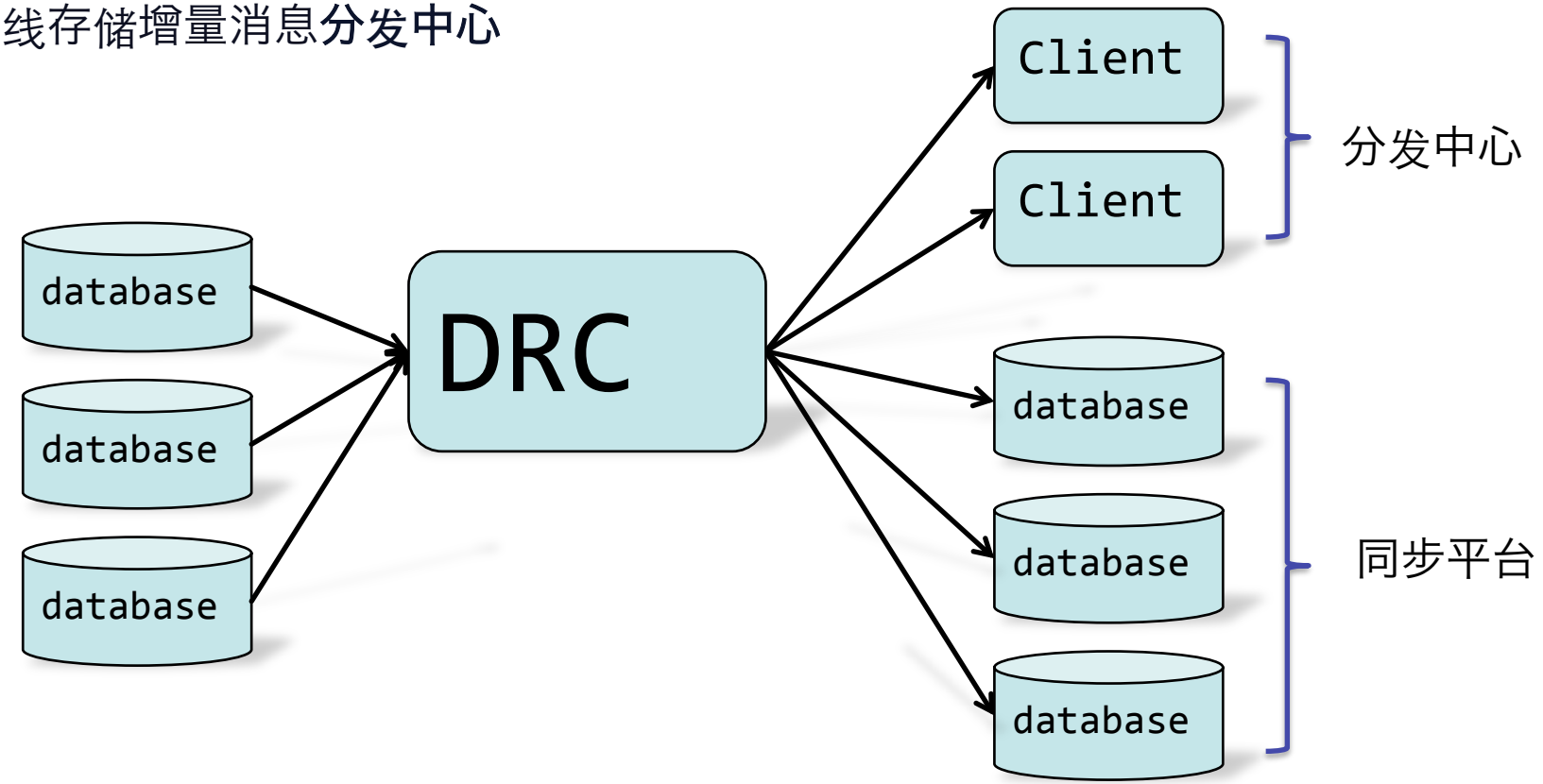
为什么是DRC

- 开源 or 其他产品
  - Tungsten
  - GoldenGate
  - MySQL5.6
  - Replicator Listener
  - MySQL Applier for Hadoop
  - …

用或者不用开源产品，定制化的业务需求都在那里，并没有本质的不同

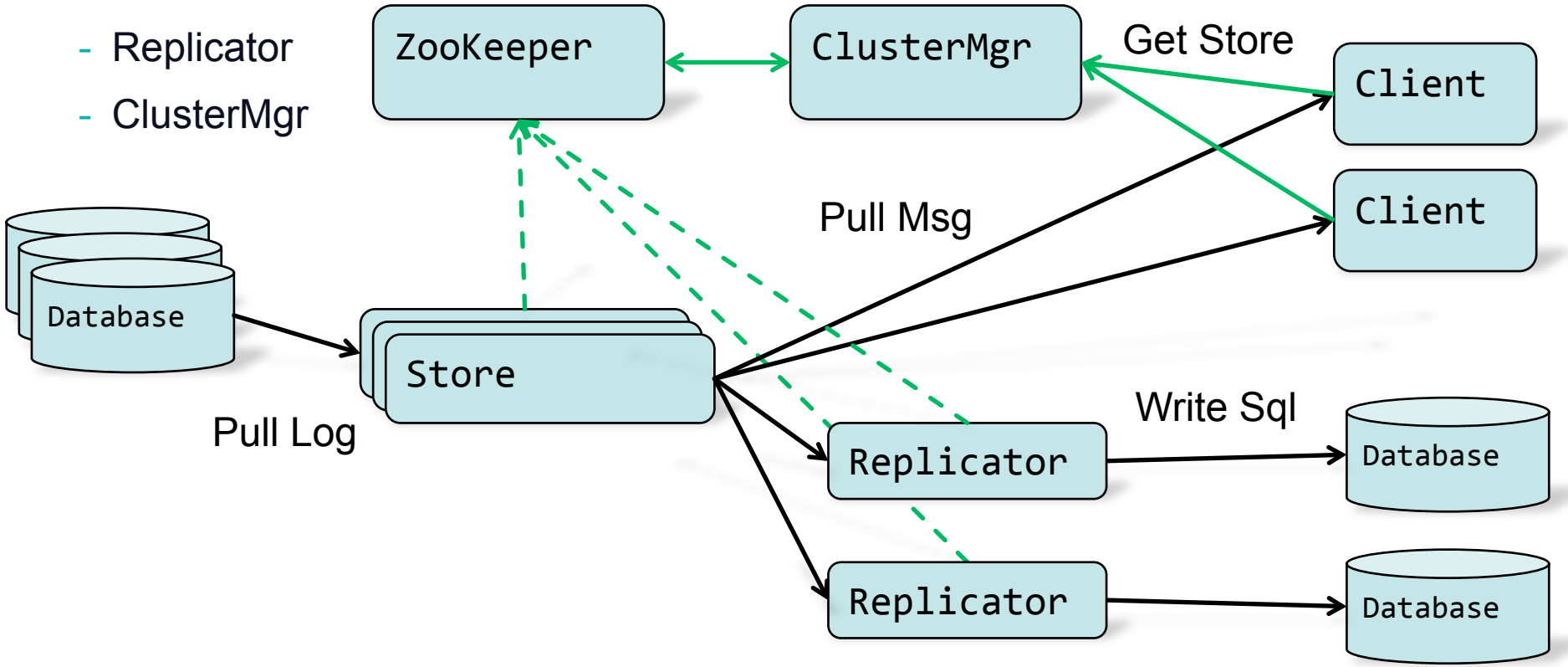
DRC长什么样

- 数据复制中心
  - 在线存储增量数据**同步平台**
  - 在线存储增量消息**分发中心**



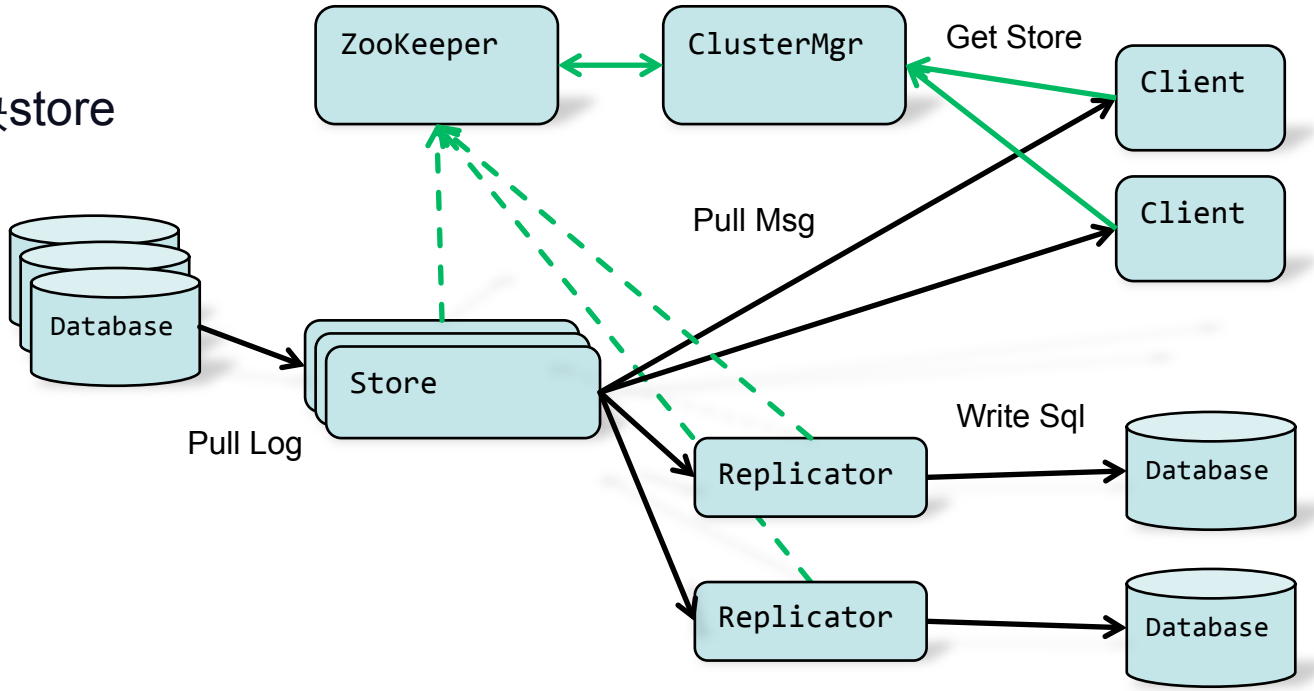
DRC的架构

- 架构
  - Zookeeper
  - Store
  - Replicator
  - ClusterMgr



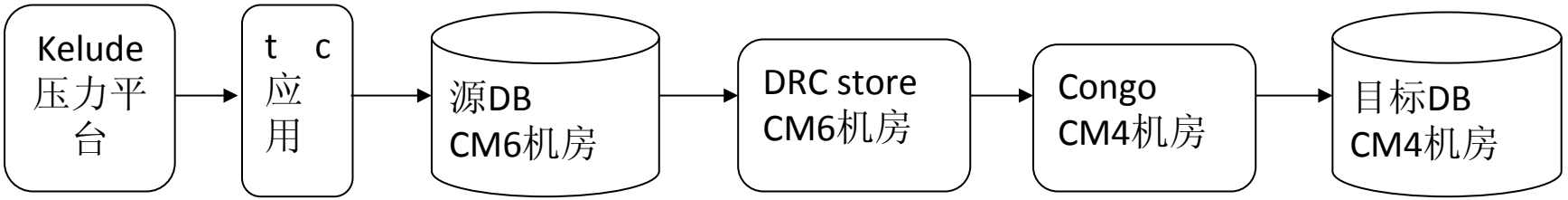
DRC的架构

- 概念
  - 主题topic——数据流
  - 集群管理模块clustermgr
    - 资源定位
    - 任务守护
  - 队列和分发引擎模块store
    - 日志解析
    - 数据持久化
    - 数据分发过滤
  - 同步模块replicator
  - 客户端client

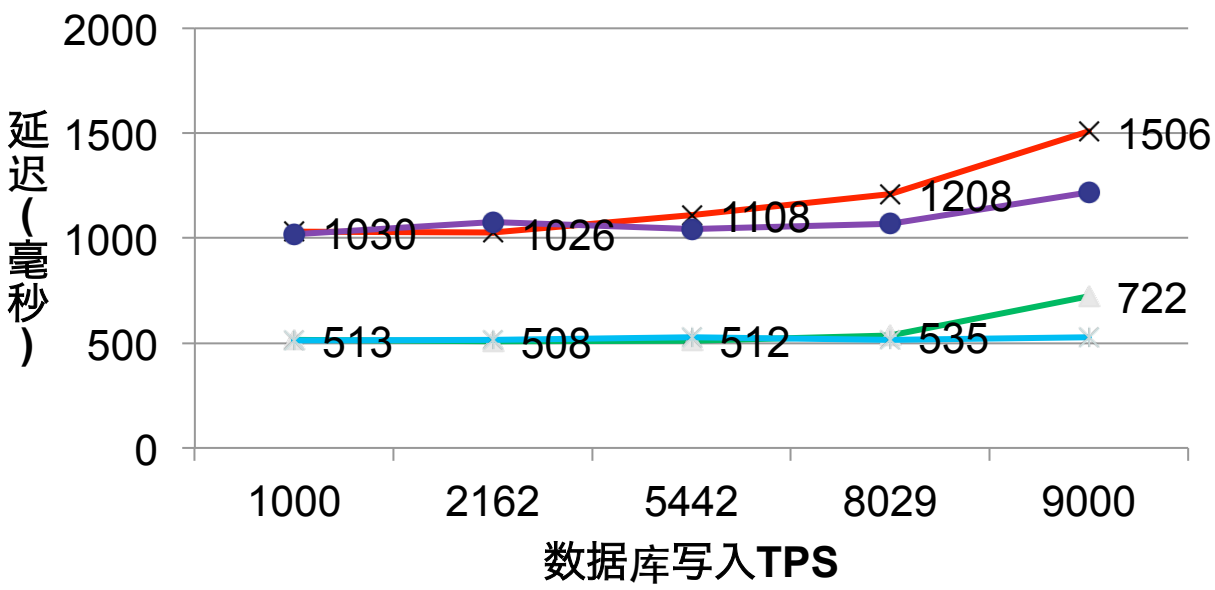


DRC的Performance

- 测试场景



- 性能指标



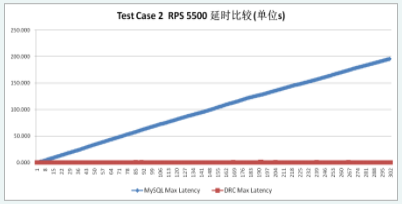
Binlog时间只能精确到秒
包含0.5秒固有统计误差

平均延迟ms/16线程
最大延迟ms/16线程
平均延迟ms/32线程
最大延迟ms/32线程

DRC的Performance

- TPS和延时的压测数据
  - TPS：同步停1小时后启动MySQL Async和Drc，观察TPS和追平时间
  - 延时：主库保持5000TPS写入，观察平均延时和最大延时

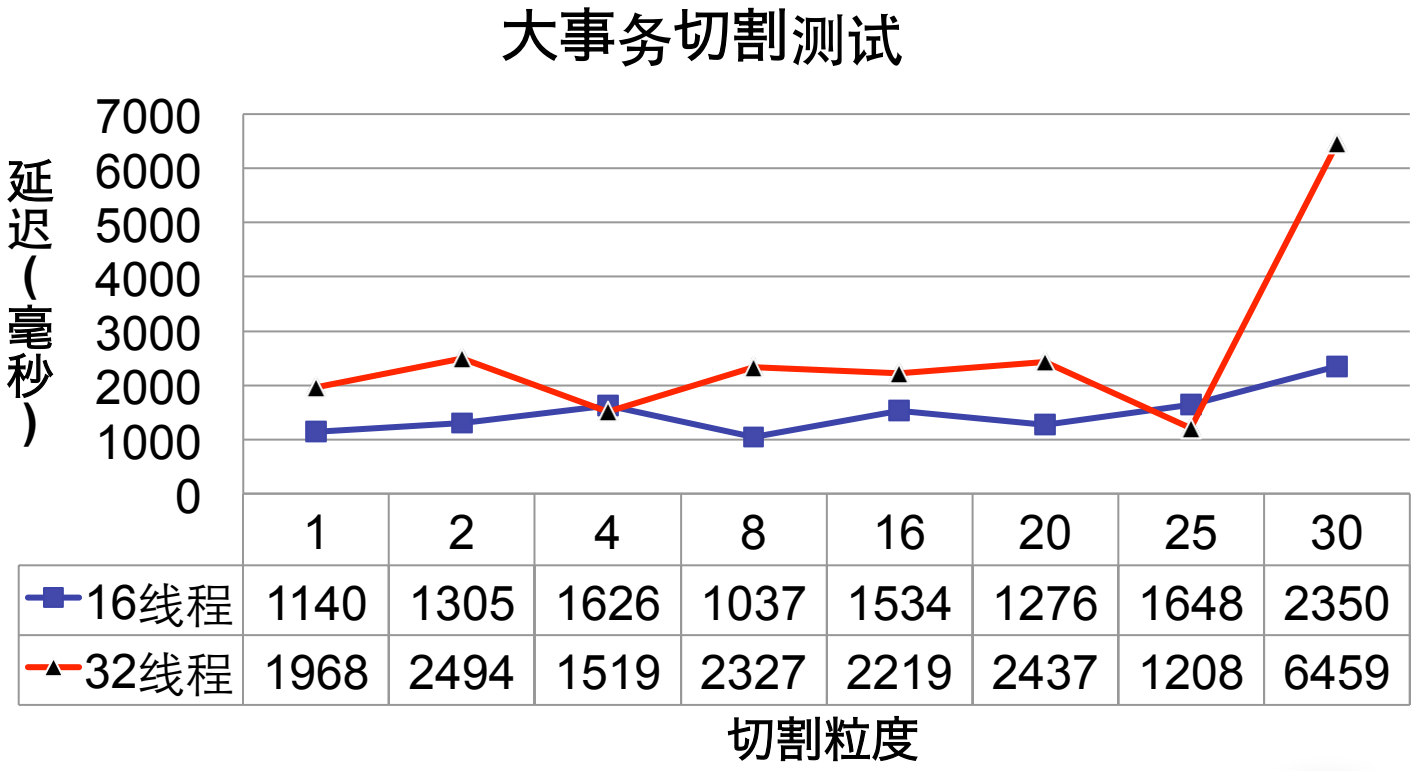
	MySQL 原始复制	DRC	备注
TPS	1236 事务数/秒	6867 事务数/秒	a) 平均单个事务1.33个更新操作 b) 延时趋势如下
追平时间	711秒	128秒	
平均延时	100秒	0.56秒	
最大延时	195秒	1.01秒	



Binlog时间只能精确到秒
包含0.5秒固有统计误差

DRC的Performance

- 大事务
  - 问题：延迟增大
  - 解决：切割大事务

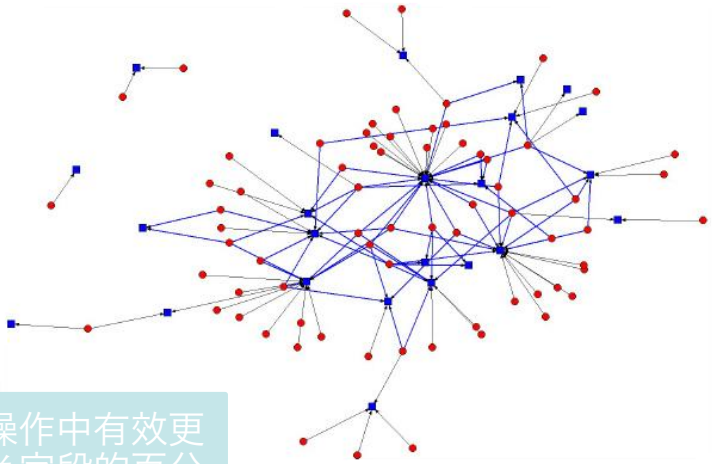


DRC的Highlights

- 数据同步
  - 并行写入
  - DDL支持
  - 双向同步
- 解耦在线存储系统
  - 对内支持不同数据源
  - 对外访问接口一致
- 运维容灾
  - 外部数据源容灾
  - 内部系统容灾
- 安全控制
- 系统管理

DRC设计和实现

- 并发复制
  - 并发策略
  - 适用场景



每秒内更新同一个pk的操作数次数分布:

- pk更新n次 总次数
- 1 13256796(99%)
- 2-5 5590(0%)
- 5-10 662(0%)
- 10-20 211(0%)
- 20-30 33(0%)
- 30-40 4(0%)
- 50-100 17578(0%)

单个tx里更新操作次数分布:

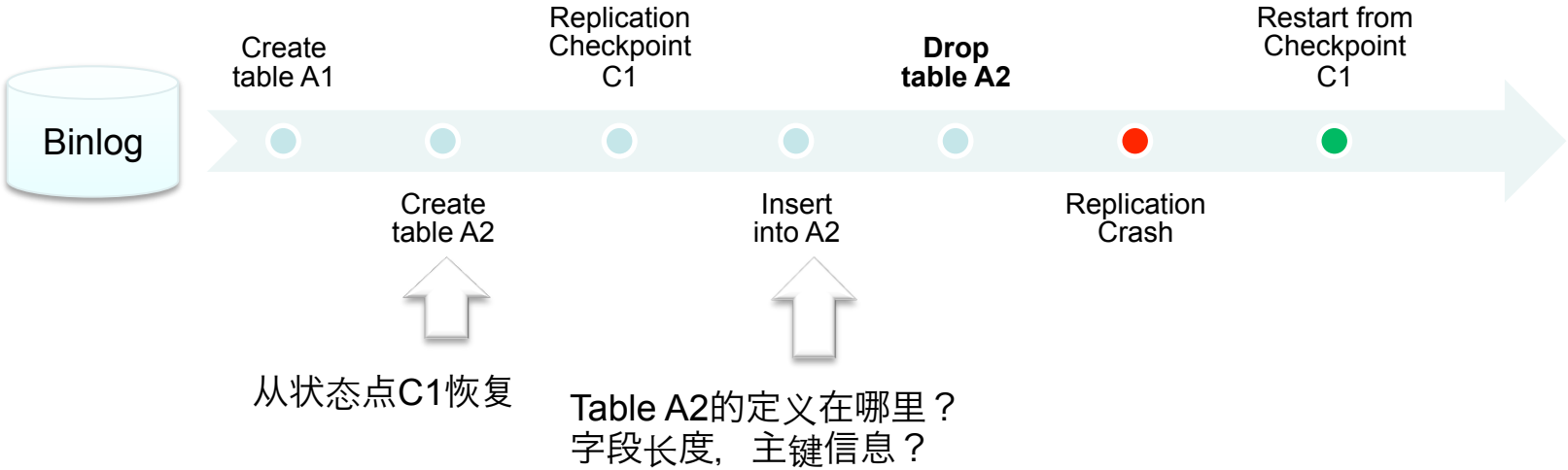
- 更新次数 总次数
- 1 9004792(87%)
- 2-5 1191460(11%)
- 6-10 96898(1%)
- 10- 73536(1%)

单个更新操作中有效更新字段占总字段的百分比分布:

- 有效字段更新百分比 总次数
- 1-10% 7358632(55%)
- 10-20% 3676699(28%)
- 20-30% 73262(1%)
- 30-40% 8092(1%)
- 40-50% 1408991(10%)
- 50-100% 794067(5%)

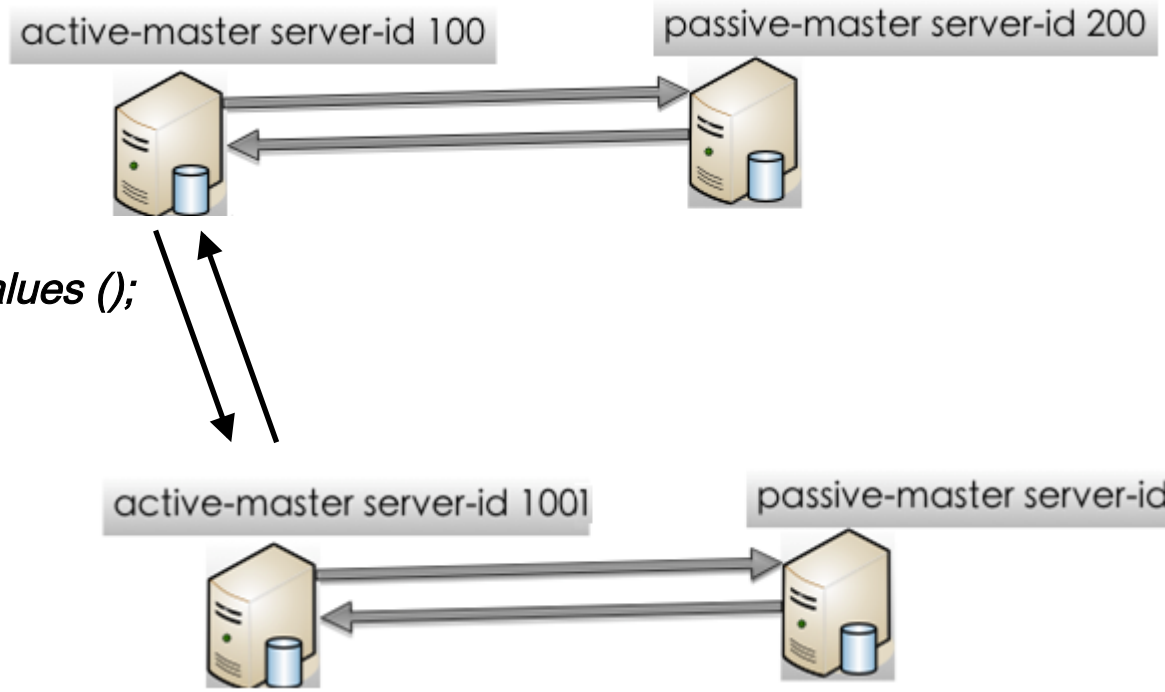
DRC设计和实现

- Meta和DDL
  - Meta的用途
  - 问题：DDL后Meta和Binlog不对应
  - 解决：Meta中心



DRC设计和实现

- 双向同步
  - 循环复制
    - serverid
    - Txn_flag

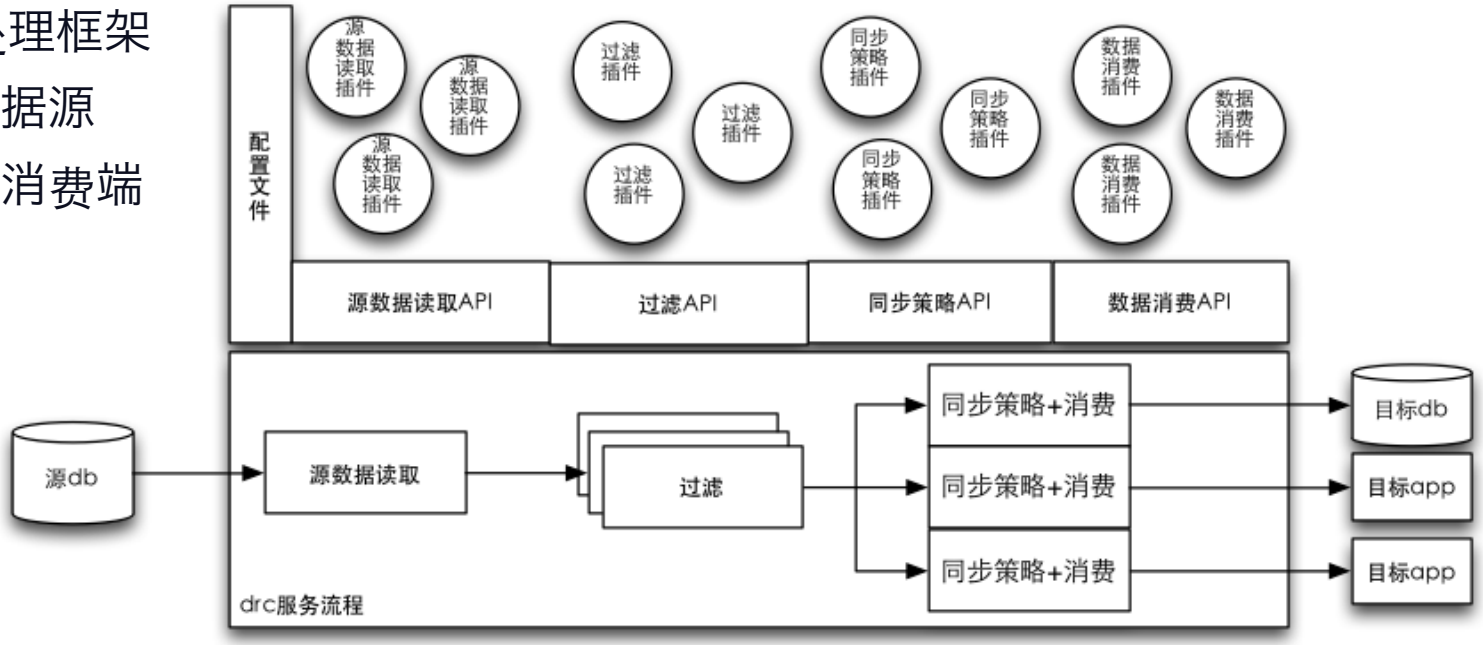


begin
insert into `db.`table values ();
…
end

  - DDL

DRC设计和实现

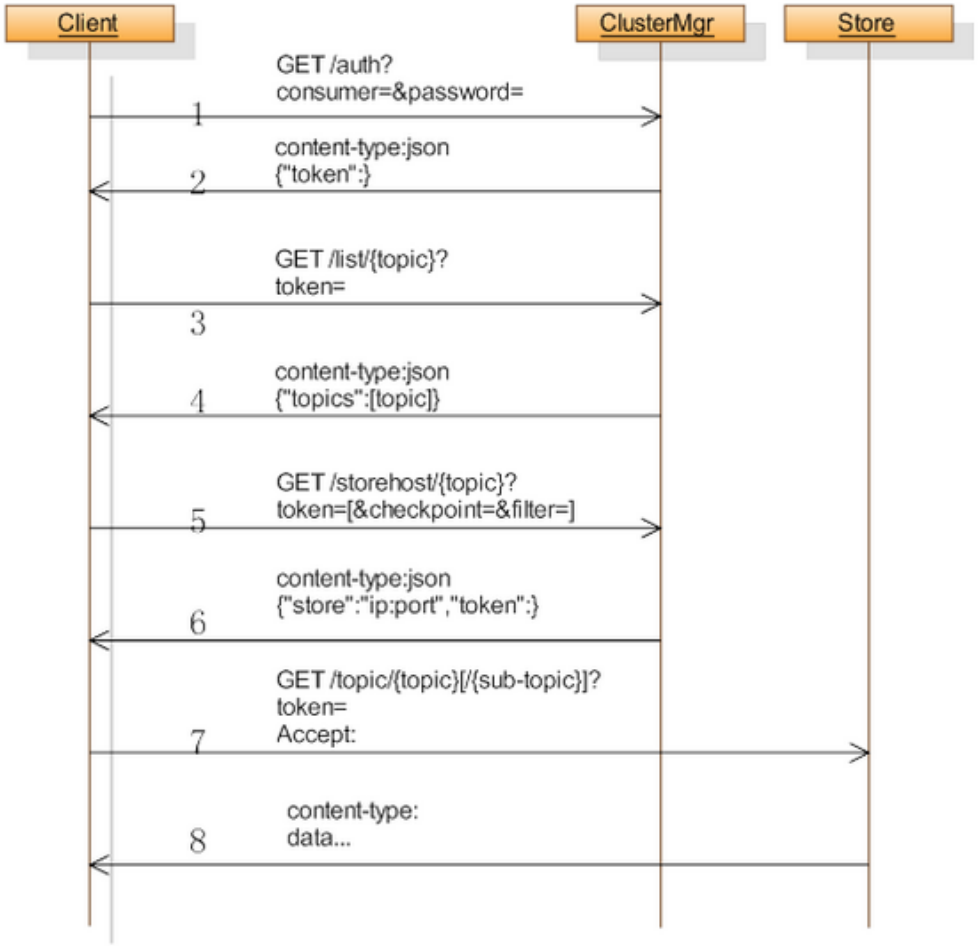
- 统一的数据处理框架
  - 兼容不同数据源
  - 兼容不同的消费端
- 性能优化
  - 多级流水
  - 队列缓存
  - 对象复用



parallelism = 1024
pipeline = reset,read|filterBeforeParse|parse|filter|consume

DRC设计和实现

- 消息订阅
  - REST API
  - 权限管理
  - 指定位点
  - 指定过滤条件
  - 结构化消息格式



DRC设计和实现

- 安全控制
  - 用户权限
- 配置管理
  - 主题管理
  - 机器管理

关键词

- 高性能数据复制技术

DRC的展望

- 打造多地数据中心分布式事务数据库引擎，可自定义数据一致性保护级别
- 解决各种异构数据库的强同步MySQL&Oceanbase&Hbase&Oracle
- 打造高可靠的分布式数据实时同步平台运维体系
- 打造数据库与数据安全体系

QA环节

- 招人么？
  - DRC等待你的加入 @tb杰睿

总结

- DRC——面向数据库复制的高性能服务组件