



2019

05

08-10

北京新云南皇冠假日酒店

数据风云 十年变迁

DTCC

第十届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2019



+

o

o

o

爱奇艺实时数据传输服务 DBIO架构设计与实践

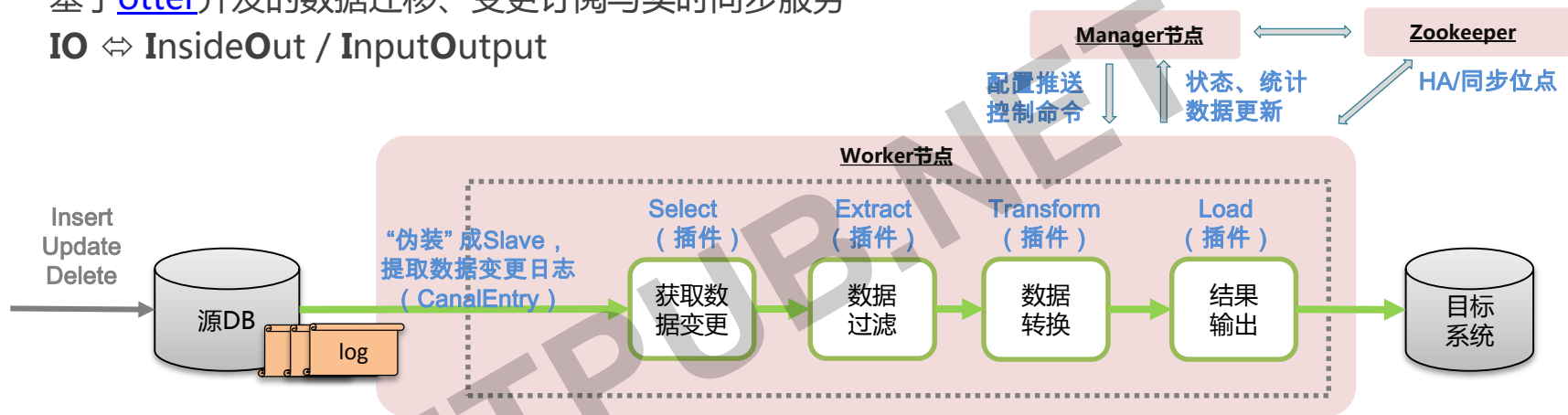
郭磊涛

guoleitao@qiyi.com

DBIO是什么？

基于`otter`开发的数据迁移、变更订阅与实时同步服务

IO ⇔ InsideOut / InputOutput



源端

- 5 种数据库：MySQL、TiDB、MongoDB、Redis、Couchbase
- 全量+增量
- 适配源端Failover

同步服务

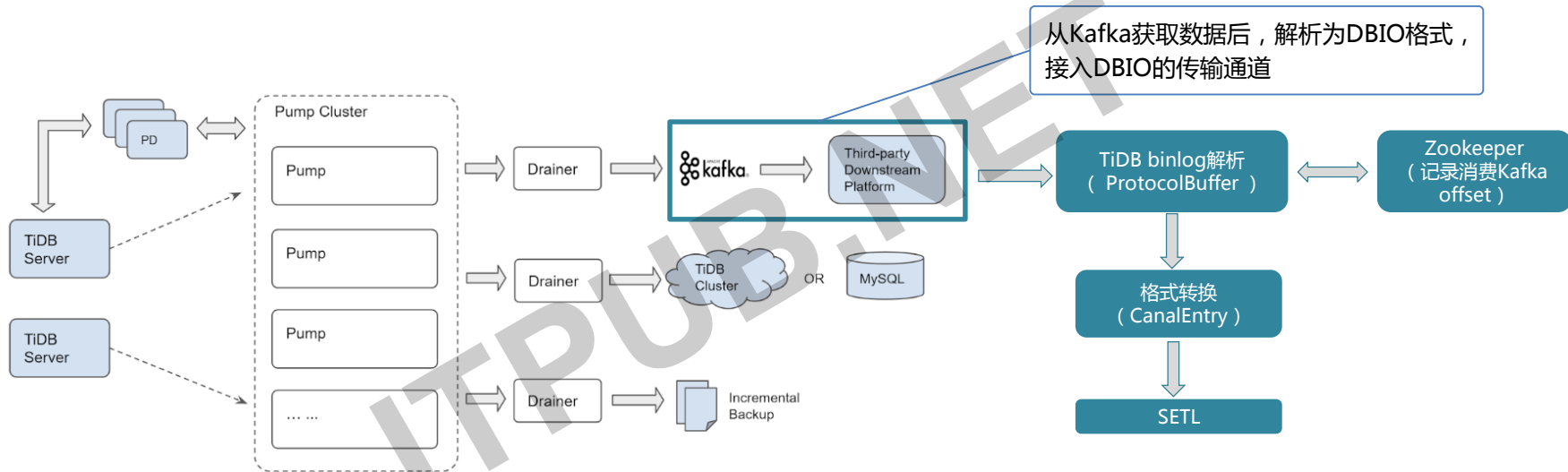
- 服务Failover
- 自动部署
- 自服务：监控、告警、运维自动化

目标端

- 11 种目标系统：ActiveMQ、RocketMQ、Kafka、MySQL、TiDB、MongoDB、Redis、Couchbase、ElasticSearch、Kudu、HBase
- MySQL分库分表
- 单向+双向同步
- 适配目标端Failover

源端支持更多数据库：TiDB

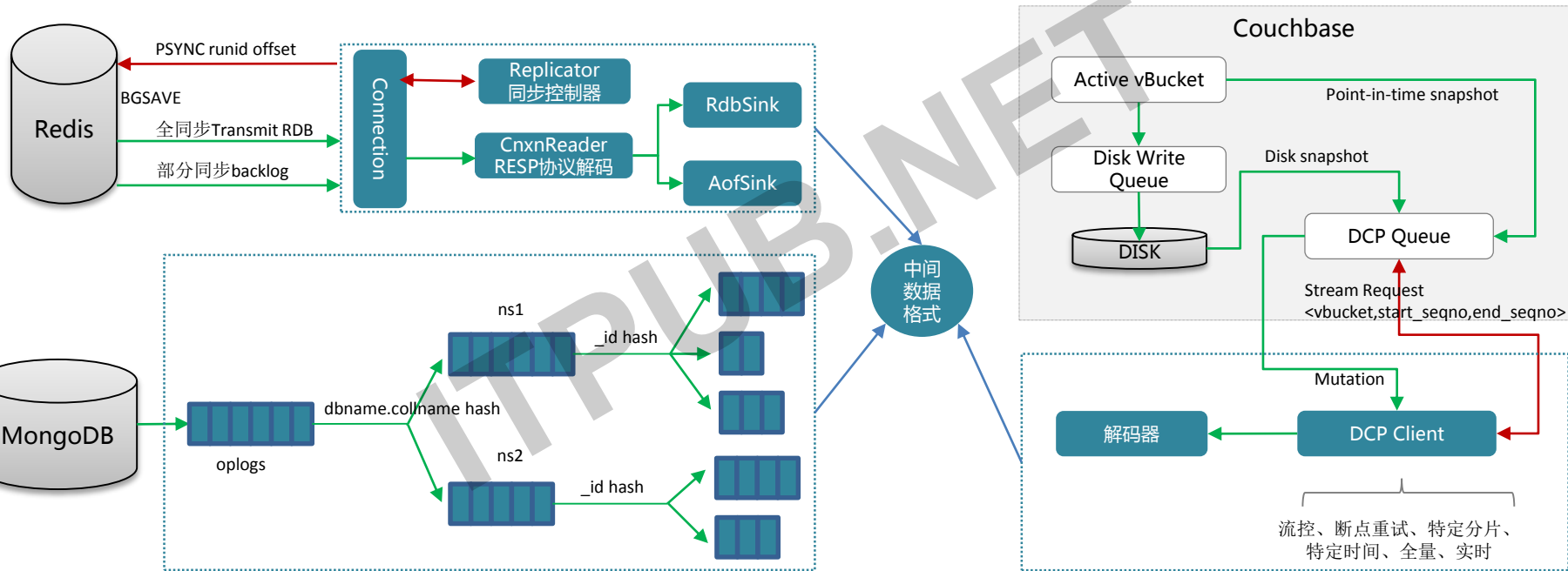
- DBIO实时获取并解析TiDB binlog至中间格式



格式匹配问题：一对多转换；header/key/column/row等的转换；I/U/D的不同处理
字段匹配问题：bit(byte[]转换整形)/整形(根据MySQL字段类型处理溢出等问题)
延迟时间计算：TSO→executetime的转换

源端支持更多数据库：其他

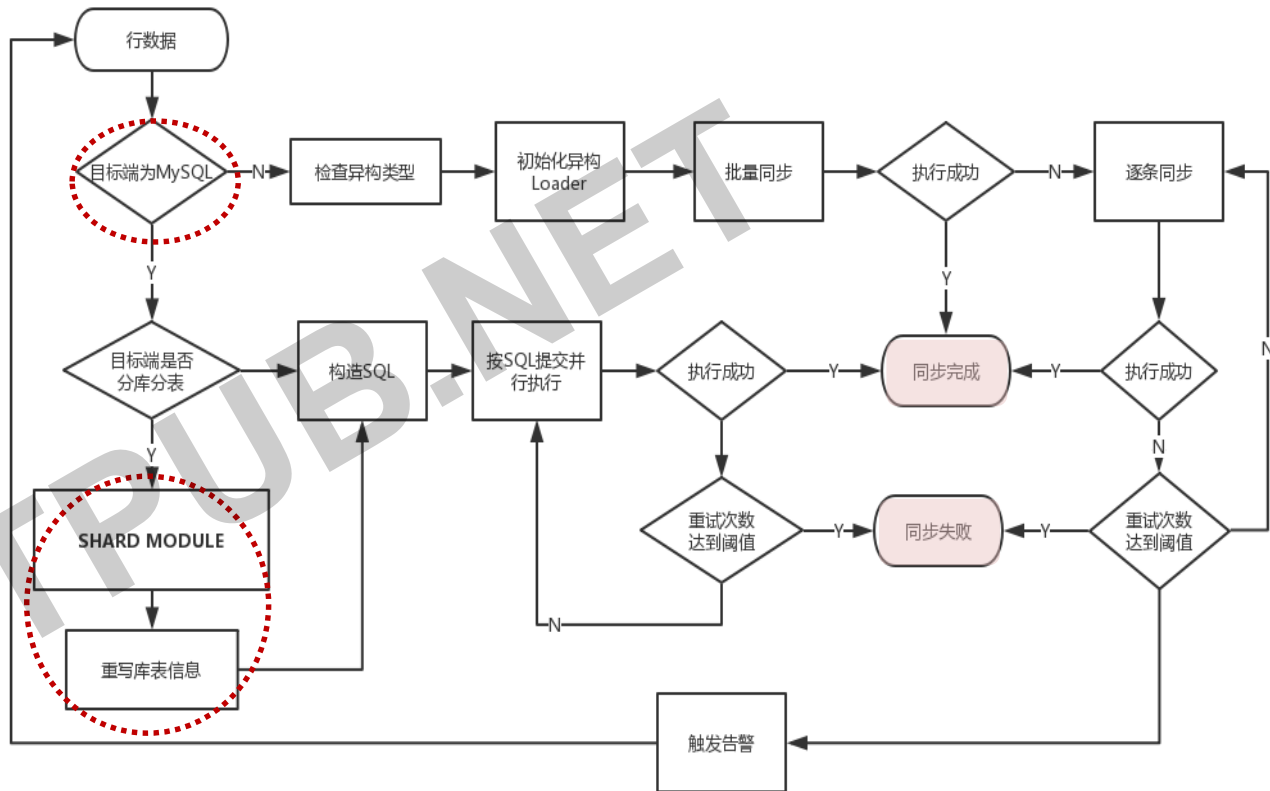
- DBIO伪装为源DB的slave，获取并解析Redis、MongoDB、Couchbase的log至中间格式



支持更多目标端系统

- 修改Load模块，使之支持更多目标端

- 支持的数据格式不统一
- 同步结果的确认方案不统一
- 异常的处理方法不统一
- API使用方法不统一
- 保证重复操作幂等



内置向MySQL分库分表同步组件

外接分库分表
中间件 ❌

向分库分表同步

内置分库分表
模块

优点：代码改造小

缺点：重复SQL解析，同步延迟和吞吐量受影响；更多依赖，带来稳定性问题

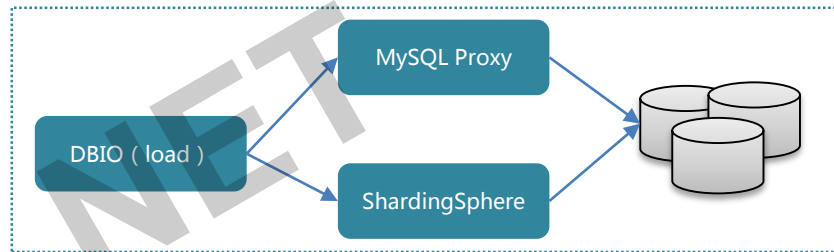
从Binlog解析出来的字段信息
+
插件化的分库分表规则

支持常见的分库分表规则：range/hash/list/crc

文本形式的分库分表规则，代码侵入性低

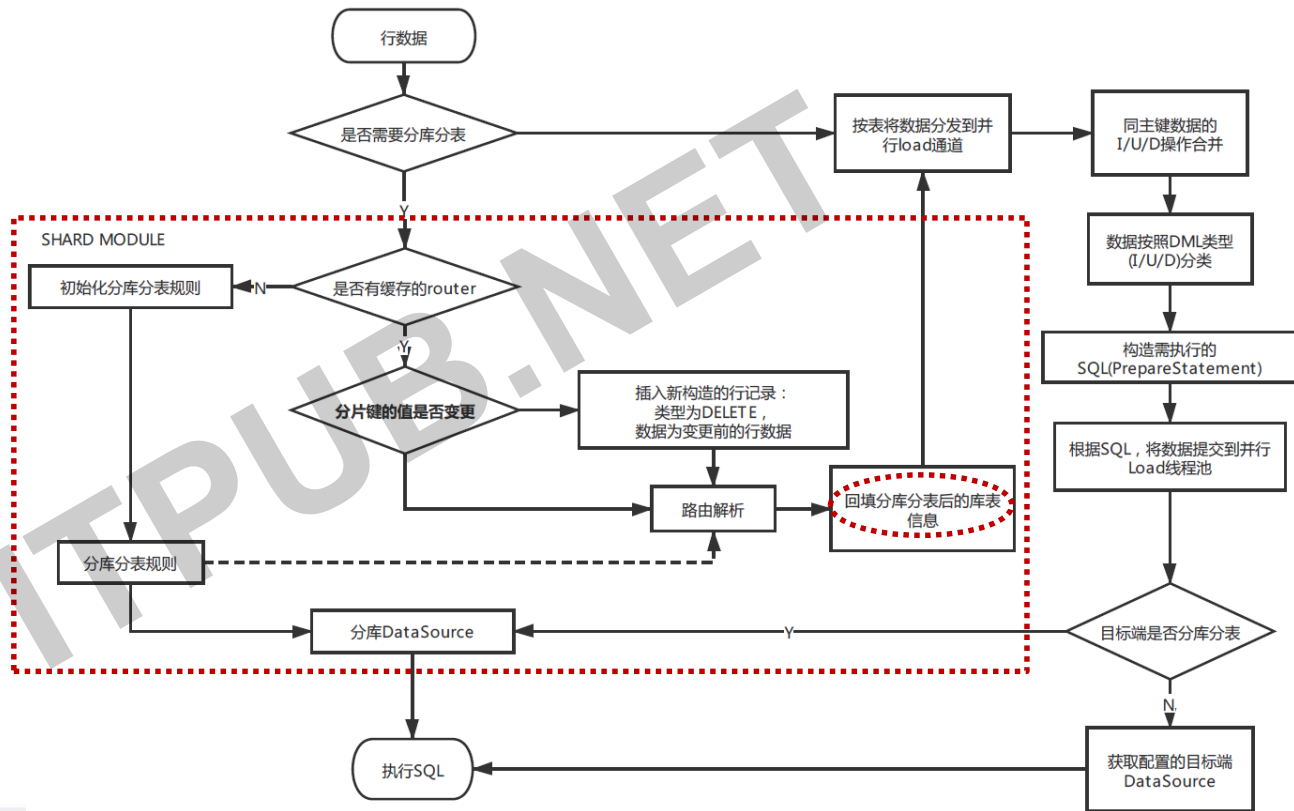
插件化：界面中配置，操作灵活，便于运维

性能影响小

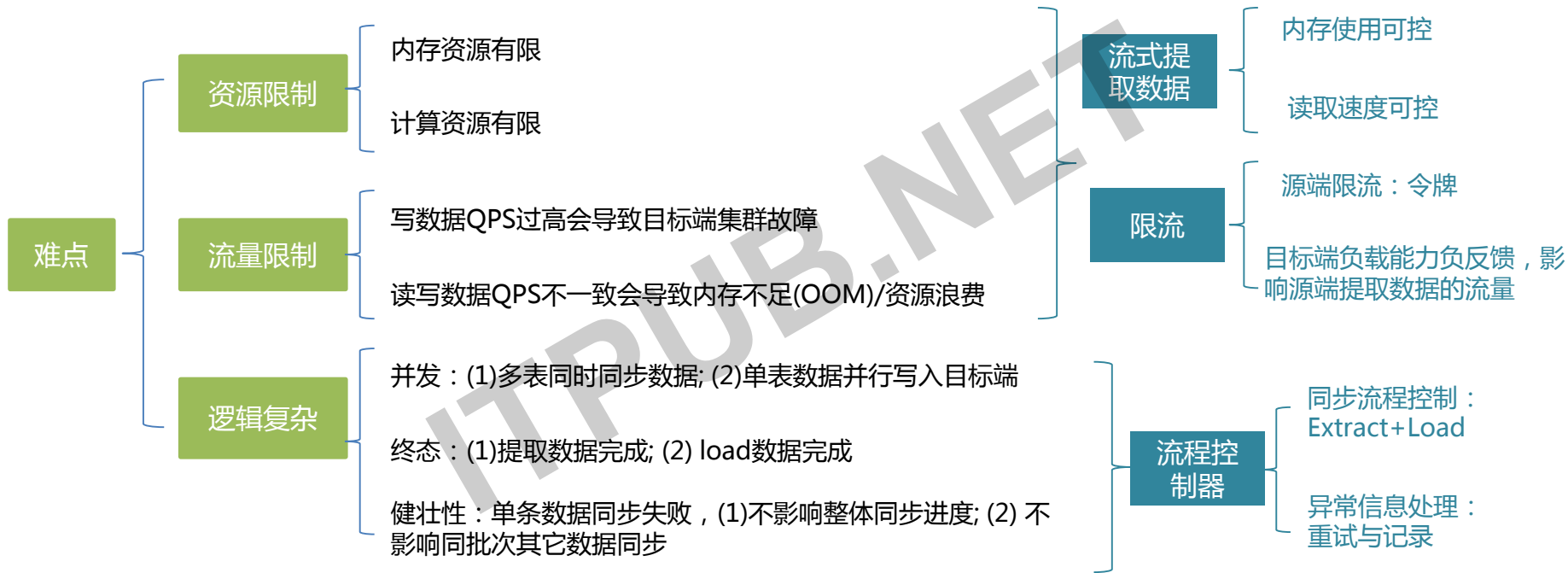


内置向MySQL分库分表同步组件

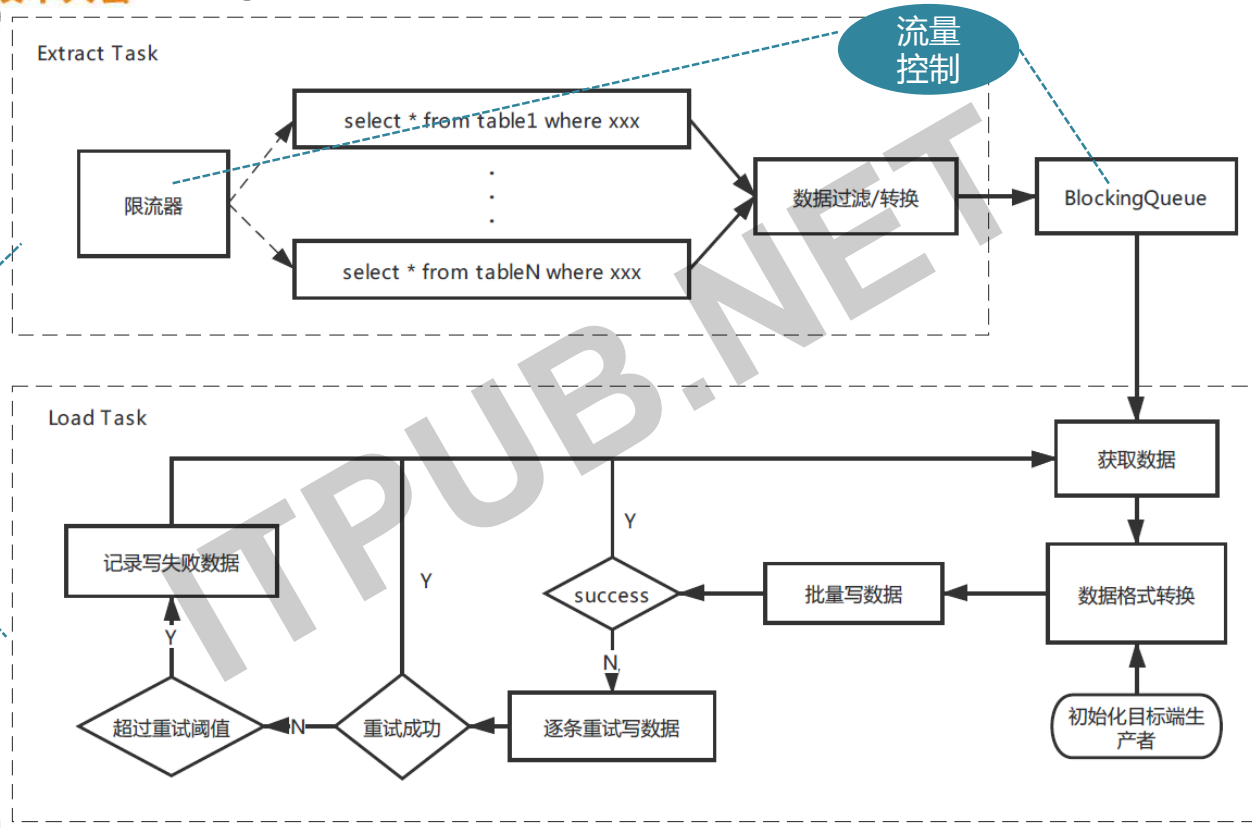
- 连接池换为HikariCP，吞吐量提升15%
- 内置分库分表组件比外接组件吞吐量提升60%
- 分表数越多，性能越低



MySQL全量数据同步



MySQL全量数据同步

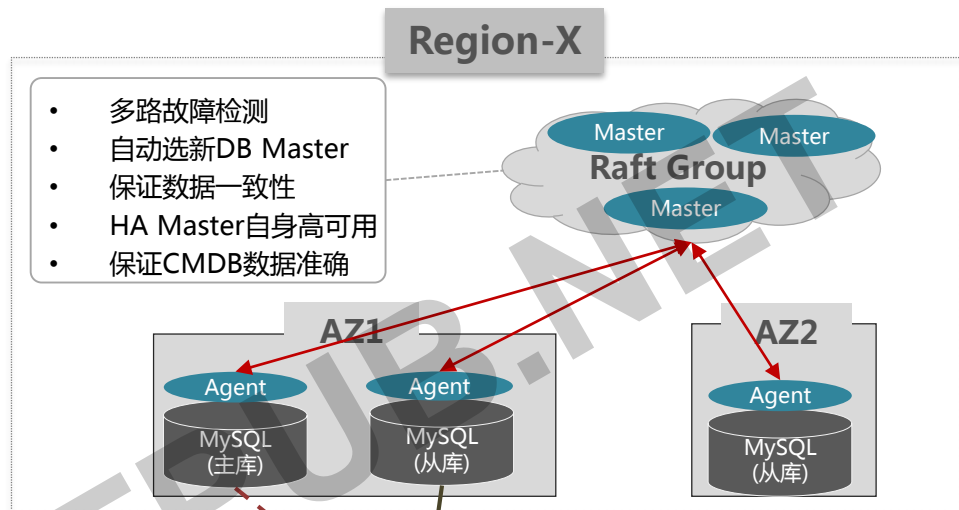


DBIO HA方案

- 涉及到的高可用问题

- 源DB HA
- 目标DB HA
- IO自身的HA

- 多路故障检测
- 自动选新DB Master
- 保证数据一致性
- HA Master自身高可用
- 保证CMDB数据准确



- 捕获DB连接异常
- 查询CMDB获取最新DB Master IP+Port
- 根据timestamp/GTID定位新Master起始位点

- 目标端通过域名访问
- Failover策略：捕获异常+重试（封装在SDK中）

MySQL间双向同步方案

Binlog event增加特定操作

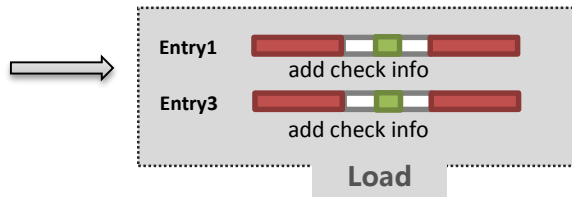
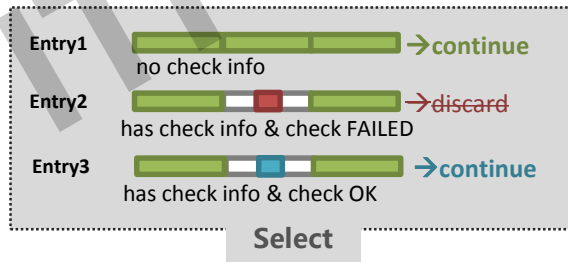
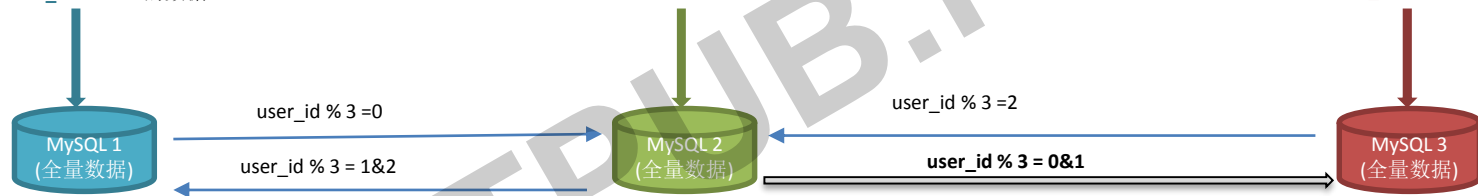


前提条件：业务数据单元化 + 冲突交由业务处理

写：user_id % 3 = 0 的数据

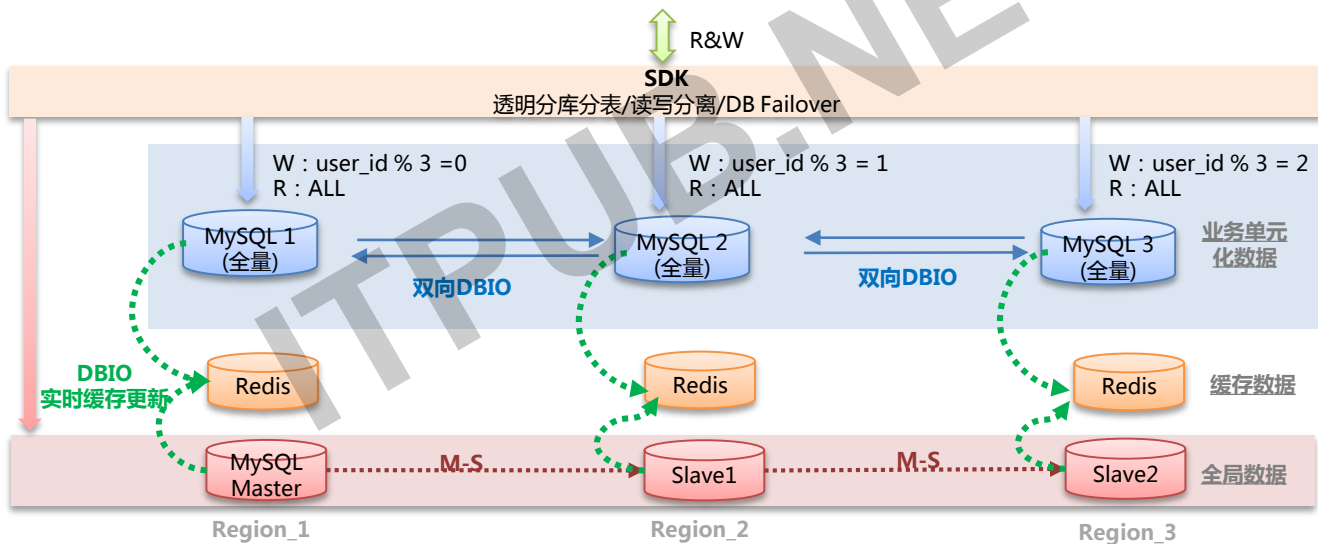
写：user_id % 3 = 1 的数据

写：user_id % 3 = 2 的数据



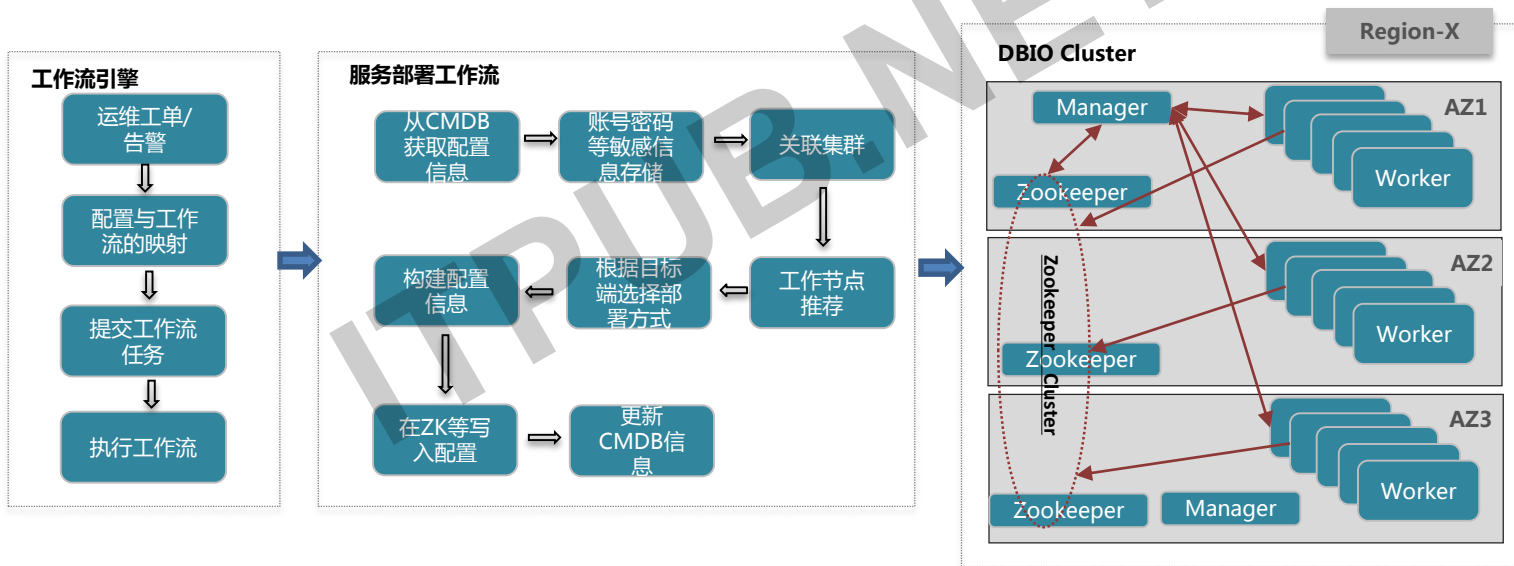
利用双向同步实现MySQL异地多活

- **数据拆分**：单元化数据（异地多活）、全局数据（同城HA）
 - **单元化数据**：每个Region部署一套MySQL集群，Region间通过DBIO双向同步，保证每个Region都可读写
 - **全局数据**：利用MySQL 同城HA保证高可用
- **SDK**：对业务屏蔽后端数据库访问细节，实现透明故障切换/运维



自服务平台

- 工作流引擎+脚本 → 运维自助/告警自愈
- 集群：同地域跨IDC部署，避免单IDC故障
- 调度：根据规则自动关联集群，选择内存/CPU利用率最小且跨IDC的双节点进行部署，节点故障重新调度



基本属性

* 服务类型:

数据同步

数据变更订阅

* 目标系统:

MySQL

MongoDB

ElasticSearch

Kudu

HBase

CouchBase

Redis

* 标题:

* 描述:

* 所属项目:

IDBA

仅列出通过审批的项目; 如未找到项目, 可以 [新建项目](#)

* Email CC:

抄送人的邮箱地址, 多个地址以 “,” 分隔

* 重要级别:

☒ 线上服务

☐ 测试

* 是否敏感:

是

否

敏感资源定义请参考 [WIKI](#)

* 同步方式:

☒ 历史全量数据同步

☒ 增量数据实时同步

其他需求:

channel-~~xxxx-1951-history-kudu~~

返回数据列表

监控信息

历史吞吐量

最近时间

历史吞吐量

2019-04-27 18:43

2019-04-28 18:42

查询

记录(总记录: ~~400000000~~)

鼠标左键可以进行缩放

180,000

80,000

40,000

每秒

同步信息

添加数据库

数据库URL 源：

服务器域名或IP:端口/数据库名

源表名信息：

源字段信息：

源同步字段：

channel-2022-10-11

监控信息

报警项编辑

订阅列表

新增

报警项编辑

监控项目

异常

位点超时(单位)



channel-1

报警信息

报警项编辑


订阅列表

报警项编辑

大小(总大小: 117.25MB)

鼠标点击可以进行缩放

2018-04-27 23:36
数据量: 100KB



异常	ERROR, EXCEPTION	
位点超时(单位: ms)	600	ON
延迟(单位: ms)	600	OFF
处理超时(单位: ms)	60	OFF

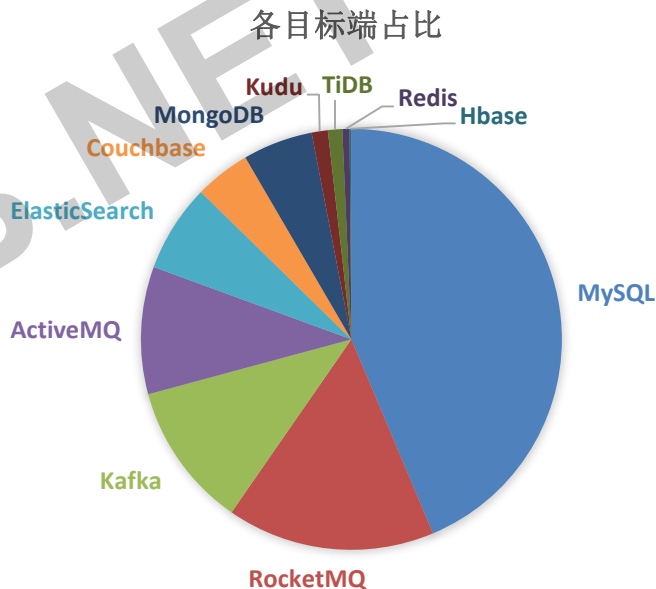
全部开启

全部关闭

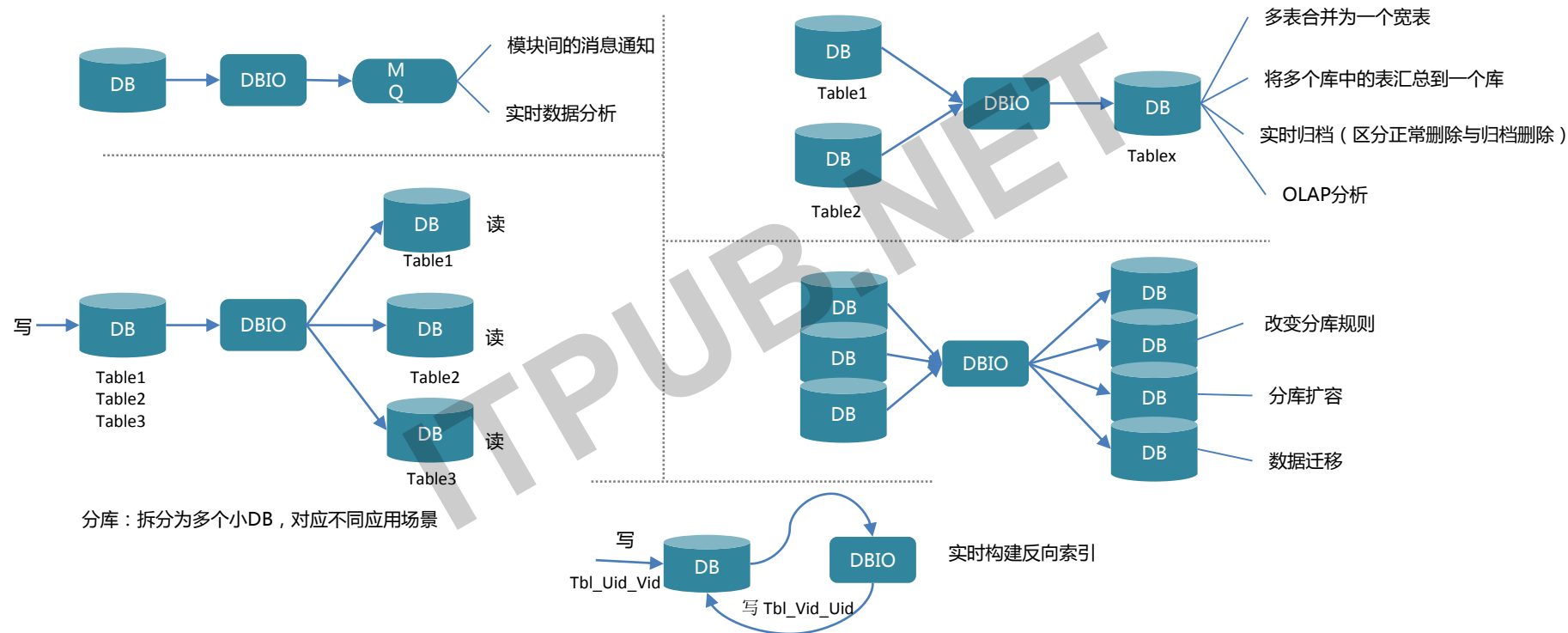
提交

DBIO在爱奇艺的应用

- DBIO已在爱奇艺稳定运行2年多，有千条实时同步流，吞吐量百万行/秒
 - MySQL间同步需求最强烈
 - 主要应用场景为：DB功能拆分
 - RMQ逐步取代ActiveMQ



DBIO常见应用场景



分库：拆分为多个小DB，对应不同应用场景

正在进行的工作

- 应对突发流量，优化性能
 - 源端太快
 - 目标端太慢（消息积压等）
- 更好的应对DDL
 - DDL自动同步到目标端
- 更多应用方案
 - 数据迁移+SDK实现透明冷热数据分级存储



加入爱奇艺！
请关注爱奇艺伯乐



THANKS

ITPUB.NET