



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



+

○

○

○

中国银联跨中心&异构数据同步 技术与实践

银联科技事业部·操作系统与数据库团队
数据库研发工程师 翁海星

目 录

- 需求背景
- 架构设计
- 案例实践
- 未来规划

需求背景

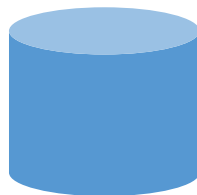
□ 简单的数据同步？

- MySQL
- 主从复制
- 主主复制
- 异步复制
- 半同步复制
-



UPSQL

A机房
/A中心



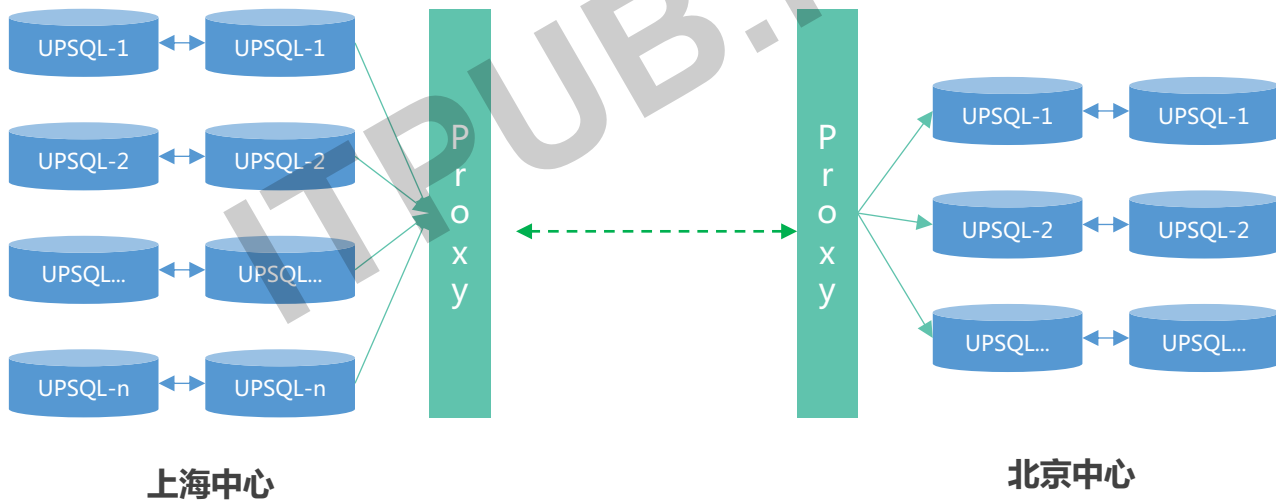
UPSQL

B机房
/B中心

需求背景

□ 分库之后.....

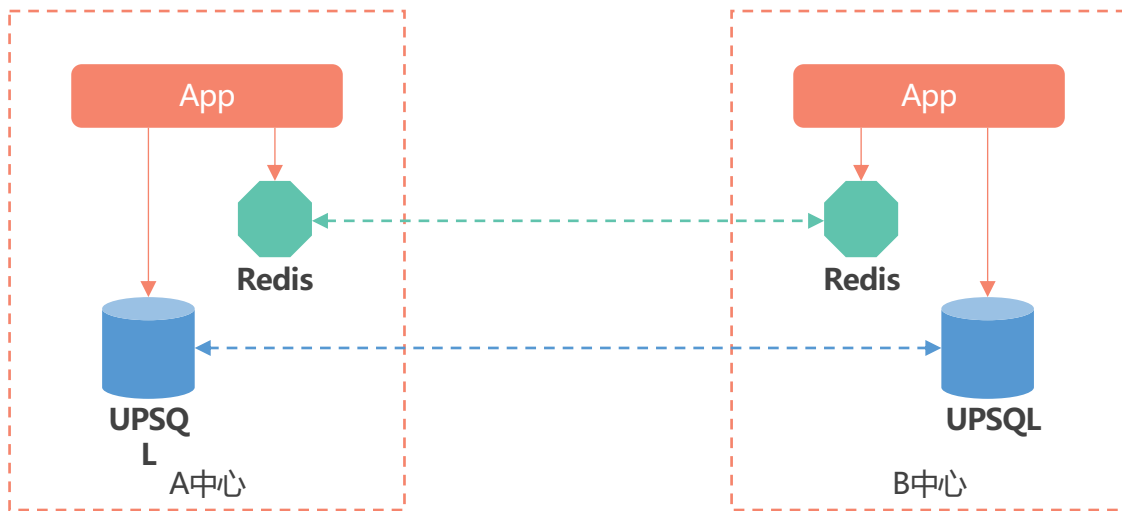
- 双中心需要转所有的表？
- 双中心的部署资源不匹配？
- 中心内主从切换？



需求背景

□ 还有.....

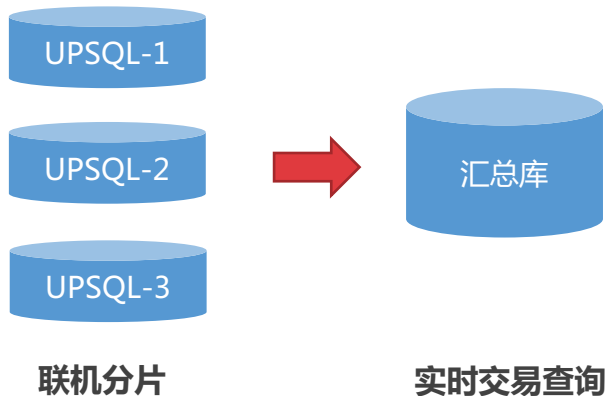
- 双中心的缓存同步
- 数据库与缓存之间同步



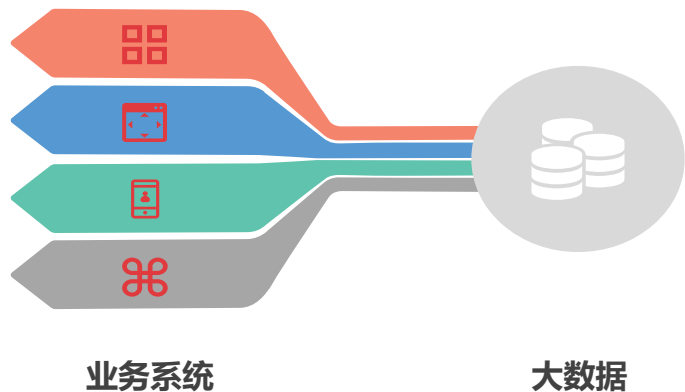
需求背景

□ 更复杂的情况？

- 联机 => 汇总库
- 形变
- 异构数据库(DB2, ElasticSearch)

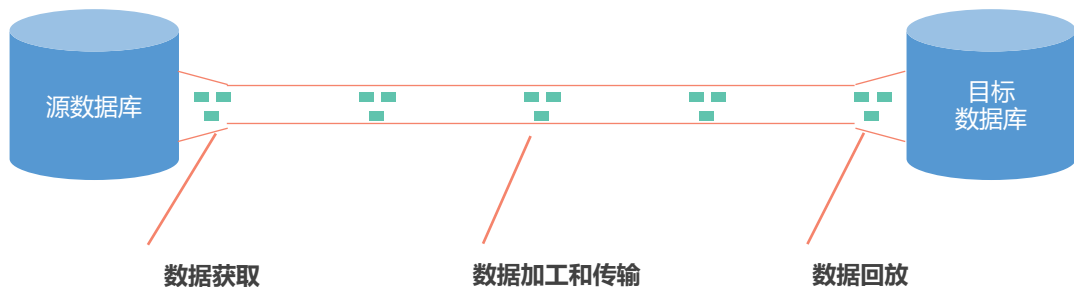


- ETL工具
- T+1 ❌



需求背景

□ 数据同步组件-Moray



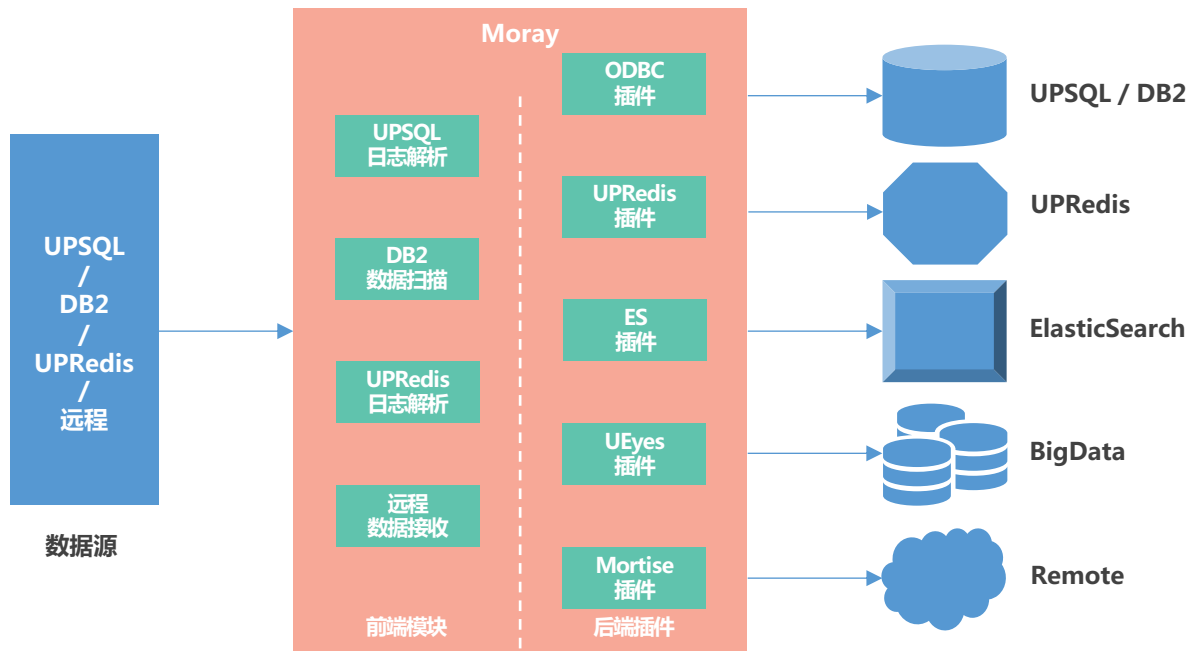
Moray是一个数据同步组件，为数据库之间提供点对点、准实时数据同步服务。

Moray 架设于源数据库与目标数据库之间，通过解析源库日志或扫表获取数据，经过内部加工，然后传输至目标库进行回放，从而实现数据同步。

- 对应用透明
- 应用无需额外开发
- 尽可能降低对主库性能的影响

架构设计

□ 前后端分离



前后端模块插件化设计，设计内部通用协议，以支持不同数据源和目标数据库的组合。

架构设计

□ 设计要点

- 尽可能不拆分事务，保证事务原子性和数据一致性。
- 尽可能不直接读源表，不与应用竞争数据库资源；数据压缩传输，节约带宽；多线程并行回放，并优化并行算法。



- 断点续传保证数据零丢失；报文校验防数据篡改。
- 采用端到端同步，去中心化，各业务之间互不影响；独立运行，降低与源库的耦合度；可根据数据库架构灵活部署。

架构设计

□ 设计要点

- 管理监控平台实现节点管理和运行状态监控。如：同步进度、时延。
- 总体分为前端模块和后端插件，模块和插件均易于扩展，未来可支持更多数据源和目标库。



- 针对多种异常场景设计了容错策略，如：自动过滤重复数据；数据异常自动转换补偿等。
- 组件之间接口异步化，无阻塞，后一组件通过异步ACK给前一组件应答，提高整体性能。

架构设计

□ 同构或异构数据库间数据同步

目标库 数据源	UPS QL	DB2	UPRedis	ElasticSearch	Ueyes (Impala)	Kafka
UPS QL	√	√		√	√	
DB2	√	√		√	√	
UPRedis	√		√			

- **UPS QL**：解析源库binlog日志获取数据，通过ODBC插件写入UPS QL或DB2数据库，或通过ES插件写入ElasticSearch等；推荐用于UPS QL实时数据同步。
- **DB2**：扫描源库表格获取数据，通过ODBC插件写入UPS QL或DB2；推荐用于一次性数据迁移。
- **UPRedis**：解析源库AOF日志，通过redis插件写入redis，或通过ODBC插件写入UPS QL；推荐用于UPRedis缓存数据实时同步，或持久化至数据库。

架构设计 - UPSQL实时同步

- Binlog dump协议，获取/解析增量日志
- Moray内部协议格式保持数据库的事务特性
- 参考MySQL组提交，将可以并发的事务归入一组
- 同时支持MySQL XA协议（分布式事务）

<GTID_EVENT> = 19fc90f5-b17a-11e8-a842-9cb6548afe90:1

<QUERY_EVENT> = XA START X'78612d31','X'',1

<TABLE_MAP_EVENT> = test.t

<WRITE_ROW_EVENT> = insert into

<QUERY_EVENT> = XA END X'78612d31','X'',1

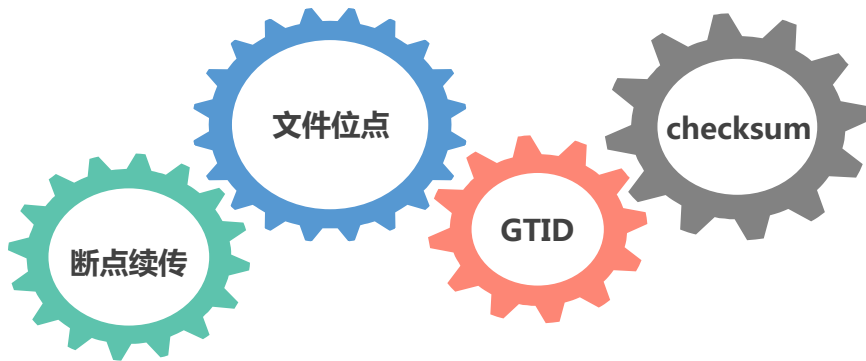
<XA_PREPARE_EVENT> = XA PREPARE X'78612d31','X'',1

<GTID_EVENT> = 19fc90f5-b17a-11e8-a842-9cb6548afe90:2

<QUERY_EVENT> = XA COMMIT X'78612d31','X'',1;

架构设计 - UPSQL实时同步

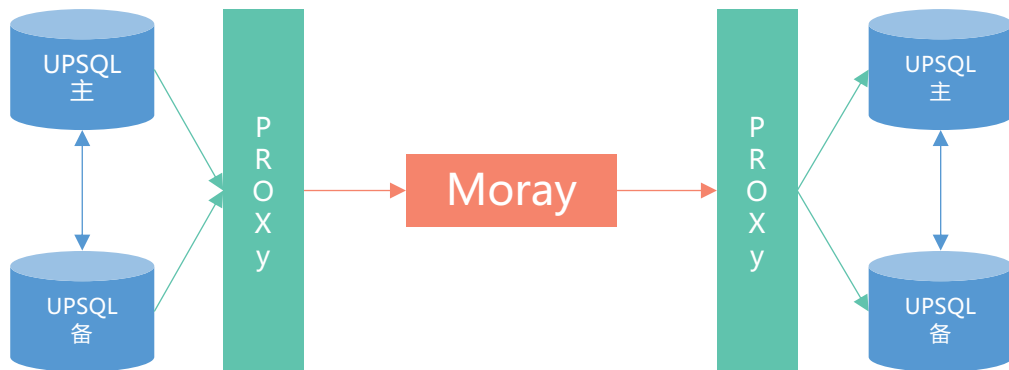
□ 数据安全



- 断点续传严格保证数据零丢失
- 文件位点或GTID策略保证数据起点精准
- 报文校验和保证数据包传输完整
- 内部协议设计上保证数据回放时的ACID特性

? ? ? ? - UPSQL ? ? ? ?

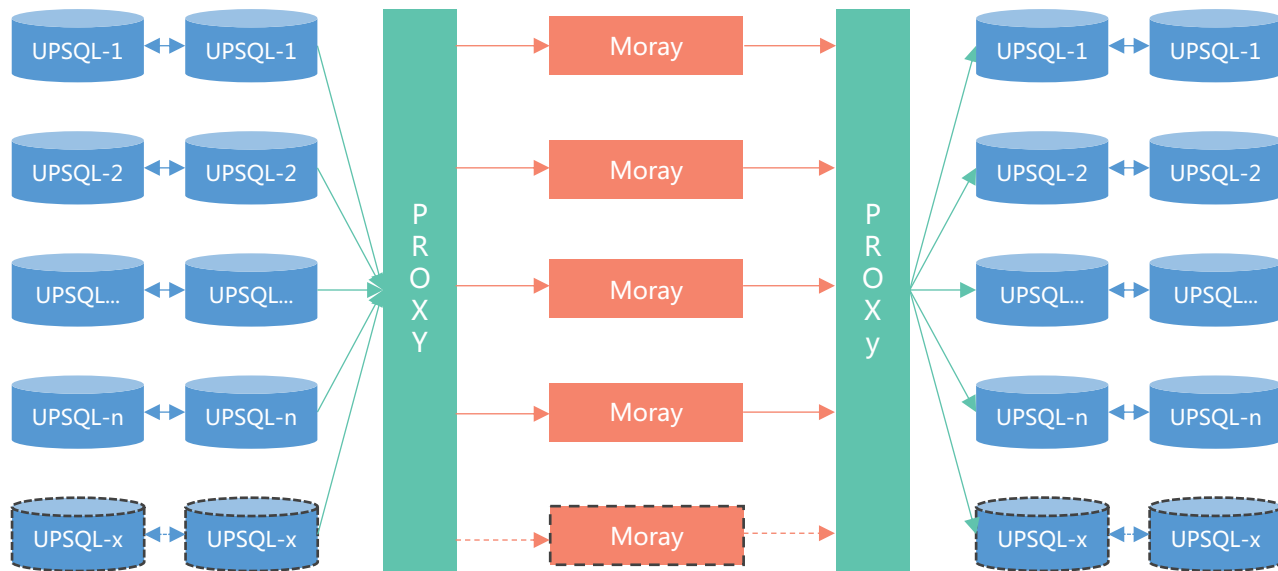
□ ? ? ? ? ? ? ? ? + UPSQL-Proxy ?



- 一组主备库共享一条同步链路
- Moray与两中心的Proxy相连，由Proxy负责数据库的高可用切换
- Moray自身为冷备，需手动切换；银联使用DBaaS平台集成，由DBaaS负责主备切换

? ? ? ? - UPSQL ? ? ? ?

□ ? ? ? ? UPSQL-Proxy ?



新增数据库分片时需新增Moray同步线路。

架构设计 - UPSQL实时同步

□ 容错机制

容错场景		容错策略	备注
重复执行	同一段日志多次回放	UPSQL根据GTID自动过滤	
数据异常	Insert记录已存在	Moray自动转换为update	
	Update记录不存在	Moray自动转换为insert	
	Delete记录不存在	Moray自动忽略	
数据冲突	切库或回切，数据延迟导致数据被双写	Moray根据时间戳或版本号保留最新的记录	1

1. 要求表上有时间戳字段(精度越高越好)或版本号字段

架构设计- UPSQL实时同步

- 数据库变更
- Moray目前不支持DDL语句
- 变更需根据如下策略

模式	灾备	双活
同步线路	单向	双向
DDL类型	新增表 新增字段 增加字段长度 ...	新增表 新增字段 增加字段长度 ...
是否停数据库	否	否
是否停Moray	否	是
操作步骤	1. 备中心变更 2. 主中心变更	1. 停止Moray同步 2. 双中心同时变更 3. 启动Moray同步

架构设计 – 数据形变

□ 使用lua脚本配置数据形变功能

```
16 function moray_df()  
17     res = {} -- 存储行记录内容  
18     origin_table_name = odbc.table_name -- 源库的表名, odbc.table_name值为ORIGIN_TABLE  
19     col_value = odbc.ORIGIN_COL -- 源库的ORIGIN_COL字段值  
20     if (string.sub(col_value,-1) >="0" and string.sub(col_value,-1) <="4")  
21     then  
22         res.table_name = origin_table_name.."01" -- 表名变换, 替换为"schema.table"格式  
23     else  
24         res.table_name = origin_table_name.."02" -- 表名变换, 替换为"schema.table"格式  
25     end  
26     return res -- 返回行记录  
27 end
```

架构设计 – 数据形变

□ 更复杂的一个例子

```
1 function moray_df()
2   res = {} -- 一行转多表: 第一张表
3   res.table_name = "sbtest.test" -- 表名变换
4   local column_value = to number(odbc.settle_dt)
5   local skipvalue = column_value % 3
6   if (skipvalue == 2) then
7     res.skip = true -- 是否跳过该条记录
8   end
9   res.change_name = {}
10  res.change_name.expire_dt = "expire_datetime" -- 列名修改
11  res.col = {}
12  res.col.expire_datetime = "1234" -- 列值配置值, 定义row后才生效, 否则不会修改列值
13  res.row = {}
14  res.row[1] = {} -- 一行转多行: 第一行
15  res.row[1].transmsn_dt_tm = "100" -- 主键修改为100
16  res.row[1].expire_datetime = "5678" -- 必须定义row才可以实现列值修改, 第一行的列值修改, 覆盖上述的通用修改值"1234"
17  res.row[2] = {} -- 一行转多行: 第二行
18  res.row[2].transmsn_dt_tm = "101" -- 主键修改为101
19
20  res2 = {} -- 一行转多表: 第二张表
21  res2.table_name = "sbtest.test2" -- 表名变换
22  res2.change_name.expire_dt = "expire_datetime222" -- 列名修改
23
24  res3 = {} -- 一行转多表: 第三张表 = 原表
25
26  return res, res2, res3 -- 一行转多表, 返回多个res
27 end
```

TPS - UPSQL

测试环境：

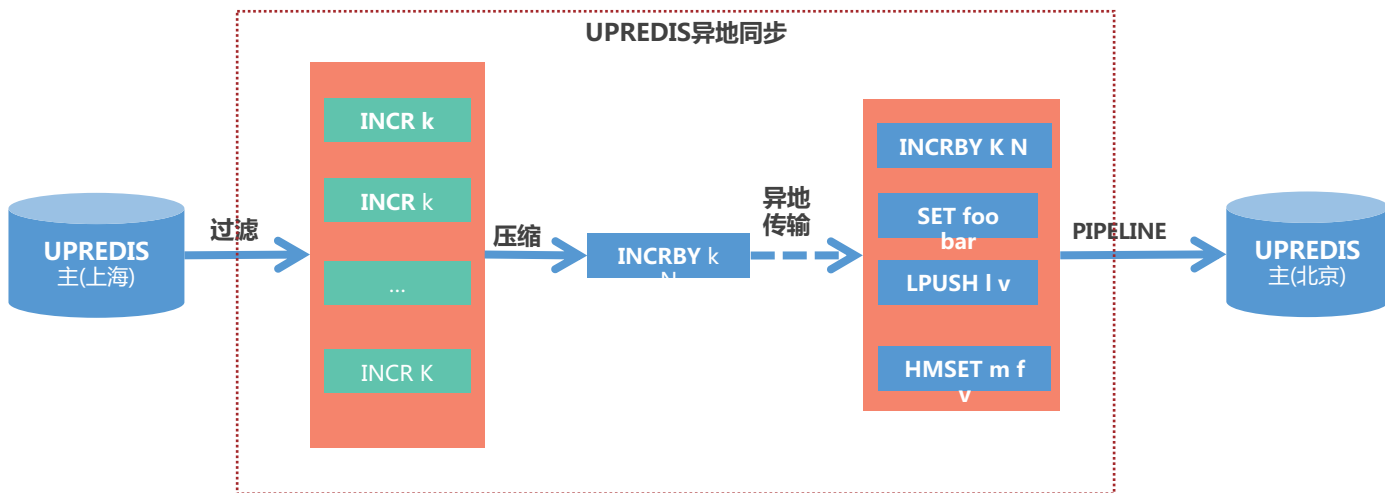
- 源端：虚机一台4C4G
- 目的端：虚机一台8C8G
- 回放线程数：64
- 银联内部通用测试场景，一条记录2K，一个事务包含一个Insert语句以及一个Update语句

压缩和校验	TPS	CPU使用量		网络传输
		源端	目的端	
无压缩，无校验	10000	2.1C	6C	195Mbps
无压缩，有校验	9800	2.2C	6.2C	191Mbps
有压缩，无校验	9500	2.4C	6.5C	18.5Mbps
有压缩，有校验	9400	2.5C	6.6C	18.3Mbps

架构设计-UPRedis实时同步

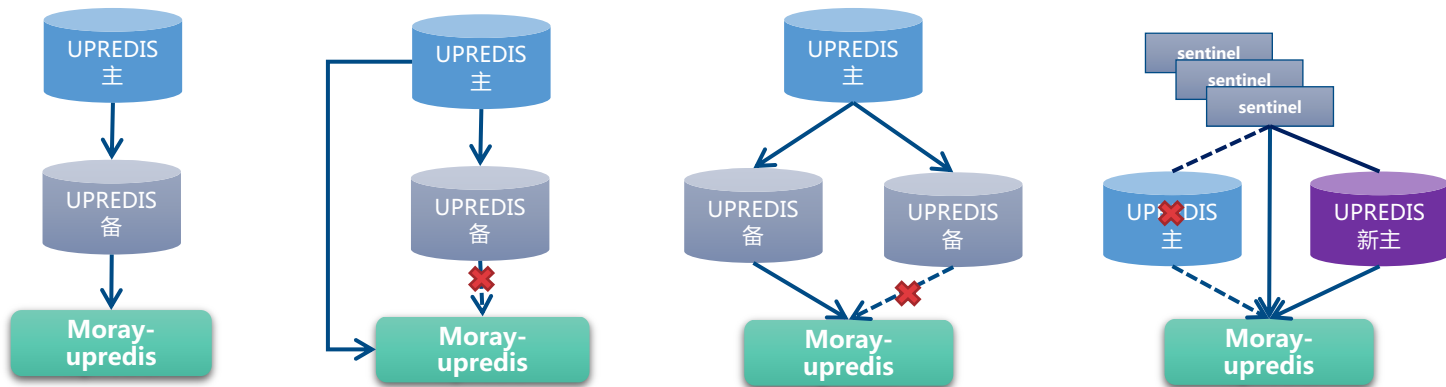
□ 高性能

- 报文压缩，减少网络带宽
- PIPELINE发送，提高收发效率
- 设计性能10万QPS，流量400Mbps，压缩后<40Mbps



架构设计-UPRedis实时同步

□ 高可用



- 正常情况下，UPREDIS从备库同步数据
- 当备库不可用时：
 - ◆ 如果1主1从，UPREDIS从主库同步数据
 - ◆ 如果1主多从，UPREDIS从其它从库同步数据
- 主备切换时，异地同步组件订阅sentinel信息，切换到新主

案例实践

- 2018.2月 投入研发
- 2018.5月 第一个版本release
- 2018.9月 第一个业务上线



未来规划

□ 更多的模块与插件

- Oracle
- Kafka

□ Moray集群化

- 简化部署配置与集中管理
- 进一步提升高可用

Contact us

<https://open.unionpay.com/tjweb/product/detail?prold=45>

wenghaixing@unionpay.com



The screenshot shows the top navigation bar of the UnionPay Open Platform with links: 开放平台, 产品API, 解决方案, 科技产品, 支持中心, and 特色专题. The main header features the Moray logo and name. Below it, a paragraph describes Moray as a real-time data synchronization component developed by UnionPay, supporting various databases like UPSQL, MySQL, DB2, Elasticsearch, HDFS, and UPREDIS. A row of three buttons is present: 业务申请, 技术支持, and 介质下载. At the bottom of the header section, there are four links: 产品特点, 应用场景, 介质下载, and 联系我们.

UnionPay 开放平台 产品API 解决方案 科技产品 支持中心 特色专题

Moray

Moray是中国银联研发的，是源数据库与目标数据库之间的实时数据同步组件，对应用透明，应用无需额外开发，且对数据库性能影响小。目前已支持UPSQL/MySQL同步到UPSQL/MySQL、DB2、elasticsearch和HDFS，支持UPREDIS同步到UPSQL/MySQL和UPREDIS，支持DB2同步到UPSQL/MySQL和DB2，更多的异构数据库数据转移正在开发中。

业务申请 技术支持 介质下载

产品特点 应用场景 介质下载 联系我们

产品特点

- 高可用

高



THANKS