

# 读写分离，主从延迟

创建：王丽月，最新修改于：四月 26, 2017

## 问题描述

同一请求，写入后，立即查询DB。DB为一主多备：Master、Slave1、Slave2，由于主从延迟，导致读入数据滞后，场景如下：

```
com.sankuai.ia.order.trade.handler.AptHandler#process 下单：已创建订单，创建预约单时，查询订单基  
com.sankuai.ia.order.trade.handler.CreateOrderAllInfoHandler#process 下单：创建订单完整信息，  
com.sankuai.ia.order.trade.handler.CreateOrderHandler#process 下单：创建订单，查询订单基础信息  
com.sankuai.ia.order.trade.handler.PrepareHandler#process （不需要实时）  
com.sankuai.ia.order.trade.handler.PaymentHandler#rollback checkOrderStatus 以防3020延迟，导  
com.sankuai.ia.order.facade.PaymentNotifyFacade#afterPaySuc 支付通知，成功后处理，以防3050延迟
```

此外，DataBus监控的是Slave1，当查询打到Slave2，也存在主从不同步问题。

## 方案调研


1. 写入后DB后，Thread.sleep，等待主从同步。→ low
2. 使用Proxy，监控Client的查询和更新。主从表都增加一个count\_table表，作用类似记录表的版本号，每次更新之后，版本号自增。查询时，检查主从表的版本号是否一致，一致，则走从表查询；否则，走主表查询。→ 一旦主从不同步，所有查询打到主库，风险大，高并发场景下，失去读写分离意义。因此，全局走主库思路不可取。
3. 新想法1：更新时，将更新数据存于集中缓存中；查询时，检查是否最近更新，是，则走缓存查询结果。→ 如何判断查询数据是否在最近更新数据范围内，难度大，出错风险高。
4. 新想法2：根据业务场景，针对需要实时查询场景，采用指定master查询。

## 具体操作

ia-mapper查询的接口支持强制走主库：

1. 若查询参数是Cnd对象，使用com.sankuai.ia.mapper.utils.Cnd#forceMaster方法指定；
2. 非Cnd对象参数，新增了一组接口

```
com.sankuai.ia.mapper.base.BaseSelectMapper#getIdFromMaster  
com.sankuai.ia.mapper.base.BaseSelectMapper#getAllFromMaster  
com.sankuai.ia.mapper.base.BaseSelectMapper#getAllCountFromMaster
```

 赞 成为第一个赞同者

无标签