QCon
全球软件开发大会【上海站】

58速运 微服务落地实践之填坑大法

任桃术@58速运

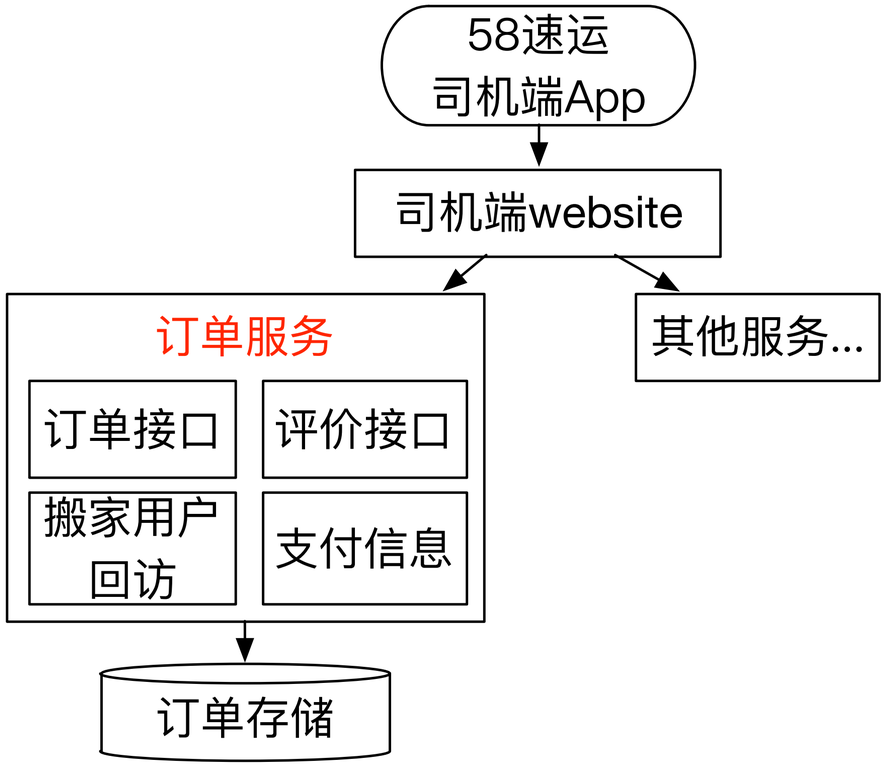
目录

- 存在问题
  耦合度高、伪服务
- 58速运微服务落地实践
  系统拆分实践&注意事项
  伪服务改造实践&注意事项
- 总结

问题 - 耦合度高

- 应用系统层

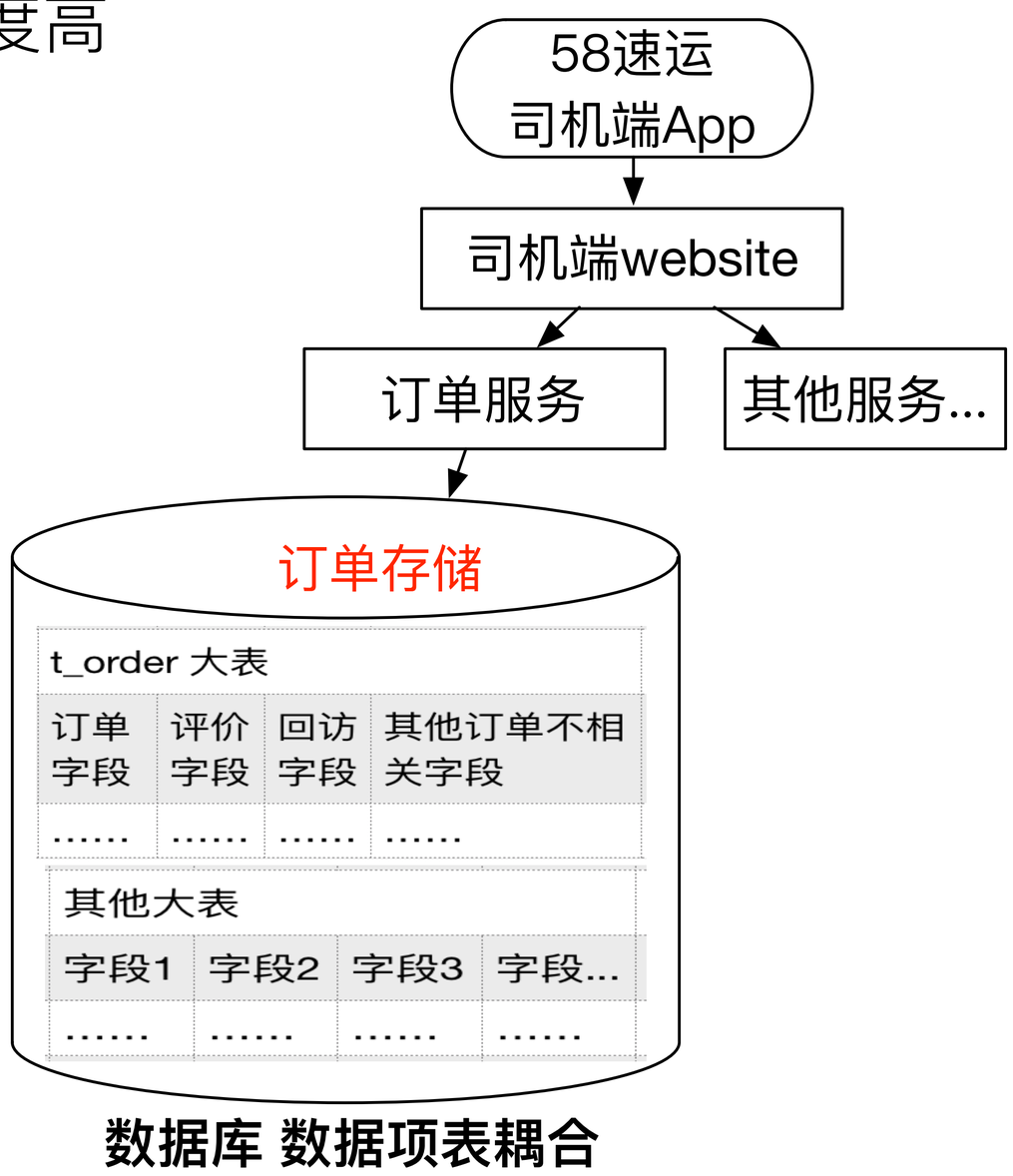
Web应用 模块耦合



服务 接口耦合

问题 - 耦合度高

- 数据库层



数据库 数据项表耦合

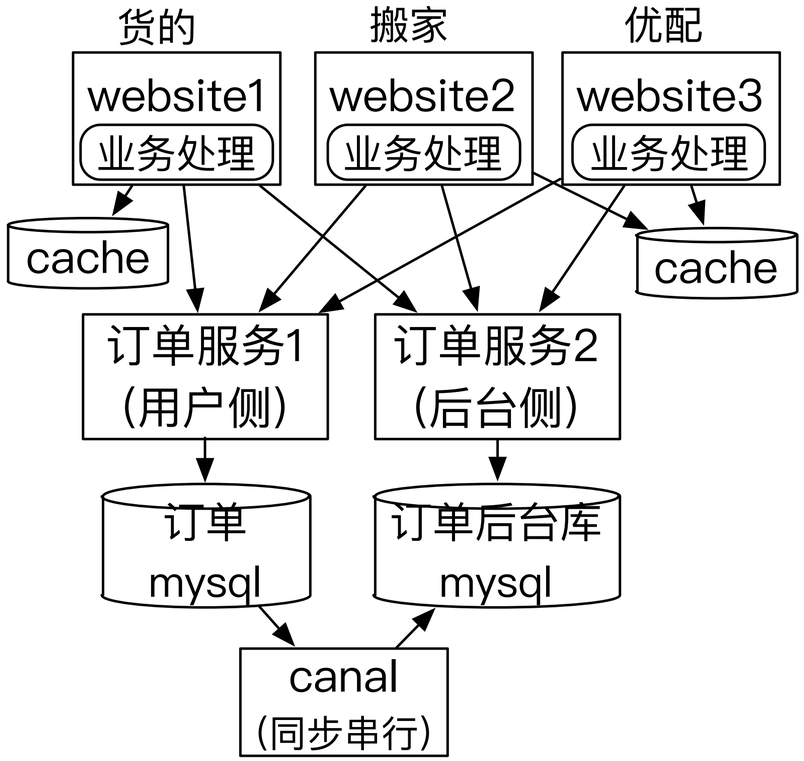
问题 - 伪服务

- 58速运伪服务特点

  1. 简单DAO层封装，不包含公共业务封装

     业务逻辑分散，复制、粘贴，一个小需求n初改、测；

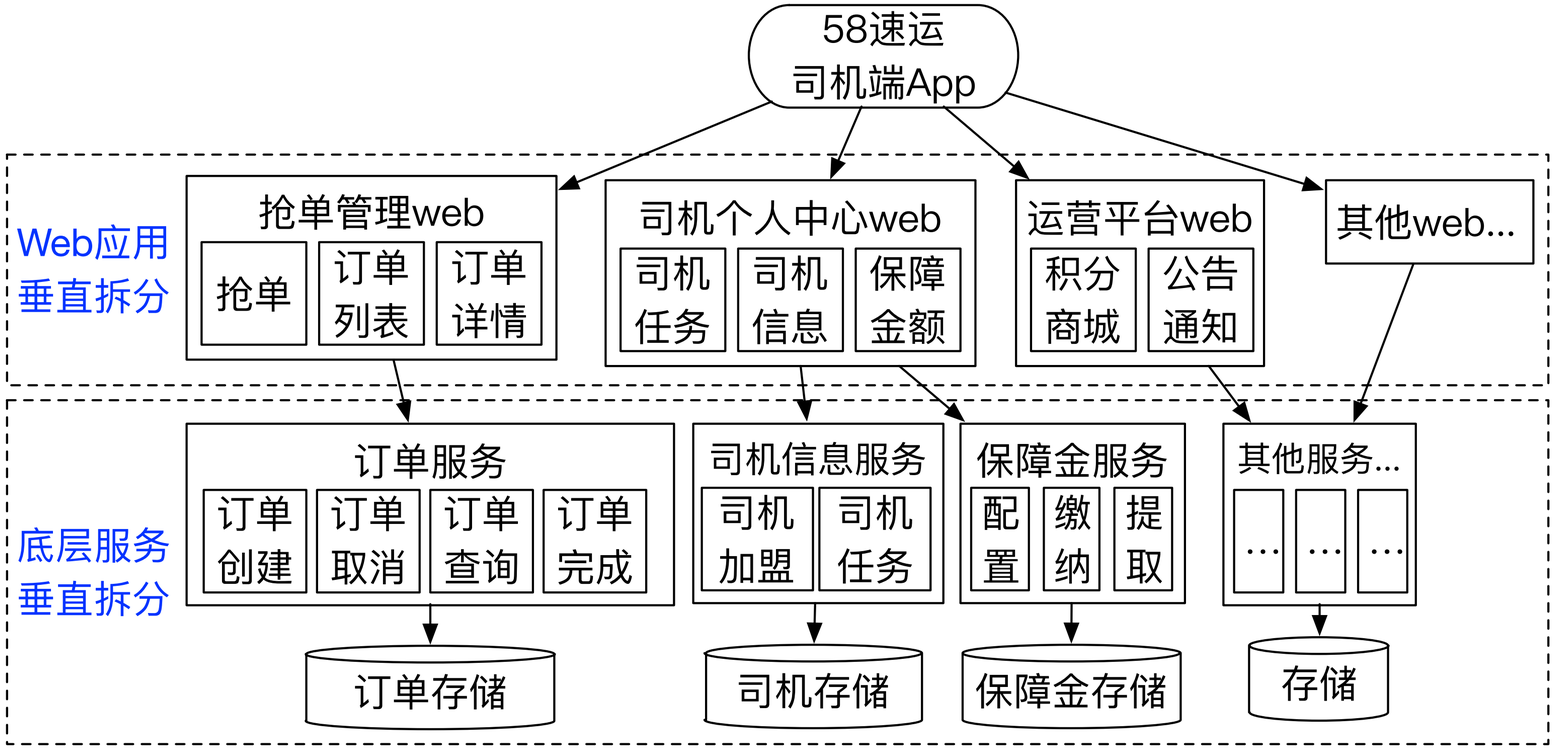
  2.缓存丢在调用方，业务方需要关注存储(用户侧、后台侧)；



怎么办?

58速运微服务落地实践 - 系统拆分

- 业务垂直拆分
  1. 业务梳理，模块归类；
  2. 依赖方梳理；
  3. 构建新系统；
  4. 推进依赖方接入新系统；
  5. 日志确认全部接入新系统；
  6. 下线掉旧系统；



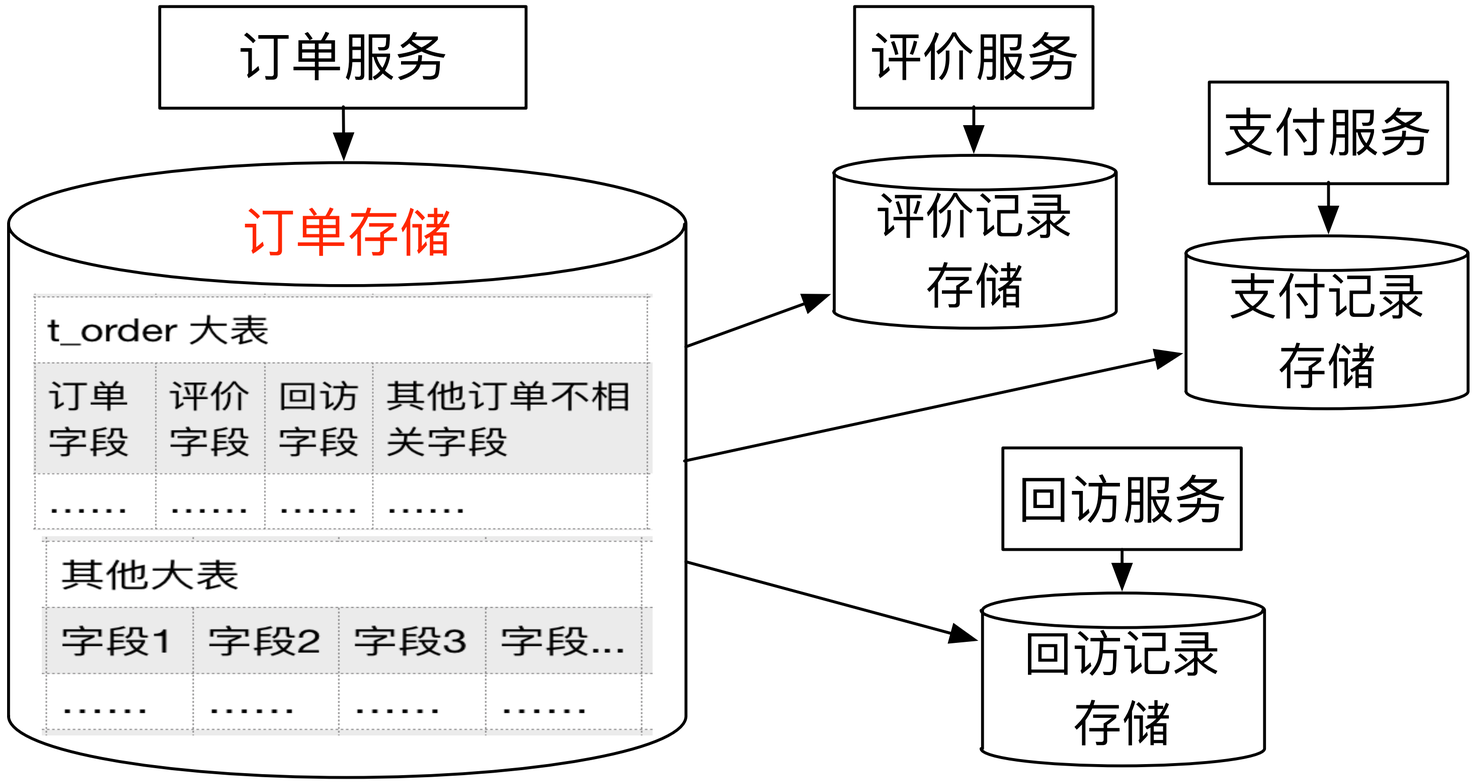
- 注意事项
  1. 避免遗漏依赖方；
  2. 协商好时间节奏，避免较长时间维护两个应用；
  3. 分步操作(step1:域名、接口不变；step2:基于新系统再重构)
  4. 切忌简单拷贝代码了事

58速运微服务落地实践 - 数据拆分

- 数据双写

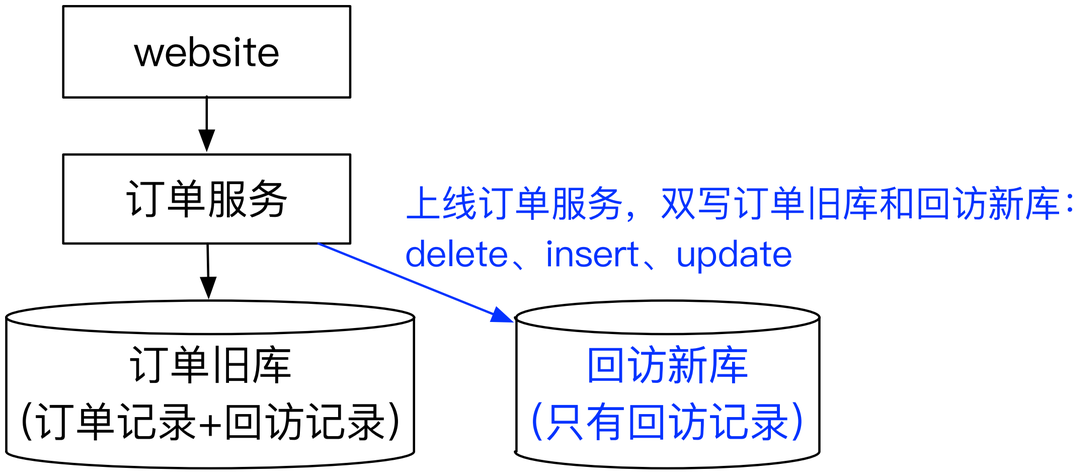
平滑迁移，不影响线上业务；

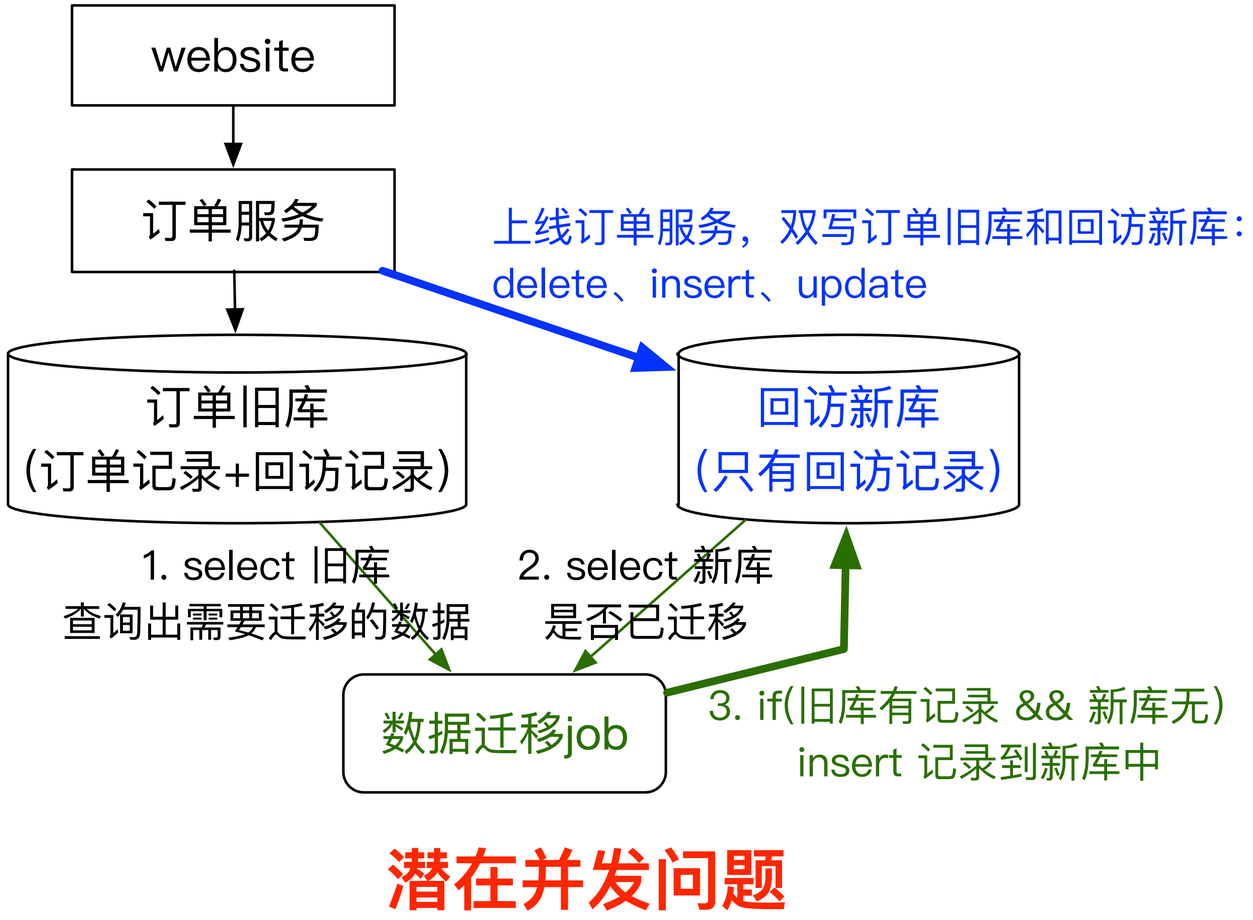
数据一致性；



58速运微服务落地实践 - 数据拆分

- 数据双写(从订单库中迁移出用户回访记录为例)



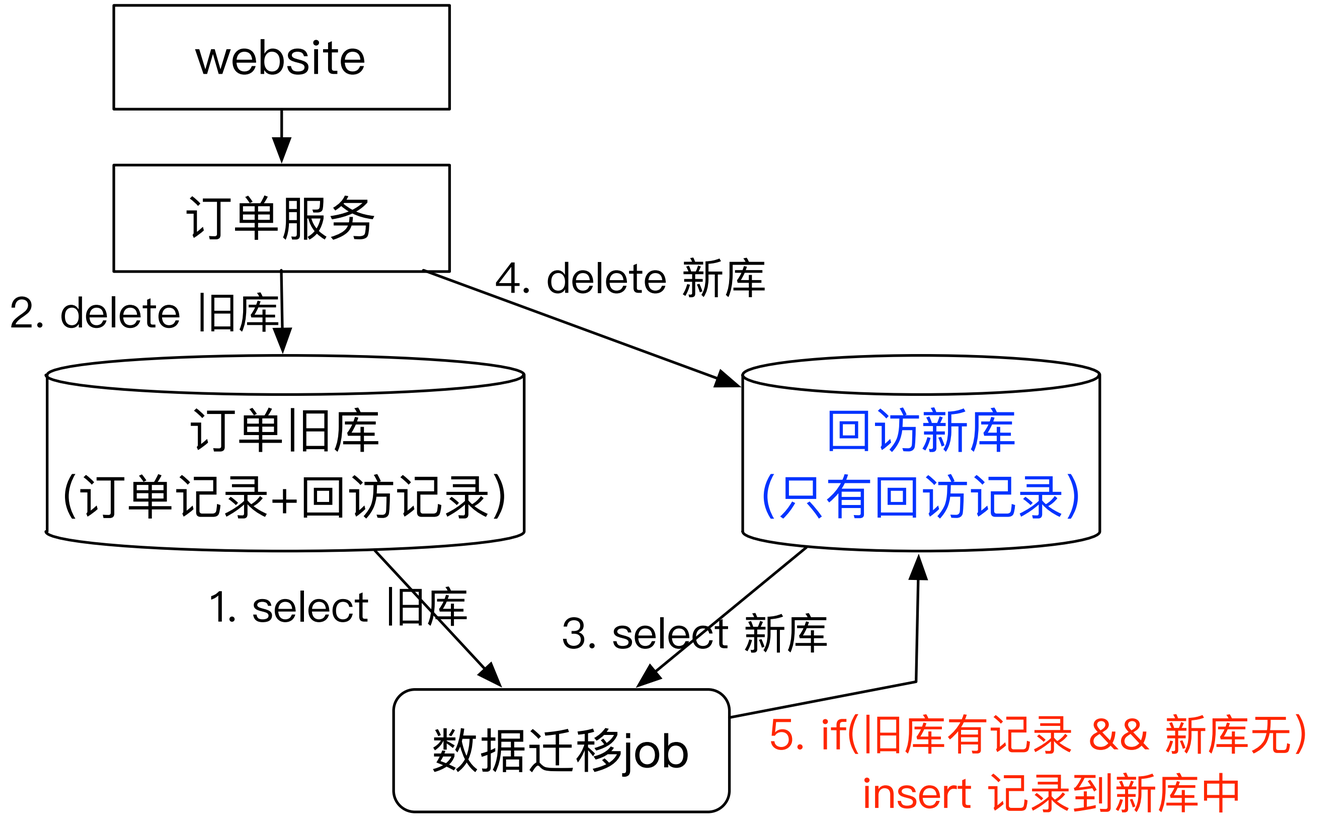


潜在并发问题

58速运微服务落地实践 - 数据拆分

- 并发情况　双写中的delete操作

正常：如果记录未迁移到新库中，只删除旧库记录即可；否则新、旧库都执行删除操作；



潜在问题：回访服务新库中，会多出一些记录

58速运微服务落地实践 - 数据拆分

- 并发情况　双写中的delete操作

潜在问题：

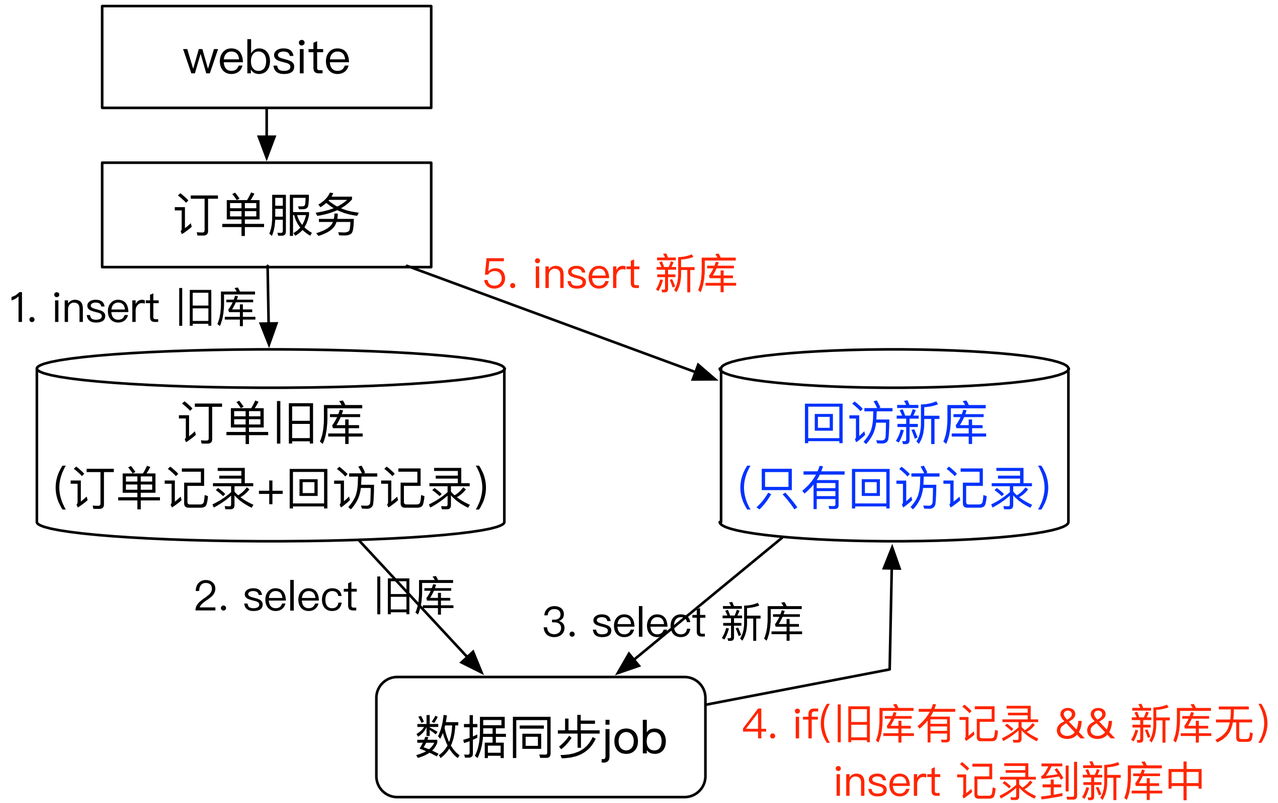
回访服务新库中，会多出一些记录(该删除，却没有删除掉)

58速运实践：

每隔n分钟，校验n分钟前进行迁移的数据，是否在新库中有且旧库中没有存在该情况，则删除新库中的记录；

58速运微服务落地实践 - 数据拆分

- 并发情况　双写中的insert操作



潜在问题：回访服务新库中，会多出一些记录

除了数据校验外，insert时 还有其他避免并发问题的方法么?

58速运微服务落地实践 - 数据拆分

- 并发情况　双写中的insert操作

潜在问题：

回访服务新库中，会多出一些记录(重复insert)

58速运实践：

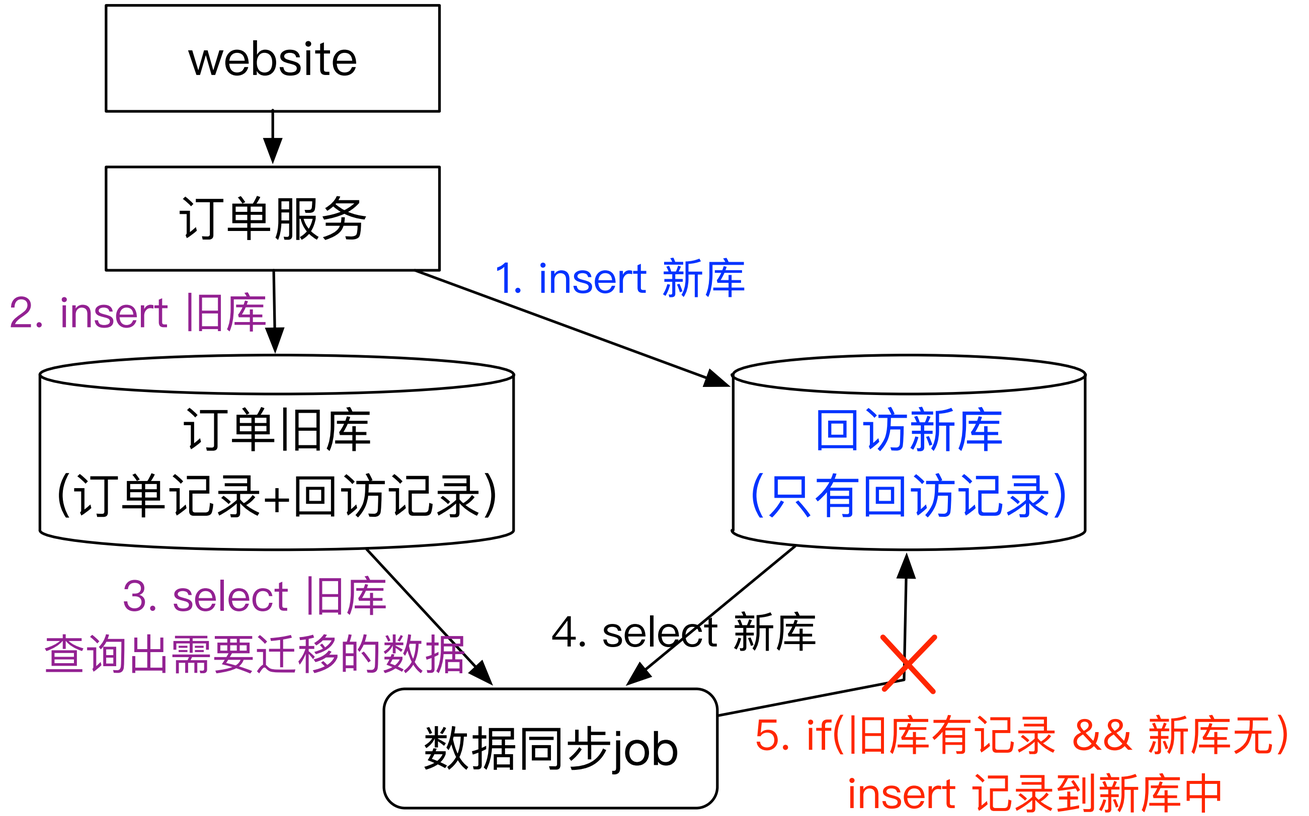
先执行新库的insert，

step5条件不成立；

(stpe2、step3串行)

如果step2 insert旧库失败，

需要回滚新库insert操作；



58速运微服务落地实践 - 数据拆分

- 并发情况　双写中的update操作

58速运实践：

update拆分为对新库先进行delete操作，再进行insert操作

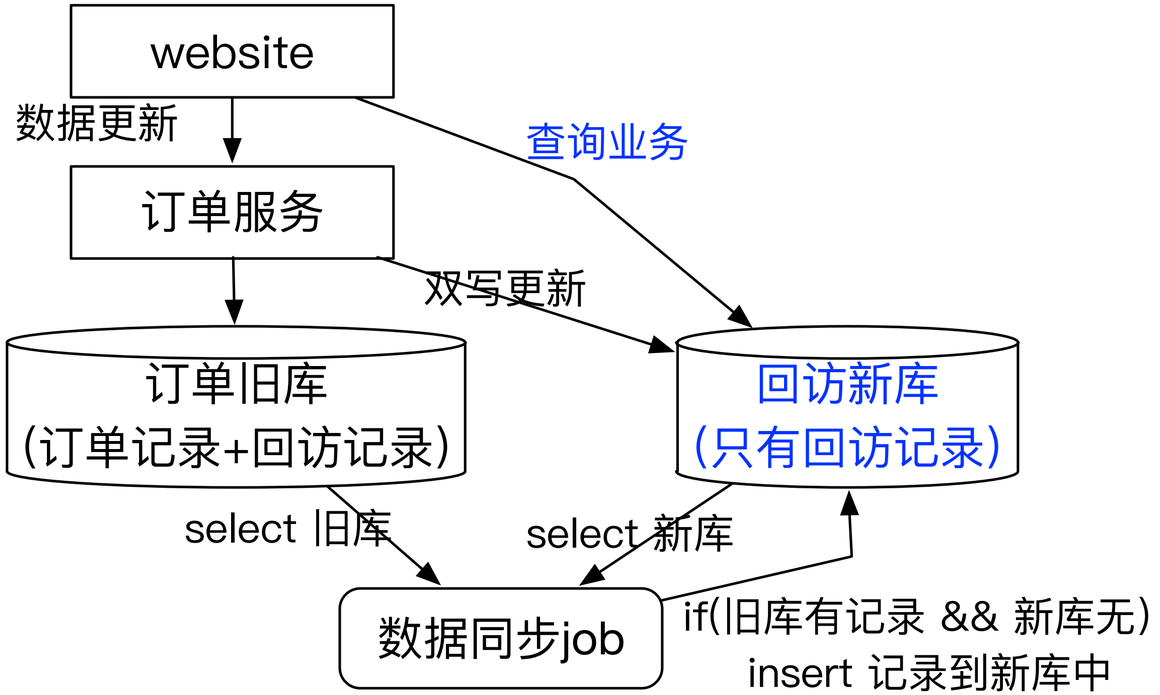
双写的delete、insert操作解决后，双写update操作也就解决了；

校验特别重要：

并发数据一致性校验；

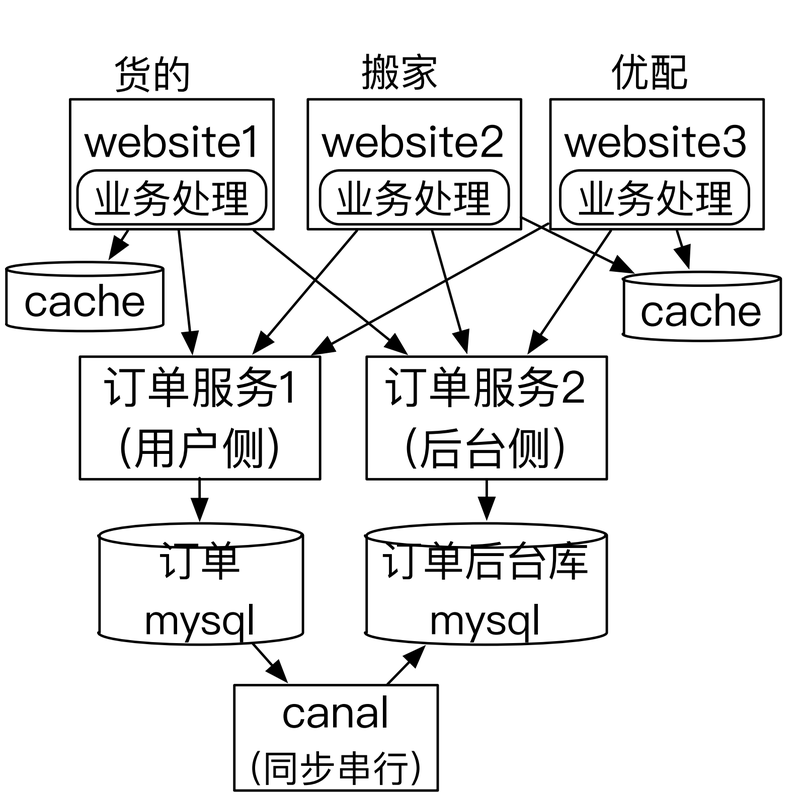
抽样校验；

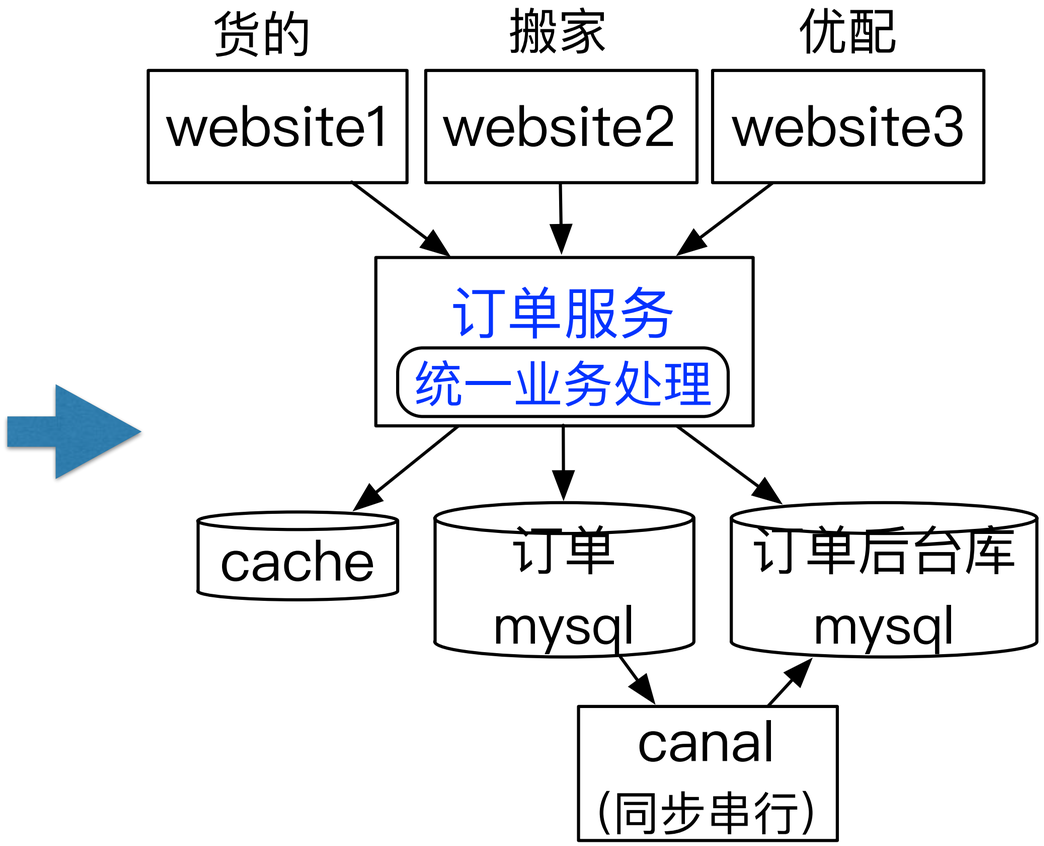
查询业务先切换；



58速运微服务落地实践 - 伪服务改造

- 统一业务下沉为服务
- 统一一个订单服务

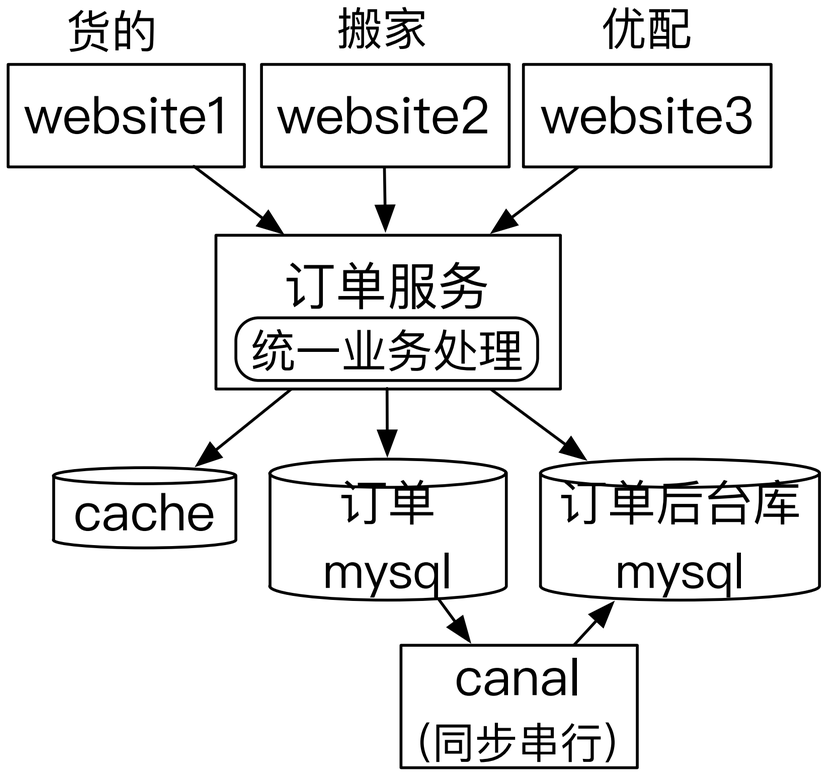


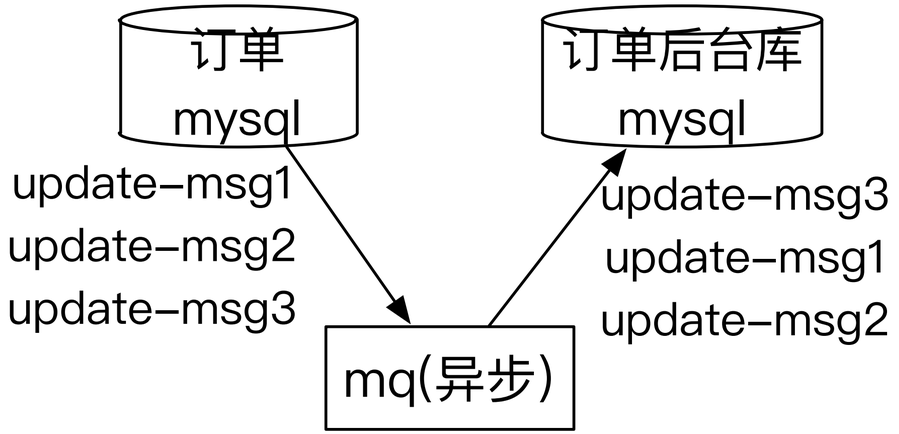



```
统一业务处理
if(货的){
    ……
}else if(搬家){
    ……
}esle if(优配){
    ……
}
```

58速运微服务落地实践 - 服务改造

- 串行同步数据，性能问题



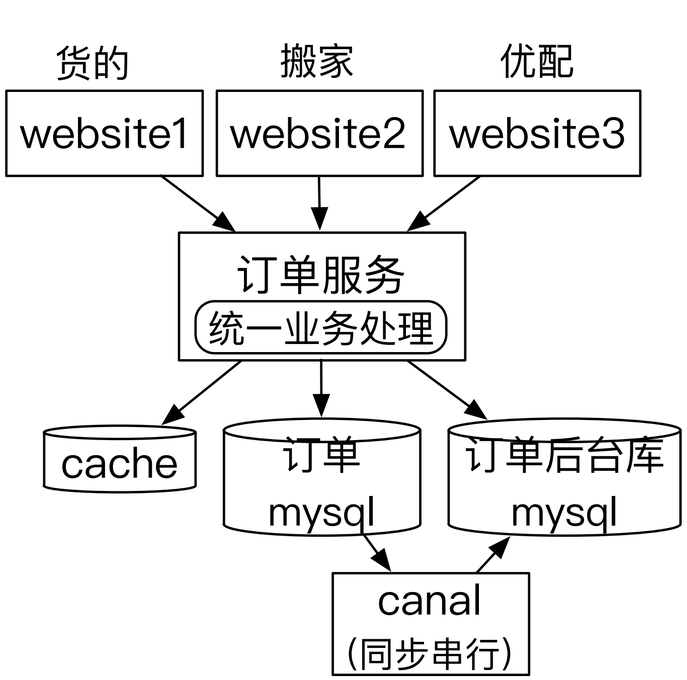


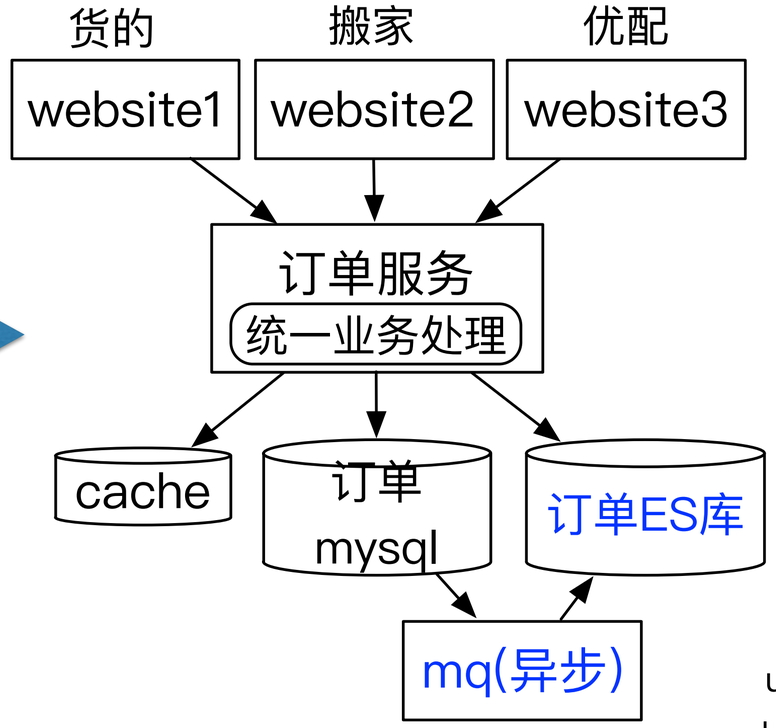
异步同步数据，乱序导致数据不一致

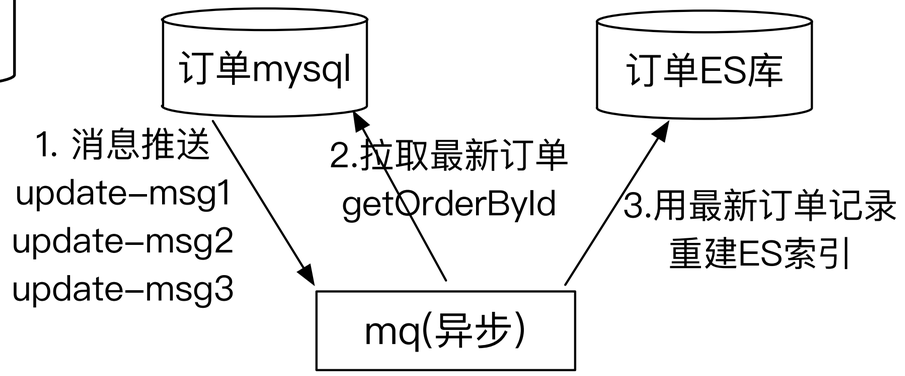
不要为小概率事件，影响整体

58速运微服务落地实践 - 服务改造

- 异步化&ES







最终一致

总结

- 存在问题

  耦合度高、伪服务
- 58速运微服务落地实践

  系统垂直拆分：

  1.业务梳理；2.沟通推进；

  数据平滑迁移：

  1.双写校验；

  服务改造：

  1.统一业务下沉为服务，一个地方改动即可；

  2.不要为小概率事件，影响整体；

  3.数据最终一致性实践；