

# 订单中心，究竟分还是合？

58沈剑



# QCon

全球软件开发大会

## 成为软件技术专家的 必经之路

[北京站] 2018

2018年4月20-22日 北京·国际会议中心

**7折** 购票中, 每张立减2040元  
团购享受更多优惠



识别二维码了解更多



# 极客时间

重拾极客精神·提升技术认知

## 下载极客时间App

获取有声IT新闻、技术产品专栏，每日更新



扫一扫下载极客时间App

主办方 **Geekbang** 极客邦科技 **InfoQ**

# AiCon

全球人工智能与机器学习技术大会

助力人工智能落地

2018.1.13 - 1.14 北京国际会议中心



扫描关注大会官网

# 关于-我



- “架构师之路” 作者，深夜写写技术文章
- 百度 - 高级工程师
- 58同城 - 高级架构师，技术委员会主席，技术学院优秀讲师
- 58到家 - 高级技术总监，技术委员会主席
- 58速运 - CTO
- 本质：技术人一枚

# 目录

- 订单中心，分分合合，架构方案
- 订单中心，由分到合，架构迁移
- 总结与启示

# 一、订单中心，分分合合，架构方案

# 订单中心，业务介绍

- 数据量大，并发量大
- 不同业务订单异构
- 前台侧，例如用户中心，有列表统一展现需求
- 后台侧，例如运用后台，各属性上都可能有检索需求
- ...



# 方案一，快速实现

家政类别，订单表：

order(oid, uid, c1, c2, c3)

通过组合索引满足组合查询需求：

index\_1(c1,c2) index\_2(c2, c3) index\_3(c1, c3)

业务发展，**新增速运类别**，订单表升级为：

order(oid, uid, c1, c2, c3, c10, c11, c12, c13)

其中：

c1,c2,c3是家政类别属性

c10,c11,c12,c13是速运类别属性

如何满足订单列表需求？

- 通过uid统一查询

新建组合索引满足新的查询需求？

- 不敢想有多少个索引才能覆盖所有两属性查询

- 三属性查询，根本玩不下去？

## 方案二，分

### 按照业务做垂直拆分

order\_jiazheng(oid, uid, c1, c2, c3)

order\_suyun(oid, uid, c10, c11, c12, c13)

每个业务有**订单服务**，**搜索服务**，分别满足垂直的数据库查询，订单检索需求

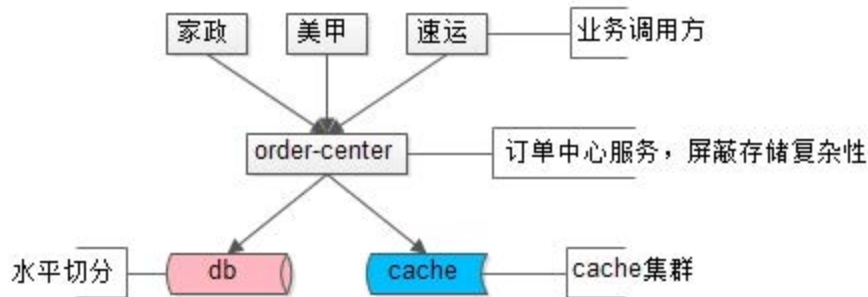
维护在不同的部门的数据库、服务、搜索，**看上去各业务线灵活性强，这恰恰是悲剧的开始：**

- 按照uid来查询怎么办，查询自己所有订单？
- 跨品类查询怎么办，按照订单状态，支付状态查询？
- 后台系统，按照不同属性维度查询怎么办？
- 统一的对账需求，风控需求，数据仓库需求怎么办？
- 技术范围的扩散，有的用mongo存储，有的用mysql存储，有的自研存储

...

多业务订单中心，要合！

# 要点一：统一订单中心



- 技术问题统一解决

- 分布式订单id生成
- 数据量大，读性能，高可用，统一技术体系
- 数据仓库

- 业务问题统一解决

- 订单列表拉取
- 订单状态，支付状态列表拉取
- 对账，风控

如何满足业务侧个性化订单属性需求？

## 要点二：可变属性存储

oid	uid	time	type	订单状态	支付状态	金额	ext
o1	u1	t1	家政	已下单	已支付	80	{"小区": "天通苑东一区"}
o2	u2	t2	速运	已下单	未支付	120	{"车型": "金杯"}

- 通用字段，有统一查询需求的字段统一存储
- 个性字段，用可变json统一存储
- 通过type来理解json的含义

如何满足业务侧**个性化订单属性查询**需求？

## 要点三：统一搜索服务



- **外置索引，统一搜索服务**，搜索解耦，屏蔽底层复杂性
- **写请求**：同步（RPC调用）或者异步（MQ）
- **普通读请求**：走db+cache
- **搜索请求**：走搜索服务
- **副作用**：一致性，需要有检测机制/定期重建机制

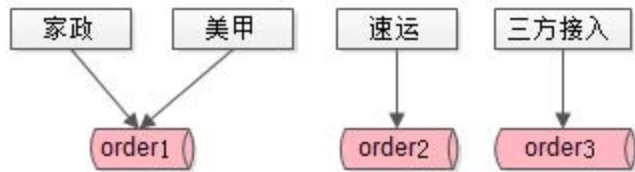


架构现状，就像生活一样，总不尽如人意...

## 二、订单中心，由分到合，架构迁移

## 当有恶化苗头时...

- 相似业务直接访问一个数据库，耦合
- 多个业务访问各自库，统一性差

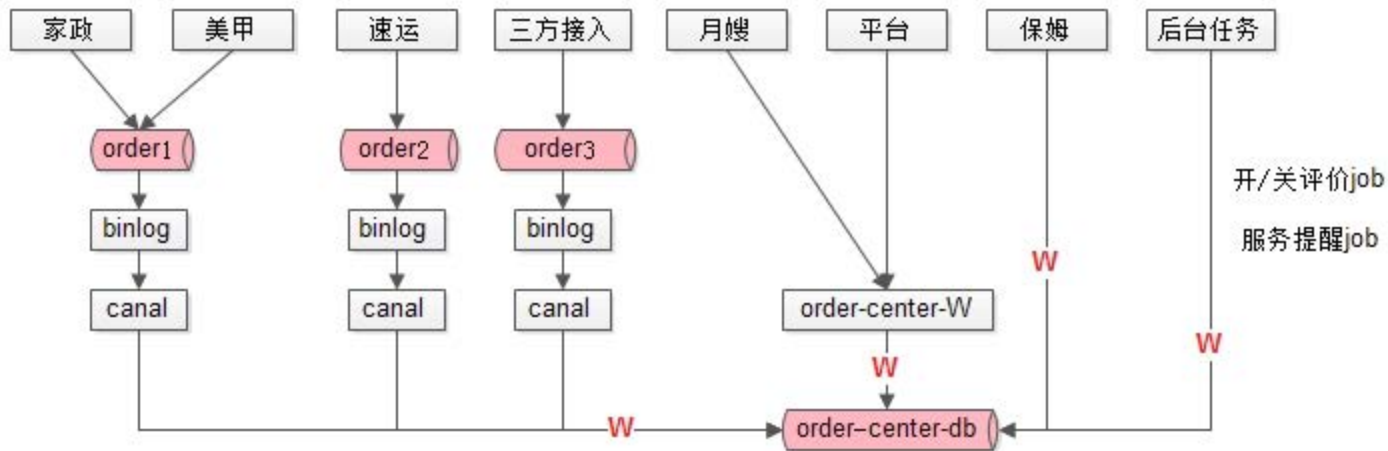


- 潜在问题
  - 58到家APP很难统一展现订单
  - 后台对订单的查询，修改需求很难满足
  - 统一仓库没法满足
  - 统一对账，风控没法满足
  - 新的业务无所适从

# 确定方向与路线

- 总的方向是**统一**
- 先**数据收口**，数据统一
- 再**服务收口**，服务统一

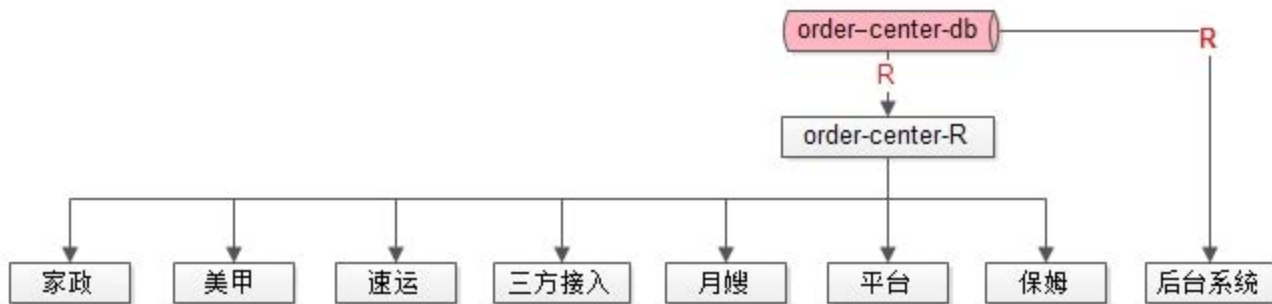
## 数据收口（写）



- 统一order-center-db
- 新建order-center-W服务
- 已有订单通过canal同步数据
- 新增业务统一使用统一数据

- 解决问题
  - 后台统一订单需求，得到满足
  - 统一数据仓库得以建设
  - 统一对账，风控可以逐步开展
  - 新的业务能迅速展开

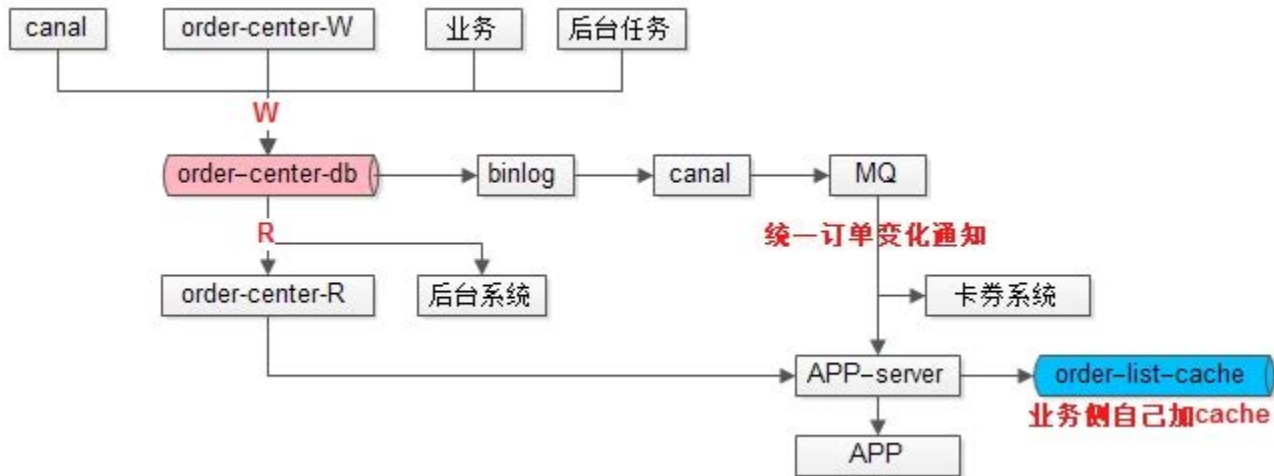
## 数据收口（读）



- 新建order-center-R服务
- 尽量通过统一服务读取订单数据

- 潜在问题
  - 写入点多
  - 读取虽然收口，但order-center-R加不了缓存
  - 订单中心压力上来之后，数据库压力大
  - 统一发卡发券关注订单变化事件

## 统一对外事件 外部增加缓存



- **被迫**再拉出一条统一分支，binlog+canal+MQ，实现**统一的订单变化事件通知**
  - 解决统一通知需求
- 由于多写入，order-center无法加缓存，**被迫调用方自己做缓存**，接收统一通知淘汰缓存
  - 解决统一订单列表需求
  - 解决缓存需求

众多补丁下，看似实现了业务功能，但是...



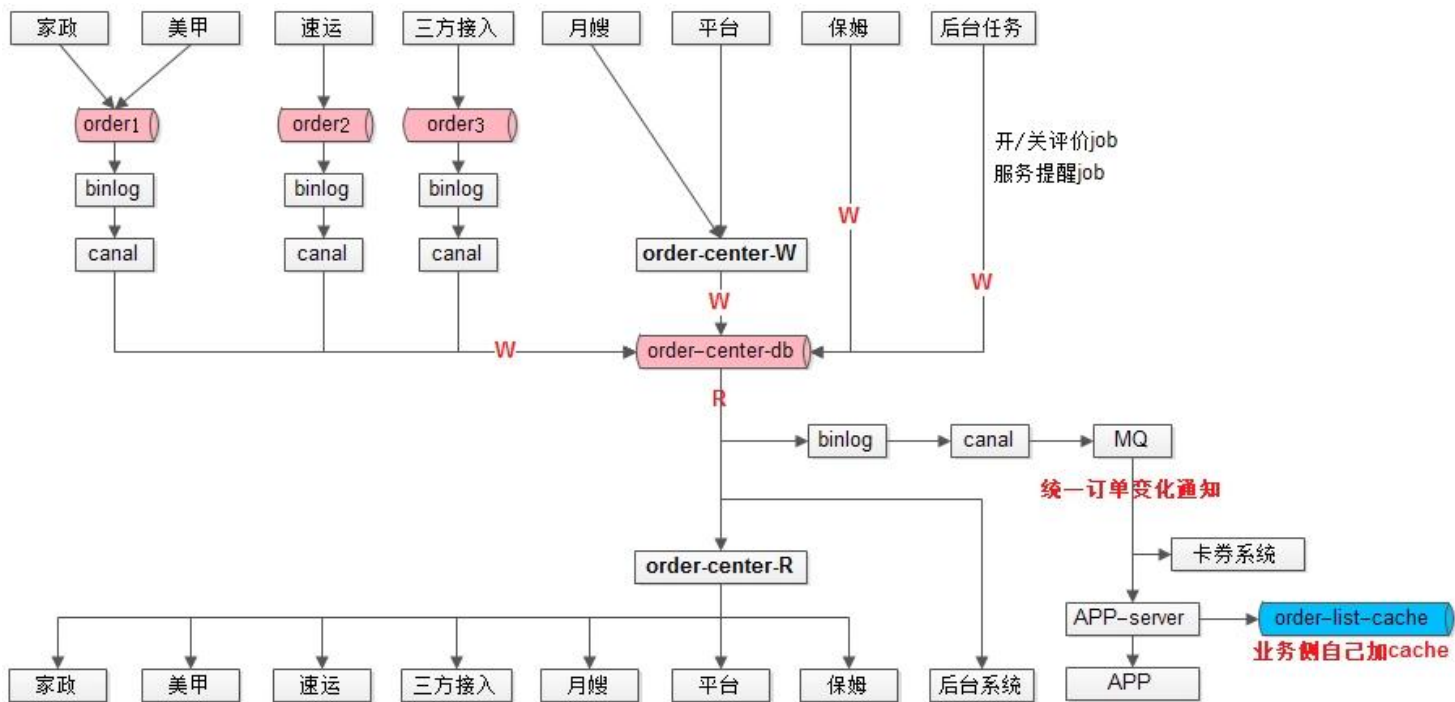
## 真的理想么？

- 直接读写数据库
- 服务加不了缓存

以上两点是服务化大忌

- RDS不支持binlog
- Canal无法使用

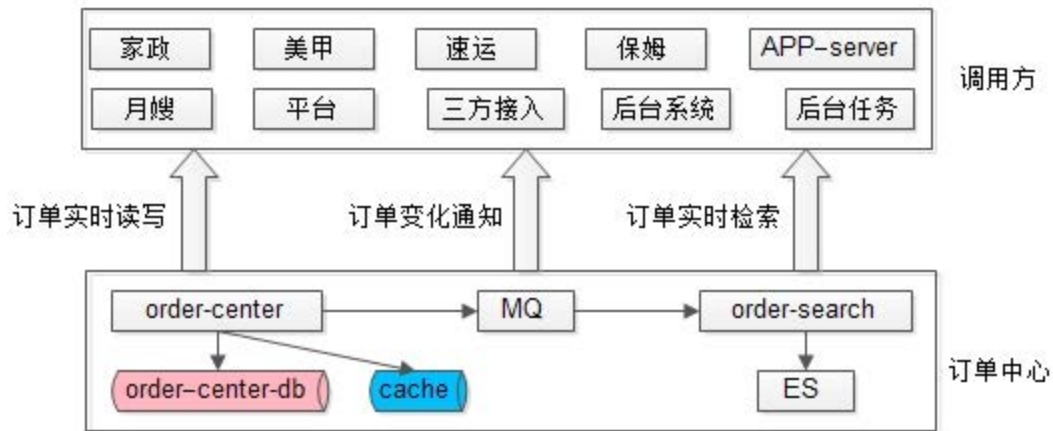
以上两点是阿里云限制



# 服务化不是这么玩的，应当...

- **数据私有**，禁止任何上游，绕过服务，读写数据库
- 职责集中，**有限功能**，**无限性能**，高内聚，低耦合
- **屏蔽复杂性**
  - 内部复杂性不可见
  - cache不可见，分库分表不可见，存储引擎不可见
  - 搜索复杂性不可见，内部数据结构不可见

# 订单中心



## • 职责集中

- **订单实时读写**，统一提供按uid/oid/status/time查询，RPC接口
- **订单通知变化**，统一提供订单变化通知
- **订单复杂检索**，统一提供按照各种个性化属性查询，RPC接口

## • 复杂性屏蔽

- 屏蔽分库分表
- 屏蔽cache
- 屏蔽内部MQ通知
- 屏蔽搜索具体实现

### 三、总结与启示

## 订单中心架构

- **快速实现**
  - 扩展列实现个性化，通过组合索引满足查询需求
- **分**
  - 扩展表实现个性化，统一业务难以实现
- **合**
  - 统一订单中心服务
  - 可变属性统一存储
  - 统一订单搜索服务

## 订单中心迁移

- 先数据收口
- 变化通知收口
- **职责与复杂性收口，完成统一订单中心**
  - 提供实时订单读写RPC
  - 提供订单通知变化CallBack
  - 提供实时订单检索RPC

# Q&A

“架构师之路” 公众号

# 谢谢！



# THANK YOU

---

如有需求，欢迎至 [ 讲师交流会议室 ] 与我们的讲师进一步交流

