58速运
百万单量用车O2O架构优化实践

58沈剑
shenjian@daojia.com

关于-我

- 百度 - 高级工程师
- 58同城 - 技术委员会主席，高级架构师，技术学院优秀讲师
- 58到家 - 技术委员会主席，技术总监
- **本质：程序员**

微博

微信

目录

- 用车O2O业务简述
- 用车O2O业务难点
- 用车O2O优化与演进
- 总结

一、用车O2O业务简述

用车O2O核心业务

- APP分端：用户端APP，司机端APP
- 核心业务

（1）用户查看司机

（2）用户下单

（3）司机抢单 + 用户反选

（4）记录里程

二、用车O2O业务难点

用车O2O业务难点

- 业务难点

为什么车在山里，在海里？

（1）用户查看司机：如何保证司机位置的准确性？

（2）用户下单：如何快速进行订单派发？

下单了木有司机抢，着急

（3）司机抢单 + 用户反选：如何高效进行订单推送？

Push的到达率好低

（4）记录里程：如何准确记录行车里程？

跑了10公里，APP里程却是5公里，司机亏大了

三、用车O2O优化与演进

（一）地理位置查询-数据库实现

- 如何简单实现位置上报 + 位置搜索

sj_uid	jingdu	weidu

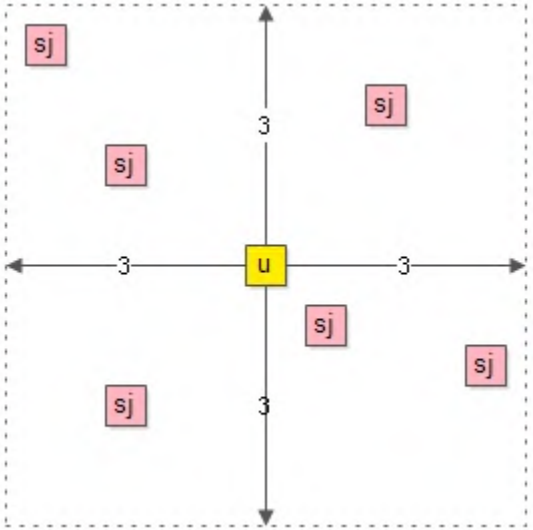
driver

（1）数据库存储司机经纬度

（2）司机每隔10s上报自己的经纬度，更新数据库

（3）用户查询，上传自己的经纬度，去数据库中查询

```
SELECT sj_uid FROM driver WHERE
        (jingdu > $jd - 3) AND (jingdu < $jd + 3) AND
        (weidu > $wd - 3) AND (weidu < $wd + 3)
```



（一）地理位置查询优化-倒排

- 存在什么问题

（1）数据库写压力大

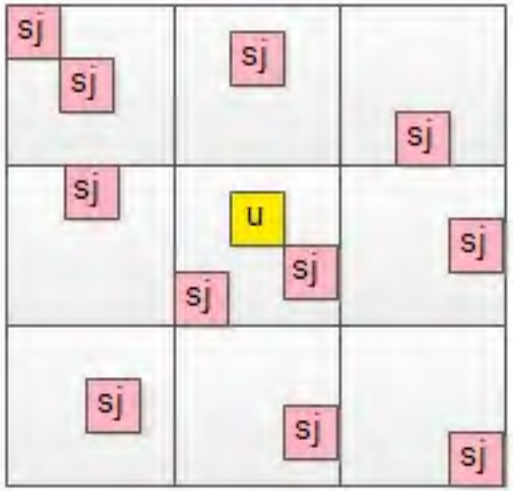
（2）查询效率低

- 优化

（1）减少写压力，HOW？

（2）优化查询复杂度，HOW？

- 正排 sj_uid => area_id
- 倒排 area_id => set<sj_uid>



（二）订单派发-for循环

- 如何简单实现订单派发

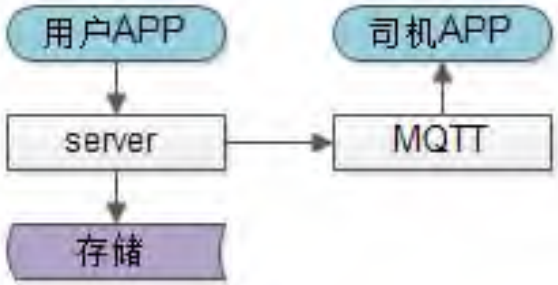
（1）下单：校验下单频率 => 校验订单合法性 => 敏感词校验 => 优惠券验证 => 插入订单 => ...

（2）派单：搜索策略 => 搜索符合条件的司机 => 搜出50个司机

for(50个司机){

计算补贴 => 写入抢单表 => MQTT推送

}

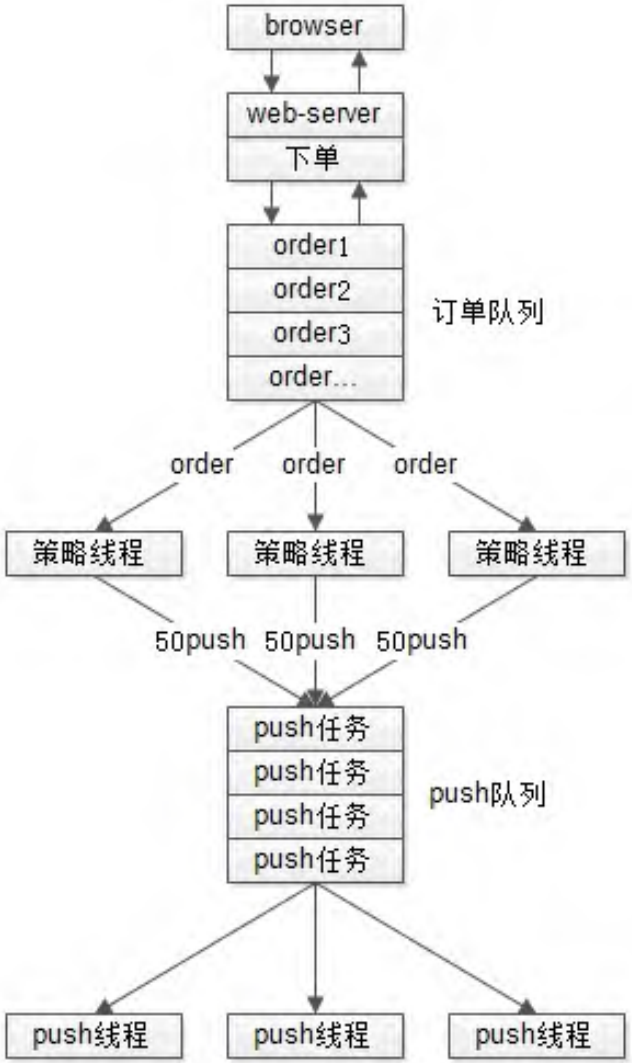


（二）订单派发优化-异步

- 存在什么问题？
- 优化

（1）下单和派单异步

（2）派单并行



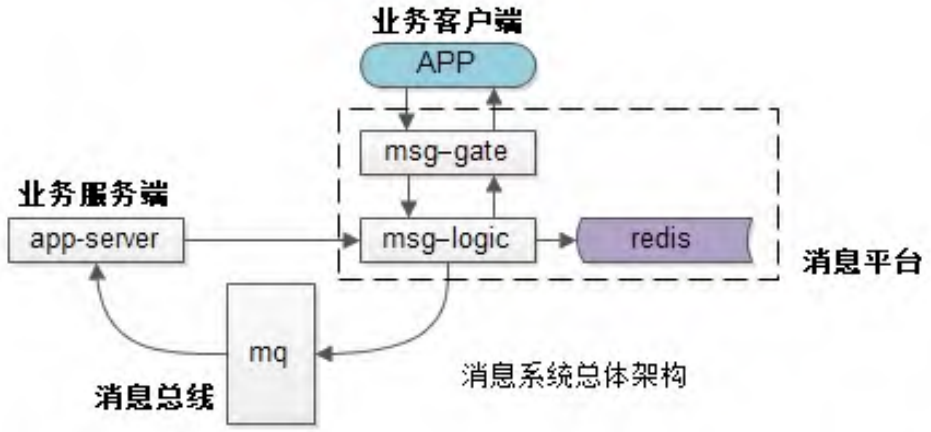
（三）订单推送优化-分布式推送通道

- 存在什么问题？MQTT是单点
- 优化：通用TCP分布式推送系统
- 核心流程

（1）登录

（2）c2s消息发送

（3）s2c消息发送



（四）里程上报优化-效率优化

- 每隔10s上报GPS，数据表记录历史轨迹
- 存在什么问题

（1）http上传GPS，web-server压力大

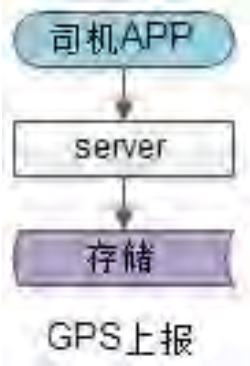
（2）数据库写压力大

- 优化手段

（1）通过TCP通道上报GPS

（2）缩短上报时间

- 存在什么问题：里程数不准
- 解决方案？



四、总结

总结

- 总结

（1）GPS上报优化：“写入缓存，定时刷库”能有效降低写压力

（2）GPS查询优化：area_id => set<sj_uid>倒排优化

（3）订单派发优化：下单与派单异步，派单任务并行

（4）订单推送优化：TCP分布式推送系统

（5）里程上报优化：客户端实时记录，批量上传 能 保持里程精度，并降低数据库写压力

Q&A

- 谢谢！

“架构师之路”公众号

Thanks!