

# 58到家消息系统设计与开发实践

58到家 - 任桃术  
rentaoshu@daojia.com



# 目录

---

- 背景
- 整体架构
- 系统重点  
协议；快速重连

# 背景 - 58速运



## 问题

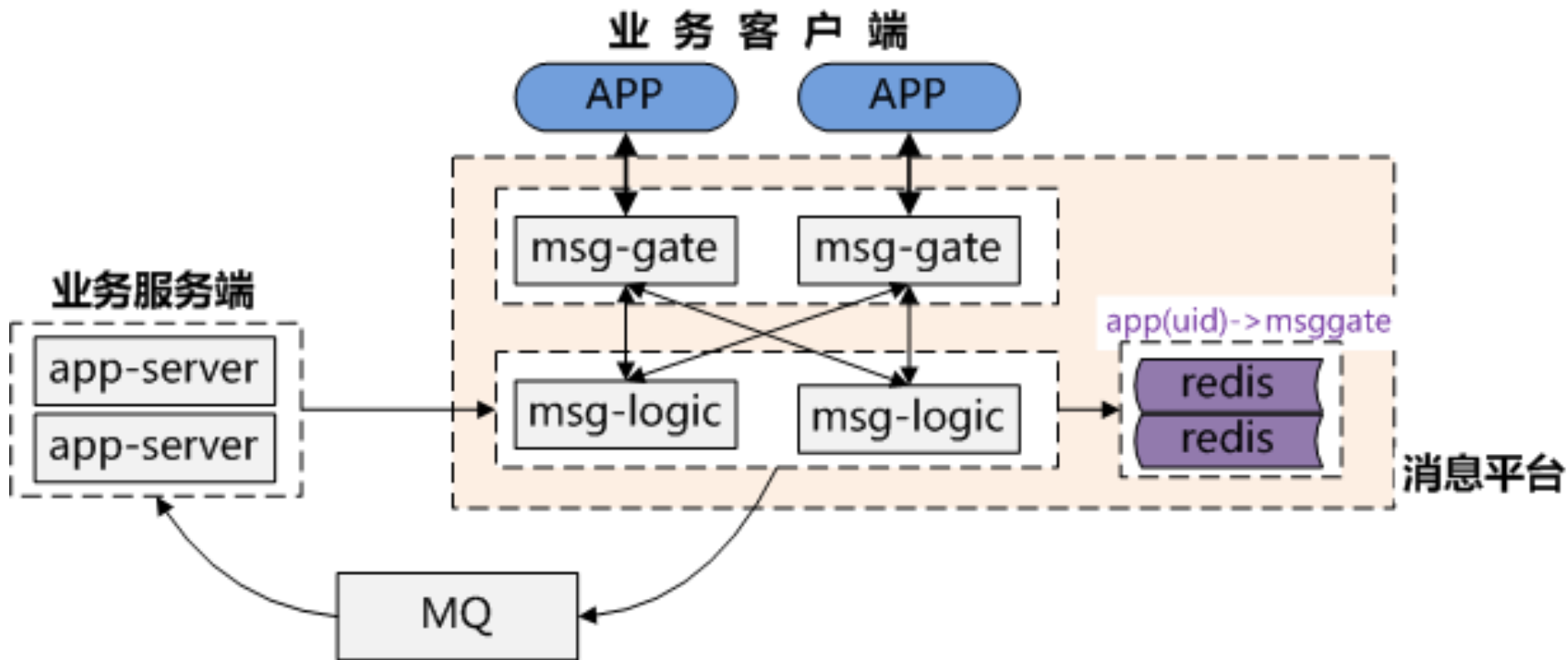
- 1、司机端GPS上报后端压力大
- 2、向多个司机推送订单，可用性、到达率、性能有提升空间

# 主要业务场景

---

- 58速运APP端(app -> appserver)
  - 往app-server发送GPS消息
  - 用户往app-server发送下单消息
  - 司机往app-server发送抢单信息
- app-server端(appserver -> app)
  - 往司机端APP群发订单消息
- 其他实时消息  
(暂时没有离线消息存储)

# 整体架构



# 系统重点

---

## 协议设计

- 扩展性
- 可调试性
- 异步支持

## 快速重连

- 无线网络不稳定，TCP时断时连；

# 协议设计

---

- 扩展性

如何支持多种类型的报文(登录、app业务、appserver推送、keepalive)?

如何支持业务包数据接口的变更?

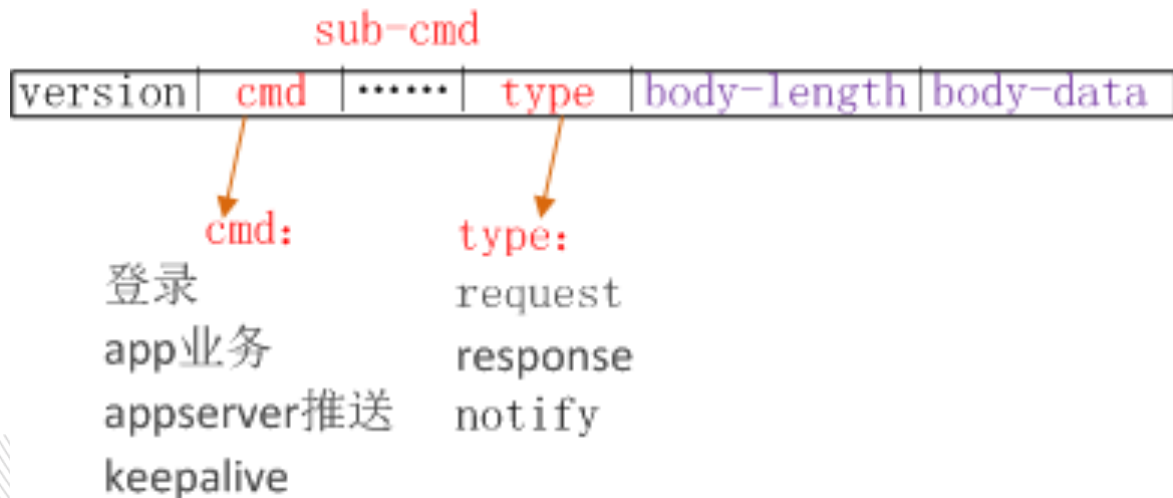
# 协议设计

---

- 扩展性

如何支持多种类型的报文(登录、app业务、appserver推送、keepalive)?

如何支持业务包数据接口的变更?





# 协议设计

---

- 可调试性

假设某个请求包处理失败，我们如何快速知道是APP端的问题，还是msg-gate的问题，还是msg-logic的问题，还是redis的问题？

# 协议设计

---

- 可调试性

假设某个请求包处理失败，我们如何快速知道是APP端的问题，还是msg-gate的问题，还是msg-logic的问题，还是redis的问题？

version	cmd	...	type	log-id	body-length	body-data
---------	-----	-----	------	--------	-------------	-----------

# 协议设计

---

- 异步支持

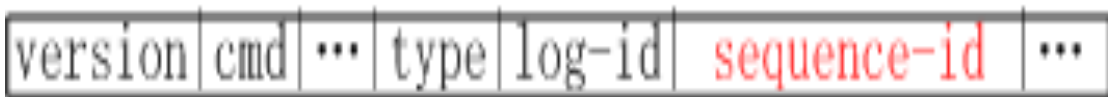
APP端向msg-gate异步投递了a, b, c三个消息, 下游服务处理完后msg-gate回复了APP端x, y, z三个包, 上游服务怎么把abc和xyz匹配上?

# 协议设计

---

- 异步支持

APP端向msg-gate异步投递了a, b, c三个消息, 下游服务处理完后msg-gate回复了APP端x, y, z三个包, 上游服务怎么把abc和xyz匹配上?



+ 消息发送队列(网络框架)

# 快速重连

---

- 无线网络不稳定，TCP时断时连；

APP端和msg-gate之间的Session保持一段时间

(新建Session需要走密钥协商、登录验证，建立安全通讯信道，成本高)

# 快速重连

---

## Session保持引发的问题

- TCP连接断开，Session保持中，app-server推送消息到APP端？
- TCP连接断开后，APP端重连到另外一台msg-gate服务器？

# 快速重连

---

## Session保持引发的问题

- TCP连接断开，Session保持中，app-server推送消息到APP端？

清除msg-gate上的session；

回调msg-logic的unreachable接口(清除redis相应记录)

# 快速重连

---

## Session保持引发的问题

- TCP连接断开后，APP端重连到另外一台msg-gate服务器？

通过redis找到上一次建立session的msg-gate服务器；

清除上一次建立的session；

在重连的新的msg-gate服务器上建立新session和TCP连接；

怎么优化？APP端需要做怎样的处理？



# 总结

---

## 协议设计

- 扩展性
- 可调试性
- 异步支持

参考方案：

version	log-id	sequence-id	cmd	type	body-length	body-data
---------	--------	-------------	-----	------	-------------	-----------

## 快速重连

- 无线网络不稳定，TCP时断时连；

参考方案：

appserver -> app，清除session，同步redis状态；

app -> appserver，通过APP端记录上一次连接的服务器 进行快速重连；

Q & A

谢谢

“架构师之路” 公众号

