

# 期中测试获奖用户名单及参考答案：通达系统架构设计

2022-04-11 李智慧

《李智慧·高并发架构实战课》

[课程介绍 >](#)



讲述：李智慧

时长 02:07 大小 1.95M



你好，我是李智慧。今天我们来公布一下期中测试的获奖用户名单和对应的答案。

我们期中测试的要求是写一个同城快送业务的系统架构设计文档，这个测试主要考察的目标包括：使用 **UML** 进行系统建模的能力，用文档表达设计思路的能力，完整思考一个系统整体架构的能力，以及识别设计落地关键技术问题及对策的能力。


在这里，感谢各位同学的积极参与，我收到了多位同学提交的答案，所有提交的答案都满足设计文档的格式要求，在设计的完整性上也基本达到了测试题目的要求，相信提交答案的同学也一定对如何从全局思考一个系统的架构设计，如何将设计思路表达出来有了感性而深刻的认识。

正如昵称为“外星人”（ ）的同学在文档中所说：“一开始，其实认为自己在理论 + 思想上，已经基本算是及格的。但是，等到真的给自己一个业务，要去做设计的时候发现，并不是自己预想的那么简单。因为，这时候，你需要思考的并不是某一个单一的业务场景：秒杀业务，地理

位置解决方案，全球化用户加速访问之类的。这时候，你面临的是一个更加真实的挑战，现在让你来做设计，你会怎么设计。”

我相信，这种对全局现实性问题的思考，会对你的思考问题方式和事物认知能力产生长远的影响，你会注意到以前不曾注意的东西，也会思考以前忽略了的一些问题。

好，话不多说，在此，我们挑选出了 3 位获赞数最多且最符合作业要求的 3 位同学，获奖名单如下：

中奖用户	奖励
	与李智慧老师1v1语音沟通
小锅米线	与李智慧老师1v1语音沟通
peter	与李智慧老师1v1语音沟通

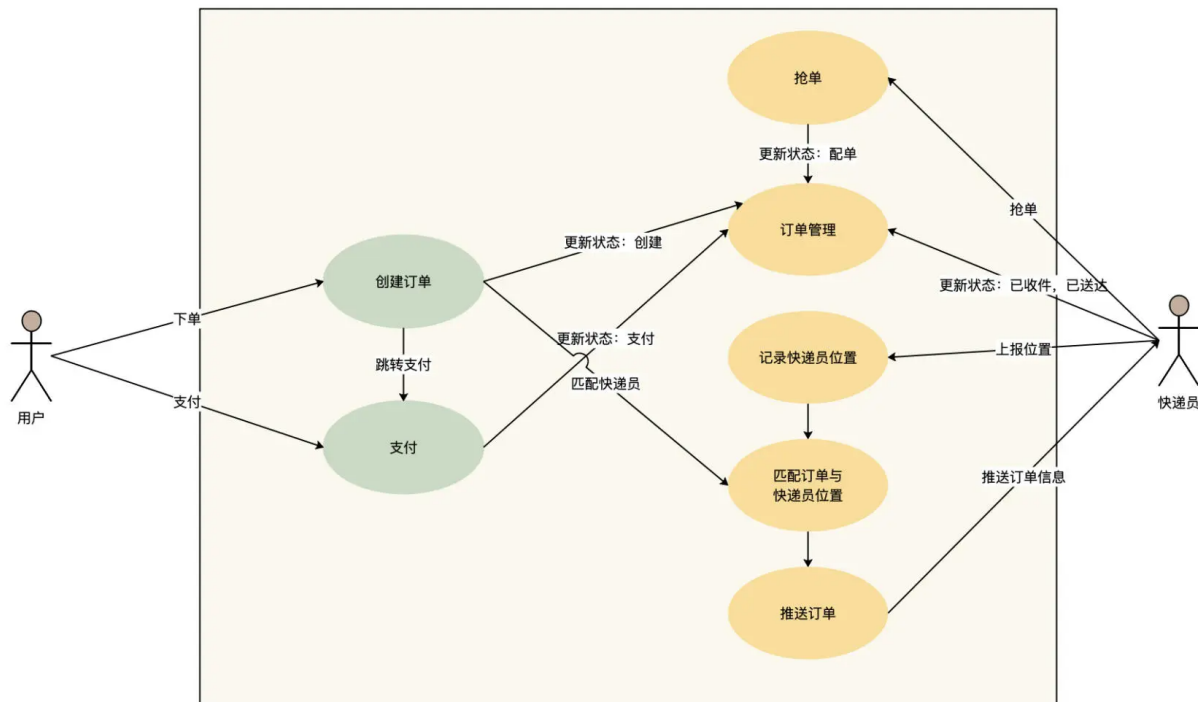
恭喜这 3 位同学！这里我也给出一个我写的答案，供你参考。

参考答案：通达系统架构设计

通达公司上级母公司是行业顶尖的物流配送企业，依托母公司的行业资源，我们将在全国多个主要城市开展一对一同城快送业务。本系统架构设计目标旨在快速利用公司现有资源，结合公司初期运营目标，开发一个能支持公司当下运营，也能支持公司未来发展的互联网应用系统。

需求分析

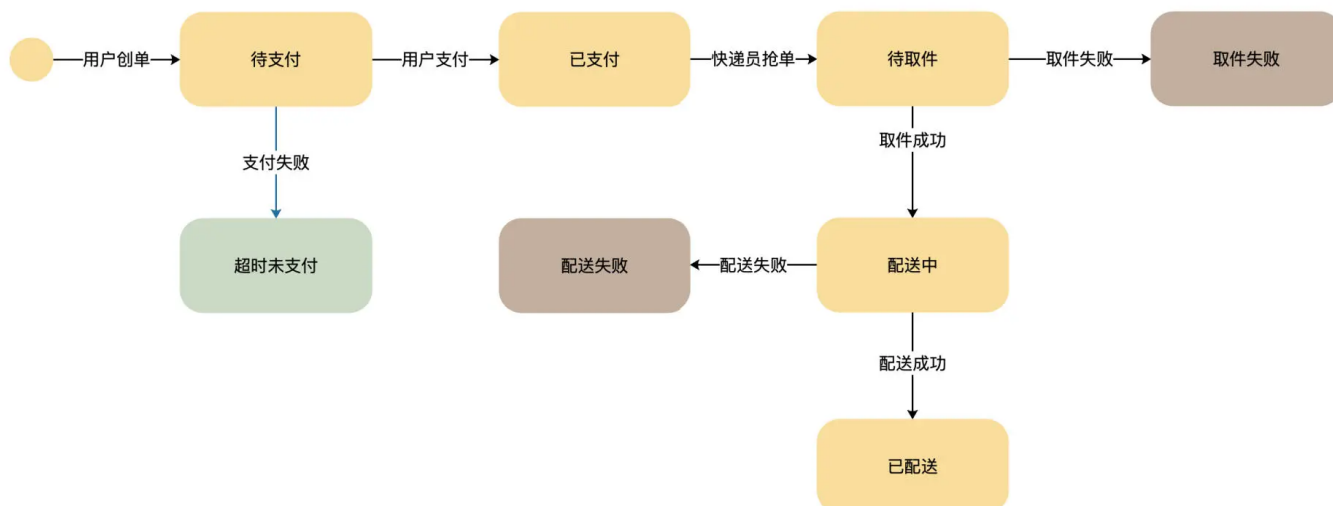
通达系统的核心使用者为需要在同城快速递送物品的用户和提供快送服务的快递员，系统用例图如下：



用户可以通过用户 App 创建快送订单，输入取件地址和收件地址，以及取件人、收件人联系方式后，系统预估订单金额，用户确认后，跳转到第三方支付页面等待用户支付。

快递员 App 实时上报自己当前位置。用户支付后，系统将订单发送给取件地点附近的所有快递员，快递员自主决定是否要抢单。系统会将用户订单信息推送给抢单成功的快递员 APP，快递员根据收件地点上门取件并完成配送。

订单管理是系统的核心功能，因此系统需要设计开发专门的订单管理模块，具体的订单状态模型图如下：



用户点击确认创建订单后，订单状态为“待支付”，用户完成支付后，订单状态为“已支付”。如果用户支付失败或者超时未支付，订单状态为“超时未支付”，超时未支付状态为订单最终状态之一，此状态订单不可以再修改操作。

“已支付”状态订单发送给附近快递员，如果有快递员抢单成功，订单状态为“待取件”。如果快递员上门联系不到发件人，快递员在 **APP** 拍照输入取件失败证据，订单状态改为“取件失败”。如果快递员上门取件成功，快递员在 **APP** 点击确认取件，订单状态改为“配送中”。

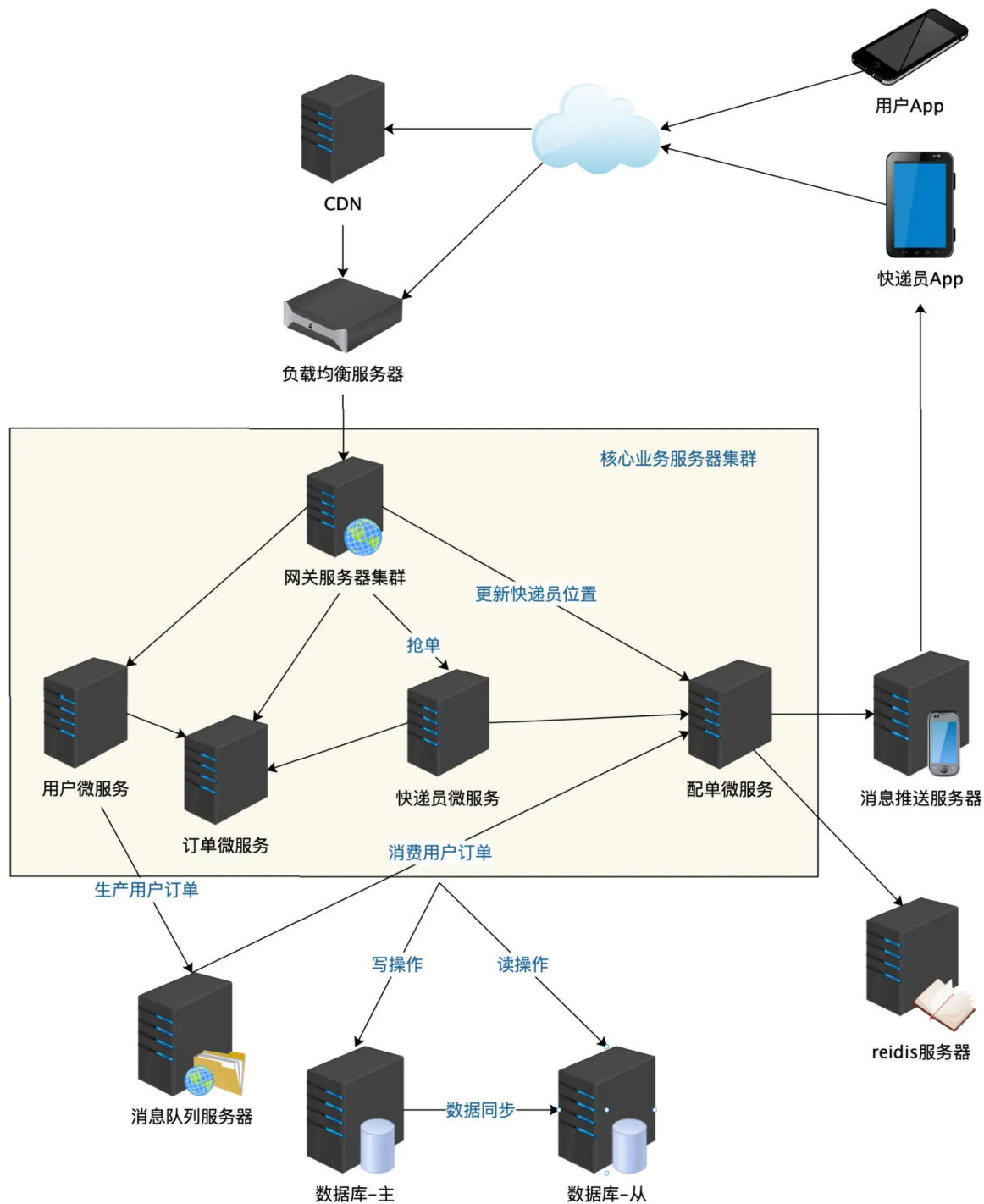
送达目的地后，如果联系不到收件人，快递员在 **APP** 拍照输入送件失败证据，订单状态改为“配送失败”，如果配送成功，订单状态改为“已配送”。

### **非功能需求：**

系统设计需要在线上后支持日订单量 **50** 万订单，一年后支持日订单量 **300** 万订单。

### **系统架构**

系统采用目前主流的微服务架构体系，整体架构图如下：



用户和快递员的请求通过负载均衡服务器进入网关服务器集群，网关服务器集群调用具体微服务完成请求处理。

用户微服务负责用户注册，个人资料修改，账户充值等业务逻辑，同时，用户所有操作都通过用户微服务集中处理，由用户微服务调用其他服务完成业务逻辑，包括下单、支付等。

快递员微服务负责为快递员提供服务，包括快递员注册、认证等，也包括快递员抢单、确认收件成功及失败、确认配送成功及失败等。

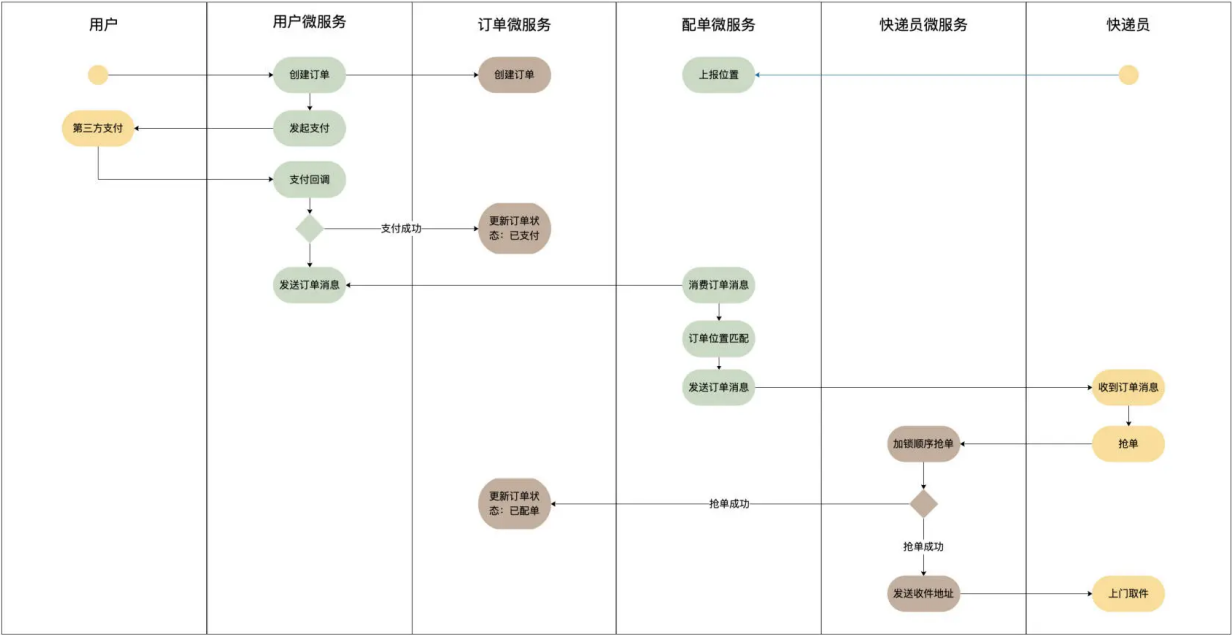
订单微服务负责订单生命周期管理，订单创建以及所有订单状态变更全部调用订单微服务来完成。所以用户微服务和快递员微服务需要依赖订单微服务。

配单微服务负责将用户订单推送给附近的所有快递员，即为订单匹配快递员。一方面，配单微服务需要记录所有快递员的当前位置，快递员位置信息通过快递员 APP 每 3 秒定时上传到网关服务器，网关服务器直接调用配单微服务，配单微服务将当前快递员位置信息更新到 Redis 中。

另一方面，配单微服务需要得到用户最新成功支付的订单信息。用户微服务在用户成功支付后，将订单信息发送给消息队列服务器，配单服务器作为消息消费者获取订单信息。配单服务器匹配到订单取件位置 5 公里范围内所有快递员，通过消息推送服务器将订单信息推送给快递员，等待快递员抢单。

系统使用关系数据库记录用户、快递员信息，以及订单信息，关系数据库采用主从复制同步的方式进行双机部署，订单写操作通过主数据库完成，订单读操作通过从数据库完成。

下单抢单逻辑是系统的核心逻辑，具体处理活动图如下：



订单处理涉及到的角色泳道包括用户、用户微服务、订单微服务、配单微服务、快递员微服务、快递员。

用户调用用户微服务请求创建订单，用户微服务调用订单微服务完成创建订单。订单微服务返回订单创建成功后，用户微服务发起支付，用户 APP 端跳转到第三方支付，用户微服务等待第三方支付回调。

收到第三方支付回调结果后，如果支付成功，调用订单微服务修改订单状态为“已支付”。同时将订单信息通过消息队列异步发送给配单微服务。配单微服务实时接受快递员上报的位置信息，根据订单信息，匹配附近的快递员。

匹配到符合条件的快递员后，发送订单信息给快递员。快递员收到推送的订单信息后，决定是否要抢单。快递员点下抢单按钮后，请求发送给快递员微服务，快递员微服务可能会同时收到多个快递员的抢单请求，所以抢单请求需要加锁顺序操作，保证第一个到达系统的抢单请求能成功抢单。

由于快递员微服务是集群部署，所以需要使用外部锁来保证请求的顺序性，系统使用 Redis 完成锁操作。抢单成功的请求调用订单微服务，更新订单状态为“已配单”，并发送收件地址给快递员，快递员根据地址上门取件。

## 主要技术选型

1. 开发语言采用 Java 开发实现，APP 端支持安卓和 IOS 两种系统。
2. 负载均衡服务器采用 Nginx 部署。
3. 网关采用 SpringBoot 开发，系统上线时，订单量较小，网关服务器采用双机部署，未来根据系统负载压力和监控指标进行扩容。
4. 微服务框架采用 Dubbo，每个微服务至少部署 3 个实例，并根据监控指标动态扩容。
5. 消息队列采用 ActiveMQ 部署。
6. 数据库采用 MySQL 并配置主从复制，实现读写分离。
7. 缓存使用 Redis，Redis 记录快递员实时位置信息并实现抢单加锁操作。

## 关键技术落地实现

抢单的锁实现通过调用 Redis 的 CAS 命令来实现，具体过程为：

1. 订单支付完成后，配单微服务收到订单消息，立即在 Redis 中创建一个 < 订单 ID, -1 > 的键值对；




2. 快递员抢单的时候，调用 Redis 的 CAS 命令：CAS < 订单 ID> <-1> < 快递员 ID>，该命令比较订单 ID 的 value 是否为 -1，如果是 -1，就将 value 设置为当前快递员 ID；
3. 只有第一个快递员调用的时候 value 为 -1，才能设置成功，也就是能抢单成功，并且记录下来抢单成功的快递员 ID。其他抢单请求都返回 CAS 失败。

由于快递员位置在不断变更，并且高并发地发送给系统。而 Redis 中 GeoHash 命令通过跳表来存储位置，这种不断更新位置的高并发请求将会对 Redis 造成巨大的计算压力。因此系统并不采用 Redis 的 GeoHash 命令来进行距离计算，而是采用专栏 [第 9 讲](#) 交友软件同样的设计方案，即采用 Hash 表加 GeoHash 编码的方式来实现，Hash 表存储在 Redis 中。

分享给需要的人，Ta 订阅超级会员，你最高得 50 元

Ta 单独购买本课程，你将得 20 元

 生成海报并分享

 赞 1  提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

[上一篇](#) 22 | 大数据平台设计：如何用数据为用户创造价值？

[下一篇](#) 结束语 | 一个架构师的一天



# 物联网遇见 RISC-V 芯片

加更 5 讲，解锁不一样的智能家居

郭朝斌

前小米生态链企业  
秒秒测 IoT 事业部总监  
前诺基亚高级工程师



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

## 精选留言 (5)

 写留言



無名小卒

2022-04-13

老师，您在系统架构里说到“配单微服务需要记录所有快递员的当前位置”，从功能职责边界角度来看的话，我觉得快递员位置信息还是放到快递员微服务会合适些。

此外，除了快递员从取件到配送完成的过程需要获取定位信息，其实从抢单到取件的过程也可以获取定位，以供用户了解快递员是否到达取件地取件，提升用户体验。

作者回复：考虑很周到，这样分析，也许我们应该拆分一个独立的快递员位置微服务，供其他微服务使用。



无名 

2022-04-13

老师，期间的图和文档能否分享一下呢



Steven

转眼间课程就结束了，感谢老师的讲解和指导，受益匪浅。

尤其这最后一课：“因为期中测试的评论区只有三个作业链接，就以为稳了，结果因为字数不够被刷了。随便拷一段文字过来也达到要求了”，所以做事不能有侥幸心理，不能大意。哈哈~

师傅领进门 修行在个人。各位江湖再见。

作者回复: 感谢你的坚持和参与，希望这个课程能给你带来收获和成长。

共 2 条评论 >



**Steven**

2022-04-11

期中测试出来，陪跑了一把。陪跑的某一方面原因是我4月3日当天赶出来，字数没到 2000 吗？

提交方式是“把文档链接提交在评论区”，奖励要求是获赞最多且符合作业要求。可是评论区现在一共只有三个作业链接，其它的作业在哪了？点赞数是多少呢？

本着公平公正公开的原则，所以我还是有必要问个明白的，是吧，小编。 :)

回到学习上来，因为参考文档没写太细，以下几个问题，想听听老师的看法：

- 1, **CDN**，我认为是不需要的，没有什么静态资源需要放到**CDN**。
- 2, 消息队列，我认为在这个量级下是没有必要采用消息中间队的，可以考虑用 **Redis** 这个轻量级的方案。
- 3, **MySQL** 数据库，我觉得主备即可，并不需要主从。
- 4, 网关考虑了双机部署，那负载均衡服务器 **Nginx** 也是需要双机高可用的。
- 5, **Redis** 以什么方式部署？哨兵足够了吧。

另外，针对同城快递这个场景：

- 1, 异地多活到什么量级需要考虑？
- 2, 订单数据的冷热数据时间分界线设置为多少是合理的？

作者回复: 谢谢参与~~ 优秀作业是我选的, 没有选中的核心原因是字数, 设计文档需要给相关者阅读, 字数少本身就无法传递出足够的设计意图, 所以我们作业是对字数有要求的。

1 快送用户发起订单的时候, 需要给快送商品拍照, 快递员可以看到商品照片, 决定是否接单, CDN用来缓存商品照片。

2 可以用redis

3 MySQL主备和主从技术上是同一回事

4 是的

5 可以

1 异地多活主要考虑是成本, 只要公司觉得多活带来的收益能够覆盖成本, 多大量级都可以搞多活

2 曾经在实践中设为1个月

共 2 条评论 >



peter

2022-04-11

请教老师几个问题啊:

**Q1: 网关为什么不用SpringCloud Gateway?**

文中说“网关采用 SpringBoot 开发”, 为什么要自己开发。SpringCloud有现成的网关“SpringCloud Gateway”。为什么不用现成的?

**Q2: Dubbo比http更好吗?**

文中说“微服务框架采用 Dubbo”。一般的SpringCloud培训, 服务间调用通常都用feign, feign实际上是http调用。选Dubbo, 是因为Dubbo比http更好吗?

**Q3: CDN使用时需要业务逻辑吗?**

CDN怎么使用的? 比如一个图片放在CDN上, APP访问此图片时, APP上的图片URL是CDN的URL, 而不是后端服务器上的URL。是这样使用的吗?

如果是这样用的, 假如CDN上没有该图片, 怎么转到后盾服务器去获取? 这个判断逻辑是谁实现的? 是CDN自己实现的吗? 还是网站开发人员需要在CDN上写业务逻辑? (或者只是在CDN上做个配置? 没有用过CDN, 瞎猜的)

**Q4: 推送服务是怎么实现的?**

推送服务是通过长连接来实现的吗? (类似于网约车)。是否可以用websocket实现?

作者回复: 1 spring boot 包含 spring gateway

2 架构师喜欢Dubbo, 团队有Dubbo经验, 核心工程师熟悉Dubbo

3 同一个URL, URL请求要经过CDN, 不管CDN有没有这张图片。有专门CDN开发的人实现CDN的逻辑

4 这里是安卓或IOS系统消息推送, 也可以用长连接。可以用websocket



