

24 | 如何在线上环境里兼容多种RPC协议?

2020-04-15 何小锋

RPC实战与核心原理

[进入课程 >](#)



讲述：张浩

时长 08:37 大小 7.91M



你好，我是何小锋。上一讲我们学习了如何在没有接口的情况下完成 RPC 调用，其关键在于你要理解接口定义在 RPC 里面的作用。除了我们前面说的，动态代理生成的过程中需要用到接口定义，剩余的其它过程中接口的定义只是被当作元数据来使用，而动态代理在 RPC 中并不是一个必须的环节，所以在没有接口定义的情况下我们同样也是可以完成 RPC 调用的。

回顾完上一讲的重点，咱们就言归正传，切入今天的主题，一起来看看如何在线上环境里兼容多种 RPC 协议。



看到这个问题后，可能你的第一反应就是，在真实环境中为什么会存在多个协议呢？我们说过，RPC 是能够帮助我们屏蔽网络编程细节，实现调用远程方法就跟调用本地一样的体

验。大白话说就是，RPC 是能够帮助我们在开发过程中完成应用之间的通信，而又不需要我们关心具体通信细节的工具。

为什么要支持多协议？

既然应用之间的通信都是通过 RPC 来完成的，而能够完成 RPC 通信的工具有很多，比如像 Web Service、Hessian、gRPC 等都可以用来充当 RPC 使用。这些不同的 RPC 框架都是随着互联网技术的发展而慢慢涌现出来的，而这些 RPC 框架可能在不同时期会被我们引入到不同的项目中解决当时应用之间的通信问题，这样就导致我们线上的生成环境中存在各种各样的 RPC 框架。

很显然，这种混乱使用 RPC 框架的方式肯定不利于公司技术栈的管理，最明显的一个特点就是维护 RPC 框架的成本越来越高，因为每种 RPC 框架都需要有专人去负责升级维护。

为了解决早期遗留的一些技术负债，我们通常会去选择更高级的、更好用的工具来解决，治理 RPC 框架混乱的问题也是一样。为了解决同时维护多个 RPC 框架的困难，我们肯定希望能够用统一用一种 RPC 框架来替代线上所有的 RPC 框架，这样不仅能降低我们的维护成本，而且还可以让我们在一种 RPC 上面去精进。

既然目标明确后，我们该如何实施呢？

可能你会说这很简单啊，我们只要把所有的应用都改造成新 RPC 的使用方式，然后同时上线所有改造后的应用就可以了。如果在团队比较小的情况下，这种断崖式的更新可能确实是最快的方法，但如果是在团队比较大的情况下，要想做到同时上线所有改造后的应用，暂且不讨论这种方式是否存在风险，光从多个团队同一时间上线所有应用来看，这也几乎是一件不可能做到的事儿。

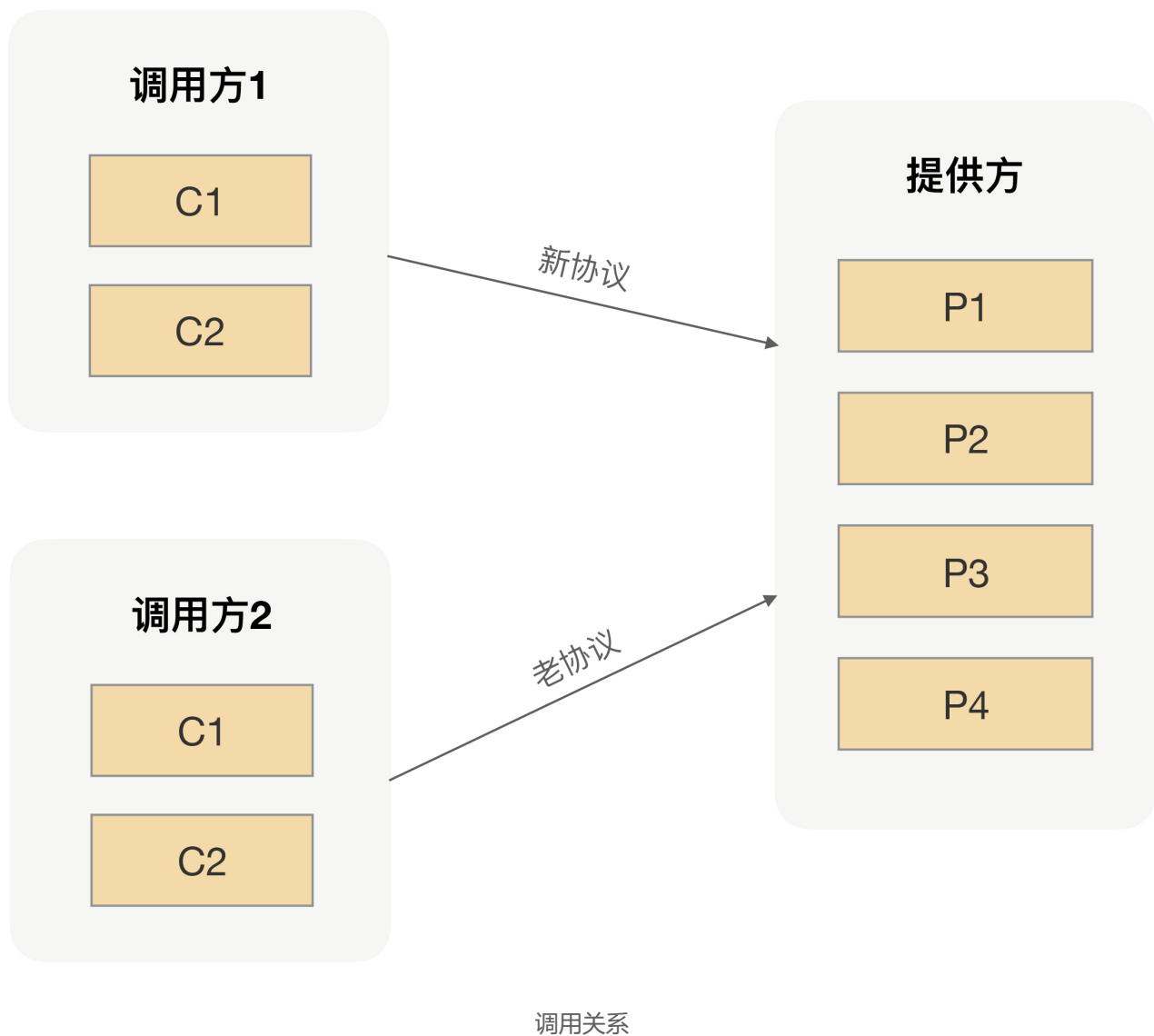
那对于多人团队来说，有什么办法可以让其把多个 RPC 框架统一到一个工具上呢？我们先看下多人团队在升级过程中所要面临的困难，人数多就意味着要维护的应用会比较多，应用多了之后线上应用之间的调用关系就会相对比较复杂。那这时候如果单纯地把任意一个应用目前使用的 RPC 框架换成新的 RPC 框架的话，就需要让所有调用这个应用的调用方去改成新的调用方式。

通过这种自下而上的滚动升级方式，最终是可以让所有的应用都切换到统一的 RPC 框架上，但是这种升级方式存在一定的局限性，首先要求我们能够清楚地梳理出各个应用之间的调用关系，只有这样，我们才能按部就班地把所有应用都升级到新的 RPC 框架上；其次要求应用之间的关系不能存在互相调用的情况，最好的情况就是应用之间的调用关系像一颗树，有一定的层次关系。但实际上我们应用的调用关系可能已经变成了网状结构，这时候想再按照这种方式去推进升级的话，就可能寸步难行了。

为了解决上面升级过程中遇到的问题，你可能还会想到另外一个方案，那就是在应用升级的过程中，先不移除原有的 RPC 框架，但同时接入新的 RPC 框架，让两种 RPC 同时提供服务，然后等所有的应用都接入完新的 RPC 以后，再让所有的应用逐步接入到新的 RPC 上。这样既解决了上面存在的问题，同时也可以让所有的应用都能有序地升级到统一的 RPC 框架上。

在保持原有 RPC 使用方式不变的情况下，同时引入新的 RPC 框架的思路，是可以让所有的应用最终都能升级到我们想要升级的 RPC 上，但对于开发人员来说，这样切换成本还是有点儿高，整个过程最少需要两次上线才能彻底地把应用里面的旧 RPC 都切换成新 RPC。

那有没有更好的方式可以让应用上线一次就可以完成新老 RPC 的切换呢？关键就在于要让新的 RPC 能同时支持多种 RPC 调用，当一个调用方切换到新的 RPC 之后，调用方和服务提供方之间就可以用新的协议完成调用；当调用方还是用老的 RPC 进行调用的话，调用方和服务提供方之间就继续沿用老的协议完成调用。对于服务提供方来说，所要处理的请求关系如下图所示：



怎么优雅处理多协议？

要让新的 RPC 同时支持多种 RPC 调用，关键就在于要让新的 RPC 能够原地支持多种协议的请求。怎么才能做到？在 [\[第 02 讲\]](#) 我们说过，协议的作用就是用于分割二进制数据流。每种协议约定的数据包格式是不一样的，而且每种协议开头都有一个协议编码，我们一般叫做 magic number。

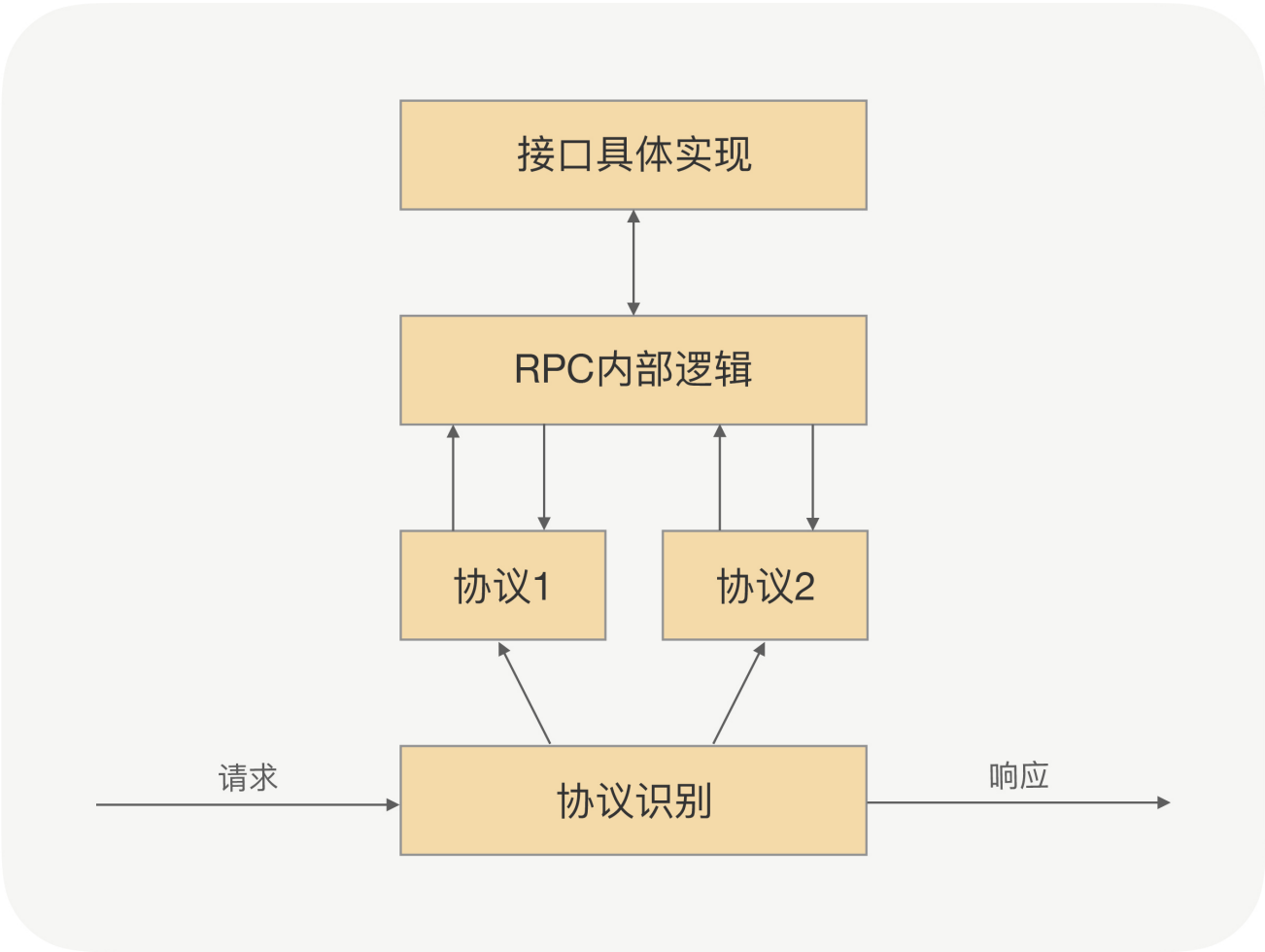
当 RPC 收到了数据包后，我们可以先解析出 magic number 来。获取到 magic number 后，我们就很容易地找到对应协议的数据格式，然后用对应协议的数据格式去解析收到的二进制数据包。

协议解析过程就是把一连串的二进制数据变成一个 RPC 内部对象，但这个对象一般是跟协议相关的，所以为了能让 RPC 内部处理起来更加方便，我们一般都会把这个协议相关的对象转成一个跟协议无关的 RPC 对象。这是因为在 RPC 流程中，当服务提供方收到反序列

化后的请求的时候，我们需要根据当前请求的参数找到对应接口的实现类去完成真正的方法调用。如果这个请求参数是跟协议相关的话，那后续 RPC 的整个处理逻辑就会变得很复杂。

当完成了真正的方法调用以后，RPC 返回的也是一个跟协议无关的通用对象，所以在真正往调用方写回数据的时候，我们同样需要完成一个对象转换的逻辑，只不过这时候是把通用对象转成协议相关的对象。

在收发数据包的时候，我们通过两次转换实现 RPC 内部的处理逻辑跟协议无关，同时保证调用方收到的数据格式跟调用请求过来的数据格式是一样的。整个流程如下图所示：



多协议处理流程

总结

在我们日常开发的过程中，最难的环节不是从 0 到 1 完成一个新应用的开发，而是把一个老应用通过架构升级完成从 70 分到 80 分的跳跃。因为在老应用升级的过程中，我们不仅

需要考虑既有的功能逻辑，也需要考虑切换到新架构上的成本，这就要求我们在设计新架构的时候要考虑如何让老应用能够平滑地升级，就像在 RPC 里面支持多协议一样。

在 RPC 里面支持多协议，不仅能让我们更从容地推进应用 RPC 的升级，还能为未来在 RPC 里面扩展新协议奠定一个良好的基础。所以我们平时在设计应用架构的时候，不仅要考虑应用自身功能的完整性，还需要考虑应用的可运维性，以及是否能平滑升级等一些软性能力。

课后思考

在 RPC 里面支持多协议的时候，有一个关键点就是能够识别出不同的协议，并且根据不同的 magic number 找到不同协议的解析逻辑。如果线上协议存在很多种的话，就需要我们事先在 RPC 里面内置各种协议，但通过枚举的方式可能会遗漏，不知道针对这种问题你有什么好的办法吗？

欢迎留言和我分享你的答案，也欢迎你把文章分享给你的朋友，邀请他加入学习。我们下节课再见！

更多课程推荐

RPC 实战与核心原理

高效解决分布式系统的通信难题

何小锋

京东技术架构部首席架构师



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。

上一篇 23 | 如何在没有接口的情况下进行RPC调用?

下一篇 特别放送 | 谈谈我所经历过的RPC

精选留言 (5)

写留言



Darren

2020-04-15

magic number和协议可以缓存到map中，或者redis中，从配置中心读取，当配置中心修改后，热更新到对应的缓存中。

展开 ▾



5



come on

2020-04-23

配置外部化，rpc框架一般都支持协议的扩展，可以通过加载外部配置的方式去构建应用支持的协议



Desmond

2020-04-20

在反序列化后，且调用API前加一个过滤器，识别是什么协议，老协议按老逻辑走，新协议按新协议逻辑走，多个过滤器构成一个调用链

展开 ▾



zero

2020-04-16

rpc的调用是微服务间直连调用，请问协议转换这层逻辑放在哪里，换了新协议是指在当前rpc框架更换协议还是直接更换了rpc框架

展开 ▾

作者回复: 为了平滑升级，需要在新rpc里面去做兼容



那个谁

2020-04-15

我想到的一种思路是，上线带监控的rpc，把未包含的协议监控起来，逐步上线支持

作者回复: 如果发现没有支持的，是不是就已经造成业务受损



💬 2

