

## 23 | 如何在没有接口的情况下进行RPC调用？

2020-04-13 何小锋

RPC实战与核心原理

[进入课程 >](#)



讲述：张浩

时长 08:20 大小 7.64M



你好，我是何小锋。上一讲我们学习了 RPC 如何通过动态分组来实现秒级扩缩容，其关键点就是“动态”与“隔离”。今天我们来聊聊如何在没有接口的情况下进行 RPC 调用。

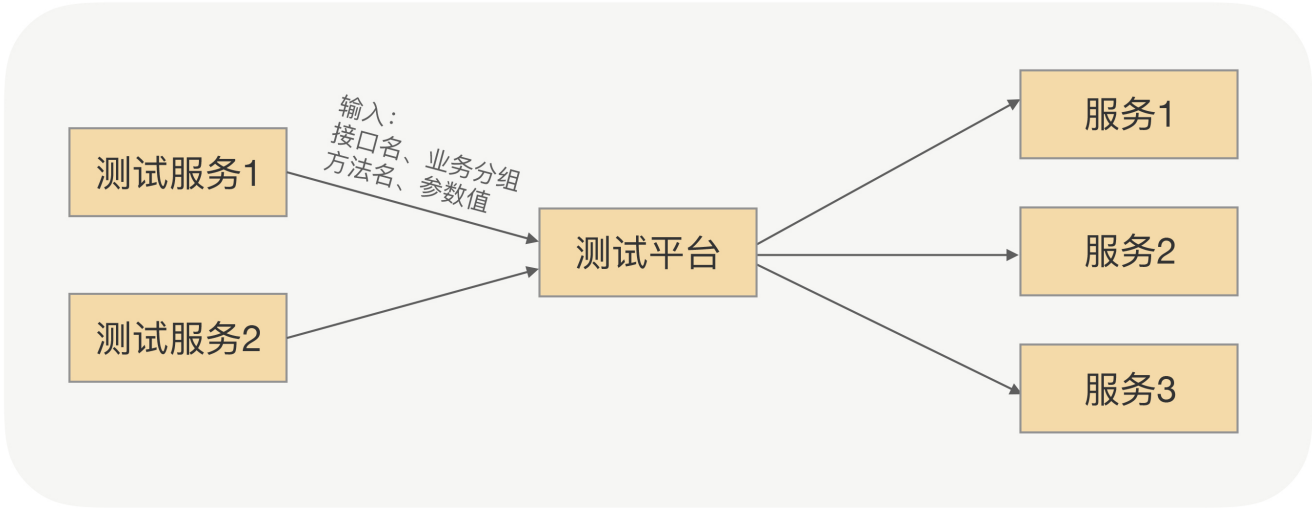
### 应用场景有哪些？

在 RPC 运营的过程中，让调用端在没有接口 API 的情况下发起 RPC 调用的需求，不只是一个业务方和我提过，这里我列举两个非常典型的场景例子。



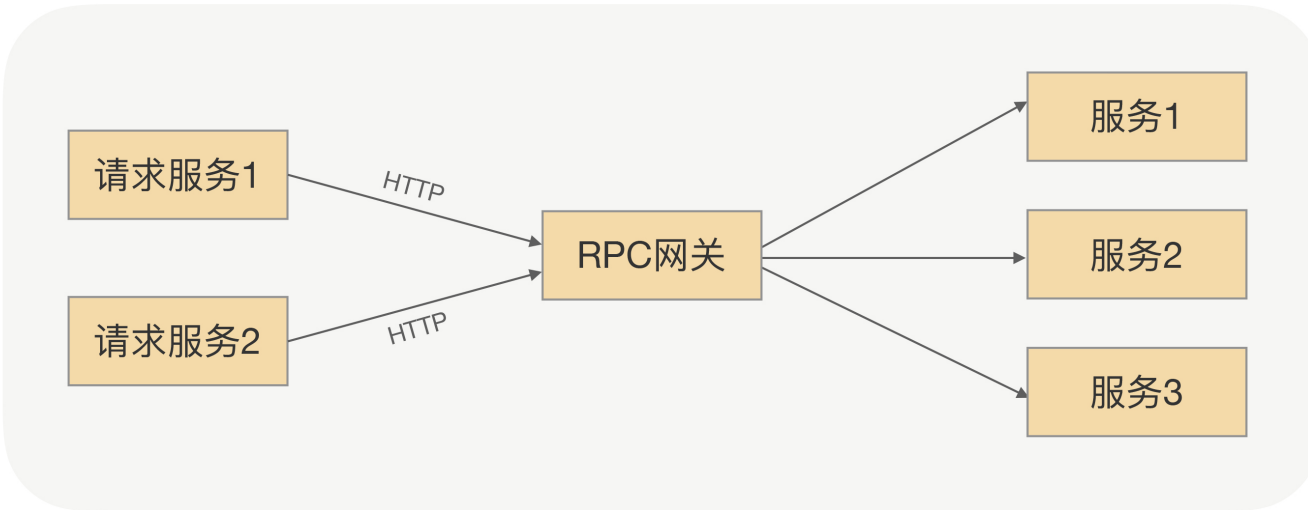
**场景一：**我们要搭建一个统一的测试平台，可以让各个业务方在测试平台中通过输入接口、分组名、方法名以及参数值，在线测试自己发布的 RPC 服务。这时我们就有一个问题要解决，我们搭建统一的测试平台实际上是作为各个 RPC 服务的调用端，而在 RPC 框架的使

用中，调用端是需要依赖服务提供方提供的接口 API 的，而统一测试平台不可能依赖所有服务提供方的接口 API。我们不能因为每有一个新的服务发布，就去修改平台的代码以及重新上线。这时我们就需要让调用端在没有服务提供方提供接口的情况下，仍然可以正常地发起 RPC 调用。



示意图

**场景二：**我们要搭建一个轻量级的服务网关，可以让各个业务方用 HTTP 的方式，通过服务网关调用其它服务。这时就有与场景一相同的问题，服务网关要作为所有 RPC 服务的调用端，是不能依赖所有服务提供方的接口 API 的，也需要调用端在没有服务提供方提供接口的情况下，仍然可以正常地发起 RPC 调用。



示意图

这两个场景都是我们会经常碰到的，而让调用端在没有服务提供方提供接口 API 的情况下仍然可以发起 RPC 调用的功能，在 RPC 框架中也是非常有价值的。

## 怎么做？

RPC 框架要实现这个功能，我们可以使用泛化调用。那什么是泛化调用呢？我们带着这个问题，先学习下如何在没有接口的情况下进行 RPC 调用。

我们先回想下我在基础篇讲过的内容，通过前面的学习我们了解到，在 RPC 调用的过程中，调用端向服务端发起请求，首先要通过动态代理，正如 [\[第 05 讲\]](#) 中我说过的，动态代理可以帮助我们屏蔽 RPC 处理流程，真正地让我们发起远程调用就像调用本地一样。

那么在 RPC 调用的过程中，既然调用端是通过动态代理向服务端发起远程调用的，那么在调用端的程序中就一定要依赖服务提供方提供的接口 API，因为调用端是通过这个接口 API 自动生成动态代理的。那如果没有接口 API 呢？我们该如何让调用端仍然能够发起 RPC 调用呢？

所谓的 RPC 调用，本质上就是调用端向服务端发送一条请求消息，服务端接收并处理，之后向调用端发送一条响应消息，调用端处理完响应消息之后，一次 RPC 调用就完成了。那是不是我们只要能够让调用端在没有服务提供方提供接口的情况下，仍然能够向服务端发送正确的请求消息，就能够解决这个问题了呢？

没错，只要调用端将服务端需要知道的信息，如接口名、业务分组名、方法名以及参数信息等封装成请求消息发送给服务端，服务端就能够解析并处理这条请求消息，这样问题就解决了。过程如下图所示：




示意图

现在我们已经清楚了解决问题的关键，但 RPC 的调用端向服务端发送消息是需要以动态代理作为入口的，我们现在得继续想办法让调用端发送我刚才讲过的那条请求消息。

我们可以定义一个统一的接口（GenericService），调用端在创建 GenericService 代理时指定真正需要调用的接口的接口名以及分组名，而 GenericService 接口的 \$invoke 方法的入参就是方法名以及参数信息。

这样我们传递给服务端所需要的所有信息，包括接口名、业务分组名、方法名以及参数信息等都可以通过调用 GenericService 代理的 \$invoke 方法来传递。具体的接口定义如下：

 复制代码

```
1 class GenericService {
2
3     Object $invoke(String methodName, String[] paramTypes, Object[] params);
4
5 }
```

这个通过统一的 GenericService 接口类生成的动态代理，来实现在没有接口的情况下进行 RPC 调用的功能，我们就称之为泛化调用。

通过泛化调用功能，我们可以解决在没有服务提供方提供接口 API 的情况下进行 RPC 调用，那么这个功能是否就完美了呢？

回顾下 [🔗\[第 17 讲\]](#) 我过的内容，RPC 框架可以通过异步的方式提升吞吐量，还有如何实现全异步的 RPC 框架，其关键点就是 RPC 框架对 CompletableFuture 的支持，那么我们的泛化调用是否也可以支持异步呢？

当然可以。我们可以给 GenericService 接口再添加一个异步方法 \$asyncInvoke，方法的返回值就是 CompletableFuture，GenericService 接口的具体定义如下：

 复制代码

```
1 class GenericService {
2
3     Object $invoke(String methodName, String[] paramTypes, Object[] params);
4
5     CompletableFuture<Object> $asyncInvoke(String methodName, String[] paramType:
6
7 }
```

学到这里相信你已经对泛化调用的功能有一定的了解了，那你有没有想过这样一个问题？在没有服务提供方提供接口 API 的情况下，我们可以用泛化调用的方式实现 RPC 调用，但是如果服务提供方没有提供接口 API，我们就没法得到入参以及返回值的 Class 类，也就不能对入参对象进行正常的序列化。这时我们会面临两个问题：

**问题 1：**调用端不能对入参对象进行正常的序列化，那调用端、服务端在接收到请求消息后，入参对象又该如何序列化与反序列化呢？

回想下 [\[第 07 讲\]](#)，在这一讲中我讲解了如何设计可扩展的 RPC 框架，我们通过插件体系来提高 RPC 框架的可扩展性，在 RPC 框架的整体架构中就包括了序列化插件，我们可以为泛化调用提供专属的序列化插件，通过这个插件，解决泛化调用中的序列化与反序列化问题。

**问题 2：**调用端的入参对象（params）与返回值应该是什么类型呢？

在服务提供方提供的接口 API 中，被调用的方法的入参类型是一个对象，那么使用泛化调用功能的调用端，可以使用 Map 类型的对象，之后通过泛化调用专属的序列化方式对这个 Map 对象进行序列化，服务端收到消息后，再通过泛化调用专属的序列化方式将其反序列成对象。

## 总结

今天我们主要讲解了如何在没有接口的情况下进行 RPC 调用，泛化调用的功能可以实现这一目的。

这个功能的实现原理，就是 RPC 框架提供统一的泛化调用接口（GenericService），调用端在创建 GenericService 代理时指定真正需要调用的接口的接口名以及分组名，通过调用 GenericService 代理的 \$invoke 方法将服务端所需要的所有信息，包括接口名、业务分组名、方法名以及参数信息等封装成请求消息，发送给服务端，实现在没有接口的情况下进行 RPC 调用的功能。

而通过泛化调用的方式发起调用，由于调用端没有服务端提供方提供的接口 API，不能正常地进行序列化与反序列化，我们可以为泛化调用提供专属的序列化插件，来解决实际问题。

## 课后思考

在讲解泛化调用时，我讲到服务端在收到调用端通过泛化调用的方式发送过来的请求时，会使用泛化调用专属的序列化插件实现对其进行反序列化，那么服务端是如何判定这个请求消息是通过泛化调用的方式发送过来的消息呢？

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。

© 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 | 动态分组：超高效实现秒级扩缩容

## 精选留言 (5)

 写留言



Darren

2020-04-13

方法名称需要特殊处理一下，参数不是很好，因为有的方法是没有参数的。



1



Reason





2020-04-13

有个问题请教下老师，希望可以得到解答：

文中说泛化调用用于统一测试平台时，可以不需要修改平台代码重新上线。不修改平台代码重新上线，怎么编写相应的泛化调用代码呢？



**Reason**

2020-04-13

能想到两种解决方法：

1. 通过泛化调用的接口名或者方法名，判断是否是泛化请求
2. 客户端发起调用时一定知道请求是泛化请求，因此可以在请求信息的附加字段中标识该请求为泛化请求

展开 ∨



**翌**

2020-04-13

这个泛化调用好抽象，对于区分泛化调用和其他调用，一个是泛化调用的函数名和第一参数方法名，可以在这两个上做文章来区分泛化方法和其他方法

展开 ∨



**高源**

2020-04-13

老师理论讲的很到位，深入理解还得去看代码，老师能否推荐一下学习rpc地方，理论听的明白，动手还差很多啊 😊

展开 ∨

