

# RPC

## 基本资料

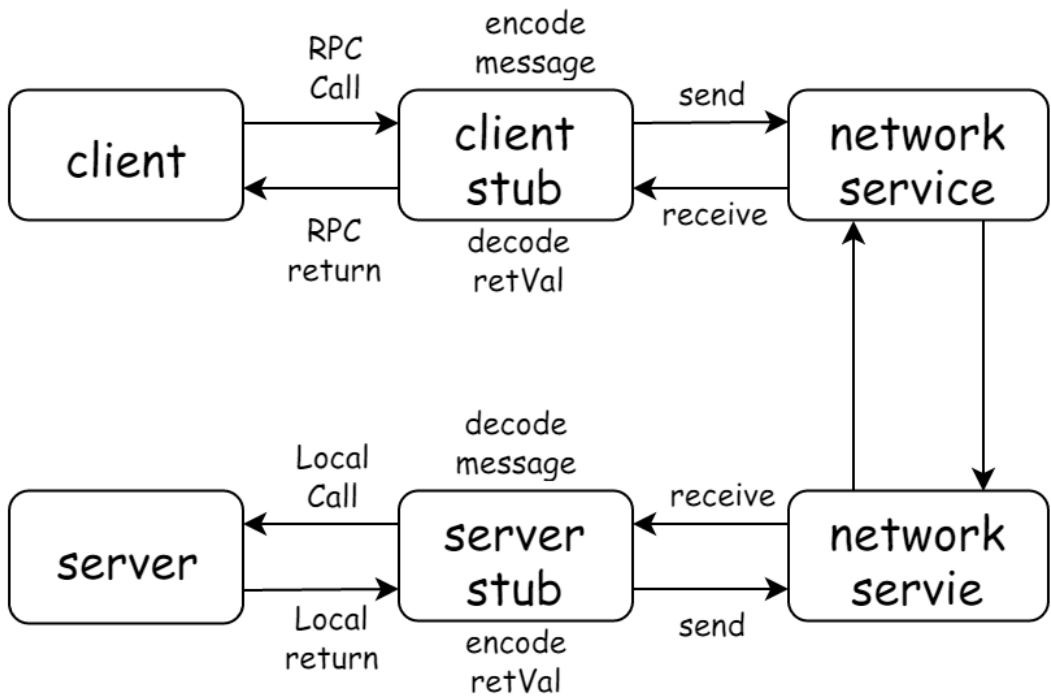
### 什么是rpc

RPC (Remote Procedure Call Protocol) ——远程过程调用协议。RPC 这个概念术语在上世纪 80 年代由 Bruce Jay Nelson 提出。

RPC定义：在分布式系统中，使用RPC调用远程服务和调用本地服务一样，即代码调用基本相同，RPC协议屏蔽了调用的底层细节。

远程调用和本地调用一样，这个是rpc的核心精髓。

### rpc调用结构



1. 调用方 (Client) 通过本地的 RPC 代理 (Proxy) 调用相应的接口
2. 本地代理将 RPC 的服务名，方法名和参数等信息转换成一个标准的 RPC Request 对象交给 RPC 框架
3. RPC 框架采用 RPC 协议 (RPC Protocol) 将 RPC Request 对象序列化二进制形式，然后通过 TCP 通道传递给服务提供方 (Server)
4. 服务端 (Server) 收到二进制数据后，将它反序列化成 RPC Request 对象
5. 服务端 (Server) 根据 RPC Request 中的信息找到本地对应的方法，传入参数执行，得到结果，并将结果封装成 RPC Response 交给 RPC 框架
6. RPC 框架通过 RPC 协议 (RPC Protocol) 将 RPC Response 对象序列化二进制形式，然后通过 TCP 通道传递给服务调用方 (Client)
7. 调用方 (Client) 收到二进制数据后，将它反序列化成 RPC Response 对象，并且将结果通过本地代理 (Proxy) 返回给业务代码

### rpc调用关键点

根据上面部分，要实现rpc整个调用流程，最主要有三个关键点：

#### 1.服务ID（去哪儿）

该ID唯一标示了某个服务，解决去哪里调用的问题，即寻址。就如C语言本地调用，该id就是函数指针，Java语言就是接口和对象。在分布式系统中，该ID可以由多种组合来唯一标示某个服务，如服务提供方的应用名和服务接口名组成服务的唯一ID。

这里就是服务寻址，在蚂蚁预支相关的概念有antvip、ldc、配置中心、zone分布

2.序列化和反序列化（携带什么、如何携带）

序列化和反序列化，解决服务消费者怎么把参数值传给远程服务提供者的问题。C语言在本地调用中，我们只需要把参数压到栈里，然后让函数自己去栈里读就行。但是在远程过程调用时，需要客户端把参数先转成一个字节流，传给服务端后，再把字节流转成自己能读取的格式。同理，从服务端返回的值也需要序列化反序列化的过程。

Chair是通过hession

3.网络传输 （怎么去）

在分布式系统中，各个服务是以网络来连接对方。网络传输即解决如何将服务ID和序列化后的数据传给对方。这里传输协议没有限制，只需完成数据携带功能即可。

TCP

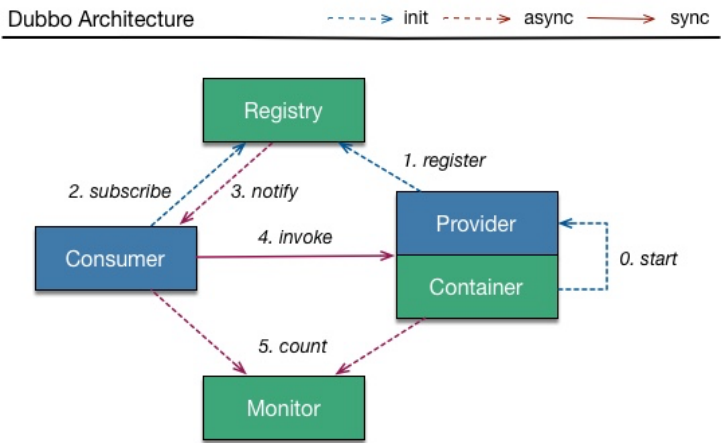
RPC框架

RPC框架职责

- 1.让调用方感觉就像调用本地函数一样调用远端函数
- 2.让服务提供方感觉就像实现一个本地函数一样来实现服务

rpc框架包含的角色

目前主流的框架都有如下图的几个角色划分



- 服务消费者：即服务订阅和调用方
- 服务提供者：即服务发布和注册方
- 注册中心：用来注册和订阅服务，监控注册者和订阅者，即实现服务发现的功能
- 监控中心：用来监控整个框架运行状况

通信框架、序列化/反序列化和协议栈的关系：协议描述了分布式服务框架的通信契约，序列化和分序列化框架用于协议消息对象和二进制数组之间的相互转换，通信框架在技术上承载协议，协议要落地，需要依赖通信框架提供的基础通信能力。

技术点

连接管理

系统keep-alive优势：

- TCP协议层面保活探测机制，系统内核完全替上层应用自动给做好了
- 内核层面计时器相比上层应用，更为高效
- 上层应用只需要处理数据收发、连接异常通知即可
- 数据包将更为紧凑

应用keep-alive优势（心跳）

关闭TCP的keepalive，完全使用业务层面心跳保活机制 完全应用掌管心跳，灵活和可控，比如每一个连接心跳周期的可根据需要减少或延长。心跳由客户端在无对tcp的读写操作后,定时发送.服务端收到则响应.如果客户端持续没有发送,达不到保活的目的,则服务端在90s之后,将关闭连接

业务心跳 + TCP keepalive一起使用，互相作为补充。

## 参考资料

RPC 相关日志：<https://lark.alipay.com/middleware/tracer/tracer-rpc> <<https://lark.alipay.com/middleware/tracer/tracer-rpc>>

RPC和RPC框架：<https://lark.alipay.com/mobilegw/wlnbvs/rpc-and-rpcframework> <<https://lark.alipay.com/mobilegw/wlnbvs/rpc-and-rpcframework>>

聊聊 Nodejs RPC（一）：<https://lark.alipay.com/xiaochen.gaoxc/gwtntkv/ufqt7k> <<https://lark.alipay.com/xiaochen.gaoxc/gwtntkv/ufqt7k>>

RPC的连接管理与心跳：<https://lark.alipay.com/middleware/sofa-rpc/heart-beat> <<https://lark.alipay.com/middleware/sofa-rpc/heart-beat>>

---