

RPC和RPC框架

RPC概述

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

Nelson 指出实现 RPC 的程序包括 5 个部分：

- 1. User：服务消费者
- 2. User-stub：request、调用参数，入参wrapper
- 3. RPCRuntime：网络传输wrapper
- 4. Server-stub：response、返回参数，出参wrapper
- 5. Server：服务提供者

这 5 个部分的关系如下图所示

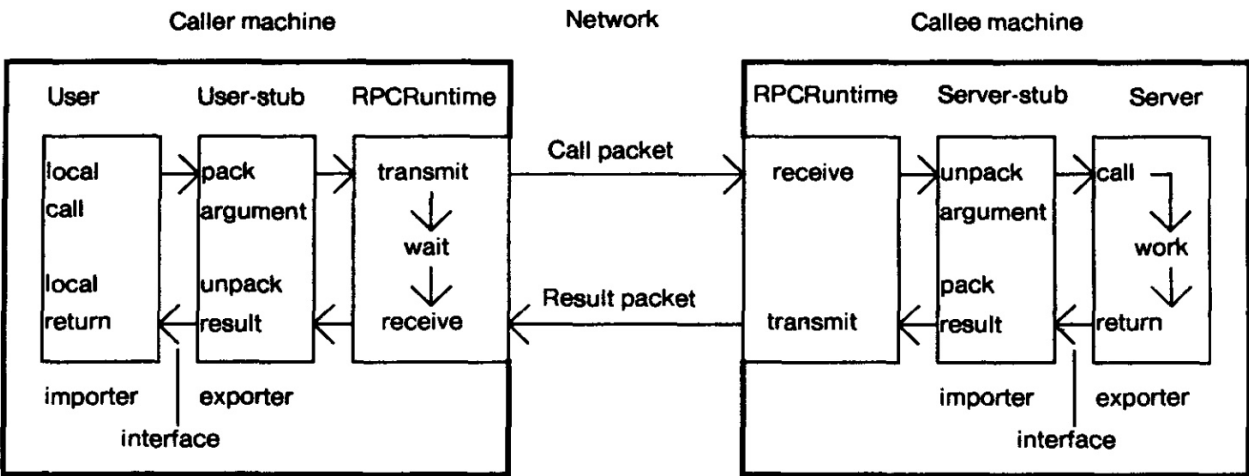


Fig. 1. The components of the system, and their interactions for a simple call.

这里 user 就是 client 端，当 user 想发起一个远程调用时，它实际是通过本地调用 user-stub。user-stub 负责将调用的接口、方法和参数通过约定的协议规范进行编码并通过本地的 RPCRuntime 实例传输到远端的实例。远端 RPCRuntime 实例收到请求后交给 server-stub 进行解码后发起本地端调用，调用结果再返回给 user 端。

rpc调用关键点

根据上面5个部分，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。

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。

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。

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

前面所讲的rpc原理，更倾向于运行时的概念。rpc框架包括了开发时期和运行时的概念。

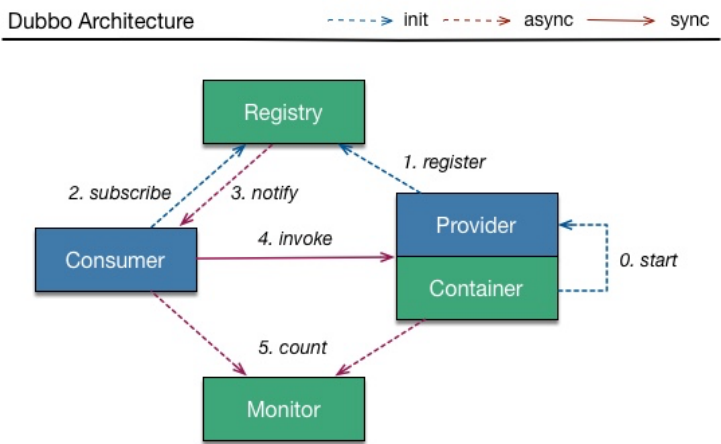
这里rpc框架分为两种类型，

- 1. 跨语言调用型，就是你提到的Thrift，Grpc之类的；
- 2. 服务治理型的，典型代表就是Ali的Dubbo，和新浪微博的Motan。
服务治理型多了服务的发现和服务的治理，偏向于soa的范畴。rpc调用只是它的一个部分。

以下讨论的都是服务治理型rpc框架。

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

rpc框架包含的主要功能

rpc框架最基本的几个功能：

协议支持

协议主要用来定义服务。协议的各个属性就是定义服务的各个属性。

序列化和反序列化

服务发布和消费

通过协议来定义服务的发布和消费

服务治理

- 流量控制：动态、静态流控制。
- 服务降级。
- 超时控制。
- 优先级调度。
- 流量迁移。
- 调用链跟踪和分析。
- 服务路由。
- SLA策略控制。

服务发现（注册中心）

服务监控

主流rpc框架对比

- 微博：Montan
- 阿里：HSF Dubbo Edas
- 百度：bRPC
- Google：gRPC
- Go语言实现的RPCX

这些rpc都大同小异，该有的功能都有，性能基本满足一般公司的各种流量场景。

	Dubbo	Montan	rpcx	gRPC	Thrift
开发语言	Java	Java	Go	跨语言	跨语言
分布式(服务治理)	√	√	√	×	×
多序列化框架支持	√	√ (当前支持Hessian2、Json,可扩展)	√	×	×
				(只支持protobuf)	(thrift格式)
多种注册中心	√	√	√	×	×
管理中心	√	√	√	×	×
跨编程语言	×	×	×	×	×
		(支持php client和C server)	http://blog.csdn.net/zixiao217		