

【RPC】慕课网RPC教程

原创

yzzheng_60125

已于 2022-06-10 16:32:49 修改

3108

收藏 7

版权

分类专栏：

Java

文章标签：

rpc

网络协议

网络



Java 专栏收录该内容

0 订阅

59 篇文章

订阅专栏

RPC

Remote Procedure Call, 远程过程调用

跨进程到跨物理机器

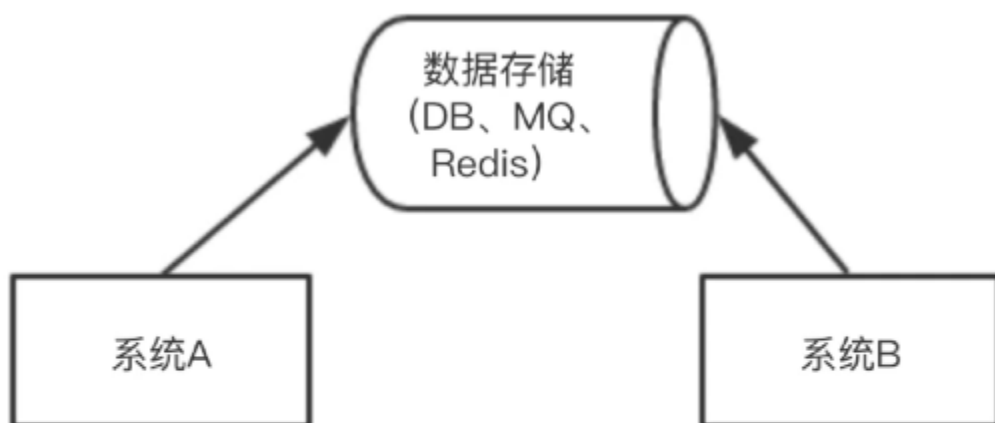
RPC是分布式系统常见的一种通信方法；

跨进程交互形式：RESTful、WebService、HTTP、基于DB做数据交换、基于MQ做数据交换、以及RPC

一、图解

1. 依赖中间件做数据交互

消息允许在数据存储中积压的
系统A和B异步执行的；



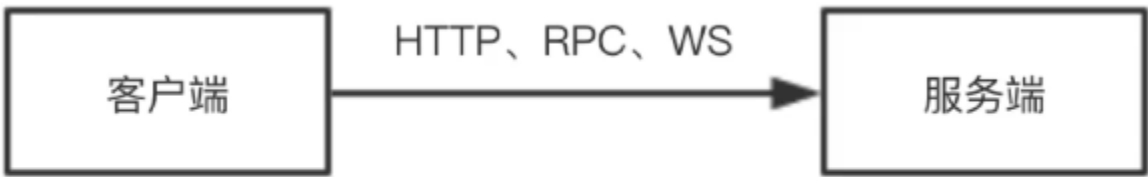
MySQL、RabbitMQ、Kafka、
Redis

CSDN @~知境~

2、直接交互

客户端会一直等着服务端返回的

客户端和服务端同步执行的；



RESTful、WebService、RPC、HTTP

RPC中

Server: Provider, 服务提供者

Client: Consumer、服务消费者

Stub: 存根、服务描述

在RPC中可以像调用本地方法, 调用远程方法

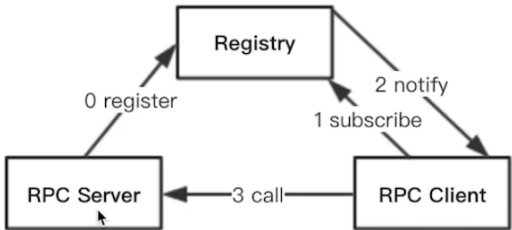
	gRPC	thrift	RMI	dubbo	HadoopRPC
开发语言	多语言	多语言	Java	Java	Java
序列化	protobuf	thrift格式	Java序列化	hession2	R/Writable
注册中心	✗	✗	JDK自带	Zk等	✗
跨语言	✓	✓	✗	✗	✗
服务定义	Protobuf文件	Thrift文件	Java接口	Java接口	Java接口
服务治理	✗	✗	✗	✓	✗

CSDN @~知境~

服务治理：RPC调用和服务管理（比如加上注册中心、服务监控、服务降级）

二、RPC核心原理

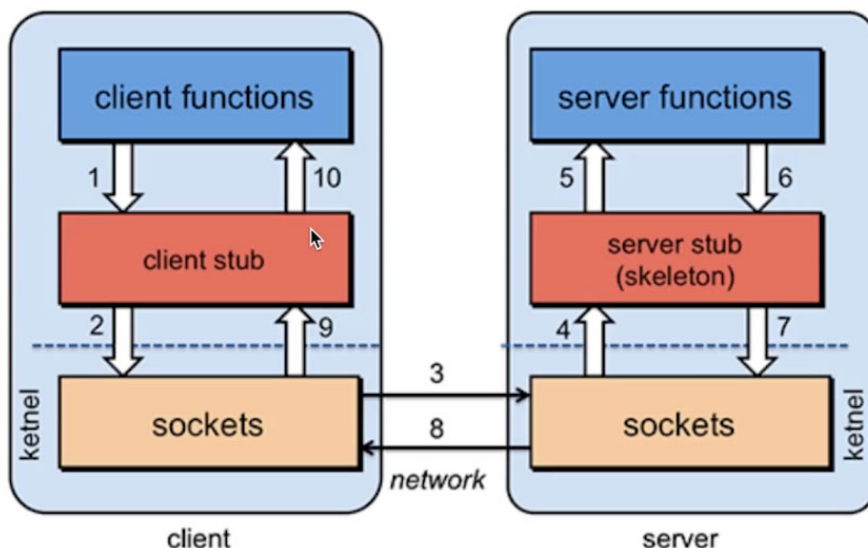
首先server会注册到注册中心去, client会订阅注册中心, 如果server地址改变会重新注册上去, 注册中心通知client端, 最后一个client可以调用rpc server。 call是rpc 调用的关键。



Server: Provider , 暴露服务 , 服务提供方
Client: Consumer , 服务消费 , 调用远程服务
Registry: 服务注册与发现

CSDN @~知境~

注册中心不一定是必须的；call是这个结构的关键部分



客户端
存根代理
服务端
网络传输
序列化

CSDN @~知境~

第2步，序列化，把对象转成网络能够传输的二进制数据

第4步，反序列化，把二进制反序列化成client的发送的对象，给server

client序列化一个对象C通过网络传输过来，这个对象C到达server端，它应该包括client调用server端的信息，明确调用哪个接口（stub），调用接口的实参。

通过反射调用这个方法，拿到结果，然后在序列化。

RPC中重要的部分，3和8的网络传输部分，RPC中肯定要有一个网络管理的模块，序列化，client端；

client中内部有stub对象，server中有对服务管理和服务反射调用的功能；

三、技术栈

基础：JavaCore、Maven、反射

Maven分成多模块管理

动态代理（生成client存根实际调用对象）

序列化（Java对象与二进制数据互换）fastjson，序列化：Java对象转位二进制数组，反序列化：二进制数组转为Java对象

网络通信（传输序列化后的数据）jetty、URLConnection

tcp长连接是RPC的主流，这里使用HTTP；

常用的RPC框架都是tcp协议的

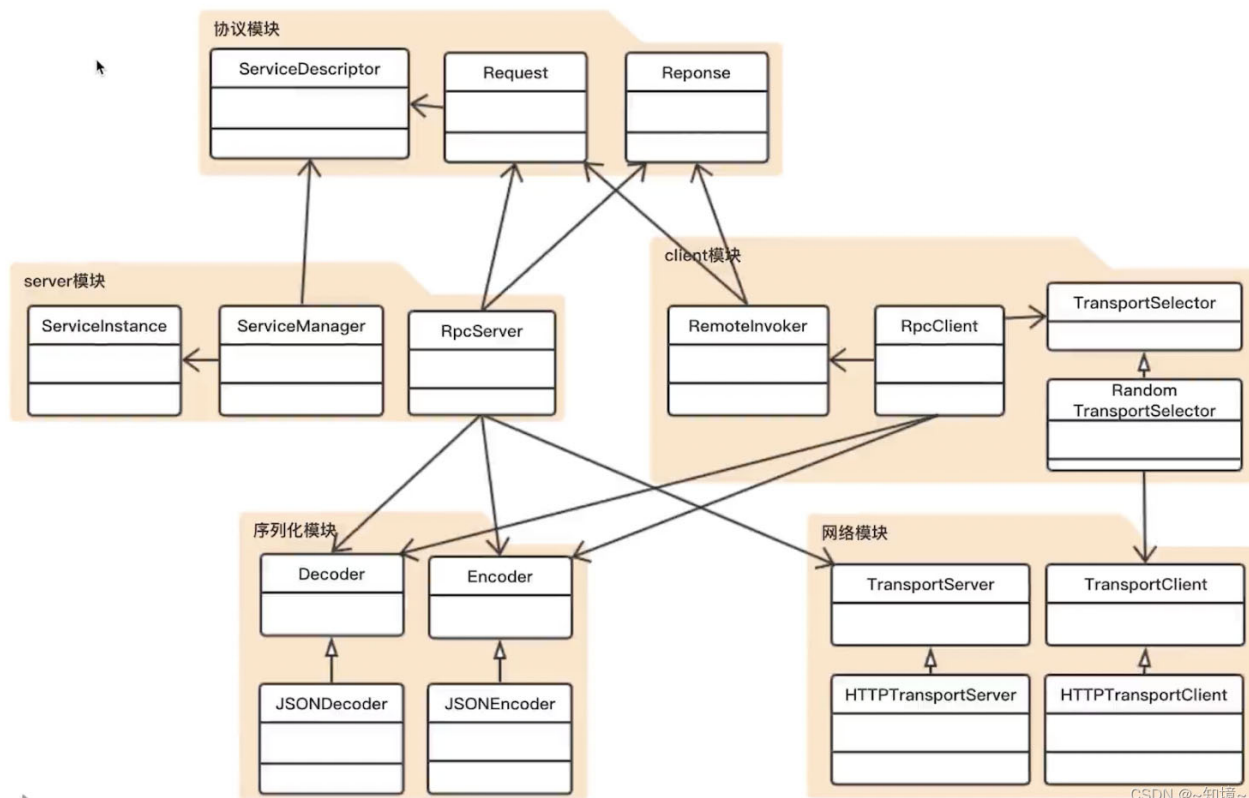
但是这里用HTTP协议，应用场景：比较简单，另一方面，假如已经有了1个http服务，但是又想开放rpc服务，同时不想去多开一个网络端口（涉及网安问题），走http

四、实战篇

步骤分析：

- step1 创建工程、制定协议、通用工具方法
- step2 实现序列化模块
- step3 实现网络模块
- step4 实现Server模块
- Step5 实现Client模块
- gk-rpc使用案列

4.1 类图



协议模块：描述server与client之间的通信协议，核心类：`Request`表示需要请求server的哪个服务，以及请求携带的参数；`Response`表示server相应client的返回信息（）、`ServiceDescriptor`：一个服务的描述信息

序列化模块：把对象进行序列化和反序列化,encoder把对象序列化二进制，decoder把二进制反序列化

网络通信模块：基于HTTP实现

Server模块：ServiceManager维护一些server需要暴露出去的服务，ServiceInstance暴露出去的服务的具体对象

client模块：RemoteInvoke和RpcServer进行交互，交互信息通过request和response进行封装的，TransportSelector（client和server建立连接的时候，可以建立多个连接，或者一个client连接不同的server）

4.2 动手实战

4.2.1 模块的定义

gk-rpc-client：用户端模块

gk-rpc-codec：序列化和反序列化模块

gk-rpc-common：通用模块

gk-rpc-protol：rpc协议模块

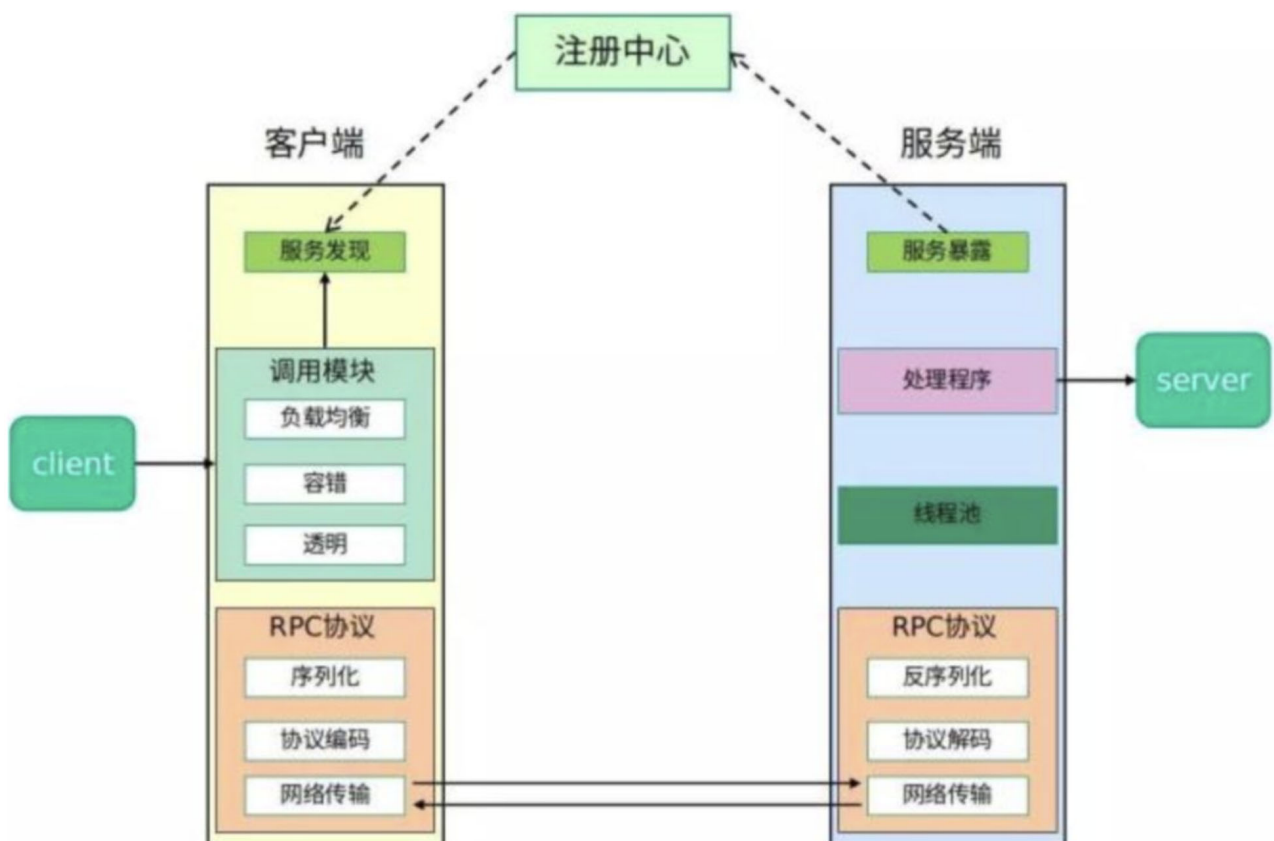
gk-rpc-server：服务端模块

gk-rpc-transport：网络传输

回顾总结

6月9日跟着 smgeek 老师学完了RPC的项目，已经debug测试完成，想着继续优化这个项目，所以近期在重新补充

看看RPC结构的全貌：



关于RPC协议这一部分的网络传输，可以使用不用的网络协议来实现

在smgeek中的rpc协议这部分，使用的是jetty，它是基于http协议实现的
在网络消息传输中可以基于TCP、UDP、http来实现，各自都有各自的特点：

基于 TCP 实现的 RPC 调用，能够灵活对协议字段进行定制，减少网络开销提高性能，实现更大的吞吐量和并发数，但要关注底层细节，在进行数据解析时更加复杂一些。

基于 HTTP 实现的 RPC 可以使用 JSON 和 XML 格式的请求或响应数据，解析工具很成熟，在其上进行二次开发会非常便捷和简单。但是 HTTP 是上层协议，所占用的字节数会比使用 TCP 协议传输所占用的字节数更高。

文章知识点与官方知识档案匹配，可进一步学习相关知识

网络技能树 跨区域网络的通信学习 网络层的作用 33015 人正在系统学习中

“相关推荐”对你有帮助么？

 非常没帮助  没帮助  一般  有帮助  非常有帮助