记得录屏

# 课程回顾

1. 分布式架构的定义以及分布式架构的演进。
2. 分布式架构和集群的区别
3. TCP/UDP、全双工、半双工、3次握手协议、4次挥手协议
4. FIN标识的报文给到server端
5. server端接收到FIN报文以后，表示Client端没有数据要发给Server端了
6. server端发送ACK给到Client端，表示Server端的数据已经发完了。准备关闭链接
7. client端收到ACK报文以后，知道可以关闭连接了，发送ACK请求到Server端，自己进入TIME-WAIT
8. Server端接收到ACK以后，表示可以断开连接了
9. Client端等待一定时间后，没有收到回复，表示Client可以关闭连接
10. TCP的非阻塞IO
11. 序列化
12. SerialVersionUID
13. 静态变量序列化问题、Transient关键字、父子类的序列化问题
14. kryo、FST、JSON、XML、protobuf、Hessian、Avro、Thrift
15. http和https协议、RESTful规范
16. 客户端发起一个https请求
    1. 客户端支持的加密方式
    2. 客户端生成的随机数（第一个随机数）
17. 服务端收到请求后，拿到随机数，返回
    1. 证书（颁发机构（CA）、证书内容本身的数字签名（使用第三方机构的私钥加密）、证书持有者的公钥、证书签名用到的hash算法）
    2. 生成一个随机数，返回给客户端（第二个随机数）
18. 客户端拿到证书以后做验证
    1. 根据颁发机构找到本地的跟证书
    2. 根据CA得到根证书的公钥，通过公钥对数字签名解密，得到证书的内容摘要 A
    3. 用证书提供的算法对证书内容进行摘要，得到摘要 B
    4. 通过A和B的对比，也就是验证数字签名
19. 验证通过以后，生成一个随机数（第三个随机数），通过证书内的公钥对这个随机数加密，发送给服务器端
20. （随机数1+2+3）通过对称加密得到一个密钥。（会话密钥）
21. 通过会话密钥对内容进行对称加密传输

# 分布式通信框架-RMI讲解

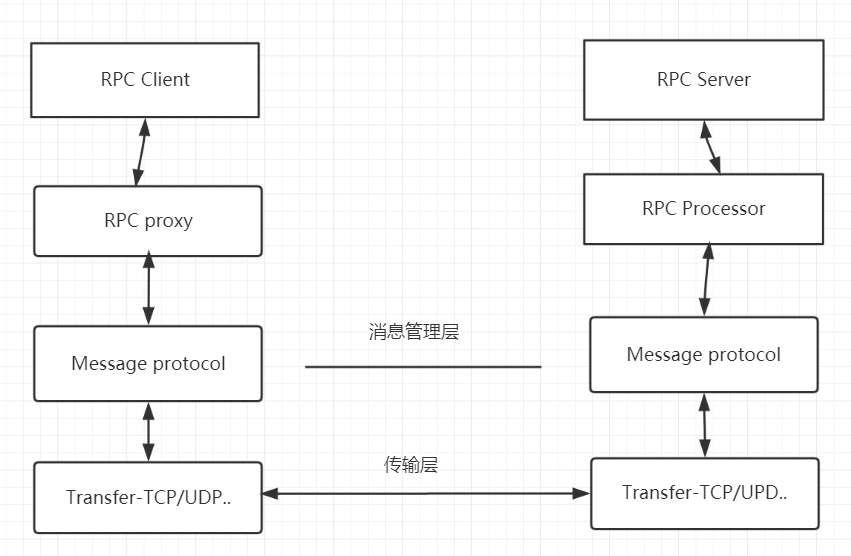
## 什么是RPC

Remote procedure call protocal

RPC协议其实是一个规范。Dubbo、Thrif、RMI、Webservice、Hessain

网络协议和网络IO对于调用端和服务端来说是透明；

## 一个RPC框架包含的要素



# RMI的概述

RMI(remote method invocation) , 可以认为是RPC的java版本

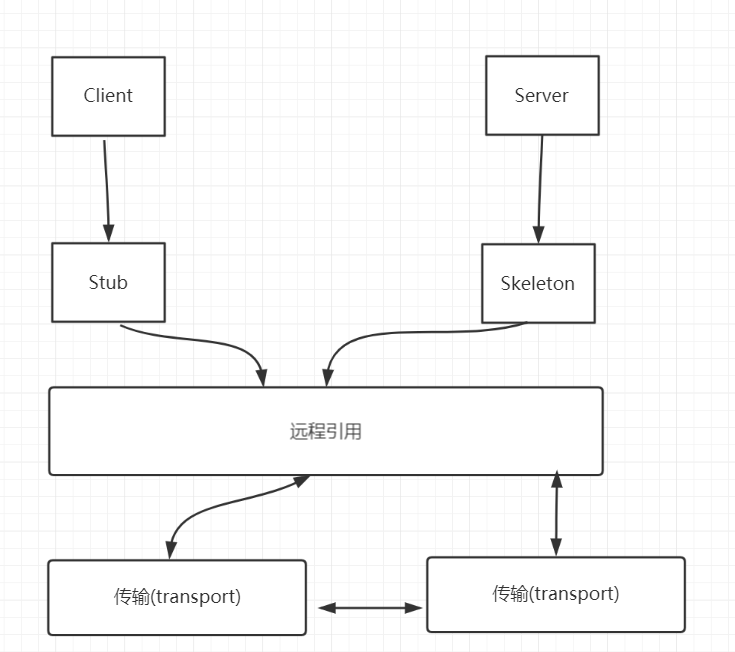
RMI使用的是JRMP（Java Remote Messageing Protocol）, JRMP是专门为java定制的通信协议，所以踏实纯java的分布式解决方案

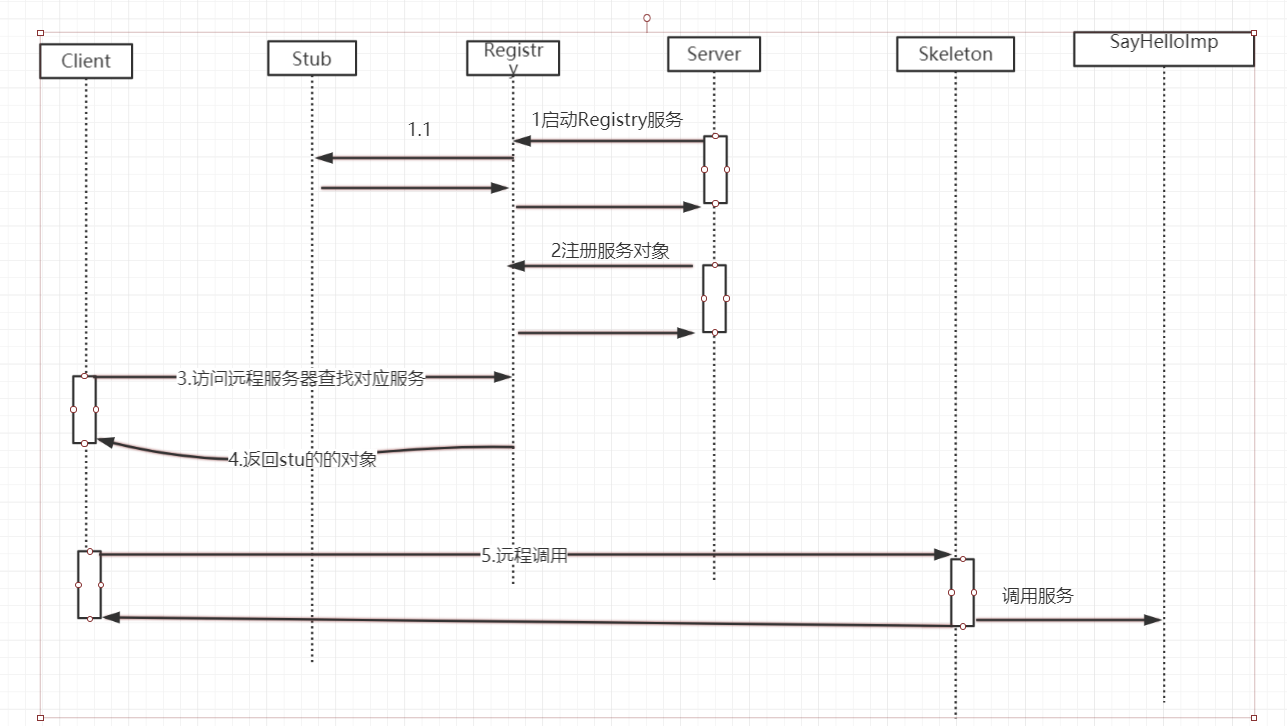
# 如何实现一个RMI程序

1. 创建远程接口， 并且继承java.rmi.Remote接口
2. 实现远程接口，并且继承：UnicastRemoteObject
3. 创建服务器程序： createRegistry方法注册远程对象
4. 创建客户端程序

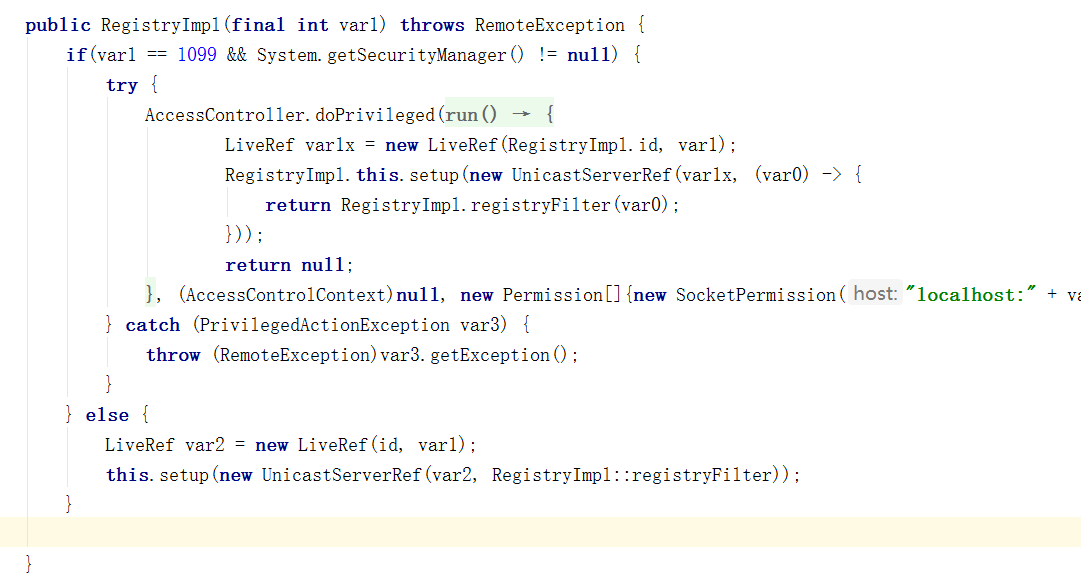
# 如果自己要去实现一个RMI

1. 编写服务器程序，暴露一个监听， 可以使用socket
2. 编写客户端程序，通过ip和端口连接到指定的服务器，并且将数据做封装（序列化）
3. 服务器端收到请求，先反序列化。再进行业务逻辑处理。把返回结果序列化返回





# 源码分析



BIO 、 NIO