**分布式:**

RPC可以提高系统稳定性，比如说，我们的订单服务程序更新出BUG，导致内存溢出，是这台服务器宕机了，但是它只会影响的整个系统的订单业务部分，对于用户注册登录等业务没有影响，同样对于系统的日志记录也没有影响。

**RPC:**

注册中心:服务端会把它的服务注册到注册中心中，包括服务名称、服务调用的ip地址、端口、协议、还有调用路径等等。

**RMI:**

RMI接口和实现类不灵活，RMI必须继承和实现Remote接口之类的。

**动态代理:**

动态代理，两个概念，一个是代理，一个是动态。

代理的本质就是代理模式，代理模式一定要有这三个要素：接口，提供服务的真实对象，代理对象。

三个要素之外还有重要的两个动作，

首先是真实对象以及代理的对象都必须同时继承一个指定的接口。

第二个，这个代理对象必须包含这个真实的对象。

Java实现：

第一个小伙伴,Proxy, 它就是一个调度器,这个是专门调度人.

第二个小伙伴就是Invocationhandler,这个是一个增强器，专门做增强的.

Invocationhandler的源码，这个代码就只有一个方法，我们把这个方法实现了就可以了。

代理模式里面一个重点，代理模式一定要把真实对象包进来。

为什么在RPC中用动态代理增强？-----增强了网络远程调用功能。

ObjectOutputStream和ObjectInputStream结合Socket通讯就是我们实现的网络增强部分的组成.