

rpc

RPC 即远程过程调用,它是分布式架构的核心,当两台服务器 A, B, 一个应用部署在 A 服务器上,想要调用 B 服务器上应用提供的函数/方法,由于不在一个内存空间,不能直接调用,此时使用 rpc 就可以像调用本地服务(方法)一样去调用服务器的服务(方法).

RPC 架构里包含如下 4 个组件:

- 1、客户端(Client): 服务调用方.
- 2、客户端存根(Client Stub): 存放服务端地址信息,将客户端的请求参数打包成网络消息,再通过网络发送给服务方.
- 3、服务端存根(Server Stub): 接受客户端发送过来的消息并解包,再调用本地服务.
- 4、服务端(Server): 真正的服务提供者.

具体实现步骤:

- 1、客户端以本地调用方式调用服务;
- 2、客户端存根接收到调用后负责将方法、参数等组装成能够进行网络传输的消息体;
- 3、客户端存根找到服务地址,并将消息通过网络发送到服务端;
- 4、服务端存根收到消息后进行解码;
- 5、服务端存根根据解码结果调用本地的服务;
- 6、本地服务执行处理逻辑;
- 7、本地服务将结果返回给服务端存根;
- 8、服务端存根将返回结果打包成消息,并将打包后的消息通过网络发送客户端;
- 9、客户端存根接收到消息,并进行解码;
- 10、客户端得到最终结果。

RPC 架构的目标就是把客户端存根将请求参数作为消息体发送给服务端,服务端那边将处理完成后的结果返回给客户端的这一过程封装起来,让用户像调用本地服务一样的调用远程服务.

REST

是一种软件架构风格,它将网络中的一切事物看成资源,通过不同的请求方式来对资源进行不同的操作并获取资源.

区别

从性能角度看,使用 REST 在网络传输时,需要携带的信息更多(因为 REST 是基于 HTTP 协议的,HTTP 本身提供了丰富的状态功能与扩展功能,但也正由于 Http 提供的功能过多导致性能低),从性能角度上讲,较为低效。而 RPC 传输上仅传输与业务内容相关的数据,传输数据更小,性能更高。

IO 多路复用作用

socketserver, 多个客户端连接,单线程下实现并发效果,就叫多路复用。

与多进程和多线程技术相比,I/O 多路复用技术的最大优势是系统开销小,系统不必创建进程/线程,也不必维护这些进程/线程,从而大大减小了系统的开销。

select、poll、epoll 模型的区别? (属于多路复用 IO 的模型)

都是 i/o 多路复用的机制,监视多个 socket 是否发生变化,本质上都是同步 i/o

select,poll 实现需要自己不断轮询所有监测对象,直到对象发生变化,在这个阶段中,可能要睡眠和唤醒多次交替,而 epoll 也需要调用 epoll_wait 不断轮询就绪链表,但是当对象发生变化时,会调用回调函数,将变化的对象放入就绪链接表中,并唤醒在 epoll_wait 中进入睡眠的进程。

虽然都会睡眠和唤醒,但是 select 和 poll 在被唤醒的时候要遍历整个监测对象集合,而 epoll 只要判断就绪链表是否为空即可,节省了大量 cpu 的时间。