软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, [doi: 10.13328/j.cnki.jos.000000]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

RPC 远程过程调用实验

周小多

(计算机软件新技术国家重点实验室(南京大学),江苏 南京 210023)
通讯作者: 周小多, E-mail: MF1733086@nju..edu. cn

摘 要: 本次实验是 RPC（远程过程调用）实验，实验目的是实现客户机与远程服务器的时间同步，实验有几点要求，一是必须要考虑在客户机读取到服务器时间后在传输过程中的时延问题，二是只有得到服务器授权的客户机才有资格进行远程过程调用。实验要求在能够跨平台环境下运行且当多个客户机同时向服务器读取时间,并且服务器端能并发执行。
关键词: RPC（远程过程调用）;网络时延;跨平台;并发
中图法分类号: TP311

RPC（Remote Procedure Call）Experiment

ZHOU Xiao-Duo

(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

Abstract: This is a experiment for RPC(Remote Process Call), the purpose of the experiment is to achieve the synchronization between client and server . The experiment has several requirements. First ,we must solve the communication delay between client and server . Second, only authorized clients could get time from server. The experiment requires that the server-side can execute concurrently when it can run across a platform environment and when multiple clients simultaneously read the time from the server.
Key words: RPC(Remote Process Call); Network Delay; Cross-Platform; Concurrent

1 背景简述

RPC（Remote Procedure Call Protocol）——远程过程调用协议，它是一种通过网络从远程计算机程序上请求服务，而不需要了解底层网络技术的协议[1]。

RPC 是一种 C/S 的编程模式，有点类似 C/S Sccket 编程模式，但要比他更高一层。当我们建立起 RPC 服务以后，客户端的调用参数通过 RPC 传输通道，可以使 UDP，也可以是 TCP，并根据传输前所提供的目的地址及 RPC 上层应用程序号转至相应的 RPC 应用服务器，此时的客户端处于等待状态，直至收到应答或超时信号。当服务器端获得请求消息，则会根据注册 RPC 时告诉 RPC 系统的例程入口地址，执行相应操作，并将结果返回客户端。一次 RPC 调用结束后，相应的线程发送信号，客户端才会继续运行[2]。

1.1 RPC结构

RPC 的程序包括 5 个部分：User、client -stub、RPCRuntime、Server-stub、Server（如图 1）

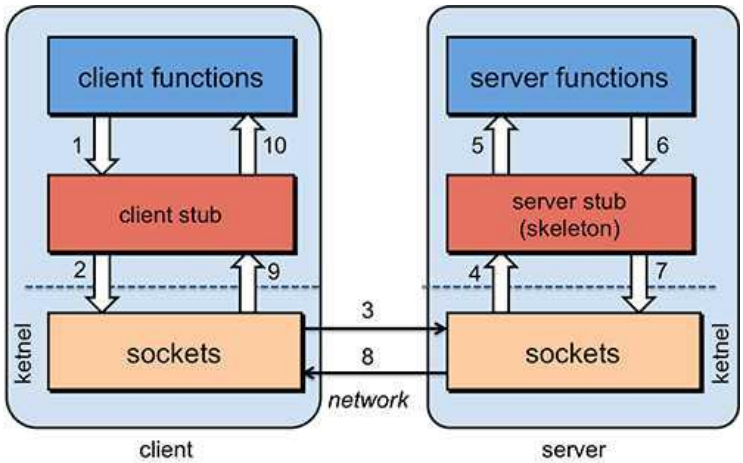


图 1 RPC 基本结构

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 client-stub。client -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 client 端。

远程过程调用主要包含如下步骤：

1、客户过程以正常的方式调用客户存根；

2、客户存根生成一个消息，然后调用本地操作系统；

3、客户端操作系统将消息发送给远程操作系统；

4、远程操作系统将消息交给服务器存根；

5、服务器存根调将参数提取出来，而后调用服务器；

6、服务器执行要求的操作，操作完成后将结果返回给服务器存根；

7、服务器存根将结果打包成一个消息，而后调用本地操作系统；

8、服务器操作系统将含有结果的消息发送给客户端操作系统；

9、客户端操作系统将消息交给客户存根；

10、客户存根将结果从消息中提取出来，返回给调用它的客户存根。

以上步骤就是将客户过程对客户存根发出的本地调用转换成对服务器过程的本地调用，而客户端和服务器都不会意识到中间步骤的存在

2 系统分析与设计

整个实验的实现主要分为客户端部分和服务器部分，下面分开来进行讲解。

2.1 服务器端的设计与分析

服务器端主要有几个重要的部分组成，RPC 注册、建立 TCP 连接、消息监听、RPC 调用（如图 2）

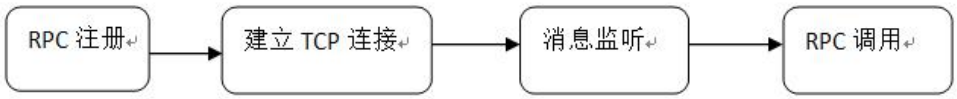


图 2　服务器端步骤

RPC 注册：

```
func (t *Rpc_time)Timerpc(a string, reply *time.Time) error{
    if(a == "123456"){
        *reply  = time.Now()
        fmt.Println(time.Now())
        return nil
    }
    reply = nil
    return  nil
}
```

建立 TCP 连接：

```
tcpAddr, err := net.ResolveTCPAddr( network: "tcp",  address: ":3339")
listener, err := net.ListenTCP( network: "tcp", tcpAddr)
```

消息监听、RPC 调用：

```
    for {
        conn, err := listener.Accept()
        if err != nil {
            continue
        }
        go rpc.ServeConn(conn)
    }
```

2.2　客户端的设计与分析

客户端主要是请求 RPC，然后利用从服务器端得到的时间来对客户端原系统时间进行同步，时间同步的过程中存在传输延时问题，因此我们进行时间同步时必须考虑网络的时延问题。

客户端的也可以分为几大块：建立 TCP 连接、远程调用 RPC 服务、时间校正（如图 3）

图 3 客户端步骤

建立 TCP：

```
client, err := rpc.Dial( network: "tcp", serverAddress )
if err != nil {
    fmt.Println( a: "连接服务端失败:", err.Error())
    return
}
fmt.Println( a: "已连接服务器")
```

远程调用 RPC 服务：

```
client.Call( serviceMethod: "Rpc_time.Timerpc", a, &reply)
```

时间校正：

时间校正这个模块需要特殊说明一下，因为从服务器的时间传输过来到客户端接收期间存在一个时延，需要特殊计算。

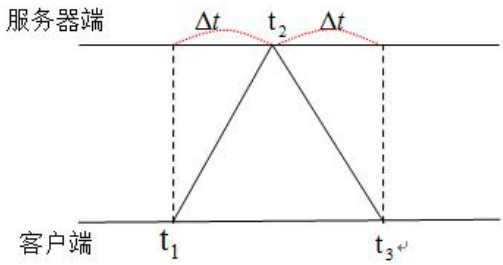


图 3 服务器与客户端的时延

如图 3，客户端发送时间请求到服务器端，服务器接收请求后再将系统时间传回客户端，这两个过程都存在时延，为了简便计算，我们假设两段传输过程中的时延相同都为 Δt，且服务器端接到请求后立即响应，我们得到以下时间计算公式：

$$t = t_2 + \Delta t$$

$$\Delta t = (t_1 + t_3)/2$$

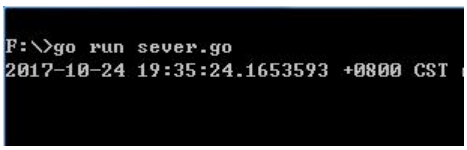
由上述两个公式得到最终客户端系统的时间 t 为：

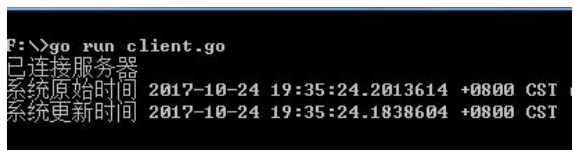
$$t = t_2 + (t_1 + t_3)/2$$

其中 t 为客户端应该同步的时间，t_1 为客户端发送请求前的系统时间，t_3 为客户端收到响应后的系统时间，t_2 为服务器发出的同步时间。

3 实现演示

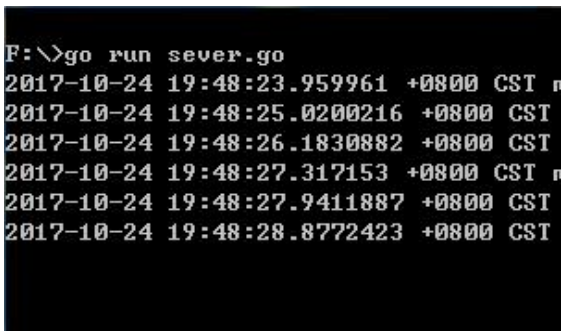
客户端进行时间同步效果：



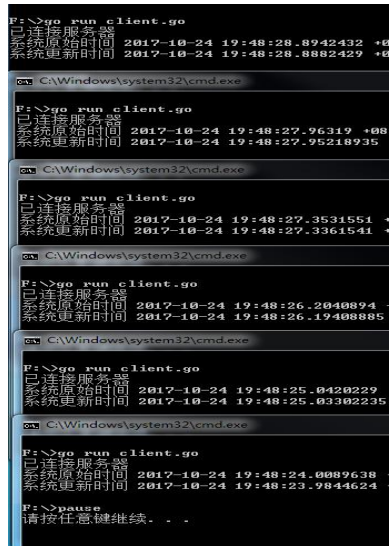


当客户端的密码错误时无法同步时间：

多客户端同时请求时服务器及客户端显示：



服务器显示



多客户端显示

4 总结

本次实验主要是为了测试 RPC 的使用，让我们初步了解 RPC（远超过程调用），实验要求利用 RPC 实现客户端与服务器的时间同步，并且要考虑网络时延问题，本研究采取了一个简单的利用来回时间差来消除网络时延的办法。同时实验要求跨平台多线程，并利用 go 语言实现，这相对是一件容易的事，本次实验整体难度比较小，没有特别棘手的问题。

References:

[1] 百度百科 “远程过程调用协议”词条.[EB]/[OL].
https://baike.baidu.com/item/%E8%BF%9C%E7%A8%8B%E8%BF%87%E7%A8%8B%E8%B0%83%E7%94%A8%E5%8D%8F%E8%AE%AE/6893245?fr=aladdin 2017.10.23

[2] 张文直, 邱旭华, 周东平. 远程过程调用(RPC)在三层分布系统中的应用[J]. 警察技术, 2005(4):31-33.