远程过程调用(RPC)在三层分布系统中的应用

公安部第一研究所 张文直 邱旭华 周东平

摘要:本文主要介绍了在三层分布式网络系统中,如何使用远程过程调用(RPC)技术来实现客户机与服务器之间的通信。

关键词:RPC Client/Server 远程过程调用 码桩 应用服务器 分布式应用

一、引言

如今,越来越多的数据库应用系统使用了三层分布架构,即在原有的客户机/服务器(Client/Server)体系结构之间加入了一层或多层应用服务程序,这种程序称为"应用服务器",系统体系结构框图如图1所示。开发人员可以将应用的商业逻辑放在中间层应用服务器上,把应用的业务逻辑与用户界面分开。在保证客户端功能的前提下,为用户提供一个简洁的界面。如果需要修改应用程序代码,只需要对中间层应用服务器进行修改,而不用修改成千上万的客户端应用程序。从而使开发人员可以专注于应用系统核心业务逻辑的分析、设计和开发,简化了应用系统的开发、更新和升级工作,有很高的可用性。

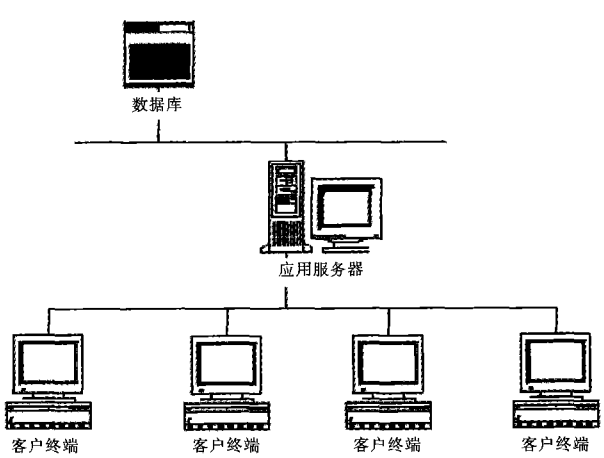

图1 三层分布架构框图

在上述的网络通信体系结构中,为了便于开发人员使用大型应用服务器作为网络服务器完成数据存储、查询等操作,微软提出了自己的远程过程调用(RPC)技术,通过安装平台SDK,将使得RPC开发环境和运行时间库自动安装。并最终使得用C语言编写的过程库可以在其他计算机上运行。一个应用程序可以与使用RPC实现的库链接,而不用向用户说明这个程序在使用RPC。

作为一种创建分布式应用程序的工具,RPC编程模型已经是熟悉的,可以很容易地将函数转换成远程过程,这简化了开发和测试周期;同时RPC对开发人员隐藏了许多关于网络接口的细节,能够不必了解具体的网络函数或低层次的网络协议,就能实现功能强大的分布式应用程序。

二、远程过程调用(RPC)技术介绍

1、RPC原理

RPC其实也是一种C/S的编程模式,有点类似C/S Socket编程模式,但要比它更高一层。当我们在建立RPC服务以后,客户端的调用参数通过底层的RPC传输通道,可以是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,且此时的客户端处于等待状态,直至收到应答或Time Out超时信号。当服务器端获得了请求消息,则会根据注册RPC时告诉RPC系统的例程入口地址,执行相应的操作,并将结果返回至客户端。当一次

RPC 调用结束后，相应线程发送相应的信号，客户端程序才会继续运行。

RPC 的体系结构如图 2 所示，RPC 工具使得客户机好像直接调用远程服务器程序的某个过程。客户机和服务器都有自己的地址空间，客户机调用的是一个本地的码桩(stub)过程，客户机的码桩代码并不包含实现远程过程的实际代码，而是：

* 从客户机的地址空间检索需要的参数；

* 将需要的参数翻译成标准网络数据表示格式(NDR)，以在网络上传输；

* 调用 RPC 客户机运行时间库中的函数，以将请求和参数传送给服务器；

* 服务器的 RPC 运行时间库函数接受请求，调用服务器的码桩过程；

* 服务器的码桩过程从网络缓冲区中检索参数，并将其从网络传输格式转换成服务器需要；

* 服务器的码桩过程调用服务器上的真实过程；

* 远程过程开始运行，可能产生输出参数并返回值；

* 远程过程完成后，通过一系列类似的步骤向客户机返回数据；

* 客户机接收网络上的数据，将其返回到调用函数，从而完成整个调用过程；

* 客户机函数继续执行，好像此过程在同一台计算机上调用的一样。

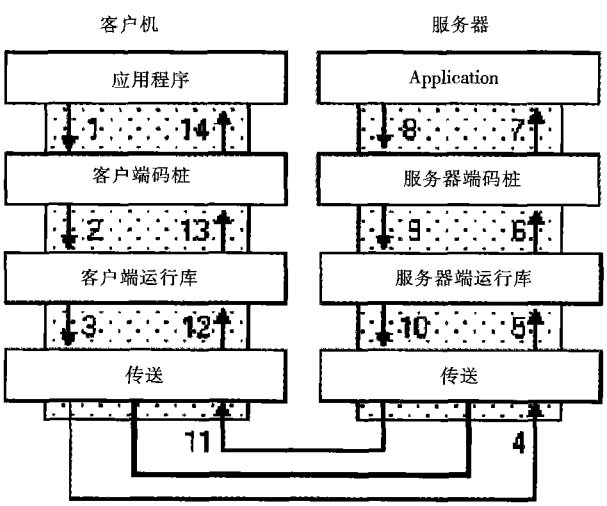

图 2　RPC 体系结构

2、创建 RPC 应用程序

在 RPC 模型中使用一个专门为此目的而设计的语言，在形式上为远程过程指定接口。这种语言叫接口定义语言(IDL)，微软对这种语言的实现叫做微软接口定义语言(MIDL)。当开发 RPC 应用程序时，需要用文本编辑器定义接口，并将其存储在以。idl 为扩展名的文本文件中。RPC 接口用来描述服务器程序实现的远程函数。这个接口确保当客户程序向服务器请求一个远程过程时，客户和服务器使用同样的规则进行通信。定义了接口以后，必须通过 MIDL 编译器对其编译。用 MIDL 编译器来生成一个头文件，程序将在客户和服务器的源文件中导入它。MIDL 编译器还生成两个源文件。将其中的一个编译并链接到客户程序中，另一个编译并链接到服务器程序中。这两个 C 源文件分别是客户码桩和服务器码桩。码桩是一些代理函数，这些函数去调用负责管理远程过程调用的运行时间库函数。

当为一个分布式应用程序创建服务器程序时，必须使用 MIDL 编译器生成的头文件和服务器码桩。在服务器 C 程序文件中导入头文件，将服务器码桩同组成应用程序的 C 源程序在一起编译。将生成的目标文件同 RPC 运行时间库链接在一起。这一过程如图 3 所示。就像图中展示的例子中看到的那样，一个叫做 MyApp.idl 的文件用来定义接口。MIDL 编译器用 MyApp.idl 生成 MyApp_s.c 服务器码桩文件、MyApp_c.c 客户码桩文件和 MyApp.h 文件。服务器程序的 C 源文件(在这个例子中是 Mysrvr.c)需要导入 MyApp.h 文件，还需要导入 RPC.h 和 RPCNDR.h 等运行时间库文件。Mysrvr.c 与 MyApp_s.c 文件一起编译生成目标文件。然后目标文件与 RPC 运行时间库以及任何可能需要的其他库文件链接在一起。结果生成一个名字为 Mysrvr.exe 的可执行服务器程序。

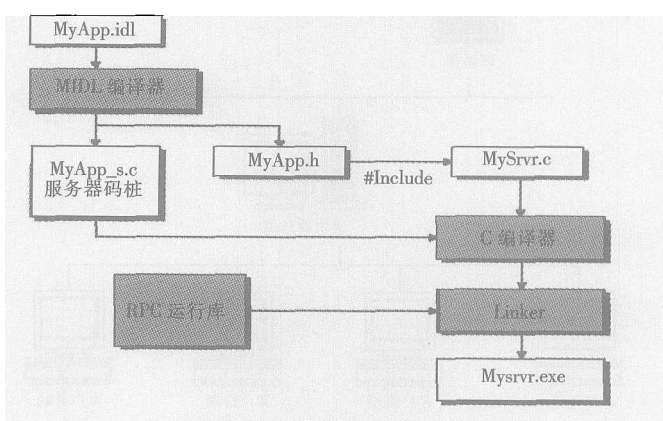

图 3　分布式应用创建服务器程序

开发客户机程序与开发服务器程序类似。客户机程序的 C 源文件 MyClnt.c 和 MyApp_c.c 文件与 RPC 运行时间库及任何客户程序需要的其他库文件编译并链接在一起，结果是生成一个名字为 MyClnt.exe 的可执行的客户程序。

最后可以在两台安装了微软 Windows NT/Windows 2000 的局域网机器上运行应用程序。在第一台机器的窗口中，键入：C:\>Mysrvr，然后，在第二个机器的窗口中，键入：C:\>MyClnt。

三、系统应用实例

第二代居民身份证的卡体初始化系统是二代证应用系统中的重要部分，卡体初始化系统接收制卡中心制成的空白卡体，通过初始化设备生成并写入每张卡体的个性化密钥。同时，系统数据库存储全部初始化卡体的编号信息。

卡体初始化系统内部结构需采用前、后台隔离的方式，分为管理控制区和操作终端区，整个系统使用局域网连接。管理控制区内配置有应用服务器、加密服务器、数据库服务器和审计服务器，实现各种核心服务功能以及数据备份和管理等。操作终端区是初始化工作的具体实施工位，由初始化设备、终端计算机和应用软件组成。

应用服务器在二代证初始化过程中起着生产调度与管理的作用。它给操作终端提供一系列程序接口，这样不仅防止操作终端对数据库进行直接的操作，而且还增加了生产过程的自动化程度。其总体结构如图 4：

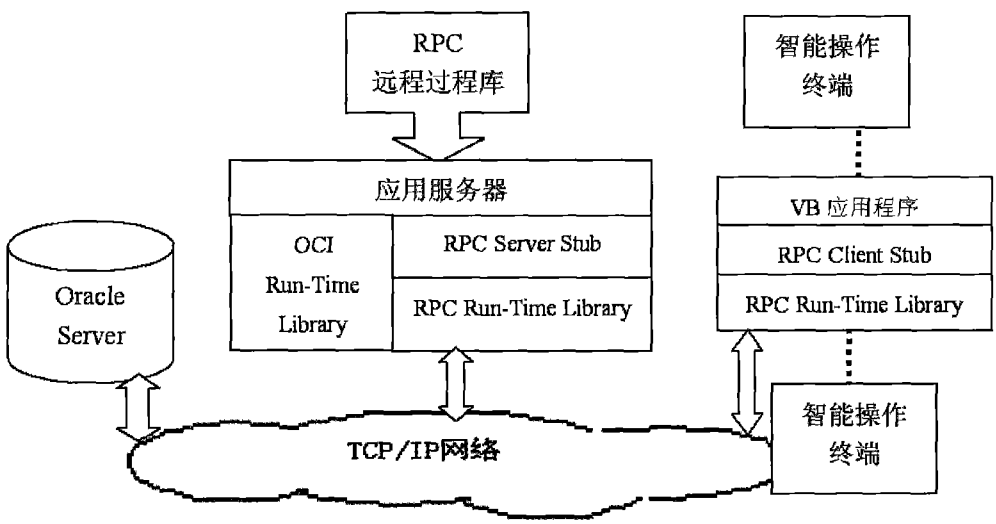

图 4　总体结构

该系统在技术上主要采用 RPC 来实现客户端与服务器的通信，而在服务器与数据库之间则采用 OCI(Oracle Call Interface)接口来实现通信。终端通过 RPC 客户端(Stub)调用应用服务器提供的远程过程，终端只需将调用接口所需要的参数传递给应用服务器，而程序的运行实际上是在应用服务器上完成的。服务器处理传来的信息，通过 OCI 接口访问 Oracle 服务器，进行必要的验证，将结果反馈给终端，将从终端传来的数据传输给 Oracle 数据库，对数据库中的表进行查询、插、删、改等操作。在实际使用中，该系统现已经进入大规模生产阶段，并且在非常频繁的并发访问中，RPC 技术的稳定性一直是值得信赖的。

四、结束语

RPC 网络通信技术作为开放软件基金会(OSF)定义的一个完整的分布式计算环境的一部分，具有三个最重要的特性：简单性、透明性和高性能，能够尽量缩短开发人员用来学习新环境的时间。相信在随后的工作中，RPC 将会得到更加广阔的应用空间。

参考文献：

1.RPC.CHM 微软开发人员网络(Microsoft Developer Network)参考库。

2. RPC 系列编程讨论 http://bbs.tsinghua.com。■