客户存根的功能是获取调用的参数并将参数打包放入消息中送往服务器存根。虽然这听起来很直截了当，但实际并不像表面上那么简单。本节我们将讨论 RPC 系统中有关参数传递的几个问题。 将参数打包形成消息的过程称为参数组装（parameter marshalling）。

（1）参数类型。理解参数类型在远程过程调用中和在本地过程调用中一样重要，因为它影响着你对结果的预期值。RPC 中有三种参数传递类型：第一种是输入参数，这种参数只用于客户向服务传递信息；第二种是输出参数，这种参数只用于服务向客户传递信息，客户不能使用它向服务发送信息；第三种既作输入又作输出的参数，客户能用这种类型的参数向服务传递信息，服务同样能用这个参数向客户传递信息。

（2）数据类型的支持。数据类型的支持指的是在远程过程调用中哪些数据类型可以作为参数使用。同各种程序设计语言一样，RPC 也有可能限制参数的复杂程度。一般的 RPC 对参数的个数进行了限制而允许较复杂的数据类型。例如只使用一个参数，但是参数可以是一个复杂的结构，这样一来，程序设计者可以通过比较方便的办法绕过这种限制。

（3）参数组装。为了进行有效的通信，参数和较大的数据结构需要进行打包。打包的功能将信息平面化和压缩，使得信息能以紧凑的方式表达，以减少通过网络传送的信息量。这种方法应该是明确的，以保证接收方能够正确地进行拆包和解压缩信息。stub 程序的功能是把这种过程调用中所带的参数打包和拆包组装和拆卸成消息形式，并进行相应的类型检查，然后把这些消息转移给 RPC runtime 子程序，后者再把它们发送到系统中的其他场点。每当调用一个远程过程的时候，都要使用代理。如图 1 所示。代理程序常用扩展库的形式提供，这不仅能减轻编程人员的负担而且能提高透明性。但是许多 RPC 的实现要求程序员实现代理。实际上，使用 RPC 可以隐匿下层的通信细节，例如 sockets 通信机制。图 1 描述了一个远程过程调用的执行过程。

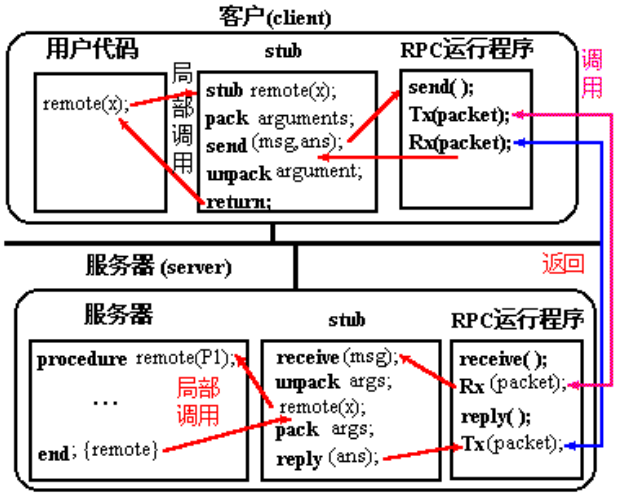

图 1　远程过程调用的执行过程

在分布式系统中，数据库一般驻存在服务器上，客户机（进程）通过远程数据库服务功能访问数据库服务器（如图所示），现有的远程数据库服务是使用 RPC 模式的，例如 Oracle 数据库提供了存储过程机制，系统与用户定义的存储过程在数据库服务器上，用户在客户端使用 RPC 模式调用存储过程获取数据，如图 2 所示。

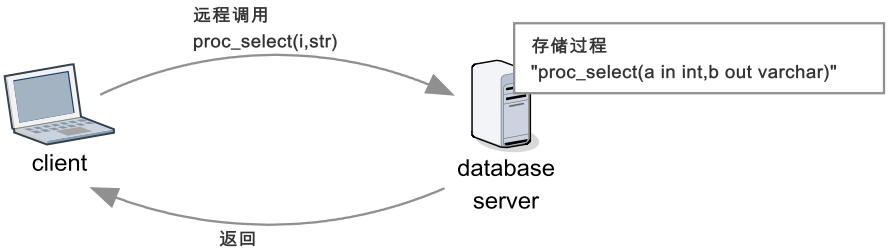

图 2　Oracle 的存储过程机制