多层C/S模式分析及使用C++Builder3的实现

中南工业大学信息工程学院　邹涛　陈志刚

一、传统双层C/S模式的特点及其不足

计算机网络计算经历几个发展阶段：从最初的主机/终端(Mainframe/Terminal)模式到文件服务器/工作站(File Server/Workstation)模式，最终发展到今天的C/S模式。

1. 传统C/S模式的特点

传统双层C/S模式的逻辑结构如图1所示。

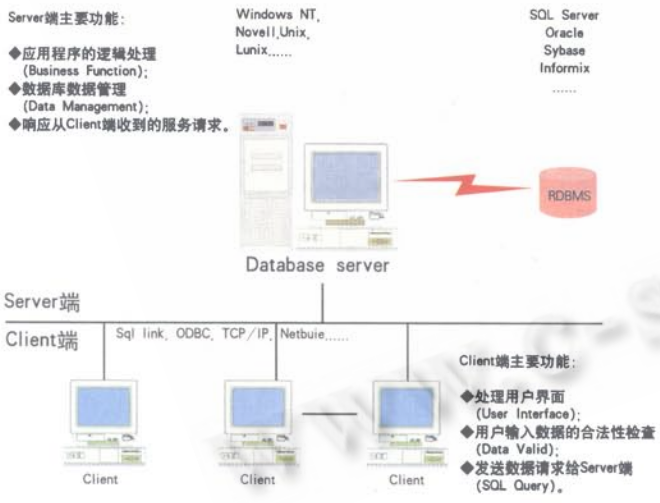

图 1　双层C/S模式的逻辑架构

对比文件服务器/工作站模式，C/S模式在充分发挥分别作为服务器以及工作站的计算机能力的基础上，极大地减少了通过网络传输的数据流量，从而大大地提高了整个应用系统的运行效率。

遗憾的是，分析表明，上述主－从结构的双层C/S模式也存在明显的缺点。

2. 传统双层C/S模式的不足之处

首先，由于客户端和服务器端直接连接，服务器将消耗部分系统资源用于处理与客户端的连接(Connection)工作。那么每当同时存在大量客户端数据请求时，服务器有限的系统资源将疲于应付与客户端之间的连接，从而无法及时响应数据请求。客户端数据请求增多的直接后果将导致系统整体运行效率的大幅降低甚至全面崩溃。

其次，主－从式的结构中，唯一在线的数据库服务器成为系统可靠性的极大隐患。如果数据库服务器因为某种原因停止工作，那么整个系统将趋于瘫痪。

最后，客户端应用程序的分发工作的烦琐程度令人难以接受。系统开发过程完毕，随之而来的程序分发除了要求为每台客户机安装客户端程序的执行文件(*.exe)以外，

还要求安装程序运行所必须的动态连接库文件(*.dll)、程序初始化文件(*.ini)等许多其他文件。另外，还必须完成每台客户机器的ODBC或BDE(Borland Database Engine)的配置工作。不仅如此，每次对客户端程序的修改和升级，又意味着上述相同分发过程的又一次重复。

为什么每年全球有超过40%的基于C/S模式的应用系统开发以失败告终?通过上述分析也就不难找出其原因了。

二、多层C/S模式的改进

正是由于上述缺点和不足，自C/S模式问世之日起，人们就不断致力于改进和完善它。

针对上述第一个问题，解决方法是在客户端与数据库服务器之间加入一个业务逻辑(Business Logic)层，该层通常存在于另一台被称为应用服务器(Application Server)的机器上，(当然也可以存在于客户端计算机中，只是这样就无法充分发挥应用服务器的优越性)。

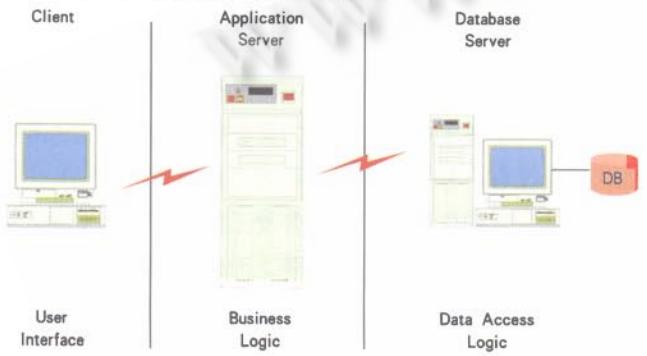

图 2 三层C/S模式逻辑架构

新加入的业务逻辑层将专门负责数据库服务器与客户端的连接工作。它一方面把客户端请求传给数据库服务器，另一方面将服务器的查询结果回送给客户端。于是数据库服务器只用专心于响应客户端数据请求。这样就大幅度减轻数据库服务器的负担，从而提高了其响应速度。所以系统的整体工作效率得以提升。

另外，还可以将原来系统中使用的一些商业逻辑规则的处理工作(既可在客户端又可在服务器端实现)也分配给业务逻辑层完成，于是客户端和服务器端程序的功能得到进一步简化。

将上述三层C/S架构稍加变形，可进一步提高系统的可靠性和运行效率。此时是在客户端与应用服务器之间再加入一种被称为代理服务器(Business Broker Server)的计算机，并增加一台或多台作用完全相同的应用服务器。这样，三层C/S架构就演化成为多层C/S架构，如图3所示。

新加入的代理服务器具有两个基本功能：一个是平衡负载(Load Balancing)，另一个是失败绕过(Fail Over)。所谓平衡负载是指代理服务器将客户端发出的众多请求，按

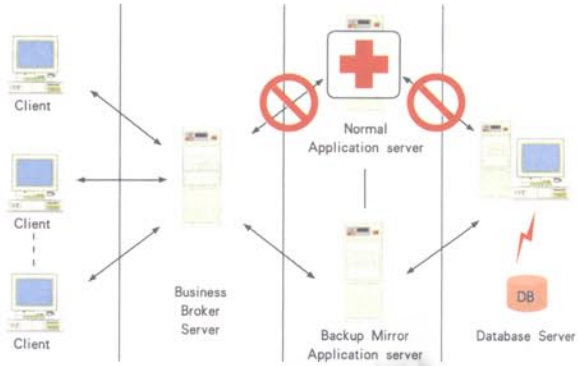

图 3 多层C/S模式逻辑架构

照当前各个应用服务器的繁忙程度，平均分配给各个应用服务器连接处理；失败绕过发生在当一台或多台应用服务器发生故障时。此时，代理服务器能够判断系统故障所在，并将客户端请求自动地发送到正常运行的应用服务器上，从而避免整个系统停止运作的窘况发生。

采用多层C/S模式架构，软件系统的层次性更加清晰。数据库服务器端特别是客户端的编程大大地简化。有些实际的多层C/S系统甚至实现了客户端零代码编程(如基于WWW的数据库应用系统，这类系统采用统一的浏览器作为用户界面)。软件的分发由此变得异常轻松，软件升级工作也变得轻而易举，因为升级常常只须针对分布在应用服务器上的中间层次软件而不用涉及客户端的用户界面。

实际的多层分布式C/S应用系统的另一个突出优势是非常具有弹性，它可以保证在充分利用公司既有资源的前提下，不断扩充公司内部的整体系统的规模。图4展示的是一个实际的基于多层C/S架构应用系统的逻辑结构图。图中的web/邮件服务器实质就是一台应用服务器。系统还可根据实际需要添加诸如互连网代理服务器(Internet Proxy Server)、防火墙(Firwall)等多种其他类型的应用服务器。

三、使用C++Builder3构造多层模式的数据库应用

C++Builder是Borland公司推出的Windows快速应用程序开发工具。它继承了Delphi完全相同的用户界面以及绝大部分Delphi控件和帮助文档。它能够方便地让使用这两

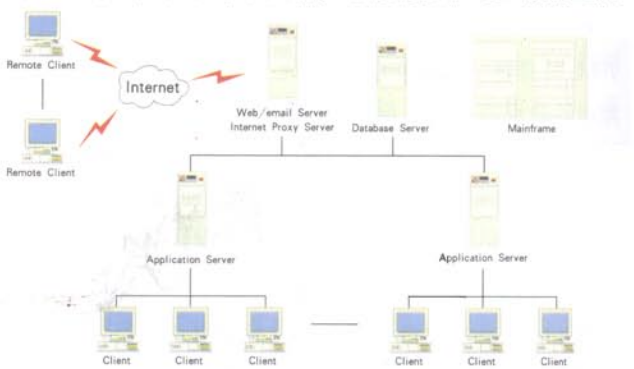

图 4 一个实际应用系统的逻辑结构图

种语言的程序员共享程序代码和其他各类资源。C++Builder以其提供的丰富的各类控件、高效的C++编程语言以及简单易学等突出优点，受到各级程序员的青睐。

1.C++Builder 3的多层C/S模式应用的架构

在C++Builder 3开发的传统双层C/S应用中，客户端的数据感知控件(Data-aware Component如TDBGrid、TDBEdit、TDBListBox等)直接通过TDataSource控件与数据存取控件(Data-access Component如TTable、TQuery、TStoredProc等)通信，后者再利用BDE或ODBC与服务器端的数据库交互，如图5所示。

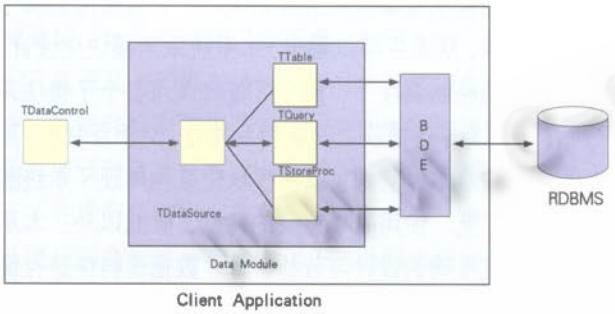

图 5 C++Builder3 双层C/S程序架构

而在多层C/S应用中，随着与数据库服务器之间的连接功能转移到应用服务器上，完成上述功能的数据存取控件也就随之移到了应用服务器上。C++Builder3为这种新的连接方式提供了相应的接口控件。如图6，在客户端，TClientDataset控件作用是提供一个用户接口，该用户接口使本地的数据感知控件能够存取通过应用服务器“代理”的数据库数据的一个本地拷贝；而TMIDASConnection控件则具体完成TClientDataset与应用服务器的连接工作。

在应用服务器中，通常会存在一个或多个TProvider控件，TProvider控件提供了客户端与数据库的一个通信接口。

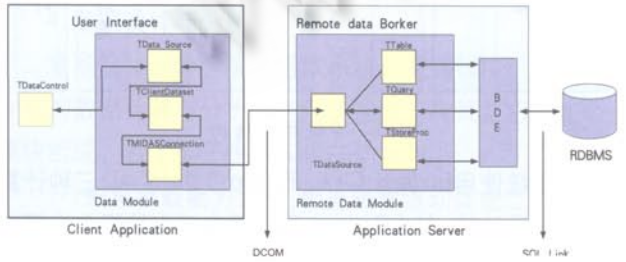

图 6 C++Builder3 多层C/S程序架构

2.使用C++Builder3开发多层C/S程序的步骤

使用C++Builder3开发一个多层C/S程序包括两方面的工作：首先必须编制，运行和注册一个应用服务器。因为C++Builder3提供的运行在应用服务器上的接口控件(TProvider)是基于DCOM(Distributed Component Object Module)方式的，所以需要在使用前对该应用服务器程序进行注册(与使用ActiveX及OLE时类似)；第二步工作才是编程实现相应的客户端程序。需要注意的是以上两步骤的前后次序是严格规定的，即在开始第二步工作之前就必须编程实现并运行一个应用服务器应用。

(1)创建应用服务器。依照以下步骤创建一个应用服务器：

①在项目中添加一个新的远程数据模块(Remote Data Module)。在模块中加入需要的数据存取控件并将它们与数据库连接起来。

②在数据模块中加入一个TProvider控件，必须显式输出数据模块中的每一个TProvider控件以使之同时在类型库(type library)中注册。选中TProvider控件，单击右键，在弹出的菜单中选择Export From<TProvider控件名称>in Data Module项。

③设置TProvider控件的DataSet属性。

④编码实现应用服务器的共享业务规则等附加功能。此步骤为可选步骤。

⑤保存，编辑并运行该应用服务器。尽管此时没有绝对必要运行它，但是因为一方面必须通过一次运行将它作为一个OLE Automation对象加以注册；另一方面，此举将方便程序员在客户端应用开发阶段的调试工作。

至此，一个应用服务器已经存在并运行。下一步工作是实现客户端功能。

(2)创建客户端应用。依据以下步骤创建客户端应用：

①在项目中添加一个新的数据模块(Data Module)。

②在数据模块中加入一个TMIDASConnection控件。

③设置TMIDASConnection控件的ConnectType属性和其相应属性以确定与应用服务器的连接方式。可选的方式有使用DCOM方式、使用TCP/IP套接方式或者OLEnterprise方式。

④设置TMIDASConnection控件其他必需的连接属性。例如，当采用DCOM方式连接应用服务器时，必须设置ComputerName属性以确定实现应用服务器功能的所在的计算机。

⑤添加需要的TClientDataSet控件，设置它们的RemoteServer属性为上面步骤中的TMIDASConnection控件的名字。

⑥设置每一个TClientDataSet控件的ProviderName属性，使之连接上远程应用服务器中的TProvider控件。

(下转32页)

(上接31页)

完成上述步骤后，客户端即可以通过应用服务器与数据库进行通信。继续加入数据感知控件和相应的TDataSource控件，通过编码以最终实现用户界面的功能。

四、结束语

C++Builder是目前众多程序开发工具中较早支持多层C/S模式的可视化开发工具，是一种公认的优秀可视化开发工具。随着以Internet技术为代表的新一代技术的成熟并融入到社会的方方面面中，也随着C/S模式本身的不断完善和发展，不难预见，将来的程序开发工具能使我们更加方便快捷地开发出构建在新一代C/S模式下的应用程序。

参考文献

[1] 陈周造著。Borland C++Builder 3.0规划与实作—进阶篇。台北：硕博文化有限公司，1998.6

[2] Borland C++Builder 3.0联机帮助。Inprise Inc.