

# 基于 DCOM 和 MIDAS 实现分布式考试系统研究

## Research On Realization of Distributed Examination System Based On DCOM And MIDAS

宋丽红 (天津大学管理学院 300072)

**摘要:**本文以网上考试系统项目为背景,对采用 DCOM 和 MIDAS 实现分布式网络考试系统中涉及到的运行机制、线程模式、实例模式及远程数据模块等技术问题进行了深入研究,提出了系统实现的框架、具体方案和编程方法。

**关键词:**DCOM 多层分布式应用服务 远程数据模块 线程 套间线程

### 1 引言

本文基于我校教学管理与学习的实际需要,设计开发了网络教学与考试系统,旨在充分利用现有的计算机和网络资源,提高考试的效率,建立数字化题库的系统,使考试标准化,公平化,同时减轻教师的负担。

### 2 分布式系统结构

分布式系统结构把传统两层结构中的客户/服务器模式扩展为客户层、中间层和数据层,克服了两层结构的缺点,提供了系统开发配置更大的灵活性、通用性和扩展性,其特点和优势突出表现为<sup>[1]</sup>:

(1) 瘦客户体系结构。客户层主要负责用户界面及交互控制,根据用户需求,连接中间层,显示从应用服务器获取的数据。客户层不再负责处理复杂业务逻辑、计算和数据库访问等关键事务。因此,客户层代码简单,程序小巧,执行效率高,易于开发、安装和维护,被称为瘦客户体系。从根本上改变了传统的二层体系结构的缺陷,不仅把客户机从沉重的负担和不断对其提高性能的要求中解放出来,也把技术维护人员从繁重的维护升级工作中解脱出来。

(2) 业务逻辑集中。中间层也称为业务层,它构成了系统应用的业务逻辑。分布式系统中客户层和数据层都不涉及复杂的业务逻辑,由中间层即应用服务器集中处理。中间层作为客户层和数据库服务器的通讯桥梁需要访问数据库服务器,完成网络连接与通讯,并为客户端提供远程调用的接口,它是整个系统的核心。当系统业务逻辑发生变化时,只需修改应用服务

程序,而不需要更新分布广泛的客户层程序,层与层之间相互独立,任何一层的改变不影响其它层的功能,降低了整个系统维护和升级的代价,系统可维护性大大提高。

(3) 大幅度减轻数据服务器的负荷。数据层一般由支持分布式应用的大型数据库服务器实现,如 Oracle、MS SQL Server 等。在两层结构中,客户层与数据库服务器直接相连,当与数据库服务器连接的客户层数目大幅度增多时,数据库服务器性能大幅度下降,甚至导致系统崩溃,这是两层结构致命的缺点。但在分布式结构中,各应用程序服务器接管了众多来自客户层的连接和数据处理请求,数据库服务器只需响应来自应用服务器的请求,这样就不会因为客户机的大幅度增加而影响整个系统的运行性能,提高了整个系统的可扩展性。

### 3 实现分布式考试系统的关键技术

#### 3.1 DCOM 组件技术

DCOM 是微软提出的分布式组件对象模型,是 COM 组件在分布式网络异构环境中的扩展。在 Windows 平台微软对 DCOM 的工作机制进行了优化,使得 DCOM 协议较其他协议具有更高的传输效率和安全保证<sup>[4]</sup>。

DCOM 为网络中组件间的通讯提供了平台,可以使分布在网络中不同计算机上应用不同开发语言开发的对象通过远程过程调用(RPC)直接通信,为实现分布式系统提供了解决方案。DCOM 不但继承了

COM 组件技术组件复用、位置独立性,语言无关性,可扩展性等优点,而且具备网络环境中分布式组件间访问的安全性,动态负载平衡,容错及链接的自我管理特点,大大简化了分布式系统的开发过程,降低了开发难度。

客户层通过 DCOM 机制链接到网络中服务器上的组件,调用其接口中的方法,这个过程中客户层并不能看到要链接的服务器,也不需要知道服务器的位置。这一点使得分布式系统的组成和配置保持了很大的灵活性,为将来增加、修改或取消各级应用服务器提供了方便。DCOM 选择了 TCP/IP 协议套件中无链接的 UDP 协议作为网络间各组件的传输协议。UDP 协议的无链接特性,使网络传输效率大幅度提高,节省了宝贵的网络资源,大大降低了分布式系统的开销。

### 3.2 MIDAS 方案

MIDAS (Multi-tier Distributed Application Services) 即多层分布式应用服务,它是由 Borland 公司提出的多层分布式系统结构的解决方案。它内置了对 DCOM Windows Socket, http, CORBA 等通信协议的支持,实现了复杂业务逻辑与客户层的分离,客户层与数据库服务器的分离<sup>[5]</sup>,如图 1 所示。

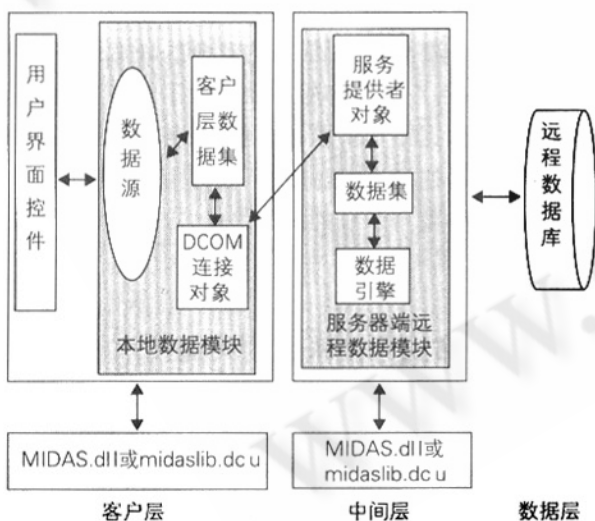


图 1 基于 MIDAS 实现分布式系统原理图

基于 MIDAS 分布式系统的运行机制:客户层通过 DCOM 连接中间层,根据用户的需求发出请求。中间层的远程数据模块相当于一个 COM 对象,为客户层提供调用的接口。中间层通过数据集组件通过数据引擎

连接数据库,把根据客户需求和业务逻辑定制的 SQL 语句对数据库进行查询或存取数据等操作。中间层把获取的数据集,压缩成数据包,通过 DCOM 对象接口传给客户层。客户层和服务器端都需要动态连接库 MIDAS.DLL 或编译文件 midaslib.dcu 的支持<sup>[5]</sup>,它负责对数据包进行管理。客户层收到数据包后,还原成数据集传给客户程序本地数据模块中的数据集组件,再由数据源作为桥梁连接界面上的显示控件。客户层程序可以直接对本地数据集中的数据进行操作。当用户对数据进行编辑修改后,可以向应用服务器申请更新数据。应用服务器收到客户程序的申请后,就向远程数据库服务器提交更新数据。

## 4 分布式考试系统的设计与实现

### 4.1 分布式考试系统体系架构

系统架构如图 2 所示。

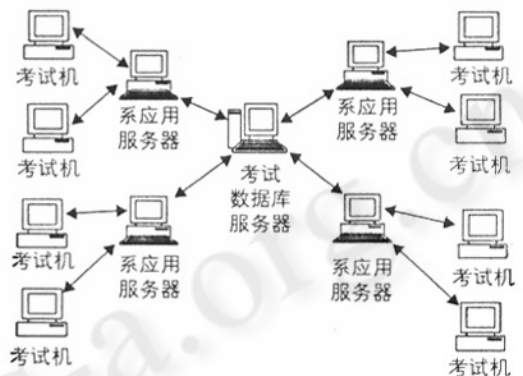


图 2 分布式考试系统体系架构图

数据层是存储大量学习资料及考试题目的数据库服务器,各系或部门设置应用服务器作为中间层,各考试机作为客户层为实际参与考试的计算机。

### 4.2 各级应用服务器的设计与实现

创建分布式系统,其核心任务是创建、注册并安装各级应用服务器。应用服务器通过添加接口,方法和属性来完成系统的各功能模块,如题库管理、考试管理,考务管理等。除此之外,它作为分布式系统的中间层,需要创建远程数据模块,完成与远程数据服务器的连接工作,并为客户层提供服务。

#### 4.2.1 建立远程数据模块 (Remote Data Module)

系统采用自动化对象 TRemoteDataModule 创建远

程数据模块。它相当于 COM 对象,提供客户层访问接口。创建时需要选择 COM 对象的实例属性和线程模式。常用的实例属性为:

(1) Single Instance。单实例模式只允许让一个客户层访问服务程序,即每个服务进程同时只能为一个客户请求服务。当每个客户程序连接到应用服务器时,服务器都会启动一个服务进程为它服务。因此应用服务器会有多个服务进程同时运行,各客户机的请求会被马上满足。这种模式对于考试机来说效率很高,但显然不适合分布式考试系统。因为考试开始时,会有很多考试机同时连接到应用服务器,这种模式会因应用服务器端启动多个进程处理各个请求,而占用大量系统资源,使应用服务器的整体运行效率大大降低。

(2) Multiple Instance。多实例模式允许连接到应用服务器的各个客户程序共享应用服务器的服务进程。当客户机连接应用服务器时,服务进程可能已经启动也可能还没有。如果没有,客户程序的请求将自动激活服务程序。这种模式下,多个客户机共享一个服务进程,从客户机角度看服务被激活和运行的速度快,响应时间短;从应用服务器角度看,系统对所有来自客户层的请求只启动一个进程,并行处理各客户层的请求,不会耗费大量的系统资源,系统的性能和稳定性都有所保证。但服务进程内部同时响应多个客户请求,需要进行多任务处理和共享变量的保护。

常用的线程模式有:

(1) Single。单线程模式,服务程序一次只能处理一个客户请求。服务程序以循环方式处理各客户机的请求,不需要对多线程的进行处理。但对于本系统不可取,因为服务器一次只能处理一个考试机的请求,其他考试机的请求需在等待队列中等待前面的客户请求处理完毕后,才可以得到服务程序的响应。

(2) Apartment。套间线程模式,每个客户的请求会动态启动一个 COM 对象来处理,一个应用服务器进程可以有多个不同的套间,在每一个套间中可以有多个 COM 对象。因此它可以服务于许多需求相同的客户层。当客户层断开与服务器的连接时,服务于它的 COM 对象自行销毁。在套间内部,各线程分别完成各自的业务处理过程。Delphi 的可视化组件库 (VCL) 对套间线程有很好的支持<sup>[2,3]</sup>,从系统的运行效率和开发

过程的复杂性考虑,目前应用最广泛,系统实现效果最好的线程模式是套间线程。每一个远程数据模块对象同时只能为一个客户服务,对象中的数据是安全的,当前客户的多重请求在各自独立的线程中处理。但是要处理多线程访问全局共享内存的问题。

在本系统中,采用了如图 3 所示的套间线程和多实例结合的方案,多个考试机和应用服务器同时通讯,提高了系统的并行处理能力。

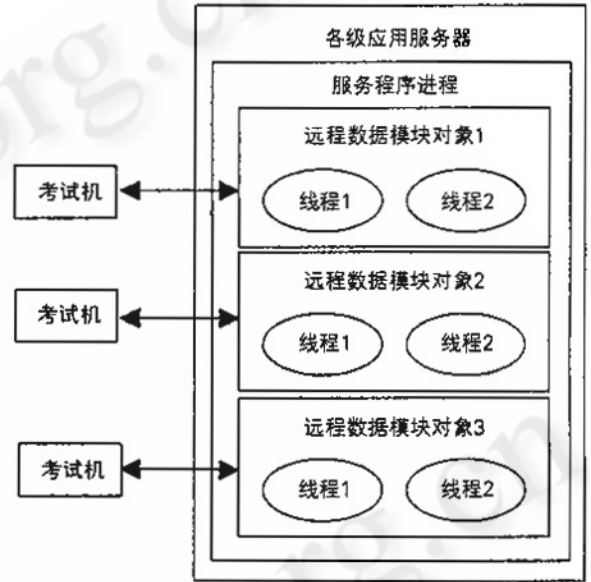


图 3 系统采用的多线程和多实例方案

4.2.2 链接远程数据库 鉴于系统选择多个实例套间线程模式创建 COM 组件,最好使用基于 BDE 数据引擎的数据集组件<sup>[4]</sup>,并且在远程数据模块中加上一个 TSession 组件对象来管理与数据库的会话,它的 AutoSessionName 属性值要设置为真。因为如果没有 TSession 对象,所有数据模块都共享一个系统缺省的 Session 对象与数据库连接,它们之间相互影响,容易造成系统瘫痪。Session 加到数据模块中,可以使每个客户层连接的远程数据模块都可以在独立的数据库连接会话中,而不会受到其他客户层的影响。远程数据模块中选择基于 BDE 数据引擎的数据集组件 TTable、TQuery 或 TStoredProc 对象连接数据库,并执行数据库访问操作。

#### 4.2.3 为客户层提供 IProvider 接口

在远程数据模块中,添加 TDataSetProvider 组件,

将其 DataSet 属性设定为当前数据集。实际上 TTable, TQuery and TStoredProc 都有一个公有属性 Provider<sup>[5]</sup>, 这个属性可通过应用服务器输出。当客户层与应用服务器取得连接后, 就可以得到指定的 TProvider 接口, 客户层与应用服务器之间就可以通过 IProvider 接口进行通讯。

#### 4.2.4 注册和配置服务程序

注册基于 DCOM 的服务有两种方法。一种方法是将创建的 DCOM 服务程序拷贝到安装 DCOM 服务的机器运行, Delphi 6.0 开发软件包中的 STDVCL32.DLL 动态连接库会自动将服务程序信息写入注册表; 另外一种方法是使用系统命令/REGSERVER 程序注册 DCOM 服务程序<sup>[5]</sup>。

分布式系统的数据安全, 特别是试题数据的信息安全和保密是系统设计实现中的一个重要问题。考试机与各级应用服务器通过 DCOM 机制来传送数据。DCOM 的安全机制是以 Windows NT 稳固的内建式安全模块为基础。用户一定要被加入应用服务器管理的用户组内, 才有资格连接服务器。Windows 系统验证用户名, 密码和公钥等必要信息, 确定用户的身份及权限。

当参考人员登录客户层程序时, 需要输入证件号码, 当客户程序连接应用服务器时, 客户层将获取本机的网卡号, 一同提交应用服务器端进行身份校验。Windows 系统会提供对 DCOM 的安全控管, 保证有权限的客户层程序才能够激活和调用 DCOM 服务。存储在数据库服务器上的数据信息是经过 DES16 位加密后的数据。各级应用服务器查询得到加密后的数据, 通过 DCOM 机制传给各考试机, 经客户层解密后, 才可显示在界面上。同样的, 参考人员考试的结果将在客户层将进行打包, 加密后传给各级应用服务器, 由应用服务器解密后, 根据业务逻辑完成自动判卷及审查。

#### 4.3 考试机客户程序的设计与实现

在分布式系统中, 考试机客户程序主要完成交互界面、现场管理及提交试卷的功能。客户程序的本地数据模块(Data Module)负责和应用服务器相连并获取数据。数据模块中包含 TDCOMConnection、TClientDataSet 和 TDataSource 组件的对象。其中 TDCOMConnection 是客户层和应用服务器通过 DCOM 协议进行通信的组件, TClientDataSet 充当客户层的数据集组

件, TDataSource 连接数据集和数据显示组件。

##### 4.3.1 连接应用服务器

通过设置 TDCOMConnection 组件的属性来实现与应用服务器连接。根据不同考试机要连接各级应用服务器的主机名, 设置 TDCOMConnection 对象的 ComputerName 属性。其次, 设置 ServerName 属性或 ServerGUID 属性来标识应用服务器。ServerName 是应用服务器 DCOM 的对象名, 即远程数据模块的名字。ServerGUID 属性用于指定 DCOM 对象的全局标识, 它是在应用服务器创建远程数据模块时生成的唯一标识。

##### 4.3.2 设置数据集组件

本地数据模块中数据集组件是 TClientDataSet。它是 TDataSet 类的子类<sup>[5]</sup>, 从 TDataSet 继承了对数据集的基本操作如插入、删除、查询等。设置它的 RemoteServer 属性为 TDCOMConnection 对象, 设置 ProviderName 属性为 IProvider 接口名称, 即服务程序远程数据模块中的 TDataSetProvider 对象的名称, 这样客户程序就可以通过 IProvider 接口与应用服务器进行通信。

## 5 结束语

系统实现了最初的设计目的: 实现考试的网络化、自动化、无纸化, 保证其公正性和高效性。该系统经过专家鉴定验收, 已全面投入使用。

### 参考文献

- 1 李维, Delphi 5. X 分布式多层应用系统篇[M], 北京机械工业出版社, 2000。
- 2 傅强、王占杰、王媛等, 基于 DCOM 的分布式收费系统设计与实现[J], 计算机工程与设计, 2004, (8): 1364。
- 3 左彦忠、盛翊智, 基于 MIDAS 分布式多层系统执行效率的研究[J], 计算机应用, 2001, (5): 85。
- 4 Delphi 开发方案比较. [EB/OL] [http://www.pcdog.com/p/html/20041218/181220045743\\_1.htm](http://www.pcdog.com/p/html/20041218/181220045743_1.htm), 2004-12-18/2005-4-13
- 5 Dan Miser. Creating Multi-tier Applications with MIDAS. <http://homepages.borland.com/dmiserDL210.htm> 2005-1-1/2005-4-13