基于RDS与MTS的N层事务性Web数据库应用程序的构造与实现

The Construct and Implent of N tier Transaction Web Database Application Based on RDS and MTS

摘　要：基于RDS与MTS的N层Web应用系统使用户在客户端能对大量数据进行复杂的交互式处理。而将定制的事务逻辑组件组织到Windows MTS事务服务中,就可以充分利用MTS服务器所提供的事物处理服务功能,以建立高效、安全的N层Web应用系统。本文阐述了基于RDS和MTS服务的N层Web应用系统的构造及其运作机制,并结合实例介绍了它的具体实现方法。

关键词：远程数据服务(RDS)　组件　MTS　Web数据库应用程序　事务

习胜丰　（益阳湖南城市学院信息与计算机科学系　413000）

随着Internet与Intranet的发展，三层(N层)结构的Web应用系统把业务逻辑独立出来,组成一层或多层,这样就形成了客户端的客户界面层(Browser)、中间业务逻辑层(可细化为多层)和后端数据库服务器层。它具有可伸缩性和满足Internet/Intranet应用的特点。基于远程数据服务(RDS)的多层Web应用系统得客户端能调用服务器端用以实现Web应用系统的业务逻辑组件，通过使用Microsoft的事务处理服务器(MTS)将业务逻辑组件组织到MTS组件包中,开发人员就可以充分利用MTS所提供的基础设施以建立N层事务性Web数据库应用程序且有很多特性有助于使基于组件的N层应用程序规模可变，这些特性包括数据库连接池、自动线程池、事务管理与对象实例管理。使用微软的N层Web结构时，分隔单元可能是COM组件，要建立组件来包含显示、业务与数据访问逻辑，可以用很多种流行的语言（如Visual Basic，J++与C++）来开发与使用组件。N层结构的代价就是相对增加了应用程序复杂性。MTS的目的是要减少这种复杂性。本文以RDS与MTS为基础，通过开发COM组件构建了基于RDS的3层Web应用程序。

1　基于RDS的三层Web结构

基于RDS的三层Web结构采用了基本的两层结构并通过增加服务器端组件来扩展它，这些服务器端组件是包含有非可视化商业逻辑的COM组件，它们可以是非事务性的，或者是事务性的。事务性组件安装在MTS下，它管理事务并处理多线程、安全性、数据库连接池等，三层结构分为Presentation Layer（PL，第一层）、Business Logic Layer（BLL，第二层）与Data Access Layer（DAL，第三层）或Data Service Tier。Microsoft的基于RDS的三层Web结构如图1。

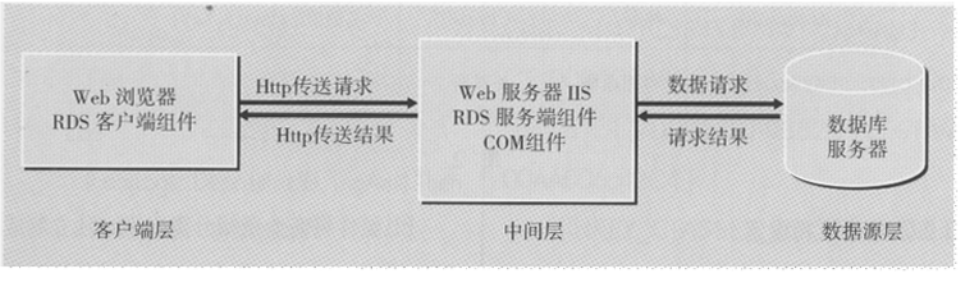

图 1　基于RDS的三层Web结构

三层结构具有扩充容易、分工合作，组件可重复使用以及安全性比较强等特点。

2　RDS的工作原理及运行机制

RDS是ADC(Advanced Data Connector)的新名称,它是ADC和ActiveX Data Objects(ADO)集成在了一起,同时采用与ADO相同的方式在编程模块内部提供远程数据。RDS将连接数据库和组织数据发布等技术引入Web应用程序中,这就大大方便了基于Web的应用程序的设计、编码和实施。RDS技术的关键在于其三层客户端/服务器端模式。这种处理方式将各种客户端/服务器端组件分成三层。客户层可以是本地计算机上支持Web远程数据操作的浏览器,也可以是独立的编译好的终端应用程序。中间层是一台NT服务器,服务器上包含了一些封装着事务流程的组件。中间层既可以是运行在IIS上的ASP程序,也可以是不基于Web的编译好的可执行应用程序。数据源层是数据库服务器。RDS是一种数据管理技术,它使用称为RDS.DataControl对象的客户端组件与服务器端组件RDS.DataFactory 对象进行记录的交互，从ODBC数据库中获取数据，使用由RDS.DataFactory对象返回修改过的记录来修改数据库。RDS.DataControl对象和RDS.

DataFactory对象可通过http、https、DCOM、In-Process COM中的任何一个协议进行通信，可通过RDS.DataControl对象的Server属性指定具体使用哪一个协议，RDS.DataControl和RDS.DataFactory对象间交换的RecordSet被处理为表格数据图的特殊格式。RDS对象的处理对应用来说是透明的。

RDS.DataSpace对象是负责与服务器进行通信的客户端对象，它能提供一种方法从客户端创建服务器组件，同时允许在客户端和服务器之间传输数据，在Web页面上创建一个DataSpace对象有二种方法：

（1）<object classid="clsid:BD96c556-65A3-11D0-983A-00C04Fc29E36"
id="dspdataspace" Height="0" width="0">
</object>

（2）Set dspdataspace=Createobject ("RDS.dataspace")

两种方法的主要区别在于：使用<object>标记时，RDS.DataSpace只在页面加载时才创建，使用脚本技术意味着当脚本运行时就创建了RDS.DataSpace对象，如想延迟对象创建以使页面加载得快一些，那么可以使用脚本技术，一旦创建了RDS.DataSpace对象，就可用它的Createobject方法创建服务器端组件对象。

Set var=dataSpace.Createobject（Progid，Connection）

其中Progid是希望创建的对象的ID，Connection是Web服务器的URL地址。

3 事务服务器MTS及其编程

MTS就是微软用于COM组件的分布式事务服务器，MTS将TP（Transaction Processing）监视器的规模可变能力与可靠性ORB（Object Request Broker对象请求代理）的方便和灵活性综合起来，ORB是处理分布式对象之间的通信细节的中间物，同时，它提供了低层的基础结构或者说是规模可变的应用程序所需要的“管件（plumbing）”：使用无状态的对象、数据库连接池、对象实例管理、简化的组件管理以及更多的功能支持事务处理。

3.1 MTS的工作原理

MTS是Windows NT 4.0 Server的微软事务服务器，它有组件事务管理器、组件管理器与安全管理器的功能，在管理事务的过程中，MTS首先检测组件是否参与事务处理，了解组件的事务处理需求是什么。一些组件忽略事务处理的过程，而一些组件应该或是必须参与事务处理。一个组件开发和部署时，开发人员可对这个组件的事务处理参数进行设置。如果设置的话，组件服务利用这个信息决定组件应该如何参与事务。除了管理组件与事务交互，MTS+还可以管理组件本身，可以使用Component Services Explorer来管理COM对象和简化编程。

3.2 MTS编程

MTS提供了一些基础结构特性，这些特性对建立基于组件的规模可变应用程序是必要的，这些特性包括数据库连接池，自动线程管理，进程隔离和改进的可管理性。

3.2.1 事务的根对象

一个事务可以征用一个或多个组件，如果这些组件的任何一个失败了，整个事务就失败了，并且所有的动作均被取消，因为所有事务都必须有一个开端，启动一个新的事务的对象称作“根对象”，这个根对象也可以创建和使用其他COM组件，而它们卷入根对象的事务要取决于新对象的事务支持设定。

3.2.2 事务支持设定

所有放入MTS中的组件都有事务支持设定，事务支持设定向MTS指明了对象自身将会如何卷入事务中，它有下面四种设定方式：

（1）Require a Transaction（请求一个事务）

（2）Require a New Transaction（请求一个新的事务）

（3）Supports Transaction（支持事务）

（4）Does Not Supports Transaction（不支持事务）

3.2.3 设计用于MTS的COM组件的技巧

（1）尽可能地减少使用对象状态，这是MTS最大化对系统资源的使用的第一步。

（2）当不需要状态时，在每个方法的结束处置显示地调用SetComplete（或必要的话调用Set Abort），这将在从该对象的方法返回时，使该对象停用并释放系统资源。

（3）避免使用属性，而代之以将必要的信息以参数的形式传递，这减少了网络回路路程，并且在使用DCOM时特别重要。

（4）为利用连接池，尽可能迟地取得数据库连接，而一旦用完了就尽可能早地释放它们。

4 开发实践

三层结构不见得一定要搭配MTS，但为了让Web应用程序具有“事务”的处理能力，将“事务“交由MTS来管理能提高效率，本文以银行存款与转帐为例研究基于RDS与MTS的三层结构的Web应用程序开发。

4.1 设计SQL的数据表Account

4.2 设计BLL（第二层）组件

字段名称	含义	数据类型	说明
Accountid	帐号	Char(8)	主键值(primary key)
MyAmount	可用余额	长整数	

图 2

BLL组件将商业逻辑分离出来，独立制成COM组件。本例中将“存款”与“转帐”作成COM组件，它们的Progid是“Mybank.

Mylogic"，提供二种方法为Deposit()（存款）与Transfer()（转帐），存款前要先检查上网者输入的帐号是否存在，然后连接数据库，增加可用余额，而转帐比存款多二种操作，它要先检查来源帐号的可用余额是否小于转帐金额，然后扣除该帐号的转帐金额且加入目的帐号中，这一切要求具有"事务"的功能。

4.3 设计DAL（第三层组件）

同一个VB的项目可以包含一个以上的类（class），作者将所有和数据库存取相关的操作打包到第二个组件，名为"Mybank.Mydata"，位于第三层结构中，与第二层BLL的"Mybank.Mylogic"组件合并后，编译、连接"Mybank.vbp"项目文件后，会在注册表（Registry）里注册二个ActiveX组件。在"MyBank.Mydata"组件中，检查可用余额用CheckAmount()函数表示，检查帐号用CheckID()函数，处理转帐金额用ProcessAmt()函数。

4.4 利用RDS设计PL（第一层）RDSCOM.Htm文件

Rdscom.Htm文件主要源代码如下：

```
<script language=vbscript>
<!-   Sub Process_Data()
 Dim objLogic, objDS, strType
  Set objDS = CreateObject("RDS.DataSpace")
  Set bjLogic = objDS.CreateObject("MyBank.MyLogic","http://127.0.0.1")
 If intType = 2 Then
  If objLogic.Transfer(srcID, lngAmt, tarID) Then
    Msgbox "转帐成功"
  Else
    Msgbox "转帐失败"
  End If
 Else
  If objLogic.Deposit(srcID, lngAmt) Then
    Msgbox "存款成功"
  Else
    Msgbox "存款失败"
  End If
 End If
End Sub
->
</script>
```

5 关键技术

5.1 将com组件交由MTS管理

编译、连接"MyBank.vbp"项目文件后，会将前三个小节建立的两个COM组件（MyBank.MyLogic和MyBank.MyData）注册到系统数据库中，可按下列步骤将刚才建立的二个组件加入到"MTS服务"中。

（1）打开"控制面板/系统管理工具/组件服务"，展开"组件服务"文件夹，一直到"MTS应用程序"。

（2）在"MTS服务"中按鼠标右键，依次执行"新建/应用程序"命令，在下一个窗口中挑选"创建空应用程序"并输入新的应用程序名称（MyBank），然后点击完成按钮。

（3）用鼠标将用VB开发生成的"MyBank.DLL"拖到上述MyBank的组件文件夹，则刚才建立的二个COM组件已交由MTS管理。

可通过在VB6开发组件设定所需的事务类型。

5.2 修改COM组件的注册信息

默认情况下，以VB编译好的ActiveX组件是允许被该部机器上的程序语言、Script其他对象调用，为了让客户端的脚本也能调用，必须适当改变COM组件的注册信息。将登录文件LaunBank.Reg导入系统数据库中，其中LaunBank.Reg的源代码如下：

```
REGEDIT4 [HKEY_CLASSES_ROOT\CLSID\{47FC5845-894A-11D3-9AD3-0080C838CC14}\Implemented Categories\{7DD95801-9882-11CF-9FA9-00AA006c42C4}]
[HKEY_CLASSES_ROOT\CLSID\{47FC5845-894A-11D3-9AD3-0080C838CC14}\Implemented Categories\{7DD95802-9882-11CF-9FA9-00AA006C42C4}]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\MyBank.MyLogic]
```

注：

（1）每个[HKE_…]开头的那一列要不空行相连。

（2）{47……}这串文字是My Bank.MyLogic的ClassID，如果重新编译MyBank.Vbp工程文件的话，要找出对应于该Progid(My Bank.MyLogic)的ClassID，然后改过来。

6 程序测试

将前述研制的二个组件"MyBank.MyLogic"与"Mybank.MyData"编译、连接并在Windows环境中注册并将其纳入MTS管理并加入"事务"设定支持以后，在操作系统为Windows NT 4.0 Server、Web服务器为IIS4.0环境下，将MSDTC服务启动以后并将RDSCOM.Htm文件放在Web根目录(C:\inetpub\wwwroot)下，在浏览器地址栏输入http://127.0.0.1/Rdscom.htm并按回车后，它的"事务"功能视当时的条件，可能是下列几种情况之一。。

（1）存款成功 "来源帐号"存在于Account数据表，而"转帐金额"也大于0。

（2）存款失败 Account数据表元"来源帐号"，或者"存款的操作因故失败"。

（3）转帐成功"来源帐号"、"目的帐号"皆存在，"转帐金额"大于0，且来源帐号的"可用余额"大于等于"转帐金额"。

（4）转帐失败 前述条件有一个不符，或者调用的DAL组件因故失败。

如想观察所有已经确认(Committed)或终止(Aborted)的事务数，可在组件服务→我的电脑→事务统计中察看。

下转第32页 >>

上接第 35 页

7 结束语

通过使用RDS’并结合客户端的脚本编程，开发人员就可以充分利用MTS所提供的基础设施以建立N层事务性Web数据库应用程序。由于缺省的商业对象(RDS Data Factory)并不包含有效性验证、商业规则逻辑等，因此在实际过程中一般不使用缺省的商业对象而需要开发专用的商户端无法得到该信息，提高了系统的安全性。最后，如果结合事务服务器(MTS)，可以大大提高系统的性能和可扩充性。RDS与MTS的结合可以充分利用客户端的计算能力，减轻Web服务器与数据库服务器的负担，缓解网络拥塞，提高Web应用程序的使用效率，可以预料，基于Web 的N层应用程序开发，将会成为Web开发的热点，具有广阔的实用前景。

参考文献

1 邓亚玲、王新房、潘永湘等，RDS在Web基信息管理系统开发中的应用[J]，计算机工作与应用，2002 (3) 。

2 廖信彦，Active Server Page3 彻底研究 [M]，中国铁道出版社，2000。

3 [美] Paul Thurrott等，潇湘工作室译，Visual InterDev6 技术内幕 [M]，人民邮电出版社，2000。

4 [美]，Nicholas D.Evanse,Ken Miller,Ken Spencer等，希望图书馆创作室译，Visual Inter Dev6.0编程指南 [M]，北京希望电子出版社，2000。

5 习胜丰，VB6中RDS数据绑定技术的实现[J]，交通与计算机，2001(5)。