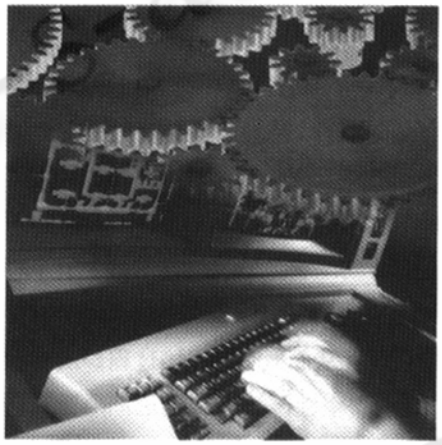
与ASP技术相结合的分布式数据库系统

袁屹 (华北石油管理局呼和浩特炼油厂设计所 010070)
王硕 (中国人民解放军第92330部队通信处 266102)

摘要：本文详细阐述了分布式层状结构数据库在网络环境下的编制要点和ASP技术在访问数据库时的技术要求，并对数据的分布结构和ASP技术的技术标准及实际应用方法进行了重点分析。同时对与之相应的技术规范和运行机制进行了系统阐述，具有良好的通用性。

关键词：分布式数据库 ASP 对象 SSL VPN

1 引言



对离散条件下存储的数据进行基于关系模型的访问和更新是现代大型分布式数据库系统的优势所在。它可以对数据进行跨地域的即时动态存取，能够及时的反映数据状态。关系模型的使用使得用户可以依照抽象的数据结构和逻辑关系与信息进行交互，而对信息存储的具体物理结构忽略不计。分布式数据库系统在跨地域异构物理环境中和关联松散的组织结构条件下，能够对局域信息进行全局共享和交互的特性更使得它在网络环境中被广泛应用。然而它对终端的硬件配置有近乎苛刻的要求，对终端所运行的操作系统和软件环境也有严格规定，同时对用户的操作技能要求较高，一般用户不经专业培训难以熟练操作，因此限制了系统的大规模应用。而且系统日常运行的资源开销较大，难以维护，客户端的故障可能会危及在服务端运行的数据库系统本身的正常运转，因此要求维护人员技术素质较高。

Microsoft公司提出的ASP(Active Server Page)技术，能够在低带宽多传输介质的广域网络环境中进行有效的数据交换和互动，对服务端的资源占用较低。用户在终端只需利用普通网络浏览器通过封装在Web页面中的ASP命令既可对远程信息进行调用而无须了解信息的物理位置，存储结构和获取信息的具体物理路径。用户调用的只是服务端信息的镜像复制，对其所做的所有客户端操作都不会直接影响到在网络中存储的信息本身的安全性。

因此，将分布式数据库系统与ASP技术结合使用将极大的改善系统用户的应用环境，有利于数据库系统的健康运行和体系扩展。本文将就客户端运行ASP页面而服务端运行数据库服务器的浏览器-服务器模式中数据库系统建设和与之相应的ASP对数据库的访问方法提出技术要点及注意事项，仅供参考。为保证系统的通用性，数据库选用Microsoft公司的SQL数据库系统。

2 浏览器-服务器模式简介

在B-S模式中，客户端对服务端运行的程序并不是直接调用而是通过Web服务器进行协调，由Web服务器向服务端程序转交客户端的调用申请，当服务器程序允许调用时，将复制一份客户端副本经Web服务器反馈回客户端并在服务端保留响应的申请记录。数据库系统只在服务端监控各个副本的运行情况并随时响应有效的客户端申请和操作。客户端只是对服务端程序所复制的客户端副本进行本地解释和操作，任何操作响应都是在客户端由副本完成的，当副本向服务端程序申请数据更新或服务端程序因数据更新而对整个网络中的客户端进行更新广播时，系统只将更新的部分经由Web服务器与客户端副本进行交互，而未更新部分则保持旧有的运行状态不变而不被交换，因此网络中传输的只是被更新的信息，从而使通信总量大为

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

降低，当用户退出数据库系统时客户端副本自动丢弃，服务端程序删除副本的相关申请记录并终止监控。在对数据的操作过程中，客户端与服务端始终是由Web服务器隔离的，这样不会因某个客户端的系统崩溃或错误操作而影响整个应用系统的正常运行，从而增加了对数据操作的技术安全性。

B-S模式的开放性还允许用户在网络中的不同位置和不同客户端上随时申请对信息的操作，而无须担心因在网络中所处物理位置的变化或所使用的终端不同而被客户端程序拒绝访问。用户也无须了解数据库系统在网络中的运行环境和存储结构，用户只须拥有有效权限就可以访问服务端程序，而这种授权机制与网络登录的授权机制是可以无缝连接的，因此用户成功登录后其在网络中的逻辑位置将是透明的。

B-S模式中网络的典型拓扑结构为：各客户端经由Web服务器向服务端程序提出访问申请，Web服务器经相应的ODBC(或JDBC)连接向服务端程序转交申请，服务端程序允许访问后将数据交给Web服务器，再由Web服务器将数据和页面相结合生成客户端副本，再送达客户端(如图1)Web服务器与服务端程序可以在一台物理服务器上，也可以分处不同物理服务器上，这样客户端、WEB服务器、服务端程序三者可以分别进行维护管理，从而减少了监管工作量和技术难度，并增加了系统的安全性。

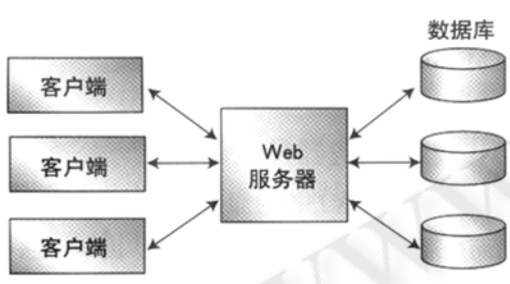


图 1 浏览器－服务器模式示意图

3 网络数据库的技术实现

3.1 网络数据库结构规划

因为网络数据库系统是基于Web环境进行分布式部署的，因此数据库架构可以是开放式的分层网状结构。数据库系统可以按功能分为几个子数据库，各数据库又按涉及内容不同分成一系列数据库。数据分处在物理位置不同，逻辑关系上相互独立的各数据库表中，在数据表间以统一标识的数据指针进行标识和连接，这个数据指针可以是标识索引，也可以是统一编制的键值，但必须是在整个数据库系统中命名唯一的且是严格一一对应的。在数据库本体中的数据结构以分层的树状结构为基础，同层的数据间以事先确定的变量标志进行联结，这个变量标志也必须是唯一的。至于数据库的数据结构和数据库表的组织方式可以根据实际情况灵活处理，这样就实现了数据库系统在物理上的分布式部署和在逻辑上的集约管理。示意图如图2。

3.2 Web服务器和数据库连接

用户使用Web站点实现对数据库的操作，因此Web服务器和数据库服务器之间必须实现建立数据关联，这一般是通过ODBC(opened database connection，开放式数据库连接)实现的。在Web服务器上配置DSN关联信息，建立对具体数据库的连接和访问用户名、访问密码等信息。如果Web服务器不会有经常性的逻辑位置调整的话，可以使用系统DSN配置，可以获得较高的效率。但为了保证整个系统的鲁棒性，最好采用文件型DSN配置，这样无论Web站点位于网络的什么物理位置，只要经过正常发布，用户在访问网站时就可以通过解释DSN文件信息获得与数据库服务器的连接。无论系统型还是文件型DSN配置，用户都不能介入(同时也查觉不到)与数据库的连接过程，所有数据交互都是在Web服务器和数据库服务器之间隐性进行的，由此保证了数据库的安全。

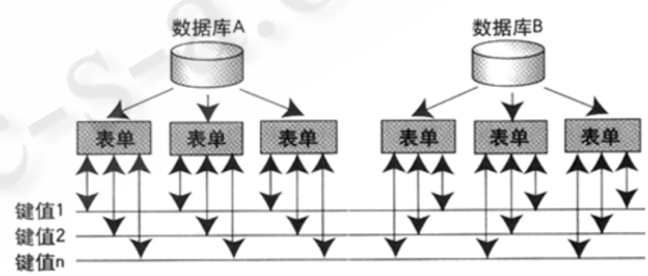


图 2 数据库系统中信息连接关系示意图

3.3 利用ASP技术对数据库的操作

用户必须以HTTP方式调用内嵌ASP命令的Web页面，只有登录到正常运行的网站上之后才可以使用ASP功能。Web服务器首先必须与数据库建立连接，成功后按事先设定遵照用户发出的正确有效的命令发出对数据库的操作申请，数据库响应申请后将结果反馈到Web服务器，Web服务器再将获得的反馈信息与特定的HTML页面结合并显示在客户端的网络浏览器上。所有这一切都必须经

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

Web服务器来协调，用户是无法干预的。

3.3.1 与数据库建立连接

ASP通过ADO(ActiveX DataBase Object)的Connection对象对数据库进行逻辑上的连接，在对数据库进行访问时，ASP首先在本地创建一个Connection对象，然后将其连接到具体的数据库服务器上。

语法：<% Set Connectionname=Server.Creatobject ("adodb.connection")

Connectionname.open "DataBasename"%>

在对数据库结束操作后与服务器断开前必须关闭所有连接，否则可能会导致数据库因客户端的非正常退出而堆积大量无宿主连接，甚至由此导致数据库的崩溃。关闭连接命令一般放在ASP操作命令的末尾。

语法：<% Connectionname.close%>

3.3.2 对数据库进行操作

在建立了与数据库的连接后，ASP通过执行封装在其内的SQL命令来对数据库进行操作，由于操作产生的执行结果往往是数据的记录集，因此ASP通过RecordSet对象操作数据集，RecordSet对象可以独立于数据库连接来创建，但ASP必须定义如何连接特定的数据库，如何提交到RecordSet对象中以及初始化对象的有关属性值，而且每进行一次数据库操作，ASP都要建立一个RecordSet对象，这样在与网络数据库的连接通道中同时打开的对象对话会很多，极大的增加服务器的负担，因此这种方法并不足取。更有效的方法是在现有的连接中建立RecordSet对象，这样一旦建立一个数据库连接就可以在整个运行期间对该数据库进行多次操作。具体的讲是使用Connection对象的Execute方法执行SQL命令。此时ASP自动将原有的connection对象重新定义为一个Recordset对象并将其作为数据库对用户操作做出响应所产生的反馈信息在客户端的存储对象，而原有的数据库连接关系会保留到与数据库的对话完毕。

语法：独立于现有连接而单独创建

<% Set Objectname=Server.Creatobject("adodb.recordset")

Objectname.open "Databasename"

sql command %>

在已有连接中创建

<% sql="sql command"

Set Objectname=Connectionname.Excute(sql) % >

3.3.3 在客户端显示数据的执行结果集

RecordSet对象是以栈表示方式显示数据的，因此可以很方便的将数据封装在返回客户端的HTML页面中。另外它还允许用户通过移动术语称之为光标的数据指针来向前或向后浏览结果集。

语法：<% do while not Objectname.eof%>

<tr>

.....

<td> <%=FormatCurrency(cdbl(Objectname(listname))) % > </td>

.....

</tr>

<% Objectname.movenext

loop %>

3.3.4 在客户端显示多媒体数据结果集

一般情况下数据库将多媒体格式的数据转换成二进制格式存储在数据库本体中，当多媒体信息被客户端申请调用时再将其还原为原来格式输出，这中间必须经过专门的工具来完成格式转换，服务器的运算负担极重而且效率低下，数据库本身也消耗了大量宝贵的存储空间记录这些数据。因此对网络环境下远程服务和数据库系统的维护要求较高，难以满足分布式数据库系统的运行要求。

ASP技术支持各种网络上通用的多媒体格式，并且可以通过内嵌ActiveX控件来提供对其他多媒体信息的支持。因此可以在创建数据库时，对多媒体信息建立基于全局环境的统一的数据指针来标识多媒体信息在网络中的存储地址和存储格式，再将数据指针封状为数据库中独立页的列(或表单中的字段)。多媒体信息并不直接存储在数据库本体中，而是在遵循分布式标识原则基础上存储在网络的任意位置，而数据库只需维护相应的数据指针标识出它的逻辑地址。ASP与数据库建立连接后，在结果集的客户端存储对象中调用相应列(或字段)中的数据指针内容，就可以访问到所选多媒体信息并将一个镜像文件复制下载到客户端的网络浏览器上。

语法：在数据库中定义多媒体数据的存储地址为

dataurl=http://www.website/saved path/dataname.datatype

其中dataURL定义为数据库表单中的一个列或一个字段。

在ASP中显示数据

<%=Objectname(dataurl)%>

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

4 分布式数据库系统的安全性支持

数据库系统的安全性要求较高，现已拥有多种方式不同但非常完善的用户认证和访问权限检验机制。当用户发出访问申请时，ASP首先是在服务端被解释和执行，客户端只接受执行后的结果而看不到源文件，因而无法修改ASP源代码，这样保证了数据库服务器中存储的数据是安全有效的而服务器本身的安全运行也不会受到客户端的影响，同时因为ASP运行在服务端，因此所有的用户申请都必须先提交到Web服务器，再由Web服务器将申请转换成各种标准的安全机制格式与数据库服务器进行协调，这样，安全机制的监控范围被限制在服务器与服务器之间，用户无法介入安全认证操作，从而保证了数据库服务器的运行安全。

4.1 用户身份验证

正常情况下，用户在申请访问数据库时，ASP页面首先询问用户登录数据库所需的用户名和身份号码等认证凭据，用户填写完毕并提交服务器，ASP将用户填写内容与数据库中所记录的有效用户登记信息进行比较以确定用户的有效性，这一切都是在服务器的安全防护机制监控之下完成的。

例如：假设数据库要求验证用户的用户名(username)和身份号码(userid)。

ASP对用户提交的信息进行操作

```
<% Set Connectionname=Server.Creatobject ("adobd.connection")
Connectionname.open "DataBasename"
sql="select * from registerlist
where username='"&Request.form("username")&"'
and userid='"&Request.form("userid")&"'"
Set test=Connectionname.execute(sql)
If test.eof Then
test.close
Connectionname.close
Else
.....
End If %>
```

4.2 数据传输中的传输信道加密

在用户向Web服务器提交访问申请时，可以对申请内容进行加密，常用的方法是SSL(secure sockets layer)加密，首先客户端向服务器发送本地端支持的密匙组合，服务器接受密匙后选择出自己所支持的密匙，并向客户端发送自己所拥有的证书，其中包括一个公匙，客户端接到公匙后开始把本地产生的一个随机传输密匙用公匙加密并送到服务器以便确认，此时客户端和服务器“握手”成功，双方开始传递加密后的数据(如图3)。这些都是在创建数据库系统时所做的安全防护机制协定，一旦系统部署完毕，安全防护机制对用户而言将是安全不可见的。

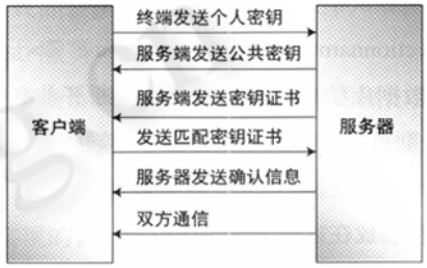


图 3 SSL加密示意图

4.3 数据传输中的网络结构加密

用户使用网络数据库时必须登录到网上，因此在网络拓扑结构上可以采取许多技术措施，加强对网络中传输数据的加密防护。常用方法是利用VPN(virtual peculiar network,虚拟专用网络)技术，在特定用户和数据库服务器之间通过路由器建立tunnel通道，再在路由器上进行配置以便对tunnel中传输数据进行加密(一般使用hash算法，它具有反向不可恢复的加密特性)。配置VPN时最好使用由用户发起的VPN连接。路由器先在普通的完整数据报基础上再封装一个特殊的数据报头，然后通过跨越公用网络的tunnel向对方传输，这样在网络中传输的数据为加密后的信息，别人在不知道具体的加密私钥时即使截取了数据报也是无法解读数据的。同时tunnel地址和实际网络地址并不相同，相互间只了解对方的tunnel地址而不了解对方的饿实际网络地址。由此可以实现本单位内部网络与外部其他特定单位网络的网络地址相互隔离，实现对特定端口的实时管理、监控和网络拓扑结构的相对独立和隔离。■

参考文献

1 *《Microsoft SQL Server 6.5 程序员指南》(美) Microsoft 著，科学技术出版社 & 龙门书局 1997年版。*

2 *《开放式数据库互连ODBC方案集粹》 (美) Robet Signore 等著，电子工业出版社，1995年版。*

3 *《Visual Interdev 6.0开发指南》(中) 任伟等著，北京航空航天大学出版社，1999年版。*

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn