

利用 RDS 访问数据库

华中理工大学控制科学和控制工程系
周茜 田忠和

本文基于对利用ADO 访问数据库的弊端的分析,提出了利用RDS 访问数据库的方法,介绍了RDS 的组件和工作原理。

随着 Web 技术的发展,基于 Internet/Intranet 的 MIS 成为当今数据库系统的主流。在一般的系统中,所有对数据库的访问操作都是由 Web 服务器上的 ADO (ActiveX Data Objects) 对象来完成。为此,客户端必须显式地将全部表单提交给服务器。在这种情况下,当用户频繁、大量地读取数据,或者多用户同时访问数据库时,服务器的负载会加重;同时,由于网络带宽的限制,客户端的浏览速度会明显地变慢,客户端的等待时间会相应地加长;另外,编程者还必须书写额外的服务器脚本以维护应用程序和会话的状态。然而,如果采用远程数据服务(RDS Remote Data Service)来代替 ADO 访问数据库,则可以在客户端直接控制对数据库的各种访问操作,从而极大地减少客户端和 Web 服务器之间的信息往返次数,减轻了服务器的负载,同时加快客户端查询和浏览数据库记录的速度,减少客户端的等待时间。

RDS 的工作原理

RDS 是 ADC (Advanced Data Connector) 的新名称,它是 ADC 和 ActiveX Data Objects (ADO) 的集成,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 Internet/Intranet 应用程序中,这就大大方便了基于

Web 和基于 LAN 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。图 1 说明了基于 Web 的 RDS 应用程序如何处理用户显示数据库信息的请求。

一个 RDS 应用程序分为三层。客户端组件一般包含在 IE 中,通过 HTTP 和服务器端组件联系。一旦用户发出请求,客户端 RDS 组件就向 Web 服务器发出查询请求。服务器端的 RDS 组件处理这个请求,并且传给数据库服务器,数据库服务器响应请求,然后传回给 Web 服务器所请求的数据。服务器端的 RDS 组件又将这些数据转换成 ADO Recordset 对象,通过网络传回给客户端。最后请求的数据便显示在如文本框、表格等可视化数据控件中。这些控件已经通过 RDS.DataControl 对象与数据绑定。客户端 RDS 组件还将得到的结果集放在高速缓存中,这样可以减少和 Web 服务器的连接次数,使得用户操纵数据非常迅速和简单。只有当用户对数据进行了更新操作或者请求新的数据时,RDS 组件才会再次向 Web 服

务器发出请求。

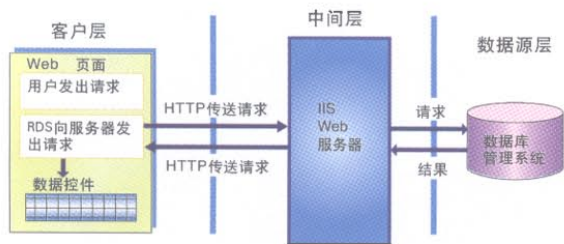


图 1 基于 Web 的 RDS 应用程序的工作原理

客户层主要是提供可视化界面来显示数据和组织数据的布局。一般在基于 Web 的 RDS 应用程序中，客户层包含下面组件：

(1)包含RDS.DataControl 对象和数据控件的Web 页面。RDS.DataControl 对象实际是一种机制，它将查询结果显示在可视化的 Web 页面的控件中。

(2)事务对象代理 (Business object proxies) 和 RDS.DataSpace 对象，这两个对象起着连接中间层组件的作用。

(3)客户端数据缓存区。客户端数据缓存由 ADO 和客户端光标引擎组成。客户端光标引擎是一套数据结构，支持客户端光标定位功能。它可以高速缓存从 DBMS 检索的数据，同时允许客户修改这些可见的数据。对于用户而言，客户端光标引擎是透明的，RDS 通过它自动完成各种操作，如导航记录、对记录进行排序和过滤、更新记录等等。图 2 说明了客户端组件是如何工作的。



图 2 客户端组件工作原理

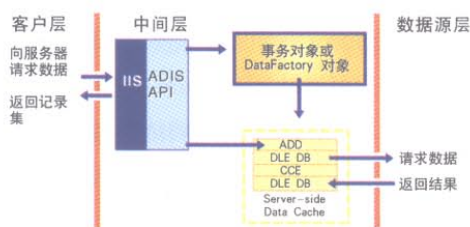


图 3 中间层组件

中间层是客户端计算机和数据库服务器的中间桥梁。中间层的组件负责接受用户端通过 HTTP 传来的请求，然后又将这些请求通过 OLE DB 供应接口发送到数据源。中间层通常包含 IIS 和 ADISAPI (Advanced Data ISAPI)、事务对象、服务器端数据缓存。事务对象是服务器端的关键组件，其中包括了应用程序逻辑、数据逻辑以及访问数据库的代码。事务对象可以是任何 VB、VC++ 等高级语言创建的自动对象。图 3 说明了 RDS 的中间层所包含的组件。

ADO 和 RDS 的比较

RDS 最重要的特点是客户端和中间层可以高速缓存查询所得的数据，从而减少了客户端应用程序和数据库服务器之间的请求次数。这点在通过 Internet 访问数据库的情况下尤为重要。应用程序不需要等待数据通过网络传输。而且 RDS 采用的是断开的 Recordset 对象，这样可以减少 OLE DB 供应接口的竞争，ADO 则是采用和数据库一直保持连接的 Recordset 对象。这一点对比在多用户同时访问数据库时尤为明显。

RDS 特别适合于那些客户需要浏览记录或访问活动数据的应用程序。因为查询的数据可以在客户计算机上进行高速缓存，所以用户可以很快地导航记录。RDS 还适合那些需要得到许多细节信息的用户。而 ADO 则适合于较高级的数据库编程。采用 ADO 可以指明创建那种类型的 Recordset 的对象，这一点比 RDS 的灵活性好。当然，ADO 是执行在 Web 服务器上，所以不需要额外的组件。在创建程序时，应该根据具体情况决定哪一种技术更适合。例如：可以运用自定义的对象采用 ADO 进行数据操纵，同时又采用 RDS 将远程数据传送到用户可以连接的中间层。

下面是利用 ADO 和 RDS 查询数据库时记录导航的过程比较：

(1)利用 ADO 查询数据库

①用户向服务器发出请求。②服务器执行程序，连接数据库，执行相应的查询或运行存储过程，得到一条或多条记录。③服务器程序指向特定的记录。④服务器从特定的记录中摘取数据，按照 HTML 格式写入页面传给浏览器。同时将有关当前记录位置的信息存储在某些全局服务器变量中，随页面一起传给了服务器。换句话说，服务器抛弃了记录集。⑤当用户点击命令按钮“下页”，按钮又将请求信息提交给服务器。此时，服务器和浏览器已经失

去连接,所以浏览器必须传送格外的信息,以告知服务器用户是想浏览下一页记录。同时还必须将当前记录的位置信息又返回给服务器。⑥服务器重新发出查询请求,定位在记录集中的正确位置,然后后移一条记录。重复④-⑤。

(2)利用 RDS 查询数据库

①用户向服务器发出请求。②服务器将请求的页面传送给客户端。③浏览器收到页面后,RDS 控件自动地打开记录集,依据 DHTML 的数据捆绑规范,将数据捆绑在相应的控件上。④进行上下翻页时,直接在记录集中进行。

两者之间最大的不同在于:当服务器将页面传送给浏览器后,浏览器就不再需要向服务器请求,明显地减少了浏览器和服务器的交互次数。另外,因为客户端可以高速缓存记录,导航不需要重新生成记录集,也不用重新生成显示数据的 DTC GRID (表格),所以导航记录的速度很快。当然,服务器上也可以不仅仅保存当前记录的位置信息,而将第一次向数据库查询得到的记录集保存在服务器的高速缓存中。但是如果这样,当多用户同时请求时,会因缓冲容量所限而使服务器的速度变慢。

RDS 应用举例

编写 RDS 的客户端程序,可以采用 VBScript 语言。程序中需要申明一个 ADC 对象,其功能类似于 ASP 中 ADO 的 Connection 对象。

```
<OBJECT classid="clsid:BD96C556-65A3-11D0-
983A-00C04FC29E33"
ID=ADC HEIGHT=1 WIDTH = 1>
</OBJECT>
<% ADC.Server = "HUSTKEJI"           'IIS
服务器的名字
ADC.Connect ="DSN=KJGL"             'IIS 服务
器上的数据源
ADC.SQL ="SELECT * FROM DM_TAB"     '设
置 SQL 语言
ADC.Refresh                          '查询
Grid.Rebind%>                       '显示在表格中
```

目前,微软公司推出的 Visual InterDev 支持一种 SOM 编程环境,在 SOM 环境中可以采用 DTC Recordset 和 DTC Grid 进行客户端数据捆绑。这种客户端数据捆绑实际上是依靠 RDS 的支持。DTC (Design Time Control) 也是可视化控件,编程者只需设置其属

性和方法,就可以自动产生运行时间的执行代码,不必要手工编写程序。采用客户端数据捆绑方式可以提高客户端的查询速度,而且因为在客户端控制各种访问数据库的操作,所以仅仅关联到页面中的某一个 DTC 控件的刷新,而不会使得整个页面被刷新,减少了客户端的延迟等待时间,给人很好的视觉感受。■

参考文献

- 1 《精通 VISUAL INTERDEV 6》北京希望电子出版社, Joseph O'Neil 著,希望创作室译。
- 2 Microsoft Corporation, Visual Interdev Online Document.
- 3 Microsoft Corporation, Internet Information Server Online Document.

