基于XML的数据交换在分布式Web系统中的应用

肖延松　孟　波　熊德林　杨子晨　(武汉大学计算机学院　430072)

摘　要　文章简要介绍了基于XML的数据交换技术和分布式Web系统的体系结构，讨论了基于XML的数据交换在分布式Web系统中的应用，并用实例说明了具体实现方法。

关键词　XML　DOM　分布式Web系统

1　引言

近年来，Web应用程序非常流行，它能适应多用户、多数据且非安全的网络环境，具有良好的灵活性和可扩展性等优点。但是随之而来它也有一些弱点：

(1) 系统的分布式特点大大增加了系统的复杂度；

(2) 应用程序不灵活——Web应用程序常常是针对特定的客户端编写的，客户端需求的改变使你必须重写服务器端程序，或者创建极其类似的新服务；

(3) 新的应用程序常常重复大量现有的代码；

(4) 向自动化Web任务转移时却发现HTML很难保存数据的含义。

XML(eXtensible Markup Language——扩展标识语言)是一种非常适于应用程序之间数据交换的格式，特别是松耦合的应用程序——如：分布式Web系统。作为一种通信协议，HTTP具有跨平台性。对于应用程序数据来说，XML具有同等的协议。XML可以促进应用程序代码的重用，提高应用程序在面对需求和程序变化时的适应能力。

2　基于XML的数据交换

XML是一种界定文本数据的简便而标准的方法，与注重数据及其表达方式的HTML不同，XML只关心数据本身。XML的优势在于其数据可以被用户定义的、有语义的标记环绕，可在数据库中实现无损的存储、检索和修改等操作。只需简单地添加标记就可以描述它们所封装的信息，XML的这种数据描述机制使得它成为一种在Internet上共享信息的强大途径，因为：

(1) 它是开放的，XML能够在不同的用户和程序之间交换数据，而不论其平台如何。

(2) 它的自描述的特性使其对于B2B和企业内部网解决方案来说是一种有效的选择。

(3) 无需事先协调，我们就可以在程序之间共享数据。

同时通过DOM API(文档对象模型API)我们可以轻松地编制读写XML的程序，在Web系统应用程序中可以灵活使用XML。

3　基于Web的分布式三层体系结构

整个系统分成如下三个层次如图1。

客户端，称为表示层。主要功能是接受用户输入和显示运行结果。通常由浏览器(如IE)组成。

应用服务器，也称为应用层。主要功能是将表示层收集的数据进行处理，并将结果返回给表示层。通常由Web服务器和应用模块组成。

数据库服务器，也成为数据层。主要功能是提供数据服务。通常由数据库服务器组成。

如：Sysbase、Oracle、Infomix、SQL Server等。

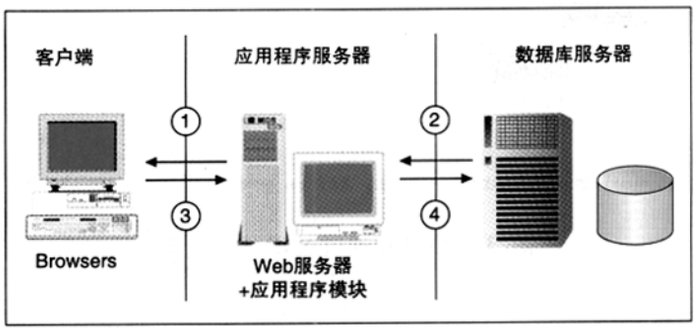

图1　三层体系结构图

说明：

(1) ① Http request ② Sql request ③ Http reply ④ Sql reply

(2) Web 服务器与数据库之间采用 ODBC 连接方式

4 XML 在 Web 系统中的应用

4.1 XML 与数据库

4.1.1 数据存储

数据库与XML对于存储数据提供了互补的功能。数据库存储数据是为了有效的恢复，而XML则提供了容易的信息交换方法，它使得应用之间可进行互操作。依靠用于数据库的DOM API，就可以使数据库看起来像是与数据库模式所衍生的DTD (Document Type Definitions)相关联的虚拟XML文档。可以从XML文档传输数据到数据库，也可从数据库到XML文档。XML可作为数据库之间数据交换的标准。

4.1.2 访问数据库

ODBC是被设计用来在不同的数据库系统上建起一座桥梁，然而拥有XML意味着更加先进。一个应用程序可以直接处理数据库中的信息——如果发生同Internet的通信，它可以使用如图2所示的结构。

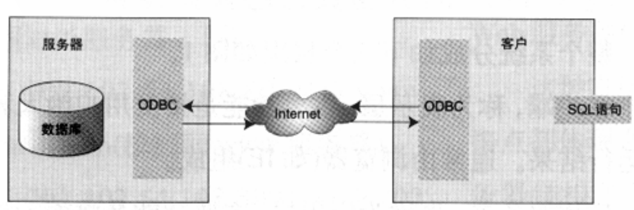

图 2　ODBC 之间的通信

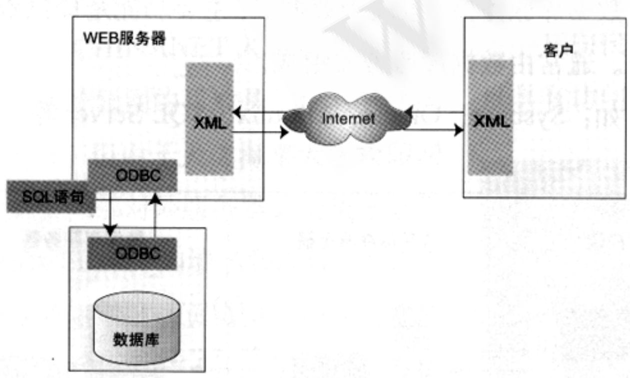

图 3　ODBC 通过 XML 接口上网

然而这种方法有几个问题：

(1) 仅有理解 ODBC 的系统可以接收这个信息。

(2) 很多防火墙不允许 ODBC 的交易。

(3) ODBC 易被黑客攻击。

通过在通信管道的两端加上XML接口，我们去除了客户端对ODBC的依赖(如图3)。同样，如果我们以某种方法封装XML，并且通过80(正常的HTTP端口)传送它，可以解决防火墙的阻挡问题。

采用 Javascript、Delphi5.0 和 ISAPI(Internet Server API)的程序实例如下：

在客户端有如下请求：

```
<XML ID="ROWDATA"SRC="queryCorp.dll/queryCyrs?cyrs='100'"'></XML>
```

```
//queryCory.dll/querycyrs 服务器端程序代码
...
Type
query:TQuery;
...
procedure TWebModulel.WebModulelActionsOAction
(Sender:TObject;
Request:TWebRequest;Response:TWebResponse;var
Handled:Boolean);
var
rs:string;
i:integer;
begin
query.close;
query.SQL.Clear;
query.SQL.Add('select qymc,dz,jyfw from corporation
where cyrs>'''+request.QueryFields.value ['cyrs'] +'''');
query.execsql;
while not query.Eof do
begin
rs:=rs+'<ROW';
for i:=0 to query.FieldCount-1 do
rs:=rs+''+query.Fields[i].FieldName+'='''+query.fields
[i].asstring+'''';
rs:=rs+'/>';
query.Next;
end;
response.content:='<ROWDATA>'+rs+'</ROWDATA>'
end;
```

这个例子允许我们简单地通过在URL中指定一个参

数cyrs(从业人数)而取回从业人数大于其值的所有企业基本资料。返回的XML数据格式将如下：

```
<ROWDATA>
<ROW qymc="公司1" dz="地址1" jyfw="网络"/>
<ROW qymc="公司2" dz="地址2" jyfw="光纤"/>
……
</ROWDATA>
```

其在客户端的显示和操作将在下面讨论。

4.2 XML在客户端的应用

利用数据绑定，在用户界面中显示查询结果。

```
〈H1〉企业资料查询〈/H1〉
〈HR〉
〈XML ID="ROWDATA "SRC="queryCorp.dll/queryCyrs?cyrs='100'""></XML>
<table ID="showtable"DATASRC="#ROWDATA"width="567"border="0">
<THEAD><tr>
td width="20%" align="center" bgcolor="#ccddFF">企业名称</td>
td width="40%" align="center" bgcolor="#ccddFF">地址</td>
td width="40%" align="center" bgcolor="#ccddFF">经营范围</td>
</tr>
</THEAD>
<TBODY>
<tr id="showtr" style="margin-top:-12"bgcolor="#F0F0FF">
<td width="20%">,input type="text" id="T1" datafld="qymc" size="20"></td>
<td width="40%">,input type="text" id="T2" datafld="dz" size="40"></td>
<td width="40%">,input type="text" id="T3" datafld="jyfw" size="40"></td>
</tr>
</TBODY>
</table>
```

利用网页中嵌入数据源对象(DSO —— Data Source Object)，只需把DATASRC属性的值指定为文档对象的名字(上例中XML的ID参数)，并把DATAFLD的属性的值指定为XML元素的一般标识符即可，这样就可把XML元素的数据内容与HTML元素联系起来，在浏览器中显示应用服务器传送过来的构造良好的XML数据。依靠Internet Explorer5.0内置的XML功能，它将分析过的XML文档以DOM表示，脚本程序可以通过这种模型来访问XML元素的数据内容，并将数据动态地插入到用户界面中，操作十分方便灵活。同样我们可以将页面用户输入数据以XML格式打包通过表单提交传送到应用服务器端处理。

4.3 客户端与多个服务器之间的数据交换

有时侯我们必须访问多个应用服务器以获取数据，如：在由A、B和C地等多个地市级分布式Web系统互联组成的省内联网中，我们想查询三地从业人数大于100的企业，这时可以用脚本程序分解查询请求，分别查询得到XML结果后再将数据源合并即可。

5 结束语

在XML下的Web系统，我们不再局限于基于浏览器的客户端。XML本身就是数据，而且可以由程序任意控制。同样的数据，既可以设定其样式化以便在浏览器中显示，也可以交给一个代理进行后台处理。在这个机制中，XML文档无需假设数据的最终用途。如果客户端需要HTML，由数据驱动的转换过程就会使用XML文档生成HTML页面，而生成XML的底层应用程序不需要任何修改。

网络中的服务器、客户机和应用程序所进行的处理都可使用这种机制交换数据，这种机制扩展起来并不困难，而且能够在运行时自动找出数据的结构。事实上，任何一种平台都支持这种机制，它使用简单，能够处理来自不同数据源的标记数据。应用程序的开发者可以使用来自非传统数据源或其他服务器的数据来满足客户端的请求，Web的开发已经从客户机-服务器计算体系迈向真正的多层模式。■

参考文献

1 Extensible Markup Language(XML) 1.0, http://www.w3.org/TR/REC-xml。

2 Shi-Ming Huang, Developing an XML gateway for business-to-business commerce, IEEE, 2000。

3 Didier Martin, Professional XML, Wrox Press, 2000。

4 Charles F. Goldfarb, Paul Prescod, 张利、王显著译, XML实用技术, 清华大学出版社, 1999。