

基于 Web 的三层 C/S 模型及其实现

祝建中 郑瑜钒 (杭州师范学院信息工程学院 310036)

摘要: 本文介绍了基于 Web 的三层 C/S 模型, 在此基础上着重讨论了如何利用 Delphi 开发基于 ISAPI 技术的 Web 服务器应用程序扩展的关键技术。

关键词: 三层 C/S 模型 ISAPI Web

1 引言

在二层 C/S (Client/Server) 体系结构中, 客户端实现用户界面和应用逻辑, 服务器端实现数据库系统的管理。与基于主机的体系结构相比, 这种二层结构具有许多优点, 但随着系统

规模的不断扩展和应用复杂程度的增加, 也逐渐显露出以下不足:

(1) 由于应该由各个客户端共同遵守的系统应用逻辑只能在客户端程序中实现, 导致客户端庞大, 适应系统应用逻辑调整的灵活性降低。

(2) 因为客户端与数据库服务器的连接只能在各个客户端单独配置和维护, 当客户机数量增大时, 系统的维护管理将变得困难。

多层 C/S 体系结构较好地解决了这些问题。多层结构的基本思想是在客户端和数据库服务器端之间增加适当得中间层, 将系统应用逻辑以及与远程数据库服务器的连接封装在这些中间层中, 从而避免了二层体系结构中由各个客户端程序单独实现应用逻辑而产生的冗余, 使得客户端程序不仅瘦小, 而且因不用涉及数据库连接配置等设置问题而更容易发布和安装; 应用逻辑在应用服务器上实现, 减少了网络数据的传输量, 降低了网络的负荷。

一个典型的三层 C/S 数据库应用程序通常划分为以下三个层次:

① 客户端程序 (Client Application): 提供客户端的用户界面。

② 应用服务器 (Application Server): 向所有的客户端提供公共的数据服务 (实现公共的应用逻辑, 实现与远程数据库的连接和访问等)。

③ 远程数据库服务器 (Remote Database Server): 提供关系式数据库管理系统。

2 基于 Web 的三层客户机 / 服务器模型

Internet 的 WWW 服务采用的是浏览器 / 服务器 (B/S) 模型, 如果只是从 Web 服务器返回静态的页面, 则它实质上也是一种二层的 C/S 模型。一端是 Web 客户, 另一端是 Web 服务器, 客户通过浏览器向 Web 服务器发出 HTTP 请求, 服务器响应客户的请求并把 HTTP 响应的结果返回给客户。

但在许多实际的 Web 应用中, 需要 Web 客户与 Web 服务器进行交互, Web 服务器能够根据客户的请求访问数据库服务器, 并把结果以动态页面的方式反馈给客户。这就需把三层 C/S 模型与 Web 技术结合起来, 使 Web 服务器同时具有应用服务器的功能, 建立如图 1 所示的基于 Web 技术的三层结构 C/S 模型:

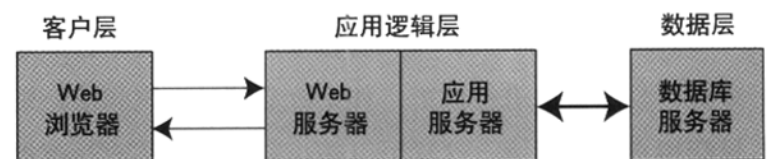


图 1 基于 Web 的三层 C/S 模型

第一层: Web 浏览器, 实现客户界面。

第二层: 兼有应用服务器功能的 Web 服务器, 响应客户请求, 实现应用逻辑, 完成对数据库的连接和访问。

第三层: 数据库服务器, 实现数据库的管理。

在这种基于 Web 技术的三层结构 C/S 模型中, 客户

端由于统一采用 Web 浏览器 (IE 或 Navigator) 而成为真正的瘦客户, 系统的应用逻辑需要调整时, 只要对 Web 服务器进行升级, 对客户端和数据库服务器几乎没有影响, 从而极大地提高了系统的灵活性和可维护性。

3 基于 Web 的三层客户机 / 服务器的实现

实现上述模型的关键在于使用 ISAPI 对 Web 服务器的功能进行扩展, 使之同时承担应用服务器的职能。

3.1 ISAPI 的基本结构和工作机理

在 Delphi 环境中, 利用 Microsoft 的 ISAPI (Internet Server API) 可以方便地实现对 Web 服务器的扩展。ISAPI 是以 DLL 的方式在第一个客户请求服务时加载到 Web 服务器的进程空间, 从而成为 Web 服务器软件的一个有机组成部分。一个 ISAPI 的 DLL 可以处理多个客户请求, 多个 ISAPI 的 DLL 可以同时加载到 Web 服务器的进程空间, 这些特征使它比 CGI 具有更高效率。图 2 给出了在 Delphi 5 中利用 ISAPI 对 Web 服务器进行扩展后的结构。

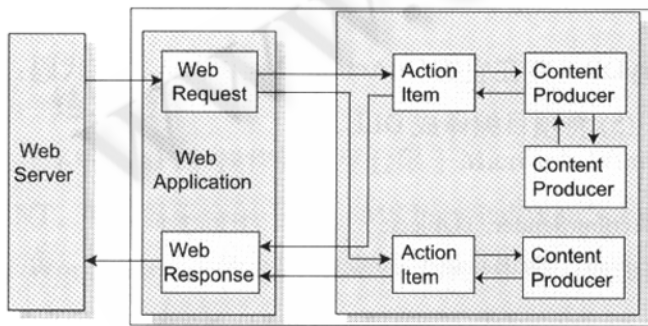


图 2 利用 ISAPI 对 Web 服务器进行扩展后的应用层逻辑结构

Delphi5 建立的 ISAPI DLL 主要由 WebApplication 和 WebModule 两大模块组成。

WebApplication 是 TWebApplication 类的对象, 它的主要功能是管理客户的请求信息和 Web 服务的回应信息。在收到一个客户的 HTML 请求时, WebApplication 创建一个存储这些请求信息的 TWebRequest 对象和一个

准备存储回应信息的 TWebResponse 对象, 然后将这两个对象传递给 WebModule。

WebModule 是 TWebModule 类的对象, 其主要功能是调度控制 ISAPI 执行流及实现应用逻辑。它管理着为实现业务规则及数据库访问而添加的各种对象, 特别要强调的是 WebModule 含有一组 TWebActionItem 对象, 每个 TWebActionItem 对象负责响应和处理一种客户的请求服务, 它的 PathInfo 属性给出了提供这个特定服务的入口, 因此一个 ISAPI DLL 能够处理多个客户的多种请求。

WebModule 在收到 WebApplication 传递过来的 TWebRequest 对象和 TWebResponse 对象后, 首先通过检查各 TWebActionItem 对象的 PathInfo 属性来判定应该由哪个 TWebActionItem 对象来处理 TWebRequest 对象的请求, 然后再把 TWebRequest 对象和 TWebResponse 对象传递给这个 TWebActionItem 对象, 由该对象负责对客户的请求作出具体的处理并形成回应信息经由 TWebResponse 对象的 Content 属性反馈给客户端。

TWebActionItem 对象是通过其 OnAction 事件中的代码完成对客户请求信息的分析处理的, 这里是实现应用逻辑和数据库访问关键所在, 它的主要任务是:

(1) 通过传递过来的 TWebRequest 对象获取客户的 HTTP 请求信息, 其类型由 MethodType 属性来判定是 GET 还是 POST, 其请求数据, 对于 GET 请求从 QueryFields 属性获得, 对于 POST 请求从 ContentFields 属性获得。

(2) 根据客户的要求实现对数据库的访问。应用 Delphi 丰富的数据库访问构件 (如 TQuery, TTable 等) 几乎可以实现任何形式数据库操作, 为远程数据库的访问提供极大的便利。

(3) 在 WebModule 模块中加入适当的标准回应内容生成构件 (Content Producer), 通过这些构件, 把诸如 TQuery, TTable 构件中的数据库访问结果转换成浏览器能够接收的形式。表 1 给出了 Delphi 5 提供的可直接使用的 5 种标准回应内容生成构件。

表 1 标准回应内容生成构件类

构件类名称	功能
TpageProducer	根据输入模板生成一个 HTML 命令串
TdataSetPageProducer	根据包含字段数据的输入模板生成一个 HTML 命令串
TdataSetTableProducer	生成一个 HTML 命令序列, 以数据表格的形式显示数据集中的记录
TqueryTableProducer	生成一个 HTML 命令序列, 以数据表格的形式显示 Query 所得的数据集中的记录
TmidasPageProducer	生成一个 HTML 文档, 表示来自应用服务器的活动数据库信息

(4) 将标准回应内容生成构件中的信息经由 TWebResponse 对象的 Content 属性反馈给客户端。

3.2 应用示例

在 Web 数据库应用系统中, Web 浏览器、Web 服务器和数据库服务器通常运行在网络上不同位置的工作站和服务器上。为了简单起见, 在本例中, Web 浏览器为 IE, Web 服务器为 MS Personal Web Server, 数据库采用 Delphi 5.0 自带的用于演示其数据库开发能力的 DBDEMOS 本地数据库 (其别名为 DBDEMOS)。

3.2.1 ISAPI DLL 的开发步骤

(1) 新建一个 Web Server Application 项目, 取名为 WebDBTestProject。

(2) 选择服务器程序类型为 ISAPI/NSAPI。

(3) 在 Delphi 5 自动生成的 WebModule1 中, 加入实现系统应用逻辑所必需的控件并编写事件代码。

本例演示如何对 DBDEMOS 本地数据库中的 employee.db 数据表动态地根据客户通过浏览器指定的年薪范围进行查询, 因此我们加入表 2 所示的控件并设置相应的属性:

表 2 加入 Web Module 模块中的控件及其属性

控件名称	属性设置	说明
Session1	AutoSessionName: True	对同时运行的多个实例进行完善的管理
Query1	DatabaseName: DBDEMOS SessionName: Session1_1	建立与数据库的连接, 实现对数据库的访问
QueryTableProducer1	Query: Query1 RowAttributes.Align: haCenter TableAttributes.Border: 1	把 Query1 的结果转换成浏览器能够接收的 HTML 格式

(4) 在 WebModule1 中创建一个负责处理客户的查询请求的 WebActionItem1, 将它的 PathInfo 属性设置为 /

QueryDB, 并在它的 OnAction 事件中加入以下代码:

(5) 将项目编译成 DLL。

```

procedure TWebModule1.WebModule1WebActionItem1Action(Sender: TObject;
    Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
var p1,p2: Double; code: integer;
begin
    val(Request.ContentFields.Values ['Min'],p1,code); // 获取客户指定的 Min 参数并转换
    val(Request.ContentFields.Values ['Max'],p2,code); // 获取客户指定的 Max 参数并转换
    with Query1 do // 添加实现查询的动态 SQL
        begin
            Close;
            SQL.Clear;
            SQL.Add('SELECT * FROM EMPLOYEE.db ');
            SQL.Add('WHERE (Salary >:Min) and (Salary < :Max)');
            Params [0].AsFloat := p1;
            Params [1].AsFloat := p2;
            Open; // 执行查询
        end;
        QueryTableProducer1.Query:=Query1; // 将查询结果作为要标准化的数据源
        Response.Content := QueryTableProducer1.Content; // 把生成的回应信息送回浏览器
        Query1.Close;
    end;
end;
    
```

(6) 把编译生成的 WebDBTestProject.dll 复制到 Web 服务器的 scripts 目录中。

浏览器中提供用户的输入界面, 通常采用 HTML 页面, 一般可使用 FrontPage 等工具来设计生成, 并存放在 Web 服务器上。下面是本例中作为客户界面的 HTML 页面代码, 以 QueryDBDemo.html 文件名存放在主域名为 zjz 的 PWS

3.2.2 客户界面的设计

为了实现客户与服务器的交互, 就必须在客户端的

的 scripts 目录中。

QueryDBDemo.html, 就会出现客户界面。

在 IE 的地址栏中输入: `http://zjz/scripts/`

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta name="GENERATOR" content="Microsoft FrontPage 3.0">
<title></title>
</head>
<body>
<font COLOR="#000000">
<p align="center">基于 ISPAI 的 Web 数据库技术示例</font></p>
<p>[例 1](数据库查询示例)查询年薪在指定范围内的员工记录.</p>
<form action="WebDBTestProject.dll/QueryDB" method="post">
  <p>年薪下限: <input name="Min" size="20"></p>
  <p>年薪上限: <input name="Max" size="20"></p>
  <p><input type="submit" value=" 确认 "></p>
</form>
</body>
</html>
```

4 结束语

尽管 ISAPI 的操作系统平台限于 Microsoft Windows 和 NT, 服务器平台限于 IIS、MS Personal Web Server 和 NT Workstation 上的 Peer Web Server。但是, 在目前新建的各种内部网中, 以 NT 为服务器, Windows 98 为客户端的系统较多, 在这样的环境下建立 Web 数据库应用系统时, 充分利用 Delphi 5 可视化快速开发能力和对数据库访问的强有力的支持, 开发采用 ISAPI 技术的基于

Web 的三层 C/S 应用系统, 无论在性能上还是在开发效率方面都具有一定的优势。■

参考文献

- 1 郑成增、陈志锋、王延珍等, WEB 数据库技术分析研究, 计算机工程与应用, 2000, 36(8): 106-108.
- 2 王兰波、雷渭侣、孙正严, Web 服务器与数据库服务器之间通信的不同方案[J], 计算机系统应用, 2000, (7): 37-39.
- 3 曾令友, Delphi 5.0 数据库应用开发 [M], 北京中国水利水电出版社, 2000.