

基于PB的Web数据库系统建设方案

李洪磊 (山东经济学院 计算机管理系 250014)

摘要:随着经济发展和社会进步,信息处理已逐渐突破局域网络的限制,向广域网应用发展,如电子政务、电子商务等。信息技术的发展,尤其是因特网及其相关技术的迅猛发展为此提供了有力的支持。本文就以PowerBuilder6.0为例介绍如何开发当前比较流行的全球化信息系统——Web数据库系统。

关键词:Web 服务器 超文本 RAD 应用服务器

一、前言

当前开发Web数据库应用系统主要有三种模式:RAD模式、Java模式和超文本模式。

1. RAD模式

RAD模式利用超文本插件(Plug-ins)技术,在Web页面中加入数据处理插件(如ActiveX)实现网际数据处理。采用这种技术开发应用系统与传统的开发方式非常相似,只不过不再是制作客户程序而是制作插件,因此开发过程相对简单,但这种实现方式缺点很多,如对客户端配置要求高(需要安装插件运行所需的各类资源),用户界面风格与WWW风格相异,插件下载慢,运行速度低等,这些原因决定了这种方式是不适合于因特网应用的,所以在开发应用系统时不宜采用这种模式。

2. Java模式

该模式采用JAVA技术(Applet和JDBC)实现Web数据处理。工作原理可用图1表示:

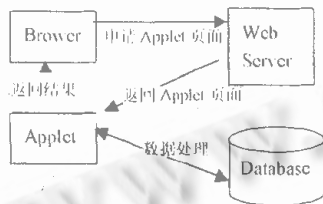


图1

3. 超文本模式

该模式利用专门设计的程序动态产生超文本信息,实现Web数据处理。这种专门设计的程序在该模式中属核心部件,它在系统中位于Web服务器和数据库服务器之间,是两种不同类型子系统联系的纽带,故称之为“中件”。该模式的工作原理如图2所示:

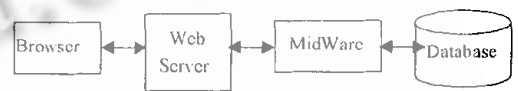


图2

二、基于PB的建设方案详述

在目前多种Web数据库系统构建技术中,Sybase公司提出的利用PowerBuilder构建Web数据库系统的技术颇具特色,笔者在经过细致研读、精心实验的基础上总结出了如下基于PB的Web数据库系统建设方案。

1. 工作原理

该方案支持两种模式,RAD模式和超文本模式。

(1)RAD模式。PB6.0提供了三种插件实现该模式下的系统开发:Window Plug-ins、Datawindow Plug-ins和Window ActiveX。本文对此模式下的系统开发不作介绍,有兴趣的读者可参阅《PowerBuilder Version 6.0 On-line Books》中的有关介绍。

(2)超文本模式。PB6.0实现了对该模式的支持,并对传统的实现方式进行了改进,新的工作模式如图3:

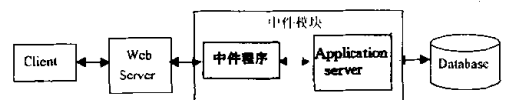


图3

在传统模式中,不同的应用需要编写不同的中件程序,由于所有的数据处理都集中在中件程序中,造成中件程序规模宏大,加重了WEB服务器负担,严重降低了系

统性能。为此,新模式引入了 Application Sever(应用服务器)和标准中件概念。应用服务器包含了原中件程序中的数据处理部分,标准中件程序仅保留了与 WEB 服务器的通信机制和调用应用服务器数据处理服务的控制机制。通过这种分离,大大降低了中件程序的规模,同时将数据处理分散到各个应用服务器中,利用 PowerBuilder 强大的分布计算功能,使系统整体性能获得了极大的提高。PB6.0 提供了这些标准中件供编程使用,这些组件是:

- (1) 标准 CGI 组件——PBCGI60.EXE
- (2) 标准 ISAPI 组件——PBISA60.DLL
- (3) 标准 NSAPI 组件(For the Netscape Commerce and Communications servers)——PBNS160.DLL
- (4) 标准 NSAPI 组件(For the Netscape FastTrack and Enterprise servers)——PBNS260.DLL
- (5) WSAPI for WebSTAR Web servers (on Power Macintosh only)

上述组件中,CGI 组件具有强大的跨平台性能但执行效率较低;ISAPI 组件目前只能在 NT 服务器上使用,但具有极高的执行效率,所以在使用 NT 和 IIS 作 Web 服务器时建议使用该组件。

2. 基于一个完整实例的开发方法介绍

由于新模式中中件已成为标准化产品,所以系统开发主要工作集中于 Application Server 的设计和配置。本文以 Windows NT4.0 和 IIS3.0 作 Web 服务器,MS Explorer4.0 作客户端程序介绍一个应用系统的开发。

(1) 设计 Application Server——AppServer

Application Server 是一种特殊的应用程序,它主要由不可视用户对象组成,外加一个通信对象 Transport,所有的数据处理均由不可视用户对象的对象函数实现,Transport 对象则用于监听来自中件程序的数据处理请求,并执行指定的对象函数,将结果返回中件程序。以下的 AppServer 实验程序完成了一个 Web 信息系统的登录功能。主操作界面如图 4 所示:

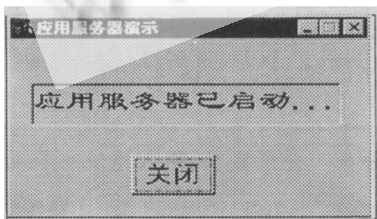


图 4

窗体名: W_main

①在 AppServer 的 Application 的 OPEN 事件脚本中写入如下代码:

```
Open(W_main)
```

②在 W_main 的 Open 事件脚本中写入如下代码:(ts 为 Transport 类型的非局部变量)

```
//设置 Transport 对象并启动监听
```

```
ts = create transport //建立一个 Transport 实例
```

```
ts.application = "AppServer" //设置应用服务器名称
```

```
ts.driver = "winsock" //设置通信驱动,该驱动具有强大的跨平台性能(推荐使用)
```

```
if ts.listen() < > 0 then //启动监听
```

```
messagebox("提示","应用服务器启动失败!") //
```

启动失败处理

```
destroy ts
```

```
close(w_main)
```

```
end if
```

st_1.text = "应用服务器已启动..." //在窗口中提示启动成功信息

在“关闭”按钮的 Clicked 事件脚本中写入如下代码:

```
ts.stoplistening() //终止监听
```

```
close(w_main) //关闭程序
```

③创建不可视用户对象 u_sys,并创建它的两个对象函数:f_login 和 f_main。

```
* f_login() 函数:
```

```
参数: UserId, Psw: String
```

```
返回值: String
```

```
程序代码:(略)
```

将上述程序代码编译为可执行程序即可生成应用服务器程序(Application Server)AppServer.exe。

(2) 配置 Application Server

①将上述可执行程序安装在寻址便捷,性能较高的 Web 服务器上(工作站或 NT 服务器上均可),假定机器名称为 AppStation。然后在该机器上配置名称为 internet (无特殊含义)的 ODBC 数据源,用户为 sa,无口令。

②用文件编辑器在 AppStation 系统配置文件目录下的 services 文件中加入如下项目。若无此文件可新建。

```
AppServer 10098/TCP (回车)
```

注:

- 系统配置文件目录对于 Windows 操作系统指其安装目录,如 C:\Windows;对于 NT 操作系统则指 \System32\drivers\etc 目录。

- AppServer 是 Transport 对象中定义的应用服务器名称。

- 10098 是其通信端口号(不能与其他服务端口冲

突)。

- TCP 为通信协议类型。
- 行末必须加回车符,否则系统视该项设置无效。

完成上述配置后便可启动应用服务器,等候标准中件传递而来的数据处理请求。

③配置 Web 服务器(假定 Web 服务器名称为 www.sie.edu.cn)

Web 服务器需进行以下配置才能建立与应用服务器的通信。

①安装标准中件程序。将 CGI 标准中件 PBCGI60.EXE 和 ISAPI 标准中件 PBISA60.DLL 安装到 Web 服务器的 \InetPub\scripts 目录中(IIS 专用目录),利用 IIS 服务管理器为该目录建立虚拟目录 scripts(系统默认),并设置其‘执行’属性,保存设置。

②配置应用服务器别名信息。在 Web 服务器所在的 NT 安装目录(如 c:\winnt)下建立一个名为 PB-WEB.INI 的 INI 文件,该文件是应用服务器的别名描述文件。在其中加入如下内容:

```
[AS1] //应用服务器别名
application = Appserver //应用服务器名称
driver = Winsock //使用的通信驱动名称
location = AppStation //运行应用服务器的机器的名称或 IP 地址
```

注:如果 location 的值不是运行应用服务器的机器的名称或 IP 地址,而是为了保密起见专门定义的特殊名称(如 px001),则需要到 NT 服务器 \System32\drivers\etc 目录下的 hosts 文件中加入对该名称的解析,否则无法使用该名称寻址。添加内容及格式如下:

应用服务器所在机器的 IP 地址 机器名(如 px001)
(回车)

(4) 制作应用系统主页

系统主页是客户端浏览的第一个页面,所以该页面必须是事先制作的静态页面。此例的主页定义如下:

```
(test.htm)
<html>
<head>
<title>a test</title>
</head>
<body>
<form method="POST" action="/scripts/pbisa60.dll/as1/u-sys/f-login">
<center><p>用户代码<input type="text" name="userid" size="10"></p></center>
<center><p>口 令<input type="password"
```

```
name="psw" size="20"></p></center>
<center><p><input type="submit" value="查询" name="B1"><input type="reset" value="返回" name="B2"></p></center>
</form>
</body>
</html>
```

说明:

对数据请求的发送由 FORM 完成,注意 action 内容的格式:

- scripts:标准中件所在 IIS 的虚拟目录名称。
- pbisa60.dll:标准 ISAPI 中件程序(采用 CGI 方式则为 pbegi60.exe)。
- AS1:pbweb.ini 中应用服务器别名。
- u-sys:包含被调用功能(函数)的不可视用户对象。
- f-login:被调用功能(函数)名称。

FORM 中的 Text Box 型元件的值可传递给被调用的函数,如上例中 userid 和 psw,其值便传递给 f-login 函数,但 f-login()中必须定义与元件同名的 String 型参数 userid 和 psw(不区分大小写)。该功能由中件程序实现,对用户透明。

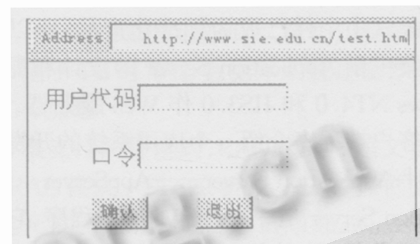


图 5 登录窗口

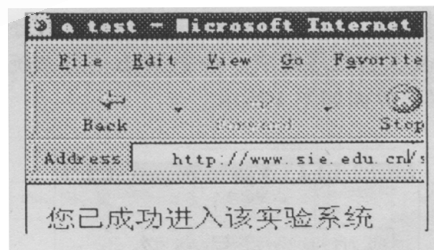


图 6 登录成功,进入系统主界面

上述 action 内容的作用是:请求 Web 服务器执行 Scripts 目录下的 pbisa60.dll 标准中件程序来调用 AS1 别名信息指定的应用服务器的 u-sys 对象的 f-login() 函数

进行数据处理

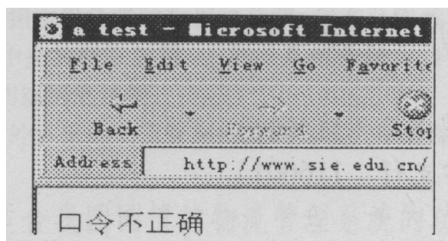


图7 登录不成功

至此一个完整的实例就完成了,以上是客户端执行结果演示。

(5)实例系统中数据表定义

```
Create Table dbo. user(
    Userid varchar(10) not null,
    Psw varchar(20))
Create unique index user on dbo. user(Userid)
```

三、构造应用系统常见错误、原因及解决

1. 发送数据请求时显示“Null object reference at line N in E event of object O”

原因:应用服务器中不可视对象 O 的 E 事件第 N 行代码处理有误

解决:修改该部分代码

2. 发送数据请求时显示“PowerBuilder method needs to return a string or blob, method 'X' in object 'O'”

原因:不可视对象函数返回值定义有误

解决:将返回值定义为 String 型或 Blob 型

3. 发送数据请求时显示“HTTP/版本号 不支持”

原因:中件所在信息服务器虚拟目录未设‘执行’属性

解决:将中件所在虚拟目录设上‘执行’属性

4. 发送数据请求时显示“PBWEBSetConfig: Could not find definition for server Alias 'X'”和“PBWEBRunRequest: PBWEBSetConfig has failed”

原因:①服务器端 PBWEB. INI 文件不存在或未放在指定目录 ②PBWEB. INI 中应用服务器别名定义有误(应用服务器名称设置有误、协议名称设置有误和应用服务器地址设置有误)

解决:按要求设置 PBWEB. INI 中应用服务器别名信息(应用服务器名称、协议名称和应用服务器地址)

5. 发送数据请求时显示“PBWEBRunRequest: Fail to connect”

原因:①应用服务器未启动监听②PBWEB. INI 中别名定义有误导致与应用服务器通信失败。

解决:①启动应用服务器监听②按要求设置 PBWEB. INI 中应用服务器别名信息(应用服务器名称、协议名称和应用服务器地址)

6. 无法启动监听

原因:①Services 文件不存在或未存放在指定目录②Services 文件中应用服务属性定义有误:应用服务名称错;端口号冲突或非法;通信协议名称错

解决:按要求设置 Services 文件

四、安全性、可维护性方面的考虑

尽管 PB 提供的超文本开发模式比较简捷的实现了系统开发,提高了运行速度,但在实践过程中也发现存在一些问题,其中比较严重的是安全性和可维护性方面的问题(这种模式的通病)。

首先在安全性方面,存在两个比较严重缺陷:(1)信息在网上传递缺乏有效的加密保护措施;(2)使用浏览器可看到页面源码,有经验的用户可避开一些步骤直截调用某些处理,如上例中用户可以定义 action = “/scripts/pbcgi60. exe/AS1/u_ sys/f_ main 而避开登录检查直截进入主操作界面(f_ main()生成的页面)。如果这些安全性方面的问题得不到有效解决,该项技术就很难获得很高的实用价值。PB6.0 提供了 SESSION 技术解决了问题(2)但无法解决客户端输入/输出信息的加密传递问题,笔者认为事关安全性方面的问题还应从 Web 低层着手解决更为合理、有效。

其次,由于采用该模式开发的系统,动态页面的生成机制被编译在应用服务器中,所以每当进行页面维护时(修改、新建、删除)就需要修改应用服务器程序中的相应源码,并需重新编译才能有效,这与其他非编译型实现模式(ASP、IDC)相比,在可维护性方面稍逊一筹。

参考文献

- [1] 《PowerBuilder version 6.0 Online Book》(美)Sybase 公司
- [2] 《现行各种 Web 数据库设计技术的分析与比较》吴良清 计算机系统应用 99 年第 3 期
- [3] 《CGI 技术与 ISAPI 技术性能比较》王元媛等 计算机应用 99 年第 4 期

(来稿时间:1999 年 7 月)