

Servlet 技术及其在网上结算系统中的应用

刘俊刚 卢孝强 黎洪生 武汉理工大学马房山西院信息学院 (430070)

Abstract

Servlet is a newborn extension web server in java, used to network service. Many computer companies are consolidating the support to servlet on their web servers software presently in view of servlet's prosperous development. Servlet's running principle, environment and functionality are presented systematically in this paper, and the typical example about internet settlement system is given to show the advantages with servlet to develop the server-side application.

Keywords: Servlet, Java, Web

摘要

Servlet 是 Java 里自带的用于网络服务的新兴扩展服务器技术, 由于其旺盛的生命力, 目前各大计算机公司几乎都在其 Web 服务器上加强了对 Servlet 的支持与服务。文章将对 Servlet 技术的工作原理, Servlet 的运行环境、功能等作一系统的介绍, 并以网上结算系统为实例探讨利用 Servlet 技术开发服务器端小应用程序的优势所在。

关键词: Servlet, Java, Web

1 引言

Sun 公司推出的 Servlet 技术是以 Java 语言编写服务器端小应用程序的有力工具, 是 Java 里自带的扩展服务器, 一般用于网络服务。Java 语言固有的特点使 Java 在开发网络环境下分布式多层结构应用中有着不可比拟的优势, 而 Java Servlet 的出现与发展, 使其在服务器端的功能有了进一步明显的提高, 目前各大公司如 ORACLE、SUN、IBM 等都在其 Web 服务器上加强了对 Servlet 的支持, 使 Java Servlet 显示出强劲的发展势头。Servlet 在扩充支持 Java 的 Web Server 上比 CGI 更加简单、方便。CGI 在每次响应一个请求时, 总要重新生成一个进程, 结果随着请求增加到一定数量时, 系统性能显著下降; 而 Servlets 是对服务器的直接扩展, 只在被动态地载入时才产生一个进程, 以后对请求的响应都由该进程产生一个线程来完成。另一个重要的不同点是, 当 Servlets 对一个响应结束之后, 并不是立即杀死该线程。这样, 在下次请求时就不用再次建立一个连接, 因为通常建立一个连接是很费时的。Servlet 也不象 ASP 技术那样基于服务器端的脚本环境, 而是直接工作在 Java 环境中, 拥有 Java 语言网络编程

所具备的一切高级特性, 如访问权限、安全性、易于进行数据库的访问、易于与 Applet 集成应用等, 同样能够实现数据库的动态访问, 并且由于 Servlets 的简便、小巧和优异的可扩展性使其在金融、贸易、电子商务等 Web 应用中倍受青睐。

2 Servlet 的工作原理、运行环境及功能

2.1 Servlet 的工作原理

从更高层上看, 如果将 Servlet 与大家熟悉的运行于客户机浏览器上的 Applet 小应用程序作比较, Servlet 是一个服务器端的 Applet, 不同之处在于它不运行在浏览器上或者是以 GUI 的形式呈现给用户, 而是以请求-应答的方式同运行在服务器上的 Servlet 引擎(engine)交互工作, 处理客户机发送的请求, 实现客户机与服务器之间的通讯(见图1)。Servlet 以 HTML 作为通讯手段与客户机进行交互。由于这种方式符合传输 Web 文档的 HTTP 协议所规定的标准, Borland 公司的 JBuilder 集成开发环境为 Servlet 提供的高效开发软件包所包含的基类正是基于 HTTP 协议的 HttpServlet 类, 大部分 Servlet 程序都是通过继承该类来创建的。在收到客户机的请求后, Servlet 将利用读者易学易懂的

HTML 语言把应答数据送回客户机。

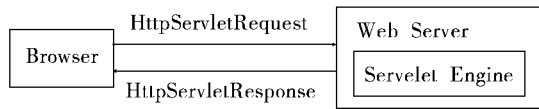


图 1 Servlet 与 Browser 的交互示意图

2.2 Servlet 的运行环境

由于 Java 语言的可跨平台特性, Servlet 可在多种平台上运行, 除 Sun 公司的 Solaris 操作系统平台外, 还可运行于 Windows、Unix、Linux 等平台之上, 当然速度是有差异的。运行与配置 Servlet 时, 系统必须具备支持 Servlet 的 Web Server, 这类服务器主要包括:

(1) JBuilder 的 Professional 版或者 Enterprise 版运行环境本身提供的 JspRunner Web 服务器。当右击 Servlet 的 .SHTML 文件时, 系统会自动启动此服务器。

(2) Sun 公司的软件包 JSDK (Java Servlet Development Kit) 提供的 servletRunner 服务器。

(3) Sun 公司出品的 Java Web Server。

(4) 其它支持 Servlet 的 Web Server 或者带有第三方 Servlet 引擎的 Web Server, 如 Apache, Netscape 等。

这些服务器软件基本可从 Sun 公司或 Borland 等公司的 Internet 网站下载得到。

2.3 Servlet 技术所提供的功能

(1) 处理 HTML 的表单标签 (FORM) 递交 (POST) 来的数据, 如电子商务系统的货物定单数据、信用卡的使用等。这种功能主要应用于网络化的商业在线支付系统中, 后面关于结算系统的实例将详细的讨论这种功能。

(2) 同时处理多方请求的功能。利用 Java 独具特色的线性安全特性可以同时处理来自多个客户机的请求, 并提供同步 (synchronize) 这些请求的功能。如在线会议系统或在线讨论系统等就可以利用 Servlet 的这个特性进行开发。

(3) 平衡服务器负载的功能。在 Servlet 中, 可以根据任务的需要将一个逻辑任务分配给几个服务器执行。当一个服务器由于多个客户机向其发送请求而应接不暇时, 它可以将一部分请求转送给其它服务器的 Servlet 或其它服务器程序, 共同分担并完成任务。

(4) 作为活动代理器的社区。程序员可以定义多个活动代理器, 一个活动代理器可以是一个 Servlet, 程序员可以在它们之间分配任务, 传递数据, 进行通讯。

3 应用 Servlet 技术设计网上在线结算系统

本设计是利用 Servlet 为一家 CocaCola 饮料超市设计的网上结算系统, 是笔者在 Windows NT 操作平台下利用 JBuilder 集成开发环境实现的。系统演示了 Servlet 与客户机间的通讯机制和 Servlet 运行周期的三个阶段, 显示了 Servlet 快速开发服务器端小应用程序的优势。

3.1 系统要完成的工作

顾客在客户机上将选中的饮料和所需饮料的数量递交给服务器后, Servlet 通过接收请求和进行运算应答客户机, 告诉顾客应付的款额。

3.2 系统实现

超市经营的饮料有 CocaCola、Sprite、Fanta、Apple、Lemmon 等多个品种, 本设计仅以这前五种为例编程。

(1) 先在客户机端 HTML 代码段的 FORM 表单中列出要提交的参数, 代码如下:

```

<FORM ACTION="http://servlet.my-servlets.firstservlet" method="POST" >
  <font size="5" color="navy" >
    <center>Select all, just left what customer don't
    want number=0
    <div> <pre>
      <div> <input type="checkbox" name="CocaCola"
      value="CocaCola" >CocaCola $1.50
      Number: <input type="text" name="CocaNum" value="0" size="8" >
      <div> .....
      <input type="submit" name="OK" value="GO" >
      <input type="reset" name="Cancel" value="Cancel" >
    </pre> </div>
    </center> </font>
  </FORM>
  
```

程序段中的省略号代表其它种类的饮料不再一一列出, 它们在形式同 CocaCola 一样, 有两个 `<INPUT>` 标签参数, 下面的 Servlet 程序中省略的代码与此相似。可见, 此例中需要提交的参数有两部分, 一部分是顾客所要购买的可口可乐的种类, 另一部分是顾客所要购买的数量。程序将递交这两部分参数给 Servlet 程序

④) 下面再看 Servlet 是怎样被创建来处理这些参数的。

建立 CocaServlet 类, 由于 JBuilder 中 Servlet 面向对象的特性, 建立该类可以直接继承 Java 软件包中基于 HTTP 协议的 HttpServlet 类, 该类提供管理 Servlet 及其与客户机通讯的方法机制。

```
public class CocaServlet extends HttpServlet {
    int connections = 0;
    public void init (ServletConfig config) throws ServletException {
        super.init (config);
    }
}
```

装载 Servlet 时, init() 方法对 Servlet 进行初始化, 这是 Servlet 生存周期的第一阶段所完成的任务。当 Servlet 被安装到服务器时, 将首先运行 Servlet 的这个初始化方法 (一般为 init () 方法), 对 Servlet 提供处理请求所需要的配置。init () 方法在 Servlet 的生存周期中仅被运行一次, 除非重新装载 Servlet (若重新装载, 必须事先退出 Servlet)。注意此时系统不能支持 Java 多线程服务的概念, 否则, Servlet 将不能正常启动。

下面 Servlet 将进入了生存周期的第二阶段, 利用 doPost () 方法接收并处理客户机表单 (FORM) 递交来的上述两部分参数。doPost () 方法中有两个参数 HttpServletRequest 类和 HttpServletResponse 类, 它们是 Servlet 收到客户机的请求时接收到的两个基本对象, HttpServletRequest 类封装了客户机向服务器发送请求时的通信机制; HttpServletResponse 类则封装了 Servlet 响应客户机请求时的通信机制。

```
public void doPost (HttpServletRequest request,
    HttpServletResponse response) throws ServletException,
    IOException {
    double volume = 0.0; // 标识顾客应付款的变量
    double Coca_price = 0.0; // 标识各种可口可乐每杯价格的变量
    .....
    String Coca = ""; // 接收顾客所要可口可乐种类的量
    .....
    int Coca_num = 1; // 接收顾客所要数量的变量
    .....
    try {
        Coca = request.getParameter ("Cocacola");
        .....
        Coca_num = Integer.parseInt (request.getParameter ("CocaNum"));
    }
```

```
.....
    } catch (Exception e) { e.printStackTrace ();
    }
    try {
        if (Coca.equals ("Cocacola"))
            Coca_price = 1.50;
        .....
        } catch (Exception e) {
            e.printStackTrace ();
        }
    }
    volume = Coca_price * Coca_num + ..... + Lemmon_price *
    Lemmon_num;
```

在第一个 try {} 块中利用 getParameter () 完成了对参数的接收, getParameter () 是 HttpServletRequest 类中专门负责接收客户机 FORM 表单中参数的一个方法。第二个 try {} 块中的代码作用是标识顾客所买饮料的价格。最后一条语句则对收到的参数进行运算, 将最后结果赋予 Volume 变量。

接下来的代码段将显示 Servlet 是如何将最后结果回复给客户机的。开始的两条语句告诉浏览器, Servlet 将使用 ServletOutputStream 类以 HTML 的形式做应答。

```
response.setContentType ("text/html");
PrintWriter out = new PrintWriter
(response.getOutputStream());
out.println ("<html>");
out.println ("<head><title>firstservlet</title></head>");
out.println ("<body BGCOLOR = \"#008080\">");
out.println ("<h1 align = \"center\"><Cocacola</h1>on line payment system</h1>");
out.println ("<p><h4>You are customer number:");
connections++;
out.println (Integer.toString (connections));
out.println ("</h4>");
out.print ("<PRE></PRE>");
out.print ("<br>");
out.println ("<font size = \"5\">");
out.print ("Final volume you should pay: ");
out.println (Double.toString (volume));
out.println ("</font>");
out.println ("</body></html>");
out.close ();
```

这段回复客户机请求的代码, 是将大家所熟悉的 HTML 代码嵌入到 Java 中, 向客户机发送信息。如同 HTML 语言的运行机制一样, 将信息显示在浏览器上, 比 ASP 通俗易懂, 更不象 CGI 需要程序员亲自编

写一些接口代码那样复杂,就能将顾客应付款的数额 volume 变量回送给浏览器,完成与客户机的交互。

```
}
}
```

到此,基本完成了对结算系统的设计。

在确认系统具备支持 Servlet 的 Web Server 后,如果读者想运行该例,还需将程序所应包含的头文件及其它代码补充完整(请参见有关 Servlet 的教材或资料),并需在 JBuilder 的 project | project properties 选项中设置好主机名和端口值。

本例演示的是一个请求时的情况。当有多个请求的时候,每个请求都有对应的 Servlet 线程方法,负责答复收到的请求。由于 Servlet 能为同时在一个进程中运行多个服务器线程方法提供线程安全,因此只要在相应的线程方法前加上线程同步关键字(synchronize)即可。线程安全在这种情况下显得特别重要,否则,系统会因线程执行的混乱而导致数据出错。如果系统没有同时处理多个服务器方法的必要,只要将 Servlet 直接实现 JSDK 软件包中的 SingleThreadModel 接口就行了。当服务器停止 Servlet 的运行时,也就是在 Servlet 生存周期的第三阶段

——也是最后一个阶段,系统会调用 destroy()方法,这个阶段可由系统本身完成,而不需程序员编写代码。同初始化方法一样,destroy()方法在 Servlet 的生存周期中也仅仅被调用一次,宣告 Servlet 运行的结束。因此需要同样注意 destroy()方法的线性安全,保证被调用期间不被其它线程介入,如果 Servlet 是直接实现的 SingleThreadModel 接口,就不用担心这个问题了。

4 结束语

综上所述可以看出,Servlet 是快速开发服务器小应用程序强有力的工具,利用它开发的银行系统软件已被国内越来越多的银行采用,系统运行稳定、速度快,应用前景广阔。Servlet 的功能还必然会随着 Java 技术的发展而得到加强。

参考文献

1. 胡泳,张志浩,陈福民,Java Applet 与 Servlet 通讯机制的探讨,计算机系统应用,2000.4
2. 方亮,金晶,虚拟教室中电子白板系统的实现,计算机与自动化,2000.1
3. <http://java.sun.com/products/servlet>

(上接第 36 页)

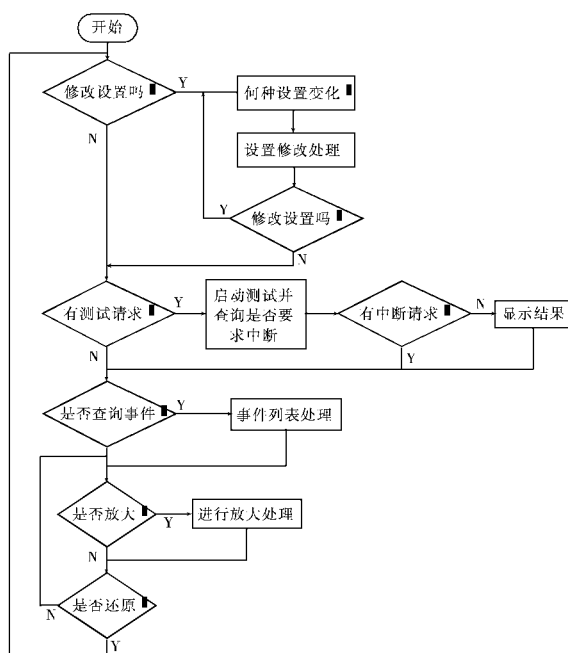


图2 程序流程图

5 结论

基于虚拟仪器开发的本系统具有完善的仪器参数的设置、数据采集、图形显示和对采集到的原始数据可以进行分析处理的功能。美国 NI 公司的 Lab-view 开发平台具有良好的品质、弹性和性能,能够提供以往传统 MMI(人机界面)或 SCADA 软件所无法想象的简洁、灵活和快速。本测试系统的最主要特点是使得仪器的功能面向用户,使用者还可以通过 GPIB 接口总线,将其它的光通信测量仪表(例如:光谱分析仪、光功率计、光万用表、稳定化光源等)集成起来,开发出集成光通信测量仪表的自动测试系统平台。在该测试平台上,使用者可根据实验和科研的具体实际需要自行编制测试程序,完成相应的自动测试功能,既操作简单,又能提高效率。更加有利于合理地、充分地使用仪器设备。

参考文献

1. 陈光禹等,现代电子测试技术~信息装备的质量卫士,国防工业出版社,2000 年
2. 邮电部电信总局主编,长途通信光缆线路维护手册,人民邮电出版社,1992 年