

用 Java 实现 Web 数据库应用

刘玲霞, 刘守义, 王 凌

(空军工程大学 电讯工程学院, 陕西 西安 710077)

摘 要:在分析比较几种 Web 数据库访问方式的基础上,指出了用 JDBC 访问数据库的优越性。然后介绍了用 Java 的 Applet、Servlet 及 JDBC 实现 Web 数据库应用的两种结构模式。最后,用一个实例说明了如何用 Java 实现 Web 数据库应用。

关键词: Applet; Servlet; 数据库

中图分类号: TP317 **文献标识码:** A **文章编号:** 1009-3516(2001)02-0030-04

随着计算机网络技术,特别是 Internet/Intranet 的不断发展,对数据库应用技术产生了巨大的影响。实现浏览器方式下的数据库访问,已成为 Internet/Intranet 应用服务的热点之一。

Java 语言所具有的平台无关性、安全性、可移植性、可以自动下载等特点,使得它成为构建 Web 数据库应用的极好的语言。尤其是 Java Servlet 的出现与发展,使 Java 在服务器端的功能有了很大的提高。将 Applet、Servlet 相结合,利用 JDBC 技术,能创建出功能完善,效率较高的 Web 数据库应用。

1 几种 Web 数据库访问方式的比较

Web 数据库访问方式通常可分为基于 Java 的 JDBC、通用网关接口(CGI)及应用编程接口(API)^[3]。

CGI 作为一个进程提供。服务器 API 一般作为一个 DLL 提供。它们的工作过程都是:(1) 客户使用某种协议与 Web 服务器建立连接并发送请求;(2) Web 服务器运行某一 CGI 或 API 程序,并把从客户收到的请求体中的信息传送给 CGI 或 API 程序;(3) 程序向数据库服务器提出请求,数据库服务器返回请求结果给该程序;(4) 该程序把得到的结果转化为 HTML 文档,返回给 Web 服务器,然后由 Web 服务器将得到的 HTML 文档传给浏览器。

JDBC 是 Javasoft 推出的一个标准 sql 数据库访问接口。通过它,Java 应用程序可以直接访问数据库。它给编程人员提供了一个统一的、无缝的接口,使开发人员可以方便地建立与数据库的连接,将 sql 语句传送给任何一种数据库,与数据库进行信息交互。各大数据库厂商都宣布支持 JDBC,已推出或准备推出各自的 JDBC 驱动程序。

CGI 和 API 方式与数据库的交互必须通过 Web 服务器才能完成,再加上自身的本质缺陷,使得其查询效率不高。JDBC 能直接访问数据库,无须 Web 服务器的介入,不仅消除了 CGI 和 API 模式中 Web 服务器带来的瓶颈现象,更提供了一种面向连接的通讯,可以支持交互式的查询和复杂的数据库事务处理,具有更好的使用效率。

2 如何用 Java Applet 及 Java Servlet 实现 Web 数据库应用

Applet 是 Java 的小应用程序,以 class 的形式嵌入 HTML,通过主页发布到 Internet。网络用户访问服务器的 Applet 时,这些 Applet 就从服务器下载到本地,然后在浏览器中由本地 Java 虚拟机解释执行。Servlet 是 Java 的小服务器程序,运行在服务端,由服务器端的 Java 虚拟机解释执行。它采用线程机制。

收稿日期:2000-07-15

作者简介:刘玲霞(1977-),女,江苏靖江人,硕士生,主要从事计算机应用研究。

用 Applet 及 Servlet 实现 Web 数据库应用,可以有两种结构模式。

2.1 两级式结构

在这种结构中,所有的功能完全由 Java Applet 实现。实现过程:

(1) 客户端访问 Web 服务器,下载嵌于 HTML 文档中的 Java Applet,通过浏览器所带的 Java 虚拟机 (JVM) 执行 Applet,此时客户端与 Web 服务器脱离。

(2) Applet 与用户交互,根据输入信息与某数据库服务器建立连接,进行查询、修改等数据库操作。数据库服务器将处理完的结果返回给客户。

2.2 三级式结构

在这种结构中,添加了一个 Java Servlet 作为中间件,功能由 Applet 和 Servlet 共同实现。Applet 负责界面处理及与 Servlet 的通讯,Servlet 则负责与数据库进行交互。实现过程:

(1) Applet 从 Web 服务器下载后在客户端运行,请求与 Servlet 建立连接。

(2) Servlet 启动一个新的子线程响应客户机的请求,建立 Servlet 与客户机的连接,通过该连接实现客户机与服务器的通信。

(3) Applet 将访问数据库的请求发送给 Servlet,Servlet 子线程根据请求通过 JDBC 调用数据库,数据库服务器将结果返回给 Servlet 子线程;

(4) Servlet 子线程将结果送回 Applet,Applet 负责按要求显示结果。

3 具体实例

下面将用一个具体实例说明如何用三级访问结构构建一个 Web 数据库应用。

Servlet 与 Applet 之间通过 socket 进行通讯。

Applet 中相关代码:Applet 中与用户交互部分等省略,重点在 Servlet 与 Applet 之间的通讯上

```
Socket socket = new Socket(getCodeBase().getHost(),SERLET_PORT);
//创建一个 Socket 对象,SERLET_PORT 为服务器端 Servlet 所监听的端口号
public acceptData{ //接收 Servlet 传来的信息
    InputStream INS = socket.getInputStream();
    DataInputStream results = new DataInputStream(new BufferedInputStream(INS));
    String queryResults = results.readLine();
}

public sendData( //将查询信息发送给 Servlet
    PrintStream toServlet = new PrintStream(socket.getOutputStream());
    toServlet.println(String queryMessage);
}
```

Servlet 中相关代码:省略了从客户端获得数据的代码,如查询条件“dbURL”,“user”,“password”,查询语句等。

```
public class Dbquery extends HttpServlet{
public void init(ServletConfig sc) throws ServletException{
    super.init(sc);
    ServerSocket serverSocket = new ServerSocket(SERLET_PORT);
    //监听客户请求
    this.handle(serverSocket.accept());
    //接受请求
}

public void handle(Socket client){
    new queryConnection(this,client).start();
}
```

```

}

public class connectionDB {           //完成连接数据库的功能
    String user;
    String dbURL;
    String password;
    Connection connect;
    public void connectionDB( String dbURL, String user, String password) {
        this. user = user;
        this. dbURL = dbURL;
        this. password = password;
        this. connect = DriverManager. getConnection( dbURL, user, password );
    }
}

public class Doquery {               //完成查询数据库的功能
    Object [ ] [ ] Arr = new Object [ ] [ ];
    String sql;
    public Object [ ] [ ] Doquery( String sql) {
        this. sql = sql;
        connectionDB connectDB = new connectionDB;
        Connection connect = connectDB. connectionDB( String dbURL, String user, String password );
        Statement sqlStatement = connect. createStatement ( );
        ResultSet resul = sqlStatement. executeQuery ( sql );
        ResultSetMeta resultMmeta = resul. getMetaData ( );
        for ( int i = 0; i < sqlStatement. getMaxRows ( ); i + + ) {
            while ( resul. nex ( ) ) {
                for ( int j = 0; j < resultMmeta. getColumnCount ( ); j + + ) {
                    Arr [ i ] [ j ] = resul. getString ( j );
                }
            }
        }
        return Arr;
    }
}

class queryConnection extends Thread { //响应客户端请求,并将结果发送到客户端
    Dbquery query;
    Socket client;
    Object [ ] [ ] sets = new Object [ ] [ ];
    queryConnection ( Dbquery query, Socket client) {
        this. query = query;
        this. client = client;
        setPriority ( NORM _ PRIORITY - 1 );
    }
    public void run ( ) {
        Doquery dbq = null;
        dbq = new Doquery ( " dbURL ", " user ", " password " );
        sets = dbq. Doquery ( sqlString1 );
    }
}

```

```
PrintStream sendToC = new PrintStream( client.getOutputStream() );
sendToC.println( sets[ i ][ j ]. toString );
sendToC.close;
client.close;
```

4 结束语

随着 Internet/Intranet 技术的不断发展,Web 与 Java 技术会更成熟,其应用前景也必定会更加广阔。Java 技术的不断成熟,将会使得其在 Web 的数据库应用中占据越来越重要的地位。

参考文献:

- [1] Ashton Hobbs. 自学 JDBC 数据库编程[M]. 丁一青,吴静. 北京:清华大学出版社,1998.
- [2] 王克宏. 最新 JAVA2 核心类库详解(下)[M]. 北京:清华大学出版社,1999.
- [3] 丁明吉. Web 服务器与数据库服务器的连接技术[J]. 计算机应用,1999,19(6):18-20.

The Applications of Web Database With Java

LIU Ling-xia, LIU Shou-yi, WANG Ling

(The Telecommunication Engineering Institute of the Air Force Engineering University(AFEU.), Xi'an 710077, China)

Abstract. On the basis of analysing and comparing some methods to access Web Database, the superiority of the JDBC technology is pointed out. Then the article introduces the two models to implement the applications of Web database by using Applet, Servlet and JDBC of Java. Finally, an example about implementing the applications by using Java is given.

Key words: Applet; Servlet; database