

面向分布式处理的 WebGIS 平台软件开发

吴健平 陈宇亮

(华东师范大学地理信息科学教育部重点实验室, 上海 200062)

摘要 目前已有许多 WebGIS 平台软件,但在分布式数据访问处理这个问题上,尚无完全成熟的解决方案。在研究和分析国内外已有 WebGIS 软件技术的基础上,探索了分布式 WebGIS 系统的实现方案,采用 Java 技术从底层开发了一个 WebGIS 平台软件。利用该软件所建立的基于 SIG 的城市空间信息服务系统原型能够实现分布式空间数据的访问操作,为进一步实现分布式异构数据的共享和互操作打下了一个基础。

关键词 WebGIS 分布式处理 平台软件 Java

中图分类号: TP301 **文献标识码:** A **文章编号:** 1006-8961(2005)06-0790-04

Development of Platform Software for WebGIS Based on Distributed Processing

WU Jian-ping, CHEN Yu-liang

(The Key Laboratory of Geoinformation Science, Ministry of Education, East China Normal University, Shanghai 200062)

Abstract There are now many platform software for WebGIS, but none provided a perfect solution to distributed processing. On the base of analyzing the existing software technology of WebGIS, we explored the solution for distributed WebGIS and developed a platform for WebGIS using Java. With this platform, a city spatial information service system based on SIG was established, which can access and process the distributed spatial data.

Keywords WebGIS, distributed processing, platform software, Java

1 引言

WebGIS,即互联网地理信息系统,是一种基于 Internet/Intranet 技术标准和通信协议、以互联网为环境,客户端以 Web 页面作为 GIS 软件的用户界面把 Internet 和 GIS 技术结合在一起,为各种地理信息应用提供 GIS 功能的技术。通过 WebGIS 系统,网络用户可以从 Internet/Intranet 上任意一个节点浏览 WebGIS 站点上的空间和属性信息,实现空间信息查询和空间分析计算。WebGIS 极大地改变了人类的生存方式,是当前 GIS 研究领域的重要方向之一^[1]。

目前已有许多 WebGIS 平台软件,如 ArcIMS、

MapXtreme、GeoSurf、GeoBeans 等,但在分布式数据访问处理这个问题上,目前都还没有完全成熟的解决方案。为此对分布式空间数据的访问与操作的解决方案进行了研究,开发了一个面向分布式处理的 WebGIS 平台软件。

2 软件开发方案

WebGIS 的实现有两种方案:瘦客户端方案和胖客户端方案。针对不同方案,有相应的平台软件。本软件按照胖客户端方案进行开发,因此软件的功能模块分为客户端和服务端两大部分。其中客户端功能模块涉及与服务器的通讯、对下载的数据

基金项目:国家高技术研究发展计划“863”资助项目(2002AA134020)

收稿日期:2004-12-07;改回日期:2005-03-24

第一作者简介:吴健平(1962~),男,教授,博士生导师。1996年于华东师范大学获博士学位。目前主要从事城市遥感与地理信息系统方面的研究。E-mail:jpwu@geo.ecnu.edu.cn

进行各种操作等;服务器端功能模块包括与客户端数据通讯模块、本地数据服务模块、服务器间数据通讯模块、元数据库访问模块等。整个软件采用 Java 技术进行开发,具有可移植性和扩展性。

2.1 客户端

胖客户端模式是通过在浏览器端嵌入 Applet 作为客户端运行环境。客户端 Applet 通过下载相应的 Java 程序(功能包)实现各种 GIS 功能,其包括空间数据读取、地图显示与地图操作等。

2.1.1 空间数据读取

软件目前支持的空间数据格式是 Shapefile 文件。

为读取 Shapefile 文件,设置了 ShapefileShape 接口,统一 Shape 文件不同元素(点、线、面)的操作,对不同类别地图元素的存储通过类扩展这个接口来实现。例如每个点元素存储在每个 ShapePoint 类的实例中,每个多边形元素存储在每个 ShapePolygon 类的实例中。

2.1.2 地图的显示

对于地图的显示,本软件在类的组织上采用类似 ArcView 的结构,分为不同的 Layer 来进行管理,不同的 Layer 组成 Theme,不同的 Theme 组成 Viewer。类结构关系如图 1 所示。

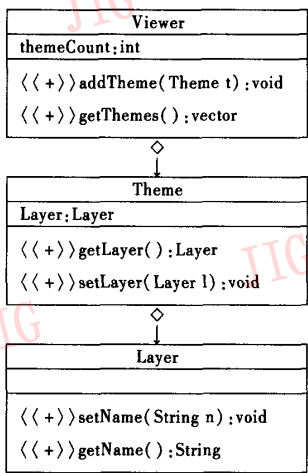


图 1 地图类关系
Fig.1 Relationship of map classes

根据点、线、面类型不同,扩展 Layer 产生相应的类,包括 PointLayer、LineLayer、PolygonLayer 等。

Java 语言提供了专门的 API 支持窗口中的绘图功能。比如 drawLine, drawPolygon, fillPolygon 等都是其固有的方法,使用非常方便。另外通过调用 Java 2D API 中提供的更强的函数可以更加方便地

实现复杂的绘图功能。

2.1.3 与服务器的通讯

由于 Java 对网络应用的良好支持,在 Internet 上可以通过不同的协议(如 HTTP, TCP, UDP 等)实现对象和数据的传输^[2,3]。在本软件中,Applet 通过 Java 提供的 URLConnection 建立与服务器的连接,并且通过此连接实现向服务器发送请求,传送数据。示例如下:

```

//建立与 Servlet 的联接,向 Servlet 传输数据
URL url = new URL(serverURL);
URLConnection con = url.openConnection();
//设置连接的参数
con.setDoOutput(true);
con.setDoInput(true);
con.setDefaultUseCaches(false);
con.setUseCaches(false);
con.setRequestProperty("Content_type", "application/x-www-form-urlencoded");
con.setRequestProperty("Content-length",
String.valueOf(queryString.length()));
//获取数据流输出通道,写入数据
DataOutputStream out =
new DataOutputStream(con.getOutputStream());
out.writeBytes(queryString);
  
```

2.2 服务器端

在服务器端,主要分为 3 个功能模块:与客户端数据通讯模块、本地数据服务模块、服务器间数据通讯模块。这些模块通过 Java Servlet 来实现。

服务器端各模块的关系及运行过程如图 2 所示。

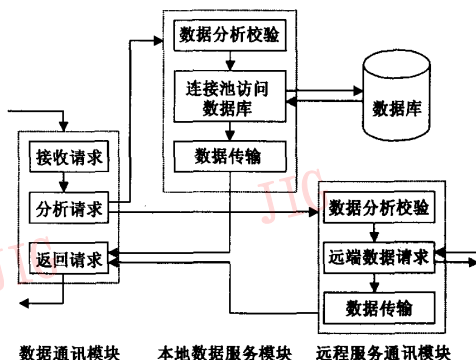


图 2 服务器端各模块的关系及运行过程

Fig.2 Relationship of modules in server and running process

2.2.1 与客户端数据通讯模块

数据通讯模块接收客户端的请求,分析请求的

数据要求,确定数据的存储地址:本地数据或者异地数据,然后调用相应的数据服务模块,接收数据服务模块传回来的数据,再把数据返回客户端。

服务器端的数据通讯模块的主要功能是建立与客户端的数据通道。另外,进行请求的解析,分析是否调用元数据库,然后把请求结果发给本地数据服务模块或者远程服务通讯模块。代码示例如下:

```
//读取客户端请求参数
String tableName = req.getParameter("table");
String columnName = req.getParameter("column");
String whereClause = req.getParameter("where");
if (whereClause.indexOf("Server") == -1) {
//调用本地数据库,返回客户端查询结果
getData(resp, tableName, columnName, whereClause);
} else {
//连接其他服务器
String serverURL = getServer(server);
ConnectRemoteServlet crs =
    new ConnectRemoteServlet(serverURL);
crs.getServerData(
    resp, tableName, columnName, whereClause);
}
```

2.2.2 本地数据服务模块

本地数据服务模块实现与本地数据库的连接,根据请求要求提取相应的数据。模块功能又可以细分为数据分析校验、数据传输、连接池访问数据库。数据分析校验检查网络传输的数据是否正确;数据传输功能完成数据库数据的传输处理;连接池访问数据库实现对连接池的访问,建立与数据库服务器的连接。连接池访问数据库部分代码示例如下:

```
//在连接池中添加新的连接
private int addConnection()
{
    int index = -1;
    try {
        int size = m_pool.size() + 1;
        fillPool(size);
        if (size == m_pool.size()) {
            index = size - 1;
        }
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
    return index;
}
```

```
//从连接池中获得新的连接
private int find(java.sql.Connection con)
{
    int index = -1;
    // Find the matching Connection in the pool
    if ((con != null) &&
        (m_pool != null)) {
        for (int i = 0; i < m_pool.size(); i++) {
            ConnectionObject co = (ConnectionObject)
                m_pool.elementAt(i);
            if (co.con == con) {
                index = i;
                break;
            }
        }
    }
    return index;
}
```

2.2.3 服务器间数据通讯模块

远程服务通讯模块实现和其他服务器的通讯连接,本模块根据元数据信息向相应的服务器发出请求,建立连接和数据传输通道,最后接收远端服务器返回的信息。它是实现分布式数据访问的关键。

数据请求发送代码示例如下:

```
//连接远程 servlet 的数据流
ByteArrayOutputStream byteOut =
    new ByteArrayOutputStream();
DataOutputStream out = new DataOutputStream(byteOut);
out.writeUTF(queryString);
out.flush();
byte buf[] = byteOut.toByteArray();
con.setRequestProperty("Content-type",
    "application/octet-stream");
con.setRequestProperty("Content-length",
    "" + buf.length);
DataOutputStream dataout =
    new DataOutputStream(con.getOutputStream());
dataout.write(buf);
dataout.flush();
dataout.close();
```

3 软件应用

利用平台软件开发了基于 SIG 框架的城市(上海)空间信息服务原型系统。在原型系统中,采用了 3 台 WebGIS 服务器来模拟分布式数据节点,服

务器 1 存储的数据是上海市水系图;服务器 2 存储的数据是畜牧场分布图;服务器 3 存储的数据是上海市行政区划图。

在服务器端,首先把各服务器功能模块安装到相应的服务器访问路径下面,分别是 Apache 网络发布目录和 Tomcat 的 Java Servlet 服务目录;然后在各个服务器设置数据源,配置数据库连接池文件,设定发布的数据库名称和密码,选择合适的连接池大小和最大连接数等参数。在本系统中,连接池大小设为 5,最大连接数设为 100。

启动 Apache 和 Tomcat 服务器之后需要针对每个服务器启动数据库连接池,通过浏览器访问管理服务器连接池的 Servlet 就可以激活启动连接池。

系统客户端的运行环境只需要支持 Java 虚拟机运行的网络浏览器就可以了,如 IE5.0 或者 NetScape5.0。用户打开门户网站的网页后,将出现 Applet 下载提示,几秒钟之后,网页会提示 Applet 程序下载完成,客户端程序自动启动,用户即可以进行操作。

客户端界面主要分为 3 个部分:地图主界面、操作工具栏和图层管理区。地图主界面即地图显示和用户交互操作的区域;操作工具栏位于上部,是一些地图操作的选择按钮,如放大、缩小、查询等功能选择按钮。图层管理区在界面右侧,可以设置需要加载的图层、显示/隐藏的图层等。

选中需要加载的图层数据,Applet 会向连接的门户网站服务器请求数据。该服务器分析客户端请求,如果请求是针对该服务器本地数据的,就调用本地数据库中的相应数据,把结果数据返回给客户端;如果是对于异地数据(其他服务器上的数据)的请求,该服务器会自动连接远端服务器,接收远端服务器返回的数据,再转发给客户端 Applet 处理。

客户端接收了连接服务器返回的数据之后,Applet 自动解析数据并在主界面上显示地图(图 3)。加载地图数据之后,即可以对这些数据进行本地操作。如可以设置各个图层是否显示。选择工具栏中的放大、缩小、移动工具,可以对地图进行放大、缩小或移动等操作。

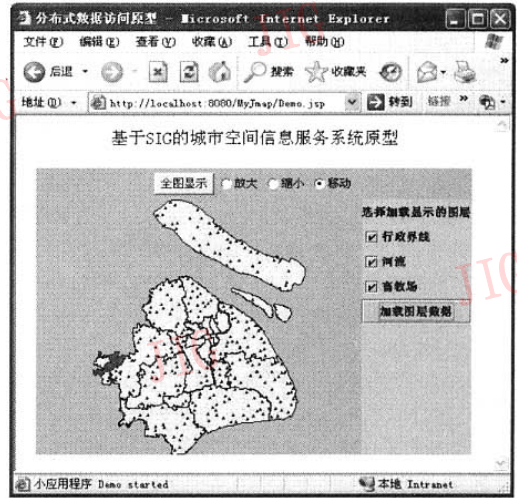


图 3 加载了地图的界面

Fig. 3 Interface after maps load

4 结 论

WebGIS 的一个发展方向就是要实现分布式数据的共享与处理,目前所开发的平台软件还只是一个原型,还有很多有待解决的问题,需要后续工作来完成,如异构数据的访问、客户端功能的进一步扩展等。

参考文献 (References)

- 1 Song G F, Zhong E S. WebGIS——GIS based on Internet [J]. Journal of Image and Graphics, 1998, 3(3):251~254. [宋关福,钟耳顺. WebGIS——基于 Internet 的地理信息系统[J]. 中国图象图形学报,1998,3(3):251~254.]
- 2 Stewart Birnam, Java and distributive system [M]. Beijing: Qinghua University Press, 2002, 8. [Stewart Birnam. Java 与分布式系统 [M]. 孟纯城译,北京:清华大学出版社,2002,8.]
- 3 Liu C J, Zhu X Y. WebGIS based on JSP technology [J]. Bulletin of Surveying and Mapping, 2002, (6):41~43. [刘春菊,朱欣焰. 基于 JSP 技术的 WebGIS [J]. 测绘通报,2002,(6):41~43.]