

采用 ECharts 可视化技术实现的数据体系监控系统^①

冀 潇, 李 杨

(中国电子科学研究院, 北京 100041)

摘 要: 针对如何将大量数据以可视化的形式呈现在用户面前, 从而能够更加清晰的传达信息的问题, 采用了 ECharts 可视化技术实现数据体系监控系统. 详细描述了该数据体系监控系统的系统设计, 并讲述了该系统的具体实现, 从而说明了使用 ECharts 可视化技术以图形化的方式对数据信息进行展示的起着十分重要的作用.

关键词: ECharts; 可视化; 编程语言; 异步调用; 监控

Realization of Data Hierarchy Monitoring System by Using ECharts Visualization

Ji Xiao, Li Yang

(China Academy of Electronics and Information Technology, Beijing 100041, China)

Abstract: Aiming at solving the problem of how to show a large number of data in graphical form, so as to convey information more clearly, the ECharts visualization technology is adopted to realize Data Monitoring System. The design of the monitoring system is described in detail and the specific implementation is also described. Thereby, it is very important to show data information in graphic form by using ECharts visualization technology.

Keywords: ECharts; visualization; programming language; asynchronous invocation; monitor

随着科技的飞速进步和网络技术的快速发展, 人们对数据和信息获取、发布、接收量也在日益增多, 相对于大量的文本信息和数字信息, 以图形图表的形式能够为用户提供更加直观形象的信息内容, 能够加快人们对信息的吸收、消化能力, 增强理解性、降低认知负担^[1]. 因此将可视化技术应用于 Web 环境下, 必然能够促进可视化技术在信息交互方面的普及和发展, 从而达到帮助人们更加迅速有效的认知、掌握并理解信息的目的, 成为目前流行的开发趋势^[2].

本文前端页面开发在使用 JavaScript 语言做的基础上, 引入了 ECharts 插件, 结合 Ajax 异步调用方式动态读取数据库, 将数据信息用可视化的图形界面展示在前台, 实现了对数据体系监控系统的可视化开发工作, 并能够通过良好的界面达到与用户友好交互的目的.

1 系统开发背景

本数据体系监控系统是针对某数据服务体系开发的一个监控系统. 该数据服务体系系统按照“网络中

心、面向服务”的设计思想, 立足于满足未来数据广域分布、海量存储、按需访问、集中管理的需求^[3], 研发出一种数据服务体系, 能够在数据中心和数据访问节点之间实现数据封装、同步、抽取、组织、维护等基础服务, 保证数据的可见性、可访问性和可理解性, 使用户能够无需关心分布结构、存储方式等物理细节, 实现数据的跨系统、跨领域、多层次共享.

数据服务体系以华北数据服务中心(北京)为主中心, 西北数据服务中心(西安)、西南数据服务中心(成都)、华东数据服务中心(上海)、中南数据服务中心(海口)为分中心的两级组网架构, 进行数据服务体系系统的构建, 每一个数据服务中心下分别挂靠 2~3 个数据访问节点, 如所示图 1. 各数据服务中心作为数据服务提供者, 对用户按需提供数据服务.

该数据体系监控系统能够对各异地数据服务中心之间的跨域数据共享与访问的状态进行监控, 从而掌握整个数据服务体系各中心之间的数据传输方向、数据传输时的网络负载值、运行状况等信息. 而在未使

^① 收稿时间:2016-09-18;收到修改稿时间:2016-10-27 [doi:10.15888/j.cnki.csa.005782]

用 ECharts 可视化技术下, 当用户登录该数据体系监控, 查询数据监控信息时, 系统通过表格的形式将各服务中心之间的数据传输详细信息展现在界面上. 如图 2、图 3 所示.

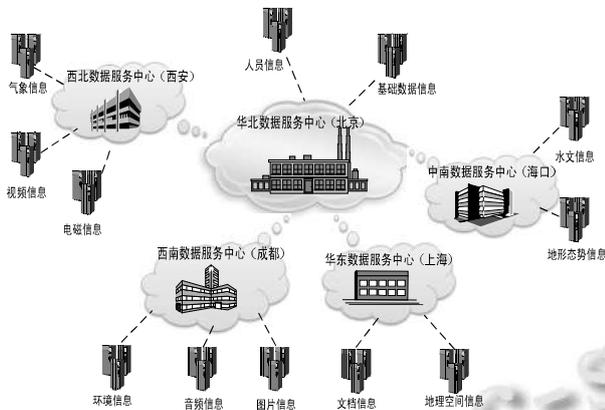


图 1 数据服务体系系统组成图



图 2 数据体系监控系统传输信息监控页面

序号	数据源	数据接收站	网络负载率	传输内容
1	北京	西安	100	基础数据同步
2	北京	上海	10	人员信息同步
3	西安	成都	60	气象数据同步
4	西安	上海	10	视频信息同步
5	西安	海口	90	电磁信息同步
6	上海	海口	60	水文信息同步
7	海口	成都	30	水文信息同步
8	成都	上海	60	环境信息同步
9	海口	北京	20	地理空间信息同步

图 3 数据体系监控系统传输信息一览表

2 技术简介

2.1 JavaScript

JavaScript 是 Net-scape 公司推出的一种基于对象和事件驱动的解释性编程语言^[4]. 其代码在执行时, 首先无需经过编译等工作而是直接发送到客户端, 客

户端接收后再由浏览器来直接执行解释工作. 从代码规范上来看, JavaScript 是一种直接嵌入在 html 中的脚本语言, 其格式不像 java 语言一样严格, 语法相对自由、格式相对松散, 例如在使用某一变量前并不需要对该变量进行严格的格式声明. 它被大量使用在客户端, 主要解决的是客户端的交互问题, 成为网页设计的一项重要技术^[5].

2.2 Ajax

Ajax 即“Asynchronous Javascript And XML”(异步 Javascript 和 XML), 是指一种新兴的网页开发技术, 是一种建立在 JavaScript、XHTML 和 CSS、DOM、HMLHttpRequest、XML 等大量成熟技术基础之上的一项综合技术, 用于创建快速动态网页的技术^[6]. 如果不使用 Ajax 技术而采用以往的传统技术来更新网页内容, 则需要对整个页面进行重新加载, 而采用 Ajax 技术就能够实现网页的异步更新, 即通过很少量的数据传递和交换, 对需要更新的网页内容进行新数据替换.

2.3 ECharts

ECharts 是 Enterprise Charts 缩写, 表示商业级数据表图, 它是一个基于 html5 Canvas 的图标库, 可以流畅的运行在 PC 和移动设备上, 兼容当前绝大部分浏览器(IE6/7/8/9/10/11, chrome, firefox, Safari 等), 底层依赖轻量级的 Canvas 类库 ZRender, 创建了坐标系、图例、提示、工具箱等基础组件, 能够提供直观、生动、可交互、可高度个性化定制的数据可视化图表. 创新的拖拽计算、数据视图、值域漫游等特性大大增强了用户体验, 赋予了用户对数据进行挖掘、整合的能力^[7].

3 系统设计与实现

3.1 数据流设计

用户要获取数据信息生成系统所需的页面, 需要通过登录前台界面触发与后台的交互, 通过 Struts 定位到需要调用的 Action 类中的方法, 并通过该类中的方法读取数据传输信息, 该信息包含了各服务中心之间的数据传输起点、终点、方向、网络负载等, 并以 Json 的格式返回至前台页面后, 再通过 ECharts 可视化方法生成 JSP 文件, 并存放在指定的位置后, 在前端生成可视化界面展现给用户. 其数据流模型图如图 4 所示.

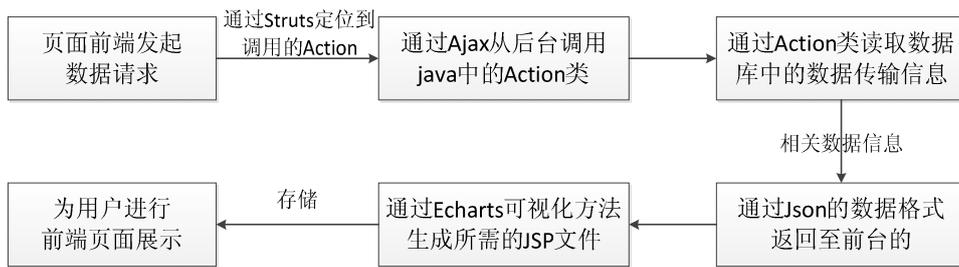


图4 页面获取数据信息生成系统数据流模型

3.2 利用 ECharts 可视化技术的系统实现

JavaScript 作为一种主要的 Web 客户端轻量级脚本语言，主要用于实现动态网页效果以及与 Web 访问者间的简单交互。ECharts 作为一个图表库，能够根据需求定制形象的可视化图表，有良好的交互性，能够被应用到需要数据展现的各类应用场景中。在该数据体系监控的应用过程中，采用 Ajax 异步调用方式，通过调用后台的 java 代码读取所需的数据信息，返回至前台页面后，使用 ECharts 可视化插件，将数据传输信息以图形化的方式呈现在用户面前。

其具体使用配置步骤为：

- (1) 在 html 文件中开辟一个如 div、span 之类的 Dom 元素，用来显示可视化的图表；
- (2) 将 echarts.js 文件引入到<script>标签中，并在此标签中配置可视化图表的使用路径；
- (3) 在<script>标签内加载 dom，初始化 ECharts 图表，对回调函数中的 option 做个性化数值设置^[8]。

具体的流程如图 5 所示。

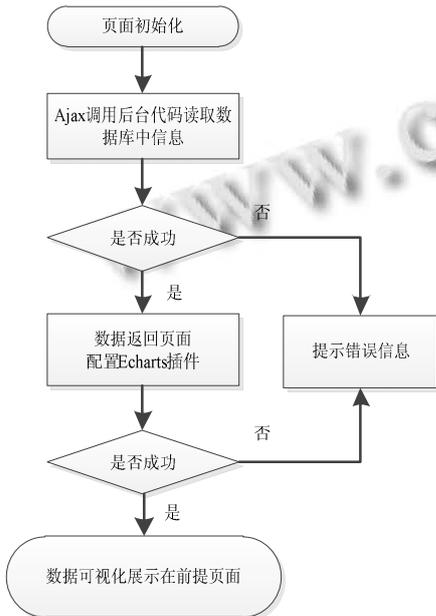


图5 数据信息读取与前台展现流程图

在 index.jsp 中，用户名登录后的 JavaScript 代码实现了以下主要功能：对页面首页进行初始化的过程中采用 Ajax 异步调用方式调用 listMapDataInfo 后台代码，读取数据库中的数据信息，生成 json 格式数据返回到前台页面，存储在 mapdataInfo 返回值当中，mapdataInfo 中包含了数据信息在各数据服务中心之间传输的起始点、终点、网络负载值等。具体代码如下：

```

$(document).ready(function(){
    var v = {}; //ECharts 要存储的数据服务中心信息
                //的对象，初始化为空
    var txt = ""; //存储两个数据服务中心之间传送的信息内容
    $.ajax({
        url:"data/listMapInfo.action", //使用 ajax 调用
                //后台的 listMapInfo 代码，获取数据服
                //务中心信息
        type:"post",
        async:false,
        cache:false,
        dataType:"json",
        success:function(data){
            mapjson = eval('(' + data.mapdataInfo + ')');
            /*返回值 mapdataInfo 中包含了各数据服务中心之间传
            输的数据信息，以 json 格式赋值给变量 mapjson*/
        }
    });
    v['trigger'] = 'item';
    v['formatter'] = function (a){
        /* mark 存储两个数据服务中心之间传送的信息
        内容，并对此信息进行验证，验证后赋予变量 txt*/
        for (var i=0; i<mapjson.length;i++){
            if (a[1] == mapjson[i].mark)
  
```

```

        txt = mapjson[i].info;
    }
    else if(a[1] == mapjson[i].begin||a[1] ==
mapjson[i].end)
        txt = a[1];
    }
    return txt;
}
return v;
});

```

接下来,对 ECharts 中的 option 对象进行设置,将 mapdataInfo 取到的返回值中包括了了华北数据服务中心(北京)、西北数据服务中心(西安)、西南数据服务中心(成都)、华东数据服务中心(上海)、中南数据服务中心(海口)五个数据服务中心,对 markPoint 对象里的 data 属性进行赋值,从而获取数据服务节点的名称和属性信息,用以进行前台界面的展示.具体代码如下:

```

function getMapPointData(){
    var points = []; //存储数据中心节点名称
    var pointnum = mapjson.length;
    for (var i=0; i<pointnum; i++){
        //数据中心起始点
        points.push({name:mapjson[i].begin});
        //数据中心终止点
        points.push({name:mapjson[i].end});
    }
    return points;
}

```

再将 mapdataInfo 中表示五个数据服务中心节点之间的数据传输方向和网络负载的值进行设置,通过对 option 对象中的 markLine 对象里的 data 属性进行赋值,用以进行前台界面的展示,具体代码如下:

```

function getMapDataLine(){
    var mapdetail = []; //存储数据中心节点信息
    if (mapjson.length!=null){
        for (var i=0;i<mapjson.length;i++){
            var t = [];
            //数据中心起始点
            t.push({name:mapjson[i].begin});
            /*end 为数据中心终止点, value 为传输负载率*/
            t.push({name:mapjson[i].end,value:mapjson[i].value});
            mapdetail.push(t);
        }
    }
}

```

最终,用 ECharts 可视化技术呈现出的数据信息传输界面如图 6 所示.根据中心选主策略从体系内的 5 个数据中心中挑选华北数据服务中心(北京)作为主数据中心,同时,主数据中心和其余 4 个数据中心之间形成镜备备份的关系,当主数据中心失效后,系统自动从其余 4 个数据中心重新选出新的主数据中心继续提供服务.在图中可以直观的看到 5 个数据中心在地图中所处的位置,它们之间的连线上动态闪烁点的运行轨迹能够一目了然的看到数据传输的方向,鼠标放在连线上能够显示出节点间的传输内容,局部放大后的图如图 7 所示.

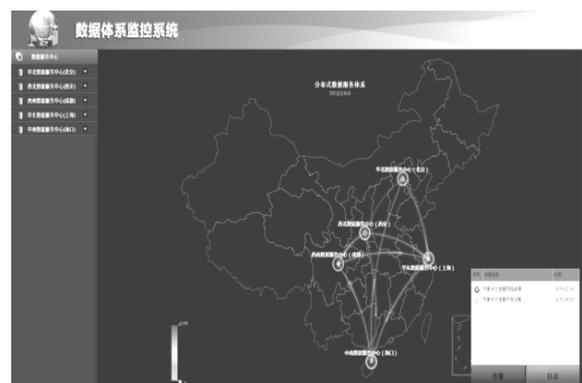


图 6 数据体系监控系统界面展示

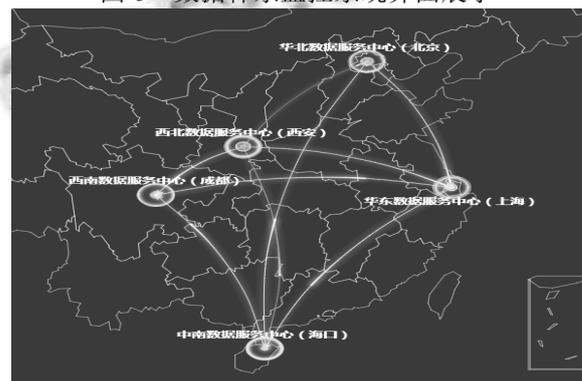


图 7 数据体系监控系统界面局部放大图

本系统拥有气象信息、水文信息、地理空间信息、电磁信息、环境信息等 12 个数据访问节点,分别挂靠在五个不同的数据中心下,部署了相应的数据服务端软件集和数据服务总线服务端软件集,负责直接处理相关各个用户系统提出的数据服务请求,并按照管理

策略接收和存储全局共享数据和相关局部共享数据。其访问节点示意图如图8所示。

在该系统中,还能够用饼状图、柱状图、折线图、力导向布局图等可视化方式对各数据服务中心的具体监控信息进行展示。以华北数据服务中心(北京)为例,点击左侧导航栏或地图上的节点,可以进入该数据服务中心的监控页面,在页面中间能够看到中心利用率、数据容量、访问节点请求数量以及所挂靠的访问节点数量等信息,并能在页面左侧看到关于该数据服务中心的属性信息和告警信息,如图9所示。具体实现技术和代码不再赘述。

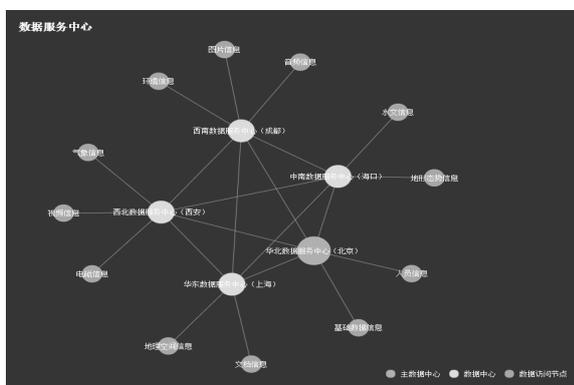


图8 数据访问节点示意图

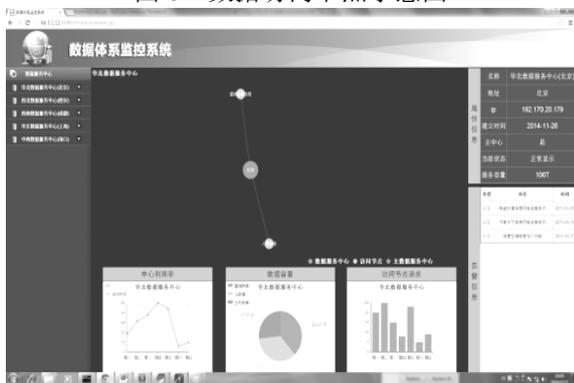


图9 华北数据服务中心监控信息图

4 结语

综上所述,本文通过对数据体系监控系统的展示与分析,可以看出,虽然不是用ECharts可视化技术也能够实现向用户展现数据监测的相关信息,但采用了ECharts可视化技术后,图形化的表现形式更加一目了然,也能够让用户更加清楚、形象、直观的了解到所查询的信息,从而提升用户的体验感,因此它在数据体系监控系统的开发中发挥了重要的作用。

参考文献

- 1 雷莹.基于 Web 的可视化数据挖掘分析平台及可视化度量模型的研究与实现[硕士学位论文].北京:北京交通大学,2014.
- 2 李伟波,鲍苏苏,李健壮.基于 Web 医学图像可视化系统应用研究.计算机与现代化,2012,203(7):92-95.
- 3 汲汾.云计算中心基础设施管理系统的设计与实现[硕士学位论文].天津:天津大学,2012.
- 4 包婧.JavaScript 并行化性能提升的研究与实现[硕士学位论文].上海:上海交通大学,2013.
- 5 冀潇,李杨.JavaScript 与 Java 在 Web 开发中的应用与区别.通信技术,2013,46(6):145-148.
- 6 佟进.基于 Ajax 的定址网络爬虫系统的研究与实现[学位论文].北京:北京邮电大学,2013.
- 7 王锡良,卿光勇,武敬锋.基于 HTML5 的数据图表在公共气象服务中的应用.电脑编程技巧与维护,2015,(15):75-76.
- 8 赵捷,谭国强.基于新浪微博的数据挖掘及可视化研究.电子技术与软件工程,2015,(18):181-182.