

基于 WSN 的室内环境监测中 Ajax 的应用

韩 维, 齐文娟

(武汉科技大学 信息科学与工程学院, 湖北 武汉 430081)

摘 要: 基于 WSN 的室内环境监测系统中, 分析了采用 B/S 模式查询监测结果时页面刷新慢的问题, 并提出采用 Ajax 技术解决上述问题的方法。实验表明: 应用 Ajax 的异步调用机制实现局部刷新, 有效提高了页面刷新速度和 Web 应用响应灵敏度。

关键词: WSN; B/S; Ajax; 异步; 局部刷新

中图分类号: TP391

文献标识码: A

文章编号: 1673-9833(2008)06-0097-03

Application for Ajax in Indoor Environment Monitoring System Based on WSN

Han wei, Qi Wenjuan

(School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China)

Abstract: In the monitoring system of indoor environment based on WSN, a solution by Ajax technology to the problem of low velocity of page refresh while querying the results on the B/S mode is put forward. The experiment shows that asynchronous call techniques of Ajax can realize the partial refresh and improve the speed of page refresh, and also make Web application has a sensitive response.

Key words: WSN; B/S; Ajax; Asynchronous; partial refresh

近年来, WSN 越来越多地用于室内环境监测, 特别是随着 ZigBee 等无线网络通讯协议的飞速发展, 使用 WSN 进行室内环境监测已成为其应用发展的必然趋势。在基于 WSN 的室内环境监测系统中, WSN 完成对监测区域中的相关数据的实时采集, 然而, WSN 没有对数据进行存储和处理的功能, 这就需要将其采集到的数据存储到外部数据库服务器, 客户通过 Internet 对这些数据进行查看和分析。基于 WSN 的室内环境监测系统的基本框架如图 1 所示。系统采集到的数据存储在数据库服务器中, 用户通过 Internet 对数据进行查看和下载。该应用采用 B/S 模式开发, 因为在 B/S 模式下, 客户端不需安装任何专门的软件, 只要有浏览器, 就可以在任何地方远程访问数据, 且不用对系统进行维护和升级, 从而大大减少了系统开销。

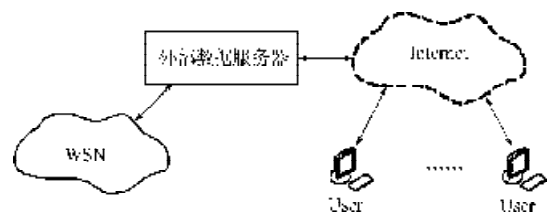


图 1 基于 WSN 的室内环境监测系统的基本框架

Fig. 1 Basic frame of the indoor environment monitoring system

由于用户面临的查询任务比较多, 查询到的数据量比较大, 特别是查询分析图形时, 数据量更大, 传统的 Web 模式页面刷新慢, 大大降低系统的反应速度、影响系统性能。本文使用 Ajax 异步调用机制实现局部刷新, 从而使问题得到解决, 提升用户体验。

收稿日期: 2008-08-24

基金项目: 湖北省教育厅科研项目 (B20071106)

作者简介: 韩 维(1986-), 女, 湖北咸宁人, 武汉科技大学学生, 主要研究方向为无线传感器网络;
齐文娟(1985-), 女, 山西大同人, 武汉科技大学学生, 主要研究方向为无线传感器网络。

1 Ajax 简介

1.1 Ajax 的定义

Ajax (Asynchronous Java Script and XML), 即异步 Java Script 和 XML, 它不是一种新技术, 而是几种蓬勃发展的网页开发技术的整合: 应用 XHTML 和 CSS 实现各种标准化展示; 实现 DOM 动态显示和交互; 应用 XML 和 XSLT 进行数据交换与处理; 应用 XMLHttpRequest 与服务器异步通信; 应用 JavaScript 绑定和处理数据^[1]。

1.2 Ajax 的核心技术

Ajax 的核心技术包括 XML Http Request、Java Script 和 DOM。Ajax 技术的关键就是 XML Http Request 对象的创建与使用, 以实现客户与服务器的异步通信。JavaScript 是用户用来创建并使用 XML Http Request 对象的脚本语言, 它将 Ajax 中的各种技术按一定方式组合在一个协作环境中共同发挥作用。DOM 以树型结构提供 Web 页面的逻辑视图, 把页面元素看成对象。JavaScript 通过 DOM 与页面进行交互, 只刷新改变的元素, 以实现页面的平滑过渡。

1.3 Ajax 工作原理

Ajax 工作原理相当于在用户和服务器之间加一个中间层——Ajax 引擎, 使用户操作与服务器响应异步化^[2]。在图 2 中, 通过传统的 Web 应用模式进行比较, 可以更好的理解其工作原理。

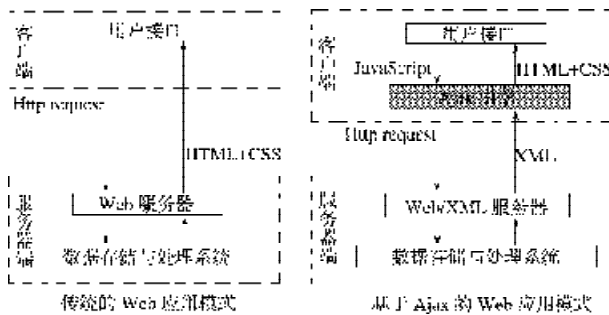


图 2 传统的 Web 应用模式与基于 Ajax 的 Web 应用模式比较

Fig. 2 Comparison between the Web mode based on Ajax and the traditional mode

1) 传统的 Web 应用。用户在界面中操作, 激发相应的事件, 浏览器便向服务器发送 http 请求。服务器进行相应的处理后, 以 html 页面的形式返回给客户端。其过程是同步的, 即每当浏览器向服务器发送请求, 用户都要等服务器响应后, 才能进行其它的操作。且每次交互都会返回一个完整的页面, 即使只是一些很简单的数据交互, 用户都必须等待很长一段时间。

2) 基于 Ajax 的 Web 应用。Ajax 引擎使用户操作与服务器响应异步化。用户激发某个事件后, 通过 JavaScript 调用 Ajax 引擎向服务器发送请求, 并不是所有

的用户请求都提交给服务器, 像一些数据验证和数据处理等都交给 Ajax 引擎自己来做, 只有确定需要从服务器读取新数据时才提交^[3]。请求提交后, 用户还可以在界面上进行其它的操作, 而不需要一直等待服务器响应。服务器做出相应处理后, 以 XML 的形式返回给客户端, 客户端的 Ajax 引擎将其转化为 HTML 文档, 利用 DOM 对象模型, 只对部分页面进行刷新, 节约了大量时间, 提高了工作效率。

2 Ajax 在本系统中的应用

2.1 系统介绍

基于 WSN 的室内环境监测系统的整体框架如图 3。其中, Web Server 和 Database Server 可以是同一台或者分布在异地的远程服务器。

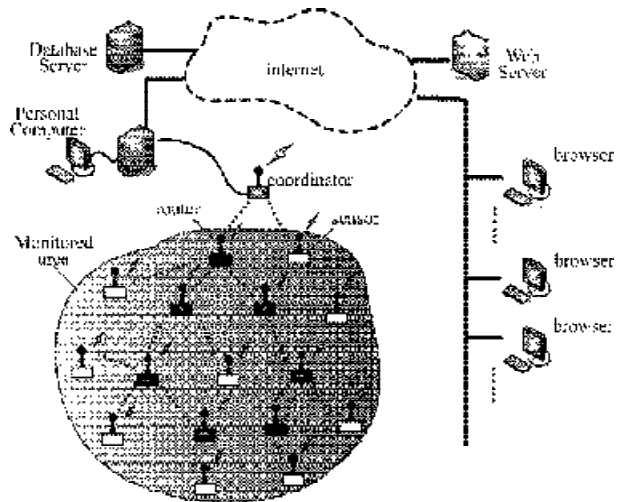


图 3 基于 WSN 的室内环境监测系统的整体框架
Fig. 3 General frame of the indoor environment monitoring system; partial refresh

工作过程: WSN 中包括协调器节点(coordinator)、路由器节点(router)和传感器节点(sensor), 它们以无线方式通信。监测人员对 WSN 进行控制, 将采集到的数据及处理结果存储到 Database Server 中。客户在浏览器中访问 Web Server 中的相应站点, 触发各种事件, 通过 internet 访问服务器中的数据^[4]。

2.2 传统的 Web 模式存在的问题

采用传统的 Web 交互模式, 会造成页面刷新慢, 系统性能差, 严重影响用户体验。主要原因有:

1) 页面元素比较多。本系统用户界面中显示的信息很多, 包括监测区域信息、网络节点信息和采样过程的信息等等。由于界面中 1 个事件请求服务器返回的信息可能只有很小的一部分, 而传统的 Web 方式每次响应都会将整个页面全部刷新。例如当查询单点采样值时, 界面中需要改变的只有显示采样值的文本框的内容, 如果将整个页面都刷新, 不仅做了无用功, 而且对页面刷新速度有很大影响。

2) 查询的数据量比较大。一般要收集许多个周期采样结果, 才能对整个室内环境质量进行比较全面的分析。若要请求查询周期采样获得的数据, 因数据量特别大, 往往需要比较长的时间, 特别是用户请求显示历史曲线图时, 响应更慢。而且用户非得等到服务器响应了才能做下一步的操作, 影响用户体验。

2.3 Ajax 程序框架

基于 Ajax 的基本原理, 针对项目中的关键问题, 在 Web 系统中引入 Ajax 技术。通过逐步的解析, 形成一个发送和接收 XML Http Request 请求的 Ajax 程序框架^[5]。Ajax 实质上也是遵循 Request/Sever 模式, 所以此框架基本的流程也是: 对象初始化→发送请求→服务器接收→服务器返回→客户端接收→修改客户端页面内容。不过该过程是异步的。

1) 捕获浏览器事件。当浏览器中的某些元素发生一些事情时, Web 浏览器就产生 1 个事件。要捕获浏览器的事件, 就是为某个事件注册 1 个处理函数。如: `<input type= "button" name= "Btn_history" value= "显示节点历史数据" onClick= "showhistory();" />`, 并为 1 个表单按钮注册了 1 个鼠标单击的处理函数 `showhistory()`, 当鼠标单击该按钮时, 就会执行 `showhistory()` 这个函数。

2) 初始化 XML Http Request 对象。捕获用户的交互事件后, 若需要与服务器交互, 需使用 XML Http Request。但是, 浏览器不同或者版本不同, 该 XML Http Request 的实例化过程实现方式也不同。IE 以 ActiveX 控件的形式提供, 而 Mozilla 等浏览器则以 XML Http Request 类的形式提供, 所以需要保证程序兼容性, 实现跨浏览器运行。

3) 向服务器发送请求。获得 1 个实例化的 XML Http Request 对象之后, 就可以向服务器发送 1 个 HTTP 的请求。发送之前需要指定响应处理函数, 即服务器响应之后的回调函数 (callback), 我们只需将 XML Http Request 对象的 `on ready state change` 属性指定为回调函数名即可, 如: `XML Http Request Object.on ready state change = call Back Handler`。指定响应处理函数之后, 调用 XML Http Request 对象的 `open` 和 `send` 方法, 就向服务器发送了 1 个 HTTP 的请求。

```
http_request.onreadystatechange = processRequest;
http request.open("GET", url, true);
http_request.send(null);
```

4) 接收服务器响应。在发送 HTTP 请求之前通过 `on ready state change` 属性指定了响应处理函数。该响

应处理函数要做的第一步是检查 XML Http Request 对象的 `ready State` 值, 判断请求目前的状态。当 `ready State` 值为 4 时, 表示服务器已经传回所有的信息。服务器返回信息后, 还需要判断返回的 HTTP 状态码, 以确定返回的页面没有错误。其中 200 代表页面正常。

```
if (http_request.readyState==4)
```

```
if (http request.status==200)
```

接下来就是对返回的数据进行处理。

5) 更新客户端页面内容。XML Http Request 对成功返回的信息有两种处理方式: 一是 `responseText`, 将信息当字符串使用, 一些状态信息等非结构化的数据, 如节点丢失之类的可以用这种方式处理; 一是 `responseXML`, 将信息当 XML 文档使用, 用 DOM 解析处理。一些结构化的数据, 如历史数据的记录, 要填入到 1 个表格中, 可以使用该方式。

3 结语

实验证明, 在基于 WSN 的室内环境监测系统中应用 Ajax 技术后, 解决了页面刷新慢的问题, 实现页面的平滑过渡。例如, 当用户多次查看历史数据时, 数据量很大, Java Script 利用 DOM 只将最新的数据更新到页面上, 用户甚至感觉不到页面已经刷新了。另外, 有效提高了反应速度。例如, 用户发送查询周期采样曲线图的请求后, 可以继续浏览页面中显示的数据, 或者发送查看历史数据等其它请求, 服务器返回曲线图后, 用户再观察周期采样曲线图。

参考文献:

- [1] 夏慧军, 魏雪辉. 深入浅出 Ajax[M]. 北京: 电子工业出版社, 2007: 18-25.
- [2] 褚法政, 周 峰, 孙更新. Ajax 开发技术原理与实践教程 [M]. 北京: 电子工业出版社, 2007.
- [3] Mark O'Neill. Mapping Ajax's weaknesses[EB/OL]. [2007-09-10]. <http://www.infosecurity-magazine.com/features/septo7/sum.html>.
- [4] Paul Ritchie. The security risks of AJAX/web 2.0 applications [J]. Network Security, 2007(3): 4-8.
- [5] Getting started —— Alexis wilke. What's AJAX? [EB/OL]. [2008-08-10]. http://developer.mozilla.org/en/docs/AJAX: Getting_Started#.

(责任编辑: 罗立宇)