文章编号:1008-0570(2006)02-3-0205-03

Ajax 引擎的原理和应用

The principle and application of Ajax engine

(北京交通大学)游丽贞 郭宇春 李纯喜
You,Lizhen Guo,Yuchun Li,Cunxi

摘要:Ajax 是 web 应用的一种新方法。它并不是一门新的语言或技术,实际上是几种已经在各自领域大行其道技术的强强结合,Ajax 混合了 XHTML/CSS,DOM,XML 及 XSLT 等几项技术,并且利用 Javascript 来整合上述技术。Ajax 为交互较多,频繁读数据,数据分类良好的 Web 应用提供了一个很好的解决方案。

关键字:Ajax,XMLHttpRequest,Web 应用,无刷新更新页面
中图分类号:TP391 文献标识码:A

Abstract: Ajax is a new approach to web application . Ajax isn' t a technology. It' s really several technologies, each flourishing in its own right, coming together in powerful new ways. Ajax offers an excellent solution to web applications ,especially, that have rich interaction and responsiveness.

Keywords: Ajax,XMLHttpRequest,Web application,updating web

当前 Web 服务逐渐渗入到人们的日常生活中,越来越多的人通过 Web 享受信息化时代带来的的各种服务。一个热门站点的 Web 服务器每时每刻要处理海量的数据请求。同时我们在浏览 Web 站点的时候,不时的通过链接从一页跳到另一页。我们会发现同一个站点上的许多页面上内容都是重复的, 譬如页头、页尾和广告等。这样一来造成了数据的重复请求,不但加大了服务器的负担,而且造成用户界面的闪烁或白屏。

针对以上问题,2005 年 2 月 Jesse James Garrett 提出了 Ajax 这个概念。Ajax 是 Asynchronous JavaScript and XML 的缩写。Ajax 并不是一门新的语言或技术,它实际上是几项技术按一定方式的组合,在共同的协作中发挥各自的作用,它包括:使用 XHTML 和 CSS 标准化呈现;使用 DOM 实现动态显示和交互;使用 XML 和 XSLT 进行数据交换与处理;使用 XMLHttpRequest 进行异步数据读取;最后用 JavaScript 绑定和处理所有数据。其中 XMLHttpRequest,Javascript 和 DOM 是 Ajax 技术的核心。

1 Ajax 的原理

在经典的浏览器与服务器的交互方式中,由用户触发一个 HTTP 请求到服务器, 服务器对其进行处理后再返回一个新的 Web 页到浏览器,每当服务器处理浏览器提交的请求时,客户都只能空闲等待,并且哪怕只是一次只需从服务器端得到很简单的一个数据的交互,都要返回一个完整的 Web 页,造成用户每次都要浪费时间和带宽去重新读取整个页面。

Ajax 的工作原理相当于在用户和服务器之间加了一个中间层——Ajax 引擎,使用户操作与服务器响应异步化——并不是所有的用户请求都提交给服务器,一些数据验证和处理由 Ajax 自己来做而不必交给服务器处理,只有确定需要从服务器读取新数据时再由 Ajax 引擎代为向服务器提交请求。在使用 Ajax 引擎后,用户从感觉上几乎所有的操作都会很快响应没有页面重载(白屏)的等待。(如图 1)

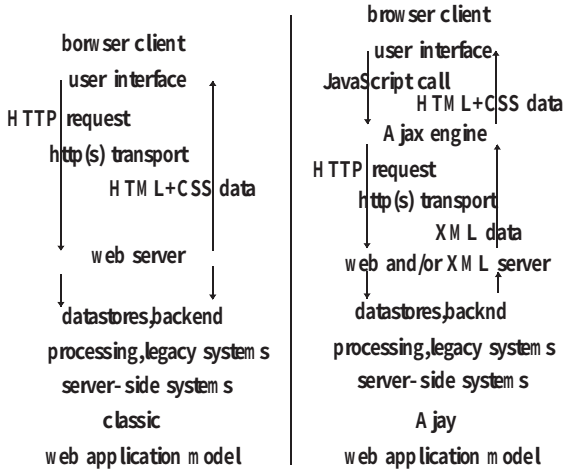

图 1

从(图-1)我们可以看出 Ajax 引擎,实际上是一个比较复杂的 JavaScript 应用程序, 用来处理用户请求然后根据需要动态读写服务器和更改 DOM 内容。以前为了使网页能无缝化重构,也就是在页面已经下载完毕后改变页面内容, 开发人员一直通过 JavaScript

游丽贞:硕士研究生
国家自然科学基金,合同号:60202001

和 DOM 来实现。但是要使网页真正动态起来,不仅要内部的互动,还需要从外部获取数据,在 XMLHTTPRequest出现以前,我们是让用户来输入数据并通过 DOM 来改变网页内容的,但现在 XMLHTTPRequest可以让我们在不重载页面的情况下读写服务器上的数据,使用户的输入达到最少。

下面介绍一下构成 Ajax 引擎的几个主要技术及其在 Ajax 引擎中所起的作用。

1.XMLHttpRequest

在 Ajax 引擎中的几项技术中 XMLHttpRequest是最核心的技术,是 Ajax 引擎解决无需刷新整个页面就可以从服务器获取新数据这个问题的关键所在。

在微软 IE 平台下 XMLHTTPRequest 是 XMLHTTP 组件一个对象,它通过允许开发人员在 Web 页面内部使用 XMLHTTP ActiveX 组件扩展自身的功能,开发人员可以不用从当前的 Web 页面导航而直接传输数据到服务器上或者从服务器取数据。这个功能是很重要的,因为它帮助减少了无状态连接的痛苦,它还可以排除下载冗余 Web 数据的需要,从而提高进程的速度。当然在其他 Web 浏览器平台下,例如 Mozilla (Mozilla1.0 以上),Konqueror 和 Opera(v7.6x+),它们都创建了它们自己的继承 XML 代理类——XMLHttpRequest 类。对于大多数情况,XMLHttpRequest 对象和 XMLHTTP 组件很相似,方法和属性也类似,只是有一小部分属性不支持。

XMLHttpRequest 对象方法

方法	描述
abort()	停止当前请求
getAllResponseHeaders()	作为字符串返问完整的 headers
getResponseHeader("headerLabel")	作为字符串返问单个的 header 标签
open("method","URL"[,asyncFlag[,"userName"[, "password"]]])	设置未决的请求的目标 URL,方法, 和其他参数
send(content)	发送请求
setRequestHeader("label", "value")	设置 header 并和请求一起发送

XMLHttpRequest 对象属性

属性	描述
onreadystatechange	状态改变的事件触发器
readyState	对象状态(integer): 0 = 未初始化 1 = 读取中 2 = 已读取 3 = 交互中 4 = 完成
responseText	服务器进程返回数据的文本版本
responseXML	服务器进程返回数据的兼容 DOM 的 XML 文档对象
status	服务器返回的状态码, 如: 404 = "文件未找到"、200 ="成功"
statusText	服务器返回的状态文本信息

2.DOM (Document Object Model)

DOM 是给 HTML 和 XML 文件使用的一组 API。它提供了文件的结构表述,让你可以改变其中的内容及可见物。其本质是建立网页与 Script 或程序语言沟通的桥梁。所有 Web 开发人员可操作及建立文件的属性、方法及事件都以对象来展现(例如,document 就代表"文件本身"这个对像,table 对象则代表 HTML 的表格对象等等)。这些对象可以由当今大多数的浏览器以 Script 来取用。

一个用 HTML 或 XHTML 构建的网页也可以看作是一组结构化的数据,这些数据被封在 DOM (Document Object Model) 中,DOM 提供了网页中各个对象的读写的支持。

3.JavaScript

JavaScript 是一在浏览器中大量使用的跨平台编程语言,常被用来做一些网页特效或表单验证。在 Ajax 中 Javascript 则是 XMLHttpRequest 和 DOM 交互的桥梁和 Ajax 引擎工作的主要推动力。Javascript 通过调用 XMLHttpRequest 的属性和方法来获取服务端数据,然后调用 DOM 的 API 来更新 Web 页面的内容。实现整个页面的无刷新更新页面。

2 Ajax 优缺点分析

任何一项技术都具有其固有的优点及缺点,对于它们的优点和缺点的事先的认识,有利于将来准确地,有效地利用这项技术。根据以上 Ajax 的原理和技术背景可以总结出 Ajax 有以下优势:

1.减轻服务器的负担

因为 Ajax 的根本理念是"按需取数据",所以最大可能在减少了冗余请求和响影对服务器空间和带宽造成的负担。同时 Ajax 可以把原来需要服务器要做的许多事情放到客户端来做。

我们在上网冲浪的时候可以看到大多数网站的很多页面至少 90% 都是一样的,比如:结构、格式、页头、页尾、广告等,所不同的只是一小部分的内容,但每次服务器都会生成所有的页面再返回给客户端,这无形之中是一种浪费,不管是对于用户的时间、带宽、CPU 耗用,还是对于 ISP 的高价租用的带宽和空间来说。如果按一页来算,只能几 K 或是几十 K 可能并不起眼,但像每天要生成几百万个页面的大 ISP 来说,可以说是损失巨大的。而 Ajax 可以所为客户端和服务器的中间层,来处理客户端的请求,并根据需要向服务器端发送请求,用什么就取什么、用多少就取多少,就不会有数据的冗余和浪费,减少了数据下载总量,而且更新页面时不用重载全部内容,只更新需要更新的那部分即可,相对于纯后台处理并重载的方式缩短了用户等待时间,也把对资源的浪费降到最低,所以 Ajax 对于用户和 ISP 来说是双赢的。

2.刷新更新页面,减少用户实际和心理等待时间

首先,"按需取数据"的理念减少了数据的实际读取量,打个很形象的比方,譬如你现从北京来到了上海,然后你再想去广州。按照经典重载的模式是从上海回到北京再到广州的话,那么 Ajax 的模式就是直接从上海到广州,而不必走已经走过的北京到上海的路。这样一来就减少了用户实际等待时间。

其次,DOM 的使用则不象传统刷新那样出现白屏的情况,而是在读取数据的过程中显示的是原来的页面状态(在实际应用过程中可以加一个 Loading 的提示框让用户了解数据读取的状态),只有当接收到全部数据后才更新相应部分的内容,而这种更新也是瞬间的,用户几乎感觉不到。

3.基于标准化的并被广泛支持和技术,并且不需要插件或下载小程序

Ajax 里面的每个技术都是基于标准化的并被广泛支持的技术,并且不需要插件或下载小程序。

4 Ajax 可以调用外部数据

Ajax 由于可以调用外部数据,也可以实现数据聚合的功能(当然要有相应授权),比如微软刚刚在 3 月 15 日发布的在线 RSS 阅读器 BETA 版;还可以利于一些开放的数据,开发自己的一些应用程序,比如用 A-mazon 的数据作的一些新颖的图书搜索应用。

5.Ajax 使 Web 中的界面与应用分离(也可以说是数据与呈现分离)

Ajax 在整个 Web 服务系统的位置决定了 Ajax 引擎只要从服务端获取 XML 或者其他格式的数据,便可定制整个 Web 界面,从而可以使的服务端只注重数据逻辑处理而不必关心 Web 界面的呈现,将数据呈现的工作交给 Ajax 引擎来做,这样有利于分工合作、减少非技术人员对页面的修改造成的 Web 应用程序错误、提高效率、也更加适用于现在的发布系统。

同时 Ajax 也存在一些的问题,例如:

1.一些手持设备(如手机、PDA 等)现在还不能很好的支持 Ajax。

2. 开发过程中用 JavaScript 作的 Ajax 引擎,JavaScript 的兼容性和 DeBug 都是让人头痛的事;另外因为这个架构刚刚提出,所以其 IDE 环境还未出现。

3.Ajax 的无刷新重载,由于页面的变化没有刷新重载那么明显,所以容易给用户带来困扰——用户不太清楚现在的数据是新的还是已经更新过的。不过对于这个问题,我们可以在相关位置做出提示,把数据更新的区域设计得比较明显,在数据更新后给用户提示。

4.对流媒体的支持没有 Flash、Java Applet 好。

3 应用

下面举个关于典型 Ajax 应用——树形菜单。我们在上网时可以看到许多树形菜单的应用,例如微软站点上的 http://msdn.microsoft.com/library/。我们以前的对于树形菜单的开发过程是这样的:

为了避免每次对菜单的操作引起的重载页面,不采用每次调用后台的方式,而是一次性将树形菜单的所有数据全部读取出来并写入数组,然后根据用户的操作用 JavaScript 来控制它的子集项目的呈现,这样虽然解决了操作响应速度、不重载页面以及避免向服务器频繁发送请求的问题,但是如果用户不对菜单进行操作或只对菜单中的一部分进行操作的话,那读取的数据中的一部分就会成为冗余数据而浪费用户的资源,特别是在菜单结构复杂、数据量大的情况下(例如微软 MSDN 站点),这种弊端就更为突出。

如果应用 Ajax 后,结果就会有所改观:在初始化页面时我们只读出它的第一级的所有数据并显示,在用户操作一级菜单其中一项时,会通过 Ajax 向后台请求当前一级项目所属的二级子菜单的所有数据,如果再继续请求已经呈现的二级菜单中的一项时,再向后面请求所操作二级菜单项对应的所有三级菜单的所有数据,以此类推.....这样,用什么就取什么、用多少就取多少,就不会有数据的冗余和浪费,减少了数据下载总量,而且更新页面时不用重载全部内容,只更新需要更新的那部分即可,相对于后台处理并重载的方式缩短了用户等待时间,也把资源的浪费降到了最低。

4 将来的研究

Ajax 是 web 应用方面的一个重要的概念,其应用也会越来越广,譬如 google,微软均已经在自己的站点或产品中应用了 Ajax。但是我们不应该对这种技术存在盲目的推崇而是应该也有我们自己的想法,Ajax 应用的最大的挑战不是技术方面的,因为 Ajax 所涉及的核心技术都已经很成熟了,它主要的挑战是 Ajax 应用的设计者:应该超越已有的思维定式,忘记已了解的 web 的限制,展开丰富的想象,把 Ajax 技术推广到更广的范围更宽的层面。

参考文献

[1][Jesse James Garrett. Ajax: A New Approach to Web Applications [EB/OL].http://www.adaptivepath.com/publications/essays/archives/000385.php

[2]车银超,刘冰.SNMPv3 安全机制的实现与性能分析[J].微计算机信息,2005,8-3:7-10

作者简介:游丽贞,(1981-),女,福建南平人,北京交通大学通信与信息系统专业硕士研究生,主要研究方向:复杂网络研究;郭宇春,(1968-),女,山西晋城人,北京交通大学 电子信息工程学院,副教授,硕士 主要研究方向:网络测量与建模,Qos 路由;李纯喜,(1970-),男,辽宁锦西人,北京交通大学 电子信息工程学院,讲师,硕士,主要研究方向:网络测量与建模,网络性能分析。(100044 北京交通大学通信工程实验室) 游丽贞 郭宇春 李纯喜

(Communication engineering lab,Beijing JiaoTong university,Beijing 100044,China) You Lizhen, Guo Yuchun, Li Chunxi

(投稿日期:2005.7.10) (修稿日期:2005.7.18)

技术创新