

AJAX 技术中 Session 服务的改进

陈 锦, 张建军

(西北大学 信息科学与技术学院, 陕西 西安 710127)

摘 要: AJAX 是近来兴起的一种 WEB 编程技术。事实上, AJAX 不是一种技术, 而是几种技术的集合。文中先简要介绍了 AJAX 的技术, 接着讨论了 AJAX 技术对原有的 session 所造成的潜在的危险: 定时刷新的 XMLHttpRequest 导致 session 永不过期; 同时刷新引起对 session 请求的竞态条件, 同时提出了解决的办法。

关键词: AJAX; session; 改进

中图分类号: TP399

文献标识码: A

文章编号: 1673-629X(2006)12-0076-03

Improvement of Session Service in AJAX

CHEN Jin, ZHANG Jian-jun

(College of Information Science and Technology, Northwest University, Xi'an 710127, China)

Abstract: Ajax is one of the recently Web programming techniques. In fact, Ajax is not a technique, but several technology assembly. Briefly introduced Ajax technologies. Then discussed Ajax technology on the session service course the potential danger: on time refresh XMLHttpRequest cause session never expired; renovated at the same time causes the session request race condition. At last, proposed the ways to solve the problems.

Key words: AJAX; session; improvement

0 引 言

AJAX 用来描述一组技术, 它使浏览器可以为用户提供更为自然的浏览体验。AJAX 提供与服务器异步通信的能力, 从而使用户从请求/响应的循环中解脱出来。借助于 AJAX, 可以在用户单击按钮时, 使用 JavaScript 和 DHTML 立即更新 UI, 并向服务器发出异步请求, 以执行更新或查询数据库。当请求返回时, 就可以使用 JavaScript 和 CSS 来相应地更新 UI, 而不是刷新整个页面。最重要的是, 用户甚至不知道浏览器正在与服务器通信: WEB 站点看起来是即时响应的。但是, AJAX 是一种新出现的技术, 它的发展还不是很完善。例如: 因为 AJAX 的异步特性, 使得 WEB 的 session 服务产生了永不过期和竞态条件的问题, 文中就是对这个问题的产生原理进行了分析, 并讨论了解决的方法。

1 AJAX 简介

1.1 AJAX 定义

AJAX 是 Asynchronous JavaScript and XML (异步 JavaScript 和 XML) 的缩写, 它其实是多种技术的综合, 包括 JavaScript, XHTML 和 CSS, DOM, XML 和 XSTL,

XMLHttpRequest^[1]。其中:

- * 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 XSTL 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。

1.2 AJAX 的工作原理

与传统的 WEB 应用不同, AJAX 采用异步交互过程。AJAX 在用户与服务器之间引入一个中间媒介, 从而消除了网络交互过程中的处理—等待—处理—等待缺点。用户的浏览器在执行任务时即装载了 AJAX 引擎。AJAX 引擎主要用 JavaScript 语言编写, 通常藏在一个隐藏的框架中。它负责编译用户界面及与服务器之间的交互。AJAX 引擎允许用户与应用软件之间的交互过程异步进行, 独立于用户与网络服务器间的交流。现在, 可以用 JavaScript 调用 AJAX 引擎来代替产生一个直接的 HTTP 请求的用户动作, 内存中的数据编辑、数据校验这些不需要重新载入整个页面的需求可以交给 AJAX 来执行。

AJAX 最核心的理念是, 传统应用中的浏览器直接和服务器交互, 现在在中间夹了一层 Script。也就是说, 原来的 Browser-Server 架构, 现在是 Browser-Ajax 引擎-Server (如图 1 所示)。假如是 100% 纯粹的 AJAX 应用, 浏览器都只向 AJAX 引擎发送消息, AJAX 引擎使用 XmlHttpRequest 向服务器发送请求, 然后服务器在 XmlHttpRequest 的回复中带上相关消息, 最后 AJAX 引擎分

收稿日期: 2006-04-04

作者简介: 陈 锦 (1981-), 男, 湖南长沙人, 硕士研究生, 主要研究方向为分布式系统; 张建军, 副教授, 硕士研究生导师, 主要研究方向为数据库、计算机网络和因特网技术、WWW 技术、多媒体通讯等。

析这些消息,用 HTML DOM 模型处理界面。如此,理论上可以完全消除按页刷新的需要。由于存在这样一个在后台的通讯机制,原有开发 WEB 程序时碰到的那些问题自然内部迎刃而解。最重要的是,使用了 AJAX 框架不会和原有开发模型相冲突,如果处理得当的话 AJAX 的优势是显而易见,加上兼容性,一定会让它有机会在 WEB 领域中占一席之地^[2]。

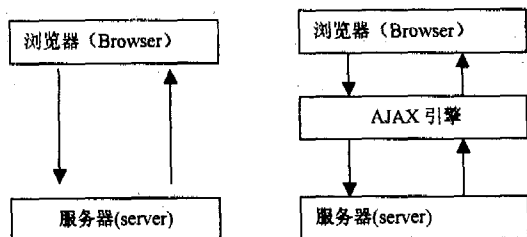


图 1 (a) 传统的 Browser-Server (b) Browser-AJAX-Server

1.3 AJAX 的优点

和传统的 WEB 服务相比,AJAX 具有明显的优势:

- * 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应对服务器造成的负担;

- * 非整页的刷新更新页面,减少用户实际和心理等待时间;

- * 更好的用户体验;

- * 也可以把以前的一些服务器负担的工作转嫁到客户端,利于客户端闲置的处理能力来处理,减轻服务器和带宽的负担,节约空间和带宽租用成本;

- * AJAX 可以异步调用外部数据;

- * 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;

- * AJAX 使 WEB 中的界面与应用分离(也可以说是数据与呈现分离);

- * 对于用户和 ISP 来说是双赢的。

2 JAX 技术所引发的 session 问题

2.1 session 介绍

session,中文经常翻译为会话,其本来的含义是指有始有终的一系列动作/消息。session 机制是一种服务器端的机制,服务器使用一种类似于散列表的结构(也可能就是使用散列表)来保存信息。

当程序需要为某个客户端的请求创建一个 session 的时候,服务器首先检查这个客户端的请求里是否已包含了一个 session 标识——称为 session id,如果已包含一个 session id 则说明以前已经为此客户端创建过 session,服务器就按照 session id 把这个 session 检索出来使用(如果检索不到,可能会新建一个),如果客户端请求不包含 session id,则为此客户端创建一个 session 并且生成一个与此 session 相关联的 session id,这个 session id 将被在本次响应中返回给客户端保存^[3]。

当一次会话结束以后,除非程序通知服务器删除一个

session,否则服务器会一直保留,程序一般都是在用户注销的时候发个指令去删除 session。然而浏览器从来不会主动在关闭之前通知服务器它将要关闭,因此服务器根本不会有机会知道浏览器已经关闭。由于关闭浏览器不会导致 session 被删除,迫使服务器为 session 设置了一个失效时间,当距离客户端上一次使用 session 的时间超过这个失效时间时,服务器就可以认为客户端已经停止了活动,才会把 session 删除以节省存储空间。

2.2 AJAX 的无刷新原理

传统的 WEB 应用采用同步交互过程,这种情况下,用户首先向 HTTP 服务器触发一个行为或请求的呼求。反过来,服务器执行某些任务,再向发出请求的用户返回一个 HTML 页面。这是一种不连贯的用户体验,服务器在处理请求的时候,用户多数时间处于等待的状态,屏幕内容也是一片空白^[4]。

无刷新更新页面工作方式,减少用户心理和实际的等待时间。特别地,当要读取大量的数据的时候,不用像 Reload 那样出现白屏的情况,AJAX 使用 XMLHttpRequest 对象发送请求并得到服务器响应,在不重新载入整个页面的情况下用 JavaScript 操作 DOM 最终更新页面。所以在读取数据的过程中,用户所面对的不是白屏,是原来的页面内容(也可以加一个 Loading 的提示框让用户知道处于读取数据过程),只有当数据接收完毕之后才更新相应部分的内容。这种更新是瞬间的,用户几乎感觉不到。总结起来就是:多次交互,每次少量更新。

2.3 定时刷新的 XMLHttpRequest 导致 session 永不过期

由于 AJAX 的工作方式是多次交互,每次少量更新,这就需要每隔不久的时间 XMLHttpRequest 都要向服务器发送请求。而 session 都有一个固定的过期时间,但是每隔不久的时间 XMLHttpRequest 向服务器发送请求,这就使得服务器为 session 设置的失效时间失效,从而引发安全问题。

要解决这个问题的方法就是编写自定义的客户端 session 监视机制,设计思路就是用一个 JS 的计时器监视距 session 过期剩余的时间,并提前用友好的方式提醒用户保存数据或者激发一个延续 session 的操作。

一开始,就利用 AJAX 向服务器发送一个请求,看 session 是否还存在,如果还在的话,就通知用户进行更新,如果不在的话,服务器也向用户发出通知,告诉用户 session 不存在。如果用户更新 session,则向服务器产生一个 XMLHttpRequest 更新 session,当应答回到客户端的时候,随即产生一个计时器。这个脚本同时还监视消息(通知用户更新)出现在屏幕上的时间。当消息在屏幕上出现的时间过长,消息将会改变(即空闲消息),通知用户他们的 session 将要过期。这样脚本将更加的人性化,让用户满意。

表 1 就是自定义的 session 监视器中可能用的属性及属性值。

表 1 自定义 session 监视器中的属性及属性值

属性	代码	默认属性值
确定框标题	ajaxTimer.title="Your Title";	超时通知
确定框内容	ajaxTimer.message="Your session is ending!";	你目前的 session 已经超时……
更新按钮文本	ajaxTimer.confirm="Update";	确定
取消按钮文本	ajaxTimer.ignore="Cancel";	忽略
最大空闲时间	ajaxTimer.maxWait=1000 * 60 * 5;	5 分钟
空闲消息	ajaxTimer.extendedMessage="You lost your data!";	你的 session 已经到期……

2.4 同时刷新引起对 session 请求的竞态条件

竞态条件是一个在设备或者系统试图同时执行两个操作的时候出现的不希望的状况,但是由于设备和系统的自然特性,为了正确地执行,操作必须按照合适顺序进行。在网络中,竞态条件会在两个用户同时试图访问同一个可用信道的时候发生,在系统同意访问之前没有计算机能得到信道被占用的提示。

编写传统的 WEB 页时,每个页面都有自己唯一的请求、唯一的执行环境,并且用户每一次都只访问一个页面。每个页面的请求都来自于一个唯一的用户,而且每个请求到达服务器的时间都有先后顺序,不和其他的页面请求共享数据。

使用 AJAX 的时候,这一切都改变了,由于 AJAX 的多次刷新,一次页面访问就可能同时产生多次的请求。不同页面的进程不能共享数据,但是同一个页面的异步请求则可以共享数据。如图 2 所示,一个页面装载时产生了有效的 session 数据,在初始化时产生了两个异步的 AJAX 请求,几乎在同一时间,一个请求想要读取 session 中的某一数据,另一个请求想要向 session 中改写这个数据的值,这就产生了竞态条件,这时,你不知道是哪个请求最先到达服务器端,最后这个 session 读取的数据的值只能由到达服务器的时间的先后来决定了,造成了 session 的数据值不确定^[5]。

产生这个问题的原因是网络传输的不确定性,这个问题的解决有两种方法。

第一,可以把这个问题的决定权交给用户。如上所示,可以给每个 Request 添加一把锁,对 session 中要访问的资源锁定,如果 Request1 先到达服务器,当 Request2 到达服务器时,会发现 Request2 所要的变量已经被 Request1 锁定,这时,Request2 返回一条消息,让用户决定是否继续 Request2 的写操作,同样,如果 Request2 先到达,则返回消

息,让用户决定是否继续 Request1 的读操作。但是,这种解决方法牺牲了用户的友好性,如果产生的消息太多的话,用户会很反感,可行性不大。

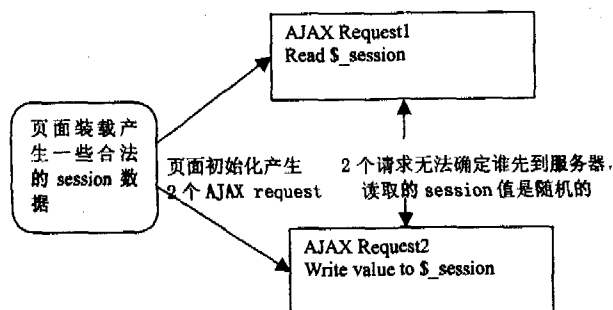


图 2 AJAX 引起 session 随机取值

第二,既然问题产生的根源是网络传输的不确定性,那么,可以去掉网络传输,把 session 从服务器端搬到客户端来,这样,所有的问题就都迎刃而解了。事实上,现在 Client/SOA 模式就是让 session 在客户端进行管理,只不过现在所用的大多是 HTML + 模式而 Client/SOA 模式离成熟应用还需要不少努力。所以,在目前最好的方法还是避免同时的产生请求对 session 的同一数据进行操作。

3 总结

文中主要研究了 AJAX 技术中对原有的 session 所造成的危险,从原理上进行了介绍并指出了解决的办法。对 AJAX 技术进行了总体的介绍,对 AJAX 的优点做出了概括。相信随着时间的推移,AJAX 技术会进一步得到改进,越来越多的网站将会使用 AJAX 技术,使得 AJAX 能够在 WEB 中站有一席之地。

参考文献:

- [1] Grance D, Pascarello E, James D. Ajax in Action[M]. USA: Manning, 2005.
- [2] 夏 桅. AJAX with ASP.NET[J]. MSDN 开发精选, 2005 (4): 230-231.
- [3] Session 详解[EB/OL]. 2005-03-12. <http://dev2dev.bea.com.cn/bbs/jishudata/ArticleShow.jsp?Id=10>.
- [4] 徐雪霖. Web 数据库访问技术探悉[J]. 微计算机信息, 2004, 20(20): 86-87.
- [5] Troubles with Asynchronous Ajax Requests and PHP Sessions [EB/OL]. 2005-03-22. <http://www.chipmunkninja.com/article/asynsessions>.

(上接第 75 页)

- [2] Knorr E, Ng R. Algorithms for mining distance-based outliers in large datasets[C]// In: Proc of the 24 VLDB Conf. New York, USA: [s. n.], 1998: 392-403.
- [3] 史东辉, 蔡庆生, 倪志伟, 等. 基于规则的分类数据离群挖掘方法[J]. 计算机研究与发展, 2000, 37(9): 1094-1100.
- [4] 范 明, 孟小峰. 数据挖掘概念与技术[M]. 北京: 机械工业出版社, 2001: 223-261.
- [5] Edwin K, Roymond N. Algorithms for mining distance-based outliers in large database[C]// In: Proc of the VLDB Conf. New York: [s. n.], 1998: 392-403.
- [6] Arning A, Agrawal A, Raghavan P. A linear method for deviation detection in large database[C]// Int Conf on knowledge Discovery in Databases and Data Mining. Portland: [s. n.], 1996: 169-184.