

基于 XML Web Service 和 Ajax 的 Web 实时分布式系统的架构及应用

Construction and application of real-time and distributed Web system based on XML Web Service and Ajax

李强 陈遵德 (顺德职业技术学院 计算机技术系 广东顺德 528300)

摘要:如果 WEB 系统的用户数比较多、数据计算比较复杂、数据库访问量比较大,人机交互量比较大,就会使传统的 Web 系统无法满足用户的实时性要求。通过 XML Web Service 可以均衡 Web 服务器的负载,创建分布式 Web 系统;通过 Ajax 可以创建无需等待、无需刷新的 Web 交互模式。本文提出了基于 XML Web Service 和 Ajax 的 Web 实时分布式系统的架构方案,并进行了具体应用。

关键词:XML Web Service Ajax 实时 分布式

1 引言

由于网络的普及以及 Web 系统相较桌面系统的一些优势,如不需安装客户端,系统维护方便等,越来越多的系统采用 B/S 模式,但是如果 Web 系统的用户数比较多、数据计算比较复杂、数据库访问量比较大,人机交互量比较大,就会对 Web 系统的实时性形成极大的挑战。传统的大学英语 Web 考试系统就面临实时性问题,如大学英语 A、B 级 Web 机试系统为了达到实时效果,每个机房配备一台服务器,这样导致系统安装、管理、维护工作量大大增加,失去了 Web 系统具有的优势。



图 1 传统 Web 系统结构模型

基于 XML Web Service 的 Web 分布式系统可以均衡 Web 服务器负载,提高 Web 系统的实时性,但是无

法解决人机交互产生的交互延时^[1]。通过 Ajax 可以创建无需等待、无需刷新的 Web 交互模式,本文提出了基于 XML Web Service 和 Ajax 的 Web 实时分布式系统的架构方案,给出了系统架构模型及实现方法,并进行了具体应用,实现了实时分布式的大学英语 Web 考试系统。

2 系统架构模型

2.1 基于 XML Web Service 的 Web 系统结构模型

传统的 Web 系统结构模型如图 1 所示,Web 服务器为浏览器用户提供 Web 服务,如有需要,数据库服务器为 Web 服务器提供数据库服务。对于访问用户数少,数据计算量小,数据库访问量小,人机交互数据量小的系统,传统的 Web 系统结构模型已经能够胜任,但是对于访问用户数多,数据计算量大,数据库访问量大,人机交互数据量大的系统,传统的 Web 系统结构模型将会产生较长延时,无法满足用户的实时性要求。

XML Web Service 是一种基于 Web 协议提供函数功能,并以 XML 文档形式返回函数执行结果的技术,XML Web Service 使创建分布式 Web 系统非常方便^[1]。

基于 XML Web Service 的 Web 系统结构模型如图

2 所示。Web 服务器为浏览器用户提供 Web 服务, XML Web Service 服务器为 Web 服务器提供数据计算或数据库访问服务, Web 服务器会根据系统设置, 均衡各个 XML Web Service 服务器的工作量。如有需要, 数据库服务器为 XML Web Service 服务器提供数据库服务。通过调整 XML Web Service 服务器数量可以满足 Web 系统的不同实时性要求^[2]。

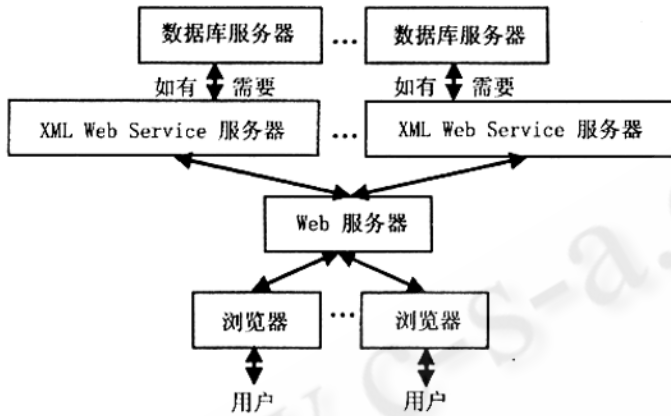


图 2 基于 XML Web Service 的 Web 系统结构模型

2.2 基于 Ajax 的 Web 系统交互模型

传统的浏览器与 Web 服务器交互模型如图 3 所示, 浏览器通过请求服务器页面取得 Web 服务器的响应, 浏览器与 Web 服务器的工作过程是一个同步的过程。浏览器的工作过程是: 首先向服务器发出服务器页面请求, 然后等待服务器页面的执行结果, 最后显示服务器页面的执行结果。WEB 服务器的工作过程是: 首先响应浏览器的页面请求, 然后执行服务器页面中的数据计算、数据库访问, 最后生成执行结果。

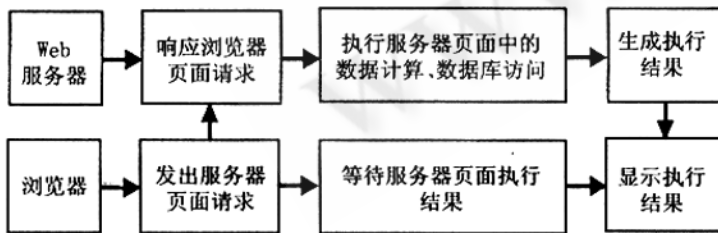


图 3 传统 Web 系统交互模型

Ajax 即异步 JavaScript 和 XML (Asynchronous JavaScript and XML), 它是现有的 DHTML、CSS、COM、XML 等技术的综合应用。其核心思路是通过浏览器端

脚本语言 (JavaScript) 以异步方式向 Web 服务器发出服务器页面请求, Web 服务器执行服务器页面后生成 XML 文档形式的执行结果并通知浏览器, 浏览器端脚本语言再根据执行结果更新部分页面^[3]。

基于 Ajax 的浏览器与 Web 服务器交互模型如图 4 所示, 浏览器与 Web 服务器的工作过程是一个异步的过程。浏览器的工作过程是: 首先通过脚本语言 (JavaScript) 向服务器发出异步服务器页面请求, 然后继续人机交互, 而无需停止工作等待服务器的处理过程, 最后通过脚本语言根据服务器执行结果更新部分页面。Web 服务器的工作过程是: 首先响应浏览器端脚本语言的页面请求, 然后执行服务器页面中的数据计算、数据库访问, 最后生成 XML 文档形式的执行结果^[4]。

2.3 基于 XML Web Service 和 Ajax 的 Web 系统结构模型

基于 XML Web Service 和 Ajax 的 Web 系统结构模型如图 5 所示, XML Web Service 服务器均衡 Web 服务器的负载, 浏览器直接与 XML Web Service 服务器进行无须等待、无须刷新的异步交互。XML Web Service 和 Ajax 结合可以综合应用两种技术的优点, 有效创建实时分布式 Web 系统。

3 系统架构的实现

基于 XML Web Service 和 Ajax 的 Web 系统架构的实现包含两大部分, 一部分是基于 XML Web Service 的系统结构实现, 一部分是基于 Ajax 的浏览器与 Web 服务器交互模型的实现。

3.1 基于 XML Web Service 的 Web 系统结构的实现

基于 XML Web Service 的系统结构的实现有如下几个步骤:

(1) 根据 WEB 系统的实时性要求和实际工作负载确定 XML Web Service 服务器数量, 方法如下:

① 确定 Web 系统的实时性要求, 即 Web 系统所能允许的最高访问延时 T_{reqmax} 。

② 确定 Web 系统中工作量最大的页面 P_{max} , 采用传统 Web 系统结构模型实现 P_{max} , 粗略计算其满负荷工作时产生的最大延时 T_{locmax} 。计算方法一

般采用单浏览器循环方法, 即将页面工作循环最大访问人数次, 取得访问前和访问完成后的时间差, 即其粗略的最大延时 T_{locmax} 。

③ 根据 T_{reqmax} 和 T_{facmax} 计算粗略的 XML Web Service 服务器数量, 计算规则是用 T_{facmax} 除以 T_{reqmax} 即为粗略的 XML Web Service 服务器数量。

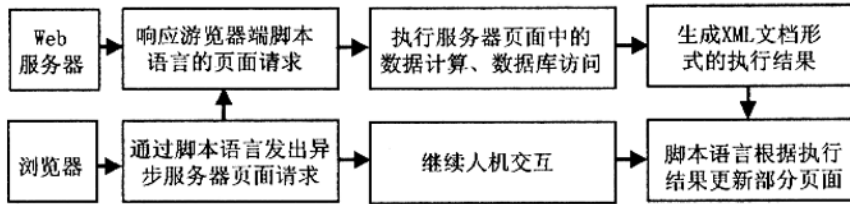


图 4 基于 AJAX 的 WEB 系统交互模型

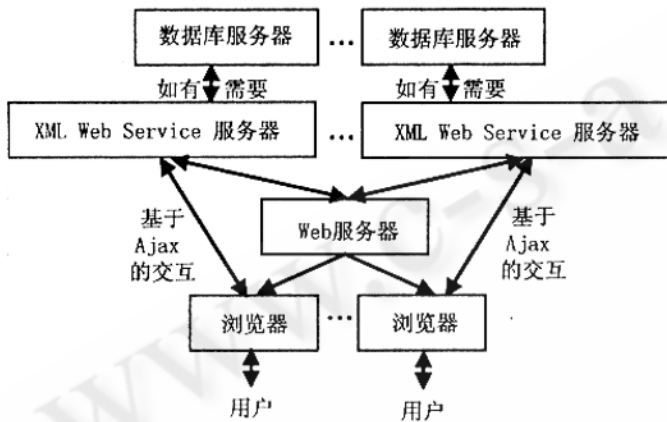


图 5 基于 XML Web Service 和 Ajax 的 Web 系统结构模型

④ 根据所计算的 XML Web Service 服务器数量, 采用基于 XML Web Service 的 Web 系统结构模型实现 P_{max} 。

⑤ 粗略计算 P_{max} 满负荷工作时产生的最大延时 T'_{facmax} ; 如果 T'_{facmax} 大于 T_{reqmax} , 则适量增加 XML Web Service 服务器数量, 直到 T'_{facmax} 小于等于 T_{reqmax} 为止。

(2) 如有需要, 在数据库服务器上实现 XML Web Service 所需要的数据库。

(3) 确定 WEB 服务器所需要的 WEB 服务函数集, 并在 XML Web Service 服务器上实现该函数集。

(4) 基于 XML Web Service 服务器所提供的 Web 服务函数集, 在 Web 服务器上实现 Web 系统页面。

3.2 基于 Ajax 的 Web 系统交互模型的实现

基于 Ajax 的浏览器与 Web 服务器交互模型的实现有如下几个步骤:

(1) 基于浏览器端脚本语言实现 Web 服务的请求程序。具体过程是首先创建 XMLHttpRequest 对象, 然后使用

XMLHttpRequest 对象发出异步方式的 Web 服务请求^[5]。

(2) 基于浏览器端脚本语言实现 Web 服务执行结果的 XML 文档解析程序。具体过程是首先使用 XMLHttpRequest 对象取得 Web 服务的 XML 文档形式的执行结果, 然后创建 XMLDocument 对象, 并使用该对象解析 XML 文档^[5]。

(3) 基于浏览器端脚本语言实现页面的更新程序。具体过程是根据执行结果利用 DHTML、CSS 技术通过设置网页元素的属性更新部分页面。

4 应用实例

本文基于 XML Web Service 和 Ajax 实现了实时分布式英语 Web 考试系统, 该系统采用基于 XML Web Service 的系统结构和基于 Ajax 的系统交互模型, 并采用了其他一些提高实时性的措施, 系统现在已经投入使用并取得了较好应用效果。

4.1 基于 XML Web Service 的英语 Web 考试系统结构

为了满足系统的实时性要求, 系统总共使用 4 台服务器, 其中一台作为 Web 服务器, 3 台作为 XML Web Service 服务器及其对应的数据库服务器。3 台 XML Web Service 服务器具有相同的 Web 服务函数集, 相应的数据库服务器具有相同的库结构。Web 服务器的数据计算任务和数据库访问任务全部调用 XML Web Service 服务器完成, 并且 Web 服务器通过考试班级数均衡各个 XML Web Service 服务器的工作负载。

4.2 基于 Ajax 的英语 Web 考试系统交互模型

英语 Web 考试系统中的考生答题页面全部基于 Ajax 交互模型。具体实现是每当考生完成一个答题操作, 则触发一个 JavaScript 事件处理程序, 该事件处理程序发出异步方式的 Web 服务请求, 并传递考生答题信息参数。同时考生可以继续答题, 相应的 Web 服务将考生答题信息存放到数据库中。WEB 服务执行完成后, 将触发 JavaScript 页面更新程序, 该页面更新程序根据执行结果更新部分页面内容。

4.3 其他一些提高系统实时性的措施

系统还采用了其他一些措施提高系统实时性, 包括:

(1) 通过增加数据冗余减少数据库访问次数。如系统中通过将题库中的题目复制到考生数据库中, 从而减少了考生访问题库数据库的次数。

(2) 既能放在用户程序端, 又能放在后台管理端

的数据计算任务,尽量放在后台管理端。如系统中将组卷工作任务放在后台管理端实现,这样就减少了用户程序端因完成组卷任务带来的延时。

(3) 尽量使用带连接的数据库访问方式。一次无连接的数据库访问耗时大约是一次带连接的数据库访问的 3 倍,并且无连接的数据库访问较带连接的数据库访问占用更多的系统内存。

4.4 系统应用效果

系统现在已经投入使用,已经约有 6000 学生使用该系统进行英语考试。系统的实时性较高,可以让 2500 学生同时考试,并且每一位考生考试过程中的最大延时不超过 2 秒。

5 结束语

通过 XML Web Service 我们可以均衡 Web 服务器的负载,创建分布式 Web 系统,提高 Web 系统页面访问的实时性;通过 Ajax 我们可以改变浏览器与服务器的交互模式,使得 Web 系统交互无需等待、无需刷新,从而提高 WEB 系统页面交互的实时性。XML Web

Service 和 Ajax 的结合可以综合应用这两种技术的优点,有效地创建实时分布式 Web 系统。

参考文献

- 1 Yoon - seop Chang, Hyeong - dong Park. XML Web Service - based development model for Internet GIS applications[J]. International Journal of Geographical Information Science, 2006(4): 371 ~ 399.
- 2 胡学骏、曾凡智,一种基于 XML Web Service 的分布式解决方案[J],计算机工程,2005(13): 204, 205, 222.
- 3 Alexei White, Ajax accelerates Web applications[J], Network World, 2006(2): 33 ~ 34.
- 4 Andrew Turner, Adding Ajax to a Website [J], MacTech Magazine, 2006(1): 48 ~ 50, 52, 54, 56, 58.
- 5 梁民、汪伟,基于 AJAX 技术开发 Web 应用[J],电脑知识与技术,2006(5): 119, 193.