

(12) 发明专利

(10) 授权公告号 CN 101669114 B

(45) 授权公告日 2012. 09. 05

(21) 申请号 200880013770. 2

(51) Int. Cl.

(22) 申请日 2008. 03. 26

G06F 17/00(2006. 01)

(30) 优先权数据

11/741, 095 2007. 04. 27 US

(56) 对比文件

CN 1851664 A, 2006. 10. 25, 全文.

(85) PCT申请进入国家阶段日

2009. 10. 27

US 2005/0240662 A1, 2005. 10. 27, 说明书第 2-24 段以及附图 1-7.

(86) PCT申请的申请数据

PCT/US2008/058212 2008. 03. 26

CN 1500328 A, 2004. 05. 26, 说明书第 4-12 页以及附图 1-5.

(87) PCT申请的公布数据

W02008/134162 EN 2008. 11. 06

审查员 孟祥岳

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 E·J·利普顿 B·C·勒鲁瓦

M·E·吉布斯 D·V·里德

S·卡尔维特 N·科塔里

M·J·哈德

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 陈斌 钱静芳

权利要求书 3 页 说明书 9 页 附图 2 页

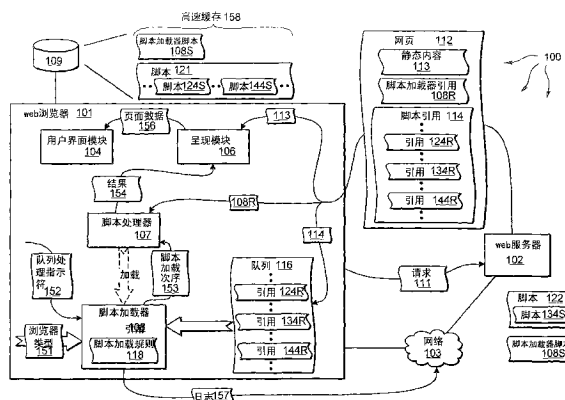
(54) 发明名称

动态加载脚本

(57) 摘要

本发明涉及用于动态地加载脚本的方法、系统和计算机程序产品。网页可以包括对控制 web 浏览器处的其它脚本的加载的脚本加载器的引用。脚本加载器可以根据覆盖 web 浏览器的默认脚本加载行为的脚本加载规则来加载脚本。脚本加载规则可以使得脚本以与在网页中遇到的次序不同的次序来加载、并行加载、以限制跨域请求的方式来加载等。脚本加载规则是可配置的,使得开发者可以创建定制的脚本加载行为。

CN 101669114 B



1. 一种在连接到网络的计算机系统处的用于在 web 浏览器处动态地加载脚本的方法, 所述计算机系统包括被配置成在所述计算机系统处呈现网页的所述 web 浏览器, 所述方法包括:

从 web 服务器接收网页的动作, 所述网页响应于来自所述 web 浏览器的对所述网页的请求被发送到所述 web 浏览器, 所述网页包括对脚本加载器的引用, 所述脚本加载器被配置成在执行时创建用于将其它脚本引用排队的数据结构, 所述脚本加载器还被配置成在接收到来自所述 web 浏览器的指示时根据覆盖所述 web 浏览器的默认脚本加载行为的脚本加载规则来动态地加载所述网页中所引用的脚本, 其中该脚本加载规则包括脚本加载规则策略, 该脚本加载规则策略指示如何基于所引用的脚本的特征和 / 或所引用的脚本之间的关系来加载所引用的脚本;

执行所述脚本加载器来创建用于将脚本引用排队的数据结构的动作;

处理所述网页以在所述数据结构中将包括在所述网页中的任何脚本引用排队的动作;

检测在所述数据结构中将包括在所述网页中的任何脚本引用排队之后没有其它脚本引用被包括在所述网页中的动作;

向所述脚本加载器指示没有其它脚本引用被包括在所述网页中以使得所述脚本加载器根据覆盖所述 web 浏览器的默认脚本加载行为的脚本加载规则来动态地加载对应于所排队的脚本引用的脚本的动作; 以及

将任何相关脚本结果包括在所述网页的呈现中的动作。

2. 如权利要求 1 所述的方法, 其特征在于, 还包括:

在从所述 web 服务器接收所述网页之前向所述 web 服务器发送 HTTP GET 的动作。

3. 如权利要求 1 所述的方法, 其特征在于, 还包括:

向所述 web 服务器发送异步 HTTP POST 的动作。

4. 如权利要求 1 所述的方法, 其特征在于, 从 web 服务器接收网页的动作包括接收在所述网页的顶部处包括脚本加载器引用的所述网页。

5. 如权利要求 1 所述的方法, 其特征在于, 向所述脚本加载器指示没有其它脚本引用被包括在所述网页中以使得所述脚本加载器动态地加载脚本的动作包括向所述脚本加载器指示没有其它脚本引用被包括在所述网页中以使得所述脚本加载器以与在所述网页中遇到脚本的次序不同的次序来加载脚本的动作。

6. 如权利要求 1 所述的方法, 其特征在于, 向所述脚本加载器指示没有其它脚本引用被包括在所述网页中以使得所述脚本加载器动态地加载脚本的动作包括向所述脚本加载器指示没有其它脚本引用被包括在所述网页中以使得所述脚本加载器并行地加载多个脚本的动作。

7. 如权利要求 1 所述的方法, 其特征在于, 向所述脚本加载器指示没有其它脚本引用被包括在所述网页中以使得所述脚本加载器动态地加载脚本的动作包括向所述脚本加载器指示没有其它脚本引用被包括在所述网页中以使得所述脚本加载器以限制对任何给定域的请求的数量的方式来加载脚本的动作。

8. 如权利要求 1 所述的方法, 其特征在于, 向所述脚本加载器指示没有其它脚本引用被包括在所述网页中以使得所述脚本加载器动态地加载脚本的动作包括向所述脚本加载

器指示没有其它脚本引用被包括在所述网页中以使得所述脚本加载器以覆盖所述 web 浏览器的默认加载行为的方式来加载脚本的动作。

9. 一种在连接到网络的计算机系统处的方法,所述计算机系统包括被配置成在所述计算机系统处呈现网页的 web 浏览器和被配置成动态地加载网页中所引用的脚本以覆盖所述 web 浏览器的默认脚本加载行为的脚本加载器,所述方法包括:

创建被配置成存储脚本引用的队列的动作;

用包括在指定网页中的脚本引用来填充所述队列的动作;

从所述 web 浏览器接收没有其它脚本引用被包括在所述指定网页中的指示的动作;以及

使得由所排队的脚本引用所引用的脚本根据覆盖所述 web 浏览器的默认脚本加载行为的脚本加载规则来加载的动作,其中该脚本加载规则包括脚本加载规则策略,该脚本加载规则策略指示如何基于所引用的脚本的特征和 / 或所引用的脚本之间的关系来加载所引用的脚本。

10. 如权利要求 9 所述的方法,其特征在于,还包括:

在使得所述脚本被加载之前枚举所述队列中的脚本引用以建立脚本引用列表的动作。

11. 如权利要求 10 所述的方法,其特征在于,还包括:

标识对应于包含在所述脚本引用列表中的脚本引用的脚本的动作;以及
确定所标识的脚本中的任一个是否彼此相关的动作。

12. 如权利要求 10 所述的方法,其特征在于,还包括:

标识至少一个脚本依赖于至少一个其它脚本的动作。

13. 如权利要求 10 所述的方法,其特征在于,还包括:

标识包括跨域请求的脚本的动作。

14. 如权利要求 9 所述的方法,其特征在于,所述使得脚本根据脚本加载规则来加载的动作包括以与在所述网页中遇到脚本的次序不同的次序来加载脚本的动作。

15. 如权利要求 9 所述的方法,其特征在于,所述使得脚本根据脚本加载规则来加载的动作包括并行地加载多个脚本的动作。

16. 如权利要求 9 所述的方法,其特征在于,所述使得脚本根据脚本加载规则来加载的动作包括以限制对任何给定域的并发请求的数量的方式来加载脚本的动作。

17. 如权利要求 9 所述的方法,其特征在于,所述使得脚本根据脚本加载规则来加载的动作包括以覆盖所述 web 浏览器的顺序加载行为的方式来加载脚本的动作。

18. 一种在连接到网络的计算机系统处的用于在 web 浏览器处动态地加载脚本的系统,所述计算机系统包括被配置成在所述计算机系统处呈现网页的所述 web 浏览器,所述系统包括:

用于从 web 服务器接收网页的装置,所述网页响应于来自所述 web 浏览器的对所述网页的请求被发送到所述 web 浏览器,所述网页包括对脚本加载器的引用,所述脚本加载器被配置成在执行时创建用于将其它脚本引用排队的数据结构,所述脚本加载器还被配置成在接收到来自所述 web 浏览器的指示时根据覆盖所述 web 浏览器的默认脚本加载行为的脚本加载规则来动态地加载所述网页中所引用的脚本,其中该脚本加载规则包括脚本加载规则策略,该脚本加载规则策略指示如何基于所引用的脚本的特征和 / 或所引用的脚本之间

的关系来加载所引用的脚本；

用于执行所述脚本加载器来创建用于将脚本引用排队的数据结构的装置；

用于处理所述网页以在所述数据结构中将包括在所述网页中的任何脚本引用排队的装置；

用于检测在所述数据结构中将包括在所述网页中的任何脚本引用排队之后没有其它脚本引用被包括在所述网页中的装置；

用于向所述脚本加载器指示没有其它脚本引用被包括在所述网页中以使得所述脚本加载器根据覆盖所述 web 浏览器的默认脚本加载行为的脚本加载规则来动态地加载对应于所排队的脚本引用的脚本的装置；以及

用于将任何相关脚本结果包括在所述网页的呈现中的装置。

19. 一种在连接到网络的计算机系统处的系统,所述计算机系统包括被配置成在所述计算机系统处呈现网页的 web 浏览器和被配置成动态地加载网页中所引用的脚本以覆盖所述 web 浏览器的默认脚本加载行为的脚本加载器,所述系统包括：

用于创建被配置成存储脚本引用的队列的装置；

用于用包括在指定网页中的脚本引用来填充所述队列的装置；

用于从所述 web 浏览器接收没有其它脚本引用被包括在所述指定网页中的指示的装置；以及

用于使得由所排队的脚本引用所引用的脚本根据覆盖所述 web 浏览器的默认脚本加载行为的脚本加载规则来加载的装置,其中该脚本加载规则包括脚本加载规则策略,该脚本加载规则策略指示如何基于所引用的脚本的特征和 / 或所引用的脚本之间的关系来加载所引用的脚本。

动态加载脚本

[0001] 背景

[0002] 1. 背景和相关技术

[0003] 计算机系统和相关技术影响社会的许多方面。的确,计算机系统处理信息的能力已改变人们生活和工作的方式。计算机系统现在通常执行在计算机系统出现以前手动执行的许多任务(例如,文字处理、日程安排和会计等)。最近,计算机系统彼此耦合并耦合到其它电子设备以形成计算机系统和其它电子设备可以在其上传输电子数据的有线和无线计算机网络。因此,许多计算任务的执行分布在多个不同的计算机系统和/或多个不同的计算组件上。

[0004] 基于网络的通信的一种常见形式是在万维网(“WWW”)上交换电子消息。万维网上的内容通常以客户机/服务器模型来访问。客户机计算机系统的“web 浏览器”发送访问由服务器计算机系统的“web 服务器”提供的内容的请求(例如,通过将统一资源定位符(“URL”)输入到该 web 浏览器中)。如果 web 浏览器的用户被授权访问该内容,则 web 服务器通常将该内容提供给该 web 浏览器。在 web 环境中,内容和对内容的请求经常使用超文本传输协议(“HTTP”)来传输。基于 web 的内容可以在超文本标记语言(“HTML”)页面、样式表、图像、脚本等中提供。

[0005] 例如,可以使用脚本来执行比仅使用 HTML 伪指令所允许的操作更复杂的操作。一般而言,脚本是可以在 web 服务器执行以将内容添加到页面或可以被发送到 web 浏览器以供在 web 浏览器处执行来将内容添加到网页的可执行代码。脚本可以用诸如例如 JavaScript、VBScript、ASP、PHP、Perl、或 ASP.Net 等脚本(编程)语言中来开发。可以维护页面的 web 服务器可以包括服务器侧和客户机侧脚本两者。服务器侧脚本可用于获得可由 web 服务器访问以包括在网页中的数据。客户机侧脚本可用于获取并呈现对客户机器唯一的数据,诸如例如本地时间和日期。然而,诸如例如向该 web 服务器或其它 web 服务器请求数据或额外脚本的更复杂的客户机侧脚本也是可能的。

[0006] 在 web 服务器接收对包括服务器侧脚本的网页的 web 浏览器请求时,该 web 服务器将该服务器侧脚本传送到合适的脚本引擎。该脚本引擎处理该脚本并返回代表该服务器侧脚本功能的对应 HTML 伪指令。该对应的 HTML 伪指令随后被注入到网页中以便返回给 web 浏览器(与任何客户机侧脚本一起)。

[0007] 例如,如果服务器侧脚本被配置成查询并返回 10 个数据库行,则该脚本引擎以 HTML 格式将这 10 个数据库行返回给 web 服务器。该 web 服务器随后将表示这 10 个数据库行的 HTML 注入到网页中的任何其它 HTML 伪指令和客户机侧脚本中。在所有服务器侧脚本被处理之后,因处理服务器侧脚本所得的 HTML、其它 HTML、以及任何客户机侧脚本被发回 web 浏览器。

[0008] 客户机侧脚本可被嵌入在网页中或包括在单独的文件中。在客户机侧脚本被包括在外部文件中时,网页可以包括引用该脚本的脚本引用(例如, <script type = " text/javascript" src = " hello.js" ></script>)。客户机侧脚本和脚本引用可以被成行地包括在发送到 web 浏览器的网页中。因此,随着 web 浏览器处理该网页,它可以遇到嵌入的

客户机侧脚本以及对客户机侧脚本的脚本引用。

[0009] 在遇到嵌入的脚本时, web 浏览器可以找出合适的脚本引擎来处理该脚本。在遇到客户机侧脚本引用时, web 浏览器可以将该所引用的脚本加载(例如,在该脚本的副本已经被高速缓存时)到脚本引擎以供处理或者它可以从 web 服务器请求该脚本。该 web 服务器可以随后将该脚本返回给 web 浏览器且该 web 浏览器可以将该脚本加载(且可能也高速缓存)到脚本引擎以供处理。

[0010] web 浏览器通常假设之后在 web 中遇到的任何脚本取决于之前在网页中所遇到的脚本。因此, web 浏览器通常以在网页中遇到脚本的次序依次加载脚本。然而在许多网页中,后续遇到的脚本不依赖任何之前遇到的脚本。因此,不幸且尤其对具有大量客户机侧脚本的网页不幸的是,依次加载是相对低效的。

[0011] 此外,在开发网页时,开发者可以选择包括之前开发的客户机侧脚本以利用客户机侧功能。在某种程度上开发者可能假设之前的客户机侧脚本如预期般工作。或者,开发者可能缺乏彻底地测试任何之前开发的客户机侧脚本的时间、愿望、或专业知识。然而,一个或多个客户机侧脚本可能不能在网页被部署时加载或运行。不幸的是,典型 web 浏览器(以及其中的脚本引擎)向网页开发者提供回极少(如果有的话)例如指示客户机侧脚本失效的环境的有用上下文。因此,网页开发者可能被要求基于涉及客户机侧脚本失效的信息的不完整记录来调试该网页。

[0012] 简要概述

[0013] 本发明涉及用于动态地加载脚本的方法、系统和计算机程序产品。web 浏览器响应于来自该 web 浏览器的对 web 网页的请求从 web 服务器接收该网页。该网页包括对脚本加载器的引用。该脚本加载器被配置成在执行时创建用于将其它脚本引用排队的数据结构。该脚本加载器还被配置成在接收来自 web 浏览器的指示时根据覆盖该 web 浏览器的默认脚本加载行为的脚本加载规则来动态地加载网页中所引用的脚本。

[0014] web 浏览器执行脚本加载器。脚本加载器创建用于将脚本引用排队的数据结构。web 浏览器处理网页以在该数据结构中将包括在网页中的任何脚本引用排队。脚本加载器用包括网页中的脚本引用来填充队列。web 浏览器检测在将包括在网页中的任何脚本引用排队之后没有其它脚本引用被包括在该网页中。

[0015] web 浏览器向脚本加载器指示没有其它脚本引用被包括在网页中。脚本加载器从 web 浏览器接收没有其它脚本引用被包括在指定网页中的指示。脚本加载器使得由排队的脚本引用所引用的脚本根据覆盖 web 浏览器的默认脚本加载行为的脚本加载规则来加载。web 浏览器将任何相关脚本结果包括在网页的呈现中。

[0016] 提供本概述是为了以简化的形式介绍将在以下详细描述中进一步描述的一些概念。该概述不旨在标识所要求保护的的主题的关键特征或必要特征,也不旨在用于帮助确定所要求保护的的主题的范围。

[0017] 本发明的附加特征和优点将在以下描述中叙述,且其部分根据本描述将是显而易见的,或可通过对本发明的实践获知。本发明的特征和优点可通过在所附权利要求书中特别指出的手段和组合来实现和获得。本发明的这些和其他特征将通过以下描述和所附权利要求书而变得更加完全明显,或可通过对下文中所述的本发明的实践来获知。

[0018] 附图简述

[0019] 为了描述可获得本发明的上述和其它优点和特征的方法,将通过引用附图中示出的本发明的具体实施例来呈现以上简要描述的本发明的更具体描述。可以理解,这些附图仅描述本发明的典型实施例,从而不被认为是对其范围的限制,本发明将通过使用附图用附加特征和细节来描述和说明,在附图中:

[0020] 图 1 示出了便于动态地加载脚本的示例计算机体系结构。

[0021] 图 2 示出了用于动态地加载脚本的示例方法的流程图。

[0022] 详细描述

[0023] 本发明涉及用于动态地加载脚本的方法、系统和计算机程序产品。web 浏览器响应于来自该 web 浏览器的对网页的请求从 web 服务器接收该网页。该网页包括对脚本加载器的引用。该脚本加载器被配置成在执行时创建用于将其它脚本引用排队的数据结构。该脚本加载器还被配置成在接收来自 web 浏览器的指示时根据覆盖该 web 浏览器的默认脚本加载行为的脚本加载规则来动态地加载网页中所引用的脚本。

[0024] web 浏览器执行脚本加载器。脚本加载器创建用于将脚本引用排队的数据结构。web 浏览器处理 web 网页以在该数据结构中将包括在网页中的任何脚本引用排队。脚本加载器用包括在网页中的脚本引用来填充队列。web 浏览器检测在将包括在网页中的任何脚本引用排队之后没有其它脚本引用被包括在该网页中。

[0025] web 浏览器向脚本加载器指示没有其它脚本引用被包括在该网页中。脚本加载器从 web 浏览器接收没有其它脚本引用被包括在指定网页中的指示。脚本加载器使得由排队的脚本引用所引用的脚本根据覆盖 web 浏览器的默认脚本加载行为的脚本加载规则来加载。web 浏览器将任何相关脚本结果包括在网页的呈现中。

[0026] 本发明的各实施例可以包括含有计算机硬件的专用或通用计算机,这将在以下做出进一步讨论。本发明的范围内的各个实施例还包括用于承载或其上储存有计算机可执行指令或数据结构的可读介质。这样的可读介质可以是可由通用或专用计算机访问的任何可用介质。作为示例而非限制,可读介质可包括物理(或可记录类型的)计算机可读存储介质,诸如 RAM、ROM、EEPROM、CD-ROM 或其它光盘存储、磁盘存储或其它磁存储设备、或可用于存储计算机可执行指令或数据结构形式的所需程序代码装置且可由通用或专用计算机访问的任何其它介质。

[0027] 在本说明书和所附权利要求书中,“网络”被定义为允许在计算机系统和 / 或模块之间传输电子数据的一个或多个数据链路。当信息通过网络或另一通信连接(硬连线、无线、或者硬连线或无线的组合)传输或提供给计算机时,该计算机将该连接适当地视为计算机可读介质。因此,作为示例而非限制,可读介质还可包括可用于承载或存储计算机可执行指令或数据结构形式的所需程序代码装置并可由通用或专用计算机访问的网络或数据链路。

[0028] 计算机可执行指令包括例如,使通用计算机、专用计算机、或专用处理设备执行某一功能或某组功能的指令和数据。计算机可执行指令可以是例如二进制代码、诸如汇编语言等中间格式指令、或甚至源代码。尽管用结构特征和 / 或方法动作专用的语言描述了本主题,但可以理解,所附权利要求书中定义的主题不必限于上述特征或动作。相反,上述特征和动作是作为实现权利要求的示例形式而公开的。

[0029] 本领域的技术人员将理解,本发明可以在具有许多类型的计算机系统配置的网络

计算环境中实践,这些计算机系统配置包括个人计算机、台式计算机、膝上型计算机、消息处理器、手持式设备、多处理器系统、基于微处理器的或可编程消费电子设备、网络 PC、小型计算机、大型计算机、移动电话、PDA、寻呼机等等。本发明也可以在其中通过网络链接(或者通过硬连线数据链路、无线数据链路,或者通过硬连线和无线数据链路的组合)的本地和远程计算机系统两者都执行任务的分布式系统环境中实践。在分布式系统环境中,程序模块可以位于本地和远程存储器存储设备中。

[0030] 图 1 示出了便于动态地加载脚本的示例计算机体系结构 100。如图所示,计算机体系结构 100 包括 web 浏览器 101、web 服务器 102、以及网络 103。web 浏览器 101 和 web 服务器 102 可以被连接到诸如例如局域网(“LAN”)、广域网(“WAN”)或甚至因特网等网络 103。因此,web 浏览器 101 和 web 服务器 102 的组件可以从连接到网络 103 的其它组件接收数据并向这些其它组件发送数据。因此,这些组件可以创建消息相关数据并通过网络交换消息相关数据(例如,网际协议(“IP”)数据报和利用 IP 数据报的其它更高层协议,诸如传输控制协议(“TCP”)、超文本传输协议(“HTTP”)、简单邮件传输协议(“SMTP”)等)。例如,各组件可交换 HTTP 请求与响应以及包含 web 服务相关数据的简单对象访问协议(“SOAP”)封装。

[0031] 一般而言,web 浏览器 101 被配置成请求(例如,经由到指定 URL 的 HTTP 消息)并呈现来自诸如例如 web 服务器 104 等可经由网络 103 访问的 web 服务器的基于 web 的内容。基于 web 的内容可以包括文本数据、图像数据、音频/视频数据、可执行代码(例如,脚本)等。

[0032] 一般而言,web 服务器 102 被配置成接收对基于 web 的内容的请求(例如,到指定 URL 的 HTTP 消息)并将所请求的基于 web 的内容返回给请求的 web 浏览器。在 web 服务器 102 接收请求时,它可以标识对应于该请求的服务器侧页面(例如,基于包含在该请求中的 URL 将该请求分派到合适的进程)。web 服务器 106 可以随后加载所标识的服务器侧页面。

[0033] 在服务器侧页面包括服务器侧脚本引用时,web 服务器 106 可以向服务器侧脚本处理器发送所引用的服务器侧脚本(例如,ASP. Net 指令)以供处理。该脚本处理器处理服务器侧脚本并将任何对应结果返回给 web 服务器 102。web 服务器 102 可以随后将这些结果(可能在附加的格式化之后)与其它内容和任何客户机侧脚本引用一起包括在网页中。web 服务器 102 可以随后将该网页返回给请求的 web 浏览器。

[0034] 因此,例如,web 浏览器 101 可以向 web 服务器 101 发送请求 111(HTTPGET)(通过利用 web 服务器 102 的合适的 URL)。web 服务器 102 可以处理请求 111 并生成/获得对应的内容(静态的或通过服务器侧脚本动态地访问)。对应的内容和任何客户机侧脚本引用可以在网页 112 中返回给 web 浏览器 101(例如,包含在 HTTP 消息中)。客户机侧脚本引用可以引用存储在 web 服务器 102 处的诸如例如大容量存储设备或组件的目录结构(例如,动态链接库(“DLL”))等各种不同位置中(例如,脚本 122 中)的脚本,或可以引用在其它 web 服务器处的脚本。web 服务器 102 可以随后响应于请求 111 将网页 112 返回给 web 浏览器 101。

[0035] 在 web 浏览器 101 接收网页时,它可以将该网页的各部分分配给被配置成处理该网页的各部分的模块。因此,在 web 浏览器 101 接收网页 112 时,诸如例如静态内容 113 等静态内容可以被分配给呈现模块 106。任何脚本(例如,JavaScript、VB 脚本等)或诸如例

如脚本引用 108R 等脚本引用可以被分配给诸如例如脚本处理器 107 等合适的脚本处理器。之前在 web 浏览器 101 处执行的脚本可被存储在诸如例如高速缓存 158 等大容量存储 109 处。因此,响应于接收客户机侧脚本引用,web 浏览器 101 可以经由网络 103 从 web 服务器(可以是或可以不是发送该客户机侧脚本引用的 web 服务器)或从高速缓存 158 获得对应的客户机侧脚本。

[0036] 脚本处理器 107 可以执行脚本并将任何所得内容(如果有的话)返回给呈现模块 106。例如,脚本处理器 107 可以将结果 154 返回给呈现模块 106。呈现模块 106 可以在用户界面 104 处呈现数据。例如,呈现模块 106 可以在用户界面 104 处呈现页面数据 156。呈现模块 106 可以随着其接收到内容来呈现内容,使得网页的某些部分(例如,静态内容)在网页的其它部分(例如,脚本结果)之前被呈现。

[0037] 图 2 示出了用于动态地加载脚本的示例方法 200 的流程图。方法 200 将参考计算机体系结构 100 中的组件和数据来描述。

[0038] 方法 200 包括从 web 服务器接收网页的动作,该网页是响应于来自 web 浏览器的对网页的请求发送到 web 浏览器的(动作 200)。该网页包括对被配置成在执行时创建用于将其它脚本引用排队的数据结构的脚本加载器的引用。该脚本加载器还被配置成在接收来自 web 浏览器的指示时根据覆盖该 web 浏览器的默认脚本加载行为的脚本加载规则来动态地加载网页中所引用的脚本。

[0039] 例如,web 浏览器 101 可以响应于请求 111 来接收网页 112。网页 112 包括指向脚本加载器脚本 108S 的脚本加载器引用 108R。脚本加载器脚本 108S 被配置成在执行时创建用于将网页 112 中的诸如例如 124R、134R 和 144R 等其它脚本引用排队的数据结构。脚本加载器脚本 108S 还被配置成在接收来自 web 浏览器 101 的指示时根据脚本加载规则 118 动态地加载网页 112 中所引用的脚本。脚本加载规则 118 可以覆盖 web 浏览器 101 的默认脚本加载行为。

[0040] 脚本加载规则 118 可以包括用于在 web 浏览器 101 处加载脚本的逻辑和策略。因此,脚本加载器 108 可以将脚本加载规则 118 应用于队列 116 中的客户机侧脚本引用以创建脚本加载次序 153。脚本加载规则逻辑可以包括用于枚举包括在队列 116 中的客户机侧脚本引用的算法。例如,脚本加载规则 118 可以枚举引用 124R、134R、144R 等。

[0041] 从所枚举的客户机侧脚本引用中,其它脚本加载规则逻辑可以包括标识对应的所引用的脚本对彼此和 / 或对其它外部资源的依赖性的算法。例如,脚本加载器 108 可以执行可以确定脚本 134S 依赖于脚本 124S 且脚本 144S 不依赖于任何其它脚本的脚本加载规则。从所枚举的客户机侧脚本引用中,其它脚本加载规则逻辑可以包括标识包括跨域请求的脚本的算法。例如,脚本加载器 108 可以执行脚本加载规则以确定在执行所引用的脚本时要发生多少对除了 web 服务器 102 以外的服务器的请求。

[0042] 其它脚本加载规则逻辑可以包括用于标识所引用的脚本的其它特征和 / 或所引用的脚本之间的关系关系的算法。脚本加载规则逻辑还是可扩展的,使得开发者可以开发他们自己的脚本加载规则逻辑以标识 web 网页中的所引用的脚本的特征和 / 或所引用的脚本之间的关系。或者,开发者可以修改现有脚本加载规则逻辑以定制用于指定目的的脚本加载规则逻辑。

[0043] 脚本加载规则策略指示如何基于所枚举的脚本的特征和 / 或所枚举的脚本之间

的关系来加载所枚举的所引用的脚本。脚本加载规则策略可被应用于从队列 116 中所枚举的脚本引用以采用指定次序来加载对应脚本。脚本加载次序可以基于所标识的脚本依赖性以及其它脚本关系和特征来指示。

[0044] 在某些实施例中,应用脚本加载规则策略以便按不同于所引用的脚本在网页中出现的次序的指定次序来加载所枚举的所引用的脚本(无序加载)。例如,脚本引用 124R、134R 和 144R 可以按此次序被包括在网页 112 中。即,引用 124R 距网页 112 的顶部更近而引用 144R 距网页 112 的底部更近。然而,脚本加载规则策略可以指示即使引用 124R 在引用 134R 之前被接收到,脚本 134S 仍然将在脚本 124S 和 144S 之前被加载。此外,脚本加载规则策略可以指示即使引用 124R 在引用 144R 之前被接收到,脚本 144S 仍然将在脚本 124S 之前被加载。

[0045] 在其它实施例中,应用脚本加载规则策略以并行地加载所枚举的所引用的脚本(并行加载)。例如,脚本加载规则策略可以指示因为脚本 124S 不依赖于脚本 144S 且脚本 144S 不依赖于脚本 124S,所以脚本 124S 和 144S 将被并行地加载。

[0046] 在其它实施例中,应用脚本加载规则策略以便按限制对指定域的并发跨域请求的数量的方式来加载所枚举的脚本。例如,脚本加载规则策略可以指示脚本将以将对除了 web 服务器 102 以外的 web 服务器的并发跨域请求的数量限制为指定阈值的次序被加载。

[0047] 脚本加载规则策略还可以是可扩展的,使得开发者可以开发他们自己的脚本加载规则策略来以指定方式加载脚本。或者,开发者可以修改现有脚本加载规则策略以定制用于指定目的的脚本加载规则策略。例如,脚本加载规则策略可以指示将对给定已知依赖性集并行地加载各脚本,而不是顺序地加载脚本。

[0048] 脚本加载器 108 还可以在公式化脚本加载次序时考虑单独的浏览器特征以在浏览器环境内适当地加载脚本。例如,脚本加载器 108 可以在公式化脚本加载次序 153 时考虑浏览器类型 151(且可以意识到浏览器类型 151 的脚本加载特征)。

[0049] 在某些实施例中,可以将脚本加载规则 118 硬编码到脚本加载器 108 中。因此,为了在这些实施例中定制脚本加载规则 118,开发者可以修改脚本加载器 108 的源代码以定制用于指定目的的脚本加载规则 118(并因此还定制脚本加载 108)。在其它实施例中,脚本加载器 108 外部地访问脚本加载规则 118。因此,为了在这些其它实施例中定制脚本加载规则 118,开发者可以修改该外部可访问脚本加载规则 118 以定制用于指定目的的脚本加载规则 118。

[0050] 脚本加载器 108 可以提供用于与浏览器 101 和脚本处理器 107 接口的 API。

[0051] 方法 200 包括执行该脚本加载器以创建用于将脚本引用排队的数据结构的动作(动作 202)。例如,脚本处理器 107 可以加载脚本加载器 108 以创建队列 116。可以将脚本加载器引用 108R 内联地包括于邻近(或在)网页 112 的顶部处。因此,脚本加载器引用 108R 是所遇到的第一个引用的可能性增加。由于 web 浏览器 101 的默认行为(例如,顺序地加载脚本),脚本处理器 107 可以在处理任何其它脚本或脚本引用之前处理脚本加载器引用 108R。因此,处理脚本引用 108S 以加载脚本加载器 108 本质上引导了覆盖 web 浏览器 101 的默认脚本加载行为的脚本加载行为。

[0052] 方法 200 包括创建被配置成存储脚本引用的队列的动作(动作 203)。例如,脚本加载器 108 可以创建队列 116 以在网页 112 中存储客户机侧脚本引用。方法 200 包括处理

网页以在数据结构中将包括在网页中的任何脚本引用排队的动作（动作 204）以及用包括在指定网页中的脚本引用来填充队列的动作（动作 205）。例如，web 浏览器 101 和脚本加载器 108 可以互操作以处理网页 112 并将队列引用 124R、134R 和 144R 存储在队列 116 中。

[0053] 方法 200 包括检测在数据结构中将包括在网页中的任何脚本引用排队之后没有其它脚本引用被包括在网页中的动作（动作 206）。例如，web 浏览器 101 可以检测在引用 124R、134R 和 144R 被排入队列 116 之后没有其它脚本引用被包括在网页 112 中。

[0054] 方法 200 包括向脚本加载器指示没有其它脚本引用被包括在网页中的动作（动作 207）。指示没有其它引用被包括使得脚本加载器根据覆盖 web 浏览器的默认脚本加载行为的脚本加载规则来动态地加载对应于排队的脚本引用的脚本。例如，在来自网页 112 的所有引用已经被排入队列 116 时，web 浏览器 101 可以向脚本加载器 107 发送队列处理指示符 152。队列处理指示符 152 可以使得脚本加载器 108 根据脚本加载规则 118 来动态地加载排入队列 116 的脚本。

[0055] 方法 200 包括从 web 浏览器接收没有其它脚本引用被包括在指定网页中的指示的动作（动作 208）。例如，脚本加载器 108 可以从 web 浏览器 101 接收队列处理指示符 152。方法 200 包括使得由排队的脚本引用所引用的脚本根据覆盖 web 浏览器的默认脚本加载行为的脚本加载规则来加载的动作（动作 209）。例如，脚本加载器 108 可以使得脚本 124S、134S 和 144S 以及网页 112 中的其它所引用的脚本根据覆盖 web 浏览器 101 的默认脚本加载行为的脚本加载规则 118 来加载。

[0056] 脚本加载次序 153 共同表示用于加载网页 112 中所引用的脚本的次序。在某些实施例中，脚本处理器和脚本加载器 108 进行通信以协调各脚本的实际加载。例如，脚本加载器 108 可以向脚本处理器 107 发送加载一个或多个脚本的命令。在加载一个或多个脚本时，脚本处理器 107 可以指示将一个或多个脚本加载到脚本加载器 108。脚本加载器 108 随后可以发送将另一个或多个脚本加载到脚本加载器 108 的另一命令。在加载该另一个或多个脚本时，脚本处理器 107 可以指示将该另一个或多个脚本加载到脚本加载器 108。类似的机制可以发生，直到网页 112 中的脚本引用被加载。

[0057] 因此，脚本加载器 108 和脚本处理器 107 可以互操作来以与在网页 112 中遇到的次序不同的次序来加载脚本、并行地加载多个脚本、以限制并发跨域请求的方式加载脚本等。脚本加载器 108 还可以使得脚本以不违反 web 浏览器 101 的脚本加载特征（基于浏览器类型 151）的方式被加载。

[0058] 某些已加载脚本可以生成内容以供包括在网页 112 中。结果 154 共同表示从 web 浏览器 101 处执行的客户机侧脚本返回的任何内容。脚本处理器 107 可以向呈现模块 106 发送结果 154。方法 200 包括将任何相关脚本结果包括在网页的呈现中的动作（动作 210）。例如，呈现模块 106 可以将结果 154 包括在页面数据 156（表示网页 112 的数据）中。呈现模块 106 可以在用户界面 104 处呈现页面数据 156。

[0059] 因此，本发明的各实施例方便了在与客户机侧脚本加载相关联的初始化（init）并加载（load）事件之前的时刻加载脚本。可以将脚本加载器引用返回给 web 浏览器以便于在不同情况下的此行为。

[0060] 在一个实施例中，在对 web 应用程序的初始请求是 HTTP GET 时，将脚本加载器引用返回给 web 浏览器。脚本加载器引用可以被内联地包括在返回的网页的顶部，诸如例如：

[0061] <script src = “ScriptLoader.js” />

[0062] 为 web 应用程序创建并注册脚本加载器, 诸如例如 :

[0063] Sys.Application.set_scriptLoader(new Sys.ScriptLoader());

[0064] //

[0065] this._loader.add_allScriptsLoaded(

[0066] this._allScriptsLoadedDelegate);

[0067] 将包括引用、内联脚本、以及处理程序的脚本引用排队到脚本加载器。如前所述, 脚本可以从多个不同位置加载。因此, 脚本可以从自定义库加载, 诸如例如 :

[0068] Sypplication.queueScriptReference(' /CustomScript.js');

[0069] Sys.Application.queueScriptReference((“Sys.Application.add_init(

[0070] \$create(Custom.UI.Control, {}, {}, {}, \$get(‘f’))

[0071]);

[0072] Sys.Application.initialize();

[0073] 随后可以指示该注册脚本加载器来加载它已经排队的脚本引用, 诸如例如 :

[0074] Sys.Application.get_scriptLoader().set_scriptLoadTimeOut(n)

[0075] Sys.Application.get_scriptLoader().loadScripts();

[0076] 该脚本加载器可以在所有脚本已经被加载时指示 web 应用程序。

[0077] 在另一实施例中, 在对 web 应用程序的异步 HTTP POST 中将脚本加载器引用返回给 web 浏览器。在此其它实施例中, 脚本加载可能已经被创建了且一请求管理器挂钩到所加载的脚本上的事件。因此可以发生注册, 诸如例如 :

[0078] Sys.Application.get_scriptLoader().

[0079] add_allScriptsLoaded(PRM._allScriptsLoadedDelegate);

[0080] 随后可以使用内部脚本加载器来将脚本引用排队。这包括脚本引用、内联脚本、以及任何处理程序。因此, 脚本可以从自定义库加载, 诸如例如 :

[0081] Sys.Application.queueScriptReference(' /CustomScript.js');

[0082] Sys.Application.queueScriptReference((“Sys.Application.add_init(

[0083] \$create(Custom.UI.Control, {}, {}, {}, \$get(‘f’))

[0084]);

[0085] 指示已注册的脚本加载器加载它的脚本引用, 诸如例如 :

[0086] Sys.Application.get_scriptLoader().loadScripts();

[0087] 在本实施例以及其它之前所述的实施例中, 页面或组件开发者能够通过脚本加载器的 API 和声明性句法来定义在 web 浏览器处加载脚本的行为。脚本管理器随后生成使用客户机侧脚本处理器的输出。

[0088] 其它脚本相关功能也可以被包括在脚本加载器中。例如, 脚本加载器可被配置成将任何脚本加载错误写入日志文件。脚本加载错误可以包括 : 超时、依赖性错误、资源错误等。关于脚本加载错误的相关上下文信息也可以在发生脚本加载错误发生时写入日志文件。日志文件可被存储在可由页面或组件开发者访问的位置。例如, 在检测到 web 浏览器 101 处的脚本加载错误时, 脚本加载器 108 可以将日志 157 存储到网络 103 内的网络位置。开发者可以从该网络位置访问日志 157 以协助测试和 / 或调试脚本加载器 108 以及其它脚

本。

[0089] 本发明可具体化为其它具体形式而不背离其精神或本质特征。所述实施例在所有方面都应被认为仅是说明性而非限制性的。从而,本发明的范围由所附权利要求书而非前述描述指示。落入权利要求书的等效方式的含义和范围内的所有改变应被权利要求书的范围涵盖。

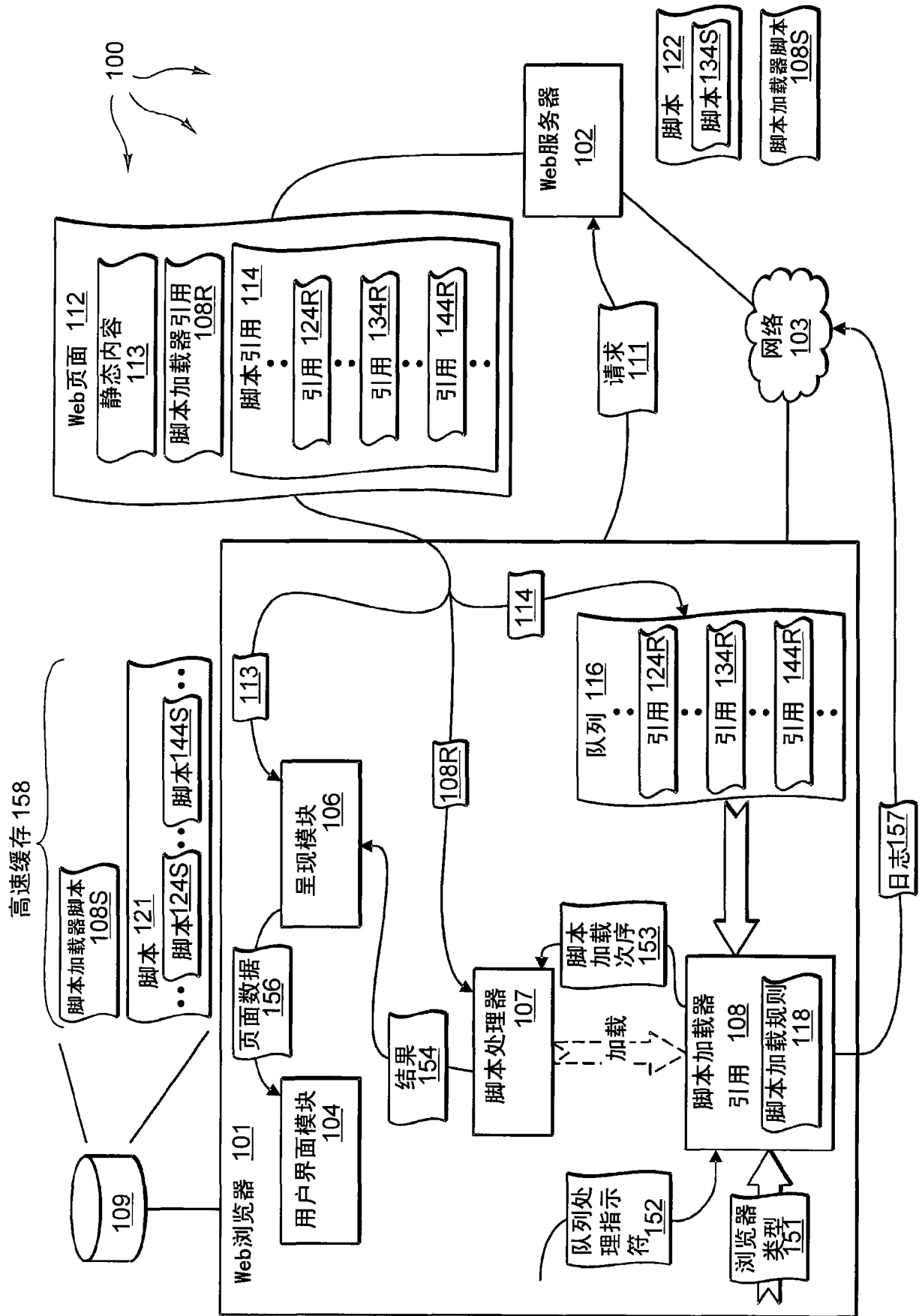


图 1

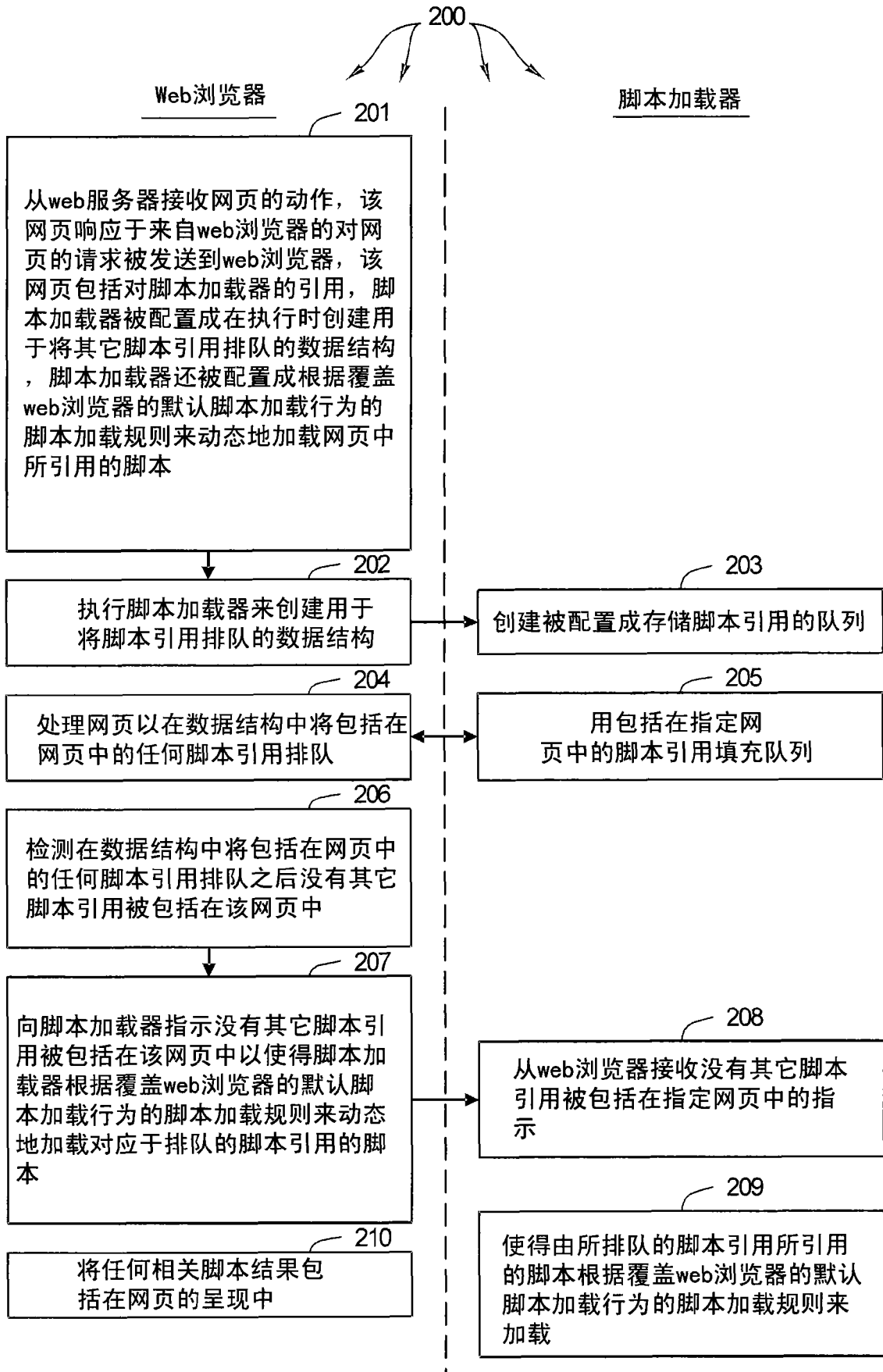


图 2