



(12) 发明专利

(10) 授权公告号 CN 106575298 B

(45) 授权公告日 2020. 10. 30

(21) 申请号 201580040337.8

V • 佩雷尔曼 T • 克拉伊舍克

(22) 申请日 2015.07.25

(74) 专利代理机构 上海专利商标事务所有限公司 31100

(65) 同一申请的已公布的文献号

代理人 唐杰敏

申请公布号 CN 106575298 A

(43) 申请公布日 2017.04.19

(51) Int.Cl.

(30) 优先权数据

G06F 16/957 (2019.01)

62/029,398 2014.07.25 US

14/809,046 2015.07.24 US

(56) 对比文件

(85) PCT国际申请进入国家阶段日

CN 101127044 A, 2008.02.20

CN 102365636 A, 2012.02.29

2017.01.22

CN 101464905 A, 2009.06.24

US 2010058172 A1, 2010.03.04

(86) PCT国际申请的申请数据

尤朝等. 一种在线的动态网页分块缓存方

PCT/US2015/042146 2015.07.25

法.《电子学报》.2009, (第5期), 第1087-1091页.

(87) PCT国际申请的公布数据

张紫薇. Web前端性能优化的研究与应用.

W02016/015031 EN 2016.01.28

《中国优秀硕士学位论文全文数据库》.2011, (第3期), 全文.

(73) 专利权人 高通股份有限公司

审查员 卜庆庆

地址 美国加利福尼亚州

(72) 发明人 B • 刘 D • 雷布尼科夫

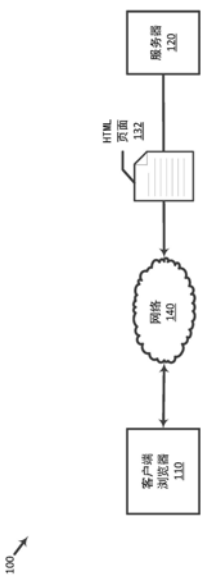
权利要求书3页 说明书13页 附图7页

(54) 发明名称

包含动态内容和陈旧内容的网站的快速呈现

(57) 摘要

一种用于呈现网页的示例方法包括基于处理由一位置参引的第一Web文档来生成第一经串行化的最终文档对象模型(DOM)。该第一经串行化的最终DOM包括第一内联的动态内容。该方法还包括基于处理由该位置参引的第二Web文档来生成第二经串行化的最终DOM。该第二经串行化的最终DOM包括第二内联的动态内容。该方法进一步包括基于第一经串行化的最终DOM与第二经串行化的最终DOM之间的差异来生成公共的经串行化的最终DOM。该公共的经串行化的最终DOM包括包含在第一经串行化的最终DOM和第二经串行化的最终DOM中的静止动态内容, 并且该差异是真实动态内容。该方法还包括将该公共的经串行化的最终DOM呈现给设备的显示器。



1. 一种用于呈现网页的方法,包括:

基于处理由一位置参引的第一Web文档来生成第一经串行化的最终文档对象模型DOM,所述第一Web文档关联于包括静止动态内容和第一真实动态内容的第一动态内容,所述第一动态内容是客户端侧生成的并且内联在所述第一经串行化的最终DOM中;

基于处理由所述位置参引的第二Web文档来生成第二经串行化的最终DOM,所述第二Web文档关联于包括所述静止动态内容和第二真实动态内容的第二动态内容,所述第二动态内容是客户端侧生成的并且内联在所述第二经串行化的最终DOM中;

基于所述第一经串行化的最终DOM与所述第二经串行化的最终DOM之间的差异来生成公共的经串行化的最终DOM,所述公共的经串行化的最终DOM包括包含在所述第一经串行化的最终DOM和所述第二经串行化的最终DOM中的静止动态内容,其中生成所述公共的经串行化的最终DOM包括:标识所述第一经串行化的最终DOM与所述第二经串行化的最终DOM之间的一个或多个差异,以及从所述第一经串行化的最终DOM中移除所述一个或多个差异,所述一个或多个差异是真实动态内容;以及

将所述公共的经串行化的最终DOM呈现给设备的显示器。

2. 如权利要求1所述的方法,进一步包括:

解析所述第一Web文档,所述第一Web文档是标记语言的形式并且包括一组元素和一组规则;

将所述一组规则应用于所述一组元素中的一个或多个元素;

根据所述解析和所述应用来生成最终DOM,所述最终DOM包括基于应用所述一组规则所生成的所述第一动态内容;

将所述最终DOM进行串行化以产生所述第一经串行化的最终DOM。

3. 如权利要求2所述的方法,其中,生成所述第一经串行化的最终DOM包括:将所述第一经串行化的最终DOM中的一个或多个非内联规则移至内联规则。

4. 如权利要求3所述的方法,其中,规则包括客户端侧脚本。

5. 如权利要求3所述的方法,其中,规则是应用于所述第一Web文档中的一个或多个元素的格式化规则。

6. 如权利要求2所述的方法,其中,所述第一经串行化的最终DOM和所述第二经串行化的最终DOM是所述最终DOM的不同版本。

7. 如权利要求1所述的方法,进一步包括:

将所述第一经串行化的最终DOM串行化为文件;以及

将所述文件高速缓存。

8. 如权利要求1所述的方法,其中,呈现所述公共的经串行化的最终DOM包括在前台中呈现所述公共的经串行化的最终DOM,所述方法进一步包括:

从高速缓存中检索所述公共的经串行化的最终DOM;

接收对于由所述位置参引的第三Web文档的当前请求,所述第三Web文档关联于包括所述静止动态内容和第三真实动态内容的第三动态内容,所述第三动态内容是客户端侧生成的;

在后台进程中,构成用于所述当前请求的当前DOM并且呈现所述当前DOM;

在所述当前DOM已经被呈现之后,用所呈现的公共的经串行化的最终DOM来替代所呈现

的当前DOM。

9. 如权利要求1所述的方法,进一步包括:

基于处理由所述位置参引的第三Web文档来生成第三经串行化的最终DOM,所述第三Web文档关联于包括所述静止动态内容和第三真实动态内容的第三动态内容,所述第三动态内容是客户端侧生成的并且内联在所述第三经串行化的最终DOM中;

基于所述第二经串行化的最终DOM与所述第三经串行化的最终DOM之间的第二差异来生成第二公共的经串行化的最终DOM,所述第二公共的经串行化的最终DOM包括包含在所述第二经串行化的最终DOM和所述第三经串行化的最终DOM中的静止动态内容并且缺少真实动态内容,所述第二差异是真实动态内容;以及

将所述第二公共的经串行化的最终DOM呈现给所述设备的所述显示器。

10. 如权利要求1所述的方法,其中,所述位置是统一资源定位符(URL)。

11. 一种用于呈现网页的系统,包括:

客户端浏览器,所述客户端浏览器基于处理由一位置参引的第一Web文档来生成第一经串行化的最终文档对象模型DOM,并且基于处理由所述位置参引的第二Web文档来生成第二经串行化的最终DOM,其中所述第一Web文档关联于包括静止动态内容和第一真实动态内容的第一动态内容,所述第一动态内容是客户端侧生成的并且内联在所述第一经串行化的最终DOM中,并且其中所述第二Web文档关联于包括所述静止动态内容和第二真实动态内容的第二动态内容,所述第二动态内容是客户端侧生成的并且内联在所述第二经串行化的最终DOM中;其中所述客户端浏览器基于所述第一经串行化的最终DOM与所述第二经串行化的最终DOM之间的差异来生成公共的经串行化的最终DOM,其中所述公共的经串行化的最终DOM包括包含在所述第一经串行化的最终DOM和所述第二经串行化的最终DOM中的静止动态内容,其中所述客户端浏览器标识所述第一经串行化的最终DOM与所述第二经串行化的最终DOM之间的一个或多个差异,并且从所述第一经串行化的最终DOM中移除所述一个或多个差异,所述一个或多个差异是真实动态内容;

其中所述客户端浏览器将所述公共的经串行化的最终DOM呈现给设备的显示器。

12. 如权利要求11所述的系统,其中,所述客户端浏览器解析所述第一Web文档,以及所述第一Web文档是标记语言的形式并且包括一组元素和一组规则,其中所述客户端浏览器将所述一组规则应用于所述一组元素中的一个或多个元素并且根据所述解析和所述应用来生成最终DOM,以及其中所述最终DOM包括基于应用所述一组规则所生成的所述第一动态内容。

13. 如权利要求12所述的系统,其中,所述客户端浏览器从所述第一经串行化的最终DOM中移除一个或多个规则,并且使所移除的一个或多个规则内联纳入到所述第一经串行化的最终DOM中。

14. 如权利要求13所述的系统,其中,规则包括客户端侧脚本。

15. 如权利要求14所述的系统,其中,所述客户端侧脚本是JAVASCRIPT编程语言的形式。

16. 如权利要求12所述的系统,其中,所述第一经串行化的最终DOM和所述第二经串行化的最终DOM是最终DOM的不同版本。

17. 一种其上存储有计算机可执行指令的计算机可读介质,所述指令用于执行以下操

作,包括:

基于处理由一位置参引的第一Web文档来生成第一经串行化的最终文档对象模型DOM,所述第一Web文档关联于包括静止动态内容和第一真实动态内容的第一动态内容,所述第一动态内容是客户端侧生成的并且内联在所述第一经串行化的最终DOM中;

基于处理由所述位置参引的第二Web文档来生成第二经串行化的最终DOM,所述第二Web文档关联于包括所述静止动态内容和第二真实动态内容的第二动态内容,所述第二动态内容是客户端侧生成的并且内联在所述第二经串行化的最终DOM中;

基于所述第一经串行化的最终DOM与所述第二经串行化的最终DOM之间的差异来生成公共的经串行化的最终文档对象模型DOM,所述公共的经串行化的最终DOM包括包含在所述第一经串行化的最终DOM和所述第二经串行化的最终DOM中的静止动态内容,其中生成所述公共的经串行化的最终DOM包括:标识所述第一经串行化的最终DOM与所述第二经串行化的最终DOM之间的一个或多个差异,以及从所述第一经串行化的最终DOM中移除所述一个或多个差异,所述一个或多个差异是真实动态内容;以及

将所述公共的经串行化的最终DOM呈现给设备的显示器。

18. 一种用于呈现网页的系统,包括:

用于基于处理由一位置参引的第一Web文档来生成第一经串行化的最终文档对象模型DOM的装置,所述第一Web文档关联于包括静止动态内容和第一真实动态内容的第一动态内容,所述第一动态内容是客户端侧生成的并且内联在所述第一经串行化的最终DOM中;

用于基于处理由所述位置参引的第二Web文档来生成第二经串行化的最终DOM的装置,所述第二Web文档关联于包括所述静止动态内容和第二真实动态内容的第二动态内容,所述第二动态内容是客户端侧生成的并且内联在所述第二经串行化的最终DOM中;

用于基于所述第一经串行化的最终DOM与所述第二经串行化的最终DOM之间的差异来生成公共的经串行化的最终DOM的装置,所述公共的经串行化的最终DOM包括包含在所述第一经串行化的最终DOM和所述第二经串行化的最终DOM中的静止动态内容,其中生成所述公共的经串行化的最终DOM包括:标识所述第一经串行化的最终DOM与所述第二经串行化的最终DOM之间的一个或多个差异,以及从所述第一经串行化的最终DOM中移除所述一个或多个差异,所述一个或多个差异是真实动态内容;以及

用于将所述公共的经串行化的最终DOM呈现给设备的显示器的装置。

包含动态内容和陈旧内容的网站的快速呈现

[0001] 发明人:F·刘、D·雷布尼科夫、V·佩雷尔曼、和T·克拉伊舍克

[0002] 公开领域

[0003] 本公开一般涉及呈现内容,尤其涉及呈现网页。

[0004] 相关申请的交叉引用

[0005] 本申请要求于2014年7月25日提交的美国临时专利申请号62/029,398的优先权权益,该申请要求于2015年7月24日提交的美国专利申请号14/809,046的申请日的优先权,上述两篇申请通过援引而纳入于此。

[0006] 背景

[0007] 用于快速网页显示的传统技术依赖于应用开发者来设计网页,以使得针对第一绘制进行最小的JavaScript和CSS处理。然而,该办法不适合于包含由JavaScript生成的大量动态内容的网页。例如,一些网站可使它们的所有用户界面(UI)元素都由JavaScript生成。另外,依赖于网页开发者来使得网页显示更快可能不足够,因为网页开发者的工具和/或库的选择是有限的。例如,网页开发者经常依赖于通常是在JavaScript中实现的第三方库。

[0008] 用于加速浏览器或内容递送网络(CDN)侧上的操作的常规技术使用高速缓存页。然而,这一技术不适合于高度动态的网站,因为陈旧的信息可能被高速缓存并且稍后被呈现给用户。如果由JavaScript生成的内容根本不被显示以防止显示“陈旧”信息,则包含大量动态内容的那些网页将具有最少的事物要显示。可能期望在避免这些缺点的同时快速地呈现网页。

[0009] 简要概述

[0010] 本公开涉及呈现内容。

[0011] 根据一些实施例,一种用于呈现网页的示例方法包括基于处理由一位置参引的第一Web文档来生成第一经串行化的最终文档对象模型(DOM)。该第一经串行化的最终DOM包括第一内联的动态内容。该方法还包括基于处理由该位置参引的第二Web文档来生成第二经串行化的最终DOM。该第二经串行化的最终DOM包括第二内联的动态内容。该方法进一步包括基于第一经串行化的最终DOM与第二经串行化的最终DOM之间的差异来生成公共的经串行化的最终DOM。该公共的经串行化的最终DOM包括包含在第一经串行化的最终DOM和第二经串行化的最终DOM中的静止动态内容,并且该差异是真实动态内容。该方法还包括将公共的经串行化的最终DOM呈现给设备的显示器。

[0012] 根据一些实施例,一种用于呈现网页的系统包括客户端浏览器,该客户端浏览器基于处理由一位置参引的第一Web文档来生成第一经串行化的最终文档对象模型(DOM),以及基于处理由该位置参引的第二Web文档来生成第二经串行化的最终DOM。第一经串行化的最终DOM包括第一内联的动态内容,并且第二经串行化的最终DOM包括第二内联的动态内容。客户端浏览器基于第一经串行化的最终DOM与第二经串行化的最终DOM之间的差异来生成公共的经串行化的最终DOM。该公共的经串行化的最终DOM包括包含在第一经串行化的最终DOM和第二经串行化的最终DOM中的静止动态内容,并且该差异是真实动态内容。客户端浏览器将公共的经串行化的最终DOM呈现给设备的显示器。

[0013] 根据一些实施例,一种其上存储有计算机可执行指令的计算机可读介质,该计算机可执行指令用于执行以下操作,包括:基于处理由一位置参引的第一Web文档来生成第一经串行化的最终文档对象模型(DOM),该第一经串行化的最终DOM包括第一内联的动态内容;基于处理由该位置参引的第二Web文档来生成第二经串行化的最终DOM,该第二经串行化的最终DOM包括第二内联的动态内容;基于第一经串行化的最终DOM与第二经串行化的最终DOM之间的差异来生成公共的经串行化的最终文档对象模型(DOM),该公共的经串行化的最终DOM包括包含在第一经串行化的最终DOM和第二经串行化的最终DOM中的静止动态内容,并且该差异为真实动态内容;以及将该公共的经串行化的最终DOM呈现给设备的显示器。

[0014] 根据一些实施例,一种用于呈现网页的系统包括:用于基于处理由一位置参引的第一Web文档来生成第一经串行化的最终文档对象模型(DOM)的装置,该第一经串行化的最终DOM包括第一内联的动态内容;用于基于处理由该位置参引的第二Web文档来生成第二经串行化的最终DOM的装置,该第二经串行化的最终DOM包括第二内联的动态内容;用于基于第一经串行化的最终DOM与第二经串行化的最终DOM之间的差异来生成公共的经串行化的DOM的装置,该公共的经串行化的最终DOM包括包含在第一经串行化的最终DOM和第二经串行化的最终DOM中的静止动态内容,并且该差异为真实动态内容;以及用于将该公共的经串行化的最终DOM呈现给设备的显示器的装置。

[0015] 附图简要说明

[0016] 形成本说明书一部分的诸附图解说了本公开的诸实施例,并且与本描述一起进一步用以解释诸实施例的原理。在诸附图中,类似的参考标号可指示相同元件或功能上类似的元件。元件在其中首次出现的附图通常由对应参考标记中的最左边的数字指示。

[0017] 图1是根据一些实施例的示例通信系统。

[0018] 图2是根据一些实施例的用于串行化最终DOM的过程流程。

[0019] 图3是根据一些实施例的用于标识web文档中的真实动态内容的过程流程。

[0020] 图4解说了根据一些实施例的诸客户端与服务器之间的用于提供两个网页之间的差异的通信的示例泳道图。

[0021] 图5解说了根据一些实施例的用于使用双呈现进程办法来显示公共的经串行化的最终DOM和实际网页的示例过程流程。

[0022] 图6是解说根据一些实施例的呈现网页的方法的流程图。

[0023] 图7是适于实现本公开的一个或多个实施例的计算机系统的框图。

[0024] 详细描述

[0025] I. 概览

[0026] II. 示例系统架构

[0027] III. 呈现网页

[0028] A. 基于原始DOM来构建最终DOM

[0029] B. 串行化最终DOM

[0030] 1. 构建与最终DOM的视觉表示相对应的经串行化的最终DOM

[0031] 2. 将最终DOM串行化为文件

[0032] C. 标识并且处理真实和静止动态内容

[0033] D.用于经串行化的最终DOM的管理和递送的机制

[0034] E.用于显示所请求的网页的机制

[0035] 1.双呈现进程办法

[0036] 2.嵌入式JavaScript办法

[0037] IV.示例方法

[0038] V.示例设备

[0039] I.概览

[0040] 将理解,以下公开提供了用于实现本公开的不同特征的许多不同的实施例或者示例。一些实施例可以在没有这些具体细节中的一些或全部的情况下被实践。以下描述了诸组件、模块和布置的具体示例以简化本公开。当然,这些仅仅是示例而并不旨在限制。

[0041] 本公开提供了浏览器/服务器框架以尽可能多地达成非陈旧动态内容的快速呈现的相互矛盾的要求,而无需要求网页开发者变化他们的网站。许多流行的网站具有使用客户端脚本语言来在客户端侧生成的动态内容。可能期望在显示最小陈旧信息的同时快速地并且尽可能多地呈现具有动态内容的网页。静止动态内容可以是指例如帧、布局、侧边栏、菜单、通知模板、朋友列表等。真实动态内容可以是指跨页面加载时相当频繁地变化的任何内容(例如,新闻推送)。

[0042] 为了更快地呈现网页,本公开提供了各种技术,诸如最终DOM的串行化、动态内容标识和处理(以避免显示陈旧信息)、用于生成经串行化的最终DOM的机制(以用于内容的管理和递送)、以及用于显示经串行化的最终DOM和实际网页的机制(以及如何使用最终DOM)。在网页被完全加载之后的DOM可被称为网页的“最终DOM”。

[0043] 包括本公开中描述的一个或多个技术的实施例可更快地显示甚至最动态的网站、避免陈旧信息被显示给用户、和/或保留动态生成但是不变化的项目以供快速显示。另外,可以在具有或者不具有服务器的辅助的情况下在浏览器中实现该解决方案。相应地,对于开发者而言改变原始网站可以是非必需的。

[0044] II.示例系统架构

[0045] 图1是根据一些实施例的示例通信系统100。系统100包括经由网络140来通信地耦合至服务器120的客户端浏览器110。尽管系统100在本文中参照一个客户端浏览器和一个服务器来描述,但是本领域技术人员将领会,系统100可包括多于一个客户端浏览器和/或服务器。

[0046] 浏览器是允许用户查看或者下载网络上(诸如在万维网上的网站或者web应用上)可用的内容的软件应用。内容可包括文本、文件、图像、音频、视频、以及个人通信。网页呈现此类内容并且使用诸如统一资源定位符(URL)之类的统一资源标识符(URI)来定位。可使用拥有网页内容的计算机的网际协议(IP)地址来检索网页。浏览器可使用数个协议和标准以获得或管理内容流。例如,浏览器可使用超文本传输协议(HTTP)来获取内容和网页。HTTP是提供基本请求/响应语义的应用级协议。HTTP可被用于请求和检索来自服务器的对象。

[0047] 服务器120举例而言可以是但不限于:web服务器、电信服务器、数据库服务器等。Web服务器可包含响应于来自客户端的HTTP请求而生成内容的web应用。Web服务器将所生成的内容进行封装并且以HTTP响应的形式来将内容供应给客户端。Web服务器可以用HTTP回复来响应HTTP请求的软件组件。作为解说性示例,web服务器可以是但不限于:

Apache HTTP服务器、Apache Tomcat、微软因特网信息服务器、JBoss应用服务器、WebLogic应用服务器、或Sun Java系统Web服务器。诸商标是其相应所有者的资产。Web服务器可以供应诸如超文本标记语言 (HTML)、可扩展标记语言 (XML)、文档、视频、图像、多媒体特征、或其任何组合之类的内容。这些示例是严格解说性的并且不限定本公开的范围。

[0048] 客户端浏览器110可以在任何计算设备上运行。类似地,服务器120可以使用能够向客户端供应数据的任何计算设备来实现。计算设备的示例包括但不限于:中央处理单元、专用集成电路、计算机、工作站、分布式计算系统、计算机集群、嵌入式系统、自立电子设备、经联网的设备、移动设备(例如,移动电话、智能电话、个人数字助理(PDA)、导航设备、平板设备或移动计算设备)、机架服务器、机顶盒、或其他类型的具有至少一个处理器和存储器的计算机系统。由集群计算环境或服务器场来执行的计算过程可以跨位于相同或不同位置的多个处理器来执行。这种计算设备可包括软件、固件、硬件、或其组合。软件可包括一个或多个应用和操作系统。硬件可包括但不限于:处理器、存储器、以及用户界面显示器。

[0049] 网络140可以是携带数据通信的任何网络或网络的组合,并且在本文中可以被称为计算机网络。此类网络140可以包括但不限于:有线(例如,以太网)或无线(例如,Wi-Fi或3G)网络、局域网、中域网、和/或广域网(诸如因特网)。网络140可支持包括但不限于万维网协议和/或服务的协议和技术。取决于特定的应用或环境,中间web服务器、网关、或其他服务器可以在系统100的诸组件之间被提供。

[0050] 客户端浏览器110可以通过一系列到服务器的请求和来自服务器的响应来检索网页。在一示例中,客户端浏览器110在网络140上向服务器120发送对于网页的请求。服务器120接收该请求,处理该请求,并且返回响应(诸如,所请求的网页的内容)。在一示例中,服务器120响应于客户端浏览器110的请求而向客户端浏览器110返回HTML页面132。HTML页面132是标记语言(HTML)形式的web文档并且包括一组一个或多个元素(例如,标签)和一组一个或多个规则。规则可以例如是被应用于一个或多个元素或级联样式表(CSS)元素的JavaScript。CSS是定义用于格式化并且呈现web文档的呈现规则、性质和辅助构造的声明性语法。通过使用CSS,文档(例如,HTML页面132)中的元素(例如,HTML元素)的外观和格式化可以被快速地改变,而无需添加或移除元素。相应地,HTML页面132的大小可以是较小的,并且操纵该文档的脚本可以是快速的。

[0051] 客户端浏览器110可以下载外部资源以呈现HTML页面132。例如,HTML页面132可包含其他组件(诸如CSS、JavaScript文件、图像、以及其他资源)的地址。客户端浏览器110依次向服务器请求这些中的每一者,从而将它们组合进网页的最终呈现中。客户端浏览器110呈现HTML页面132可能是耗时的,因为客户端浏览器110在能够呈现HTML页面132之前可执行多个动作。例如,HTML页面132可包括JavaScript,在该情形中,客户端浏览器110可以下载并且执行JavaScript以在客户端处生成动态内容,这花费时间。所生成的动态内容随后成为该网页的一部分。

[0052] HTML页面132中的一些内容可以是“真实”动态内容,其很有可能针对每一页面加载而变化(例如,新闻推送)。真实动态内容在对由相同URL参引的页面进行加载时可能不同,并且通常可能在每一次页面加载之后变得陈旧。例如,在第一天,用户可仔细阅读社交媒体网站,并且请求显示由该用户的朋友贴出的新闻推送的网页。布局、侧边栏、以及模板的显示跨页面加载可以是相同的,但是显示给用户的所贴出的新闻推送可能在用户每次重

新刷新或者请求该网页时变化。

[0053] HTML页面132中的一些内容可以是“静止”动态内容,在静止内容在网页加载之间不变化的意义上,这些内容是“静止”的,但是在该内容是由客户端侧脚本(例如,用JavaScript编写的脚本)生成的意义上,其是“动态”的。静止动态内容的示例是用户登录之后的网站的布局、色彩、横幅、格式控制(文本框或按钮)、菜单、背景等。此类网站的整个用户界面可以使用例如JavaScript来生成,并且除非发生不会频繁发生的主要设计变化,否则该内容都保持相同。

[0054] 单个网页具有真实动态内容和静止动态内容两者是常见的。尽管真实动态内容和静止动态内容两者在它们两者都是在运行时生成的意义上是“动态内容”,但是静止动态内容具有跨页面加载是相同的较高概率。可能期望标识网页中的真实动态内容以及静止动态内容并且使它们彼此区分开来,以便更快地呈现网页。

[0055] 为了更快地呈现网页,客户端浏览器110可以将网页高速缓存。如果客户端浏览器110从服务器120接收到与经高速缓存的网页相对应的HTML页面132,则客户端浏览器110可以显示与HTML页面132相对应的经高速缓存的网页。然而,陈旧信息可能被呈现给用户(例如,可能从高速缓存显示诸如昨天的新闻之类的错误的“真实”动态内容)。本公开提供了使用浏览器/服务器框架来在显示最小陈旧信息的同时达成非陈旧动态内容的快速呈现的互相矛盾的要求的技术。

[0056] III. 呈现网页

[0057] 为了完全加载以及显示网页,客户端浏览器110下载HTML页面132、解析HTML页面132、构建HTML页面132的结构文档对象模型(DOM)、并且呈现DOM。在具有大量动态内容的网页中,最耗时的动作可以是下载动态内容(例如,来自外部源的JavaScript脚本)、执行动态内容(例如,执行JavaScript)、以及样式演算。在这些动作被执行之后,客户端浏览器110可以呈现DOM。呈现DOM将实际网页内容描绘在屏幕上。

[0058] 图2是根据一些实施例的用于串行化最终DOM的过程流程。图2提供由服务器120发送至客户端浏览器110以供呈现的“原始”HTML页面132的示例。客户端浏览器110处理HTML页面132,并且将作为HTML页面132的表示的原始DOM 202构建为一组结构化的具有属性和方法的节点和对象。原始DOM 202被形象化为HTML页面132内的标记元素的阶层的树形结构。

[0059] HTML页面132中的对象属性被直接映射到HTML元素的属性。在HTML语法与DOM之间存在直接映射。在原始DOM 202中,“html”节点204对应于HTML页面132中的<html>元素,“主体”节点206对应于HTML页面132中的<主体>元素,“文本1”节点208对应于HTML页面132中的“文本1”,“Js1.js”节点210对应于HTML页面132中的<js1.js>元素,并且“Css1.css”节点212对应于HTML页面132中的<链接css1.css>元素。HTML页面132还包括两个结束标签</主体>和</html>。DOM提供对样式的编程化访问。DOM中的CSS对象是“活动的”,亦即,样式表中的变化在经计算和实际的样式中反映。相应地,可以使用简单的基于脚本的编程来改变个体元素和样式规则上的内联样式。脚本可以按例如JavaScript、微软JScript、或者微软Visual Basic脚本编辑(VBScript)来编写。

[0060] A. 基于原始DOM来构建最终DOM

[0061] 客户端浏览器110下载HTML页面132中发现的子资源。示例子资源是可被应用于包

括在HTML页面132中的一个或多个元素的规则。示例规则包括CSS元素和JavaScript脚本。客户端浏览器110可以在HTML页面132加载的同时或者在HTML页面132加载之后立即编译和执行子资源。子资源的编译和执行可以是同步操作。在子资源的编译和执行期间,客户端浏览器110可以处理例如CSS元素、处理图像、和/或执行JavaScript脚本。

[0062] 作为执行子资源的结果,客户端浏览器110可以生成动态内容、从html标签构建附加的DOM节点、或者从该DOM中移除DOM节点。在具有动态内容的网页被完全加载之后,处于该阶段的DOM可被称为最终DOM,其不仅包含原始HTML页面132中定义的静态内容,还包含由子规则生成的新添加和/或修改的动态内容(例如,JavaScript脚本、CSS中指定的样式等)。在此阶段,最终DOM元素的样式已经被解决。

[0063] 在图2中,客户端浏览器110处理HTML页面132并且生成纳入网页的动态内容的最终DOM 220。执行JavaScript可添加不在发送自服务器120的原始HTML页面132中的附加DOM节点。为了生成最终DOM 220,客户端浏览器110执行由“Js1.js”节点210(其与HTML页面132中的“js1.js”元素相关联)表示的JavaScript,以及响应于执行该脚本而生成由“表单”节点222表示的动态内容。由“Js1.js”节点210表示的JavaScript脚本通过将“<表单>”标签添加在HTML字节流(例如,文件.写入())中或者通过直接向原始DOM 202添加一“表单”DOM节点来动态地向原始DOM 202添加一“表单”。如图2中解说的,最终DOM 220包括不包含在原始DOM 202中的“表单”节点222。类似于原始DOM 202,最终DOM 220包括诸节点:“html”节点204、“主体”节点206、“文本1”节点208、“Js1.js”节点210、以及“Css1.css”节点212。除了这些节点之外,最终DOM 220还包括“表单”节点222。客户端浏览器110将由“Css1.css”节点212指定的样式应用于HTML页面132中的恰适HTML标签并且向用户呈现最终DOM 220。

[0064] B. 串行化最终DOM

[0065] 客户端浏览器110通过构建对应于最终DOM的视觉表示的DOM来呈现最终DOM 220,并且在耦合至客户端浏览器110的设备的显示器上描绘最终DOM。客户端浏览器110可以通过构建经串行化的最终DOM 230并且将其串行化为文件来串行化最终DOM 220。将经串行化的最终DOM串行化以标识真实动态内容和静止动态内容并且快速地呈现网页可以是有利的,如本公开中进一步讨论的。

[0066] 1. 构建与最终DOM的视觉表示相对应的经串行化的最终DOM

[0067] 客户端浏览器110可以通过标识最终DOM 220中的动态内容、移除表示包括来自最终DOM 220的动态内容的一组规则的节点(例如,“Js1.js”节点210和“Css1.css”节点212)、以及将动态内容内联到经串行化的最终DOM 230中来生成经串行化的最终DOM 230。相应地,客户端浏览器110从最终DOM 220中移除JavaScript脚本和CSS样式并且将此动态内容内联在经串行化的最终DOM 230中。在一些示例中,客户端浏览器110通过将最终DOM 220中的一个或多个非内联规则移至内联规则来生成经串行化的最终DOM 230。

[0068] 经串行化的最终DOM 230包括节点204、206、208、和222。最终DOM 220中的节点204和206可以与经串行化的最终DOM 230中的节点204和206相同。由最终DOM 220中的“Css1.css”节点212表示的恰适样式被应用于恰适的HTML标签/DOM节点(例如,由最终DOM 220中的“文本1”节点208和/或“表单”节点222表示的文本)。内联样式是已经使用样式属性被应用于(诸)HTML元素的CSS样式指派。在一些示例中,客户端浏览器110可以通过复制存储在由HTML页面132中的“<链接css1.css>”元素表示的位置处的外部CSS文件的内容,并且

在HTML页面132的头部部分中的样式标签之间粘贴该内容来内联CSS样式。

[0069] 2. 将最终DOM串行化为文件

[0070] 客户端浏览器110可以将经串行化的最终DOM 230串行化为经串行化的文件240 (例如, HTML文件), 其中所有元素样式都是内联的。作为最终DOM 220的串行化的经串行化的文件240是具有正确样式的、包含静态和动态内容两者的“平坦化”文件。经串行化的文件240可以是将相同内容作为最终显示给用户的“实际网页”来示出的HTML文件。客户端浏览器110在高速缓存中存储经串行化的文件240以用于稍后的检索和加载。下一次, 在接收到对于显示由与HTML页面132相关联的URL参引的网页的请求之际, 客户端浏览器110可以加载经串行化的文件240。经串行化的文件240具有与HTML页面132相同的外观, 但是可以被显示地快得多, 因为不必应用规则和/或搜索外部资源。例如, 对于客户端浏览器110而言, 可以不必执行JavaScript或者应用CSS元素, 这些在显示网页中通常是耗时的步骤。相应地, 在下一次请求网页时可以达成更快的页面显示。

[0071] 如果相同网页的经串行化的最终DOM是可用的, 则经串行化的文件可以首先 (较快地) 被显示, 而同时实际网页正在后台中加载。实际网页可以在其在后台中完成加载时被显示。然而, 经串行化的文件240可能显示陈旧信息。例如, 如果网站包括新闻推送, 则可能向用户显示过时的新闻推送。为了避免陈旧信息的显示, 客户端浏览器110可以标识经串行化的文件240中的动态内容, 并且移除一些动态内容或者用通用和/或虚拟的占位符来替代动态内容。

[0072] 为了标识动态内容, 客户端浏览器110可以在应用HTML文档中的任何规则 (例如, JavaScript和CSS元素) 之前标识最终DOM与原始主HTML的对应的原始DOM之间的差异。客户端浏览器110可以确定最终DOM 220与原始DOM 202之间的差异, 从最终DOM中移除该差异, 并且随后显示该最终DOM, 其中该差异是动态内容。然而, 这一办法可能移除所有的动态内容 (真实动态内容和静止动态内容两者)。该净效应可类似于禁用浏览器中的JavaScript并且随后加载原始HTML。另外, 该净效应也可能是不同的, 因为被保持在最终DOM中而不在最初原始HTML中的样式也可由JavaScript修改。对于具有许多静止动态内容的网站而言, 几乎没有什么可被显示给用户, 因为没有或者几乎没有用户界面元素被显示给用户。相应地, 在真实动态内容和静止动态内容之间作出区分并且仅移除真实动态内容可能是期望的。

[0073] 在一些实施例中, 用于生成经串行化的DOM的方法包括解析第一web文档。第一web文档可以是标记语言的形式并且包括一组元素和一组规则。该方法还可包括根据解析web文档并且将该组规则应用于web文档来将该组规则应用于该组元素中的一个或多个元素并且构建最终DOM。该方法可进一步包括在计算设备的显示器上呈现最终DOM并且标识包括在最终DOM中的一组规则。该方法还可包括从最终DOM中移除该组规则并且将最终DOM串行化为第二web文档。经串行化的最终DOM可以在浏览器的高速缓存中被高速缓存。

[0074] C. 标识并且处理真实和静止动态内容

[0075] 为了避免陈旧信息的显示, 客户端浏览器110标识经串行化的文件240中的真实动态内容和静止动态内容, 并且移除至少一些静止动态内容或者用通用和/或虚拟的占位符来替代静止动态内容。为了标识真实动态内容, 客户端浏览器110可以维护与网页加载相关联的最终DOM, 并且在每次客户端浏览器110加载对应于一个或多个相同URL的网页时存储相应的最终DOM。在每次页面加载之后, 客户端浏览器110可以保存经串行化的文件240 (例

如,串行化在HTML文件中的经串行化的最终DOM)以供稍后使用。

[0076] 在一示例中,客户端浏览器110生成与对于相同网页(由相同URL参引的网页)的请求相关联的多个最终DOM,并且在每次客户端浏览器110加载相同网页时,在高速缓存中存储相应的最终DOM。客户端浏览器110随后确定用于相同网页的多个最终DOM之间的差异。在经比较的多个最终DOM中的不同部分可被标识为真实动态内容。

[0077] 图3是根据一些实施例的用于标识web文档中的真实动态内容的过程流程。图3包括经串行化的最终DOM 302和经串行化的最终DOM 304。经串行化的最终DOM 302和304可以如以上所指示的那样来生成并且可以与相同的URL相关联。经串行化的最终DOM 302和304可以是相同网页的经串行化的最终DOM的诸版本,并且被维护和存储在存储器中。

[0078] 经串行化的最终DOM 302包括“html”节点306、“主体”节点308、“<菜单>”节点310、“<馈送>”节点312、“项目1”节点314、“项目2”节点316、以及“新闻A”节点318。经串行化的最终DOM 304包括“html”节点306、“主体”节点308、“<菜单>”节点310、“<馈送>”节点312、“项目1”节点314、“项目2”节点316、以及“新闻B”节点320。“html”节点306、“主体”节点308、“<菜单>”节点310、以及“<馈送>”节点312可以是表示HTML元素/子树的“静态”节点。“项目1”节点314、“项目2”节点316、“新闻A”节点318、以及“新闻B”节点320可以是由JavaScript生成的动态内容。如果客户端浏览器110要在第1和第2天从社交媒体网站加载网页,则客户端浏览器110可以将与这两个网页加载相关联的经串行化的最终DOM 302和304的两个版本存储到高速缓存或者数据库中。

[0079] 在一示例中,客户端浏览器110比较经串行化的最终DOM 302与经串行化的最终DOM 304,并且标识已经变化的一个或多个部分。该一个或多个所标识的部分是跨诸经串行化的最终DOM的真实动态内容。经串行化的最终DOM 302包括“新闻A”内容并且经串行化的最终DOM 304包括“新闻B”内容。相应地,“新闻X”被标识为真实动态内容。该真实动态内容(例如,新闻推送和该新闻的内容)在这两个经串行化的最终DOM之间是不同的,但是静止动态内容(例如,用户界面元素)保持相同。如果客户端浏览器110要高速缓存经串行化的最终DOM 302,并且向用户显示经串行化的最终DOM 302而不是经串行化的最终DOM 304,则可以向用户呈现陈旧的新闻推送(由“新闻A”内容来表示)而不是最新近的新闻推送(由“新闻B”的内容来表示)。为了避免向用户显示陈旧的信息,客户端浏览器110移除真实动态内容以产生公共的经串行化的最终DOM 330。公共的经串行化的最终DOM 330是具有从经串行化的最终DOM 302或者经串行化的最终DOM 304中移除的真实动态内容的DOM。

[0080] 在一些示例中,客户端浏览器110通过从经串行化的最终DOM 302(或者经串行化的最终DOM 304)中移除表示真实动态内容的节点来移除真实动态内容。如果表示真实动态内容的节点被移除,则网页的布局可以变化。例如,大横幅图像可以是真实动态内容,并且如果该大横幅图像被移除,则该网页布局的其余部分可能对于用户而言显得奇怪。在图3中解说的示例中,不是移除表示真实动态内容的节点(例如,节点318),而是客户端浏览器110可以用充当占位符元素的虚节点332来替代表示真实动态内容的节点。虚节点332可以具有相同的大小(或者甚至色彩方案,如果举例而言它是大图像的话)以保持网页布局与之前相同。

[0081] 真实动态内容还可以从子DOM中被移除。例如,诸iFrame可能具有它们自己的子DOM,并且HTML文档一般是不可高速缓存的。相应地,如果源URL没有变化,则客户端浏览器

110可以在公共的经串行化的最终DOM中保持iFrame或者用相等大小的虚节点来替代iFrame。

[0082] 在通过移除表示真实动态内容的节点或者用虚节点替代该节点来移除真实动态内容(例如,“新闻X”)之后,经串行化的最终DOM的其余部分可以用尽可能少的陈旧内容来显示。公共的经串行化的最终DOM 330不包括潜在地陈旧的内容。通过标识原始HTML页面(从网站中下载的实际HTML页面)与经串行化的最终DOM的HTML之间的差异,或者通过标识经串行化的最终DOM的不同版本之间的差异,可以生成公共的经串行化的最终DOM而不包含任何真实动态内容。经串行化的最终DOM的比较可以在不同时间被执行以完善最终DOM,以便排除真实动态内容。此外,多于两个不同版本的经串行化的最终DOM可以被比较。

[0083] D. 用于经串行化的最终DOM的管理和递送的机制

[0084] 在一些实施例中,在客户端侧上,客户端浏览器110可以将经串行化的最终DOM保存到由网页的URL索引的本地高速缓存或者数据库中。客户端浏览器110还可以为经串行化的最终DOM设置有效期。标识不同版本的经串行化的最终DOM之间的(诸)差异可以在后台中周期性地执行。在一示例中,每次网页被加载时,可以使用URL来查询本地高速缓存或者数据库以确定相同URL的经串行化的最终DOM是否可以被使用。如果服务器从客户端接受要在下一次相同的URL被请求时使用的元数据,则客户端浏览器110也可以向服务器提交经串行化的最终DOM。

[0085] 如以上描述的,在每个页面加载之后,经串行化的最终DOM HTML可以被生成以供稍后使用和存储。该经串行化的最终DOM的生成和比较可以在客户端侧(在客户端浏览器110中)或者服务器侧(例如,web代理或者CDN边缘服务器)处执行。相应地,尽管客户端浏览器110被描述为生成和/或存储最终DOM和经串行化的最终DOM,但是还应当理解,服务器也可以生成和/或存储最终DOM和/或经串行化的最终DOM(例如,在内容递送网络(CDN)中)。例如,每次网页被请求时,CDN可以在被命中之时高速缓存经串行化的最终DOM。

[0086] 在一些实施例中,在服务器侧上,服务器(例如,CDN边缘服务器或者代理服务器)可以从多个用户接收对于相同URL的请求。在一示例中,每次由服务器服务URL请求时,该服务器在服务器侧生成经串行化的最终DOM并且在服务器上保存针对该URL的经串行化的最终DOM。该服务器可以通过例如使用诸如PhantomJS之类的无头浏览器来生成最终DOM。

[0087] 图4解说了根据一些实施例的诸客户端与服务器之间的用于提供两个网页之间的差异的通信的示例泳道图。在图4中,服务器“a.com”406从客户端浏览器110和410接收多个请求。客户端浏览器110可以发送对于由URL“http://a.com”402参引的网页408的请求。该请求可以从客户端浏览器110发送至CDN服务器404,该CDN服务器404将该请求传递至存储由URL“http://a.com”402参引的网页408的服务器“a.com”406。服务器“a.com”406将网页408(由索引.html(1)表示)发送至CDN服务器404,该CDN服务器404将网页传递至客户端浏览器110。CDN服务器404可以基于“索引.html(1)”来生成经串行化的最终DOM 420并且在CDN服务器404上保存关于URL“http://a.com”402的经串行化的最终DOM 420。

[0088] 在稍后的时间点,客户端浏览器410可以发送对于由URL“http://a.com”402参引的网页408的请求。该请求可以从客户端浏览器410发送至CDN服务器404,该CDN服务器404将该请求传递至服务器“a.com”406。服务器“a.com”406将网页408(由索引.html(2)表示)发送至CDN服务器404,该CDN服务器404将网页传递至客户端浏览器110。CDN服务器404可以

基于“索引.html (2)”来生成经串行化的最终DOM 430并且在CDN服务器404上保存关于URL “http://a.com”402的经串行化的最终DOM 430。

[0089] 相同URL (例如, “http://a.com”) 但是用于不同用户 (例如, 客户端浏览器110和410) 的经串行化的最终DOM的不同版本 (例如, 经串行化的最终DOM 420和经串行化的最终DOM 430) 可以彼此比较以标识不同的内容, 以使得最后的经串行化的最终DOM可以仅包含对于所有用户而言公共的内容。最后的不包含真实动态内容的经串行化的最终DOM可被称为公共的经串行化的最终DOM。

[0090] CDN服务器404可继续生成经串行化的最终DOM并且将经串行化的最终DOM的不同版本彼此进行比较。在图4中, 如关于图3所讨论的, CDN服务器404生成经串行化的最终DOM 420和430, 并且比较它们以确定它们的 (诸) 差异。CDN服务器404可以将经串行化的最终DOM 420和430之间的 (诸) 差异 (该差异可以被写为“Diff=S (索引1) - (索引2)”) 标识为真实动态内容, 并且从经串行化的最终DOM 420或430之一中移除真实动态内容 (例如, 通过从经串行化的最终DOM 420或430中移除表示真实动态内容的节点, 或者通过用经串行化的最终DOM 420或430中的虚节点来替代表示真实动态内容的节点)。

[0091] 在经串行化的最终DOM的 (诸) 未改变部分稳定之后, CDN服务器404可以生成移除经串行化的最终DOM 420与430之间的 (诸) 差异的公共的经串行化的最终DOM 440, 并且为客户端供应公共的经串行化的最终DOM。公共的经串行化的最终DOM 440是由“s. 索引.html = S (索引1) - Diff”来表示的。在稍后的时间点, 客户端浏览器410可以发送对于网页408的另一请求。

[0092] E. 用于显示所请求的网页的机制

[0093] 使用本公开中讨论的技术, 网页可以按各种各样的方式来被呈现。在一示例中, 响应于此请求, CDN服务器404可以向客户端浏览器410供应不包括任何动态内容的公共的经串行化的最终DOM 440。CDN服务器404可以使用不同的URL 442来将公共的经串行化的最终DOM 440递送至客户端浏览器410。在另一示例中, CDN服务器404可将真实动态内容嵌入或内联到主HTML中以由JavaScript显示管理器使用。相应地, 公共的经串行化的最终DOM可以在客户端处和/或在服务器处生成, 并且通过单独的URL来递送至客户端或者被嵌入在主HTML文档中。

[0094] 1. 双呈现进程办法

[0095] 在一些实施例中, 客户端浏览器110可以使用双呈现进程办法来显示网页。图5解说了根据一些实施例的用于使用双呈现进程办法来显示公共的经串行化的最终DOM和实际网页的示例过程流程。在图5中, 客户端浏览器110向服务器请求“foo.com”。响应于该请求, 客户端浏览器110可以创建两个呈现进程, 一个呈现进程用于从高速缓存或者从服务器 (例如, CDN服务器404或代理服务器) 请求和呈现公共的经串行化的最终DOM, 以及一个呈现进程用于要被显示给用户的实际网页。

[0096] 第一呈现进程是“快速进程”并且被用于达成网页正被快速地呈现的错觉。在第一呈现进程中, 客户端浏览器110从高速缓存或者从服务器 (使用“s.foo.com”) 加载公共的经串行化的最终DOM, 并且首先向用户显示公共的经串行化的最终DOM。相应地, 客户端浏览器110首先显示由“s.foo.com”参引的网页。用户能够快速查看网页中的静止动态内容, 因为该动态内容已经被移除 (例如, JavaScript或者CSS元素)。第二呈现进程是“完整进程”并

且被用于在后台中请求和呈现实际网页。在第二呈现进程中,客户端浏览器110加载实际网页(使用“foo.com”),但是实际网页还未向用户示出。客户端浏览器110在其完成对实际网页的呈现时向用户显示实际网页,从而确保最终向用户示出的网页不是破损的。

[0097] 参照回到图4并且使用双呈现进程办法,在下次由URL“http://a.com”402参引的网页被加载时,客户端浏览器110可以在前台中向用户显示公共的经串行化的最终DOM 440,而同时后台进程在后台中呈现实际网页。公共的经串行化的最终DOM 440是经串行化的最终DOM的当前版本,并且客户端浏览器110在后台中执行经串行化的最终DOM的先前版本与经串行化的最终DOM的当前版本之间的另一差异。相应地,客户端浏览器110可以呈现诸进程以处理经串行化的最终DOM和实际网页两者,但是首先示出经串行化的最终DOM。

[0098] 2. 嵌入式JavaScript办法

[0099] 在一些实施例中,客户端浏览器110可以使用嵌入式JavaScript办法。在嵌入式JavaScript办法中,经串行化的最终DOM内容被嵌入在网页的主HTML中。相应地,网页还包括JavaScript。客户端浏览器110首先显示经串行化的最终DOM内容,随后转变成显示实际网页。如果经串行化的最终DOM是从服务器侧获得的,则该嵌入式JavaScript办法可被执行。服务器可以通过将经串行化的最终DOM嵌入在原始HTML中来修改原始HTML,并且客户端浏览器110可以使用JavaScript在不同的时间加载和显示经串行化的最终DOM和实际网页。

[0100] IV. 示例方法

[0101] 图6是解说根据一些实施例的呈现网页的方法600的流程图。方法600并不旨在进行限制并且可以用于其他应用中。

[0102] 图6中,方法600包括框602-608。在框602中,基于处理由一位置参引的第一Web文档来生成第一经串行化的最终DOM,第一经串行化的最终DOM包括第一内联的动态内容。在一示例中,客户端浏览器110基于处理由URL参引的HTML页面132来生成经串行化的最终DOM 302,经串行化的最终DOM 302包括内联的动态内容。在框604中,基于处理由该位置参引的第二Web文档来生成第二经串行化的最终DOM,第二经串行化的最终DOM包括第二内联的动态内容。在一示例中,客户端浏览器110基于处理由该位置参引的第二Web文档来生成经串行化的最终DOM 304,经串行化的最终DOM 304包括第二内联的动态内容。

[0103] 在框606中,基于第一经串行化的最终DOM与第二经串行化的最终DOM之间的差异来生成公共的经串行化的最终DOM,该公共的经串行化的最终DOM包括包含在第一经串行化的最终DOM和第二经串行化的最终DOM中的静止动态内容,并且该差异是真实动态内容。在一示例中,客户端浏览器110基于经串行化的最终DOM 302与304之间的差异来生成公共的经串行化的最终DOM 330,该公共的经串行化的最终DOM 330包括包含在经串行化的最终DOM 302与304中的静止动态内容,并且该差异是真实动态内容。在框608中,该公共的经串行化的最终DOM被呈现给设备的显示器。在一示例中,客户端浏览器110将公共的经串行化的最终DOM 330呈现给设备的显示器。

[0104] 在一些示例中,可以针对由客户端浏览器110加载的任何数目的网页来执行框602-608中解说的一个或多个动作。另外,也要理解,附加的过程可以在以上所讨论的框602-608之前、期间或之后执行。还应理解,本文描述的方法600的一个或多个框可按期望被省略、组合或以不同的顺序执行。

[0105] 图1-6仅仅是示例,其不应当不恰当地限制权利要求的范围。例如,为了简明起见,

本公开可以将客户端应用描述为客户端浏览器,但是这并不旨在进行限制并且应当理解其他客户端应用在本公开的范围之内。

[0106] 本公开提供了在避免陈旧内容的同时达成对高度动态网站的快速呈现的技术。另外,来自不同方面的办法可以被组合以形成工作框架。在一示例中,客户端浏览器可以采用双呈现办法以利用在服务器侧生成的经串行化的最终DOM,其中所有的真实动态内容被移除。在另一示例中,浏览器可以高速缓存所有的不具有动态内容的经串行化的最终DOM。另外,通过查看高速缓存中的内容和由客户端浏览器所请求的内容以及由服务器返回的内容,客户端浏览器110可以标识是否正在递送和处理用以促成快速呈现的任何经串行化的最终DOM或者任何形式的资源。

[0107] V. 示例设备

[0108] 图7是适于实现本文公开的任何实施例的示例计算机系统700的框图。在各种实现中,计算机系统700可以是服务器。在各种实现中,客户端浏览器110可以在计算机系统700中运行。

[0109] 计算机系统700包括耦合至输入/输出(I/O) 704组件的控制单元701。控制单元701可包括一个或多个处理单元(CPU) 770,并且可附加地包括一个或多个存储设备,该一个或多个存储设备各自选自包括以下各项的组:软盘、柔性盘、硬盘、磁带、任何其它磁介质、CD-ROM、任何其它光学介质、随机存取存储器(RAM)、可编程只读存储器(PROM)、可擦除ROM(EPROM)、闪速EPROM、任何其它存储器芯片或卡带、和/或处理器或计算机被适配成从其读取的任何其它介质。这一个或多个存储设备可包括对于使用计算机网络(未示出)耦合至计算机系统700的一个或多个计算设备和/或计算机程序(例如,客户端)可用的所存储信息。计算机网络可以是任何类型的网络,包括LAN、WAN、内联网、因特网、云、和/或它们的能够互连系统中的计算设备和/或计算机程序的任何网络组合。

[0110] 计算机系统700包括总线702或者其他用于在计算机系统700的各个组件之间传达信息数据、信号以及信息的通信机制。组件包括用于处理用户动作(诸如,从按键板/键盘选择按键或者选择一个或多个按钮或链接等,以及向总线702发送相应的信号)的输入/输出(I/O) 组件704。I/O组件704还可包括输出组件(诸如显示器711)和输入控件(诸如光标控件713(诸如,键盘、按键板、鼠标等))。还可包括音频I/O组件705以允许用户使用语音通过将音频信号转换为信息信号来输入信息。音频I/O组件705可允许用户听见音频。

[0111] 收发机或网络接口706经由至网络的通信链路718在计算机系统700与其他设备之间传送和接收信号。在一实施例中,该传输是无线的,尽管其他传输机制和方法也可以是适合的。另外,显示器711可经由通信链路718耦合至控制单元701。

[0112] CPU 770(其可以是微控制器、数字信号处理器(DSP)、或其他处理组件)处理这些各种信号,诸如以供在计算机系统700的显示器711上显示或者经由通信链路718传输至其他设备。CPU 770还可控制信息(诸如cookies或IP地址)至其他设备的传输。计算机系统700的组件还包括系统存储器组件714(例如,RAM)、静态存储组件716(例如,ROM)、和/或计算机可读介质717。计算机系统700通过由CPU 770和其他组件执行在系统存储器组件714中包含的一个或多个指令序列来执行特定操作。逻辑可被编码在计算机可读介质717中,该计算机可读介质717可以指代参与向CPU 770提供指令以供执行的任何介质。此类介质可采取多种形式,包括但并不限于非易失性介质、易失性介质、和传输介质。在各种实现中,非易失

性介质包括光盘或磁盘、或固态驱动器,易失性介质包括动态存储器(诸如系统存储器组件714),并且传输介质包括同轴电缆、铜线和光纤(包括包含总线702的导线)。在一实施例中,逻辑被编码在非瞬态计算机可读介质中。计算机可读介质717可以是能包含、存储、传达、传播或传输由CPU 770使用或者结合其使用的指令的任何装置。计算机可读介质717可以是电子、磁性、光学、电磁、红外、或半导体设备或传播介质、或任何其他存储器芯片或卡带、或计算机被适配成从其读取的任何其他介质。在一示例中,传输介质可采用声波或光波的形式,诸如那些在无线电波、光和红外数据通信期间生成的波。

[0113] 在本公开的各种实施例中,用于实践本公开的指令序列(例如,方法600)的执行可以由计算机系统700执行。在本公开的各种其他实施例中,由通信链路718耦合至网络(举例而言,诸如LAN、WLAN、PTSN、和/或各种其他有线或无线网络,包括电信、移动、和蜂窝电话网络)的多个计算机系统700可以执行指令序列以彼此协同地实践本公开。

[0114] 在适用的情况下,由本公开提供的各种实施例可使用硬件、软件、或硬件和软件的组合来实现。同样在适用的情况下,本文阐述的各种硬件组件和/或软件组件可被组合成包括软件、硬件、和/或两者的复合组件而不脱离本公开的精神。在适用的情况下,本文阐述的各种硬件组件和/或软件组件可被分成包括软件、硬件、或两者的子组件而不脱离本公开的精神。另外,在适用的情况下,构想了软件组件可被实现为硬件组件,反之亦然。

[0115] 根据本公开的应用软件可被存储在一个或多个计算机可读介质上。还构想了本文所标识的应用软件可使用一个或多个通用或专用计算机和/或经联网和/或其他形式的计算机系统来实现。在适用的情况下,本文描述的各个框的排序可以改变、组合成复合框、和/或分成子框以提供本文描述的特征。

[0116] 前述公开并不旨在将本公开限定为所公开的精确使用形式或特定使用领域。如此,构想了对本公开的各种替换实施例和/或修改(无论是在本文中显式描述还是暗示)鉴于本公开是可能的。可在形式和细节上作出改变而不脱离本公开的范围。因此,本公开仅由权利要求来限定。

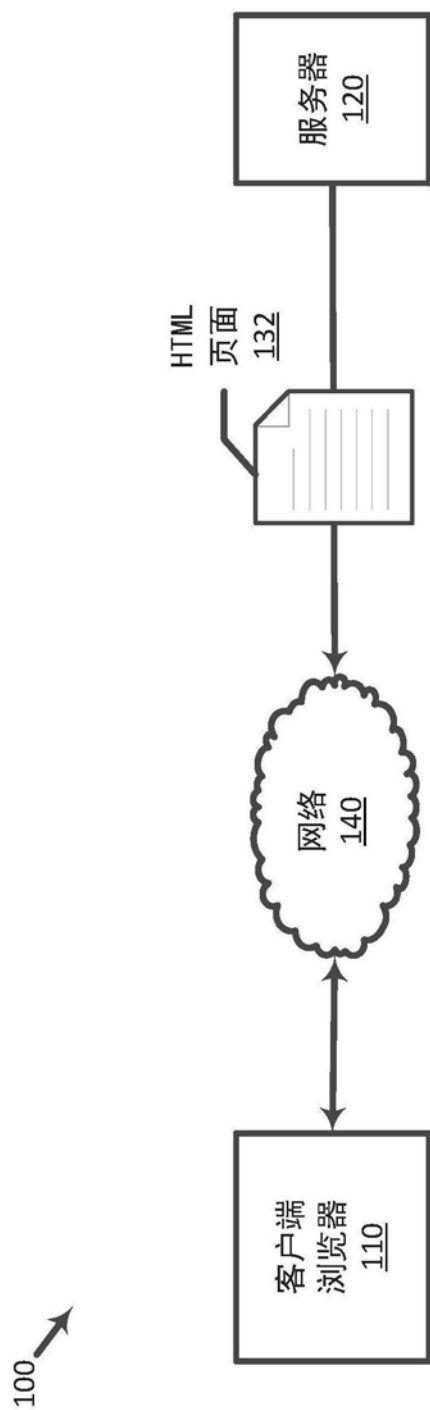


图1

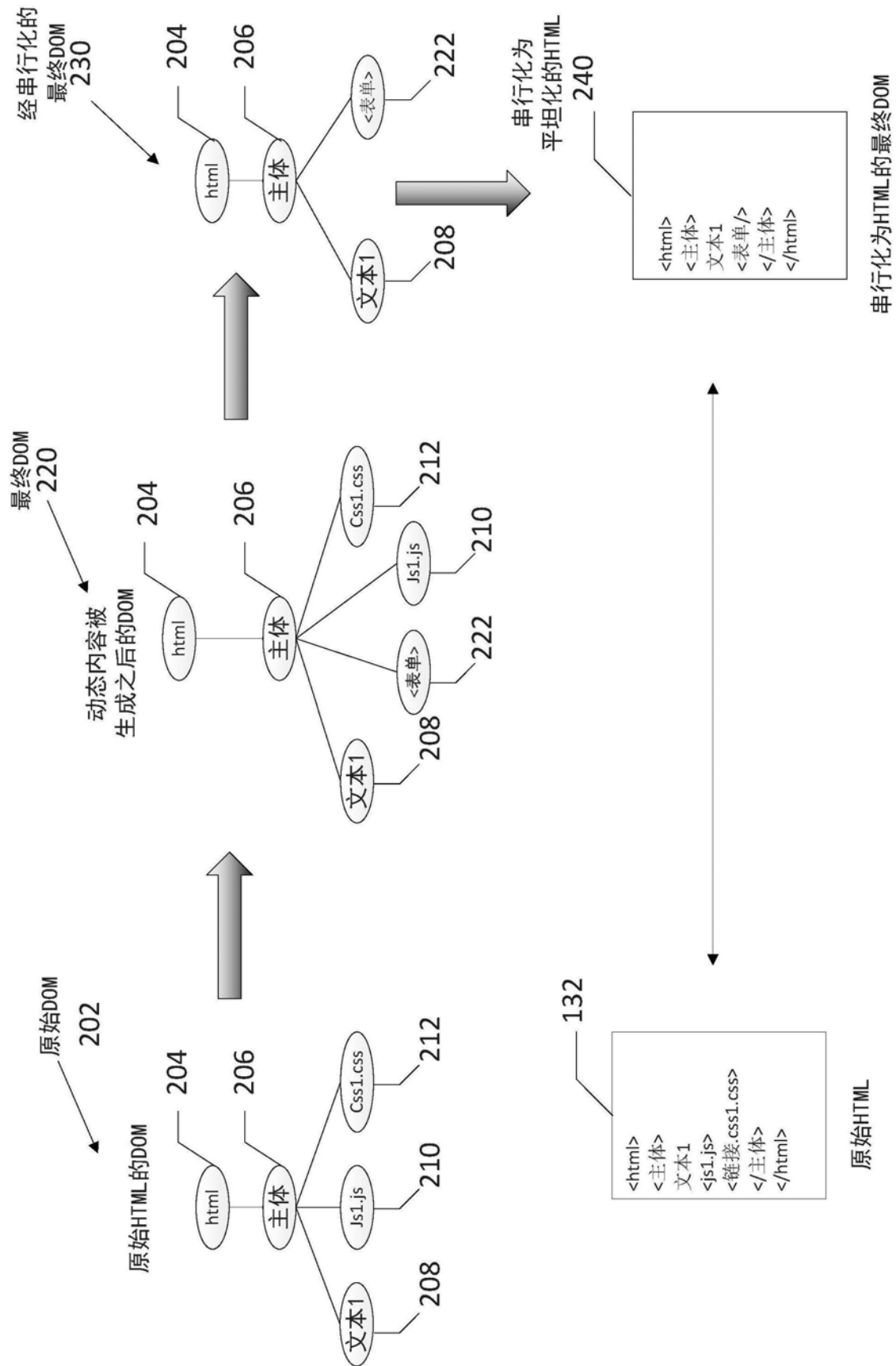


图2

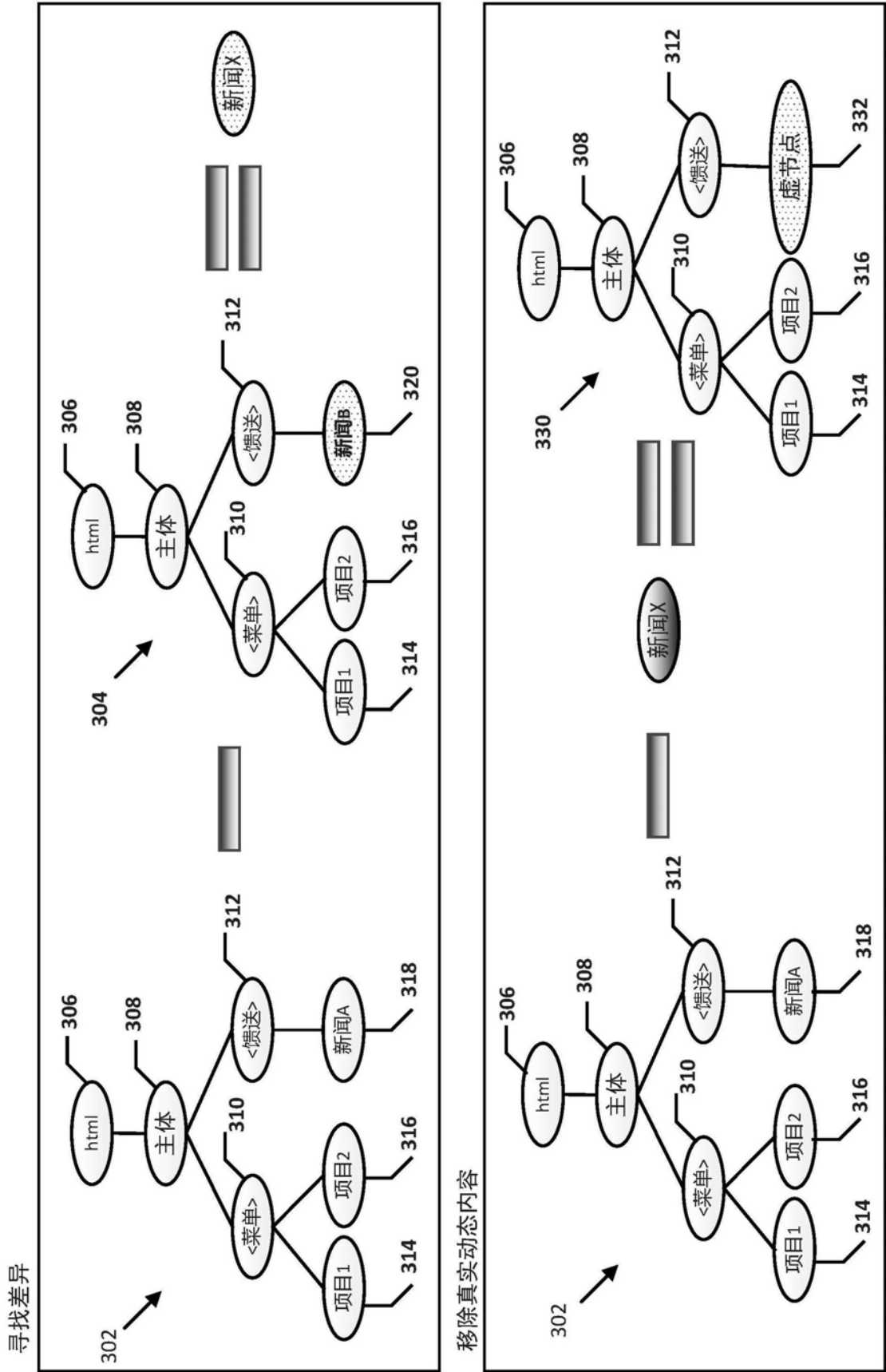


图3

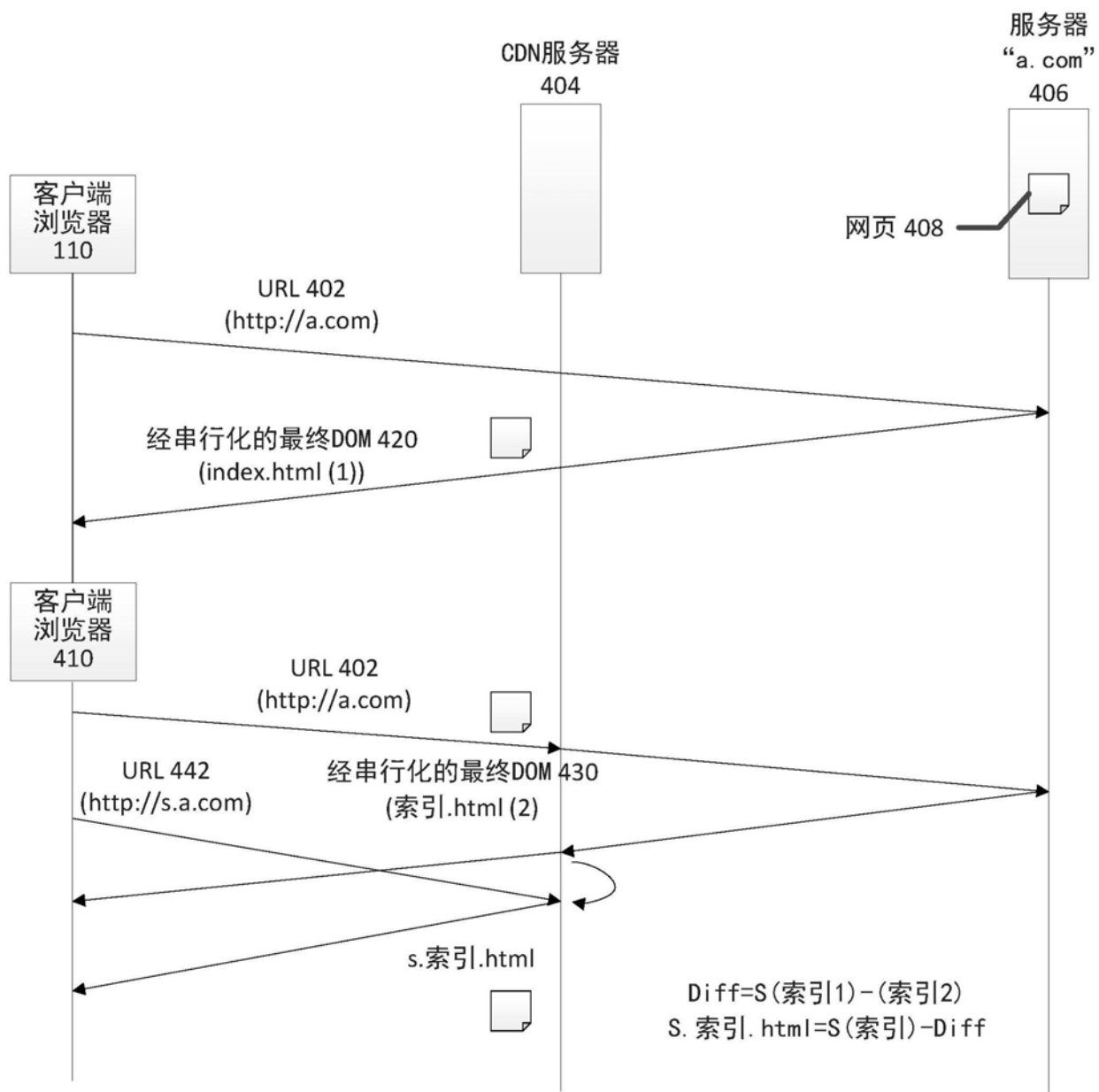


图4

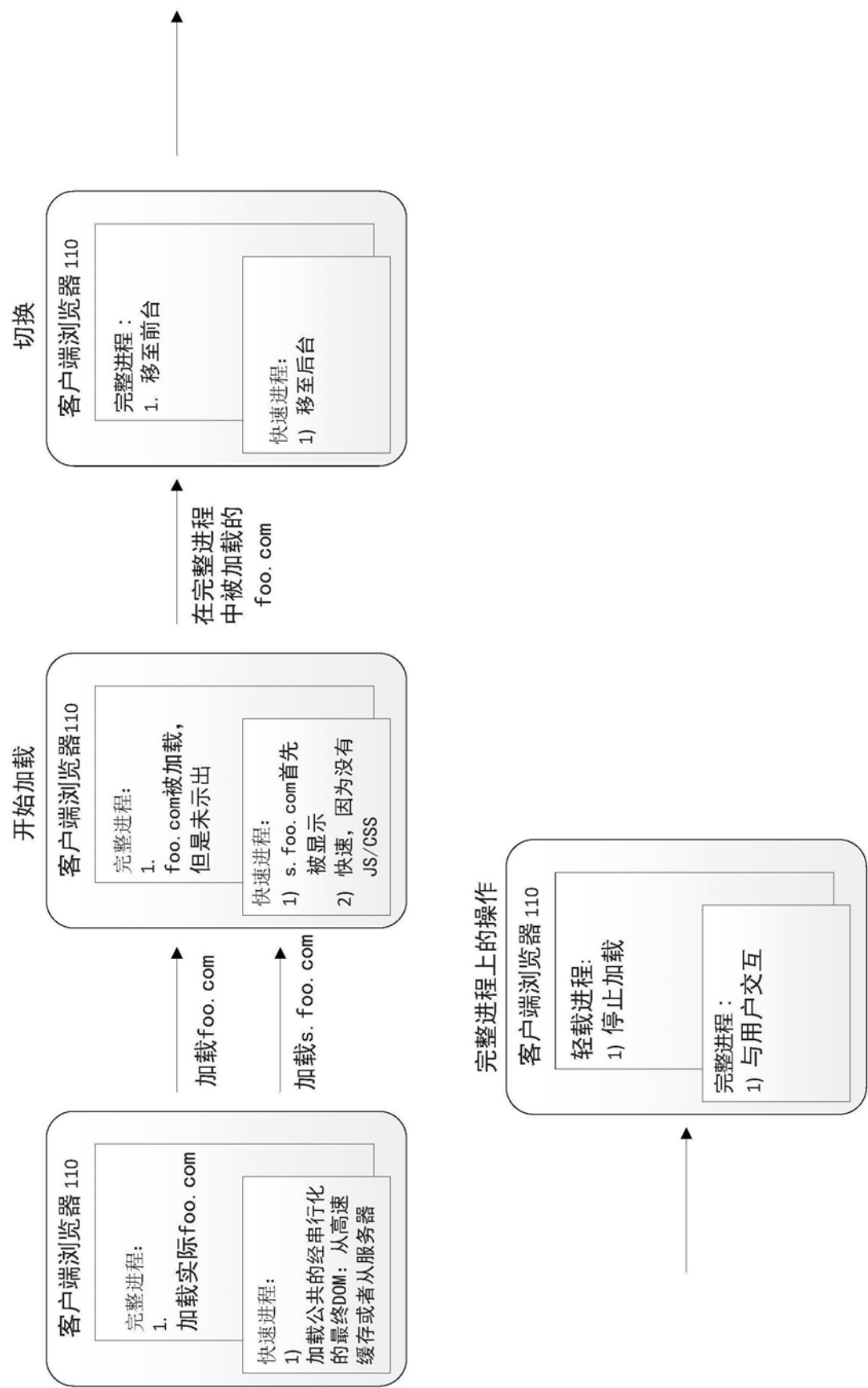


图5

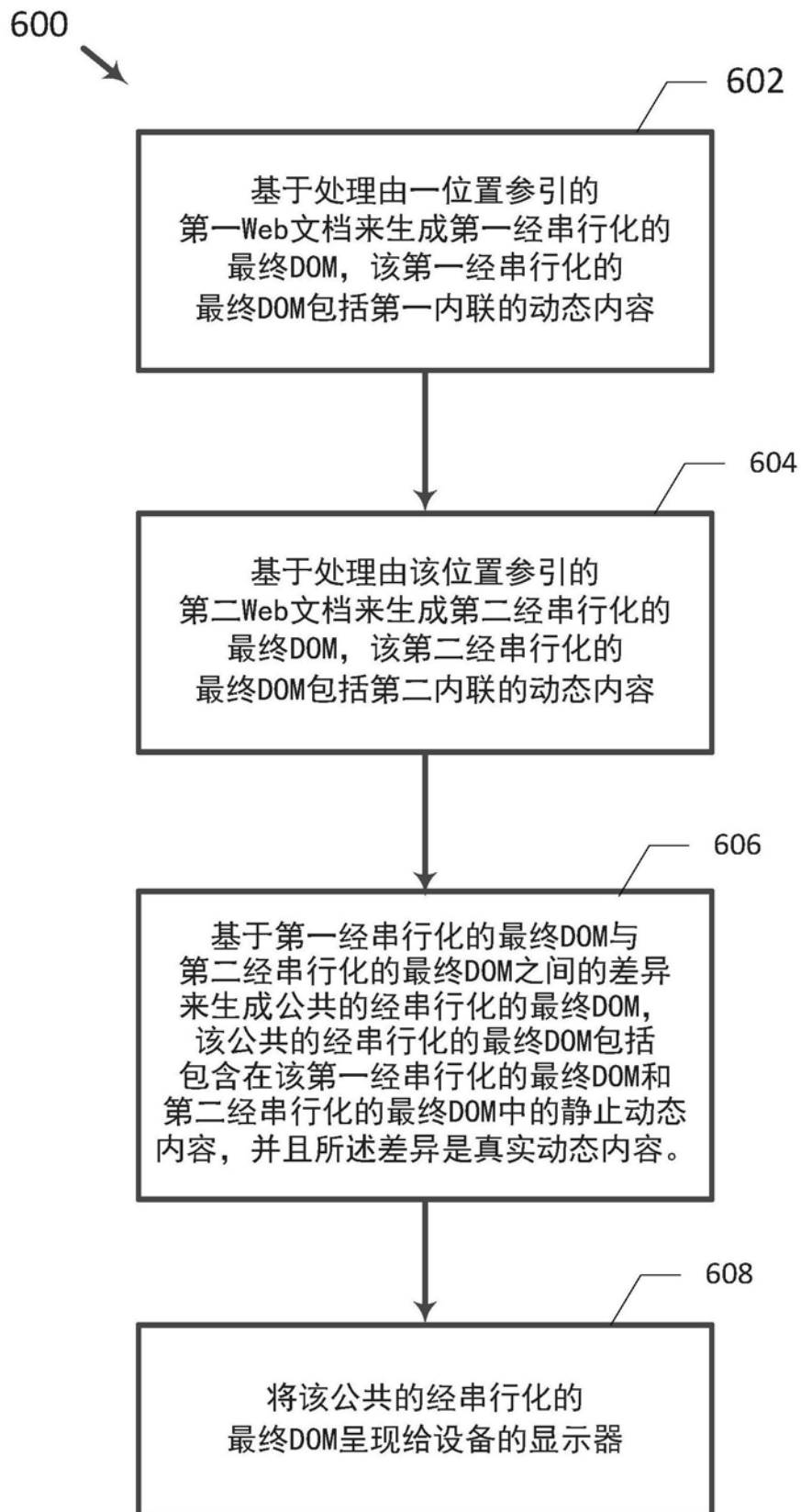


图6

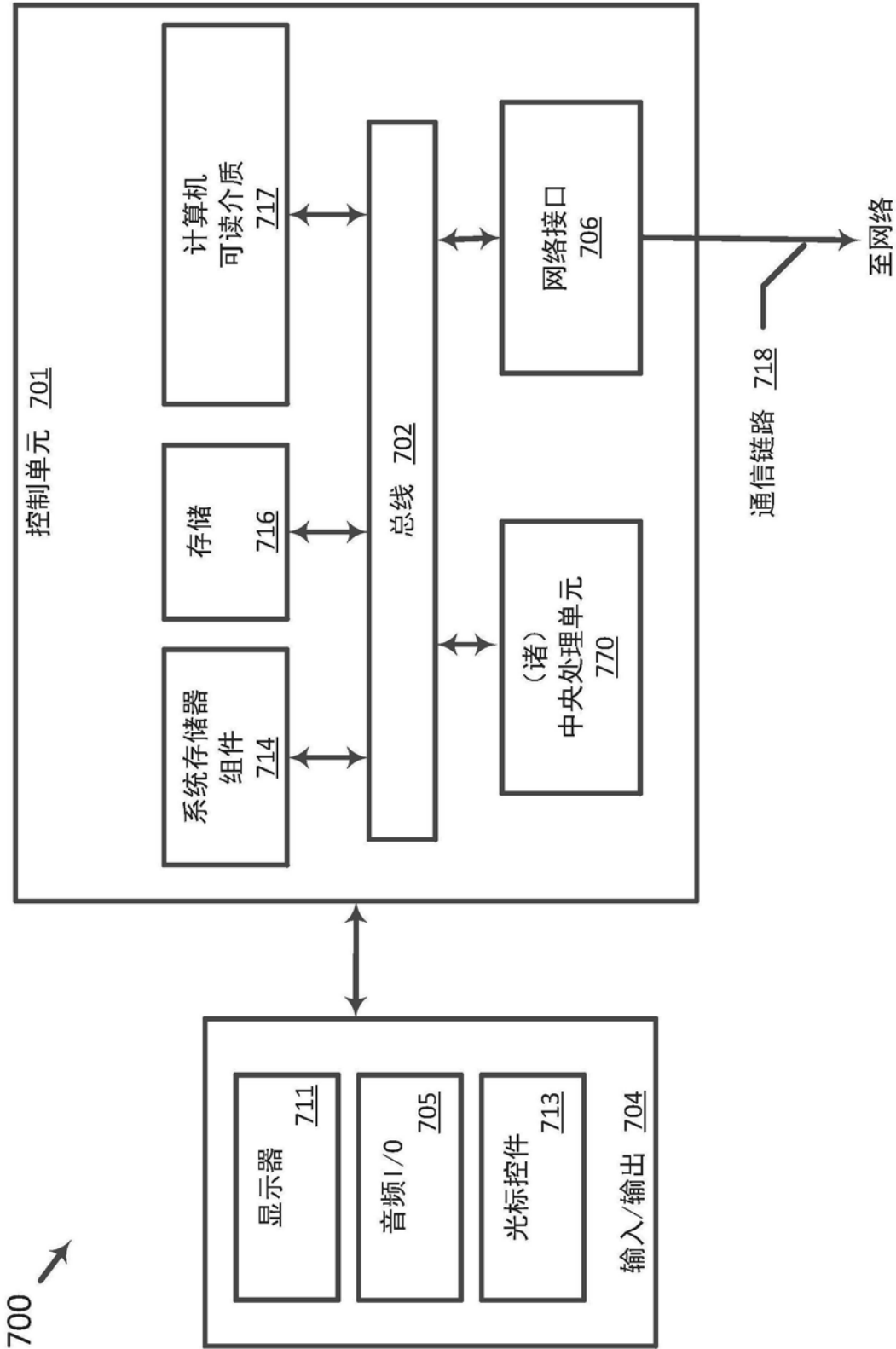


图7