



(12)发明专利

(10)授权公告号 CN 103502983 B

(45)授权公告日 2017.02.08

(21)申请号 201280020821.0

(72)发明人 L·西泽 G·C·卡希瓦尔 B·王

(22)申请日 2012.04.12

M·P·马翰 C·S·德希伦
W·罗蒂斯 M·维克拉姆

(65)同一申请的已公布的文献号

申请公布号 CN 103502983 A

(74)专利代理机构 上海专利商标事务所有限公司 31100

(43)申请公布日 2014.01.08

代理人 唐杰敏

(30)优先权数据

13/096,131 2011.04.28 US

(51)Int.Cl.

G06F 17/30(2006.01)

(85)PCT国际申请进入国家阶段日

2013.10.28

(56)对比文件

US 6377957 B1,2002.04.23,

(86)PCT国际申请的申请数据

US 6377957 B1,2002.04.23,

PCT/US2012/033344 2012.04.12

US 2009063500 A1,2009.03.05,

(87)PCT国际申请的公布数据

US 2003217169 A1,2003.11.20,

W02012/148692 EN 2012.11.01

审查员 胡斌

(73)专利权人 高通股份有限公司

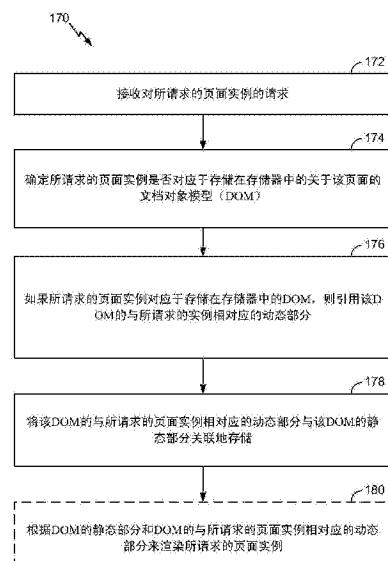
权利要求书5页 说明书22页 附图21页

(54)发明名称

利用基于DOM的同构来备忘缓存Web浏览计算

(57)摘要

用于加速浏览器对网页的渲染的方法和设备，其存储所渲染的页面的文档对象模型(DOM)树结构和计算，并且将正被渲染的页面的DOM树的诸部分进行比较以确定这些DOM树结构是否有部分与之匹配。如果要被渲染的网页的DOM树与存储在存储器中的DOM树相匹配，则可从存储器调用与匹配DOM树相关联的计算，从而避免了对执行计算以渲染页面的需要。树同构算法可被用于识别存储在存储器中的、与要被渲染的网页的DOM树相匹配的DOM树。重用渲染计算可显著减少为渲染网页所需要的时间和资源。基于DOM树同构来标识计算结果的可重用部分使浏览器即使在URL不匹配时也能够重用存储着的网页渲染计算。



1. 一种用于在计算设备上渲染页面的方法,包括:

在所述计算设备的处理器中接收显示具有第一通用资源定位符(URL)的网页的请求;

接收与所请求的网页相对应的软件代码;

解析所接收到的软件代码以生成文档对象模型(DOM)树;

确定所生成的DOM树的一部分是否与所存储的多个DOM树中的所有存储的特定DOM树的一个或多个部分同构,其中所述多个DOM树中的每个DOM树包括页面的不同实例的关系数据,并且所存储的特定DOM树的所述一个或多个部分中的每个部分涉及所述页面的所述不同实例所独有的一一个或多个动态存储DOM部分;

在确定了所生成的DOM树的一部分与所存储的特定DOM树的一个或多个部分同构时从存储器检索先前存储的用于处理与所存储的特定DOM树的同构部分相关联的HTML代码的HTML计算,其中用于处理HTML代码的所述先前存储的HTML计算包括解析、格式化、和渲染所请求的网页的具有不同于所述第一URL的第二URL的第二实例的基于web的内容中的一者或多者;

检索被确定为包括与所生成的DOM树的一部分同构的一个或多个部分的所有存储的特定DOM树;

使用检索到的HTML计算、检索到的所存储的特定DOM树的所述一个或多个部分、和检索到的所存储的特定DOM树的动态存储DOM部分来构成所请求的网页;以及

在所述计算设备的电子显示器上显示所构成的所请求的网页。

2. 如权利要求1所述的方法,其特征在于,

接收与所请求的网页相对应的软件代码包括接收与所请求的网页相对应的HTML代码;以及

解析所接收到的软件代码以生成文档对象模型(DOM)树包括解析所接收到的HTML代码以生成DOM树。

3. 如权利要求2所述的方法,其特征在于,确定所生成的DOM树的一部分是否与所存储的多个DOM树中的所有存储的特定DOM树的一个或多个部分同构包括通过使用树同构比较算法来将所生成的DOM树与所存储的特定DOM树的DOM树结构作比较。

4. 如权利要求2所述的方法,其特征在于,确定所生成的DOM树的一部分是否与所存储的多个DOM树中的所有存储的特定DOM树的一个或多个部分同构包括确定所生成的DOM树的所述一部分是否具有与所存储的特定DOM树的一个或多个部分相同的结构。

5. 如权利要求2所述的方法,其特征在于,确定所生成的DOM树的一部分是否与所存储的多个DOM树中的所有存储的特定DOM树的一个或多个部分同构包括确定所生成的DOM树的所述一部分是否与所存储的特定DOM树的一个或多个部分在结构上相似。

6. 如权利要求2所述的方法,其特征在于,进一步包括确定用于创建所存储的特定DOM树的计算结果的键值和输入值是否等于所生成的DOM树的键值和输入值,

其中从所述存储器检索所述先前存储的用于处理与所存储的特定DOM树的同构部分相关联的HTML计算是仅在确定了用于创建所存储的特定DOM树的计算结果的键值和输入值等于所生成的DOM树的键值和输入值时才被执行。

7. 如权利要求2所述的方法,其特征在于,从所述存储器检索所述先前存储的用于处理与所存储的特定DOM树的同构部分相关联的HTML计算包括访问存储所述DOM树的所述一部

分的计算结果的所述存储器,以使得所述计算结果基于与所述DOM树的诸部分相对应的DOM子树结构按照计算出的散列值来索引。

8. 如权利要求2所述的方法,其特征在于,进一步包括:

在确定所生成的DOM树与所存储的特定DOM树的所述一个或多个部分中的任何部分都不同构时完成关于所生成的DOM树的HTML代码计算;以及

在存储器中存储用所生成的DOM树来索引的所述HTML计算的结果。

9. 如权利要求8所述的方法,其特征在于,存储所生成的DOM树包括在键-值数据结构中存储所生成的DOM树的至少一部分,其中DOM树元素与相应的HTML计算结果相关联地存储在所述键-值数据结构中。

10. 如权利要求9所述的方法,其特征在于,所述键-值数据结构包括散列映射数据结构。

11. 如权利要求10所述的方法,其特征在于,从所述存储器检索先前存储的计算包括评估与所述存储着的DOM树中的和所生成的DOM树的所述部分同构的节点相关联的散列键。

12. 一种计算设备,包括:

存储器;

耦合至所述存储器的处理器;

用于接收显示具有第一通用资源定位符(URL)的网页的请求的装置;

用于接收与所请求的网页相对应的软件代码的装置;

用于解析所接收到的软件代码以生成文档对象模型(DOM)树的装置;

用于确定所生成的DOM树的一部分是否与所存储的多个DOM树中的所存储的特定DOM树的一个或多个部分同构的装置,其中所述多个DOM树中的每个DOM树包括页面的不同实例的关系数据,并且所存储的特定DOM树的所述一个或多个部分中的每个部分涉及所述页面的所述不同实例所独有的一一个或多个动态存储DOM部分;

用于在确定了所生成的DOM树的一部分与所存储的特定DOM树的一个或多个部分同构时从存储器检索先前存储的用于处理与所存储的特定DOM树的同构部分相关联的HTML代码的HTML计算的装置,其中用于处理HTML代码的所述先前存储的HTML计算包括解析、格式化、和渲染所请求的网页的具有不同于所述第一URL的第二URL的第二实例的基于web的内容中的一者或者者;

用于检索被确定为包括与所生成的DOM树的一部分同构的一个或多个部分的所有存储的特定DOM树的装置;

用于使用检索到的HTML计算、检索到的所存储的特定DOM树的所述一个或多个部分、和检索到的所存储的特定DOM树的动态存储DOM部分来构成所请求的网页的装置;以及

用于在电子显示器上显示所构成的所请求的网页的装置。

13. 如权利要求12所述的计算设备,其特征在于:

所述用于接收与所请求的网页相对应的软件代码的装置包括用于接收与所请求的网页相对应的HTML代码的装置;以及

所述用于解析所接收到的软件代码以生成文档对象模型(DOM)树的装置包括用于解析所接收到的HTML代码以生成DOM树的装置。

14. 如权利要求13所述的计算设备,其特征在于,用于确定所生成的DOM树的一部分是

否与所存储的多个DOM树中的所存储的特定DOM树的一个或多个部分同构的装置包括用于通过使用树同构比较算法来将所生成的DOM树与所存储的特定DOM树的DOM树结构作比较的装置。

15. 如权利要求13所述的计算设备,其特征在于,用于确定所生成的DOM树的一部分是否与所存储的多个DOM树中的所存储的特定DOM树的一个或多个部分同构的装置包括用于确定所生成的DOM树的所述一部分是否具有与所存储的特定DOM树的一个或多个部分相同的结构的装置。

16. 如权利要求13所述的计算设备,其特征在于,用于确定所生成的DOM树的一部分是否与所存储的多个DOM树中的所存储的特定DOM树的一个或多个部分同构的装置包括用于确定所生成的DOM树的所述一部分是否与所存储的特定DOM树的一个或多个部分在结构上相似的装置。

17. 如权利要求13所述的计算设备,其特征在于,进一步包括用于确定用于创建所存储的特定DOM树的计算结果的键值和输入值是否等于所生成的DOM树的键值和输入值的装置,

其中用于从所述存储器检索所述先前存储的用于处理与所存储的特定DOM树的同构部分相关联的HTML代码的HTML计算的装置包括用于仅在确定了用于创建所存储的特定DOM树的计算结果的所述键值和输入值等于所生成的DOM树的键值和输入值时才检索与所存储的特定DOM树的同构部分相关联的HTML计算的装置。

18. 如权利要求13所述的计算设备,其特征在于,用于从所述存储器检索所述先前存储的用于处理与所存储的特定DOM树的同构部分相关联的HTML代码的HTML计算的装置包括用于访问存储所述DOM树的所述一部分的所述计算结果的所述存储器以使得所述计算结果基于与所述DOM树的诸部分相对应的DOM子树结构按照计算出的散列值来索引的装置。

19. 如权利要求13所述的计算设备,其特征在于,进一步包括:

用于在确定了所生成的DOM树与所存储的特定DOM树的所述一个或多个部分中的任何部分都不同构时完成关于所生成的DOM树的HTML代码计算的装置;以及

用于在存储器中存储用所生成的DOM树来索引的所述HTML计算的结果的装置。

20. 如权利要求19所述的计算设备,其特征在于,用于存储所生成的DOM树的装置包括用于在键-值数据结构中存储所生成的DOM树的至少一部分的装置,其中DOM树元素与相应的HTML计算结果相关联地存储在所述键-值数据结构中。

21. 如权利要求20所述的计算设备,其特征在于,用于在键-值数据结构中存储所生成的DOM树的至少一部分的装置包括用于在散列映射数据结构中存储所生成的DOM树的至少一部分的装置。

22. 如权利要求21所述的计算设备,其特征在于,用于从所述存储器检索先前存储的计算的装置包括用于评估与所述存储着的DOM树中的和所生成的DOM树的所述部分同构的节点相关联的散列键的装置。

23. 一种计算设备,包括:

存储器;以及

耦合至所述存储器的处理器,其中所述处理器配置有处理器可执行指令以执行包括以下操作的操作:

接收显示具有第一通用资源定位符(URL)的网页的请求;

接收与所请求的网页相对应的软件代码；

解析所接收到的软件代码以生成文档对象模型(DOM)树；

确定所生成的DOM树的一部分是否与所存储的多个DOM树中的所有存储的特定DOM树的一个或多个部分同构，其中所述多个DOM树中的每个DOM树包括页面的不同实例的关系数据，并且所存储的特定DOM树的所述一个或多个部分中的每个部分涉及所述页面的所述不同实例所独有的一一个或多个动态存储DOM部分；

在确定了所生成的DOM树的一部分与所存储的特定DOM树的一个或多个部分同构时从存储器检索先前存储的用于处理与所存储的特定DOM树的同构部分相关联的HTML代码的HTML计算，其中用于处理HTML代码的所述先前存储的HTML计算包括解析、格式化、和渲染所请求的网页的具有不同于所述第一URL的第二URL的第二实例的基于web的内容中的一者或多者；

检索被确定为包括与所生成的DOM树的一部分同构的一个或多个部分的所有存储的特定DOM树；

使用检索到的HTML计算、检索到的所存储的特定DOM树的所述一个或多个部分、和检索到的所存储的特定DOM树的动态存储DOM部分来构成所请求的网页；以及

显示所构成的所请求的网页。

24. 如权利要求23所述的计算设备，其特征在于，所述处理器配置有处理器可执行指令以执行操作，以使得：

接收与所请求的网页相对应的软件代码包括接收与所请求的网页相对应的HTML代码；以及

解析所接收到的软件代码以生成文档对象模型(DOM)树包括解析所接收到的HTML代码以生成DOM树。

25. 如权利要求24所述的计算设备，其特征在于，所述处理器配置有处理器可执行指令，以使得确定所生成的DOM树的一部分是否与所存储的多个DOM树中的所有存储的特定DOM树的一个或多个部分同构包括通过使用树同构比较算法来将所生成的DOM树与所存储的特定DOM树的DOM树结构作比较。

26. 如权利要求25所述的计算设备，其特征在于，所述处理器配置有处理器可执行指令，以使得确定所生成的DOM树的一部分是否与所存储的多个DOM树中的所有存储的特定DOM树的一个或多个部分同构包括确定所生成的DOM树的所述一部分是否具有与所存储的特定DOM树的一个或多个部分相同的结构。

27. 如权利要求25所述的计算设备，其特征在于，所述处理器配置有处理器可执行指令，以使得确定所生成的DOM树的一部分是否与所存储的多个DOM树中的所有存储的特定DOM树的一个或多个部分同构包括确定所生成的DOM树的所述一部分是否与所存储的特定DOM树的一个或多个部分在结构上相似。

28. 如权利要求25所述的计算设备，其特征在于，所述处理器配置有处理器可执行指令以执行进一步包括以下操作的操作：确定用于创建所存储的特定DOM树的计算结果的键值和输入值是否等于所生成的DOM树的键值和输入值，

其中所述处理器配置有处理器可执行指令，以使得从所述存储器检索所述先前存储的用于处理与所存储的特定DOM树的同构部分相关联的HTML代码的HTML计算是仅在确定了用

于创建所存储的特定DOM树的计算结果的所述键值和输入值等于所生成的DOM树的键值和输入值时才被执行的。

29. 如权利要求25所述的计算设备,其特征在于,所述处理器配置有处理器可执行指令,以使得从所述存储器检索所述先前存储的用于处理与所存储的特定DOM树的同构部分相关联的HTML代码的HTML计算包括访问存储所述DOM树的所述一部分的所述计算结果的所述存储器,以使得所述计算结果基于与所述DOM树的诸部分相对应的DOM子树结构按照计算出的散列值来索引。

30. 如权利要求25所述的计算设备,其特征在于,所述处理器配置有处理器可执行指令以执行进一步包括以下操作的操作:

在确定了所生成的DOM树与所存储的特定DOM树的所述一个或多个部分中的任何部分都不同构时完成关于所生成的DOM树的HTML代码计算;以及

在存储器中存储用所生成的DOM树来索引的所述HTML计算的结果。

31. 如权利要求30所述的计算设备,其特征在于,所述处理器配置有处理器可执行指令,以使得存储所生成的DOM树包括在键-值数据结构中存储所生成的DOM树的至少一部分,其中DOM树元素与相应的HTML计算结果相关联地存储在所述键-值数据结构中。

32. 如权利要求31所述的计算设备,其特征在于,所述处理器配置有处理器可执行指令,以使得所述键-值数据结构包括散列映射数据结构。

33. 如权利要求32所述的计算设备,其特征在于,所述处理器配置有处理器可执行指令,以使得从所述存储器检索先前存储的计算包括评估与所述存储着的DOM树中的和所生成的DOM树的所述部分同构的节点相关联的散列键。

利用基于DOM的同构来备忘缓存Web浏览计算

[0001] 相关申请

[0002] 本申请是2010年3月24日提交的题为“Apparatus and Methods for Rendering a Page(用于渲染页面的装置和方法)”的美国专利申请No.12/730,967的部分继续申请,该美国专利申请要求2009年10月6日提交的题为“Apparatus and Methods of Rendering a Page(渲染页面的装置和方法)”的美国临时申请No.61/248,909和2009年3月31日提交的题为“Method and Apparatus for Facilitating Efficient Local Caching(用于促成高效本地高速缓存的方法和装置)”的美国临时申请No.61/165,238的优先权权益,所有这些申请均被转让给本申请受让人并且由此通过援引全部纳入于此。

发明领域

[0003] 以下描述一般涉及计算设备通信,并且尤其涉及用于渲染页面的装置和方法。

背景技术

[0004] 在计算设备上渲染诸如网页之类的页面会受到该页面的文档对象模型(DOM)树的影响。DOM树定义呈子-父关系的一个或多个节点,包括各自相应的DOM节点的性质和/或属性及其样式属性。当页面被启动时,其.html文件被从高速缓存读取、解析,DOM树被构造,并且随后嵌入式或外部级联样式表(css)被应用。每当页面被加载时都要执行这些操作,这花费了大量时间,由此导致用户体验不尽人意。

[0005] 不仅如此,在资源受约束的环境中,诸如对于移动计算设备(例如,与非移动计算设备相比具有相对有限的处理能力、存储器和/或通信吞吐量的移动电话、个人数字助理(PDA)或其他相对较小的便携式设备)而言,加剧了此时间量。因此,对于移动计算设备而言,用户体验甚至更差。

[0006] 因此,希望改善页面的渲染。

[0007] 概述

[0008] 以下给出一个或多个方面的简要概述以提供对这些方面的基本理解。此概述不是所有构想到的方面的详尽综览,并且既非旨在标识出所有方面的关键性或决定性要素亦非试图界定任何或所有方面的范围。其唯一的目的是要以简化形式给出一个或多个方面的一些概念以作为稍后给出的更加详细的描述之序。

[0009] 一个方面涉及用于在计算机设备上渲染页面的方法。该方法可包括在计算机设备处接收对所请求的页面实例的请求。另外,该方法可包括确定所请求的页面实例是否对应于存储在存储器中的关于该页面的文档对象模型(DOM)。存储在存储器中的DOM对应于存储着的页面实例并且包括对于存储着的页面实例和所请求的页面实例两者而言相同的静态部分。另外,该方法还可包括如果所请求的页面实例对应于存储在存储器中的DOM,则检索该DOM的与所请求的实例相对应的动态部分。该动态部分是所请求的页面实例所独有的。另外,该方法可包括将该DOM的与所请求的页面实例相对应的动态部分与该DOM的静态部分关联地存储。

[0010] 另一方面涉及被配置成在计算机设备上渲染页面的至少一个处理器。该处理器可包括用于在计算机设备处接收对所请求的页面实例的请求的第一模块。该处理器还可包括用于确定所请求的页面实例是否对应于存储在存储器中的关于该页面的文档对象模型(DOM)的第二模块。存储在存储器中的DOM对应于存储着的页面实例并且包括对于存储着的页面实例和所请求的页面实例两者而言相同的静态部分。另外，该处理器可包括用于如果所请求的页面实例对应于存储在存储器中的DOM则检索该DOM的与所请求的实例相对应的动态部分的第三模块。该动态部分是所请求的页面实例所独有的。该处理器还可包括用于将该DOM的与所请求的页面实例相对应的动态部分与该DOM的静态部分关联地存储的第四模块。

[0011] 又一方面涉及用于在计算机设备上渲染页面的计算机程序产品。该计算机程序产品可包括计算机可读介质，该计算机可读介质至少包括用于使计算机在计算机设备处接收对所请求的页面实例的请求的指令。该计算机可读介质还可包括用于使计算机确定所请求的页面实例是否对应于存储在存储器中的关于该页面的文档对象模型(DOM)的至少一条指令。存储在存储器中的DOM对应于存储着的页面实例并且包括对于存储着的页面实例和所请求的页面实例两者而言相同的静态部分。该计算机可读介质可进一步包括用于如果所请求的页面实例对应于存储在存储器中的DOM则使计算机检索该DOM的与所请求的实例相对应的动态部分的至少一条指令。该动态部分是所请求的页面实例所独有的。另外，该计算机可读介质还可包括用于使计算机将该DOM的与所请求的页面实例相对应的动态部分与该DOM的静态部分关联地存储的至少一条指令。

[0012] 另一方面涉及一种设备。该设备可包括用于在计算机设备处接收对所请求的页面实例的请求的装置。另外，该设备可包括用于确定所请求的页面实例是否对应于存储在存储器中的关于该页面的文档对象模型(DOM)的装置。存储在存储器中的DOM对应于存储着的页面实例并且包括对于存储着的页面实例和所请求的页面实例两者而言相同的静态部分。该设备还可包括用于如果所请求的页面实例对应于存储在存储器中的DOM则检索该DOM的与所请求的实例相对应的动态部分的装置。该动态部分是所请求的页面实例所独有的。该设备可进一步包括用于将该DOM的与所请求的页面实例相对应的动态部分与该DOM的静态部分关联地存储的装置。

[0013] 又一方面涉及用于在计算机设备上渲染页面的装置。该装置可包括门户组件，该门户组件被配置成接收对所请求的页面实例的请求。该装置还可包括web组件，该web组件被配置成确定所请求的页面实例是否对应于存储在存储器中的关于该页面的文档对象模型(DOM)。存储在存储器中的DOM对应于存储着的页面实例并且包括对于存储着的页面实例和所请求的页面实例两者而言相同的静态部分。另外，该装置可包括被配置成如果所请求的页面实例对应于存储在存储器中的DOM则检索该DOM的与所请求的实例相对应的动态部分的所述web组件。该动态部分是所请求的页面实例所独有的。该装置可进一步包括被配置成将该DOM的与所请求的页面实例相对应的动态部分与该DOM的静态部分关联地存储的所述web组件。

[0014] 所描述的诸方面涉及用于在计算机设备上渲染页面的方法、装置或计算机程序产品，包括用于执行以下操作的动作、装置或指令：在计算机设备处接收对所请求的页面实例的请求；确定所请求的页面实例是否对应于存储在存储器中的关于该页面的文档对象模型

(DOM),其中存储在存储器中的DOM对应于存储着的页面实例并且包括对于存储着的页面实例和所请求的页面实例两者而言相同的静态部分;如果所请求的页面实例对应于存储在存储器中的DOM,则检索该DOM的与所请求的实例相对应的动态部分,其中该动态部分是所请求的页面实例所独有的;以及将该DOM的与所请求的页面实例相对应的动态部分与该DOM的静态部分关联地存储;并且可任选地包括用于根据DOM的静态部分和该DOM的与所请求的页面实例相对应的动态部分来渲染所请求的页面实例的动作、装置或指令。

[0015] 其他方面涉及通过将存储着的页面渲染计算与DOM树结构相关联并藉由识别存储着的与要被渲染的页面的DOM树的至少一部分相匹配的DOM树模式来标识可重用的计算的方式,来重用web浏览器的计算。这些方面通过除了传统的对HTML数据的高速缓存之外还使得能够重用HTML处理计算,以使得即使在所保存的计算的页面URL与要被渲染的当前页面不同时也可在渲染未来页面时重用这些HTLM处理计算的方式,来改善浏览器的性能。在此类方面,用于在计算设备上渲染页面的方法可包括在计算设备的处理器中接收显示网页的请求;接收与所请求的网页相对应的HTML代码;解析接收到的HTML代码以生成文档对象模型(DOM)树;确定所生成的DOM树的一部分是否与存储在存储器中的DOM树的一个或多个部分同构;当确定所生成的DOM树的一部分与存储在存储器中的DOM树的一个或多个部分同构时从存储器检索先前存储着的与DOM树的同构部分相关联的计算;使用检索到的计算来构成所请求的网页;以及在计算设备的电子显示器上显示所请求的网页。在此类方法中,确定所生成的DOM树的一部分是否与存储在存储器中的DOM树的一个或多个部分同构可包括通过使用树同构比较算法来将所生成的DOM树与存储在存储器中的DOM树结构作比较;确定所生成的DOM树的所述部分是否具有与存储在存储器中的DOM树的一个或多个部分相同的结构,或者确定所生成的DOM树的所述部分是否在结构上类似于存储在存储器中的DOM树的一个或多个部分。在一方面,从存储器检索先前存储着的与DOM树的同构部分相关联的计算可包括访问存储DOM树的这些部分的计算结果的高速缓存机构,以使得所述计算结果基于与DOM树的诸部分相对应的DOM子树结构按照计算出的散列值来索引。

[0016] 在一方面,该方法可进一步包括确定用于创建存储器中的DOM树的计算结果的键值和输入值是否等于所生成的DOM树的键值和输入值,在这种情形中,可仅在确定了用于创建存储器中的DOM树的计算结果的键值和输入值等于所生成的DOM树的键值和输入值时才执行从存储器检索先前存储着的与DOM树的同构部分相关联的计算的操作。另外,该方法可进一步包括在确定了所生成的DOM树与存储在存储器中的DOM树的这一个或多个部分中的任何部分均不同构时完成对所生成的DOM树的HTML代码计算,并且在存储器中存储用所生成的DOM树来索引的HTML计算结果。在此类方面,存储所生成的DOM树可包括在键-值数据结构中存储所生成的DOM树的至少一部分,其中DOM树元素与相应的HTML计算结果相关联地存储,在这种情形中,结构化的高速缓存可使用散列映射数据结构来实现,并且键-值数据结构可包括散列映射数据结构。在此类方面,从存储器检索先前存储着的计算可包括评估与存储着的DOM树中的和所生成的DOM树的所述部分同构的那些节点相关联的散列键。

[0017] 其他方面涉及包括处理器的计算设备,该处理器被配置有用于执行上述方法的操作的处理器可执行指令。另一些方面涉及包括用于达成上述方法的功能的装置的计算设备。又一些方面涉及其上存储有处理器可执行指令的非瞬态计算机可读存储介质,这些指令被配置成使计算设备的处理器执行上述方法的诸操作。

[0018] 为能达成前述及相关目的,这一个或多个方面包括在下文中充分描述并在所附权利要求中特别指出的特征。以下描述和附图详细阐述了这一个或多个方面的某些解说性特征。但是,这些特征仅仅是指示了可采用各种方面的原理的各种方式中的若干种,并且本描述旨在涵盖所有此类方面及其等效方案。

附图说明

[0019] 纳入本文并构成本说明书的一部分的附图解说了本发明的示例性方面。这些附图与以上给出的一般描述以及下文给出的详细描述一道用于解释本发明的特征而不是限定所公开的诸方面。

[0020] 图1是用于渲染页面的系统的一方面的示意图。

[0021] 图2是共享DOM的一方面的示意图。

[0022] 图3是图1的计算机设备的一方面的示意图。

[0023] 图4是渲染页面的方法的一方面的流程图。

[0024] 图5是渲染第一页面实例的一方面的流程图。

[0025] 图6是渲染第二页面实例的一方面的流程图。

[0026] 图7是根据一方面的用于促成高效率本地高速缓存的示例性方法。

[0027] 图8是根据一方面的用于在高速缓存中进行存储之前通过处理指令来促成内容修改的示例性方法。

[0028] 图9是根据一方面的用于在高速缓存中促成结构化存储的示例性方法。

[0029] 图10是根据一方面的示例主页面的示意图。

[0030] 图11是根据一方面的示例目录页面的示意图。

[0031] 图12是根据一方面的商品详情页面的示例的示意图。

[0032] 图13是根据一方面的商品详情页面的另一示例的示意图。

[0033] 图14是用于渲染页面的装置的一方面的示意图。

[0034] 图15是通过浏览器解析所请求的网页的HTML代码所生成的样本DOM树的示意图。

[0035] 图16是具有为每个节点存储的计算值的样本DOM树的示意图。

[0036] 图17是用于在浏览器的首次执行时和/或具有未标识出的DOM结构的网页首次出现时创建、处理和存储DOM树的方法的一方面的流程图。

[0037] 图18是用于在浏览器的后续执行时创建、处理、存储、和/或检索DOM树的诸部分的方法的一方面的流程图。

[0038] 图19是用于在浏览器的后续执行时创建、处理、存储、和/或检索DOM树的诸部分的方法的另一方面的流程图。

[0039] 图20是适用于各种实施例的示例接收机设备的组件框图。

[0040] 图21是适用于各种实施例的示例服务器的组件框图。

具体实施方式

[0041] 现在参照附图描述各个方面。在以下描述中,出于解释目的阐述了众多具体细节以提供对一个或更多个方面的透彻理解。但是显然的是,没有这些具体细节也可实践此(诸)方面。

[0042] 所描述的诸方面涉及获取和渲染页面,诸如具有web资源和数据项的网页。另外,所描述的诸方面提供了在网页之间转换的虚拟效果,例如,为用户提供链接以在网页之间来回移动。网页是根据相应的文档对象模型(DOM)来渲染的,该DOM定义了定义页面的各个组件的组件、相对结构、以及行为。相同页面的不同实例可共享DOM的一些共有部分,而同时具有将页面的一个实例与下一实例区分开来的其他独有部分。

[0043] 根据一个或多个方面,用于渲染页面的装置和方法提供了可操作用于创建可由页面的两个或更多个实例使用的共享DOM的web引擎或其他组件。此共享DOM包括页面的不同实例共有的静态DOM部分,以及该页面的相应一个或多个实例所独有的一一个或多个动态DOM部分。由此,所描述的方面通过在渲染页面的新实例时重用与存储着的或高速缓存着的DOM(该DOM可基于相同页面的先前处理过的不同实例)相对应的静态DOM部分来改善页面渲染的效率,由此避免了不得不为页面的每个实例创建完全新的DOM。

[0044] 例如,这些方面可参照诸如由移动电话执行的移动购物应用之类的使用情形来描述。然而,应当理解,此示例使用情形仅是为解说目的,并且存在许多其他使用情形。在移动购物应用示例中,与移动购物应用相对应的网页可包括不同类型页面的树或阶层,诸如具有目录列表的主页面、具有商品列表和/或其他目录列表的一个或多个目录页面、以及还可链接至其他商品详情页面的一个或多个商品详情页面、和/或诸如定价或购买页面之类的附加详情页面。相同页面的不同实例(诸如第一游戏的商品详情页面和第二游戏的商品详情页面)可共享类似的结构并可能共享一些类似的数据,但是它们还可各自具有独有的数据(诸如每个相应游戏的名称)。由此,通常,每个商品详情页面具有不同的相应的DOM。然而,在本申请的诸方面,商品详情页面取而代之具有共享DOM,该共享DOM包括与该页面的不同实例之间共有的DOM元素相对应的静态DOM部分、以及将独有数据项与该页面的相应实例相关联的一个或多个动态DOM部分。例如,在这种情形中,静态DOM部分可包括第一游戏商品详情页面和第二游戏商品详情页面的相同布局,而各动态数据项(诸如第一游戏名称和第二游戏名称)则由动态DOM部分来表示。

[0045] 换言之,在一方面,当页面的独特实例(例如,“商品详情”页面)从未在计算机设备上被取回或高速缓存、并且该页面的该实例被请求时,所描述的诸方面提供导致取回页面数据(例如,包括超文本标记语言(html)或可扩展html(xhtml))、级联样式表(css)、和JavaScript(js)并且从xhtml和css创建文档对象模型(DOM)的静态部分的行为。DOM的此静态部分被存储在高速缓存中并且可被重用。对于该页面,随后执行js,从而导致一个或多个数据请求(例如,XMLHttp请求(XHR))。初始的js和对一个或多个响应中的每一个响应的处置导致为页面的该实例创建DOM的一个或多个动态部分。随后,当该页面的第二实例被请求时,静态DOM部分可被重用,并且为该请求的与页面的第二实例相对应的新查询字符串执行js,从而导致一个或多个新数据请求以及为该页面的新实例创建DOM的一个或多个动态部分。作为结果,共享DOM包括至少一个静态部分和一个或多个动态部分,从而使DOM能够定义供在渲染该页面的不同实例时使用的共享DOM。

[0046] 因此,在一方面,对DOM的初始或静态状态或部分的重用减少了页面转换时间。在一个或多个方面,状态或DOM部分重用可通过维护分开的(或者替换地集成的)、与DOM的静态和动态状态或部分相对应的数据结构来达成。

[0047] 另外,在一个或多个方面,一个或多个优点可包括:由于仅复制属性表,因而没有

存储器开销或者存储器开销较小;由于仅发生一次对HTML的解析和对CSS规则的应用,因而页面加载很快;由于不需要为相同页面的不同实例创建不同的DOM,因而与其他实现相比具有更干净和复杂性更低的代码;由于鉴于静态DOM部分不变,渲染可集中在更新各动态DOM部分上,因而不需要对DOM树的额外走查;并且动态属性是在页面加载之际按需要来创建的,且在导航回到该页面的实例之际在对DOM的现有走查上被移除。

[0048] 可任选地,在一些方面,DOM和相应的页面内容可被高速缓存,并且高速缓存可包括推导式高速缓存或结构化高速缓存。推导式高速缓存可存储从接收到的内容推导的内容。例如,在一个方面,一旦接收到原始内容(诸如但不限于xhtml、级联样式表或javascript),计算机设备就可进一步处理该原始内容以生成推导式内容。例如,在一方面,应当领会,推导式内容可包括DOM的静态部分,例如,DOM的静态部分可从接收到的xhtml和css创建和/或推导。可由于为生成推导式内容所花费的资源相对昂贵而确定将推导式内容例如存储在推导式高速缓存中。在一个方面,对推导式内容的这种高速缓存可通过使用此预先准备的推导式内容(诸如如果该内容被频繁地使用)来辅助高效率地促成对经高速缓存内容的后续使用。

[0049] 推导式内容的另一示例可包括计算机设备取回在从高速缓存检索时要被运行的程序。计算机设备可编译所取回的程序,而不是每当计算机设备从高速缓存检索该程序时编译所取回的程序。经编译的程序(或即推导式内容)可存储在推导式高速缓存中准备好供后续使用。另外,推导式内容可与接收到的源相关联。

[0050] 在另一方面,计算机设备高速缓存还可包括结构化的高速缓存。结构化的高速缓存允许以如下方式来存储内容:检索程序或类似物可选择性地选取高速缓存着的内容的有用部分供检索。例如,在没有本发明的装置和方法的情况下,高速缓存着的内容可能看起来是语义不透明的内容,由此阻碍了对高速缓存着的内容的诸部分的选择性检索。在一个方面,结构化高速缓存可按如下方式来组织:允许检索存储在结构化高速缓存中的内容项的选定的已知部分而不需要检索整个内容项。

[0051] 例如,假定具有众所周知结构的内容(诸如java script对象表示法(JSON)内容)被存储在结构化高速缓存中。进一步假定程序或类似物仅可使用该JSON内容的一小部分,诸如与制造商、标题、或类似物相关联的内容。结构化高速缓存允许仅选择性地移除对于该程序有用的内容部分,由此减少可能本来要求的任何中间处理。

[0052] 在另一示例中,内容提供商可用向计算机设备传送的内容来引入由发源服务器生成的入口点。由此,当具有由发源服务器生成的入口点的内容被存储在结构化高速缓存中时,可促成对由这些入口点定义的部分的后续移除。

[0053] 可任选地,在一些方面,所描述的装置和方法可进一步包括可在接收到对页面的请求之前执行以取回和处理该页面的预取操作。

[0054] 可任选地,在一些方面,所描述的装置和方法可进一步包括可在接收到对页面的请求之前执行以将网页解析成DOM并且高速缓存相应的DOM的预推导操作。

[0055] 可任选地,在一些方面,所描述的装置和方法可进一步包括预取和预推导操作两者。

[0056] 预取操作和/或预推导操作可进一步增加渲染页面实例的速度,由此改善从一个页面转换至另一个页面时的用户体验。

[0057] 在另一方面,用于加速浏览器对网页的渲染的方法和设备存储所渲染的页面的文档对象模型(DOM)树结构和计算,并且将正在渲染的页面的DOM树的诸部分进行比较以该DOM树结构是否有部分与之匹配。如果要被渲染的网页的DOM树与存储在存储器中的DOM树相匹配,则可从存储器重调用与匹配DOM树相关联的计算,从而避免了对执行计算以渲染页面的需要。树同构算法可被用于识别存储在存储器中的、与要被渲染的网页的DOM树相匹配的DOM树。如以上所讨论的,重用渲染计算可显著减少为渲染网页所需要的时间和资源。基于DOM树同构来标识计算结果的可重用部分使得即使在URL不匹配时浏览器也能够重用存储着的网页渲染计算,由此增加了可由浏览器重用的渲染计算的量。

[0058] 因此,本发明的诸方面可通过重用DOM的静态部分来高效率地渲染页面的不同实例,由此节省处理资源或通信资源或者两者。

[0059] 如本申请中所使用的,术语“组件”、“模块”、“系统”及类似术语旨在包括计算机相关实体,诸如但并不限于硬件、固件、硬件与软件的组合、软件、或执行中的软件。例如,组件可以是但不限于在处理器上运行的进程、处理器、对象、可执行件、执行的线程、程序、和/或计算机。作为解说,在计算设备上运行的应用和该计算设备两者皆可以是组件。一个或多个组件可驻留在进程和/或执行的线程内,且组件可以本地化在一台计算机上和/或分布在两台或更多台计算机之间。此外,这些组件能从其上存储着各种数据结构的各种计算机可读介质来执行。这些组件可藉由本地和/或远程进程来通信,诸如根据具有一个或多个数据分组的信号来通信,这样的数据分组诸如是来自藉由该信号与本地系统、分布式系统中另一组件交互的、和/或跨诸如因特网之类的网络与其他系统交互的一个组件的数据。

[0060] 另外,本文结合终端来描述各个方面,终端可以是有线终端或无线终端。终端也可被称为系统、设备、订户单元、订户站、移动站、移动台、移动设备、远程站、远程终端、接入终端、用户终端、终端、通信设备、用户代理、用户设备、或用户装备(UE)。无线终端可以是蜂窝电话、卫星电话、无绳电话、会话发起协议(SIP)电话、无线本地环路(WLL)站、个人数字助理(PDA)、具有无线连接能力的手持式设备、计算设备、或连接到无线调制解调器的其他处理设备。此外,本文结合基站来描述各个方面。基站可用于与无线终端进行通信,且也可被称为接入点、B节点、或其它某个术语。

[0061] 另外,措词“示例性的”在本文中被用于表示用作示例、实例或解说。本文中描述为“示例性”的任何方面或设计不必被解释为优于或胜过其他方面或设计。相反,词语“示例性”的使用旨在以具体方式给出概念。此外,术语“或”旨在表示包含性“或”而非排他性“或”。即,除非另外指明或从上下文能清楚地看出,否则短语“X采用A或B”旨在表示任何自然的可兼排列。即,短语“X采用A或B”藉由以下实例中任何实例得到满足:X采用A;X采用B;或X采用A和B两者。另外,本申请和所附权利要求书中所用的冠词“一”和“某”一般应当被理解成表示“一个或多个”,除非另外声明或者可从上下文中清楚看出是指单数形式。

[0062] 本文中所描述的技术可用于各种无线电通信系统,诸如CDMA、TDMA、FDMA、OFDMA、SC-FDMA和其他系统。术语“系统”和“网络”常被可互换地使用。CDMA系统可实现诸如通用地面无线电接入(UTRA)、cdma2000等无线电技术。UTRA包括宽带CDMA(W-CDMA)和其他CDMA变体。另外,cdma2000涵盖IS-2000、IS-95和IS-856标准。TDMA系统可实现诸如全球移动通信系统(GSM)之类的无线电技术。OFDMA系统可实现诸如演进型UTRA(E-UTRA)、超移动宽带(UMB)、IEEE802.11(Wi-Fi)、IEEE802.16(WiMAX)、IEEE802.20、Flash-OFDM等无线电技术。UTRA和

E-UTRA是通用移动电信系统(UMTS)的部分。3GPP长期演进(LTE)是使用E-UTRA的UMTS版本，其在下行链路上采用OFDMA而在上行链路上采用SC-FDMA。UTRA、E-UTRA、UMTS、LTE以及GSM在来自名为“第三代伙伴项目”(3GPP)的组织的文档中进行了描述。另外，cdma2000和UMB在来自名为“第三代伙伴项目2”(3GPP2)的组织的文献中描述。此外，此类无线通信系统还可另外包括常使用非配对无执照频谱、802.xx无线LAN、蓝牙以及任何其他短程或长程无线通信技术的对等(例如，移动对移动)自组织(ad hoc)网络系统。

[0063] 各个方面或特征将以可包括数个设备、组件、模块、及类似物的系统的形式来呈现。将理解和领会，各种系统可包括附加设备、组件、模块等，和/或可以并不包括结合附图所讨论的设备、组件、模块等的全体。也可以使用这些办法的组合。

[0064] 参照图1，在一个方面，用于渲染诸如网页之类的页面的装置和方法包括具有门户组件12的计算机设备10，该门户组件12与web引擎14交互以使得能够在显示器15上渲染诸如网页之类的页面13的一个或多个实例。例如，所显示的页面13可经由通信网络17从web服务器16和数据服务器18获得。可以是分开的或者是集成的组件的web服务器16和数据服务器18分别存储分别定义第一页面实例28和第二页面实例30的web资源20和22以及数据项24和26。例如，web资源20和22可包括为构造页面的静态部分所需要的任何东西(例如，可用数据填充的空页面模板)，而数据项24和26可包括用于填充页面的任何内容(例如，页面的一个或多个动态部分)。由此，web资源20和22的全部或一些部分是为第一页面实例28和第二页面实例30两者所共有的，而数据项24和26的全部或一些部分是各自相应的页面实例所独有的。

[0065] 更具体地，门户组件12诸如基于用户输入或者门户组件12的启动来传达接收到的请求32，该请求32向web引擎14标识所请求的页面实例34。例如，接收到的请求32可包括定义web资源(诸如通用资源定位符(URL))的第一部分，和定义要取回的具体数据(诸如查询字符串)的第二部分。web引擎14包括匹配确定器36，该匹配确定器36执行以确定如由请求32标识的所请求的页面实例34是否对应于存储着的关于该页面的文档对象模型(DOM)38。DOM38定义呈子-父关系的一个或多个节点的树或阶层，包括各自相应的DOM节点的性质和/或属性及其样式属性，这使得能够渲染该页面。如果所请求的页面实例34对应于存储着的DOM38，则迭代器组件40标识DOM38的为所请求的页面实例34和一个或多个存储着的页面实例44两者所共有的一个或多个静态部分42。可任选地，应当注意，DOM38可进一步包括因页面44的这一个或多个存储着的实例中的每一个实例而异的一个或多个动态部分46。例如，存储着的页面实例44可对应于第一页面实例28，而所请求的页面实例34可对应于第二页面实例30。另外，基于请求32，迭代器组件40标识、创建和存储因所请求的页面实例34而异的一个或多个动态部分48。由此，DOM38定义了页面的不同实例之间的关系，以及为每个页面实例所独有的并且进一步与由不止一个页面实例共享的一个或多个静态DOM部分相对应的一个或多个相应的动态DOM部分。因此，DOM38可被称为共享DOM，这是因为DOM38包括对于存储着的页面实例44和所请求的页面实例44两者而言相同的静态DOM部分。

[0066] 替换地，如果匹配确定器36确定所请求的页面实例44并不匹配或者相关于任何存储着的DOM，则匹配确定器36操作用于创建和存储关于该页面的新DOM39。可具有与共享DOM38类似的结构的新DOM39则可稍后由匹配确定器36参考以确定与对页面实例的后续请求的匹配。

[0067] 在这些方面,共享DOM38可允许更快速地渲染所请求的页面实例34,这是因为web引擎14不需要为与已存储的DOM相对应的新的页面实例重构整个DOM结构。取而代之的是,web引擎14可重用静态DOM部分42,并且仅需要执行与对应于所请求的页面实例34的这一个或多个动态DOM部分48有关的处理。

[0068] 另外,计算机设备10可包括可操作用于执行脚本化功能性的脚本化引擎50。例如,web引擎14与脚本化引擎50交互以加载对应于页面的脚本,该脚本化引擎50执行该脚本以注册相应的脚本功能。响应于从web引擎14接收到的加载事件,脚本化引擎50进一步运行页面脚本并且生成发送调用以发起对该网页实例的相应动态数据的取回。另外,在接收到动态数据之后,脚本化引擎50可操作用于例如经由DOM应用程序接口(API)应用该数据并且将该数据写入相应的DOM。相应地,基于为相应页面实例所写入的动态数据,web引擎14创建并将一个或多个动态DOM部分与各自相应的页面实例相关联地存储。

[0069] 另外,计算机设备10包括与web引擎14交互的渲染引擎52。例如,响应于来自web引擎14的请求,渲染引擎52可操作用于绘制相应页面的与一个或多个静态DOM部分相对应和/或与相应页面实例的一个或多个动态DOM部分相对应的一个或多个部分。另外,渲染引擎52与显示器15交互以呈现所绘制的页面部分,由此呈现相应的页面实例。

[0070] 因此,通过提供具有共享静态部分的DOM,所描述的方面可在取回与每个新的页面实例相对应的web资源时节省通信资源,或者所描述的方面可通过重用静态DOM部分在创建DOM的诸部分时节省处理资源,或者可节省两者。

[0071] 参照图2,在一个方面,共享DOM38可被进一步解释为表示由多个节点(诸如节点62、64、66、68、70和72)定义的模板结构60并且进一步表示不同的页面实例(诸如第一页面实例28、第二页面实例30和第三页面实例74)。每个页面实例(例如,28、30和74)共享一个或多个共同节点,诸如节点62、64、66、68和70。然而,分别与每个页面实例的每个节点相关联的页面数据可以不同。例如,页面数据82、84、86、88和90分别对应于第一页面实例28的节点62、64、66、68和70,而页面数据92、94、96、98和100分别对应于第二页面实例30的节点62、64、66、68和70,而页面数据102、104、106、108和110分别对应于第三页面实例74的节点62、64、66、68和70。另外,应当注意,一些页面实例可具有比其他页面实例多或者少的节点。例如,在图2中,第三页面实例74包括具有相应页面数据112的在任何其他页面实例中不存在的附加节点72。

[0072] 共享DOM38定义诸节点、页面数据和页面实例之间的关系,该关系提供用以标识和构造不同的页面实例的简单且高效率的机制。例如,共享DOM38标识各个页面实例28、30和74的所有节点(包括共享节点62、64、66、68和70)及其彼此之间的相应关系,以及为任何实例所独有的任何节点(诸如节点72)。另外,共享DOM38包括与每个节点相对应的静态DOM部分122、124、126、128和130(例如,共有部分)以及与每个节点相对应的动态DOM部分或动态页面数据,诸如分别对应于节点62的实例28、30和74的动态数据82、92和102,分别对应于节点64的实例28、30和74的动态数据84、94和104,分别对应于节点66的实例28、30和74的动态数据86、96和106,分别对应于节点68的实例28、30和74的动态数据88、98和108,以及分别对应于节点70的实例28、30和74的动态数据90、100和110。另外,共享DOM38包括各独有节点的动态DOM部分,诸如与节点72的第三页面实例74相对应的动态数据112。

[0073] 参照图3,在一个方面,诸如移动或蜂窝电话之类的计算机设备10可包括用于执行

与本文中所描述的一个或多个组件和功能相关联的处理功能的处理器150。处理器150可包括单个或多个处理器或多核处理器集合。另外，处理器150可被实现为集成处理系统和/或分布式处理系统。

[0074] 计算机设备10可进一步包括诸如用于存储正由处理器150执行的应用的本地版本和/或用于对当前并非正被处理器150执行或利用的数据、指令和/或应用的长期存储的存储器152。在一个方面，存储器152可进一步包括推导式高速缓存155。一旦接收到旨在被高速缓存的内容，计算机设备10就可进一步处理所接收到的内容以生成推导出的内容并且将推导式内容157存储在推导式高速缓存155中。在一个方面，此处理可通过预先准备经常使用的推导式内容157来辅助高效率地促成对高速缓存着的内容的后续使用。在另一方面，存储器152可包括结构化高速缓存159。结构化高速缓存159可允许以如下方式来存储内容：可通过选取内容的选定已知部分供检索来选择性地进行检索。在一个方面，结构化高速缓存159是按如下方式来组织的：允许检索存储在结构化高速缓存159中的内容项的选定已知部分而不需要检索整个内容项。存储器152可附加地存储与估计和/或利用信道（例如，基于性能、基于容量等）相关联的协议和/或算法。在一个方面，存储器152可进一步包括原始内容存储151。原始内容存储151可允许存储接收到的原始内容。

[0075] 应当领会，本文中所描述的数据存储（例如，存储器152）可以是易失性存储器或非易失性存储器、或者可包括易失性和非易失性存储器两者。在一个方面，可使用双存储器模型，其中第一层是基于文件系统的非易失性存储器并且第二层是堆结构易失性存储器。在此类方面，原始收到的内容可存储在第一层中并且推导式内容可存储在第二层中。另外，在此类方面，第二层可比第一层更快速地访问。作为解说而非限定，非易失性存储器可包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦除PROM(EEPROM)、或闪存。易失性存储器可包括充当外部高速缓存存储器的随机存取存储器(RAM)。藉由解说而非限定，RAM有许多形式可用，诸如同步RAM(SRAM)、动态RAM(DRAM)、同步DRAM(SDRAM)、双倍数据率SDRAM(DDR SDRAM)、增强型SDRAM(ESDRAM)、同步链路DRAM(SLDRAm)、以及直接存储器总线(Rambus)RAM(DRRAM)。本主题系统和方法的存储器152可包括但不限于这些以及任何其他合适类型的存储器。

[0076] 另外，计算机设备10可包括通信组件154，该通信组件154利用本文中所描述的硬件、软件和服务来提供对与一个或多个实体的通信的建立和维护。通信组件154可承载计算机设备10上的诸组件之间以及计算机10与外部设备（诸如位于通信网络另一边的设备和/或串联或本地连接至计算机设备10的设备）之间的通信。例如，通信组件154可包括一条或多条总线，并且可进一步包括可操作用于与外部设备交互的接口、一个或多个发射机或发射链组件以及一个或多个接收机或接收链组件。例如，此类发射机和接收机可根据一种或多种技术或协议来实现有线或无线通信。

[0077] 计算机设备10可附加地包括可操作用于从计算机设备10的用户接收输入158并且进一步可操作用于生成供呈现给用户的输出160的用户接口组件156。用户接口组件156可包括一个或多个输入设备，包括但不限于键盘、数字小键盘、鼠标、触敏显示器、导航键、功能键、话筒、语音识别组件、能够从用户接收输入的任何其他机构、或其任何组合。另外，用户接口组件156可包括一个或多个输出设备，包括但不限于显示器、扬声器、触觉反馈机构、打印机、能够向用户呈现输出的任何其他机构、或其任何组合。

[0078] 另外,如以上所提及的,计算机设备10可包括门户组件12,该门户组件12可以是web浏览器、窗口小部件或移动窗口小部件、或web应用中的一者或任何组合。门户组件12可操作用于与计算机设备10的用户以及与计算机设备10的其他组件接口以使得能够渲染网页。

[0079] 如以上所讨论的,web引擎14管理对DOM匹配的确定以及对DOM的创建和存储。可任选地,在一些方面,web引擎14可包括用于管理取回操作的定时和性能的取回管理器162。例如,取回管理器162可使得能够预取具有刚从引用页面点击的链接的页面的内容,其中引用页面包括URL以执行预取。在一方面,例如,取回管理器162可在引用页面的javascript中执行预取发现调用。在另一方面,例如,取回管理器162可对页面上要取回的URL加标签或者排序,因此在页面渲染时可在接收对相应页面的请求之前处理这些标签或URL。例如,获取管理器162可包括用于确定诸页面之间的关系(例如,列表中毗邻的页面、阶层中毗邻的页面等)的算法或功能以确定要预取何物和何时预取。在其他方面,所描述的装置和方法可在当前页面的HTML头中包括将由取回管理器162分析以供预取的已知URL的清单。在另一方面,所描述的装置和方法可在引用页面的HTML头中包括每个可到达页面的URL的结构化清单,该结构化清单将由取回管理器162分析以对每个可到达页面执行预取。在此方面的进一步衍变中,所描述的装置和方法可将本地的“下一页”清单从页面元数据中分离出来,从而将该清单作为对GRS的getPageManifest()调用来插入。这将允许服务器调谐预取次序而不必更新页面的xhtml页面。这是另一个URL,所以expiry(过期)和ifmodified-since(自从……是否修改过)可控制对此“高速缓存元数据”资源的高速缓存。在另一方面,所描述的装置和方法可包括用更宽泛的预取策略调用来检索此元数据。此类策略更新可包括对整个模板集合的所有预取排序。预取排序可最初用自引导清单来键控和取回。

[0080] 可任选地,在一些方面,web引擎14可包括推导管理器164以在对相应页面实例的请求之前预推导DOM或DOM的诸部分。例如,渲染延迟中有很大部分与使web文档被解析成DOM并且准备好供渲染有关。推导管理器164包括提供在管理在网页被需要的时间之前将网页推导成DOM并高速缓存该DOM的操作中的层的算法或功能。推导管理器164使得此类推导能够以智能方式进行,即推导和高速缓存很有可能不是在经由训令从清单首次下载时进行,而是取而代之地可在web引擎14相对而言不太被使用并且相应页面正被需要的概率很高时进行。例如,推导管理器164的算法或功能可确定诸页面之间的关系(例如,列表中毗邻的页面、阶层中毗邻的页面等)以确定要预推导何物以及何时预推导一个或多个页面。

[0081] 应当注意,取回管理器162和推导管理器164可组合在相同的计算机设备10中以改善会话中页面首次运行的性能。

[0082] 计算机设备10可进一步包括内容处理机166以辅助在高速缓存之前对内容进行处理。在一个方面,内容处理机166允许计算机设备102确定是否要从接收到的原始内容151生成推导式内容157。另外,内容处理机166可从通信网络接收处理指令168。在此类方面,通信网络可提供处理指令168连同要在计算机设备102处存储的内容。在一个方面,处理指令168可诸如通过通信网络来允许一些服务器侧控制,该服务器侧控制传递至计算机设备102以在计算机设备102处按与没有处理指令168的情况相比更高效率的方式对原始内容151执行变换或类似操作。在一个方面,处理指令168可促成将推导式内容157与接收到的原始内容151相关联以允许高效率的后续检索。例如,处理指令168可随图像一起传送并且可提供用

于重新映射图像像素的指令。由此,与处理指令一起传递的内容的大小可以很小。并且一旦由计算机设备102接收到大小很小的原始内容和处理指令,处理指令168就可促成将该内容扩展开来成为更丰富的图像或者更丰富的内容集合。内容处理机166可进一步包括结构参考169以辅助对收到内容的处理。结构参考169可处理收到内容以促成在结构化高速缓存159中的结构化存储。

[0083] 譬如javascript引擎之类的脚本化引擎50管理脚本功能的注册和给定页面实例的脚本的运行。渲染引擎52生成或绘制给定页面实例的全部或诸部分,并且将结果传达至用户接口156的组件,诸如在显示器15上呈现的所显示的页面实例13(图1)。

[0084] 现在转到图4,在操作中,在一个方面,用于渲染页面的方法170包括在计算机设备处接收对所请求的页面实例的请求(框172)。

[0085] 该方法进一步包括确定所请求的页面实例是否对应于存储在存储器中的关于该页面的文档对象模型(DOM)。存储在存储器中的DOM对应于存储着的页面实例并且包括对于存储着的页面实例和所请求的页面实例两者而言相同的静态部分(框174)。

[0086] 另外,该方法包括如果所请求的页面实例对应于存储在存储器中的DOM,则引用该DOM的与所请求的实例相对应的动态部分。该动态部分是为所请求的页面实例所独有的(框176)。

[0087] 该方法还包括将该DOM的与所请求的页面实例相对应的动态部分与该DOM的静态部分关联地存储(框178)。可任选地,该方法进一步包括根据DOM的静态部分和DOM的与所请求的页面实例相对应的动态部分来渲染所请求的页面实例(框180)。

[0088] 参照图5和图6,在一方面,用于渲染页面的相应第一实例和第二实例的方法包括诸如基于用户输入或门户应用的启动来请求第一页面实例(202、204、302、304)。该方法进一步包括web引擎确定该页面实例是否对应于存储在存储器或高速缓存中的DOM(206、306)。在207处,如果该页面实例对应于存储在存储器或高速缓存中的DOM(例如,预先推导出的DOM),则该方法继续行至绘制过程(218)。然而,如果该页面实例不对应于存储在存储器或高速缓存中的DOM,则分别向web服务器传送相应的GET并从web服务器接收相应的响应(208、210)。该响应包括例如xhtml、css和javascript。

[0089] 在一些方面,在211处,可执行对收到内容的预处理,例如,推导收到内容以生成推导式内容。在一个方面,从web服务器获得的原始内容可伴随有处理指令。一旦从web服务器接收到原始内容,内容处理机就可确定是否可在高速缓存接收到的原始内容之前或者与高速缓存接收到的原始内容同时地执行附加处理。在一个方面,这些处理指令辅助至少确定是否要进一步处理接收到的原始内容。在另一方面,接收到的原始内容可存储在由与原始内容相关联的已知结构决定(诸如可由结构参考、服务器生成的入口点、或类似物来定义)的结构化高速缓存中。一旦内容处理机确定可处理接收到的原始内容的至少一部分,则内容处理机就处理接收到的原始内容的选定部分以生成推导式内容。推导式内容可存储在推导式高速缓存中。在一个方面,推导式内容进一步与接收到的原始内容相关联并且被相应地存储。例如,推导式内容可包括经编译的javascript和/或经编译的css以供随后在渲染相应的第一和第二页面实例时使用。如以下所讨论的,图7-9解说了用于预处理内容的各种方法体系。

[0090] 随后,在高速缓存中存储响应数据(212),并且创建和存储静态DOM部分(214)。创

建和存储静态DOM部分可包括从该响应推导内容并且该内容变换成能够被web引擎处理的形式。例如,从该响应推导内容可包括解析该响应(例如,xhtml代码)以及从经解析信息创建DOM。另外,规则可被构造并被应用于DOM以辅助静态DOM的创建。这些规则可包括例如CSS规则以及其他类型的规则。应当领会,静态DOM可被存储在存储器或高速缓存中以供稍后使用。在一个方面,预处理静态DOM的全部或一部分(211),并且因此静态DOM部分可包括推导式内容。替换地,对于第二实例,寻找和使用静态DOM部分(316)。在这两种情形中,均开始绘制过程以绘制和显示一个或多个静态DOM部分。更具体地,web引擎开始绘制过程(218、318)并且发起渲染引擎以绘制该静态部分(220、320)。渲染引擎向显示器转发所绘制的静态部分(222、322)以显示一个或多个静态DOM部分(224、324)。注意,此绘制过程的定时可以变化。另外,推导出的css内容(211)可被用于在绘制过程期间创建静态结构和文本(220、320)。

[0091] 另外,在获得响应和存储着的静态DOM部分之后,发起加载脚本操作(226、326),其中脚本引擎注册脚本功能(228、328)、接收加载事件(230、330)、并且随后运行页面脚本(232、332)。推导出的javascript(211)可在脚本化功能期间使用(228、232、328、332)。XHR发送事件(234、334)随后被传送给web引擎,该web引擎进行检查以查明数据是否已存储在高速缓存中(236、336)。如果是,则在237和337处,存储着的数据可被利用并被传递至脚本化引擎(244、344)。如果不是,则分别向数据服务器传送数据取回和从数据服务器接收响应(238和240、338和340)。这可以是XHR调用(238)和XHR结果(240)。在一些方面,在241和341处,可如以上在211中讨论的那样执行对收到内容的预处理。如以下所讨论的,图7-9解说了用于预处理内容的各种方法体系。

[0092] 响应数据存储在高速缓存中(242、342),并且结果传递至脚本化引擎(244、344),该脚本引擎经由DOM API应用数据并且将该数据写入web引擎中的高速缓存(246和248、346和348)。web引擎随后为该数据的一个或多个动态部分来为给定实例创建和存储动态DOM部分(250、350)。由于绘制操作可能是资源昂贵的,因而web引擎可缓冲此DOM部分(252、352)。随后,web引擎发起与以上针对静态部分的操作类似的操作,这导致给定的页面实例被创建。web引擎开始绘制过程(254、354)并且发起渲染引擎以绘制静态部分(256、356)。渲染引擎将所绘制的静态部分转发给显示器(258、358)以用于显示一个或多个静态DOM部分(260、360),从而导致给定的页面实例被创建(262、362)。另外,注意,推导出的css内容(241、341)可被用于在绘制过程期间创建静态结构和文本(256、356)。

[0093] 可任选地,在一些方面,图5和图6的方法可包括如以上参照取回管理器162(图3)所描述的一个或多个预取操作(264、364)。应当注意,预取操作的在图5和图6中的操作顺序中的定时可以基于对何时想要此类预取的确定而相对于所解说的动作发生变化。另外,应当注意,预取操作可进一步包括附加的GET和响应、和/或XHR调用和结果、以及相应的预先处理和后续处理,如以上所描述的。

[0094] 可任选地,在一些方面,图5和图6的方法可包括如以上参照推导管理器164(图3)所描述的一个或多个预推导操作(266、366)。应当注意,预推导操作的在图5和图6中的操作顺序中的定时可以基于对何时想要此类预推导的确定而相对于所解说的动作发生变化。另外,应当注意,预推导操作可进一步包括附加的GET和响应、和/或附加的XHR调用和结果、以及相应的预先处理和后续处理,如以上所描述的。

[0095] 现在转到图7,解说了用于促成高效率的本地高速缓存的示例性方法1000。在附图标记1002处,接收原始数据。在一个方面,该数据可伴随有处理指令。该数据可由计算机设备或类似物接收,并且该数据可发源自服务器、通信网络等。例如,在一方面,原始数据可包括css和javascript。

[0096] 在附图标记1004处,该方法包括确定收到数据是否是用于本地处理的候选。例如,收到数据可包括提示进一步处理的处理指令。在另一方面,收到数据可以属于提示进一步处理的已知数据类型,诸如css或javascript。如果确定不需要进一步的处理,则在附图标记1006处,收到数据可被高速缓存。附加地或可任选地,在一个方面,收到数据可被高速缓存,而不论该数据是否是用于进一步处理的候选。相比之下,如果在附图标记1004处确定提示了进一步处理,则在附图标记1008处,从收到内容生成本地推导出的内容。在一个方面,处理指令可引导内容处理机生成推导式内容。例如,收到图像可被处理以生成在计算设备上常用的一个或多个大小。在另一示例中,接收到的程序文件可被处理以生成经编译程序以供计算设备随后使用。在又一示例中,接收到的字体类型可被处理以生成各种常用的颜色和/或大小变型以供计算设备随后使用。在附图标记1010处,至少将推导式内容存储在高速缓存中。在另一方面,推导式内容和收到内容被存储并且相关联以辅助后续的检索。在一个方面,对原始数据和一个或多个推导出的内容项的高速缓存可通过对推导式内容项消除了检索后的处理要求来改善感知到的应用性能。在又一方面,处理指令随推导式内容和收到内容一起被存储并且被用于使推导出的内容与收到内容相关联。

[0097] 现在转到图8,解说了根据一个方面的用于在高速缓存中进行存储之前通过处理指令来促成内容修改的示例性方法1100。在一个方面,在附图标记1102处,计算设备可请求、取回要在高速缓存中存储的内容,或进行类似操作。在此类方面,该请求可伴随有协商指令。例如,该请求可针对图像,其中伴随的指令提供了优选图像的可能的像素范围。一般地,在附图标记1104处,接收原始数据。此类数据可伴随有处理指令。在一个方面,这些处理指令是至少部分地响应于请求协商指令所生成的。数据可由计算机设备或类似物接收,并且数据可发源自服务器、通信网络等。

[0098] 接下来要描述的是,在附图标记1106处,通过分析伴随的处理指令来确定收到数据是否是用于本地处理的候选。如果处理指令不提示进一步的处理,则在附图标记1108处,收到数据可被高速缓存。作为对比,如果在附图标记1106处,处理指令提示了附加处理,则在附图标记1112处,可向内容处理机提供处理指令和收到内容。在附图标记1112处,通过将处理指令用作指导来从原始内容生成本地推导出的内容(例如,推导式内容)。由此,服务器或类似物可促成在传输之后对原始内容的处理。例如,通过由处理指令提示内容处理机要平滑收到图像的渐变,收到图像可被处理以生成具有较小粒度的图像。

[0099] 在附图标记1114处,至少将推导式内容存储在高速缓存中。在另一方面,推导出的内容和收到内容被存储并且相关联以辅助后续的检索。在又一方面,处理指令随推导出的内容和收到内容一起被存储并且被用于使推导出的内容与收到内容相关联。

[0100] 现在转到图9,解说了用于促成在高速缓存中的结构化存储的示例性方法900。一般地,在附图标记902处,接收语义不透明的数据。在一个方面,该数据可伴随有由服务器生成的进入点以允许对选定部分的结构化检索。在另一方面,该数据可以属于已知结构。该数据可由计算机设备或类似物接收,并且数据可发源自服务器、内容提供商等。

[0101] 接下来要描述的是,在附图标记904处,确定收到数据是否可以与已知结构相关联。例如,收到数据可包括由服务器生成的入口点或者可以属于已知结构。如果确定该数据不是以允许进一步处理的方式被结构化的,则在附图标记906处,收到数据可被高速缓存。作为对比,如果在附图标记904处确定提示了进一步处理,则在附图标记908处,可确定数据结构。例如,某些媒体类型可用允许处理的已知结构来定义以促成选择性的后续检索。在附图标记910处,参照接收到的结构化内容的已知内容来存储该结构化内容。

[0102] 在一个方面,在附图标记912处,可接收对结构化内容的一部分的请求。在此类方面,在附图标记914处,可检索结构化内容的被请求部分。例如,可从高速缓存中的结构化内容中选择性地检索与制造商或标题或类似物相关联的信息而不移除整个内容项。

[0103] 图10-13中解说了根据一方面的由移动电话来执行的移动购物应用的一个解说性示例。现在转到图10,在一方面,主页面500(例如,第一页面实例)可由用户请求并且在移动设备上显示,如以上关于图1-7所讨论的。主页面500可包括目录列表504,该目录列表504具有与目录中的各个商品有关的信息。目录列表504可包括目录中的数据类别的列表(例如,506a-506n)、关于目录中的商品的促销信息、目录中的商品的预览和/或描述、或是要从目录中购买的商品、以及其他信息。如果用户想要查看不同的目录列表,则用户可例如通过选择按钮502来移至购物应用中的不同目录列表。替换地或补充地,用户可以例如选择目录中的数据类别之一(例如,“本周新品”506c)并且移至所选类别的目录页面。图11中解说了目录页面(例如,第二页面实例)的示例。

[0104] 现在参照图11,在一方面,目录页面600可由用户请求并且在移动设备上显示,例如,用户从主页面500(图10)选择目录页面600。当目录页面600被请求时,移动设备可确定目录页面600是否具有与主页面500和/或购物应用中的其他页面相对应的相似结构。一旦确定目录页面600具有与主页面500相似的结构,移动设备就可检索并重用来自主页面500的DOM的目录页面600的相似结构。另外,移动设备可确定目录页面600具有不同于主页面500的结构并且可为目录页面600所独有的商品创建DOM。

[0105] 目录页面600可包括描述供购买的商品列表608a-608n和/或商品类别606的标题604(例如,本周新品)。类别可包括例如游戏、音乐、视频、畅销货、特色商品、和/或推荐商品、以及其他类别。应当领会,标题604可对应于用户在主页面500(图10)上的选择。另外,供购买的商品608a-608n可包括:描述供购买的商品的名称612a-612n;商品的价格614a-614n;解说和/或描述供购买的商品的图标610a-610n;所显示的商品评级616a-610n(例如,概述与供购买的商品的质量和/或性能有关的数据的星形或文本);以及商品的供应商;以及与供购买的商品有关的其他信息。应当领会,供购买的商品可包括例如游戏、音乐、视频或书籍、以及其他商品。

[0106] 例如,移动设备可确定标题604、供购买的商品列表608a-608n和/或商品类别606可具有与主页面500中的特征相似的结构(例如,标题504、按钮502和列表506a-506n)并且可检索并重用来自主页面500DOM的相似结构。另外,移动设备可确定标题名称604(例如,“本周新品”)、游戏名称612a-612n、游戏图标608a-608n、游戏价格614a-614n、和/或游戏评级616a-616n为目录页面600所独有,并且由此可为目录页面600所独有的这些特征创建动态DOM。

[0107] 因此,移动设备为目录页面600中的为目录页面600所独有的部分创建动态DOM并

且重用来自主页面500的静态DOM的相似结构和/或实例。因此,由于移动设备不需要为目录页面600重构整个DOM结构,因而允许更快速地渲染目录页面600。

[0108] 如果用户想要查看不同的目录页面或者返回到主页面500,则用户可例如通过选择按钮602移至购物应用中的不同目录页面。替换地或补充地,用户可例如选择目录页面600上列出的类别和/或商品之一(例如,游戏“游戏#5名称”612e)并且移至商品页面(例如,另一个(第二)页面实例)。图12和13中解说了商品页面(例如,第二页面实例)的示例。

[0109] 现在参照图12,在一方面,商品详情页面700可由用户请求并且在移动设备上显示,例如,用户从目录页面600(图11)选择商品详情页面700。当商品详情页面700被请求时,移动设备可确定商品详情页面700是否具有与目录页面600、主页面500和/或购物应用中的其他页面相对应的相似结构和/或实例。如以上参照图11所讨论的,移动设备可从主页面500和/或目录页面600的DOM检索并且重用目录页面600和/或主页面500的相似结构。另外,移动设备可为商品详情页面700中的与目录页面600和/或主页面500中的结构或项目不匹配的部分创建DOM。

[0110] 例如,移动设备可确定商品详情页面700上的按钮702、游戏名称706、图标704、价格710、游戏评级708可具有与目录页面600的特征相似的结构并且可从目录页面600DOM检索和重用相似的结构。另外,移动设备可为商品详情页面700中的与目录页面600和/或主页面500中的结构或项目不匹配的部分(例如,对此商品评级选项720、更多类似商品推荐716)创建DOM。因此,移动设备为商品详情页面700中的为商品详情页面700所独有的部分创建动态DOM并且重用来自目录页面600和/或主页面500的相似结构和/或实例。因此,由于移动设备不需要为商品详情页面700重构整个DOM结构,因而允许更快速地渲染商品详情页面700。

[0111] 如果用户想要查看前一页面,则用户可选择按钮702以返回购物应用中的前一页面。替换地或补充地,用户可选择例如商品详情页面700上的另一商品(例如,游戏“吉他英雄III”718)并且移至另一商品详情页面(例如,图13中解说了商品详情页面800)。

[0112] 现在转到图13,解说了根据一方面的商品详情页面800的另一示例。如以上参照图12所讨论的,移动设备可确定商品详情页面800是否具有与商品详情页面700、目录页面600、主页面500和/或购物应用中的其他页面相对应的相似结构和/或实例。移动设备可从商品详情页面700、主页面500和/或目录页面600的DOM检索并且重用详情页面700、目录页面600和/或主页面500的相似结构。另外,移动设备可为商品详情页面800中的与商品详情页面700、目录页面600和/或主页面500中的结构或项目不匹配的部分创建动态DOM。例如,在此情形中,静态DOM部分可包括商品详情页面700的相同布局,而动态数据项(诸如游戏名称806、游戏评级808、游戏价格810)则由动态DOM部分来表示。

[0113] 参照图14,解说了渲染经由内容提供商从web服务器和数据服务器接收到的页面(诸如网页)的装置400。在一些方面,例如,装置400可驻留在移动设备内。如所描绘的,装置400包括可表示由处理器、或软件、或其组合(例如,固件)实现的功能的功能框。装置400包括促成接收与页面相对应的web资源和数据并且渲染页面实例的电组件的逻辑编组402。逻辑编组402可包括用于诸如在计算机设备处接收对所请求的页面实例的请求的装置404。另外,逻辑编组402可包括用于确定所请求的页面实例是否对应于存储在存储器中的关于该页面的文档对象模型(DOM)的装置406,其中存储在存储器中的DOM对应于存储着的页面实例并且包括对于存储着的页面实例和所请求的页面实例两者而言相同的静态部分。另外,

逻辑编组402还可包括用于如果所请求的页面实例对应于存储在存储器中的DOM则检索该DOM的与所请求的实例相对应的动态部分的装置408，其中该动态部分为所请求的页面实例所独有。另外，逻辑编组402可包括用于将DOM的与所请求的页面实例相对应的动态部分与该DOM的静态部分关联地存储的装置410。可任选地，逻辑编组402可包括用于根据DOM的静态部分和该DOM的与所请求的页面实例相对应的动态部分来渲染所请求的页面实例的装置412。因此，装置400可在渲染页面时高效率地重用静态部分，并且创建和存储共享DOM，该共享DOM具有定义与相应的一个或多个页面实例相对应的一个或多个动态DOM部分的关系。

[0114] 另外，装置400可包括存储器414，其保存用于执行与电组件404、406、408、410和(可任选地)412相关联的功能的指令。虽然被示出为在存储器414外部，但是应当理解，电组件404、406、408、410和(可任选地)412可以存在于存储器414之内。

[0115] 除了以上提及的高速缓存和处理技术之外，其他方面提供了用于通过除了重用HTML代码之外还重用HTML计算来改善浏览器的性能的系统、方法和设备，其中HTML计算被链接或索引至相应的DOM树结构。在加载网页时，浏览器通常执行两个高等级任务：通过因特网取回HTML代码(本文中称“网络元素”)以及执行HTML计算以处理HTML代码的内容(本文中“计算元素”)。常规浏览器通过在本地存储器中高速缓存接收自特定URL的HTML代码由此节省下载HTML的时间并且改善网络元素的方式，来改善网页渲染过程。通过高速缓存HTML代码，浏览器一接收到访问相应的HTML代码被高速缓存着的统一资源定位符(URL)或统一资源标识符(URI)的请求就可开始渲染页面。因此，常规浏览器存储HTML代码，以使得这些浏览器一确定所请求的URL/URI与具有至存储着的HTML代码的链接的URL/URI组件相匹配就可开始渲染页面，如同已在因特网上接收到该HTML那样。然而，此类浏览器未能针对计算元素，从而每当渲染相同页面时都需要重新执行计算。作为结果，常规浏览器仅将渲染时间改善到联网元素是性能瓶颈的程度。

[0116] 如果数据传输速率慢于本地处理速度，则取回HTML代码就是性能瓶颈。这通常发生在需要在具有有限带宽的网络上取回许多数据的时候。然而，近年来，网络带宽已大为改善。这因3.5G和4G移动网络变得越来越可用而得到证明，其提供超过14.4Mbps的带宽。另外，一旦为URL高速缓存了HTML代码，常规浏览器就不能够再更快速地渲染页面，这对于用户而言可能看起来似乎升级至较快速网络几乎不能改善网页渲染时间。网页正变得日益愈加复杂。现代网页需要大量计算以解析、格式化和渲染收到网页的内容。因此，渲染网页的瓶颈已从网络元素移至计算元素。

[0117] 各个方面提供了将HTML渲染计算高效率地映射和关联至其DOM树结构的方式，以使得为一个页面的一部分演算的计算可在渲染后续页面的一个或多个部分时不是基于URL和/或URI而是基于结构相似性来被高效率地重用。

[0118] web浏览器通过处理HTML代码以计算和定义进入显示的所有元素的方式生成供显示的页面。为计算网页元素的样式、格式和布局要求所需要的时间往往是大量的。例如，布局计算和样式格式化的组合通常占本地网页处理中的总计算时间的一半以上。另外，计算样式属性并且将其应用于DOM元素的过程通常是每当网页被请求和/或渲染时都必须执行的递归且耗时的过程。各个方面通过存储为第一页面演算的计算并且重用存储着的计算以渲染具有相同或相似DOM树结构(即，同构)的第二页面而无需第一和第二页面的网页URL之间有任何关联性的方式来消除这些冗余且耗时的过程。

[0119] 以上参照图1和图2所讨论的方面存储DOM树,以使得(共享URL和/URI的)相同页面的不同实例可共享相同DOM树的各静态部分。在那些方面,当页面的独特实例(例如,“商品详情”页面)被请求时,页面数据(例如,超文本标记语言(html)、级联样式表(css)和javascript(js))可被取回并被用于创建DOM树的静态部分。这些静态部分可被存储在高速缓存中,以使得它们可被重用。也就是说,当存储着的页面的第二实例被请求时,重用第一实例的静态DOM部分并且专门为第二实例创建一个或多个动态部分。作为结果,这些方面允许定义DOM,以使得DOM的诸部分可被重用于渲染相同页面的不同实例。以下参照图15–20描述的方面将这些能力扩展到使得即使在URL或URI不同的情况下也能够重用HTML计算。

[0120] 在渲染网页时,浏览器首先解析HTML代码(无论是经由网络接收的还是从存储器调用的)以生成DOM树。常规浏览器以及各个实施例的浏览器在该DOM树不匹配存储器中的任何DOM树结构时则处理该DOM树以执行为定义要被渲染的页面的特征和特性所需要的计算。这些计算可包括例如为要在页面上显示的每个子元素计算尺寸和样式。有许多对于任何给定的网页而言可能都需要的众所周知的计算。这些计算可包括计算布局、标识样式、执行CSS匹配、以及需要处理器和/或处理密集的计算的其他操作。在各个方面,在存储器中存储这些计算连同DOM树的相应部分。随后,每当渲染页面时,在执行计算之前将所生成的DOM树与高速缓存着的DOM树作比较以确定是否有任何匹配。如果在存储器中标识出匹配的DOM树结构,则在存储器中调用并且重用相应的HTML计算结果,而且该页面的DOM树中仅与存储着的树不匹配的那些部分被处理以完成为渲染该页面所需要的HTML计算。浏览器可使用众所周知的树同构算法来标识存储着的DOM树部分与所请求的页面的DOM树之间的结构相似性。因此,此可重用性确定是独立于URL和/或URI地作出的。

[0121] 以下描述的方面通过在存储器中的DOM树中存储结果得到的计算并且基于DOM树结构进行识别以标识存储着的计算的可被重用于渲染所访问的网页的诸部分,由此增加可供浏览器重用的HTML渲染计算的量的方式,来进一步改善浏览器性能。为了达成此举,除了高速缓存DOM树的静态部分和动态部分之外,在存储器中还存储实际的计算。在各个方面,结果得到的计算可被高速缓存,以使得高速缓存着的计算被索引至其DOM树(即,在完成这些计算时所执行的DOM树)。在一方面,结果以键-值(即,标签和值)对的形式存储在存储器中。键或标签标识树结构元素,而值是该元素的HTML计算的结果。通过将计算索引至DOM树,可在渲染具有结构上相似的DOM树部分的任何页面时以独立于与原始高速缓存着的页面或者要被渲染的页面相关联的URL或URI的方式重用高速缓存着的计算。存储索引至DOM树的计算还允许快速地分析存储着的DOM树的结构而无需任何特殊标签、关于URL和/或URI的信息、或者任何其他附加的标识机制。

[0122] 各个方面使用各种众所周知的树同构算法来识别和标识正被渲染的页面的DOM树和存储在存储器中的DOM树部分的同构部分(即,共享相似的树结构的部分)以识别可被重用的高速缓存着的计算。例如,浏览器可使用诸如在“The Design and Analysis of Computer Algorithms(计算机算法的设计和分析)”,Addison-Wesley,1974中描述的Alogtime、Aho、Hopcroft、Ullman和其他同构算法之类的关于树同构的公知算法来标识匹配的DOM树部分,该文献的全部内容纳入于此。

[0123] 当浏览器识别出当前DOM树的诸部分与存储着的DOM树或者存储着的DOM树部分的诸部分相匹配时,与标识出的部分相对应的高速缓存着的计算结果可被快速地且高效率地

从存储器调用。这可通过以键值和输入值的形式来存储计算结果的方式达成。在这些方面，可将用于创建高速缓存着的计算结果的键值和输入值与正被渲染的当前页面的键值和输入值作比较以确定这些值是否相同，以使得存储着的计算结果的一个或多个部分可被重用。如果确定了存储着的计算的一个或多个部分是可重用的，则从存储器调用存储着的计算值并且由浏览器使用这些计算值。当页面的所有可重用的HTML计算已从存储器调用并且所有其他计算已由浏览器执行时，所检索的和所执行的计算结果可被用于显示该页面。将此能力与常规的HTML高速缓存能力相组合使得浏览器能够非常快速地显示页面而无需下载或计算新请求的HTML页面。

[0124] 图15解说了通过浏览器解析所请求的网页的HTML代码所生成的DOM树1500的样本部分。如图15中所解说得，在解析HTML代码时，浏览器可为每个标头创建DOM元素和/或节点。在所解说得的示例中，HTML代码包含报头部分（例如，`<HEAD>...</HEAD>`）和主体部分（例如，`<BODY>...</BODY>`）。主体部分包含至少一个Div（章节）部分和至少一个Span（跨度）部分，如由DOM树1500中的Div（章节）和Span（跨度）节点所解说得。Div和Span节点中的每一者还可包含子部分，如由Div部分的Paragrpah（段落）部分所解说得。在完成用于渲染页面的计算的过程中，DOM树元素（例如，Body、Div、Paragraph等）可被遍历以计算各个页面组件的属性和特性。例如，对于计算所请求的页面的布局，DOM树可被遍历以提取每个元素的高度、宽度和位置的绝对值（例如， $h=50$ 、 $w=50$ 、 $x=500$ 、 $y=10$ ），这些绝对值通常取决于运行时和环境因素（例如，窗口大小）并且不能提前确定。在各个方面，这些计算出的位置值可随后存储在链接至DOM树的存储器中以供在渲染包括相同或相似计算的未来页面时重用。

[0125] 图16解说了可为DOM树1500中的每个元素存储计算出的值。在所解说得的示例中，表示每个元素的高度、宽度、水平位置（X）和垂直位置（Y）的绝对位置值可作为散列表、散列映射、链接表、二进制表、相关联的数组被存储在存储器1610中。在各个方面，该信息可作为散列表和/或散列映射被存储在存储器1610中，以使得存储着的信息的键-值对映射至DOM树中的相应节点。例如，键-值对可被存储，以使得键元素标识DOM树元素并且（例如，经由散列映射的散列函数）定位相应的值元素（例如，该DOM元素的绝对位置值）。在各个方面，散列表和/或散列映射的维度可被良好地设计，以使得为标识出值（例如，绝对值）所需要的指令的数目独立于DOM树中的元素的数目。在各个方面，散列表和/或散列映射可被用于在单个统一的数据结构中存储这些DOM树及其相关联的属性。本领域中已知的其他数据结构或者数据结构的组合也可被使用。

[0126] 图17解说了用于在浏览器的首次执行时和/或具有未标识出的DOM结构的特定网页的首次渲染时创建、处理和存储DOM树的方法1700的一方面。在方法1700中，在框1702中，浏览器可从与期望网页相对应的远程或本地服务器接收HTML代码。在框1704中，浏览器可解析接收到的HTML并且创建DOM树。在框1706中，浏览器可处理DOM树以完成所有必要的计算。在框1708中，浏览器可在存储器中高速缓存计算结果，并且在框1710中构成DOM树和/或经处理的HTML代码以供输出给显示器。在框1712中，浏览器可在其上运行该浏览器的计算设备的电子显示器上显示所构成且经处理的HTML代码连同所有关联的图像和文件。

[0127] 图18解说了用于在浏览器的后续执行时创建、处理、存储、和/或检索DOM树的诸部分的方法1800的一方面。在方法1800中，在框1802中，浏览器可从服务器接收HTML代码和/或从高速缓存存储器调用HTML代码。在框1804中，浏览器可解析HTML代码并且为接收到的

HTML代码创建DOM树的至少一部分。在确定框1806中,浏览器可通过确定DOM树结构是否同构来确定DOM树的所生成部分是否与存储在存储器中的一个或多个DOM树的任何部分相匹配。在各个方面,浏览器可通过评估一个或多个存储着的DOM树的索引来作出此确定。在各个方面,浏览器可对照存储在存储器中的DOM树的一个或多个部分的散列键来检查与DOM树的所生成部分相关联的散列键。在各个方面,浏览器可执行一个或多个同构算法或发起对一个或多个同构算法的执行以确定所生成的DOM树的诸部分是否与存储在存储器中的DOM树的任何部分相匹配。确定框1806中同构算法的执行可确定所生成的DOM树的诸部分是否与存储在存储器中的DOM树的诸部分在结构上相同和/或在结构上相似。

[0128] 返回图18,如果确定DOM树的所生成部分不与存储在存储器中的任何DOM树的结构相匹配(即,确定框1806=“否”),则在框1808中,浏览器可处理DOM树的所生成部分并且执行相关联的计算(例如,计算元素的绝对位置、样式、执行CSS匹配等)。在框1810中,浏览器可高速缓存计算结果,其中这些结果被索引至DOM树的所执行部分。

[0129] 另一方面,如果确定了DOM树的所生成部分与存储在存储器中的DOM树的任何部分在结构上相同和/或在结构上相似(即,确定框1806=“是”),则在框1812中,浏览器可从存储器调用与匹配的DOM索引相对应的高速缓存着的计算结果。在确定框1814中,浏览器可确定存储在存储器中的匹配部分的键和/或输入数据是否与所生成部分的键和/或输入数据相同,从而指示存储着的计算可在生成所请求的HTML页面时是可重用的。如果存储在存储器中的匹配部分的键和/或输入数据与所生成部分的键和/或输入数据并不相同(即,确定框1814=“否”),则在框1818中,浏览器可处理所生成的DOM树并且执行计算。如果键和/或输入数据匹配(即,确定框1814=“是”),则在确定框1816中,浏览器可确定是否所有必要的计算都已被计算或者从存储器调用。如果不是所有计算均完成了(即,确定框1816=“否”),则浏览器可通过返回框1804来解析HTML代码的另一部分以生成DOM树的附加部分。如果所有计算均已完成(即,确定框1816=“是”),则浏览器可在框1818中验证数据并且构成供渲染的页面,以及在框1820中在计算设备的电子显示器上显示该页面。

[0130] 图19解说了用于在浏览器的后续执行之际创建、处理、存储、和/或检索DOM树的诸部分的方法1900的替换方面。在方法1900中,在框1902中,浏览器可从高速缓存存储器和/或远程服务器接收HTML代码。在框1904中,浏览器可解析接收到的HTML并且基于接收到的HTML来生成DOM树的一个或多个部分。在框1908中,浏览器可通过执行实现同构算法以确定所生成的DOM树的诸部分是否与存储在存储器中的DOM树的任何部分相匹配(例如,在结构上与存储在存储器中的DOM树的任何部分相似)的线程/进程来处理所生成的DOM树部分并且将其与存储在存储器中的DOM树作比较。如果在高速缓存存储器中没有找到结构上相似和/或相同的DOM树部分(例如,同构部分),则浏览器可使用诸如以上所描述的示例结构之一之类的数据结构在存储器中存储被索引至DOM树的计算。如果在存储器中找到结构上相似和/或相同的DOM树部分(例如,同构部分),则在确定框1910中,浏览器可从高速缓存存储器提取和/或检索这些同构部分并且确定键和/或数据输入值是否匹配。在各个方面,DOM树可被存储在散列表和/或散列映射中,以使得无论何时只要所比较的DOM树的一个或多个部分被确定为同构,则键和/或输入数据就将匹配。在这些方面,可将框1908可与确定框1910和框1912组合,如由框1920所解说得。

[0131] 如果确定DOM树的所生成部分的键和/或数据输入值不与存储在高速缓存存储器

中的DOM树的各部分的任何键和/或数据输入值相匹配(即,确定步骤1910=“否”),则浏览器可通过遍历该DOM树并且执行如以上参照图17在方法1700中所描述的相关联的计算来创建、处理和/或存储所生成部分。另一方面,如果一个或多个DOM树部分被确定具有匹配的键和/或输入值(即,确定步骤1910=“是”),则在框1912中,浏览器可从存储器拉取与匹配的键和/或输入值相关联的值。在确定框1914中,可使用一个或多个已知的数据和/或HTML验证方案来验证所拉取的值(例如,从存储器检索到的存储着的计算)。如果一个或多个验证方案失败(即,确定框1914=“否”),则浏览器可通过遍历DOM树并且执行如以上参照图17在方法1700中所描述的相关联的计算来创建、处理和存储HTML计算的所生成部分。如果数据被确定为是有效的(即,确定框1914=“是”),则浏览器可在框1916中构成网页并且在框1918中在电子显示器上显示结果得到的HTML网页。

[0132] 图20是适于与任何方面联用的移动计算设备的系统框图。典型的移动计算设备2000可包括耦合到内部存储器2002、显示器2003并耦合到扬声器2054的处理器2001。另外,移动计算设备2000可包括可连接到无线数据链路的用于发送和接收电磁辐射的天线2004、和/或耦合到处理器2001的蜂窝电话收发机2005、以及耦合到处理器2001的移动计算设备2024。移动计算设备2000通常还包括用于接收用户输入的菜单选择按钮或摇杆开关2008。

[0133] 以上所描述的各个方面还可在诸如图21中所解说的膝上型计算机2100之类的各种计算设备内实现。许多膝上型计算机包括充当计算机的点示设备的触摸板触摸表面,并且由此可接收与在装备有触摸屏显示器的移动计算设备上实现的拖曳、滚动和轻拂手势类似的拖曳、滚动和轻拂手势。膝上型计算机2100将通常包括耦合至易失性存储器2102和诸如盘驱动器2103之类的大容量非易失性存储器的处理器2101。计算机2100还可包括耦合至处理器2101的软盘驱动器2104和压缩盘(CD)驱动器2105。计算机设备2100还可包括耦合至处理器2101的用于建立数据连接或接收外部存储器设备的诸如USB或FireWire[®]连接器插口之类的数个连接器端口,或者用于将处理器2101耦合至网络的其它网络连接电路2106。在笔记本配置中,计算机外壳包括均耦合至处理器2101的触摸板2107、键盘2108、和显示器2109。计算设备的其它配置可包括众所周知的(例如,经由USB输入)耦合至处理器的计算机鼠标或轨迹球。

[0134] 处理器2001、2101可以是能通过软件指令(应用)配置成执行包括以下描述的各种实施例的功能在内的各种功能的任何可编程微处理器、微型计算机或一个或多个多处理器芯片。在一些移动接收机设备中,可提供多个处理器2101,诸如一个处理器专用于无线通信功能并且一个处理器专用于运行其他应用。通常,在软件应用被访问并被加载到处理器2001、2101中之前,这些软件应用可被存储在内部存储器2002、2102、2103中。处理器2001、2101可包括足以存储应用软件指令的内部存储器。

[0135] 结合本文所公开的实施例描述的各种说明性逻辑、逻辑块、模块、和电路可用通用处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)或其它可编程逻辑器件、分立的门或晶体管逻辑、分立的硬件组件、或其设计成执行本文所描述功能的任何组合来实现或执行。通用处理器可以是微处理器,但在替换方案中,处理器可以是任何常规的处理器、控制器、微控制器、或状态机。处理器还可以被实现为计算设备的组合,例如DSP与微处理器的组合、多个微处理器、与DSP核心协作的一个或多个微处理器、或任何其他此类配置。此外,至少一个处理器可包括可作用于执行以上描述的一个或多个步骤和/或

动作的一个或多个模块。

[0136] 此外,结合本文中所公开的方面描述的方法或算法的步骤和/或动作可直接在硬件中、在由处理器执行的软件模块中、或在这两者的组合中实施。软件模块可驻留在RAM存储器、闪存、ROM存储器、EPROM存储器、EEPROM存储器、寄存器、硬盘、可移动盘、CD-ROM、或本领域中所知的任何其他形式的存储介质中。示例性存储介质可被耦合到处理器以使得该处理器能从/向该存储介质读写信息。在替换方案中,存储介质可以被整合到处理器。另外,在一些方面,处理器和存储介质可驻留在ASIC中。另外,ASIC可驻留在用户终端中。在替换方案中,处理器和存储介质可作为分立组件驻留在用户终端中。另外,在一些方面,方法或算法的步骤和/或动作可作为代码和/或指令之一或其任何组合或集合驻留在可被纳入到计算机程序产品中的机器可读介质和/或计算机可读介质上。

[0137] 在一个或更多个方面中,所描述的功能可在硬件、软件、固件或其任何组合中实现。如果在软件中实现,则这些功能可作为一条或多条指令或代码存储或传送在非瞬态计算机可读介质上。非瞬态计算机可读介质包括任何形式的、促成计算机程序从一个地方转移至另一个地方的计算机存储介质。非瞬态存储介质可以是能被计算机访问的任何可用介质。作为示例而非限定,此类非瞬态计算机可读介质可以包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储、磁盘存储或其它磁存储设备、或者可用以携带或者存储指令或数据结构形式的期望程序代码且可由计算机访问的任何其它介质。如本文中所使用的盘和碟包括压缩碟(CD)、激光碟、光碟、数字多用碟(DVD)、软盘和蓝光碟,其中盘(disk)往往以磁的方式再现数据,而碟(disc)往往用激光以光学方式再现数据。以上的组合也应被包括在非瞬态计算机可读介质的范围内。

[0138] 尽管前面的公开讨论了解说性的方面和/或实施例,但是应当注意,在其中可作出各种变更和改动而不会脱离所描述的这些方面和/或实施例的如由所附权利要求定义的范围。此外,尽管所描述的方面和/或实施例的要素可能是以单数来描述或主张权利的,但是复数也是已构想了的,除非显式地声明了限定于单数。另外,任何方面和/或实施例的全部或部分可与任何其他方面和/或实施例的全部或部分联用,除非另外声明。

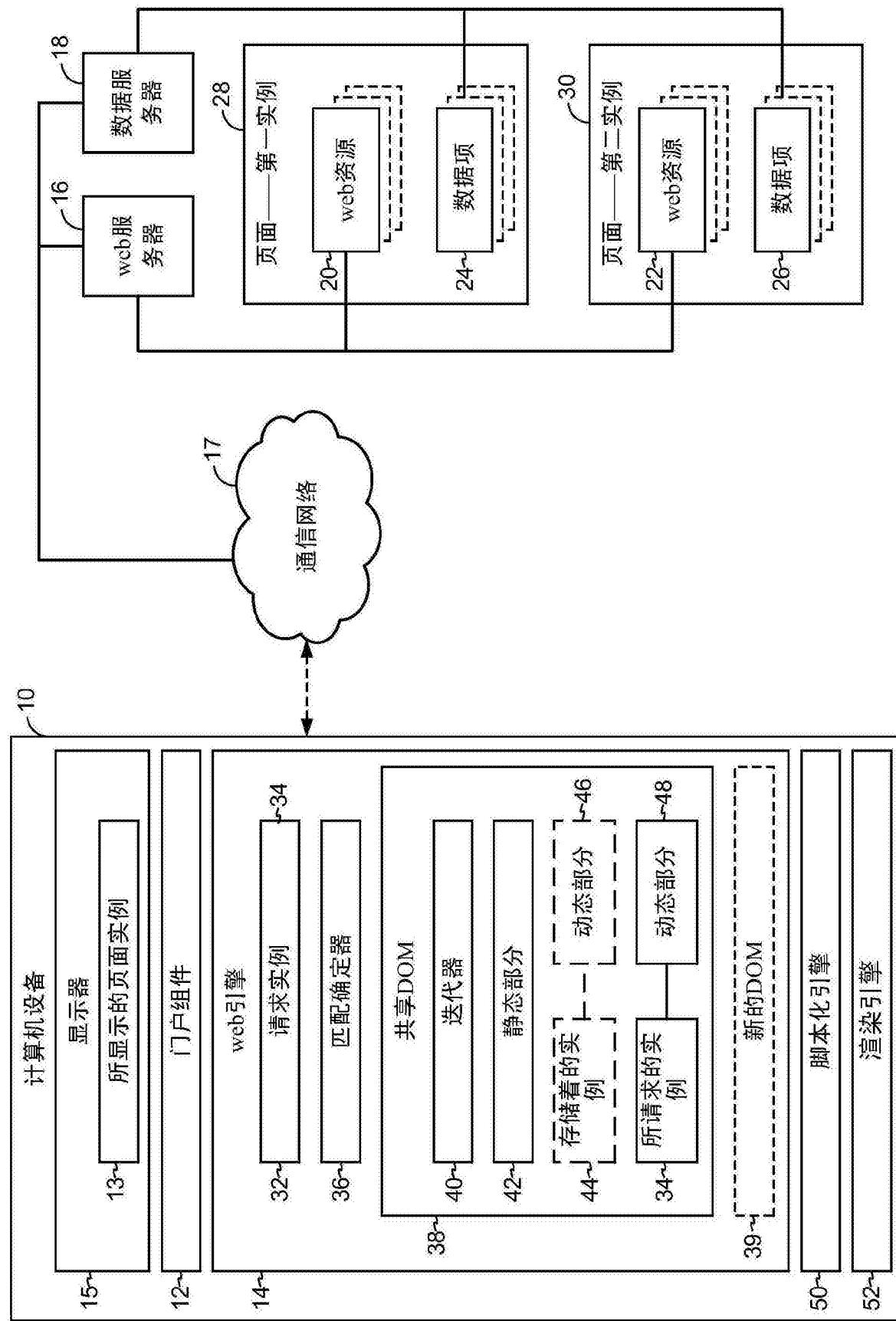


图 1

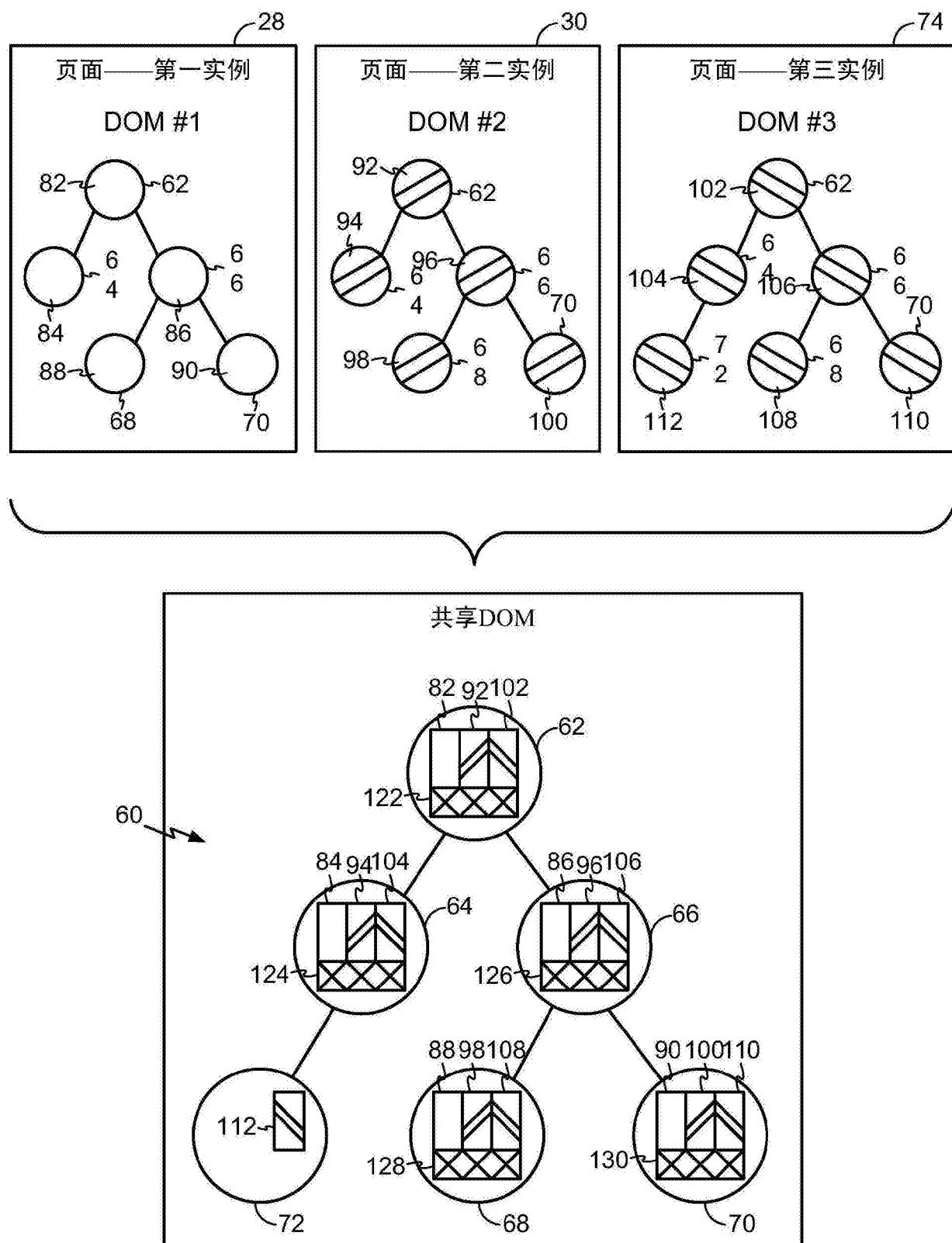


图2

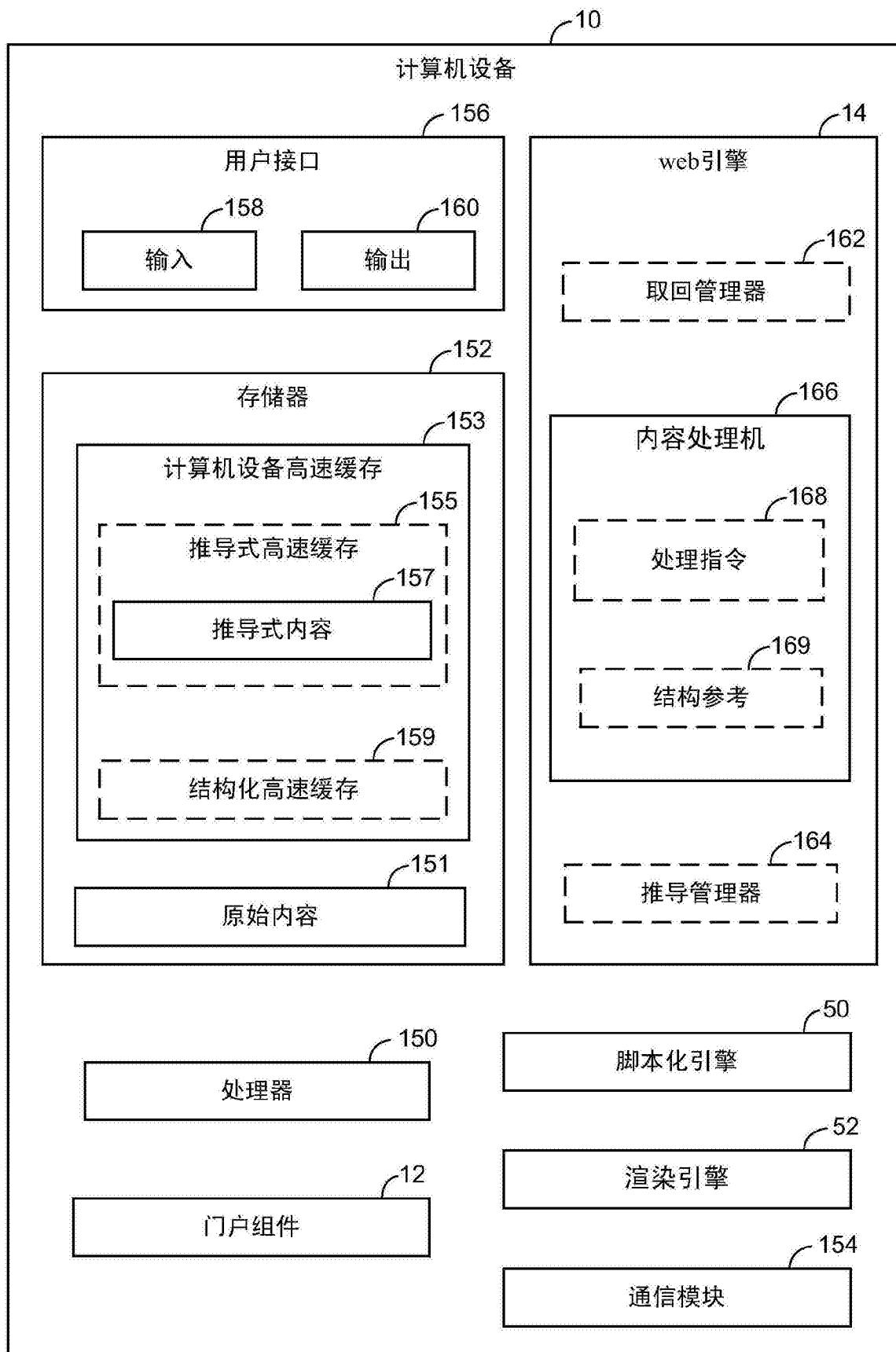


图3

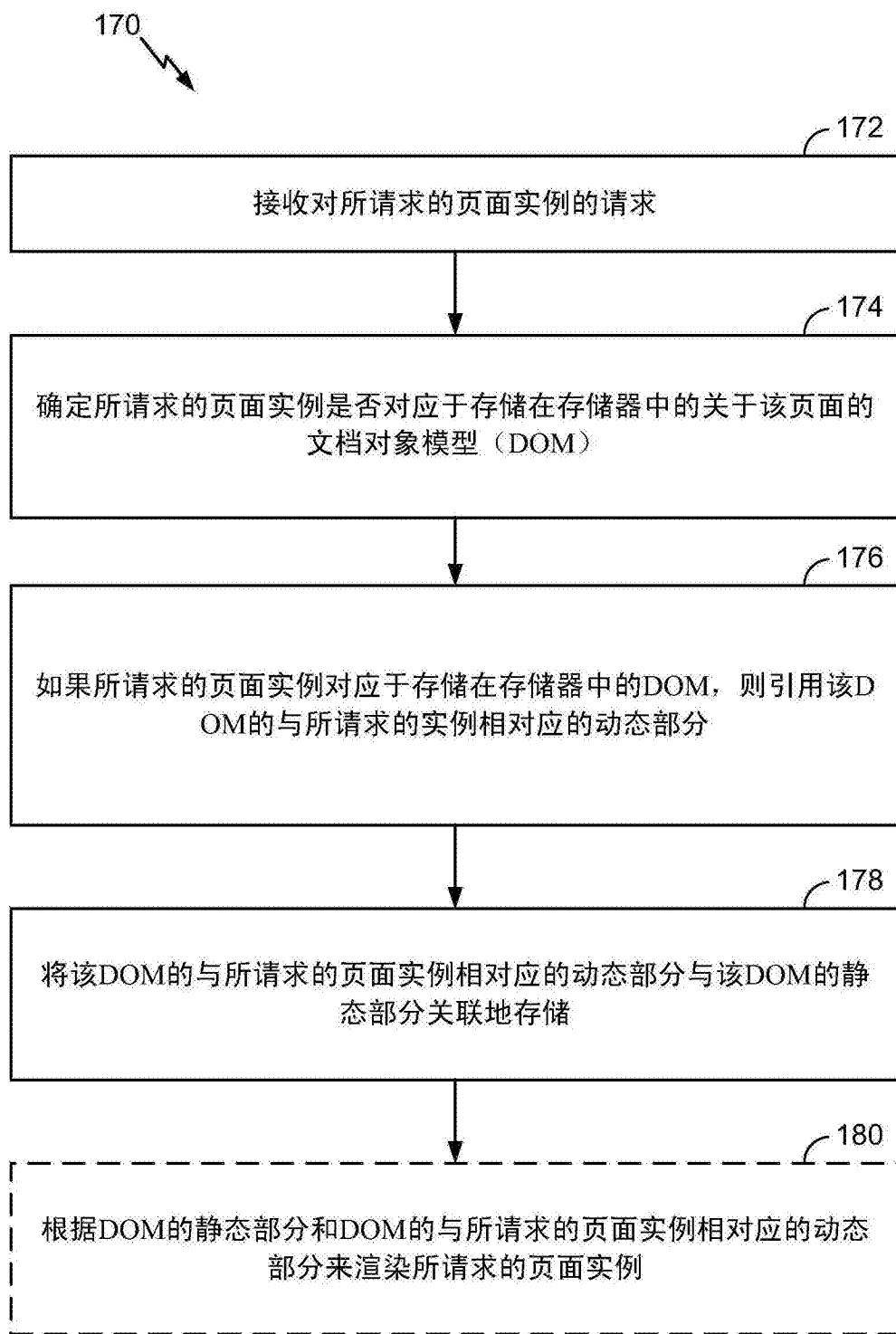


图4

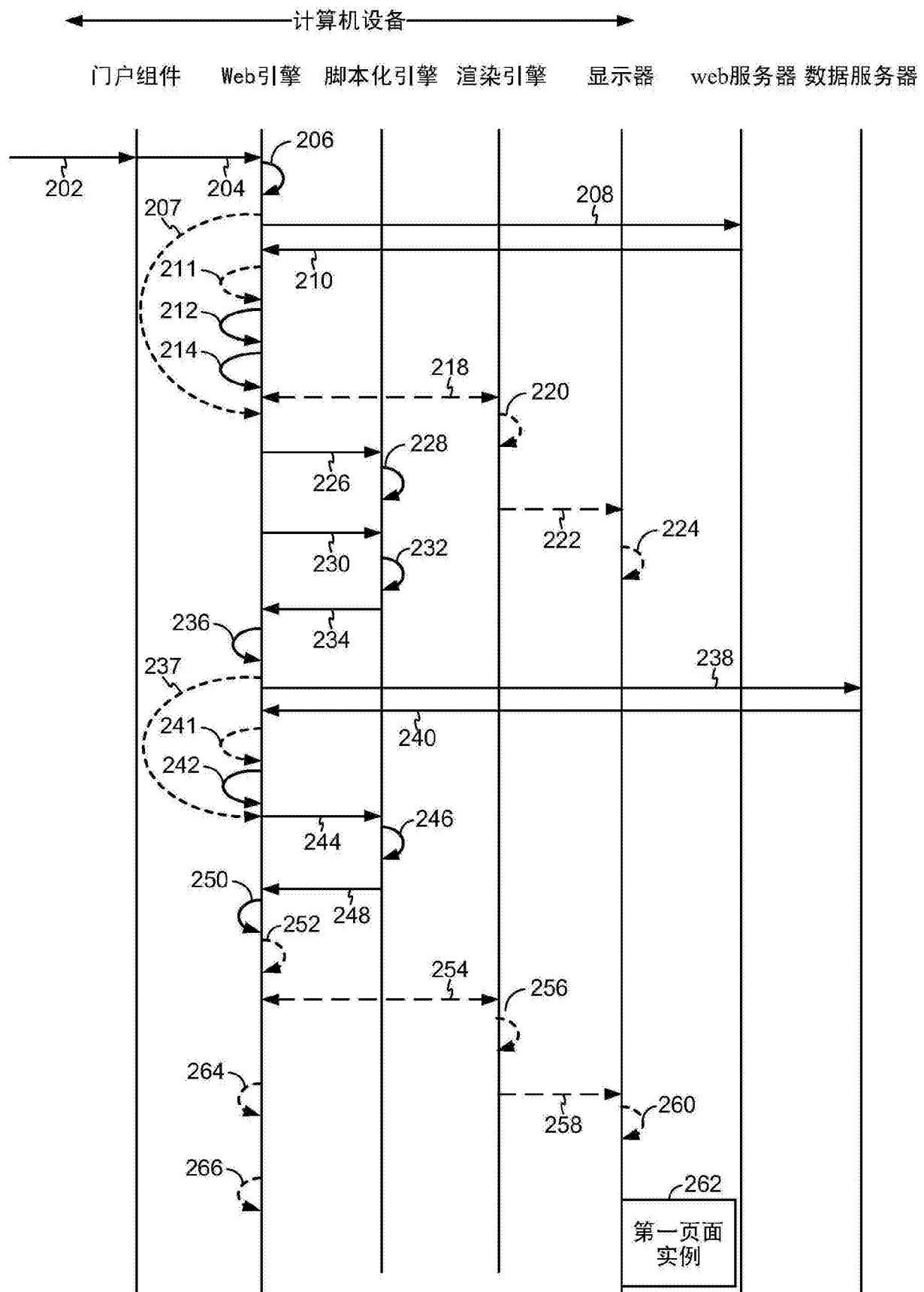


图5

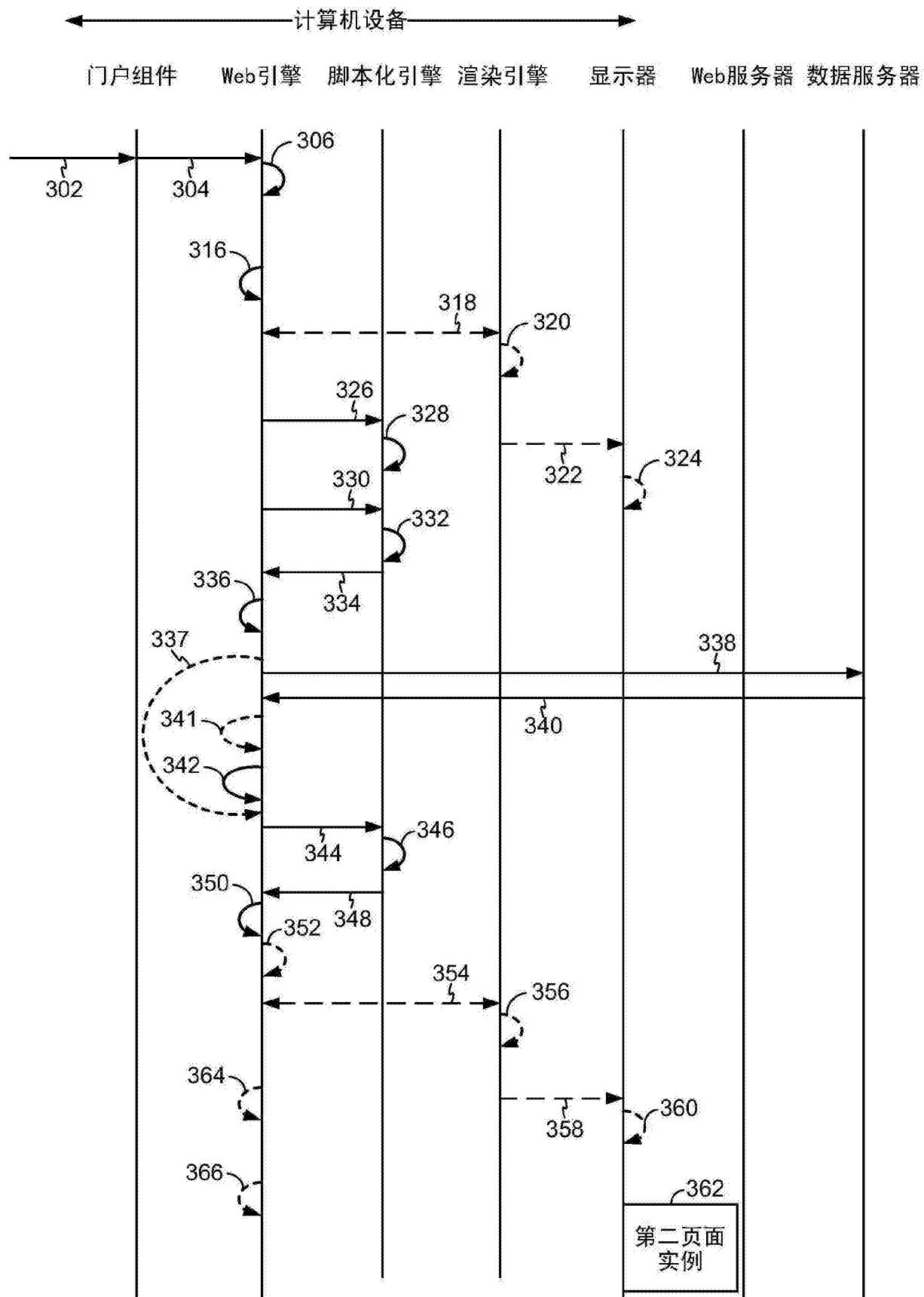


图6

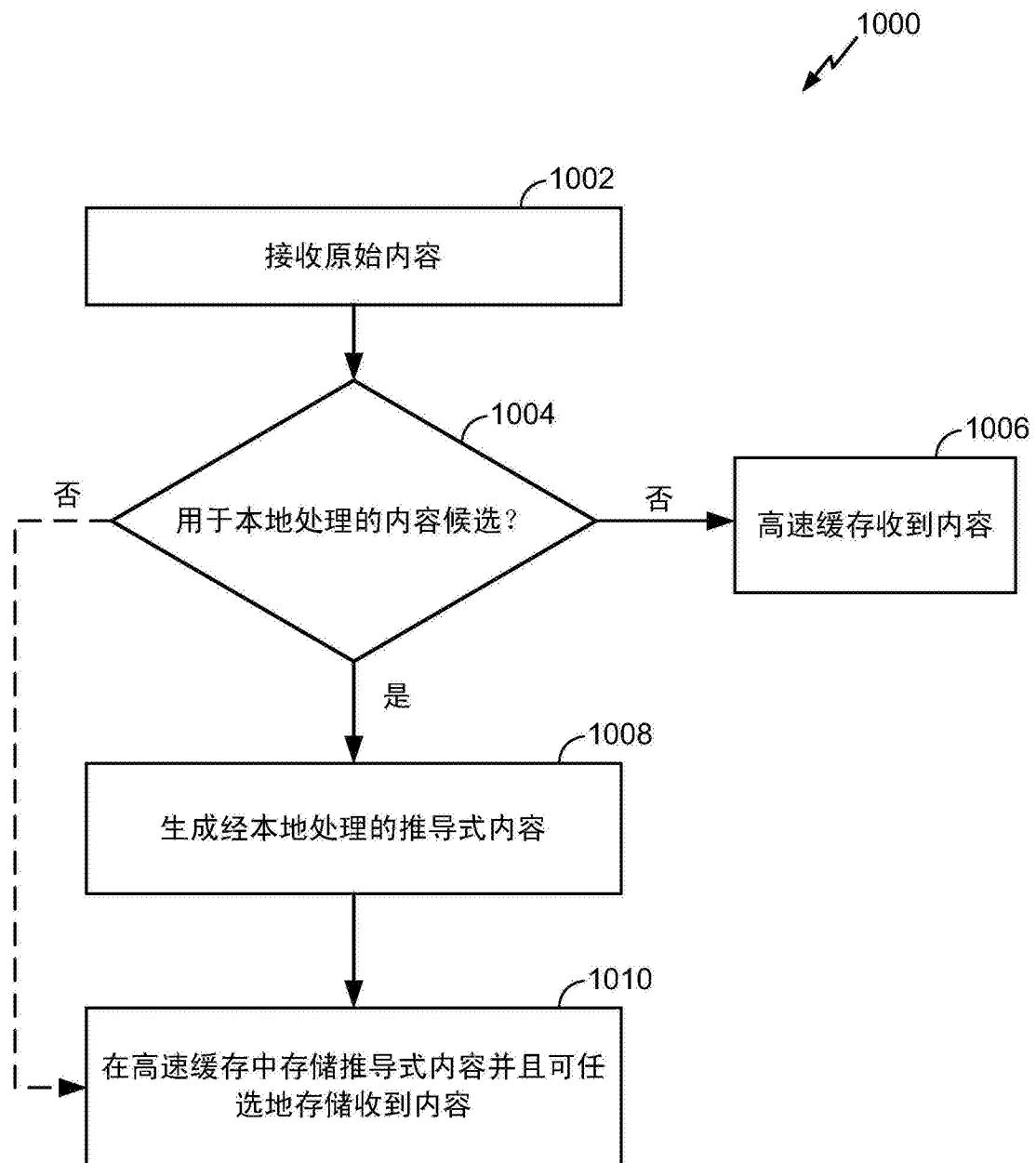


图7

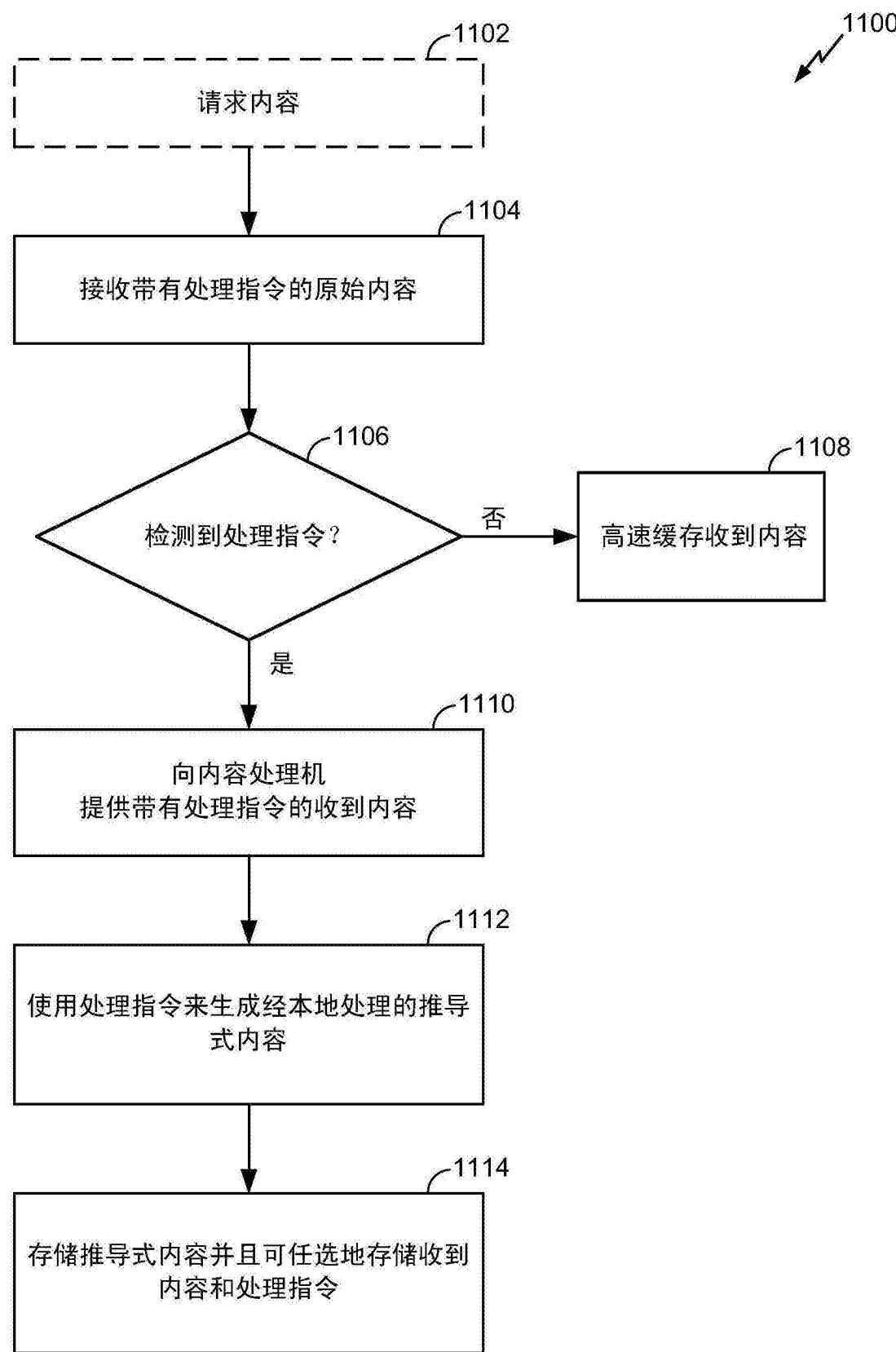


图8

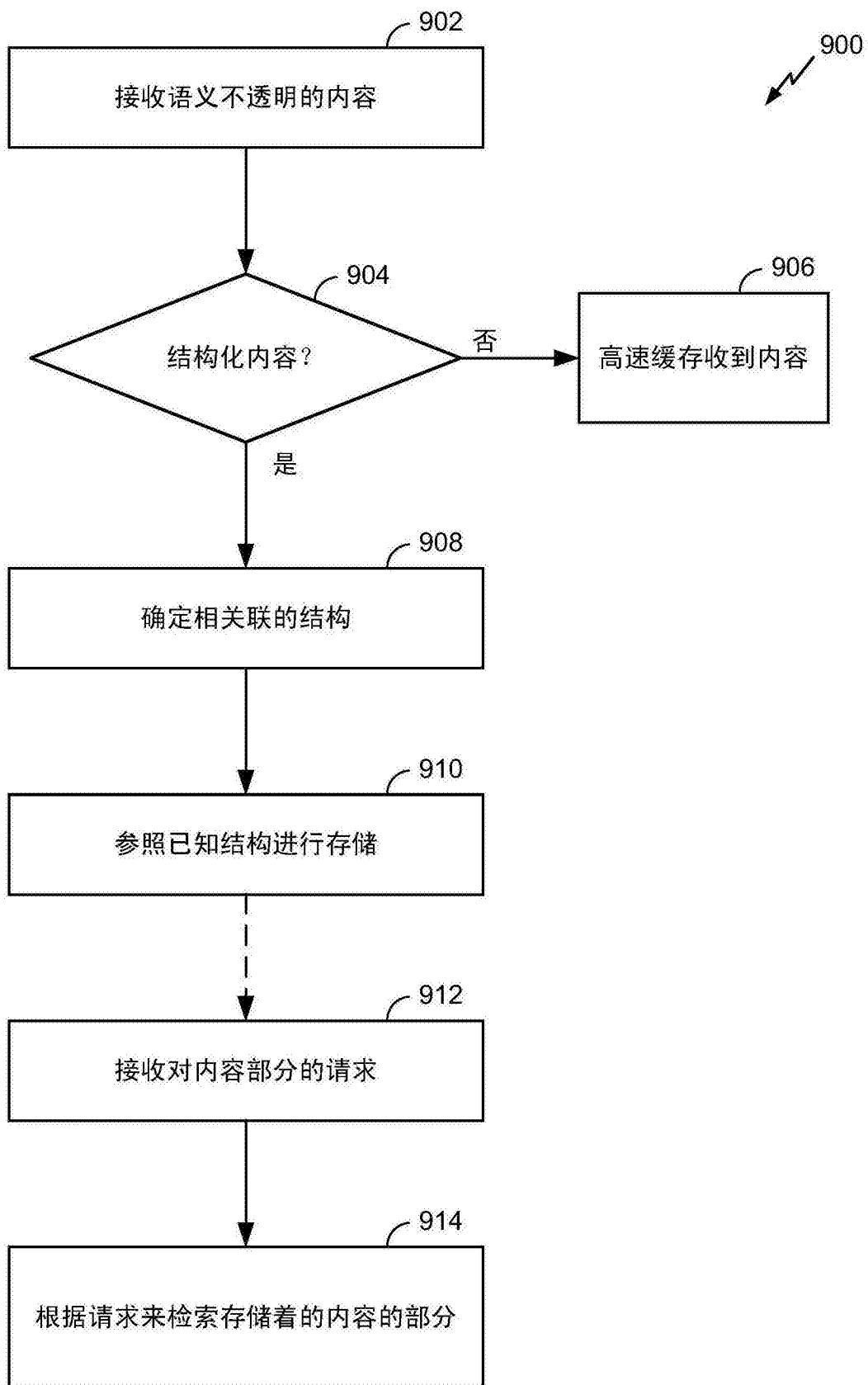


图9

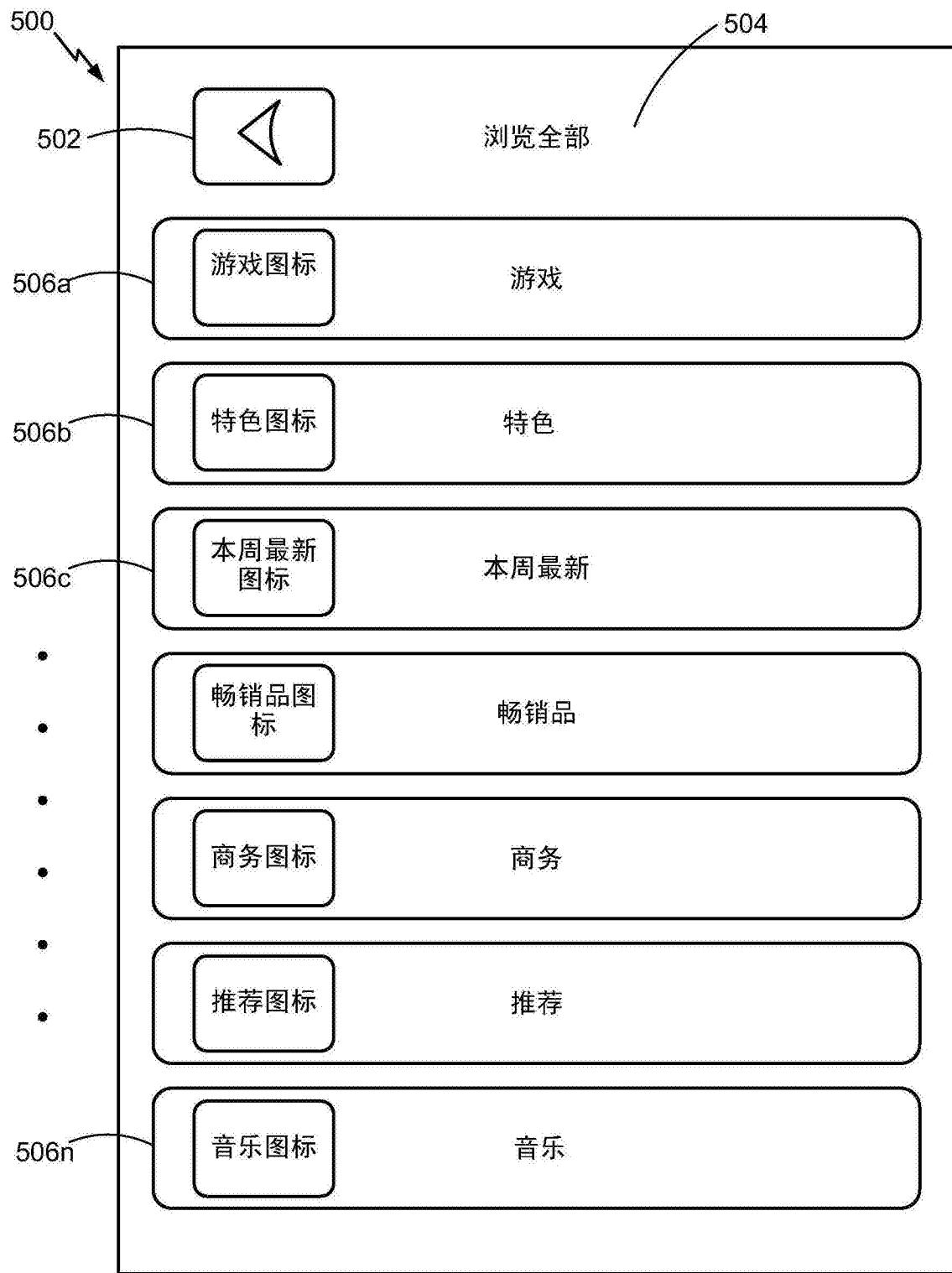


图10

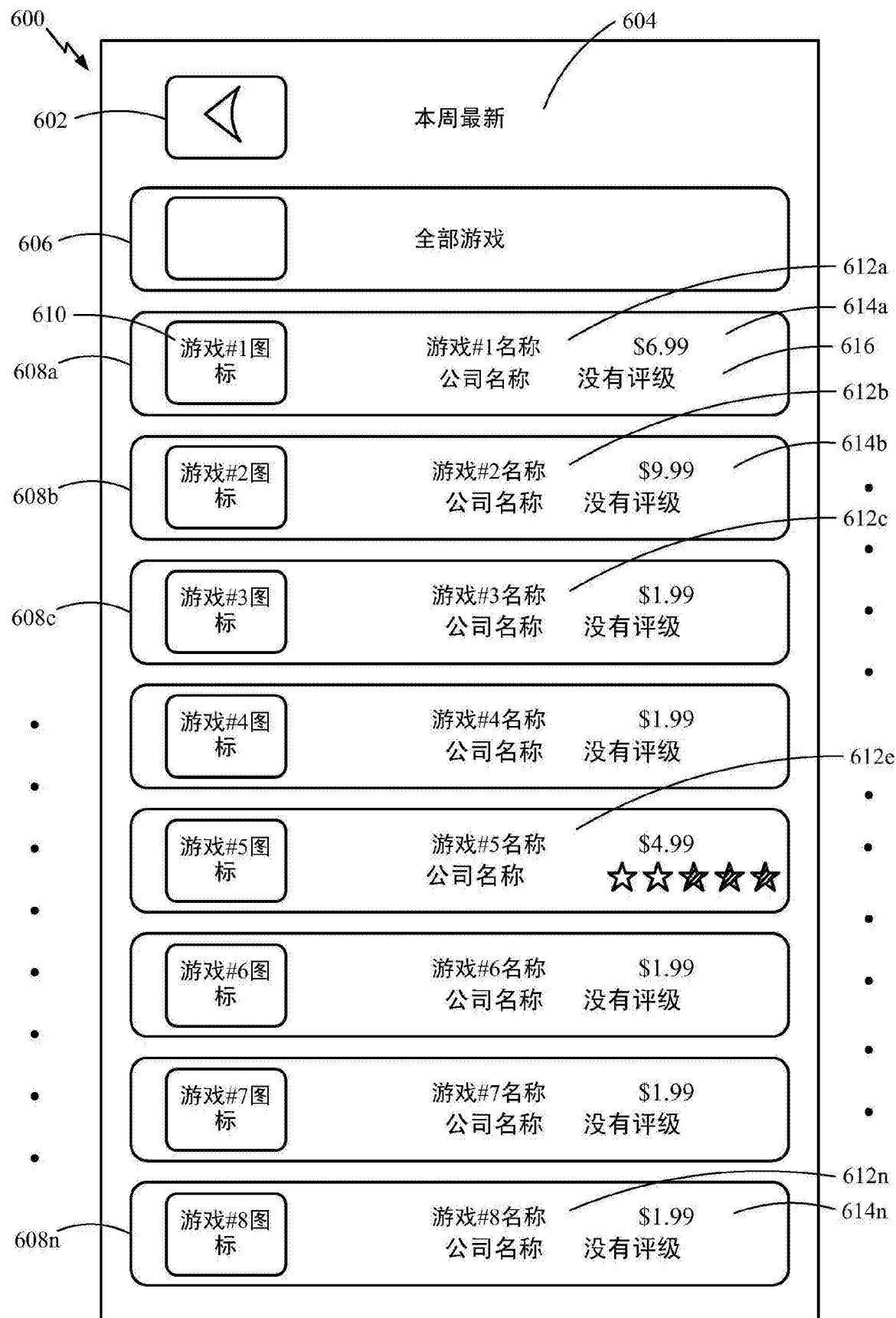


图11

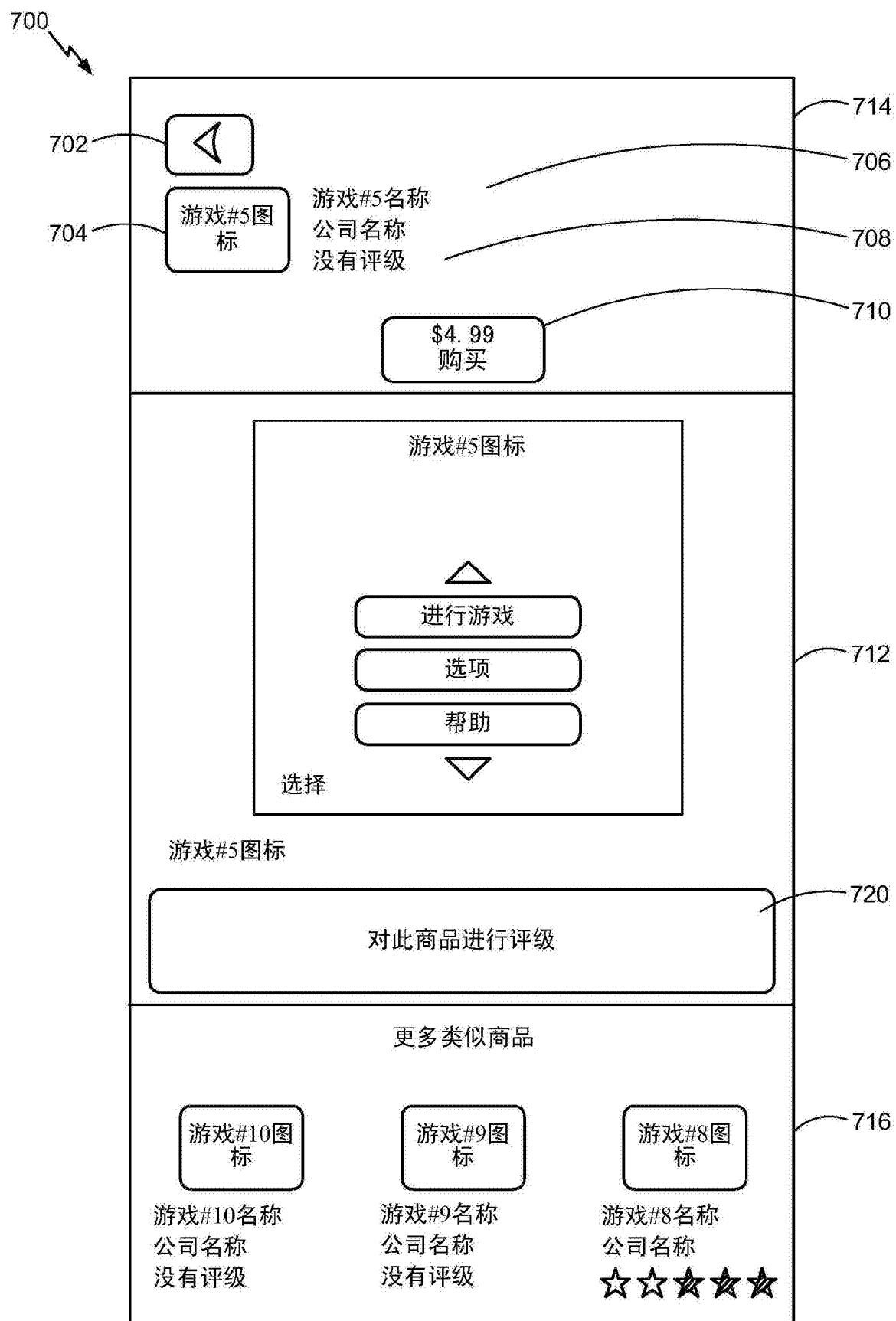


图12

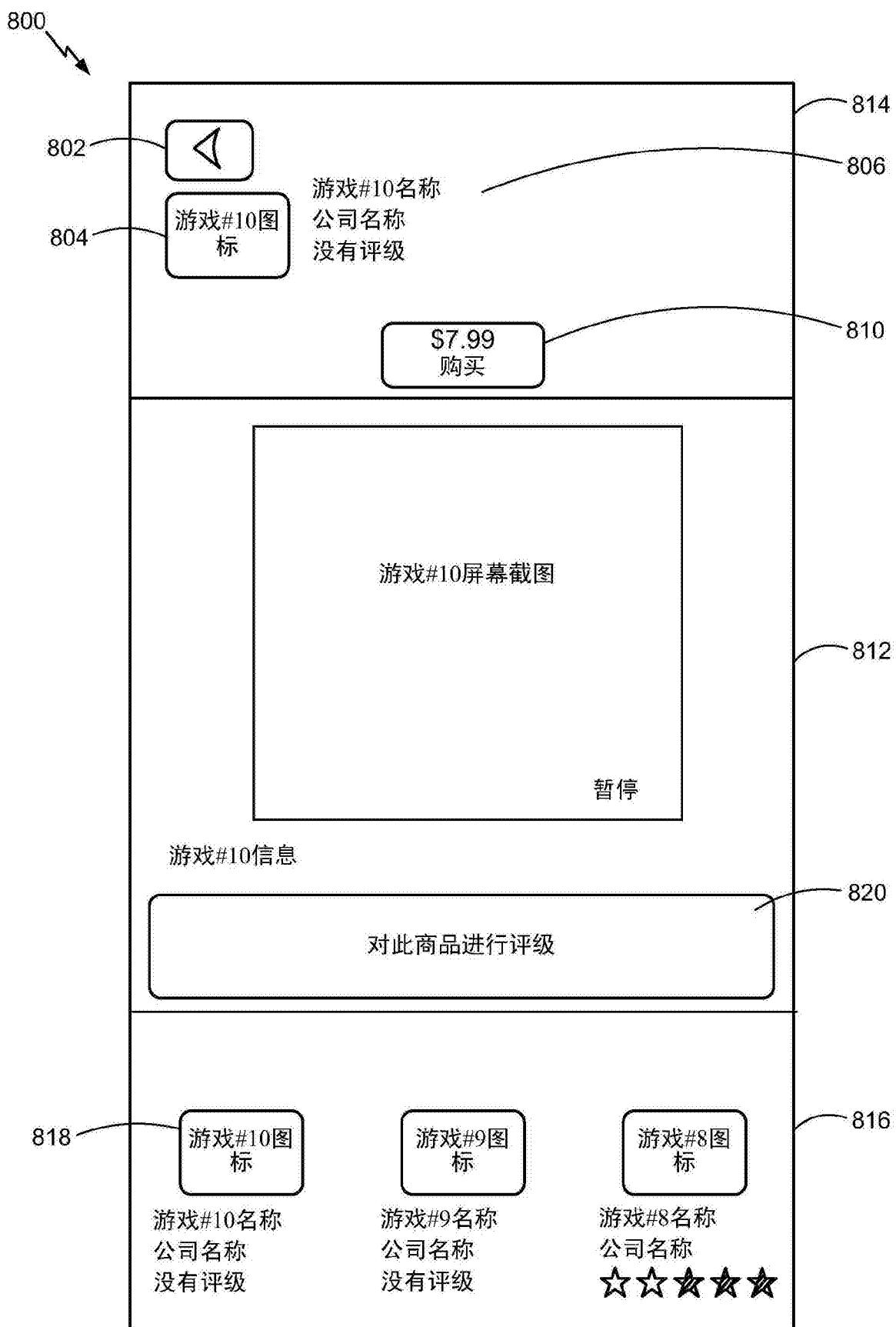


图13

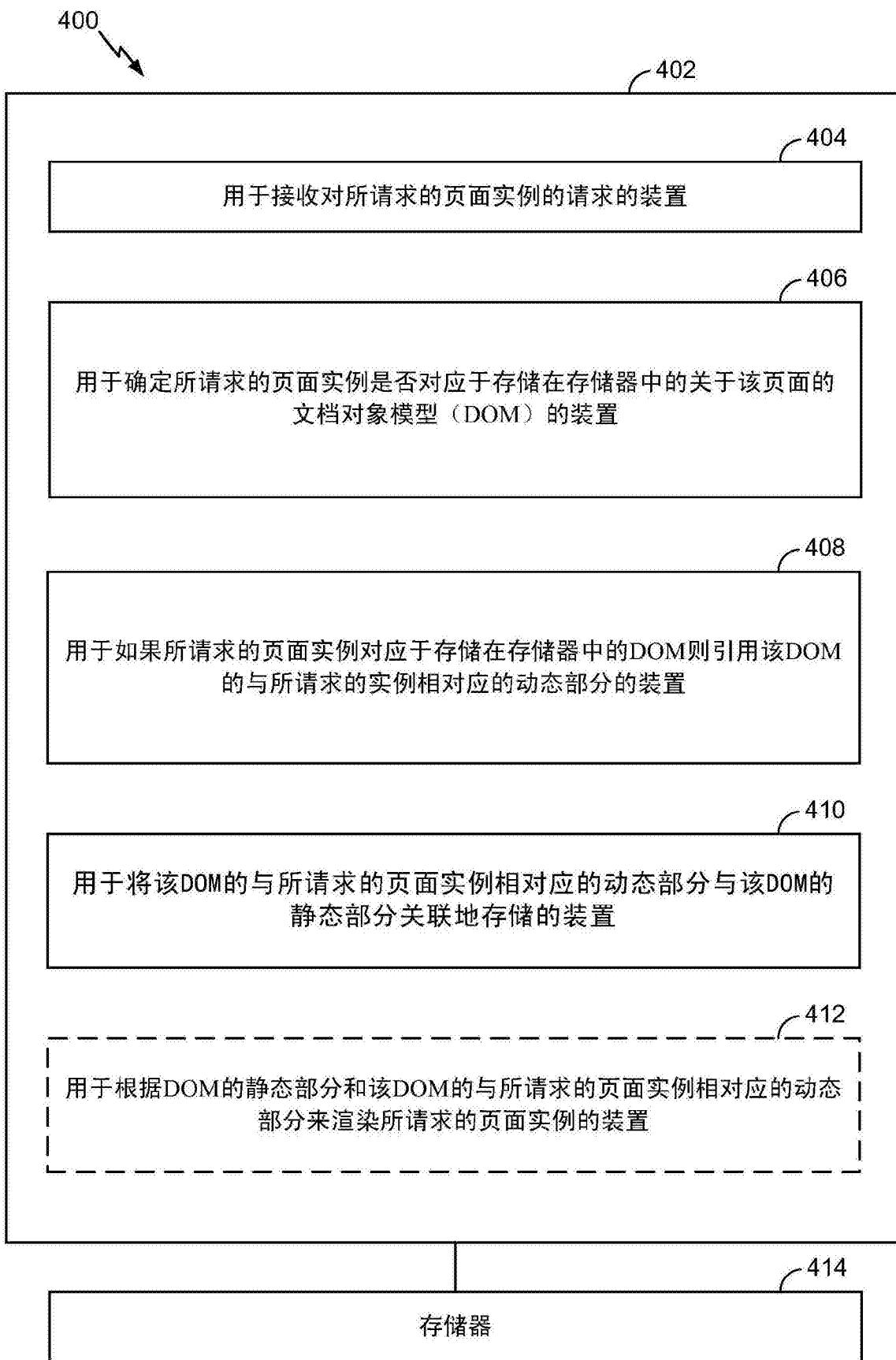


图14

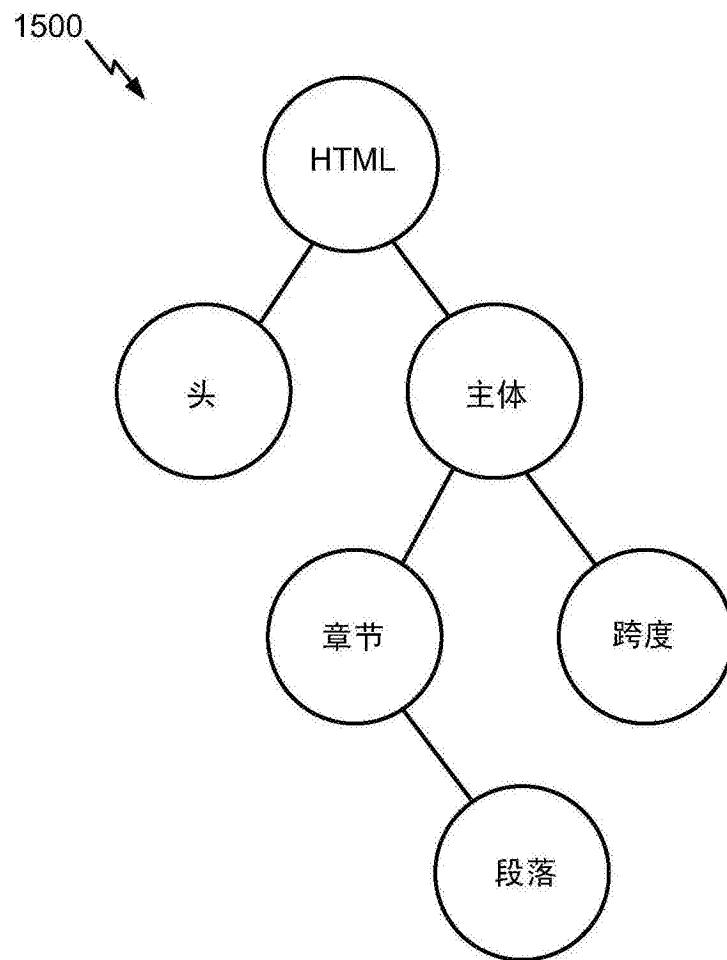


图15

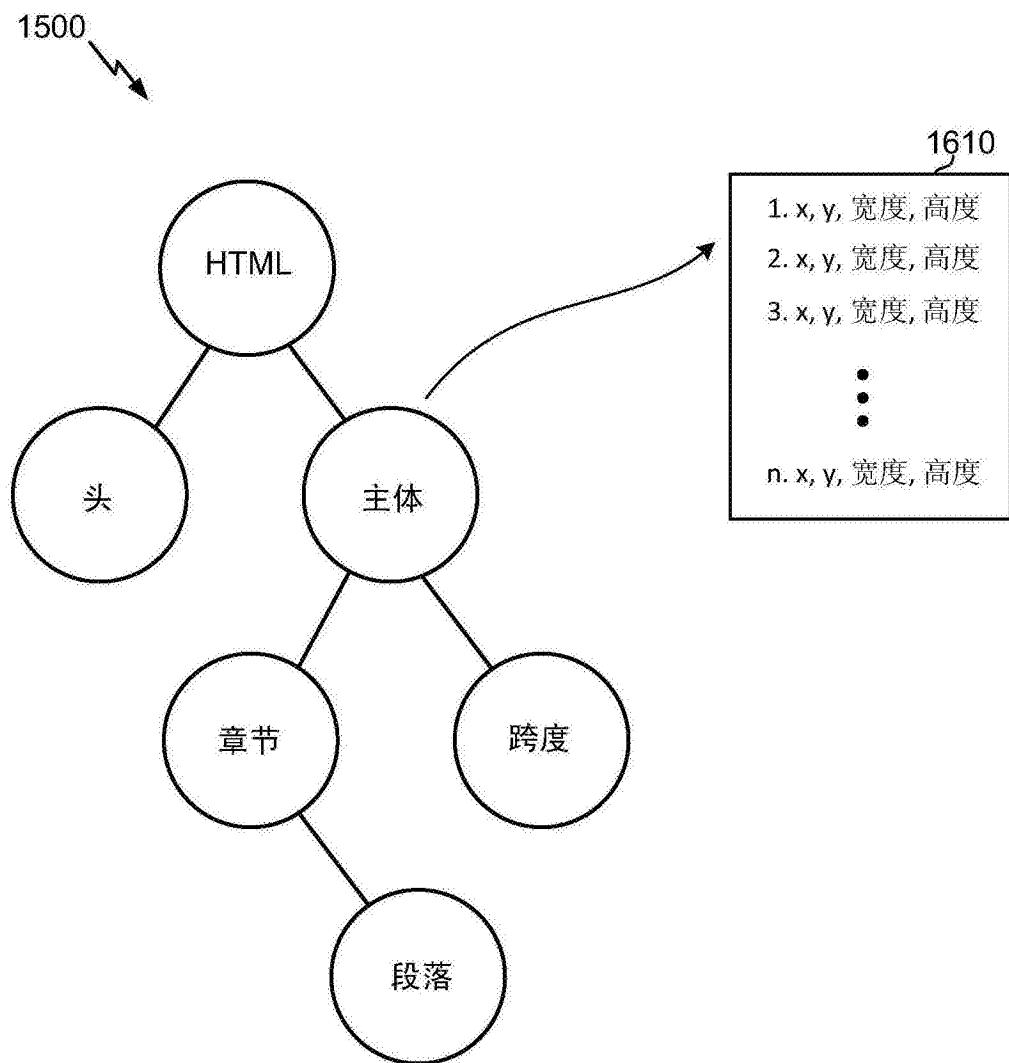


图16

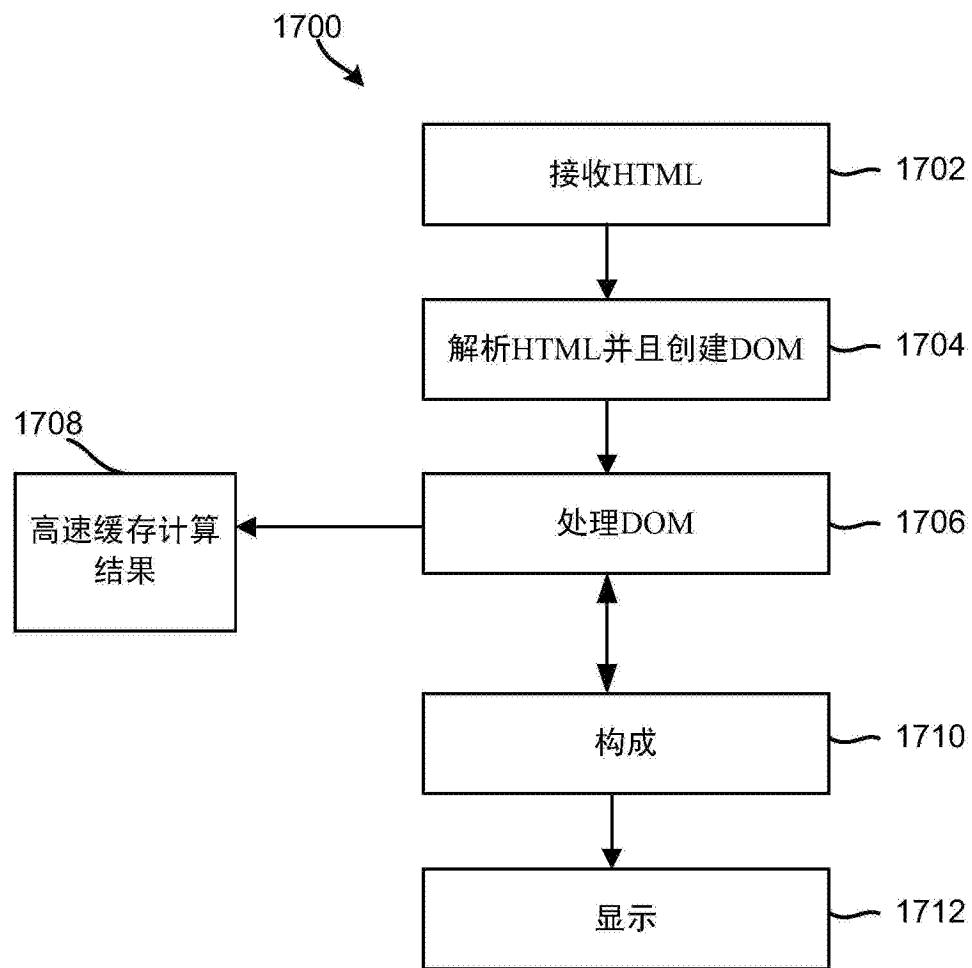


图17

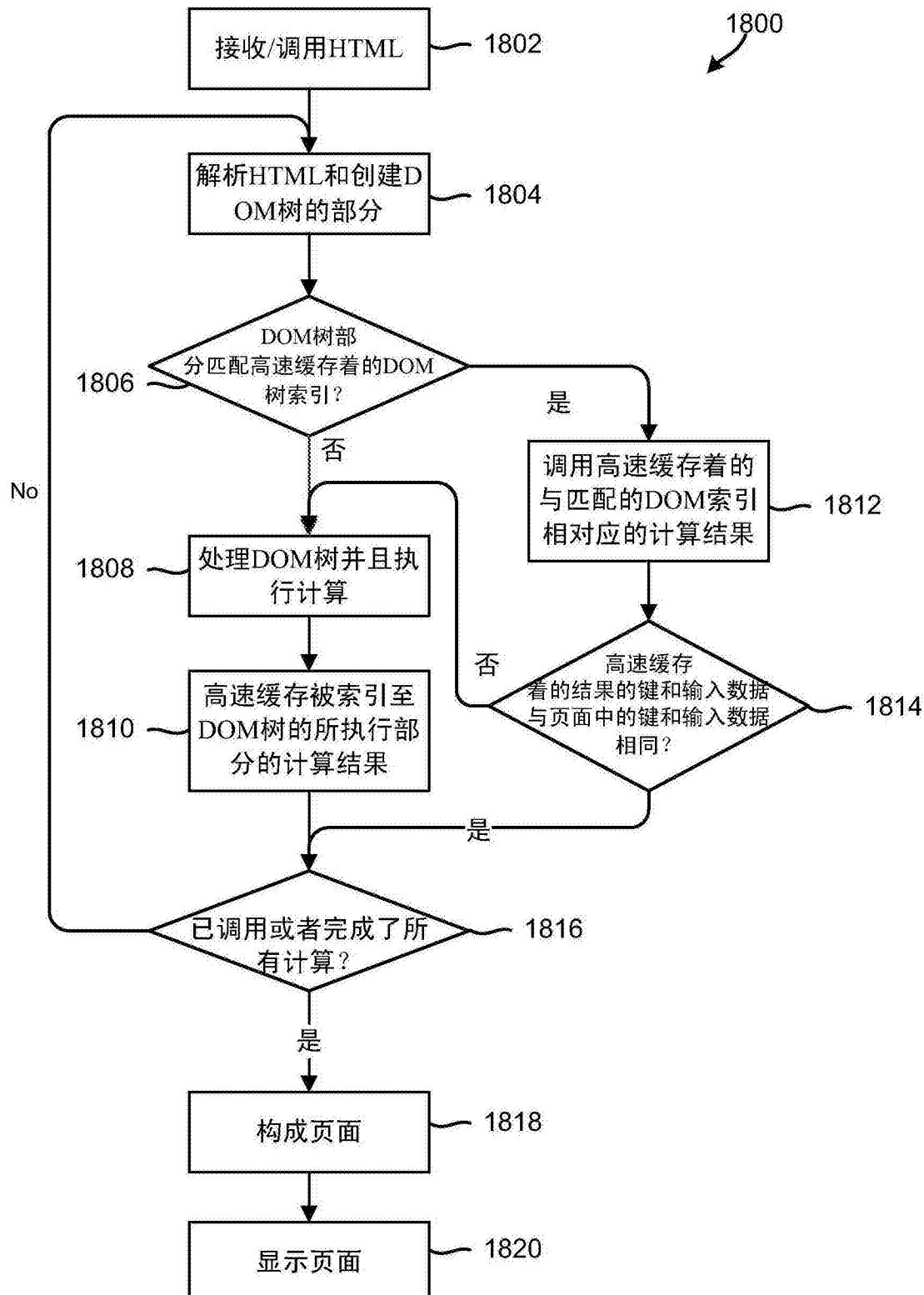


图18

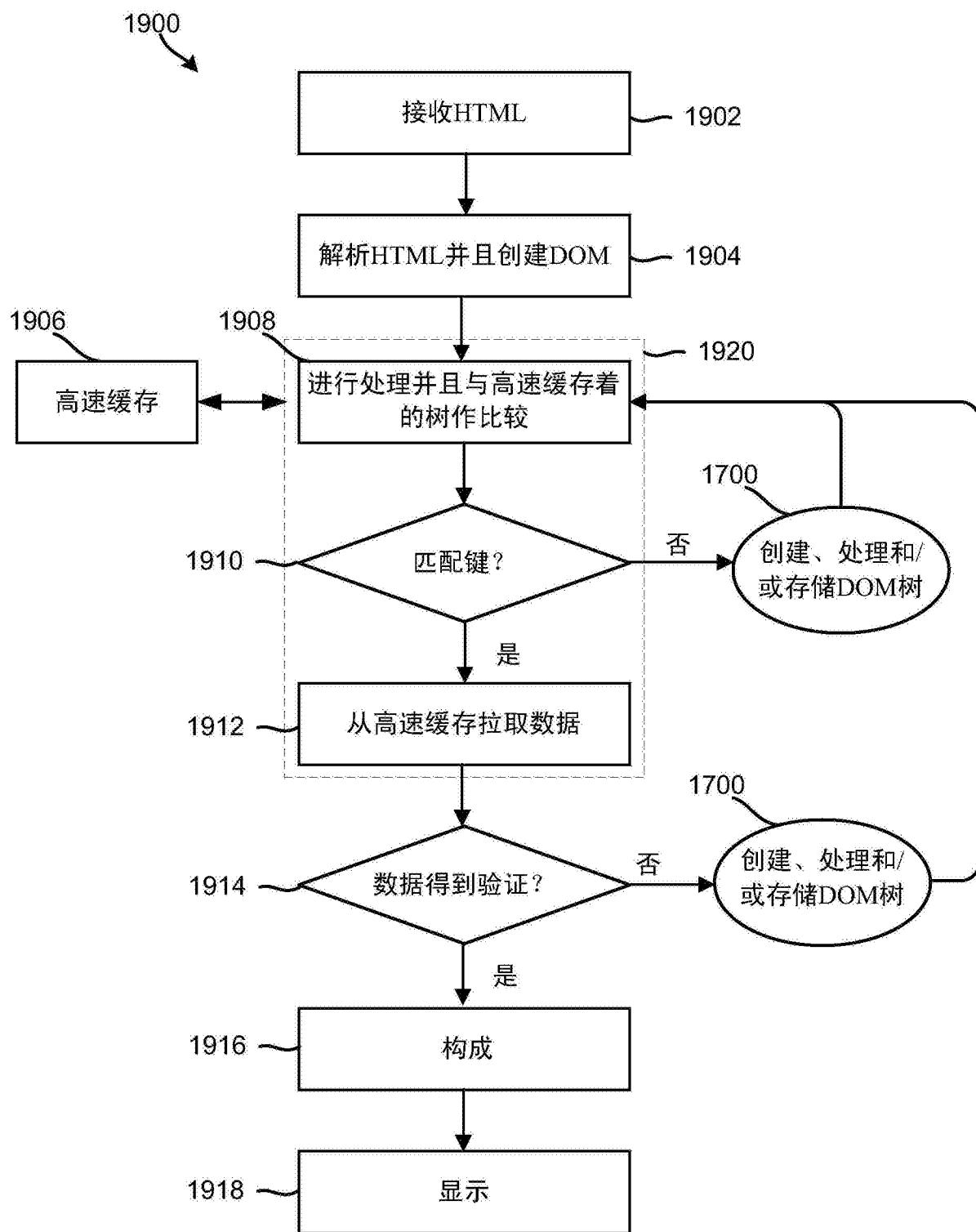


图19

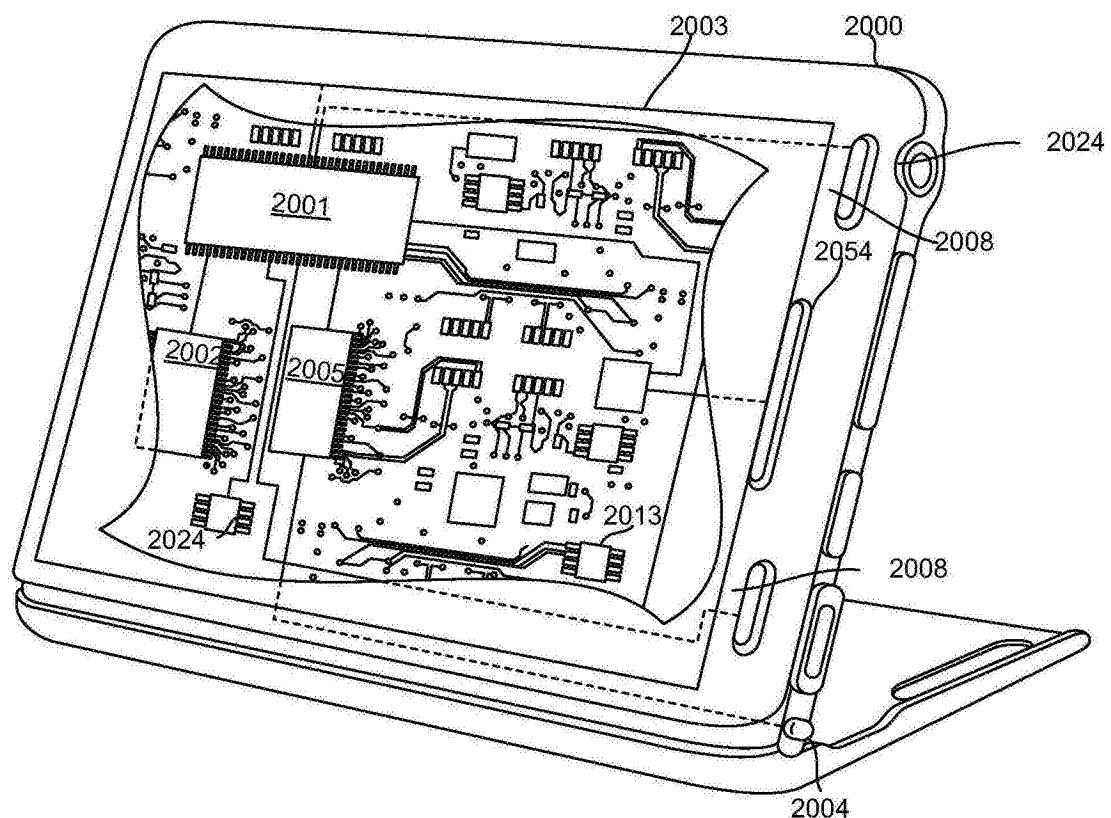


图20

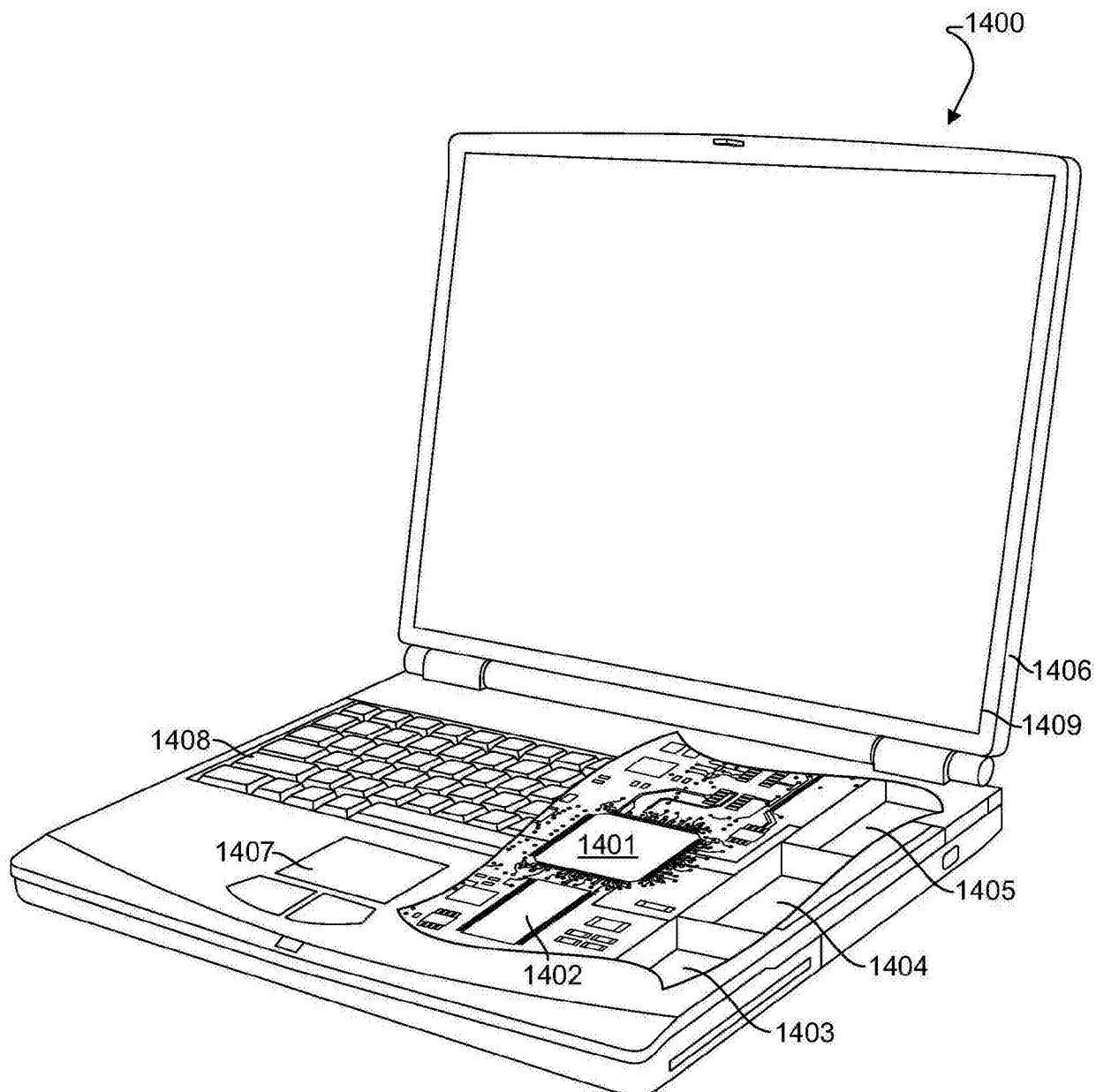


图21