



(12)发明专利

(10)授权公告号 CN 106662986 B

(45)授权公告日 2019.06.21

(21)申请号 201480079987.9

周小博

(22)申请日 2014.06.26

(74)专利代理机构 中原信达知识产权代理有限公司 11219

(65)同一申请的已公布的文献号

申请公布号 CN 106662986 A

代理人 李宝泉 周亚荣

(43)申请公布日 2017.05.10

(51)Int.Cl.

G06F 3/14(2006.01)

(85)PCT国际申请进入国家阶段日

2016.12.19

(56)对比文件

(86)PCT国际申请的申请数据

PCT/CN2014/080801 2014.06.26

CN 102917067 A, 2013.02.06,

US 2011055314 A1, 2011.03.03,

US 7054917 B1, 2006.05.30,

US 2003229677 A1, 2003.12.11,

WO 03012576 A2, 2003.02.13,

US 2005055644 A1, 2005.03.10,

(87)PCT国际申请的公布数据

W02015/196405 EN 2015.12.30

(73)专利权人 谷歌有限责任公司

地址 美国加利福尼亚州

审查员 王晨霞

(72)发明人 迈克尔·伊斯特汉 孟必平 许晖

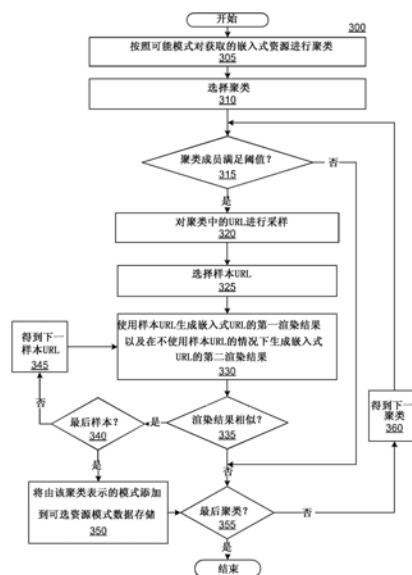
权利要求书2页 说明书12页 附图7页

(54)发明名称

优化的浏览器渲染过程

(57)摘要

实现方式通过识别内容中性嵌入式项目并且在不获取内容中性项目的情况下渲染网页来优化浏览器渲染过程。示例方法包括识别存储在获取记录中的多个URL共有的URL模式并且从所述多个URL选择URL的样本。该方法还包括针对所述样本中的每个URL,使用该URL的内容生成第一渲染结果并且在不使用该URL的内容的情况下生成第二渲染结果,通过比较第一渲染结果与第二渲染结果来计算该URL的相似度分值,从而确定该URL是否是可选的,当相似度分值大于相似度阈值时确定该URL是可选的。所述方法还可包括当所述样本中的大多数URL是可选的时,将所述URL模式存储在可选资源模式的数据存储中。



1. 一种计算机系统,包括:
至少一个处理器;以及
存储指令的存储器,所述指令在由所述至少一个处理器执行时使得所述系统:
识别存储在获取记录中的多个URL共有的URL模式;
从所述多个URL中选择URL的样本;
针对所述样本中的每个URL,通过以下来确定该URL是否是可选的:
使用该URL的内容来生成该URL的嵌入器的第一渲染结果;
在不使用该URL的内容的情况下生成所述嵌入器的第二渲染结果,以及
通过比较所述第一渲染结果与所述第二渲染结果来计算该URL的相似度分值;
当所述相似度分值大于相似度阈值时确定该URL是可选的;以及
当所述样本中的预定量的URL是可选的时,将所述URL模式存储在可选资源模式的数据存储中。
2. 根据权利要求1所述的系统,所述指令进一步包括指令,该指令在由所述至少一个处理器执行时使得所述系统:
接收对所请求的嵌入式资源的内容的请求;
确定所请求的嵌入式资源是否与所述数据存储中的模式匹配;以及
当所请求的嵌入式资源与所述模式匹配时,返回所请求的嵌入式资源是可选的指示。
3. 根据权利要求2所述的系统,其中,所述指示是未找到URL错误。
4. 根据权利要求1所述的系统,其中,所述URL的量等于所述样本中的URL的量。
5. 根据权利要求1所述的系统,其中,识别存储在所述获取记录中的所述多个URL共有的所述URL模式包括:
针对存储在所述获取记录中的相应URL,通过从该URL移除查询串的至少一部分来生成组URL;以及
按照组URL对URL进行聚类。
6. 根据权利要求1所述的系统,其中,识别存储在所述获取记录中的所述多个URL共有的所述URL模式包括:
针对存储在所述获取记录中的相应URL,通过从该URL移除查询串的至少一部分来生成组URL;
按照组URL对URL进行聚类;
选择具有最高数量的成员的聚类的组URL作为所述URL模式。
7. 根据权利要求1所述的系统,其中,计算所述相似度分值包括:
确定所述第一渲染结果的DOM树和所述第二渲染结果的DOM树的最长共有序列;以及
使用所述最长共有序列来确定所述相似度分值。
8. 一种用于优化浏览器渲染的方法,包括:
使用至少一个处理器识别存储在获取记录中的多个URL共有的URL模式;
使用所述至少一个处理器从所述多个URL选择URL的样本;
针对所述样本中的每个URL,通过以下来确定该URL是否是可选的:
使用该URL的内容生成该URL的嵌入器的第一渲染结果并且在不使用该URL的内容的情况下生成所述嵌入器的第二渲染结果,以及

通过比较所述第一渲染结果与所述第二渲染结果来计算该URL的相似度分值,当所述相似度分值大于相似度阈值时该URL是可选的;以及

当所述样本中的预定量的URL是可选的时,将所述URL模式存储在可选资源模式的数据存储中。

9. 根据权利要求8所述的方法,还包括:

接收对所请求的嵌入式资源的内容的请求;

确定所请求的嵌入式资源是否与所述数据存储中的模式匹配;以及

当所请求的嵌入式资源与所述模式匹配时,返回所请求的嵌入式资源是可选的指示。

10. 根据权利要求8所述的方法,还包括:在确定所述样本中的URL是否是可选的之前,确定由所述多个URL表示的量超过大小阈值。

11. 根据权利要求8所述的方法,还包括:当所述样本中的所有URL均是可选的时,将所述URL模式存储在可选资源模式的所述数据存储中。

12. 根据权利要求8所述的方法,其中,识别所述获取记录中的所述多个URL共有的所述URL模式包括:

针对所述获取记录中的相应URL,通过从该URL移除查询串的至少一部分来生成所述获取记录中的该URL的组URL;以及

按照组URL进行聚类。

13. 根据权利要求8所述的方法,其中,识别所述获取记录中的所述多个URL共有的URL模式包括:

针对所述获取记录中的相应URL,通过从该URL移除查询串的至少一部分来生成所述获取记录中的该URL的组URL;

按照组URL进行聚类;以及

当聚类中的成员的量满足阈值时选择所述聚类的所述组URL。

14. 一种用于优化浏览器渲染方法,包括:

接收渲染网页的请求;

使用至少一个处理器识别所述网页中需要获取的至少一个嵌入式资源;

通过确定所述嵌入式资源的URL与可选资源的数据存储中的模式匹配,来确定所述嵌入式资源是可选资源;以及

使用所述至少一个处理器在不获取所述嵌入式资源的内容的情况下将所述网页渲染为如同所述嵌入式资源不可用一样。

15. 根据权利要求14所述的方法,还包括:从服务接收可选资源模式的所述数据存储。

16. 根据权利要求14所述的方法,其中,确定所述嵌入式资源是可选资源包括:

通过从URL移除查询串的至少一部分来重写所述嵌入式资源的所述URL;以及

将所重写的URL与可选资源的所述数据存储中的所述模式进行匹配,其中,当所重写的URL与所述模式匹配时,所述嵌入式资源是可选的。

优化的浏览器渲染过程

背景技术

[0001] 许多网页动态地创建,这意味着由web浏览器的用户观看的最终页面至少部分地通过浏览器渲染过程来生成。诸如脚本、图像和样式表的嵌入式对象经常被嵌入网页中,并且可影响所渲染的网页的内容和外观或者可出于与生成动态内容无关的其它目的而被包括。例如,Google Analytics JavaScript代码可以是帮助跟踪网站的流量,但是不影响内容的嵌入式对象。嵌入式对象本身可包括附加嵌入式对象。类似地,一些嵌入式对象可被隐藏且对于用户而言不可见。网页可具有数以百计的嵌入式对象,并且浏览器的渲染引擎为用户渲染网页所花费的许多时间花在了等待浏览器获取(fetch)这些嵌入式资源上。

发明内容

[0002] 实现方式识别可选嵌入式资源并且允许渲染引擎跳过这些资源,这改善了渲染时间。可选资源表示可被跳过而不会以有意义或显著的方式影响所渲染的网页的内容的嵌入式对象。浏览器可在渲染过程期间跳过这样的可选嵌入式资源,导致加载时间更快。

[0003] 在一个方面,一种系统包括至少一个处理器和存储有指令的存储器,所述指令在由所述至少一个处理器执行时使得所述系统执行操作。所述操作可包括识别存储在获取记录中的多个URL共有的URL模式并且从所述多个URL中选择URL的样本。所述操作还包括针对所述样本中的每个URL,通过以下操作来确定该URL是不是可选的:使用该URL的内容渲染该URL的嵌入器的第一渲染结果;在不使用该URL的内容的情况下渲染嵌入器的第二渲染结果,以及通过比较第一渲染结果与第二渲染结果来计算该URL的相似度分值。所述方法还包括当所述相似度分值大于相似度阈值时确定该URL是可选的,并且当所述样本中的预定量的URL是可选的时,将URL模式存储在可选资源模式的数据存储中。

[0004] 本文所述的主题的一个或多个实现方式可包括下列特征中的一个或多个。例如,所述操作还可包括接收对所请求的嵌入式资源的内容的请求,确定所请求的嵌入式资源是否与数据存储中的模式匹配,并且当所请求的嵌入式资源与所述模式匹配时,返回所请求的嵌入式资源是可选的指示。在一些实现方式中,所述指示可以是未找到URL错误。作为另一示例,识别存储在获取记录中的多个URL共有的URL模式的操作可包括:针对存储在所述获取记录中的相应URL,通过从该URL移除查询串的至少一部分来生成该URL的组URL;以及按照组URL对URL进行聚类。作为另一示例,识别存储在所述获取记录中的所述多个URL共有的所述URL模式的操作可包括:针对存储在所述获取记录中的相应URL,通过从该URL移除查询串的至少一部分来生成该URL的组URL;按照组URL对URL进行聚类;选择具有最高数量的成员的聚类的组URL作为URL模式。作为另一示例,计算相似度分值的操作可包括:确定所述第一渲染结果的DOM树和所述第二渲染结果的DOM树的最长共有序列;以及使用所述最长共有序列来确定相似度分值。

[0005] 在另一方面,一种计算机实现的方法包括识别存储在获取记录中的多个URL共有的URL模式以及使用至少一个处理器从所述多个URL中选择URL的样本。所述方法还可包括针对所述样本中的每个URL,通过以下操作确定该URL是否是可选的:使用该URL的内容生成

该URL的嵌入器的第一渲染结果,在不使用该URL的内容的情况下生成嵌入器的第二渲染结果,并且通过比较所述第一渲染结果与所述第二渲染结果来计算该URL的相似度分值,当所述相似度分值大于相似度阈值时该URL是可选的。所述方法还可包括当所述样本中的大多数URL是可选的时,将所述URL模式存储在可选资源模式的数据存储中。

[0006] 本文所述的主题的一个或更多个实现方式可包括下列特征中的一个或更多个。例如,所述方法还可包括在确定所述样本中的URL是否是可选的之前,确定由所述多个URL表示的量超过大小阈值。作为另一示例,所述方法可包括当所述样本中的所有URL均是可选的时,将URL模式存储在可选资源模式的所述数据存储中。作为另一示例,识别获取记录中的所述多个URL共有的URL模式的步骤可包括针对所述获取记录中的相应URL,通过从该URL移除查询串的至少一部分来生成获取记录中的该URL的组URL,按照组URL进行聚类,并且当聚类中的成员的量满足阈值时选择所述聚类的所述组URL。

[0007] 在另一方面,一种方法包括:接收渲染网页的请求;识别所述网页中需要获取的至少一个嵌入式资源;确定所述嵌入式资源是可选资源;以及在不获取所述嵌入式资源的内容的情况下将所述网页渲染为如同所述嵌入式资源不可用一样。

[0008] 本文所述的主题的一个或更多个实现方式可包括下列特征中的一个或更多个。例如,确定所述嵌入式资源是可选资源可包括确定所述嵌入式资源的URL与可选资源的数据存储中的模式匹配。作为另一示例,所述方法还可包括从服务接收可选资源模式的数据存储,其中,确定所述嵌入式资源是可选资源的步骤包括确定所述嵌入式资源的URL与所述数据存储中的模式匹配。作为另一示例,确定所述嵌入式资源是可选资源的步骤可包括:通过从URL移除查询串的部分来重写所述嵌入式资源的所述URL;以及将所重写的URL与可选资源的数据存储中的模式进行匹配,其中,当所重写的URL与所述模式匹配时,所述嵌入式资源是可选的。作为另一示例,确定所述嵌入式资源是可选资源的步骤可包括:通过从URL移除查询串的至少一部分来重写所述嵌入式资源的URL;以及将所重写的URL与可选资源的数据存储中的模式进行匹配,其中,当所重写的URL与所述模式匹配时,所述嵌入式资源是可选的。

[0009] 在另一方面,非暂时性计算机可读介质可包括可由形成在基底中的至少一个处理器执行的指令,其使得计算机系统执行上述一个或更多个方法。

[0010] 本文所述的主题的一个或更多个实现方式可被实现为实现下列优点中的一个或更多个。作为一个示例,跳过可选嵌入式资源允许渲染引擎在不丢失内容的情况下更快速地渲染网页。浏览器不仅可跳过获取可选嵌入式资源,浏览器还跳过获取嵌入所跳过的可选嵌入式资源中的对象。所公开的实现方式因此节省了网络带宽和电池消耗。由于网页可包括数以百计的嵌入式对象,这样的优化可为显著的。

附图说明

[0011] 图1图示出根据所公开的主题的示例系统。

[0012] 图2是根据实现方式的所渲染的网页的渲染结果的框图。

[0013] 图3是图示出根据实现方式的用于识别可选资源模式的示例过程的流程图。

[0014] 图4是图示出根据实现方式的用于确定两个渲染结果是否相似的示例过程的流程图。

- [0015] 图5是图示出根据实现方式的用于提供优化的渲染的信息的示例过程的流程图。
- [0016] 图6示出可用于实现所描述的技术的计算机装置的示例。
- [0017] 图7示出可用于实现所描述的技术的分布式计算机装置的示例。

具体实施方式

[0018] 所公开的实现方式识别可选嵌入式对象或者不显著影响所渲染的网页的内容的嵌入式对象,从而允许浏览器渲染引擎在不获取可选嵌入式对象的情况下渲染这样的网页。这改善了渲染时间和获取系统的负担,这对诸如智能电话和平板的具有有限资源的计算装置而言很重要。当利用可选嵌入式对象渲染的嵌入器网页的渲染结果与在没有可选嵌入式对象的情况下渲染的嵌入器网页的渲染结果相似时,嵌入式资源可以是用于优化渲染的候选。可通过页面中的令牌的相似度、外链的相似度、页面布局的相似度等来确定相似度。在一些实现方式中,系统可使用最长共有序列测量来确定渲染结果是否相似。在一些实现方式中,当快照的主要组件的相似度分值满足相似度阈值时,快照可被认为相似。如果系统确定嵌入式资源是可选的,则系统可将嵌入式资源(例如,其统一资源定位符或URL)存储在可选嵌入式资源的列表中。在一些实现方式中,系统可聚合可选嵌入式资源URL以识别可用于识别其它可选嵌入式资源的模式。

[0019] 当浏览器开始渲染网页时,如果任何嵌入式资源为可选的,则它可请求服务。在一些实现方式中,这可作为获取请求的一部分来完成。服务可将所请求的嵌入式资源的URL与可选嵌入式对象的数据存储进行匹配(精确匹配或者与数据存储中的模式进行匹配)。如果服务发现匹配,则服务可告知浏览器哪些嵌入式资源是可选的,并且浏览器可在不获取这些嵌入式资源的情况下渲染网页。这样的优化浏览器渲染使得浏览器渲染过程的资源效率更高。在一些实现方式中,可选嵌入式资源数据存储可在浏览器类型(例如,移动浏览器和桌面浏览器)之间区分。

[0020] 图1是根据示例实现方式的系统的框图。系统100可用于识别可选嵌入式资源和可选资源模式,并且使用所述模式来优化浏览器渲染过程。系统100可包括采取若干不同装置的形式计算装置。例如,系统100可包括作为如图6中所描述的计算机装置600以及如图7中所描述的计算机装置700的示例的计算装置。

[0021] 系统100可包括批渲染系统110。批渲染系统110可以是标准服务器、一组这样的服务器、客户端-服务器系统、或者机架式服务器系统。另外,批渲染系统110可被实现于个人计算机中。批渲染系统110可包括被配置为执行存储在存储器中的一个或多个机器可执行指令或者软件片段、固件或其组合的一个或多个处理器。例如,批渲染系统110可包括渲染引擎120、获取服务122、web爬取引擎124、和可选资源识别引擎126。批渲染系统110可按照批处理模式生成渲染的网页(例如,作为索引过程的一部分),并且在此过程中生成获取记录132,批渲染系统110可使用记录132来生成可选资源模式130。

[0022] 向下传输(downstream)过程可请求批渲染系统110以批处理模式渲染网页。在一些实现方式中,批渲染系统110可包括诸如可选资源识别引擎126或者索引引擎(未示出)的向下传输过程。在一些实现方式中,向下传输过程可在不同于批渲染系统110的计算装置上执行。例如,向下传输过程可以是经由远程过程调用向批渲染系统发送请求的索引引擎或广告平台。向下传输过程可请求批渲染引擎120之一生成特定网页的渲染结果。每个批渲染

引擎120可被配置为模拟用于个人web浏览器的渲染器,但是针对批渲染进行了优化。批渲染系统110可包括数以千计的批渲染引擎120并且使用负载平衡来选择批渲染引擎120之一以响应请求特定。所请求的网页可包括诸如样式表、JavaScript、图像等的嵌入式对象。批渲染引擎120可利用获取服务122来请求嵌入式对象的内容。

[0023] 获取服务122可确定需要从主机服务器190获取哪些嵌入式资源、可从缓存返回哪些嵌入式资源以及哪些资源不需要返回。主机服务器190可以是可经由互联网访问的托管嵌入一个或更多个网页中的一个或更多个网页或资源的任何类型的计算装置。如果嵌入式资源需要被获取,则获取服务122可利用传统技术经由web爬取引擎124请求嵌入式对象的内容。诸如索引引擎的向下传输过程还可经由web爬取引擎124向服务器190请求内容。作为经由web爬取引擎124的获取请求的结果,批渲染系统110可生成获取记录132。获取记录132可包括关于向主机服务器190请求并从其检索了哪些网页和嵌入式对象的信息。获取记录132还可包括诸如请求的时间、嵌入器网页(如果有的话)等的附加信息。

[0024] web爬取引擎124、批渲染引擎120和获取服务122一起工作以有效地渲染大量的网页(诸如可在万维网上找到的网页)。网页的渲染是包括可用于以及或者不可用于向下传输请求过程的各种数据元素的渲染结果。可选资源识别引擎126可使用批渲染引擎120来生成可选资源模式130。可选资源识别引擎126可使用获取记录132作为输入定期地(例如,每日,一周两次等)运行。可选资源识别引擎126可分析获取记录132并且生成在先前时段中获取的URL的模式。例如,可选资源识别引擎126可将查询串从获取记录中的URL中去除,从而生成每个相应URL的组URL。查询串可以是URL中在问号(?)之后的任何字符。在一些实现方式中,可仅去除查询串的部分以生成URL的组URL。然后,可选资源识别引擎126可对组URL进行分类或聚类并且确定哪些组URL具有来自该时间段的最大数量的获取请求。

[0025] 可选资源识别引擎126可选择具有最大数量的获取请求的可能模式组URL。这样的模式表示具有最大可能优化浏览过程的嵌入式资源,因为它们最经常被获取。可选资源识别引擎可确定与可能模式匹配的URL或者与可能模式匹配的URL的样本是不是可选的。例如,可选资源识别引擎126可从获取记录132识别与可能模式匹配的嵌入式资源,并且请求批渲染引擎120针对该嵌入式资源渲染嵌入器网页的第一渲染结果。嵌入器网页可例如从嵌入式资源的获取记录来识别。然后,可选资源识别引擎126可请求批渲染引擎120在不获取嵌入式资源的情况下渲染嵌入器网页的第二渲染结果。

[0026] 为了确定所渲染的嵌入器网页的内容是否受跳过特定嵌入式资源的影响,可选资源识别引擎126可比较第一渲染结果和第二渲染结果。在一些实现方式中,如果内容受到显著影响,则该特定嵌入式资源可被添加到需要的资源的列表。如果内容不受显著影响(例如,内容相似),则可选资源识别引擎126可将嵌入式资源识别为可选资源。在一些实现方式中,嵌入式资源(例如,它的URL)可被存储在诸如可选资源模式130的数据存储中。在一些实现方式中,嵌入式资源可被存储在可选资源的临时列表中,该临时列表在稍后的时间用于确定可选资源的模式。可选资源识别引擎126可关于获取记录中与可能模式匹配的每一个嵌入式资源或者针对与该可能模式匹配的嵌入式资源的样本执行此测试(渲染结果的比较)。在一些实现方式中,如果与可能模式匹配的任何嵌入式资源是相关的,则该可能模式不是可选资源模式。在一些实现方式中,如果仅较小百分比的与可能模式匹配的嵌入式资源(例如,1%或更少)是需要的资源,则该可能模式被认为是可选资源模式,并且可选资源

识别引擎126可包括该可能模式作为可选资源模式130中的模式。当然,如果与可能模式匹配的所有测试的嵌入式资源均为可选的,则可选资源识别引擎126可将该可能模式添加到可选资源模式130。

[0027] 在一些实现方式中,可选资源识别引擎126还可验证可选资源模式130中的模式仍是可选的。例如,可选资源识别引擎126可选择与模式匹配的URL的样本、要求渲染引擎在有与模式匹配的嵌入式资源以及没有所述嵌入式资源的情况下对渲染结果进行渲染,并且比较两个渲染结果。如果与可选资源模式130中的模式匹配的URL不再是可选的,则该模式可被移除。

[0028] 在一些实现方式中,可选资源识别引擎126可与完整浏览器分离地针对移动浏览器确定模式是不是可选。例如,一些网站针对移动版本的网页和完整版本的网页加载不同的内容,并且嵌入式资源可对于移动浏览器而言是可选的,但是对完整浏览器而言是所需的。因此,可选资源识别引擎126可针对嵌入式资源渲染四个渲染结果。前两个渲染结果可由模拟完整浏览器的渲染引擎渲染,而后两个渲染结果可由模拟移动装置(例如,智能电话或平板)上的移动浏览器的渲染引擎渲染。因此,可选资源模式130可包括按照浏览器类型(例如,针对移动浏览器或者针对完整浏览器)指示模式是否是可选的数据。

[0029] 尽管为了简明起见在图1中未示出,在一些实现方式中,批渲染系统110可分布于两个或更多个分立的计算装置上。另外,批渲染引擎120、获取服务122、web爬取引擎124、和可选资源识别引擎126中的一个或更多个可跨一个或更多个计算装置分布。在一些实现方式中,批渲染引擎120、获取服务122、web爬取引擎124、和可选资源识别引擎126中的一个或更多个可与批渲染系统110的其它组件共享诸如存储器或硬件处理器的资源。类似地,获取记录132和可选资源模式130也可被存储在跨多个计算装置分布的存储器中。在一些实现方式中,批渲染系统110的各种组件可共享计算装置的硬件组件,或者可以是同一计算装置的逻辑分区。

[0030] 批渲染系统110可经由网络160与客户端180和服务器190通信。网络160可以是例如互联网,或者网络160可以是利用例如网关装置、网桥、开关等实现的有线或无线局域网(LAN)、广域网(WAN)、这些的组合等。经由网络160,批渲染系统110可与客户端180和/或主机服务器190通信并且向其发送数据/从其接收数据。例如,批渲染系统110可将可选资源模式130提供给客户端180,或者可从客户端180接收询问特定嵌入式资源是否是可选的请求并且可基于可选资源模式130提供响应。

[0031] 客户端180可以是安装有个人web浏览器140的个人计算装置,例如图6中所描绘的计算装置600。个人web浏览器的示例包括CHROME、SAFARI、INTERNET EXPLORER、FIREFOX等,无论用于诸如智能电话或平板的移动装置,还是诸如膝上型或桌面型的个人计算机。浏览器140可被配置为当渲染网页时使用服务来确定是否获取特定嵌入式资源。在一些实现方式中,系统可将可选资源模式130的副本推送给客户端180以便于浏览器140使用。从这个意义上讲,服务可以是本地服务。在一些实现方式中,浏览器140可被配置为要求基于服务器的服务确定特定嵌入式资源是否与可选资源模式130中的模式之一匹配。浏览器140通过利用该服务来执行优化的渲染以在渲染网页时跳过可选嵌入式资源。因此,浏览器140可被配置为选择性地跳过嵌入式资源以改善渲染时间并且节省内容不受影响的资源。

[0032] 图2是根据实现方式的由批渲染引擎120渲染的渲染结果200的框图。渲染结果200

可包括各种组件,图2中并未图示出其中的全部。例如,渲染结果200可包括渲染的页面的图像205。图像205可以是经由web浏览器经由显示装置显示给用户的图像。图像205可用于例如将渲染的页面的缩略图显示给用户并且确定网页的元素出现在显示器上的何处(例如,基于其x和y坐标)。渲染结果200还可包括文档对象模型(DOM)树210。DOM树210表示网页的HTML结构。渲染结果200还可包括布局215。布局215通常包括用于网页的元素的框,所述框指定图像205中的元素的x和y坐标。因此,布局215提供元素出现在网页上的何处、它在网页上占据多少空间等的指示。因此,布局215提供关于有多少网页是广告、段落有多重要(例如,首屏或非首屏)、元素是否可见等的信息。换言之,布局215提供关于所渲染的网页的元素的几何信息。渲染结果200还可包括错误220。错误220包括作为运行脚本对象(例如,JavaScript)的结果遇到的错误。渲染结果200还可包括在渲染期间获取的嵌入式资源225的列表,并且可包括作为渲染过程的部分生成的其它元素(未示出)。因此,渲染结果200提供并非仅仅对经由从托管服务器获取内容可用的信息。诸如可选资源识别引擎的向下传输请求过程可将渲染结果信息用于各种目的,诸如确定运行脚本对象是否显著改变所渲染的网页的内容。例如,网页的广告部分中的内容可被认为不重要,因此即使广告部分中存在差异,URL仍可为可选的。在一些实现方式中,仅渲染结果的主要组件的差异可被认为重要。主要组件可以是布局215中的最大的框(例如,具有最大高度和宽度的框)。

[0033] 图3是图示出根据实现方式的用于识别可选资源模式的示例过程300的流程图。过程300可由诸如图1的系统110的系统执行。该系统可使用过程300来从获取记录识别可选资源的模式。该系统可基于在有资源和没有资源的情况下渲染的嵌入式网页的渲染结果比较来确定可选资源的模式。模式可被存储在数据存储中并且用作允许浏览器(无论是完整浏览器、移动浏览器还是批渲染引擎)通过跳过(例如,不获取)可选资源来优化渲染的服务的一部分。

[0034] 过程300可开始于系统按照可能模式对所获取的嵌入式资源进行聚类(305)。系统可选择在某一先前时间段内(例如,一天、两天、一周、自过程300最后一次运行起等)获取的所有嵌入式资源。可能模式可通过将所有或部分的查询串从在获取记录中找到的URL去除来生成。获取记录可能已通过web爬取引擎或获取服务生成。在一些实现方式中,获取记录可作为索引过程(例如,经由互联网可获得的文档的索引过程)的一部分生成。嵌入式资源可在获取记录中通过诸如URL的标识符来识别。因此,网页或嵌入式资源也可称作URL,将理解如本文所用,对嵌入式资源(或网页)和URL的引用通常是同义的。可能模式可针对URL通过去除所有或部分的查询串来生成,所述查询串是嵌入式资源的URL中跟随在问号(?)之后的任何字符。可能模式还可针对嵌入式资源通过去除或忽略子域名、忽略路径的组件(例如,介于第一正斜杠("/")和查询串("?")之间的组件)、或者忽略查询串中的特定参数来生成。在一些实现方式中,来自获取记录的嵌入式资源可与从其URL生成的多于一个的可能模式关联。可能URLs可通过分类或者通过其它已知的聚类技术来聚类。

[0035] 系统可选择由可能模式之一表示的聚类(310),并且确定聚类成员的量是否大于或等于阈值(315)。聚类成员的量表示在先前时间段期间与可能模式匹配的嵌入式资源被获取的次数的量。因此,一个特定嵌入式资源如果被获取多于一次,则可在量中被表示多次。如果聚类的量低于阈值(315,否),则系统可转向下一聚类(355)。如果可能模式具有足够量的成员(315,是),则系统可选择与可能模式匹配的嵌入式资源的样本(320)。换言之,

系统可对聚类的独特成员进行采样。当然,在一些实现方式中,系统可选择聚类中的所有独特嵌入式资源作为样本(例如,样本大小为100%)。然后,系统可选择样本中的嵌入式资源中的一个嵌入式资源(325)并且利用所选择的嵌入式资源生成嵌入资源的网页(例如,嵌入器网页)的第一渲染结果(330)。嵌入器网页可从获取记录确定。系统还可在不获取所选择的嵌入式资源的情况下生成嵌入器网页的第二渲染结果(330)。系统可利用例如批渲染引擎来生成两个渲染结果。然后,系统可比较第一渲染结果和第二渲染结果以确定渲染结果是否相似(335)。

[0036] 系统可使用各种测试来确定渲染结果是否相似,如关于图4更详细讨论的。如果渲染结果不相似(335,否),则系统可移至下一聚类(355)。如果渲染结果相似(335,是),则系统可确定聚类样本中是否存在其它嵌入式资源待检查(340)。如果存在其它嵌入式资源(340,否),则系统可选择样本的下一嵌入式资源(345)并且重复步骤330和335,从而生成渲染结果并比较结果。如果样本中的所有嵌入式资源均已被检查(340,是),则系统可将聚类的可能模式添加到可选资源模式数据存储(350)。此数据存储可由服务用来识别不需要获取的嵌入式资源,如关于图5更详细说明的。然后,系统可确定是否存在其它聚类待检查(355)。如果留有其它聚类待分析(355,否),则系统可选择下一聚类,针对与下一聚类的可能模式匹配的嵌入式资源重复步骤315至355。如果不再有聚类待分析(355,是),则过程300完成。

[0037] 示例过程300图示出一个非可选的嵌入式资源将导致可能模式被排除在可选资源模式数据存储之外(例如,335,否)的实现方式。然而,在一些实现方式中,系统可保存可能模式的可选的嵌入式资源的列表以及可能模式的需要(例如,两个渲染结果被认为不相似)的嵌入式资源的列表。在这样的实现方式中,系统可计算与可能模式匹配的需要的嵌入式资源的百分比。当该百分比非常低(例如,1%或更小)时,系统仍可将该可能模式添加到可选资源模式数据存储(350)。因此,实现方式不限于图3所示的精确过程300。

[0038] 在一些实现方式中,系统可利用通过不同的浏览器类型渲染的渲染结果来重复过程300的一些或全部。例如,浏览器类型可以是完整浏览器或移动浏览器。如果第一和第二渲染结果由完整浏览器渲染,则系统可利用移动浏览器来重复步骤330至350。因此,可选资源模式的列表可包括浏览器类型的指示,数据存储中的模式也可包括浏览器类型的指示。这允许系统指示嵌入式资源是否对于一个浏览器类型(例如,移动浏览器)是可选的,对于不同的浏览器类型(例如,完整浏览器)不是可选的。

[0039] 图4是图示出根据实现方式的用于确定两个渲染结果是否相似的示例过程400的流程图。过程400可例如作为图3的步骤335的一部分来执行。尽管过程400图示出用于确定渲染结果相似度的三个测试,将理解实现方式可包括所图示的测试中的一个、两个或全部,并且可包括没有图示出的附加相似度测试。

[0040] 过程400可包括比较两个渲染结果中的令牌(405)。令牌包括文档的文本,例如,对用户可见的词语。令牌可通过处理DOM树(诸如图2的DOM树210)来生成。在一些实现方式中,可在执行比较之前移除停止词和数字。如果第一渲染结果的令牌不同于第二渲染结果的令牌(410,是),则可选地,系统可确定通过在打开脚本编写的情况下渲染而添加的独特令牌的量是否小于令牌阈值(415)。换言之,如果在打开脚本编写的情况下的渲染仅导致几个独特令牌,则系统可认为差异不显著。在一些实现方式中,令牌阈值可为5。如果所添加的独特

令牌的量满足令牌阈值 (415, 否), 则系统可认为渲染结果不相似 (430)。当渲染结果不相似时, 用于生成渲染结果的嵌入式资源不是可选的。在一些实现方式中, 系统可跳过测试 415 且令牌的任何差异可被认为是显著的并且渲染结果不相似。如果所述量未能满足令牌阈值 (415, 是), 则系统可认为令牌的差异不显著。因此, 如果第一渲染结果的独特令牌与第二渲染结果中的令牌相同 (410, 否) 或者独特令牌的量未能满足令牌阈值 (415, 是), 则渲染结果可被认为相似, 并且系统可继续执行其它测试。在仅包括令牌测试的实现方式中, 系统可直接前进至步骤 445, 确定渲染结果相似。

[0041] 在一些实现方式中, 如果令牌相同 (410, 否) 或者 (415, 是), 则系统可比较外链 (420)。外链表示从渲染的页面至其它网页或其它文档的链接。外链也可从渲染结果的 DOM 树的锚定标签节点 (<a> 标签) 提取。如果系统发现外链的差异 (425, 是), 则系统可认为该差异显著并且渲染结果不相似 (430)。因此, 嵌入式资源被认为不是可选的。如果两个渲染结果中的外链相同 (425, 否), 则系统可认为渲染结果相似 (例如, 通过前进至步骤 450), 或者可转向执行其它相似度测试。在一些实现方式中, 系统可在步骤 405 和 415 之前并且独立于步骤 405 和 415 执行步骤 420 和 425。

[0042] 在可除了其它测试之外或者代替其它测试执行的另一相似度测试中, 系统可确定图像 (例如, 快照)、DOM 树或布局之间的相似度。例如, 系统可计算 DOM 树、布局、图像或者渲染结果的另一组件的最长共有序列 (LCS) (435)。系统可使用 LCS 来计算表示相同的组件的百分比的相似度分值 (440)。当百分比满足阈值 (445, 是) 时, 系统可确定渲染结果相似 (450)。当百分比未能满足阈值 (445, 否) 时, 系统可确定渲染结果不相似 (430)。一旦系统确定可渲染结果是相似 (450) 还是不相似 (430), 过程 400 结束。

[0043] 当然, 系统可作为过程 400 的一部分执行其它测试。例如, 在另一相似度测试 (图 4 中未示出) 中, 系统可确定布局的主要组件。渲染结果的布局由框构成——每个框由屏幕坐标定义。通常, 每个框对应于网页的相应元素。例如, 布局可包括 DOM 树中的 DOM 节点的框表示 (尽管并非每一个 DOM 元素均可具有对应渲染框)。框可按照树结构 (也被称作渲染树) 来组织。因此, 例如, 表可由布局中的框表示, 段落可由布局中的另一框表示。网页的主要组件是布局中具有最大框 (由屏幕坐标定义) 的那些元素。在一些实现方式中, 例如, 系统可将预定的主要组件包括在集合中, 以使得该集合包括最大的主要组件。在一些实现方式中, 系统可将构成一定百分比的屏幕的主要组件包括在集合中。在这样的实现方式中, 如果最大主要组件占据超过该百分比, 则最大主要组件可以是集合的仅有成员。在一些实现方式中, 框大小高于阈值大小的任何组件可被包括在主要组件的集合中。不在主要组件的集合中的组件可被认为是次要组件。在一些实现方式中, 系统可使用剥洋葱技术来找到主要组件。例如, 从渲染树的根框开始, 系统可执行广度优先搜索并且识别根框的最大子框。然后, 系统可选择最大子框并且进一步深入, 以找到当前框 (例如, 根框的最大子框) 的最大子框。当没有子框占优势 (例如, 占据父框面积的一半以上) 时, 系统可停止深入。当系统停止深入时, 主要组件是未能具有优势子框的框。

[0044] 系统可计算集合中的主要组件之间的相似度分值。例如, 系统可使用主要组件框的交叠分值。在交叠分值中, 系统可计算主要组件的交叠面积, 其表示与第二渲染结果中的对应主要组件的面积交叠的第一渲染结果中的主要组件的面积。然后, 系统可计算相似分值, 其表示交叠的调和平均相对于每个主要组件的总面积。例如, 系统可使用式

$\frac{2}{((a1/oa) + (a2/oa))}$, 其中oa是交叠面积,a1是第一渲染结果中的主要组件框的总面积,且a2是第二渲染结果中的主要组件框的总面积。当然,系统可使用其它相似度量来计算分值。这样的相似度量度的示例可包括(但不限于)Katz相似度。如果相似度分值满足相似度阈值,则渲染结果被认为相似,因此嵌入式资源是可选的。在一些实现方式中,相似度阈值可较高,例如80%或更高。如果相似度分值未能满足阈值,则系统可认为渲染结果不相似。在一些实现方式中,如果相似度分值满足相似度阈值,则系统可基于渲染结果执行附加相似度测试(图4中未图示出)。在一些实现方式中,系统还可针对次要组件的差异计算相似度分值,但是使这些差异的权重小很多(例如,使得它们在渲染结果的最终相似度分值中占据很小部分)。

[0045] 在一些实现方式中,系统可尝试最小化或消除过程400上的不可预测性的影响。当即使嵌入式资源相同渲染结果也不同时,发生不可预测性。不可预测性可导致可选的嵌入式资源看起来如同它们是需要的一样。因此,考虑不可预测性可显著添加可选资源的覆盖范围。为了考虑不可预测性,系统可生成第三渲染结果。第三渲染结果可以是使用从第一渲染结果的生成返回的资源(例如,包括样本URL)的渲染。第三渲染结果与第一渲染结果之间导致的任何差异均归因于不可预测性。系统可通过移除归因于不可预测性的任何差异(例如,在第一和第三渲染结果中不同的外链、令牌、图像、渲染树等的差异)来净化相似度。例如,系统可计算第一量和第二量,第一量表示在第一渲染结果的图像与第三渲染结果的图像之间不同的像素的数量,第二量表示在第一渲染结果和第二渲染结果的图像之间不同的像素的数量。系统可从第二量减去第一量以生成差值。该差值越靠近第二量,第一和第二渲染结果之间的任何差异不归因于不可预测性的可能性越大。因此,系统可基于该差值来调节LCS的相似度分值(例如,如果该差值等于第二量则不调节,如果该差值是第二量的一半则调节一半,等等)。作为另一示例,系统可确定在第一渲染结果和第三渲染结果之间不同的DOM节点,并且在比较第一渲染结果与第二渲染结果(例如,计算LCS)时忽略这些节点。

[0046] 图5是图示出根据实现方式的提供用于优化的渲染的信息的示例过程500的流程图。过程500可作为浏览器的服务运行。浏览器可以是诸如图1的浏览器140的客户端浏览器(移动的或完整的)或者诸如图1的批渲染引擎120的批渲染引擎。在一些实现方式中,服务可以是基于云的服务。换言之,浏览器可经由网络向可向浏览器提供响应的基于云的服务发送请求。在一些实现方式中,服务可在服务器(例如,图1的批渲染系统110)上运行。在其它实现方式中,服务可以是运行浏览器的计算装置本地的。例如,可选资源模式的数据存储可被推送至运行浏览器的计算装置,服务可在计算装置处执行。在一些实现方式中,服务可在客户端(诸如图1的客户端180)上运行。

[0047] 过程500开始于服务从浏览器接收URL(505)。浏览器可以是移动浏览器、完整浏览器,并且可在客户端上运行或者作为批渲染引擎运行。服务可确定来自请求的URL是否与可选资源模式匹配(510)。在一些实现方式中,模式可表示完整URL,以使得对整个请求的URL进行匹配。在一些实现方式中,模式可表示部分URL(例如,移除了查询串或者各种部分被通配符替换)。如果所请求的URL与可选资源数据存储中的至少一个模式匹配(510,是),则服务提供指示URL是可选的响应,并且嵌入器网页可在不获取内容的情况下渲染(515)。在一些实现方式中,这可以是与当无法定位资源时浏览器可接收的“没有找到URL”响应相似的

响应。响应于接收到该指示,浏览器可跳过嵌入式资源渲染嵌入器网页,从而节省处理和带宽资源。如果所请求的URL不与可选资源数据存储中的模式匹配(510,否),则服务提供指示嵌入式资源是需要的响应(520)。在一些实现方式中,服务可实际执行获取并且提供嵌入式资源的内容。在一些实现方式中,服务可提供告知浏览器继续进行嵌入式资源的获取请求的响应。因此,浏览器可利用嵌入式资源的内容来渲染嵌入器网页。然后,过程500结束,提供了浏览器可用于优化渲染过程的信息。在一些实现方式中,浏览器可向服务提供浏览器类型,服务可在与可选资源数据存储中的模式进行匹配时使用浏览器类型。

[0048] 图6示出可用于这里所描述的技术的一般计算机装置600的示例,其可作为图1的批渲染系统120和/或客户端180操作。计算装置600旨在表示各种示例形式的计算装置,例如膝上型计算机、桌面型计算机、工作站、个人数字助理、蜂窝电话、智能电话、平板、服务器、以及其它计算装置(包括可穿戴装置)。这里所示的组件、其连接和关系以及它们的功能意在仅为示例,而不意在限制本申请中所描述和/或要求保护的发明的实现方式。

[0049] 计算装置600包括经由接口608连接的处理器602(例如,基于硅氧烷的硬件处理器)、存储器604、存储装置606、以及扩展端口610。在一些实现方式中,计算装置600可包括经由接口608连接的收发器646、通信接口644、和GPS(全球定位系统)接收器模块648以及其它组件。装置600可通过通信接口644无线地通信,其可在需要的地方包括数字信号处理电路。组件602、604、606、608、610、640、644、646和648中的每一个可被安装在公共母板上或者酌情以其它方式来安装。

[0050] 处理器602可处理用于在计算装置600内执行的指令,包括存储在存储器604中或者存储装置606上以在外部输入/输出装置(例如,显示器616)上显示GUI的图形信息的指令。显示器616可以是监视器或者平板触摸屏显示器。在一些实现方式中,连同多个存储器和多种类型的存储器一起,可酌情使用多个处理器和/或多条总线。另外,可连接多个计算装置600,其中每个装置提供必要操作的部分(例如,作为服务器阵列、一组刀片服务器或者多处理器系统)。

[0051] 存储器604将信息存储在计算装置600内。在一个实现方式中,存储器604是易失性存储器单元。在另一实现方式中,存储器604是非易失性存储器单元。存储器604还可以是另一形式的计算机可读介质,例如磁盘或光盘。在一些实现方式中,存储器604可包括通过扩展接口提供的扩展存储器。

[0052] 存储装置606能够为计算装置600提供大容量存储。在一个实现方式中,存储装置606可以是或者可包括诸如软盘装置、硬盘装置、光盘装置或磁带装置、闪存存储器或者其它相似的固态存储器装置或者装置阵列的计算机可读介质,包括存储区域网络或者其它配置的装置。计算机程序产品可被有形地具体实现于此类计算机可读介质中。计算机程序产品还可包括指令,所述指令在被执行时执行一个或多个方法,例如上述那些方法。计算机或者机器可读介质是诸如存储器604、存储装置606、或者处理器602上的存储器的存储装置。

[0053] 接口608可以是管理计算装置600的带宽密集操作的高速控制器或者管理带宽较不密集操作的低速控制器,或者这些控制器的组合。可提供外部接口640以允许装置600与其它装置的附近区域通信。在一些实现方式中,控制器608可耦合至存储装置606和扩展端口614。可包括各种通信端口(例如,USB、蓝牙、以太网、无线以太网)的扩展端口可例如通过网络适配器来耦合至诸如键盘、指点装置、扫描仪的一个或多个输入/输出装置或者诸如交

换机或路由器的网络装置。

[0054] 如图所示,计算装置600可按照多种不同的形式来实现。例如,它可被实现为标准服务器630或者多次实现于一组此类服务器中。它还可被实现为机架服务器系统的一部分。另外,它可被实现于诸如膝上型计算机632、台式计算机634、或者平板或智能电话636的个人计算装置中。整个系统可由彼此通信的多个计算装置600构成。其它配置也是可能的。

[0055] 图7示出可用于这里所描述的技术的一般计算机装置700的示例,其可以是图1的系统110。计算装置700旨在表示各种示例形式的大型数据处理装置,诸如服务器、刀片服务器、数据中心、大型机以及其它大型计算装置。计算装置700可以是具有多个处理器的分布式系统,可能包括通过一个或多个通信网络互连的附接网络的存储节点。这里所示的组件、其连接和关系以及它们的功能意在仅为示例,而并非意在限制本申请中所描述和/或要求保护的发明的实现方式。

[0056] 分布式计算装置700可包括任何数量的计算装置780。计算装置780可包括经局域网或广域网、专用光学链路、调制解调器、桥、路由器、交换机、有线或无线网络等通信的服务器或机架服务器、大型机等。

[0057] 在一些实现方式中,每个计算装置可包括多个机架。例如,计算装置780a包括多个机架758a-758n。每个机架可包括一个或多个处理器,例如处理器752a-752n和762a-762n。处理器可包括数据处理器、附接网络的存储装置以及其它计算机控制的装置。在一些实现方式中,一个处理器可用作主处理器并且控制调度和数据分发任务。处理器可通过一个或多个机架交换机758来互连,并且一个或多个机架可通过交换机778来连接。交换机778可处理多个连接的计算装置700之间的通信。

[0058] 每个机架可包括诸如存储器754和存储器764的存储器以及诸如756和766的存储装置。存储装置756和766可提供大容量存储,并且可包括易失性或非易失性存储装置,例如附接网络的盘、软盘、硬盘、光盘、磁带、闪存或者类似的固态存储器装置或者装置阵列,包括存储区域网络或者其它配置的装置。存储装置756或766可在多个处理器、多个机架或者多个计算装置之间共享,并且可包括存储可由一个或多个处理器执行的指令的计算机可读介质。存储器754和764可包括例如易失性存储器单元、非易失性存储器单元和/或其它形式的计算机可读介质,例如磁盘或光盘、闪存、高速缓存、随机存取存储器(RAM)、只读存储器(ROM)和其组合。诸如存储器754的存储器还可在处理器752a-752n之间共享。可例如在存储装置756和存储器754上诸如索引的数据结构。计算装置700可包括未示出的其它组件,例如控制器、总线、输入/输出装置、通信模块等。

[0059] 整个系统(例如系统110)可由彼此通信的多个计算装置700构成。例如,装置780a可与装置780b、780c和780d通信,并且这些装置可被统称为系统100。作为另一示例,图1的系统100可包括一个或多个计算装置700。一些计算装置可在地理上彼此靠近,其它计算装置可地理上彼此远离。计算装置700的布局仅是示例,系统可采取其它布局或配置。

[0060] 各种实现方式可包括可在可编程系统上执行和/或解释的一个或多个计算机程序中的实现方式,所述可编程系统包括耦合以从存储系统接收数据和指令以及向存储系统发送数据和指令的形成在基底上的至少一个可编程处理器(可以是专用的或通用的)、至少一个输入装置和至少一个输出装置。

[0061] 这些计算机程序(也称作程序、软件、软件应用或代码)包括用于可编程处理器的

机器指令,并且可利用高级过程化和/或面向对象的编程语言和/或汇编/机器语言来实现。如本文所用,术语“机器可读介质”、“计算机可读介质”表示用于向可编程处理器提供机器指令和/或数据的任何非暂时性计算机程序产品、设备和/或装置(例如,磁盘、光盘、存储器(包括读取存取存储器)、可编程逻辑器件(PLD))。

[0062] 这里所述的系统和技术可被实现于计算系统中,该计算系统包括后端组件(例如,作为数据服务器),或者包括中间件组件(例如,应用服务器),或者包括前端组件(例如,具有图形用户接口或者Web浏览器的客户端计算机,用户可通过其来与这里所述的系统和技术实现方式交互),或者此类后端、中间件或前端组件的任何组合。系统的组件可通过任何形式或介质的数字数据通信(例如,通信网络)来互连。通信网络的示例包括局域网(“LAN”)、广域网(“WAN”)和互联网。

[0063] 计算系统可包括客户端和服务端。客户端和服务端通常彼此远离,并且通常通过通信网络来交互。客户端和服务端的关系由于在相应计算机上运行并且彼此具有客户端-服务器关系的计算机程序而产生。

[0064] 描述了多个实现方式。然而,在不脱离本发明的精神和范围的情况下,可进行各种修改。另外,图中所描绘的逻辑流程不要求所示的特定顺序或者依次顺序来实现期望的结果。另外,可提供其它步骤,或者可从所描述的流程排除步骤,并且可向所描述的系统添加其它组件或者从其移除组件。因此,其它实现方式在以下权利要求的范围内。

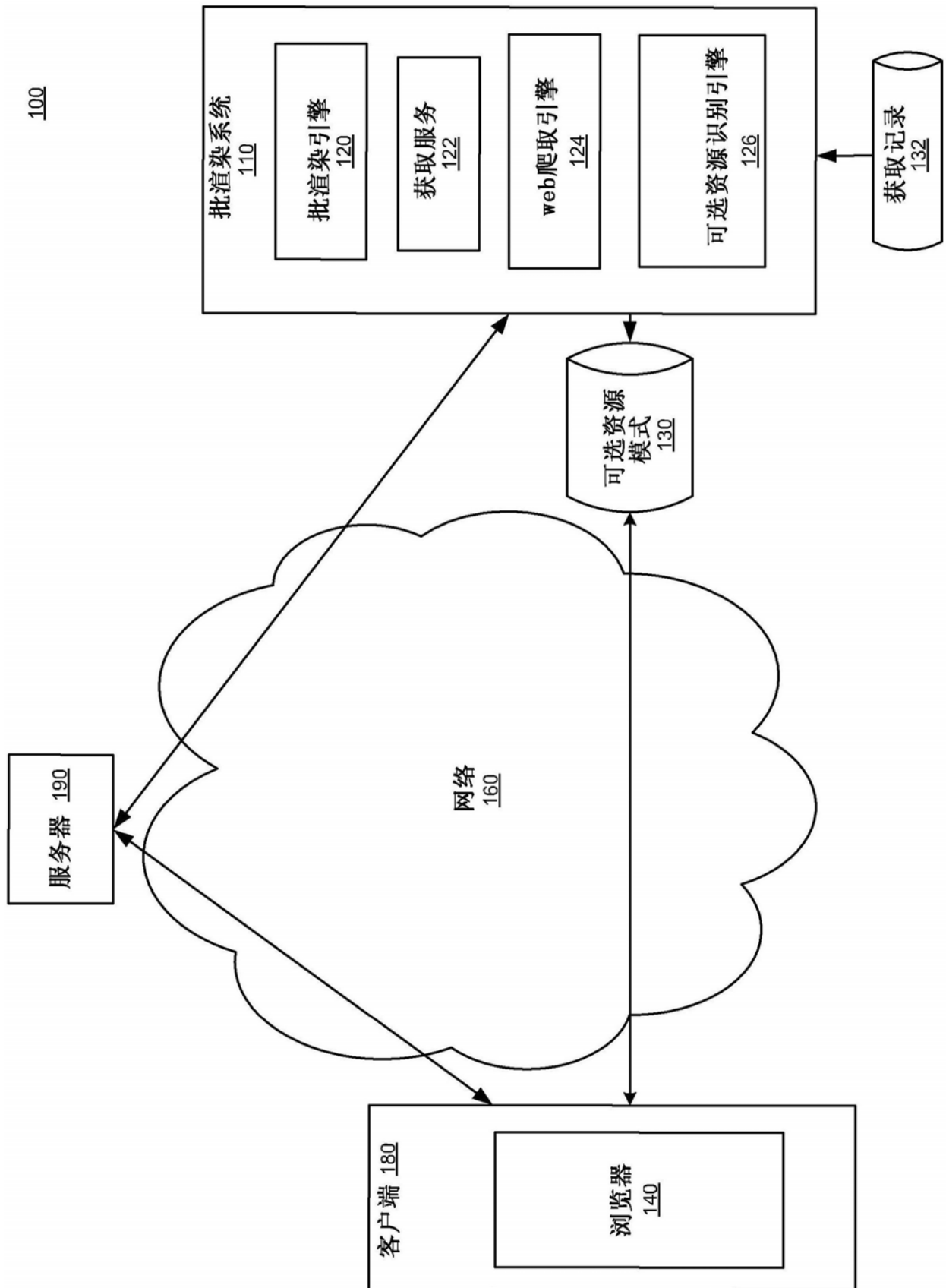


图1

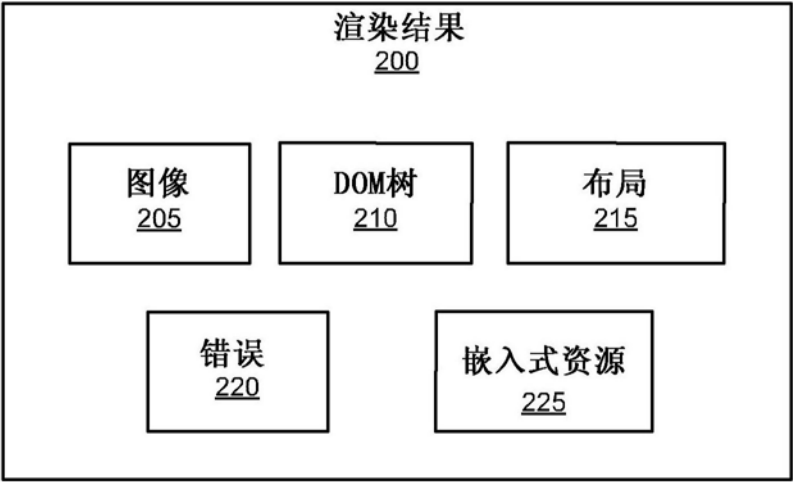


图2

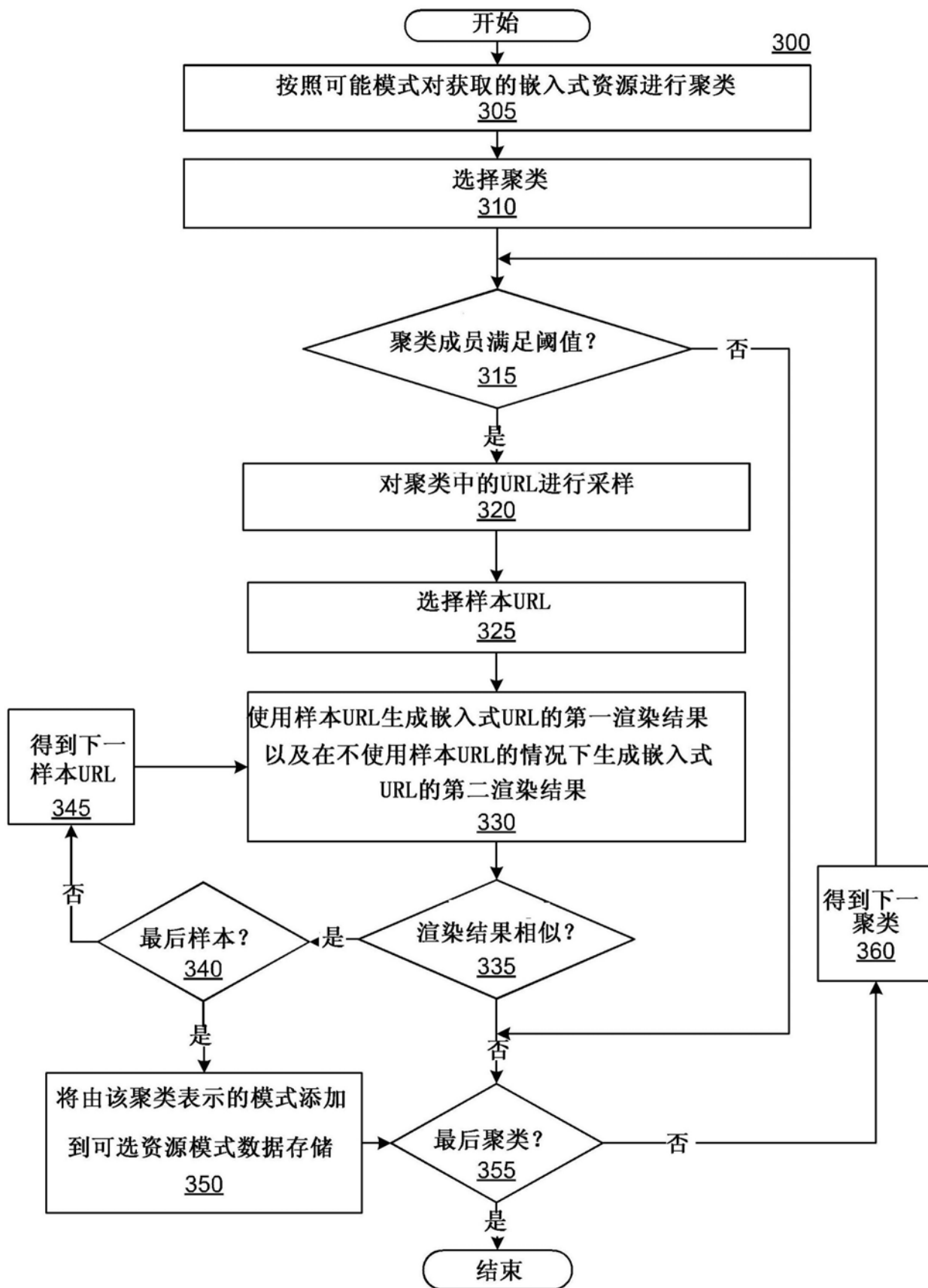


图3

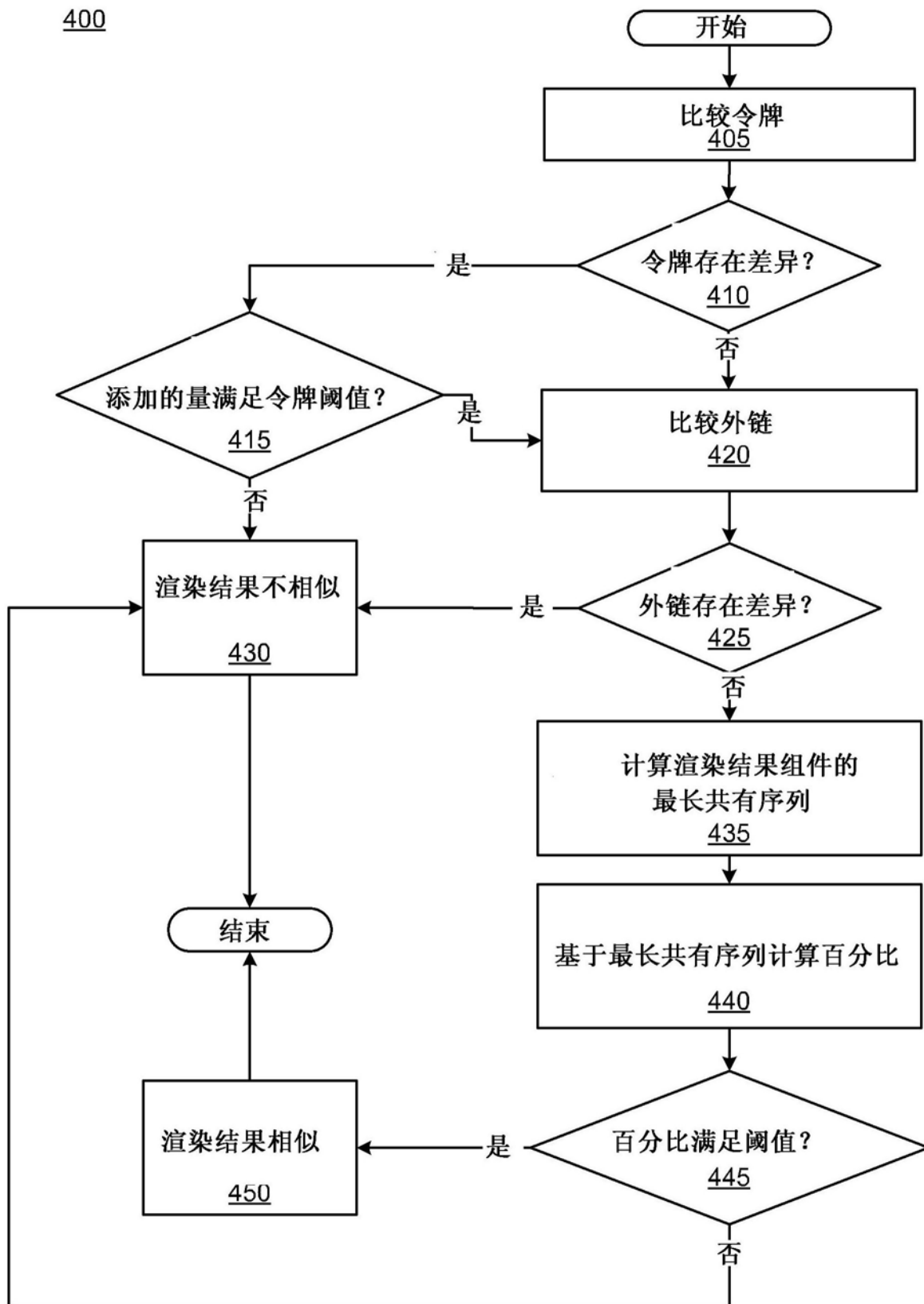


图4

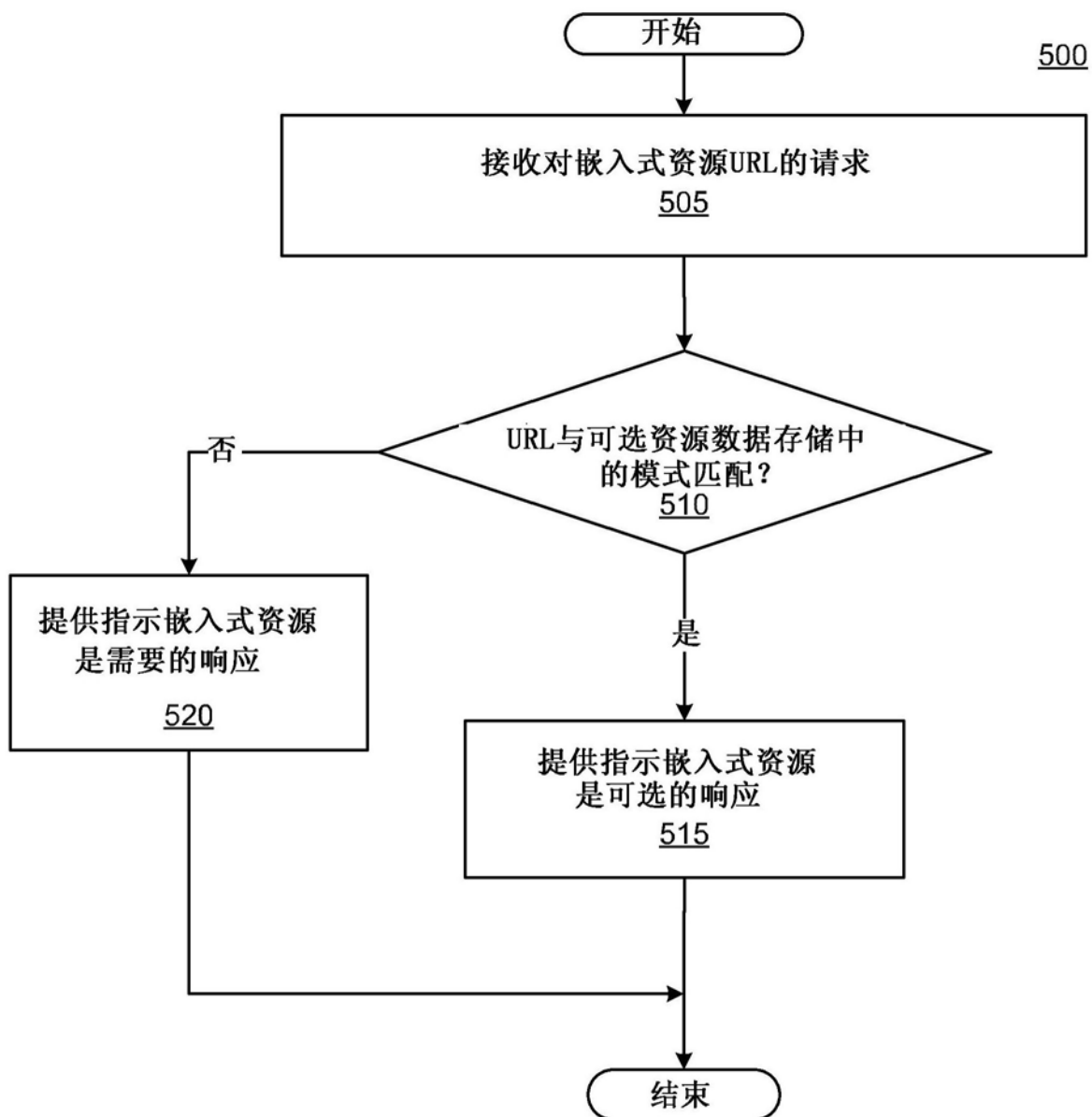


图5

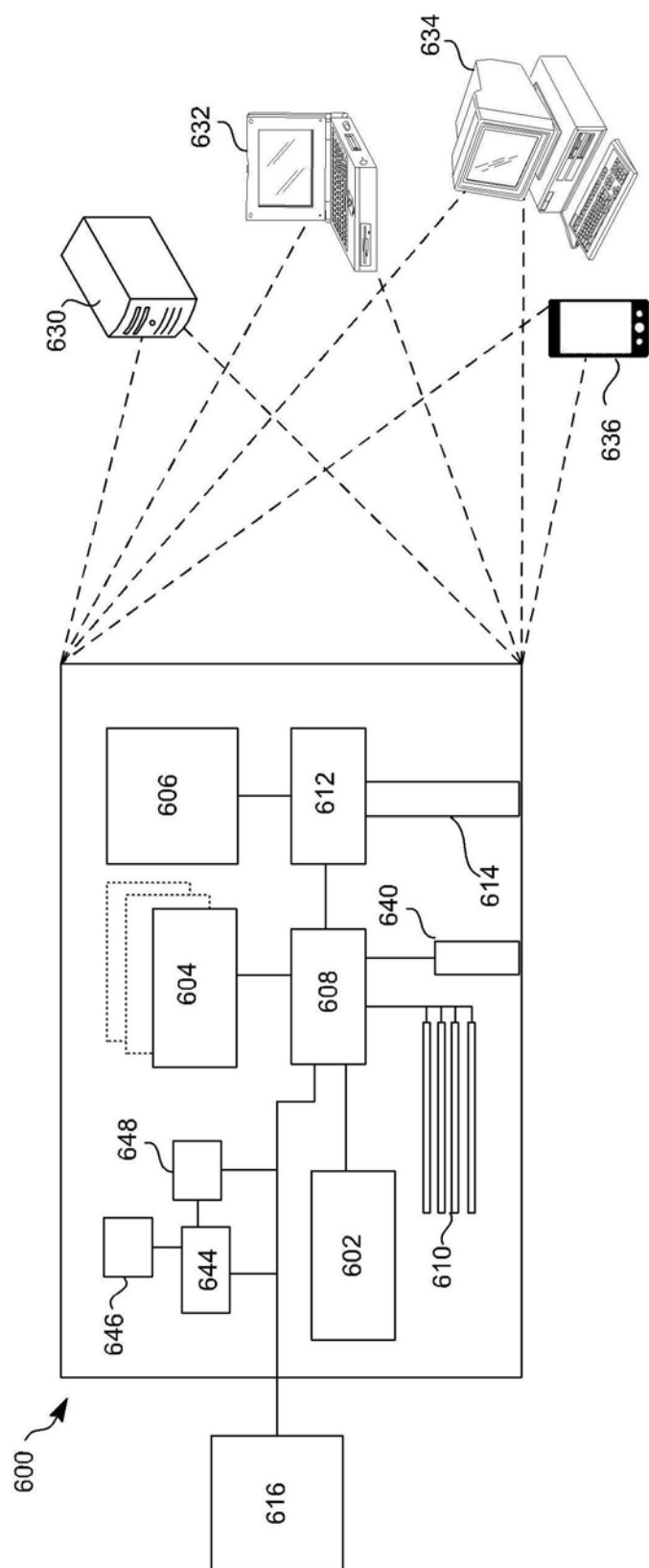


图6

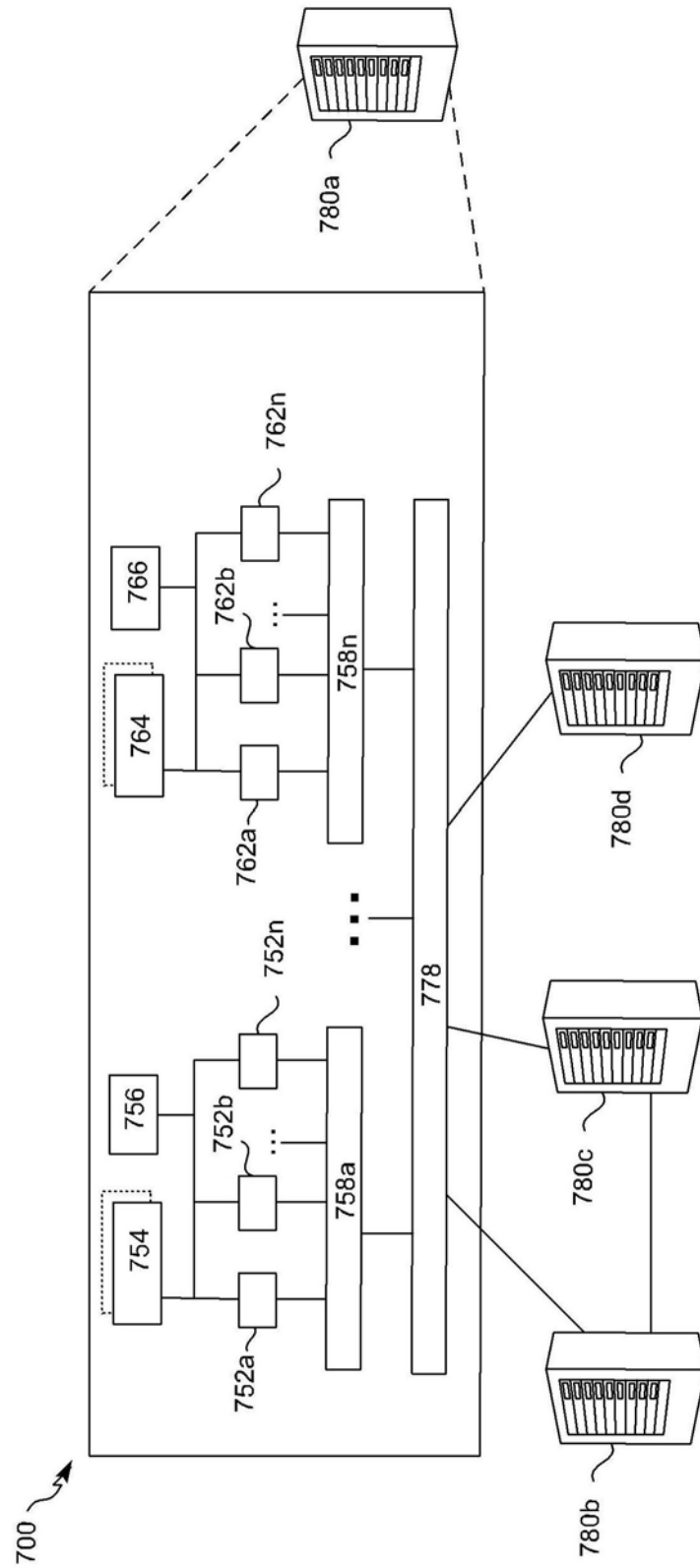


图7