



(12) 发明专利申请

(10) 申请公布号 CN 104484235 A

(43) 申请公布日 2015. 04. 01

(21) 申请号 201410681624. 4

(22) 申请日 2014. 11. 24

(71) 申请人 北京奇虎科技有限公司

地址 100088 北京市西城区新街口外大街  
28号D座112室(德胜园区)

申请人 奇智软件(北京)有限公司

(72) 发明人 罗正烨

(74) 专利代理机构 北京鼎佳达知识产权代理事  
务所(普通合伙) 11348

代理人 王伟锋 刘铁生

(51) Int. Cl.

G06F 9/50(2006. 01)

G06F 9/48(2006. 01)

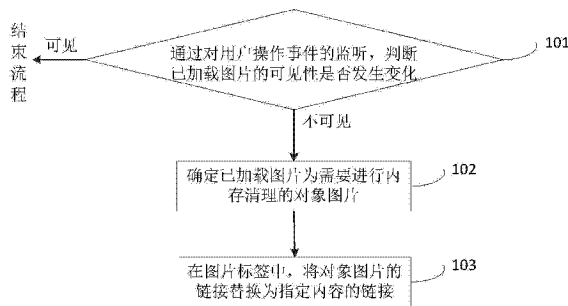
权利要求书2页 说明书10页 附图2页

(54) 发明名称

一种基于瀑布流技术的内存优化方法及装置

(57) 摘要

本发明公开了一种基于瀑布流技术的内存优化方法及装置,涉及互联网技术领域,解决了瀑布流技术过度占用内存的问题。本发明的方法包括:通过对用户操作事件的监听,当判断已加载图片变为不可见时,确定已加载图片为需要进行内存清理的对象图片;在图片标签中,将对象图片的链接替换为指定内容的链接,以便浏览器将指定内容加载到对象图片的位置上,并且将对象图片的数据从内存中清除掉,其中,指定内容的数据量小于对象图片的数据量。本发明能够降低瀑布流页面的内存占用。



1. 一种基于瀑布流技术的内存优化方法,其特征在于,所述方法包括:  
通过对用户操作事件的监听,判断已加载图片的可见性是否发生变化;  
当所述已加载图片变为不可见时,确定所述已加载图片为需要进行内存清理的对象图片;

在图片标签中,将所述对象图片的链接替换为指定内容的链接,以便浏览器将所述指定内容加载到所述对象图片的位置上,并且将所述对象图片的数据从内存中清除掉,其中,所述指定内容的数据量小于所述对象图片的数据量。

2. 根据权利要求1所述的方法,其特征在于,所述对用户操作事件的监听包括:对滚轮事件、滚动条事件或者方向键事件进行监听。

3. 根据权利要求1所述的方法,其特征在于,所述方法进一步包括:

统计用户浏览的图片数量;

当用户浏览的图片数量超过预设数值时,启动对用户操作事件的监听。

4. 根据权利要求1所述的方法,其特征在于,所述判断已加载图片的可见性是否发生变化,包括:在所述已加载图片移动出可视区域后,判断所述已加载图片变为不可见。

5. 根据权利要求1所述的方法,其特征在于,所述方法进一步包括:

将未加载过的图片确定为所述对象图片。

6. 根据权利要求1所述的方法,其特征在于,所述方法还包括:将所述对象图片的链接保存在内存中。

7. 根据权利要求6所述的方法,其特征在于,当所述对象图片重新变为可见时,所述方法进一步包括:

从内存中读取所述对象图片的链接;

在图片标签中,将所述指定内容的链接替换为所述对象图片的链接,以便所述浏览器重新获取并加载所述对象图片。

8. 根据权利要求7所述的方法,其特征在于,所述浏览器重新获取并加载所述对象图片,包括:

所述浏览器通过所述对象图片的链接从缓存中读取所述对象图片;

所述浏览器加载从缓存中读取的所述对象图片。

9. 根据权利要求8所述的方法,其特征在于,在所述浏览器通过所述对象图片的链接从缓存中读取所述对象图片之前,进一步包括:

所述浏览器将所述对象图片的链接发送给服务器;

所述浏览器通过所述对象图片的链接从缓存中读取所述对象图片,包括:

若所述服务器判断所述对象图片没有更新,则所述浏览器通过所述对象图片的链接从缓存中读取所述对象图片;

所述方法进一步包括:

若所述服务器判断所述对象图片存在更新,则所述浏览器向所述服务器请求更新后的对象图片。

10. 一种基于瀑布流技术的内存优化装置,其特征在于,所述装置包括:

监听单元,用于通过对用户操作事件的监听,判断已加载图片的可见性是否发生变化;

确定单元,用于当所述监听单元判断所述已加载图片变为不可见时,确定所述已加载图片为需要进行内存清理的对象图片;

替换单元,用于在图片标签中将所述确定单元确定的所述对象图片的链接替换为指定内容的链接,以便浏览器将所述指定内容加载到所述对象图片的位置上,并且将所述对象图片的数据从内存中清除掉,其中,所述指定内容的数据量小于所述对象图片的数据量。

## 一种基于瀑布流技术的内存优化方法及装置

### 技术领域

[0001] 本发明涉及互联网技术领域,特别是涉及一种基于瀑布流技术的内存优化方法及装置。

### 背景技术

[0002] 随着互联网技术的发展,浏览器展现网页的方式也呈现出多样性的特点,尤其是对于浏览图片信息来说,为了提升图片浏览的连贯性和便捷性,现有技术中多采用瀑布流的布局方式实现图片的加载。基于瀑布流技术,用户在浏览图片的过程中无需翻页,通过不断下拉滚动条或滑动滚轮的方式,用户可以对网页中的图片进行无限浏览。

[0003] 在现有的瀑布流技术中,发明人发现:用户浏览过的图片数据都会被存储到终端的内存中,随着用户的不断浏览,终端内存的占用将会越来越严重,使得终端无法运行其他任务。特别是对于移动终端而言,当有限的内存资源被过度占用时,移动终端的性能将会受到严重的影响,导致卡顿甚至死机等情况出现。

### 发明内容

[0004] 有鉴于此,本发明提出了一种基于瀑布流技术的内存优化方法及装置,主要目的在于解决瀑布流技术过度占用内存的问题。

[0005] 依据本发明的第一个方面,本发明提供了一种基于瀑布流技术的内存优化方法,包括:

[0006] 通过对用户操作事件的监听,判断已加载图片的可见性是否发生变化;

[0007] 当已加载图片变为不可见时,确定已加载图片为需要进行内存清理的对象图片;

[0008] 在图片标签中,将对象图片的链接替换为指定内容的链接,以便浏览器将指定内容加载到对象图片的位置上,并且将对象图片的数据从内存中清除掉,其中,指定内容的数据量小于对象图片的数据量。

[0009] 依据本发明的第二个方面,本发明提供了一种基于瀑布流技术的内存优化装置,该装置包括:

[0010] 监听单元,用于通过对用户操作事件的监听,判断已加载图片的可见性是否发生变化;

[0011] 确定单元,用于当监听单元判断已加载图片变为不可见时,确定已加载图片为需要进行内存清理的对象图片;

[0012] 替换单元,用于在图片标签中将确定单元确定的对象图片的链接替换为指定内容的链接,以便浏览器将指定内容加载到对象图片的位置上,并且将对象图片的数据从内存中清除掉,其中,指定内容的数据量小于对象图片的数据量。

[0013] 借由上述技术方案,本发明实施例提供的基于瀑布流技术的内存优化方法及装置,能够把不可见状态下的已加载图片确定为需要进行内存清理的对象图片,并在图片标签中将对象图片的链接替换为数据量较小的指定内容的链接,由于对象图片的链接被替换

掉了,因此可以使得浏览器将对象图片从内存中清理掉,因而与现有技术相比,能够及时释放已加载图片在内存中的占用空间。同时,由于使用指定内容对原对象图片的位置进行了占位,因此本发明还可以保持瀑布流页面的原有布局格式,避免页面出现上窜。

[0014] 上述说明仅是本发明技术方案的概述,为了能够更清楚了解本发明的技术手段,而可依照说明书的内容予以实施,并且为了让本发明的上述和其它目的、特征和优点能够更明显易懂,以下特举本发明的具体实施方式。

## 附图说明

[0015] 通过阅读下文优选实施方式的详细描述,各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。附图仅用于示出优选实施方式的目的,而并不认为是对本发明的限制。而且在整个附图中,用相同的参考符号表示相同的部件。在附图中:

[0016] 图 1 示出了本发明实施例提供的一种基于瀑布流技术的内存优化方法的流程图;

[0017] 图 2 示出了本发明实施例提供的判断已加载图片可见性的网页示意图;

[0018] 图 3 示出了本发明实施例提供的已加载图片由不可见变为可见时的显示页面示意图;

[0019] 图 4 示出了本发明实施例提供的一种基于瀑布流技术的内存优化装置的结构示意图。

## 具体实施方式

[0020] 下面将参照附图更加详细地描述本公开的示例性实施例。虽然附图中显示了本公开的示例性实施例,然而应当理解,可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反,提供这些实施例是为了能够更透彻地理解本公开,并且能够将本公开的范围完整的传达给本领域的技术人员。

[0021] 为了解决瀑布流技术过度占用内存的问题,本发明实施例提供了一种基于瀑布流技术的内存优化方法,如图 1 所示,该方法包括:

[0022] 101、通过对用户操作事件的监听,判断已加载图片的可见性是否发生变化。

[0023] 通常在用户浏览基于瀑布流技术的网页中的图片时,为了快速浏览大量的图片,用户会持续下拉网页侧边滑动条或滚动鼠标滚轮的操作,在操作的过程中,大量图片被加载并呈现在网页上,随着用户的继续操作,后续未浏览过的图片会被继续加载到页面的可视区域中。当图片出现在可视区域时,用户可以看到该图片;随着用户继续浏览图片,已加载图片被上翻移出可视区域后,用户看不到该图片。对于用户而言,图片的可见状态或不可见状态可以通过肉眼直接判断,但是对于客户端一侧而言,判断已加载图片的可见状态不可能像人一样用肉眼判断,其必须通过监听的方式判断已加载图片的可见状态。由于图片的加载与否与用户的操作有直接关系,所以可以通过对用户操作事件进行监听的方式判断已加载图片的可见状态。

[0024] 在实际的浏览图片过程中,由于用户浏览图片速度过快,可能存在返回操作重新查看已加载图片的情况。因此,已加载图片的状态不是一成不变的,一旦用户需要重新查看该图片时,就可以通过相应的操作使已加载的图片从不可见状态重新作为显示页面呈现可见状态。因此,这里所说的对用户操作事件进行监听是一个持续的过程,而且监听用户的操

作事件不只涉及监听图片可见与否的状态的改变,还涉及其他的操作,例如点击各种链接。

[0025] 102、已加载图片变为不可见时,确定已加载图片为需要进行内存清理的对象图片。

[0026] 在现有技术中,为了满足用户快速重新查看图片的需求,网页中已加载的图片通常被保存在内存中。随着用户浏览图片的数量的增加,内存占用会越来越大,因此可以适当释放内存中存储的图片数据。由于已加载的图片被用户看过,在绝大多数情况下,这些图片都是可以从内存中释放的图片。在已加载的图片中,如果已加载图片不可见时,则可以将其图片数据从内存中清理,因此不可见状态的已加载图片可以确定为需要进行内存清理的对象图片。

[0027] 103、在图片标签中,将对象图片的链接替换为指定内容的链接。

[0028] 通常情况下,网页上显示的图片并不是直接在网页中插入图片,而是在网页中链接图片。链接是一个网页指向一个目标的连接关系,所述的目标可以是另一个网页,也可以是相同网页上的不同位置,还可以是图片、电子邮件地址、文件甚至是应用程序。网页中进行链接的文本可以是一段文本或者是一个图片。用户可以点击已经链接的文本或图片,使链接目标显示在浏览器上。例如在HTML<img>标签中,<img src =“/i/eg\_tul ip. jpg”>的链接目标是 tul ip,即一幅郁金香图片。

[0029] 当步骤 102 确定了已加载图片为需要进行内存清理的对象图片后,可以执行步骤 103 将对象图片的链接替换为指定内容的链接,由于对象图片的链接被指定内容的链接替换掉了,所以利用浏览器自身的内存清理机制,可以将原来的对象图片的数据从内存中清除掉,该指定内容的数据量远小于原来的对象图片的数据量,通过这种实施方式释放内存中的数据,能够保证瀑布流页面的流畅运行。这里所说的指定内容,可以是一幅提示图片,也可以是原图片的链接地址或一条提示字符等。

[0030] 由于瀑布流页面的布局具有大量图片分别首尾相接、左右相连的显示特点,所以当对象图片的数据被浏览器自身的内存清理机制从内存中清除后,该对象图片在页面中的位置就会出现空位,位于该对象图片下方的图片会自动向上移动至被清除对象图片原来所占的位置处,当后续图片依次不断上移时就会导致页面上蹿。为了避免出现页面上蹿情况,本方案用数据量较小的指定内容替换掉需要清除的对象图片,实现对对象图片原始位置的占位,使得下方图片无法上移,从而保持原有的瀑布流页面布局格式,防止页面上蹿。

[0031] 本发明实施例提供的基于瀑布流技术的内存优化方法,能够把不可见状态下的已加载图片确定为需要进行内存清理的对象图片,并在图片标签中将对象图片的链接替换为数据量较小的指定内容的链接,由于对象图片的链接被替换掉了,因此可以使得浏览器将对象图片从内存中清理掉,因而与现有技术相比,本发明实施例提供的方法能够及时释放已加载图片在内存中的占用空间。同时,由于使用指定内容对原对象图片的位置进行了占位,因此本发明还可以保持瀑布流页面的原有布局格式,避免页面出现上蹿。

[0032] 进一步的,作为对上述实施例的细化和扩展,在上述步骤 101 对用户操作事件进行监听,通过这种方式判断已加载图片的可见状态时,监听的事件可以为滚轮滚动、滚动条滚动或者键盘上方向键的按压等。以滚轮滚动事件为例,通常情况下,监听操作是持续进行的过程,因此当用户滚动滚轮时,就会被监听到滚轮滚动的事件,从而触发判断图片是否移出可视区域的步骤。

[0033] 在用户浏览网页的过程中,刚浏览网页时加载的图片不是很多,内存占用很少,浏览相对顺畅,但随着浏览图片数量的增多,就需要对已加载图片进行适当的清理。因此,在另一种实现方式中,可以根据终端性能或浏览器性能预先设置浏览图片的数值,在浏览器加载网页时开始统计加载图片的数量,当统计用户浏览的图片数量超过预设的数值时,开始启动对用户操作事件的监听。

[0034] 在判断已加载图片的可见性是否发生变化时,判断已加载图片是否可见的依据在于,判断已加载图片是否位于可视区域中。本实施例所述的可视区域为网页页面中真正显示网页内容的区域,如图 2 所示,图 2 中虚线包围的区域为可视区域,不包括网页页面的边框以及工具栏。只要已加载图片位于图 2 所示的可视区域内,则确定该图片可见(例如图 2 中可以见到的 6 张图片),反之,不可见。需要说明的是,本实施例中所指的不可见的已加载图片,是已经被用户浏览过、并已被移出可视区域的图片,实际应用中,可视区域外的图片还包括未加载的图片,即可以简单理解为可视区域下方的、尚未被浏览的图片。

[0035] 当然,不只可以根据上述已加载图片是否处于可视区域来判断已加载图片是否需要内存清理的对象图片,在确定对象图片的范围上也可以根据其他区域来判定。例如,可以规定已加载图片处于两倍可视区域之外时,已加载图片被确定为需要内存清理的对象图片。所述两倍可视区域包括可视区域的一屏和可视区域以上或以下的一屏。处于两倍可视区域内的已加载图片和上述处于可视区域内的已加载图片一样,不会被作为对象图片处理。当图片处于可视区域以下部分且未被浏览过时,也可以将可视区域以下的未浏览图片作为对象图片进行处理。

[0036] 在浏览图片的过程中,从内存中获取图片信息能够提高获取图片的速度,但是如果把浏览过的图片信息都存储在内存中,又会导致内存占用越来越大,反而会影响浏览器的浏览性能,降低浏览速度。因此,为了既能从内存中快速获取图片信息又能不占用内存过多的容量,本实施例提供的一种实施方式可以将对象图片的链接保存在内存中,以便浏览器可以快速从内存中得到对象图片的链接并根据链接指定的位置获取对象图片。由于内存中存储的对象图片的链接只是几个字节的地址,与现有技术中存储的图片数据相比,占用的内存可以忽略不计。当浏览器通过内存中的对象图片的链接获取对象图片时,由于对象图片之前已经被加载过,因此浏览器可以从硬盘的缓存中获取对象图片,既节省了服务器处理资源又加快了获取图片的速度。

[0037] 在图片的浏览过程中,当用户重新查看某些图片时,即已加载的不可见状态的对象图片又处于可见状态时,由于已经进行了步骤 103 在图片标签中,将所述对象图片的链接替换为指定内容的链接,因此需要将图片标签中的指定内容的链接替换回保存在内存中的对象图片的链接,浏览器只有根据图片标签中的对象图片的链接才能加载出浏览过的原图片。即已加载图片处于不可见状态时,图片标签中记录有指定内容的链接,其中指定内容的数据量远小于对象图片的数据量;当已加载图片重新处于可见状态时,图片标签中的指定内容的链接被替换为对象图片的链接,其中对象图片的链接可以保存在内存中。

[0038] 在上述图片标签中的指定内容的链接被替换为对象图片的链接后,浏览器可以根据图片标签的内容获取并加载对象图片。浏览器具有缓存机制,会在硬盘上对最近请求过的文档、图片或网页等进行存储,这种缓存的方式节约了网络资源,提高了网络效率。虽然在步骤 103 中将已加载的图片数据已经从内存中清理掉,但是由于浏览器自身的缓存机制

使得内存中删除的图片数据还存储于缓存中,因此,在浏览器根据图片标签的内容获取并加载对象图片时可以利用自身的缓存机制从硬盘的缓存中获取对象图片,并对对象图片进行加载。

[0039] 在图片的浏览过程中还可能存在着一种情况,即服务器对已加载的图片进行了更新,如果存在着服务器更新已加载图片的情况,那么根据对象图片的链接就得不到更新后的图片。因此,基于这种情况,在浏览器通过对象图片的链接从缓存中读取对象图片的步骤之前,可以先将对象图片的链接发送给服务器,供服务器判断该对象图片是否存在更新,若对象图片没有更新,则向浏览器返回该对象图片的链接,以便浏览器通过该链接从缓存中读取该对象图片;若对象图片存在着更新,则可以向浏览器下发获取新图片的指令,浏览器根据该指令向服务器请求更新后的对象图片。

[0040] 当浏览器向服务器获取到更新对象图片后,与首次加载对象图片并处理对象图片数据的方式一样,更新对象图片的数据被保存在缓存里,替换掉缓存中原有的更新前的对象图片的数据。同样的,图片标签中的更新对象图片的链接被指定内容的链接替换掉,利用浏览器自身的内存清理机制自动将更新对象图片的数据从内存中清理掉,同时将更新对象图片的链接保存在内存中,以便再次加载该更新对象图片时,能快速将指定内容的链接替换回更新对象图片的链接。

[0041] 如图 3 所示,为了不影响瀑布流页面中的图片布局,以上实施例中所指的指定内容可以为提示图标或对象图片的链接地址,当然也可以为其他标记或符号,它们都能够起到占据原图片位置的作用。其中,提示图标无实际内容,通常可以为数据量很小的“裂图”图标或“加载”图标。当然,为了更清楚的显示各自图片的区别并方便用户按照需要打开相应的图片,可以进一步的在提示图标下方或上方添加链接地址或简短的图片描述,例如图片是关于风景的,就可以添加“scenery”;图片是关于人物的,就可以添加“figure”;图片是关于动物的,就可以添加“animal”等,用户通过描述可以点击相应的提示图标的链接地址获取需要查看的图片。

[0042] 本发明实施例在浏览器通过对象图片的链接从缓存中读取对象图片之前,将对象图片的链接发送给服务器,供服务器判断该对象图片是否有更新,若没有更新,则指示浏览器从缓存中读取对象图片;若有更新,则指示浏览器向服务器请求更新的对象图片,从而保证了浏览器获取图片的准确性。同时,使用提示图标或链接地址作为指定内容并附加描述信息,不仅在页面中能起到图片占位作用,而且能够使用户根据需要查看图片,避免浏览器再次获取全部的已加载图片。

[0043] 进一步的,作为对上述方法的实现,本发明实施例还提供了一种基于瀑布流技术的内存优化装置,该装置可以位于终端中,也可以独立于终端但与终端之间具有数据交互关系,用以实现上述实施例提供的不同实施方式所示的方法。如图 4 所示,该装置包括:监听单元 41、确定单元 42 以及替换单元 43,其中,

[0044] 监听单元 41,用于通过对用户操作事件的监听,判断已加载图片的可见性是否发生变化;

[0045] 确定单元 42,用于当监听单元 41 判断已加载图片变为不可见时,确定已加载图片为需要进行内存清理的对象图片;

[0046] 替换单元 43,用于在图片标签中将确定单元 42 确定的对象图片的链接替换为指



定内容的链接,以便浏览器将指定内容加载到对象图片的位置上,并且将对象图片的数据从内存中清除掉,其中,指定内容的数据量小于对象图片的数据量。

[0047] 进一步的,该监听单元 41 对用户操作事件的监听包括:对滚轮事件、滚动条事件或者方向键事件进行监听。

[0048] 进一步的,该装置还包括:统计单元 44,用于统计用户浏览的图片数量。该监听单元 41 当统计单元 44 统计用户浏览的图片数量超过预设数值时,启动对用户操作事件的监听。

[0049] 进一步的,该监听单元 41 用于在已加载图片移动出可视区域后,判断已加载图片变为不可见。

[0050] 进一步的,该确定单元 42 用于将未加载过的图片确定为对象图片。

[0051] 进一步的,该装置还包括:保存单元 45,用于将确定单元 42 确定的对象图片的链接保存在内存中。

[0052] 进一步的,该装置还包括:读取单元 46,用于当监听单元 41 判断对象图片重新变为可见时,从内存中读取对象图片的链接。

[0053] 该替换单元 43 还用于在图片标签中将指定内容的链接替换为对象图片的链接,以便浏览器重新获取并加载对象图片。

[0054] 进一步的,该替换单元 43 替换的指定内容为提示图标,该提示图标用于提示对象图片未加载。

[0055] 进一步的,该替换单元 43 替换的指定内容为对象图片的链接地址。

[0056] 进一步的,该装置还包括:加载单元 47;

[0057] 该监听单元 41,用于判断对象图片重新变为可见时,监听用户是否点击该对象图片的链接地址;

[0058] 该加载单元 47 用于当监听单元 41 监听到用户点击对象图片的链接地址时,通过浏览器重新获取并加载对象图片。

[0059] 本发明实施例提供的基于瀑布流技术的内存优化装置,能够把不可见状态下的已加载图片确定为需要进行内存清理的对象图片,并在图片标签中将对象图片的链接替换为数据量较小的指定内容的链接,由于对象图片的链接被替换掉了,因此可以使得浏览器将对象图片从内存中清理掉,因而与现有技术相比,本发明实施例提供的装置能够及时释放已加载图片在内存中的占用空间。同时,由于使用指定内容对原对象图片的位置进行了占位,因此本发明还可以保持瀑布流页面的原有布局格式,避免页面出现上窜。

[0060] 此外,本发明实施例在浏览器通过对象图片的链接从缓存中读取对象图片之前,将对象图片的链接发送给服务器,供服务器判断该对象图片是否有更新,若没有更新,则指示浏览器从缓存中读取对象图片;若有更新,则指示浏览器向服务器请求更新的对象图片,从而保证了浏览器获取图片的准确性。同时,使用提示图标或链接地址作为指定内容并附加描述信息,不仅在页面中能起到图片占位作用,而且能够使用户根据需要查看图片,避免浏览器再次获取全部的已加载图片。

[0061] 本发明的实施例公开了:

[0062] A1、一种基于瀑布流技术的内存优化方法,其特征在于,所述方法包括:

[0063] 通过对用户操作事件的监听,判断已加载图片的可见性是否发生变化;

[0064] 当所述已加载图片变为不可见时,确定所述已加载图片为需要进行内存清理的对象图片;

[0065] 在图片标签中,将所述对象图片的链接替换为指定内容的链接,以便浏览器将所述指定内容加载到所述对象图片的位置上,并且将所述对象图片的数据从内存中清除掉,其中,所述指定内容的数据量小于所述对象图片的数据量。

[0066] A2、根据权利要求 A1 所述的方法,其特征在于,所述对用户操作事件的监听包括:对滚轮事件、滚动条事件或者方向键事件进行监听。

[0067] A3、根据权利要求 A1 所述的方法,其特征在于,所述方法进一步包括:

[0068] 统计用户浏览的图片数量;

[0069] 当用户浏览的图片数量超过预设数值时,启动对用户操作事件的监听。

[0070] A4、根据权利要求 A1 所述的方法,其特征在于,所述判断已加载图片的可见性是否发生变化,包括:在所述已加载图片移动出可视区域后,判断所述已加载图片变为不可见。

[0071] A5、根据权利要求 A1 所述的方法,其特征在于,所述方法进一步包括:

[0072] 将未加载过的图片确定为所述对象图片。

[0073] A6、根据权利要求 A1 所述的方法,其特征在于,所述方法还包括:将所述对象图片的链接保存在内存中。

[0074] A7、根据权利要求 A6 所述的方法,其特征在于,当所述对象图片重新变为可见时,所述方法进一步包括:

[0075] 从内存中读取所述对象图片的链接;

[0076] 在图片标签中,将所述指定内容的链接替换为所述对象图片的链接,以便所述浏览器重新获取并加载所述对象图片。

[0077] A8、根据权利要求 A7 所述的方法,其特征在于,所述浏览器重新获取并加载所述对象图片,包括:

[0078] 所述浏览器通过所述对象图片的链接从缓存中读取所述对象图片;

[0079] 所述浏览器加载从缓存中读取的所述对象图片。

[0080] A9、根据权利要求 A8 所述的方法,其特征在于,在所述浏览器通过所述对象图片的链接从缓存中读取所述对象图片之前,进一步包括:

[0081] 所述浏览器将所述对象图片的链接发送给服务器;

[0082] 所述浏览器通过所述对象图片的链接从缓存中读取所述对象图片,包括:

[0083] 若所述服务器判断所述对象图片没有更新,则所述浏览器通过所述对象图片的链接从缓存中读取所述对象图片;

[0084] 所述方法进一步包括:

[0085] 若所述服务器判断所述对象图片存在更新,则所述浏览器向所述服务器请求更新后的对象图片。

[0086] A10、根据权利要求 A1 至 A9 中任一项所述的方法,其特征在于,所述指定内容为提示图标,所述提示图标用于提示所述对象图片未加载。

[0087] A11、根据权利要求 A1 至 A6 中任一项所述的方法,其特征在于,所述指定内容为所述对象图片的链接地址。

[0088] A12、根据权利要求 A11 所述的方法，其特征在于，当所述对象图片重新变为可见时，所述方法进一步包括：

[0089] 若用户点击所述对象图片的链接地址，则通过所述浏览器重新获取并加载所述对象图片。

[0090] B13、一种基于瀑布流技术的内存优化装置，其特征在于，所述装置包括：

[0091] 监听单元，用于通过对用户操作事件的监听，判断已加载图片的可见性是否发生变化；

[0092] 确定单元，用于当所述监听单元判断所述已加载图片变为不可见时，确定所述已加载图片为需要进行内存清理的对象图片；

[0093] 替换单元，用于在图片标签中将所述确定单元确定的所述对象图片的链接替换为指定内容的链接，以便浏览器将所述指定内容加载到所述对象图片的位置上，并且将所述对象图片的数据从内存中清除掉，其中，所述指定内容的数据量小于所述对象图片的数据量。

[0094] B14、根据权利要求 B13 所述的装置，其特征在于，所述监听单元对用户操作事件的监听包括：对滚轮事件、滚动条事件或者方向键事件进行监听。

[0095] B15、根据权利要求 B13 所述的装置，其特征在于，所述装置还包括：

[0096] 统计单元，用于统计用户浏览的图片数量；

[0097] 所述监听单元当所述统计单元统计所述用户浏览的图片数量超过预设数值时，启动对用户操作事件的监听。

[0098] B16、根据权利要求 B13 所述的装置，其特征在于，所述监听单元用于在所述已加载图片移动出可视区域后，判断所述已加载图片变为不可见。

[0099] B17、根据权利要求 B13 所述的装置，其特征在于，所述确定单元用于将未加载过的图片确定为所述对象图片。

[0100] B18、根据权利要求 B13 所述的装置，其特征在于，所述装置还包括：保存单元，用于将所述确定单元确定的所述对象图片的链接保存在内存中。

[0101] B19、根据权利要求 B18 所述的装置，其特征在于，所述装置还包括：

[0102] 读取单元，用于当所述监听单元判断所述对象图片重新变为可见时，从内存中读取所述对象图片的链接；

[0103] 所述替换单元还用于在图片标签中将所述指定内容的链接替换为所述对象图片的链接，以便所述浏览器重新获取并加载所述对象图片。

[0104] B20、根据权利要求 B13 至 B19 中任一项所述的装置，其特征在于，所述替换单元替换的所述指定内容为提示图标，所述提示图标用于提示所述对象图片未加载。

[0105] B21、根据权利要求 B13 至 B18 中任一项所述的装置，其特征在于，所述替换单元替换的所述指定内容为所述对象图片的链接地址。

[0106] B22、根据权利要求 B21 所述的装置，其特征在于，所述装置还包括：

[0107] 加载单元；

[0108] 所述监听单元，用于判断所述对象图片重新变为可见时，监听用户是否点击所述对象图片的链接地址；

[0109] 所述加载单元，用于当所述监听单元监听到用户点击所述对象图片的链接地址

时,通过所述浏览器重新获取并加载所述对象图片。

[0110] 在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中未详述的部分,可以参见其他实施例的相关描述。

[0111] 可以理解的是,上述方法及装置中的相关特征可以相互参考。另外,上述实施例中的“第一”、“第二”等是用于区分各实施例,而并不代表各实施例的优劣。

[0112] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统、装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0113] 在此提供的算法和显示不与任何特定计算机、虚拟系统或者其它设备固有相关。各种通用系统也可以与基于在此的示教一起使用。根据上面的描述,构造这类系统所要求的结构是显而易见的。此外,本发明也不针对任何特定编程语言。应当明白,可以利用各种编程语言实现在此描述的本发明的内容,并且上面对特定语言所做的描述是为了披露本发明的最佳实施方式。

[0114] 在此处所提供的说明书中,说明了大量具体细节。然而,能够理解,本发明的实施例可以在没有这些具体细节的情况下实践。在一些实例中,并未详细示出公知的方法、结构和技术,以便不模糊对本说明书的理解。

[0115] 类似地,应当理解,为了精简本公开并帮助理解各个发明方面中的一个或多个,在上面对本发明的示例性实施例的描述中,本发明的各个特征有时被一起分组到单个实施例、图、或者对其的描述中。然而,并不应将该公开的方法解释成反映如下意图:即所要求保护的本发明要求比在每个权利要求中所明确记载的特征更多的特征。更确切地说,如下的权利要求书所反映的那样,发明方面在于少于前面公开的单个实施例的所有特征。因此,遵循具体实施方式的权利要求书由此明确地并入该具体实施方式,其中每个权利要求本身都作为本发明的单独实施例。

[0116] 本领域那些技术人员可以理解,可以对实施例中的设备中的模块进行自适应性地改变并且把它们设置在与该实施例不同的一个或多个设备中。可以把实施例中的模块或单元或组件组合成一个模块或单元或组件,以及此外可以把它分成多个子模块或子单元或子组件。除了这样的特征和/或过程或者单元中的至少一些是相互排斥之外,可以采用任何组合对本说明书(包括伴随的权利要求、摘要和附图)中公开的所有特征以及如此公开的任何方法或者设备的所有过程或单元进行组合。除非另外明确陈述,本说明书(包括伴随的权利要求、摘要和附图)中公开的每个特征可以由提供相同、等同或相似目的的替代特征来代替。

[0117] 此外,本领域的技术人员能够理解,尽管在此所述的一些实施例包括其它实施例中包括的某些特征而不是其它特征,但是不同实施例的特征的组合意味着处于本发明的范围之内并且形成不同的实施例。例如,在下面的权利要求书中,所要求保护的实施例的任意之一都可以以任意的组合方式来使用。

[0118] 本发明的各个部件实施例可以以硬件实现,或者以在一个或者多个处理器上运行的软件模块实现,或者以它们的组合实现。本领域的技术人员应当理解,可以在实践中使用微处理器或者数字信号处理器(DSP)来实现根据本发明实施例的发明名称(如确定网站内链接等级的装置)中的一些或者全部部件的一些或者全部功能。本发明还可以实现为用于执行这里所描述的方法的一部分或者全部的设备或者装置程序(例如,计算机程序和计算

机程序产品)。这样的实现本发明的程序可以存储在计算机可读介质上,或者可以具有一个或者多个信号的形式。这样的信号可以从因特网网站上下载得到,或者在载体信号上提供,或者以任何其他形式提供。

[0119] 应该注意的是上述实施例对本发明进行说明而不是对本发明进行限制,并且本领域技术人员在不脱离所附权利要求的范围的情况下可设计出替换实施例。在权利要求中,不应将位于括号之间的任何参考符号构造成对权利要求的限制。单词“包含”不排除存在未列在权利要求中的元件或步骤。位于元件之前的单词“一”或“一个”不排除存在多个这样的元件。本发明可以借助于包括有若干不同元件的硬件以及借助于适当编程的计算机来实现。在列举了若干装置的单元权利要求中,这些装置中的若干个可以是通过同一个硬件项来具体体现。单词第一、第二、以及第三等的使用不表示任何顺序。可将这些单词解释为名称。

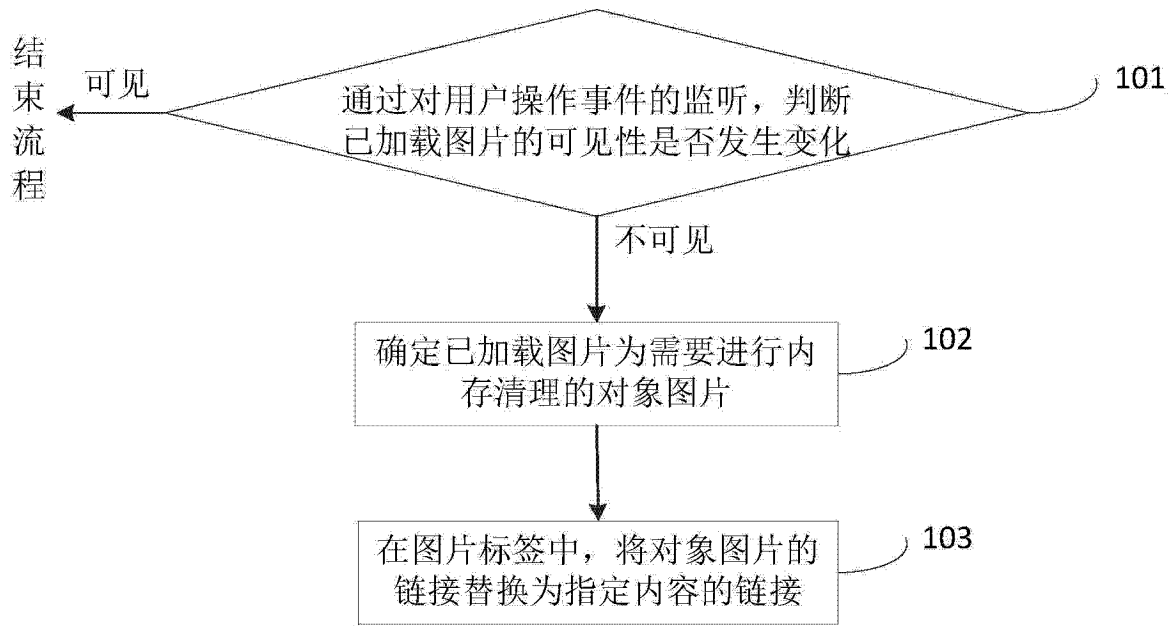


图 1

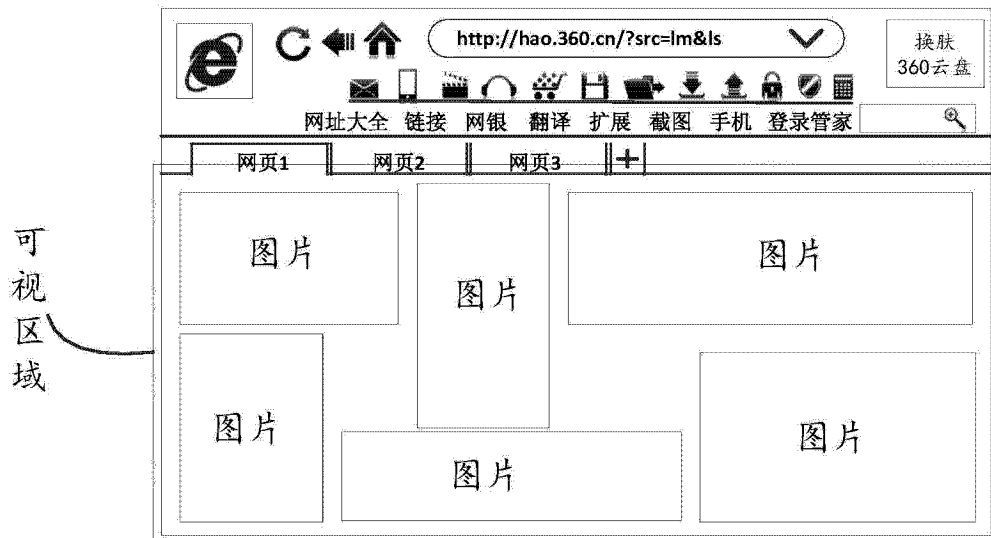


图 2

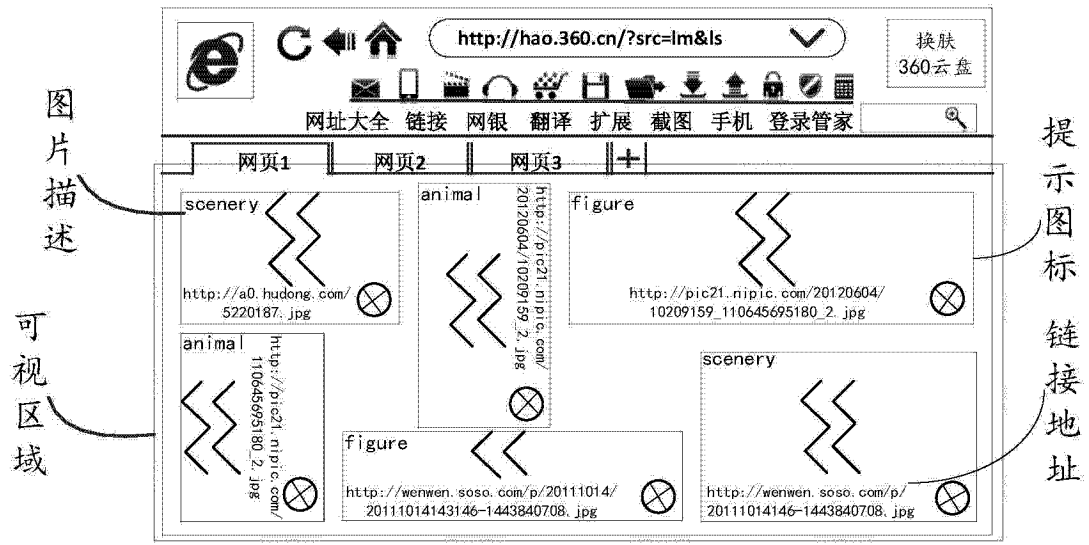


图 3

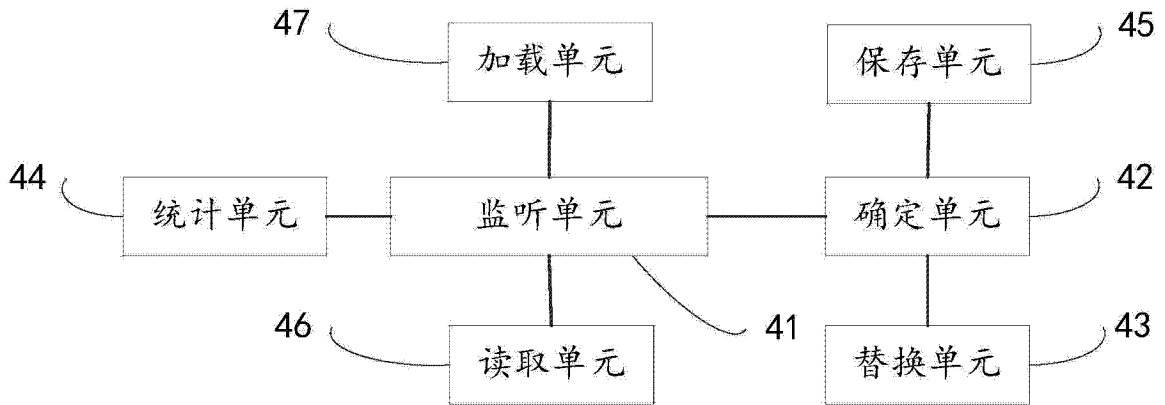


图 4