(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112395535 A
(43)申请公布日 2021.02.23

(21)申请号 201910754124.1

(22)申请日 2019.08.15

(71)申请人 腾讯科技(深圳)有限公司
地址 518000 广东省深圳市南山区高新区科技中一路腾讯大厦35层

(72)发明人 莫宝军

(74)专利代理机构 深圳市隆天联鼎知识产权代理有限公司 44232
代理人 刘抗美

(51)Int.Cl.
G06F 16/958(2019.01)
G06F 16/957(2019.01)
G06F 9/445(2018.01)

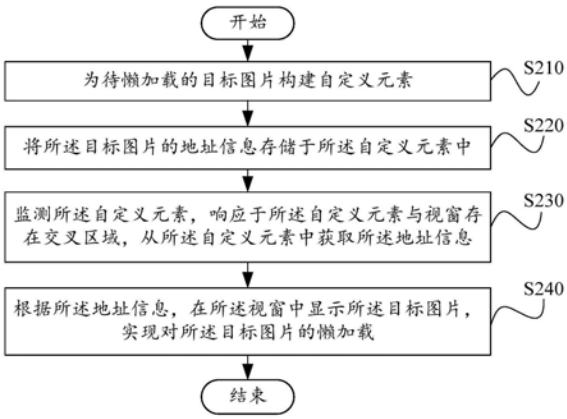
权利要求书2页 说明书12页 附图5页

(54)发明名称

图片的懒加载方法及装置、介质和电子设备

(57)摘要

本公开涉及网页优化技术领域，提供了一种图片的懒加载方法、图片的懒加载装置，以及实现上述图片的懒加载方法的计算机可读存储介质及电子设备。其中，该方法包括：为待懒加载的目标图片构建自定义元素；将目标图片的地址信息存储于自定义元素中；监测自定义元素，响应于自定义元素与视窗存在交叉区域，从自定义元素中获取上述地址信息；根据地址信息在视窗中显示上述目标图片，实现对目标图片的懒加载。本技术方案无需监听滚动事件，从而避免了监听滚动事件造成的计算资源的浪费，有利于提升图片的懒加载效率。同时，缓解浏览器的图片加载压力并增强用户浏览体验。



CN 112395535 A

1.一种图片的懒加载方法,其特征在于,所述方法包括:

为待懒加载的目标图片构建自定义元素;

将所述目标图片的地址信息存储于所述自定义元素中;

监测所述自定义元素,响应于所述自定义元素与视窗存在交叉区域,从所述自定义元素中获取所述地址信息;

根据所述地址信息,在所述视窗中显示所述目标图片,实现对所述目标图片的懒加载。

2.根据权利要求1所述的图片的懒加载方法,其特征在于,

所述为待懒加载的目标图片构建自定义元素,包括:

为待懒加载的目标图片构建超文本标记语言格式的自定义元素;

在所述自定义元素中设置钩子函数,以用于绑定交叉观察器和所述自定义元素;

所述监测所述自定义元素,包括:

通过所述交叉观察器检测所述自定义元素。

3.根据权利要求2所述的图片的懒加载方法,其特征在于,所述将所述目标图片的地址信息存储于所述自定义元素中,包括:

将所述目标图片的地址信息存储于所述自定义元素的任一属性中。

4.根据权利要求1至3中任意一项所述的图片的懒加载方法,其特征在于,所述根据所述地址信息,在所述视窗中显示所述目标图片,包括:

实例化图片标签对象;

从所述自定义元素中获取所述地址信息,并将所述地址信息赋值给所述图片标签对象的属性,以在所述视窗中显示所述目标图片。

5.根据权利要求1至3中任意一项所述的图片的懒加载方法,其特征在于,

所述为待懒加载的目标图片构建自定义元素,包括:

为待懒加载的目标图片构建超文本标记语言格式的自定义元素;

通过所述自定义元素继承超文本标记语言的元素类;

所述从所述自定义元素中获取所述地址信息,包括:

通过超文本标记语言的元素类的属性获取方法获取所述地址信息。

6.根据权利要求5所述的图片的懒加载方法,其特征在于,所述根据所述地址信息,在所述视窗中显示所述目标图片,包括:

实例化图片标签对象;

通过超文本标记语言的元素类的属性获取方法从所述自定义元素中获取所述地址信息,并将所述地址信息赋值给所述图片标签对象的属性;

通过超文本标记语言的元素类的子节点添加方法,将赋值后的图片标签对像添加至所述自定义元素的孩子节点中,以在所述视窗中显示所述目标图片。

7.根据权利要求2所述的图片的懒加载方法,其特征在于,在所述视窗中显示所述目标图片之后,所述方法还包括:

解绑所述交叉观察器和所述自定义元素,以结束对所述自定义元素的监测。

8.一种图片的懒加载装置,其特征在于,所述装置包括:

自定义元素构建模块,被配置为为待懒加载的目标图片构建自定义元素;

图片地址存储模块,被配置为将所述目标图片的地址信息存储于所述自定义元素中;

自定义元素监测模块，被配置为监测所述自定义元素，响应于所述自定义元素与视窗存在交叉区域，从所述自定义元素中获取所述地址信息；

图片显示模块，被配置为根据所述地址信息，在所述视窗中显示所述目标图片，实现对所述目标图片的懒加载。

9.一种计算机存储介质，其特征在于，其上存储有计算机程序，所述计算机程序被处理器执行时实现如权利要求1至7中任意一项所述的图片的懒加载方法。

10.一种电子设备，其特征在于，所述电子设备包括：

一个或多个处理器；

存储装置，用于存储一个或多个程序，当所述一个或多个程序被所述一个或多个处理器执行时，使得所述一个或多个处理器实现如权利要求1至7中任意一项所述的图片的懒加载方法。

图片的懒加载方法及装置、介质和电子设备

技术领域

[0001] 本公开涉及网页优化技术领域，具体而言，涉及一种图片的懒加载方法、图片的懒加载装置，以及实现上述图片的懒加载方法的计算机可读存储介质和电子设备。

背景技术

[0002] 图片作为一种网络资源，在被请求时将占用网络资源。若一次性将整个页面的所有图片加载完，将需耗费较长的首屏加载时间。为了解决这种问题，通过网页前端仅加载在浏览器当前视窗内出现时图片，达到减少首屏加载时间的技术称为“图片懒加载”。

[0003] 相关技术提供的图片懒加载方法包括：图片标签的源文件(如，src属性)中不赋图片地址，并在图片标签上自定义一个属性来存放图片地址。在逻辑层初始的执行流程中，对所有图片进行一次额外的扫描，使得已经在视区的图片能正常显示，相当于手动触发一次滚动事件。监听页面的滚动事件。在滚动事件的回调里面选择页面所有的图片标签，依次判断是否在视区范围内，如果在视区范围，则设置图片标签的属性值为图片地址，实现图片的显示。

[0004] 然而，相关技术提供的图片的懒加载方案计算量大，导致图片的懒加载效率低。

[0005] 需要说明的是，上述背景技术部分公开的信息仅用于加强对本公开的背景的理解。

发明内容

[0006] 本公开的目的在于提供一种图片的懒加载方法、图片的懒加载装置、计算机存储介质及电子设备，能够避免监听滚动事件导致的计算资源的浪费，进而至少在一定程度上提升对图片的懒加载效率。

[0007] 本公开的其他特性和优点将通过下面的详细描述变得显然，或部分地通过本公开的实践而习得。

[0008] 根据本公开的一个方面，提供一种图片的懒加载方法，包括：

[0009] 为待懒加载的目标图片构建自定义元素；将上述目标图片的地址信息存储于上述自定义元素中；监测上述自定义元素，响应于上述自定义元素与视窗存在交叉区域，从上述自定义元素中获取上述地址信息；根据上述地址信息，在上述视窗中显示上述目标图片，实现对上述目标图片的懒加载。

[0010] 在本公开的一些实施例中，基于前述方案，上述为待懒加载的目标图片构建自定义元素，包括：为待懒加载的目标图片构建超文本标记语言格式的自定义元素；在上述自定义元素中设置钩子函数，以用于绑定交叉观察器和上述自定义元素；

[0011] 上述监测上述自定义元素，包括：通过上述交叉观察器检测上述自定义元素。

[0012] 在本公开的一些实施例中，基于前述方案，上述将上述目标图片的地址信息存储于上述自定义元素中，包括：将上述目标图片的地址信息存储于上述自定义元素的任一属性中。

[0013] 在本公开的一些实施例中，基于前述方案，上述根据上述地址信息，在上述视窗中显示上述目标图片，包括：实例化图片标签对象；从上述自定义元素中获取上述地址信息，并将上述地址信息赋值给上述图片标签对象的属性，以在上述视窗中显示上述目标图片。

[0014] 在本公开的一些实施例中，基于前述方案，上述为待懒加载的目标图片构建自定义元素，包括：为待懒加载的目标图片构建超文本标记语言格式的自定义元素；通过上述自定义元素继承超文本标记语言的元素类；

[0015] 上述从上述自定义元素中获取上述地址信息，包括：通过超文本标记语言的元素类的属性获取方法获取上述地址信息。

[0016] 在本公开的一些实施例中，基于前述方案，上述根据上述地址信息，在上述视窗中显示上述目标图片，包括：实例化图片标签对象；通过超文本标记语言的元素类的属性获取方法从上述自定义元素中获取上述地址信息，并将上述地址信息赋值给上述图片标签对象的属性；通过超文本标记语言的元素类的子节点添加方法，将赋值后的图片标签对像添加至上述自定义元素的孩子节点中，以在上述视窗中显示上述目标图片。

[0017] 在本公开的一些实施例中，基于前述方案，在上述视窗中显示上述目标图片之后，上述方法还包括：解绑上述交叉观察器和上述自定义元素，以结束对上述自定义元素的监测。根据本公开的一个方面，提供了一种图片的懒加载装置，包括：

[0018] 自定义元素构建模块，被配置为为待懒加载的目标图片构建自定义元素；图片地址存储模块，被配置为将上述目标图片的地址信息存储于上述自定义元素中；自定义元素监测模块，被配置为监测上述自定义元素，响应于上述自定义元素与视窗存在交叉区域，从上述自定义元素中获取上述地址信息；以及，图片显示模块，被配置为根据上述地址信息，在上述视窗中显示上述目标图片，实现对上述目标图片的懒加载。

[0019] 在本公开的一些实施例中，基于前述方案，上述自定义元素构建模块，包括：第一构建单元和钩子函数设置单元。

[0020] 其中，上述第一构建单元被配置为：为待懒加载的目标图片构建超文本标记语言格式的自定义元素；以及，上述钩子函数设置单元被配置为：在上述自定义元素中设置钩子函数，以用于绑定交叉观察器和上述自定义元素。

[0021] 在此基础上，上述自定义元素监控模块，具体被配置为：通过上述交叉观察器检测上述自定义元素。

[0022] 在本公开的一些实施例中，基于前述方案，上述图片地址存储模块，被具体配置为：将上述目标图片的地址信息存储于上述自定义元素的任一属性中。

[0023] 在本公开的一些实施例中，基于前述方案，上述自定义元素监测模块，包括：第一图片标签对象实例化单元和图片显示单元。

[0024] 其中，上述第一图片标签对象实例化单元被配置为：实例化图片标签对象；以及，上述图片显示单元被配置为：从上述自定义元素中获取上述地址信息，并将上述地址信息赋值给上述图片标签对象的属性，以在上述视窗中显示上述目标图片。

[0025] 在本公开的一些实施例中，基于前述方案，上述自定义元素构建模块，包括：第二构建单元和类继承单元。

[0026] 其中，上述第二构建单元被配置为：为待懒加载的目标图片构建超文本标记语言格式的自定义元素；以及，上述类继承单元被配置为：通过上述自定义元素继承超文本标记

语言的元素类。

[0027] 在此基础上，上述自定义元素监控模块，具体被配置为：通过超文本标记语言的元素类的属性获取方法获取上述地址信息。

[0028] 本公开的一些实施例中，基于前述方案，上述自定义元素监测模块，包括：第二图片标签对象实例化单元、赋值单元和孩子节点添加单元。

[0029] 其中，上述第二图片标签对象实例化单元被配置为：实例化图片标签对象；上述赋值单元被配置为：通过超文本标记语言的元素类的属性获取方法从上述自定义元素中获取上述地址信息，并将上述地址信息赋值给上述图片标签对象的属性；以及，上述孩子节点添加单元被配置为：通过超文本标记语言的元素类的子节点添加方法，将赋值后的图片标签对像添加至上述自定义元素的孩子节点中，以在上述视窗中显示上述目标图片。

[0030] 本公开的一些实施例中，基于前述方案，上述图片的懒加载装置包括：解绑模块。

[0031] 其中，上述解绑模块被配置为：在上述图片显示模块在视窗中显示上述目标图片之后，解绑上述交叉观察器和上述自定义元素，以结束对上述自定义元素的监测。

[0032] 根据本公开的一个方面，提供了一种计算机存储介质，其上存储有计算机程序，上述计算机程序被处理器执行时实现上述第一方面上述的图片的懒加载方法。

[0033] 根据本公开的一个方面，提供一种电子设备，包括：处理器；以及存储器，用于存储上述处理器的可执行指令；其中，上述处理器配置为经由执行上述可执行指令来执行上述第一方面上述的图片的懒加载方法。

[0034] 由上述技术方案可知，本公开示例性实施例中的图片的懒加载方法、图片的懒加载装置、计算机存储介质及电子设备至少具备以下优点和积极效果：

[0035] 在本公开的一些实施例所提供的技术方案中，为待懒加载的目标图片构建自定义元素，如超文本标记语言(Hyper Text Markup Language，简称：HTML)，并将图片的地址信息存储于上述自定义元素中。由于存储图片地址信息的为自定义元素，因此图片并不会被加载。而是通过监测自定义元素，并在自定义元素与视窗(viewport)存在交叉区域时，获取所述图片的地址信息实现图片的显示。可见，本技术方案通过自定义元素和交叉观察的方式实现高效优雅地图片懒加载，而无需监听滚动事件，从而避免了监听滚动事件造成的计算资源的浪费，有利于提升图片的懒加载效率。进而，缓解了浏览器的图片加载压力，增强用户浏览体验。

[0036] 本公开应当理解的是，以上的一般描述和后文的细节描述仅是示例性和解释性的，并不能限制本公开。

附图说明

[0037] 此处的附图被并入说明书中并构成本说明书的一部分，示出了符合本公开的实施例，并与说明书一起用于解释本公开的原理。显而易见地，下面描述中的附图仅仅是本公开的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。在附图中：

[0038] 图1示出了可以应用本公开实施例的一种图片的懒加载方法及装置的示例性应用环境的系统架构的示意图；

[0039] 图2示出本公开一示例性实施例中图片的懒加载方法的流程示意图；

[0040] 图3示出本公开一示例性实施例中自定义元素的构建方法的流程示意图；

[0041] 图4示出的本公开一实施例中自定义元素与视窗位置关系的示意图；

[0042] 图5示出了本公开一示例性实施例中目标图片的显示方法的流程示意图；

[0043] 图6示出本公开另一示例性实施例中自定义元素的构建方法的流程示意图；

[0044] 图7示出了本公开另一示例性实施例中目标图片的显示方法的流程示意图；

[0045] 图8示出本公开示例性实施例中图片的懒加载装置的结构示意图；

[0046] 以及，

[0047] 图9示出本公开示例性实施例中电子设备的结构示意图。

具体实施方式

[0048] 现在将参考附图更全面地描述示例实施方式。然而，示例实施方式能够以多种形式实施，且不应被理解为限于在此阐述的范例；相反，提供这些实施方式使得本公开将更加全面和完整，并将示例实施方式的构思全面地传达给本领域的技术人员。

[0049] 此外，所描述的特征、结构或特性可以以任何合适的方式结合在一个或更多实施例中。在下面的描述中，提供许多具体细节从而给出对本公开的实施例的充分理解。然而，本领域技术人员将意识到，可以实践本公开的技术方案而没有特定细节中的一个或更多，或者可以采用其它的方法、组元、装置、步骤等。在其它情况下，不详细示出或描述公知方法、装置、实现或者操作以避免模糊本公开的各方面。

[0050] 附图中所示的方框图仅仅是功能实体，不一定必须与物理上独立的实体相对应。即，可以采用软件形式来实现这些功能实体，或在一个或多个硬件模块或集成电路中实现这些功能实体，或在不同网络和/或处理器装置和/或微控制器装置中实现这些功能实体。

[0051] 附图中所示的流程图仅是示例性说明，不是必须包括所有的内容和操作/步骤，也不是必须按所描述的顺序执行。例如，有的操作/步骤还可以分解，而有的操作/步骤可以合并或部分合并，因此实际执行的顺序有可能根据实际情况改变。

[0052] 图1示出了可以应用本公开实施例的一种图片的懒加载方法及装置的示例性应用环境的系统架构的示意图。

[0053] 如图1所示，系统架构100可以包括终端设备101、102、103中的一个或多个，网络104和服务器105。网络104用以在终端设备101、102、103和服务器105之间提供通信链路的介质。网络104可以包括各种连接类型，例如有线、无线通信链路或者光纤电缆等等。终端设备101、102、103可以是具有显示屏的各种电子设备，包括但不限于台式计算机、便携式计算机、智能手机和平板电脑等等。应该理解，图1中的终端设备、网络和服务器的数目仅仅是示意性的。根据实现需要，可以具有任意数目的终端设备、网络和服务器。比如服务器105可以是多个服务器组成的服务器集群等。

[0054] 本公开实施例所提供的图片的懒加载方法一般由服务器105执行，相应地，图片的懒加载装置一般设置于服务器105中。但本领域技术人员容易理解的是，本公开实施例所提供的图片的懒加载方法也可以由终端设备101、102、103执行，相应的，图片的懒加载装置也可以设置于终端设备101、102、103中，本示例性实施例中对此不做特殊限定。

[0055] 举例而言，在一种示例性实施例中，可以是终端设备101、102、103确定目标图片以及图片的地址信息并发送至服务器105，从而，服务器105为待懒加载的目标图片构建自定

义元素;然后,服务器105将上述目标图片的地址信息存储于上述自定义元素中。进一步地,服务器105监测上述自定义元素,响应于上述自定义元素与视窗存在交叉区域,从上述自定义元素中获取上述地址信息;服务器105根据上述地址信息,在上述视窗中显示上述目标图片,实现对上述目标图片的懒加载。服务器105还可以将图片懒加载后的视窗发送至终端设备101、102、103,从而,用户可以通过终端设备101、102、103方便地观看上述图片懒加载后的视窗。

[0056] 应该理解,图1中的终端、网络和服务端的数目仅仅是示意性的。根据实现需要,可以具有任意数目的终端、网络和服务端。

[0057] 相关技术提供的对图片的懒加载方案中,通过页面的监听滚动事件,以进一步计算出现在视窗范围内的图片。然而,一般情况下,滚动事件的触发频次较高,从而导致相关技术在监听滚动事件的过程中产生大量无用计算,影响图片懒加载的执行效率。同时,还可能导致用户终端发热、卡顿。

[0058] 针对相关技术中的问题,另一相关技术中提供了一种对监听滚动事件进行节流的方案,以降低滚动事件的触发频次。然而,节流阈值过大,将会导致图片显示不及时,造成用户体验差;节流阈值过小,则节流效果不明显。可见,对上述触发频次的节流程度并不容易控制。

[0059] 另外,在通过监听滚动事件实现图片懒加载的方案中,监听滚动事件还存在导致图片懒加载逻辑与其他业务逻辑耦合的风险,造成代码易读性差且维护难度增大。

[0060] 在本公开的实施例中,提供了一种图片的懒加载方法,至少在一定程度上克服上述相关技术中所存在的缺陷。

[0061] 图2示出本公开一示例性实施例中图片的懒加载方法的流程示意图。参考图2,该实施例提供的图片的懒加载方法,包括:

[0062] 步骤S210,为待懒加载的目标图片构建自定义元素;

[0063] 步骤S220,将所述目标图片的地址信息存储于所述自定义元素中;

[0064] 步骤S230,监测所述自定义元素,响应于所述自定义元素与视窗存在交叉区域,从所述自定义元素中获取所述地址信息;以及,

[0065] 步骤S240,根据所述地址信息,在所述视窗中显示所述目标图片,实现对所述目标图片的懒加载。

[0066] 在图2所示实施例提供的技术方案中,一方面,通过自定义元素和交叉观察的方式实现图片懒加载,实现了高效优雅地图片懒加载。另一方面,相较于相关技术,本方案摒弃对滚动事件的监听,从而,节省了监听滚动事件所需的计算资源,提升了图片懒加载效率,也无需对于监听滚动事件的触发频次的节流程度进行控制。同时,通过自定义元素有利于提升图片懒加载的语义化程度,以及提升了代码的可维护性。

[0067] 以下对图2所示实施例的各个步骤的具体实施方式进行详细阐述:

[0068] 在示例性的实施例中,在步骤S210中为待懒加载的目标图片构建自定义元素,并在步骤S220中,将所述目标图片的地址信息存储于所述自定义元素中。

[0069] 在示例性的实施例中,上述自定义元素可以为HTML自定义元素(Custom Element)。为了实现对目标图片的懒加载,则目标图片的源文件的任一属性(如,src属性)中不应存在地址信息,本实施例中,将目标图片的地址信息存储于上述自定义元素中的任

一属性中。示例性的，为待懒加载的目标图片P1构建HTML自定义元素CE1，将目标图片P1的src属性设为空，并将地址信息存储于元素CE1中。从而，浏览器获取资源定位符后，在构建文档对象模型(Document Object Model，简称:DOM)树的过程中，可以根据元素标签确定元素CE1是HTML自定义元素。因此，虽然元素CE1中包含目标图片P1的地址信息，图片也不会后续被加载至视窗中。

[0070] 示例性的，标签为lazy-img的HTML自定义元素可以表示为:<lazy-img src＝”targetSrc”></lazy-img>，其中的“targetSrc”表示目标图片的地址信息。示例性的，将目标图片的地址信息存储于自定义元素的src属性中。

[0071] 图3示出本公开一示例性实施例中自定义元素的构建方法的流程示意图。参考图3，该实施例提供的方法，包括步骤S310和步骤S320。

[0072] 在步骤S310中，为待懒加载的目标图片构建超文本标记语言格式的自定义元素。

[0073] 在示例性的实施例中，可以通过Web Components中的自定义元素接口(Custom Elements API)可自定义HTML元素。其中，API(Application Programming Interface，应用程序编程接口)。

[0074] 示例性的，一种构建HTML自定义元素的可实施方式为：根据接口window.customElements.define确定HTML自定义元素CE2。其中，window.customElements.define的第一个参数是自定义元素名称，如命名为lazy-img。window.customElements.define的第二个参数是自定义元素构造器A。

[0075] 示例性的，另一种构建HTML自定义元素的可实施方式为：根据接口document.registerElement确定HTML自定义元素CE3。其中，document.registerElement的第一个参数也是自定义元素名称，如命名为lazy-img。document.registerElement的第二个参数是一个对象B。

[0076] 在步骤S320中，在所述自定义元素中设置钩子函数，以用于绑定交叉观察器和所述自定义元素。

[0077] 在示例性的实施例中，构建HTML元素的过程中设置钩子函数，以用于绑定交叉观察器和上述自定义元素。其中，上述交叉观察器用于监测上述自定义元素是否与视窗存在交叉区域，以最终确定是否加载目标图片，相关实施例将在后续实施例中进行详细说明。

[0078] 在示例性的实施例中，为上述自定义元素中设置钩子函数的具体实施方式如下：

[0079] 对于根据接口window.customElements.define确定的HTML自定义元素CE2，在其自定义元素构造器A中实现一个钩子函数connectedCallback。由于在自定义元素CE2插入DOM树时，浏览器将自动调用钩子函数connectedCallback，从而达到交叉观察器实例与自定义元素CE2实例的绑定效果。

[0080] 对于根据接口document.registerElement确定的HTML自定义元素CE3，在其对象B中实现一个钩子函数attachedCallback。由于在自定义元素CE3插入DOM树时，浏览器将自动调用钩子函数attachedCallback，从而达到交叉观察器实例与自定义元素CE3实例的绑定效果。

[0081] 由于图3所示实施例将交叉观察器与自定义元素绑定，则在浏览器构建DOM树的过程中，将自定义元素插入DOM树时，能够调用钩子函数，从而实现自定义元素实例被交叉观察器监测。进而，浏览器能够自动调用实例化交叉观察器时定义的回调，从而实现懒加载。

[0082] 示例性的，在步骤S230中通过交叉观察器监测上述自定义元素。一种具体实施方式为：实例化一个交叉观察器window.IntersectionObserver，可以命名为io。具体地，为实例化交叉观察器提供一个相交回调函数，函数里面的动作是：遍历与视窗存在交叉区域的自定义元素并将其作为目标元素DOM；为每个目标元素DOM实例化一个图片标签对象，以及，获取存储在目标元素DOM上的图片地址，并赋值给图片标签对象的src属性；进一步地，再将这个图片标签对象插入到目标元素DOM中，从而实现图片显示。参考图4，当前页面的自定义元素包括：自定义元素CE1、自定义元素CE2和自定义元素CE3，其中，与视窗400存在交叉区域的目标元素为CE1和CE2。

[0083] 在示例性的实施例中，图5示出了本公开一示例性实施例中目标图片的显示方法的流程示意图。具体提供了一种实现目标元素为CE1和CE2对应的图片显示方法。参考图5，该实施例提供的方法包括：

[0084] 步骤S510，实例化图片标签对象；以及，步骤S520，从所述自定义元素中获取所述地址信息，并将所述地址信息赋值给所述图片标签对象的属性，以在所述视窗中显示所述目标图片。

[0085] 示例性的，分别为目标元素CE1和CE2实例化图片标签对象，并获取目标图片P1的地址信息S1以及获取目标图片P2的地址信息S2，然后，通过地址信息S1实例化目标元素CE1的图片标签对象的属性，以及通过地址信息S2实例化目标元素CE2的图片标签对象的属性。以实现在视窗400中显示目标图片P1和目标图片P2。

[0086] 示例性的，实现目标图片的显示之后，解绑上述交叉观察器io和上述自定义元素(在本实施例中即为目标元素CE1和目标元素CE2)，以结束对自定义元素的监测。

[0087] 在示例性的实施例中，图6示出本公开另一示例性实施例中自定义元素的构建方法的流程示意图。参考图6，该实施例提供的方法包括：

[0088] 步骤S610，为待懒加载的目标图片构建超文本标记语言格式的自定义元素；以及，步骤S620，通过所述自定义元素继承超文本标记语言的元素类。

[0089] 在示例性的实施例中，步骤S610的具体实施方式与步骤S310的具体实施方式相同，在此不再赘述。

[0090] 示例性的，本实施例在步骤S620中，通过自定义元素继承超文本标记语言的元素类，来实现对目标元素(与视窗400存在交叉区域的自定义元素)的操作。

[0091] 示例性的，针对上述两种不同的自定义元素，继承HTMLElement类的具体实施方式包括：

[0092] 对于根据接口window.customElements.define确定的HTML自定义元素CE2，通过上述构造器A继承HTMLElement。

[0093] 对于根据接口document.registerElement确定的HTML自定义元素CE3，对于在其对象B提供一个描述自定义元素原型属性对象B.prototype，该B.prototype需要继承HTMLElement.prototype。

[0094] 根据图6所示实施例提供的技术方案，自定义元素中继承了HTMLElement类之后，可以通过HTMLElement类的方式实现目标元素的图片显示。

[0095] 在示例性的实施例中，图7示出了本公开另一示例性实施例中目标图片的显示方法的流程示意图。具体是在图6所示实施例的基础上实现目标元素为CE1和CE2对应的图片

显示方法。参考图7,该实施例提供的方法包括步骤S710-步骤S730。

[0096] 在步骤S710中,实例化图片标签对象。

[0097] 仍以图4为例进行说明:为与视窗400有交叉区域的自定义元素CE1实例化一个图片标签对象T1,以及为定义元素CE2实例化一个图片标签对象T2。其中,实例化图片标签对象的具体实施方式可以是通过函数new Image()实现。

[0098] 在步骤S720中,通过超文本标记语言的元素类的属性获取方法从所示自定义元素中获取所述地址信息,并将所述地址信息赋值给所述图片标签对象的属性。

[0099] 在示例性的实施例中,通过HTMLElement类的属性获取方法getAttribute()获取目标图片的地址信息,并将其赋值给图片标签对象的src属性。具体的,通过自定义元素CE1的方法getAttribute()获取目标图片P1的地址信息S1,并将其赋值给图片标签对象T1的src属性。

[0100] 需要注意的是,为了准确获取的目标图片并准确显示目标图片,上述方法getAttribute("srcQQ")中的"srcQQ"须要与对应的自定义元素中存储的地址信息属性名相同。例如,对应的自定义元素为:<lazy-img srcQQ="targetSrcQQ"></lazy-img>。

[0101] 在步骤S730中,通过超文本标记语言的元素类的子节点添加方法,将赋值后的图片标签对像添加至所述自定义元素的孩子节点中,以在所述视窗中显示所述目标图片。

[0102] 在示例性的实施例中,通过HTMLElement类的子节点添加方法appendChild()将图片标签对象添加到目标元素的孩子节点中,实现目标图片的展示。示例性的,通过自定义元素CE1的方法appendChild()将图片标签对象T1添加到目标元素CE1的孩子节点中,从而实现目标图片P1的展示。

[0103] 在示例性的实施例中,实现目标图片的显示之后,解绑上述交叉观察器io和上述自定义元素(在本实施例中即为目标元素CE1和目标元素CE2),以结束对自定义元素的监测。

[0104] 本技术方案提供的图片的懒加载方案中,通过HTML自定义元素来存储目标图片的真实地址信息,并通过交叉观察器监测自定义元素与视窗是否重叠的方式确定目标元素,从而实现在不依赖监听滚动事件的情况下的图片懒加载。相较于相关技术,本技术方案节省了监听滚动事件所需的计算资源,提升了图片懒加载效率。同时,摒弃监听滚动事件方式有利于避免懒加载逻辑与其他的业务逻辑杂糅,也不需要在逻辑层初始的执行流程中,对所有图片进行一次额外的扫描,提升代码的可维护性和可读性。

[0105] 另外,HTML自定义元素更具语义化,从而提高代码的可维护性,符合HTML5标准所提倡的元素语义化要求。

[0106] 本领域技术人员可以理解实现上述实施方式的全部或部分步骤被实现为由CPU执行的计算机程序。在该计算机程序被CPU执行时,执行本公开提供的上述方法所限定的上述功能。所述的程序可以存储于一种计算机可读存储介质中,该存储介质可以是只读存储器,磁盘或光盘等。

[0107] 此外,需要注意的是,上述附图仅是根据本公开示例性实施方式的方法所包括的处理的示意性说明,而不是限制目的。易于理解,上述附图所示的处理并不表明或限制这些处理的时间顺序。另外,也易于理解,这些处理可以是例如在多个模块中同步或异步执行的。

[0108] 以下介绍本公开的接口适配系统实施例，可以用于执行本公开上述的即可适配方法。

[0109] 图8示出本公开示例性实施例中图片的懒加载装置的结构示意图。如图8所示，上述图片的懒加载装置800包括：自定义元素构建模块801、图片地址存储模块802、自定义元素监测模块803，以及图片显示模块804。

[0110] 其中，上述自定义元素构建模块801，被配置为为待懒加载的目标图片构建自定义元素；上述图片地址存储模块802，被配置为将上述目标图片的地址信息存储于上述自定义元素中；上述自定义元素监测模块803，被配置为监测上述自定义元素，响应于上述自定义元素与视窗存在交叉区域，从上述自定义元素中获取上述地址信息；以及，上述图片显示模块804，被配置为根据上述地址信息，在上述视窗中显示上述目标图片，实现对上述目标图片的懒加载。

[0111] 在本公开的一些实施例中，基于前述方案，上述自定义元素构建模块801，包括：第一构建单元8011和钩子函数设置单元8012。

[0112] 其中，上述第一构建单元8011被配置为：为待懒加载的目标图片构建超文本标记语言格式的自定义元素；以及，上述钩子函数设置单元8012被配置为：在上述自定义元素中设置钩子函数，以用于绑定交叉观察器和上述自定义元素。

[0113] 在此基础上，上述自定义元素监控模块803，具体被配置为：通过上述交叉观察器检测上述自定义元素。

[0114] 在本公开的一些实施例中，基于前述方案，上述图片地址存储模块802，被具体配置为：将上述目标图片的地址信息存储于上述自定义元素的任一属性中。

[0115] 在本公开的一些实施例中，基于前述方案，上述自定义元素监测模块803，包括：第一图片标签对象实例化单元8031和图片显示单元8032。

[0116] 其中，上述第一图片标签对象实例化单元8031被配置为：实例化图片标签对象；以及，上述图片显示单元8032被配置为：从所述自定义元素中获取所述地址信息，并将上述地址信息赋值给上述图片标签对象的属性，以在上述视窗中显示上述目标图片。

[0117] 在本公开的一些实施例中，基于前述方案，上述自定义元素构建模块801，包括：第二构建单元8013和类继承单元8014。

[0118] 其中，上述第二构建单元8013被配置为：为待懒加载的目标图片构建超文本标记语言格式的自定义元素；以及，上述类继承单元8014被配置为：通过上述自定义元素继承超文本标记语言的元素类。

[0119] 在此基础上，上述自定义元素监控模块803，具体被配置为：通过超文本标记语言的元素类的属性获取方法获取上述地址信息。

[0120] 本公开的一些实施例中，基于前述方案，上述自定义元素监测模块803，包括：第二图片标签对象实例化单元8033、赋值单元8034和孩子节点添加单元8035。

[0121] 其中，上述第二图片标签对象实例化单元8033被配置为：实例化图片标签对象；上述赋值单元8034被配置为：通过超文本标记语言的元素类的属性获取方法从上述自定义元素中获取上述地址信息，并将上述地址信息赋值给上述图片标签对象的属性；以及，上述孩子节点添加单元8035被配置为：通过超文本标记语言的元素类的子节点添加方法，将赋值后的图片标签对像添加至上述自定义元素的孩子节点中，以在上述视窗中显示上述目标图

片。

[0122] 本公开的一些实施例中，基于前述方案，上述图片的懒加载装置800包括：解绑模块805。

[0123] 其中，上述解绑模块805被配置为：在上述图片显示模块804在视窗中显示上述目标图片之后，解绑上述交叉观察器和上述自定义元素，以结束对上述自定义元素的监测。

[0124] 上述图片的懒加载装置中各单元的具体细节已经在对应的图片的懒加载方法中进行了详细的描述，因此此处不再赘述。

[0125] 图9示出了适于用来实现本公开实施例的电子设备的计算机系统的结构示意图。

[0126] 需要说明的是，图9示出的电子设备的计算机系统900仅是一个示例，不应对本公开实施例的功能和使用范围带来任何限制。

[0127] 如图9所示，计算机系统900包括中央处理单元(Central Processing Unit，CPU)901，其可以根据存储在只读存储器(Read-Only Memory，ROM)902中的程序或者从储存部分908加载到随机访问存储器(Random Access Memory，RAM)903中的程序而执行各种适当的动作和处理。在RAM 903中，还存储有系统操作所需的各种程序和数据。CPU 901、ROM 902以及RAM 903通过总线904彼此相连。输入/输出(Input/Output，I/O)接口905也连接至总线904。

[0128] 以下部件连接至I/O接口905：包括键盘、鼠标等的输入部分906；包括诸如阴极射线管(Cathode Ray Tube，CRT)、液晶显示器(Liquid Crystal Display，LCD)等以及扬声器等的输出部分907；包括硬盘等的储存部分908；以及包括诸如LAN(Local Area Network，局域网)卡、调制解调器等的网络接口卡的通信部分909。通信部分909经由诸如因特网的网络执行通信处理。驱动器910也根据需要连接至I/O接口905。可拆卸介质911，诸如磁盘、光盘、磁光盘、半导体存储器等等，根据需要安装在驱动器910上，以便于从其上读出的计算机程序根据需要被安装入储存部分908。

[0129] 特别地，根据本公开的实施例，下文参考流程图描述的过程可以被实现为计算机软件程序。例如，本公开的实施例包括一种计算机程序产品，其包括承载在计算机可读介质上的计算机程序，该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中，该计算机程序可以通过通信部分909从网络上被下载和安装，和/或从可拆卸介质911被安装。在该计算机程序被中央处理单元(CPU)901执行时，执行本申请的系统中限定的各种功能。

[0130] 需要说明的是，本公开实施例所示的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于：具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(Erasable Programmable Read Only Memory，EPROM)、闪存、光纤、便携式紧凑磁盘只读存储器(Compact Disc Read-Only Memory，CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本公开中，计算机可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本公开中，计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其

中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质，该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括但不限于：无线、有线等等，或者上述的任意合适的组合。

[0131] 附图中的流程图和框图，图示了按照本公开各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分，上述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意，在有些作为替换的实现中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个接连地表示的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图或流程图中的每个方框、以及框图或流程图中的方框的组合，可以用执行规定的功能或操作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0132] 描述于本公开实施例中所涉及到的单元可以通过软件的方式实现，也可以通过硬件的方式来实现，所描述的单元也可以设置在处理器中。其中，这些单元的名称在某种情况下并不构成对该单元本身的限定。

[0133] 作为另一方面，本申请还提供了一种计算机可读介质，该计算机可读介质可以是上述实施例中描述的电子设备中所包含的；也可以是单独存在，而未装配入该电子设备中。上述计算机可读介质承载有一个或者多个程序，当上述一个或者多个程序被一个该电子设备执行时，使得该电子设备实现上述实施例中所述的方法。

[0134] 例如，所述的电子设备可以实现如图2中所示的：步骤S210，为待懒加载的目标图片构建自定义元素；步骤S220，将所述目标图片的地址信息存储于所述自定义元素中；步骤S230，监测所述自定义元素，响应于所述自定义元素与视窗存在交叉区域，从所述自定义元素中获取所述地址信息；以及，步骤S240，根据所述地址信息，在所述视窗中显示所述目标图片，实现对所述目标图片的懒加载。

[0135] 又如，所述的电子设备可以实现如图3至图7中任意一图所示的各个步骤。

[0136] 应当注意，尽管在上文详细描述中提及了用于动作执行的设备的若干模块或者单元，但是这种划分并非强制性的。实际上，根据本公开的实施方式，上文描述的两个或更多模块或者单元的特征和功能可以在一个模块或者单元中具体化。反之，上文描述的一个模块或者单元的特征和功能可以进一步划分为由多个模块或者单元来具体化。

[0137] 通过以上的实施方式的描述，本领域的技术人员易于理解，这里描述的示例实施方式可以通过软件实现，也可以通过软件结合必要的硬件的方式来实现。因此，根据本公开实施方式的技术方案可以以软件产品的形式体现出来，该软件产品可以存储在一个非易失性存储介质(可以是CD-ROM，U盘，移动硬盘等)中或网络上，包括若干指令以使得一台计算设备(可以是个人计算机、服务器、触控终端、或者网络设备等)执行根据本公开实施方式的方法。

[0138] 本领域技术人员在考虑说明书及实践这里公开的发明后，将容易想到本公开的其

它实施方案。本申请旨在涵盖本公开的任何变型、用途或者适应性变化，这些变型、用途或者适应性变化遵循本公开的一般性原理并包括本公开未公开的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的，本公开的真正范围和精神由下面的权利要求指出。

[0139] 应当理解的是，本公开并不局限于上面已经描述并在附图中示出的精确结构，并且可以在不脱离其范围进行各种修改和改变。本公开的范围仅由所附的权利要求来限制。

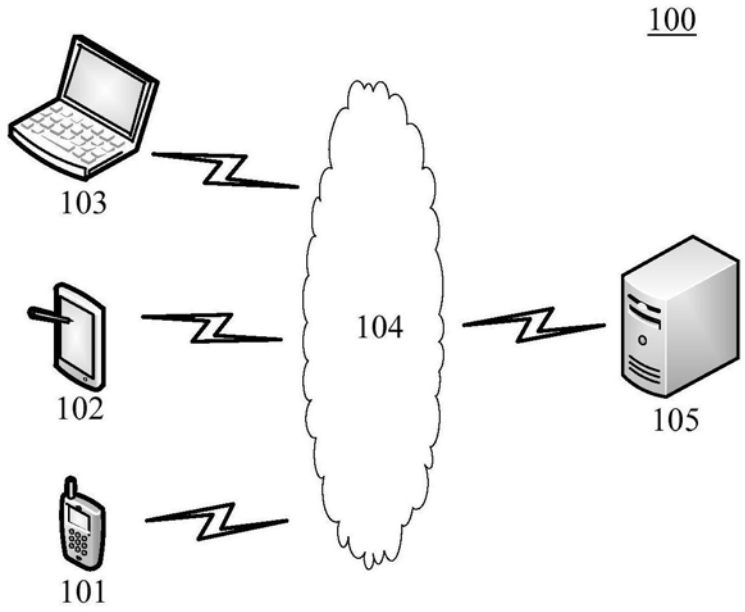

100
103
102
101
104
105

图1

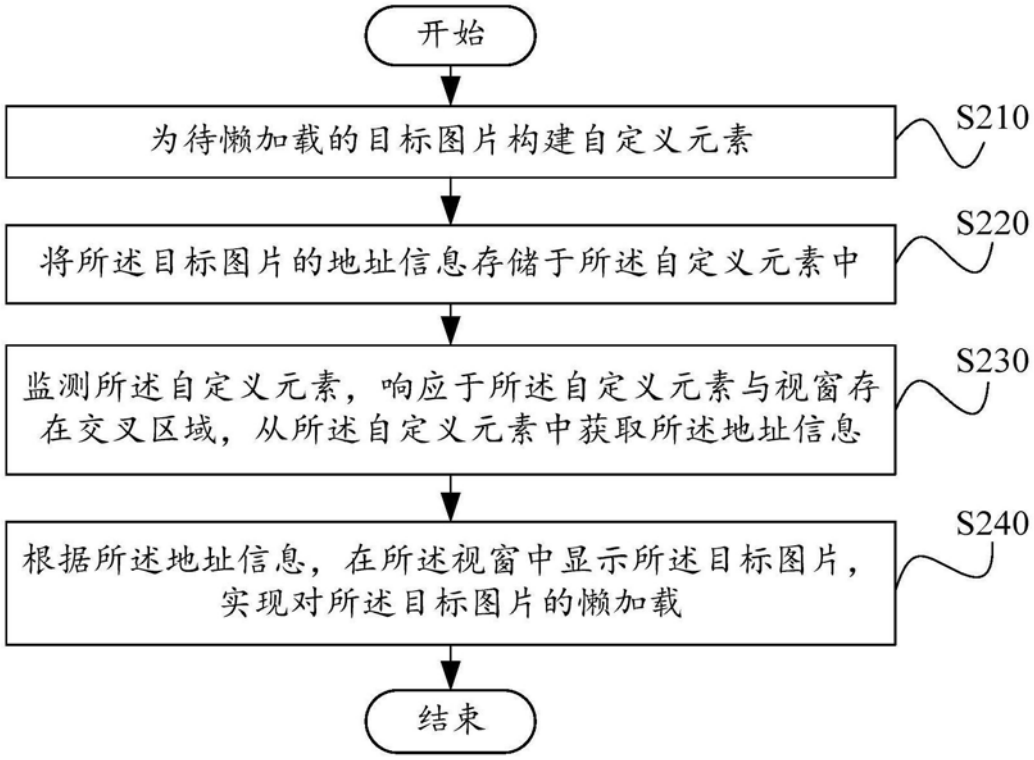

开始
为待懒加载的目标图片构建自定义元素
S210
将所述目标图片的地址信息存储于所述自定义元素中
S220
监测所述自定义元素，响应于所述自定义元素与视窗存在交叉区域，从所述自定义元素中获取所述地址信息
S230
根据所述地址信息，在所述视窗中显示所述目标图片，实现对所述目标图片的懒加载
S240
结束

图2

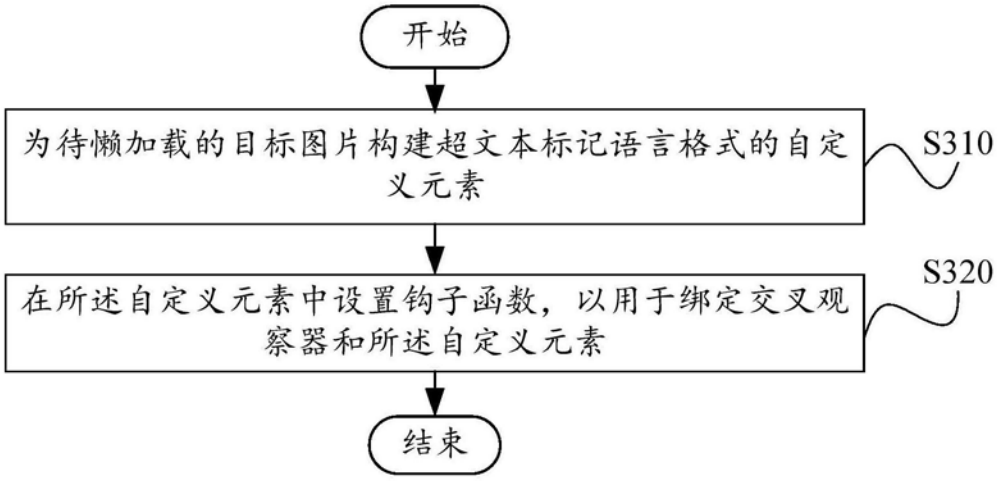

开始
为待懒加载的目标图片构建超文本标记语言格式的自定义元素
S310
在所述自定义元素中设置钩子函数，以用于绑定交叉观察器和所述自定义元素
S320
结束

图3

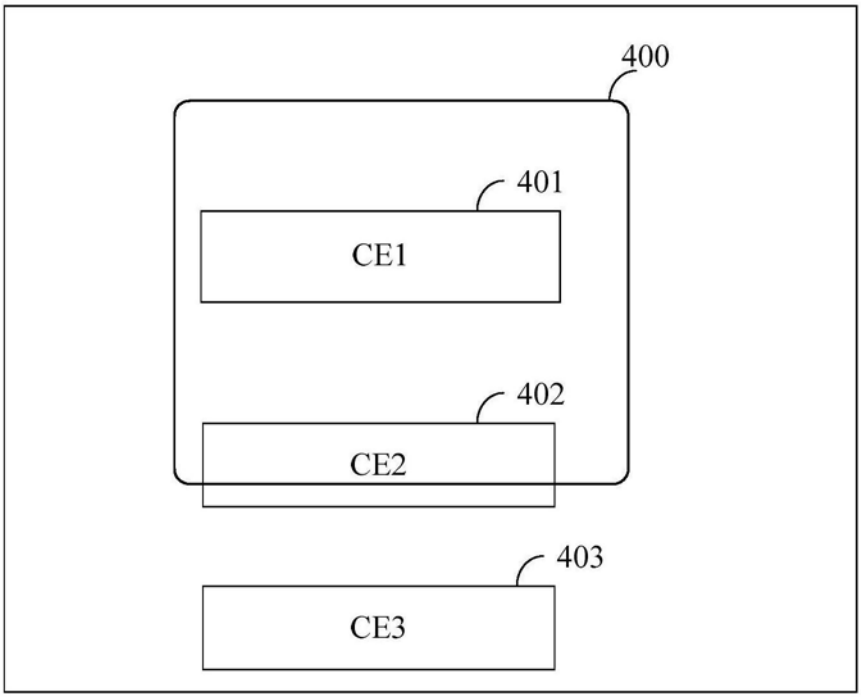

400
401
CE1
402
CE2
403
CE3

图4

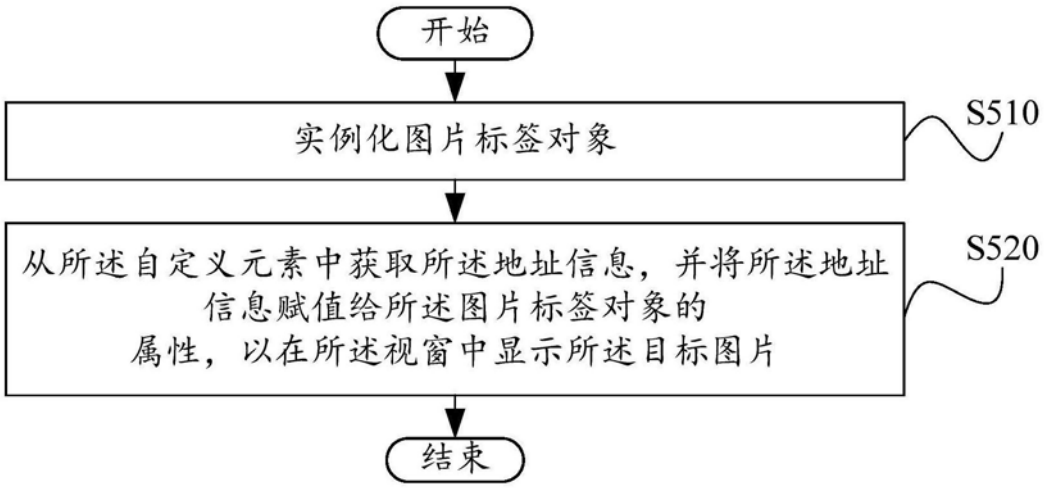

开始
实例化图片标签对象
S510
从所述自定义元素中获取所述地址信息，并将所述地址信息赋值给所述图片标签对象的属性，以在所述视窗中显示所述目标图片
S520
结束

图5

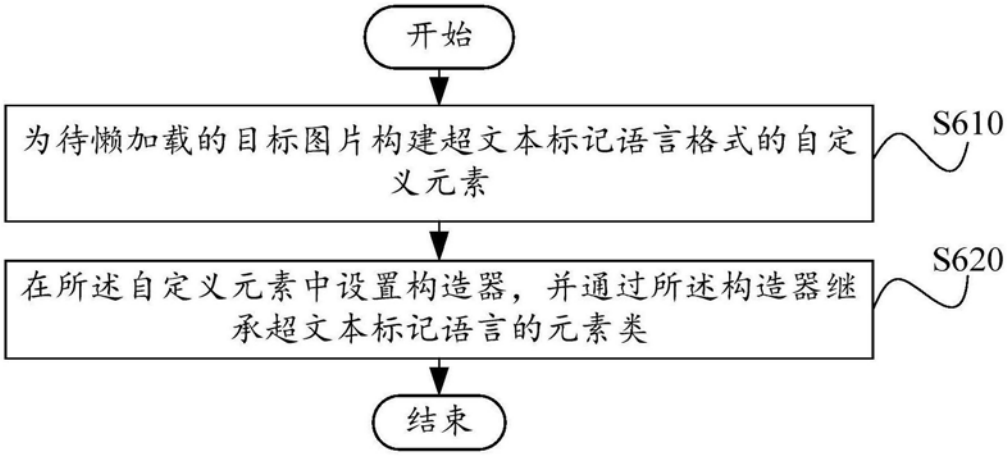

开始
为待懒加载的目标图片构建超文本标记语言格式的自定义元素
S610
在所述自定义元素中设置构造器，并通过所述构造器继承超文本标记语言的元素类
S620
结束

图6

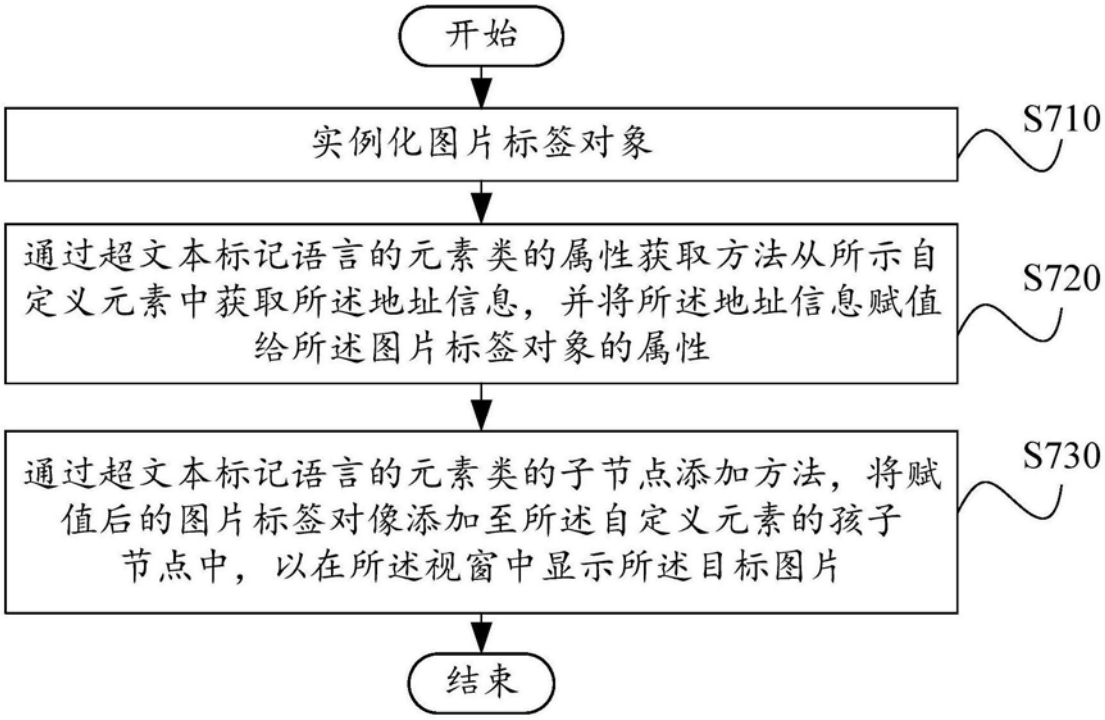

开始
实例化图片标签对象
S710
通过超文本标记语言的元素类的属性获取方法从所示自定义元素中获取所述地址信息，并将所述地址信息赋值给所述图片标签对象的属性
S720
通过超文本标记语言的元素类的子节点添加方法，将赋值后的图片标签对像添加至所述自定义元素的孩子节点中，以在所述视窗中显示所述目标图片
S730
结束

图7

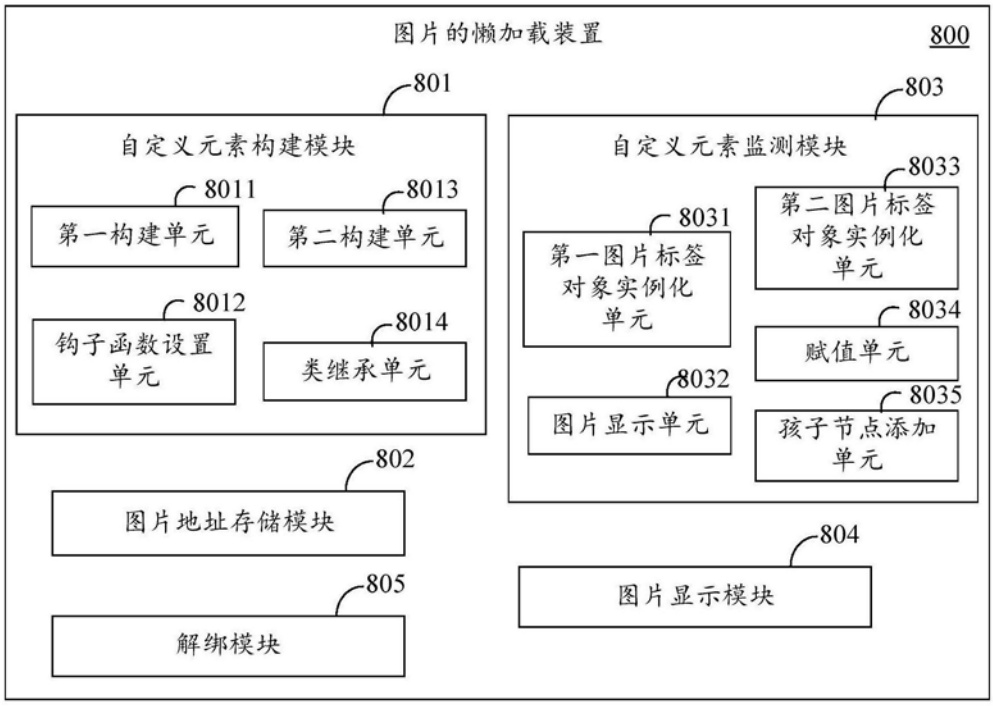

图片的懒加载装置
800
801
自定义元素构建模块
8011
第一构建单元
8013
第二构建单元
8012
钩子函数设置单元
8014
类继承单元
803
自定义元素监测模块
8033
第二图片标签对象实例化单元
8031
第一图片标签对象实例化单元
8034
赋值单元
8032
图片显示单元
8035
孩子节点添加单元
802
图片地址存储模块
804
图片显示模块
805
解绑模块

图8

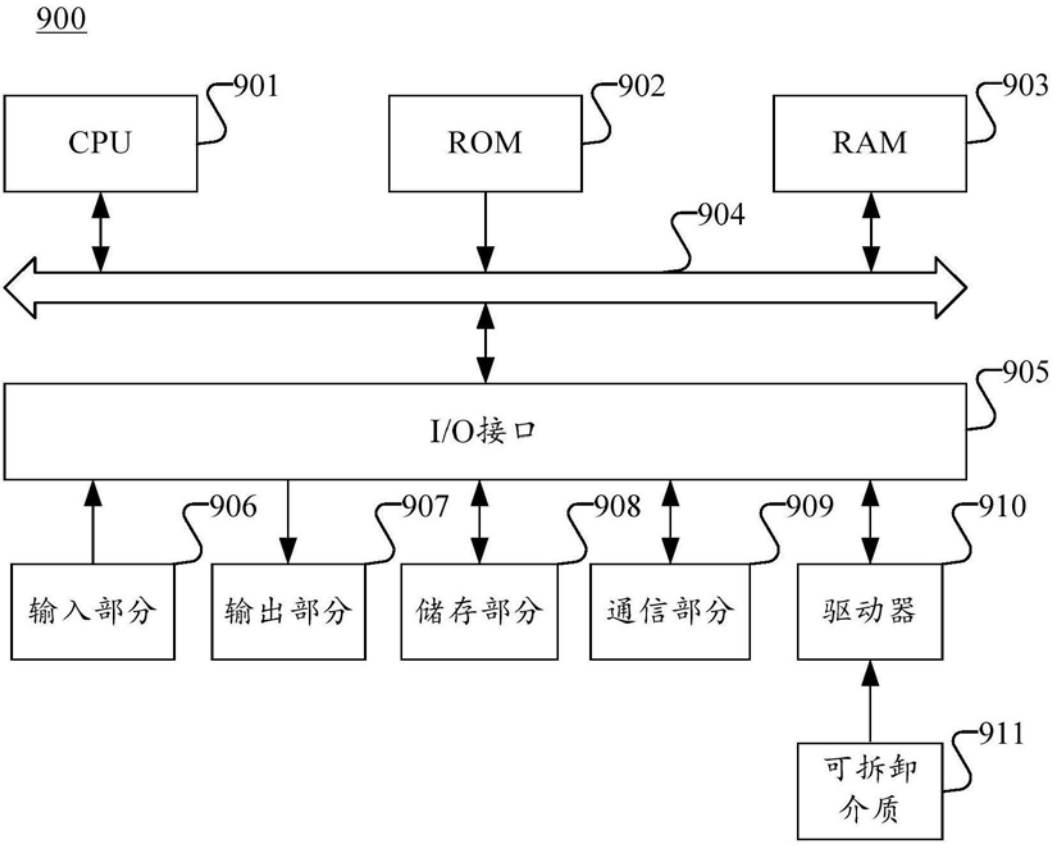

900
CPU
901
ROM
902
RAM
903
904
I/O接口
905
输入部分
906
输出部分
907
储存部分
908
通信部分
909
驱动器
910
可拆卸介质
911

图9