



(12) 发明专利申请

(10) 申请公布号 CN 112256995 A

(43) 申请公布日 2021.01.22

(21) 申请号 202011146963.4

(22) 申请日 2020.10.23

(71) 申请人 四川长虹电器股份有限公司
地址 621000 四川省绵阳市高新区绵兴东路35号

(72) 发明人 曾祥阳

(74) 专利代理机构 成都虹桥专利事务所(普通合伙) 51124

代理人 吴中伟

(51) Int. Cl.

G06F 16/957 (2019.01)

G06F 16/955 (2019.01)

G06F 16/901 (2019.01)

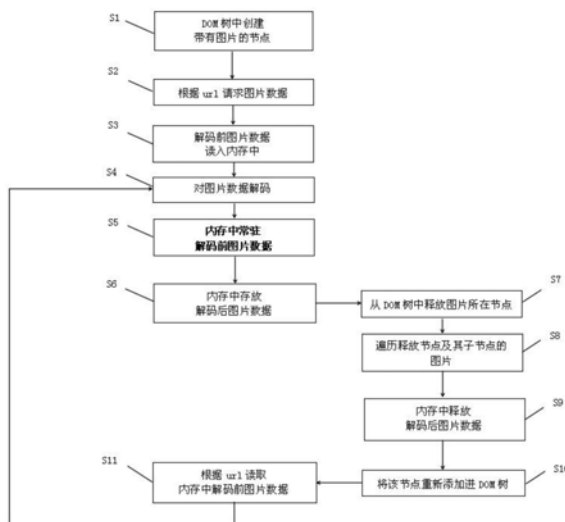
权利要求书1页 说明书3页 附图2页

(54) 发明名称

改善浏览器内存的方法

(57) 摘要

本发明涉及web领域,具体涉及一种改善浏览器内存的方法,解决了浏览器的内存占用不会有明显减少的问题,改善了浏览器内存的使用,提升了使用流畅度和用户体验。本发明改善浏览器内存的方法,解析url请求的图片数据;然后将请求到的图片数据进行解码并保存在内存中;判断页面的不可视区域,并将不可视区域的DOM树中的节点移除;遍历被移除的节点,获取对应的包含图片的url;根据获取对应的包含图片的url,清除内存中该url对应的解码后数据,并将该url与对应解码前的数据进行关联;对页面的可视区域进行判断,若被移除节点的原不可视区域重新回到可视区域,则将移除的节点重新添加回DOM树。本发明适用于改善浏览器内存。



1.改善浏览器内存的方法,其特征在于,包括:

步骤(1)、解析url请求的图片数据;

步骤(2)、将请求到的图片数据进行解码并保存在内存中;

步骤(3)、判断页面的不可视区域,并将不可视区域的DOM树中的节点移除;

步骤(4)、遍历被移除的节点,获取对应的包含图片的url;

步骤(5)、根据获取对应的包含图片的url,清除内存中该url对应的解码后数据,并将该url与对应解码前的数据进行关联;

步骤(6)、对页面的可视区域进行判断,若被移除节点的原不可视区域重新回到可视区域,则将移除的节点重新添加回DOM树,并进入步骤(1)。

2.根据权利要求1所述的改善浏览器内存的方法,其特征在于,在步骤(1)中,解析url请求的图片数据的具体方法包括:在DOM树中创建和添加带有图片的节点,浏览器根据图片url请求图片数据。

3.根据权利要求1所述的改善浏览器内存的方法,其特征在于,在步骤(1)中,解析url请求包括网络请求以及本地请求。

4.根据权利要求1所述的改善浏览器内存的方法,其特征在于,在步骤(2)中,将请求到的图片数据进行解码并保存在内存中的具体方法包括:将请求到的图片数据进行解码,获得解码后图片数据,然后将解码后图片数据存放于内存中,直到图片所在节点从DOM树中移除,同时将解码前的图片数据也存放于内存中。

5.根据权利要求1所述的改善浏览器内存的方法,其特征在于,在步骤(4)中,遍历被移除的DOM节点的具体方法包括:浏览器遍历被移除的节点,节点是树形结构,遍历完整的树形结构后得到对应包含图片的url。

改善浏览器内存的方法

技术领域

[0001] 本发明涉及web领域,具体涉及一种改善浏览器内存的方法。

背景技术

[0002] 随着web技术的普及,对于浏览器应用性能和用户体验要求也越来越高,内存占用作为浏览器一个重要性能指标,更是直接影响着浏览器的体验。在如今web页面越来越丰富的背景下,页面中图片占用是使得浏览器内存占用增加的主要原因。在不影响功能的情况下,降低浏览器中图片数据的内存占用,对于提高用户体验具有决定性作用。

[0003] Web页面的节点数量过多,会使DOM树结构复杂和影响页面的流畅度,所以前端需要移除不必要的节点。

[0004] 现有技术如图1,首先在DOM树中创建带有图片的节点,根据url请求图片数据,将请求的图片数据进行解码并读入内存中,然后释放内存中解码前的图片数据此时内存中只有解码后的图片数据,在不可视区域页面的DOM树中释放图片所在的节点,内存中就常驻解码后的图片数据,后续不可视区域页面回到可视区域后,再将图片所在的节点重新添加进DOM树,然后根据url读取内存中解码后的图片数据。

[0005] 可以看出,图片所在节点从DOM树中被移除后,浏览器的内存占用不会有明显减少,因为解码后的图片数据仍然常驻内存中。

发明内容

[0006] 本发明的目的是提供一种改善浏览器内存的方法,解决了图片所在节点从DOM树中被移除后,浏览器的内存占用不会有明显减少的问题,提升了使用流畅度和用户体验。

[0007] 本发明采取如下技术方案实现上述目的,改善浏览器内存的方法,包括:

[0008] 步骤(1)、解析url请求的图片数据;

[0009] 步骤(2)、将请求到的图片数据进行解码并保存在内存中;

[0010] 步骤(3)、判断页面的不可视区域,并将不可视区域的DOM树中的节点移除;

[0011] 步骤(4)、遍历被移除的节点,获取对应的包含图片的url;

[0012] 步骤(5)、根据获取对应的包含图片的url,清除内存中该url对应的解码后数据,并将该url与对应解码前的数据进行关联;

[0013] 步骤(6)、对页面的可视区域进行判断,若被移除节点的原不可视区域重新回到可视区域,则将移除的节点重新添加回DOM树,并进入步骤(1)。

[0014] 进一步的是,在步骤(1)中,解析url请求的图片数据的具体方法包括:在DOM树中创建和添加带有图片的节点,浏览器根据图片url请求图片数据。

[0015] 进一步的是,在步骤(1)中,解析url请求包括网络请求以及本地请求。

[0016] 进一步的是,在步骤(2)中,将请求到的图片数据进行解码并保存在内存中的具体方法包括:将请求到的图片数据进行解码,获得解码后图片数据,然后将解码后图片数据存放于内存中,直到图片所在节点从DOM树中移除,同时将解码前的图片数据也存放于内存

中。

[0017] 进一步的是,在步骤(4)中,遍历被移除的DOM节点的具体方法包括:浏览器遍历被移除的节点,节点是树形结构,遍历完整的树形结构后得到对应包含图片的url。

[0018] 本发明在对图片数据处理的过程中,清除了不可视区域解码后的图片数据的同时,将解码前的图片数据与不可视区域的节点对应的url进行了关联,解码前的图片数据内存占用通常只有解码后的几十分之一大小,因此极大地降低了内存占用;页面交互,当不可视区域重新回到可视区域后,再根据之前的关联,请求图片数据并进行解码,因此改善了内存的使用,提升了使用流畅度和用户体验。

附图说明

[0019] 图1是本发明现有技术中图片数据处理流程示意图。

[0020] 图2是本发明图片数据处理流程的一种实施例示意图。

具体实施方式

[0021] 本发明改善浏览器内存的方法,包括:

[0022] 步骤(1)、解析url请求的图片数据;

[0023] 步骤(2)、将请求到的图片数据进行解码并保存在内存中;

[0024] 步骤(3)、判断页面的不可视区域,并将不可视区域的DOM树中的节点移除;

[0025] 步骤(4)、遍历被移除的节点,获取对应的包含图片的url;

[0026] 步骤(5)、根据获取对应的包含图片的url,清除内存中该url对应的解码后数据,并将该url与对应解码前的数据进行关联;

[0027] 步骤(6)、对页面的可视区域进行判断,若被移除节点的原不可视区域重新回到可视区域,则将移除的节点重新添加回DOM树,并进入步骤(1)。

[0028] 步骤(1)中,解析url请求的图片数据的具体方法包括:在DOM树中创建和添加带有图片的节点,浏览器根据图片url请求图片数据。

[0029] 其中,节点:又称DOM节点,HTML网页中的所有内容都是DOM节点,整个HTML文档也是一个节点,前端可以通过JavaScript对节点进行移除和添加。

[0030] DOM树:所有DOM节点彼此有关系,构成树形结构,通过移除DOM树中暂时不使用的节点,可以使DOM树保持更简单的结构,增加页面流畅度。

[0031] 步骤(1)中,解析url请求包括网络请求以及本地请求。

[0032] 步骤(2)中,将请求到的图片数据进行解码并保存在内存中的具体方法包括:将请求到的图片数据进行解码,获得解码后图片数据,然后将解码后图片数据存放于内存中,直到图片所在节点从DOM树中移除,同时将解码前的图片数据也存放于内存中。相比解码后图片数据占用更少内存,用于节点移除后,替代现有技术中保存的解码后图片数据。

[0033] 其中,解码前图片数据:即编码的图片,是压缩过的数据,例如jpeg、png等格式的图片文件数据。

[0034] 解码后图片数据:将编码的图片解码成像素数据,在内存中比解码前图片大几倍到几十倍;以JPEG图片格式为例,压缩比率通常在10:1到40:1之间。

[0035] 步骤(4)中,遍历被移除的DOM节点的具体方法包括:浏览器遍历被移除的节点,节

点是树形结构,遍历完整的树形结构后得到对应包含图片的url。

[0036] 图2是本发明改善浏览器内存的方法中图片数据处理流程的实施例示意图。

[0037] 步骤s1、在DOM树中新创建和添加带有图片的节点;

[0038] 步骤s2、根据节点中图片的url,对图片数据进行网络请求,或者是对本地图片文件进行请求;

[0039] 步骤s3、将s1中请求到的图片数据读入内存中,这时的图片数据是待解码的压缩状态;

[0040] 步骤s4、对s2中读入内存的图片数据进行解码,得到解码后的图片像素数据;

[0041] 步骤s5、在内存中保存解码前的图片数据,这部分数据常驻内存;

[0042] 步骤s6、在内存中保存解码后的图片数据,对应节点从DOM树中移除前一直保存在内存中;

[0043] 步骤s7、前端通过JavaScript编程进行操作,在不可视区域的DOM树移除含有图片的节点;

[0044] 步骤s8、遍历在s6中被释放的节点和子节点,寻找出包含的图片url;

[0045] 步骤s9、根据s7中遍历得到的图片url,浏览器找到内存中对应的解码后数据,进行清除释放;

[0046] 步骤s10、前端通过JavaScript编程进行操作,原不可视区域重新回到可视区域时,将s6中被移除的节点添加进DOM树中;

[0047] 步骤s11、根据s10中添加进节点的图片url,获取内存中对应的解码前图片数据,然后重复读取流程,进入s4进行解码。

[0048] 浏览器清理移除节点中保存的解码后图片数据,保存解码前图片数据;以JPEG图片格式为例,压缩比率通常在10:1到40:1之间,即优化后图片数据内存占用仅为之前几分之一;图片数据内存占用是浏览器内存占用的主要部分,所以本技术达到优化浏览器应用的内存占用的效果,提升了使用流畅度和用户体验。

[0049] 综上所述,本发明解决了图片所在节点从DOM树中被移除后,浏览器的内存占用不会有明显减少的问题,提升了使用流畅度和用户体验。

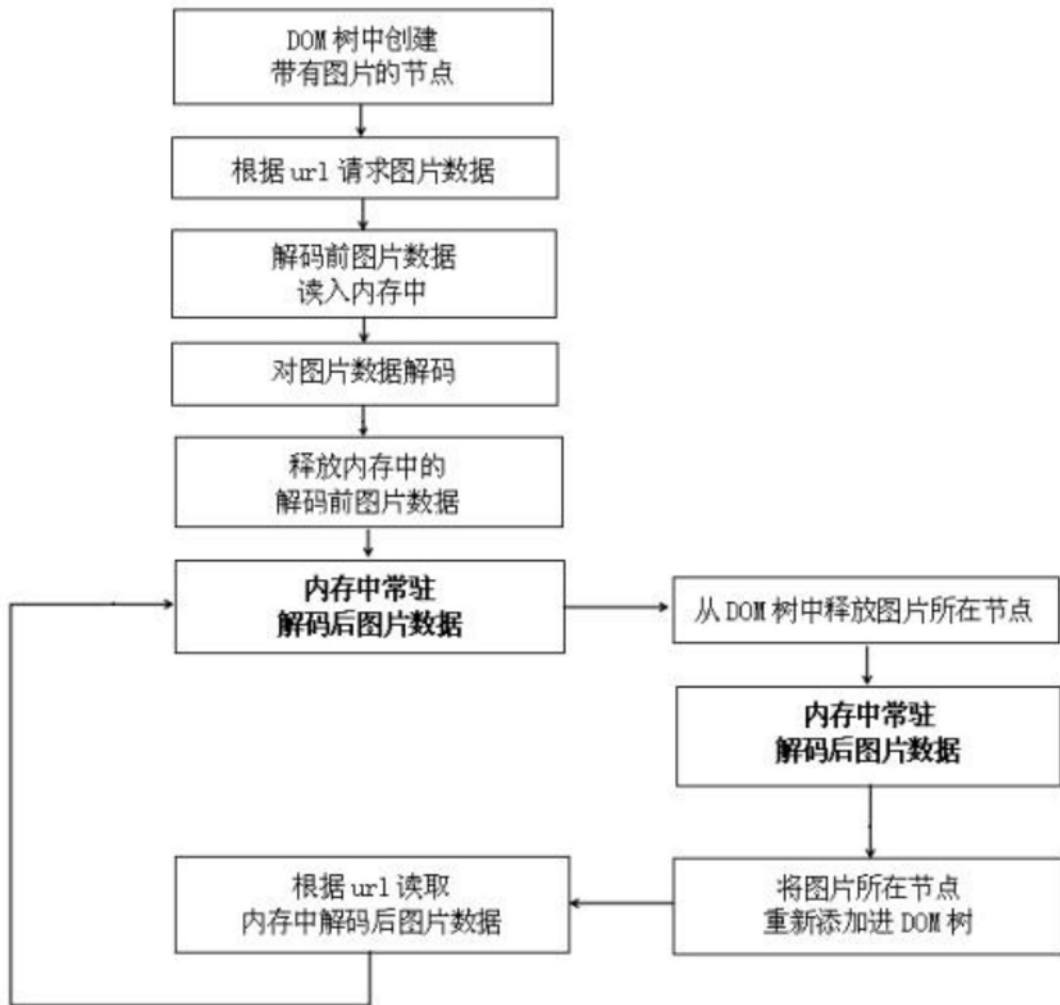


图1

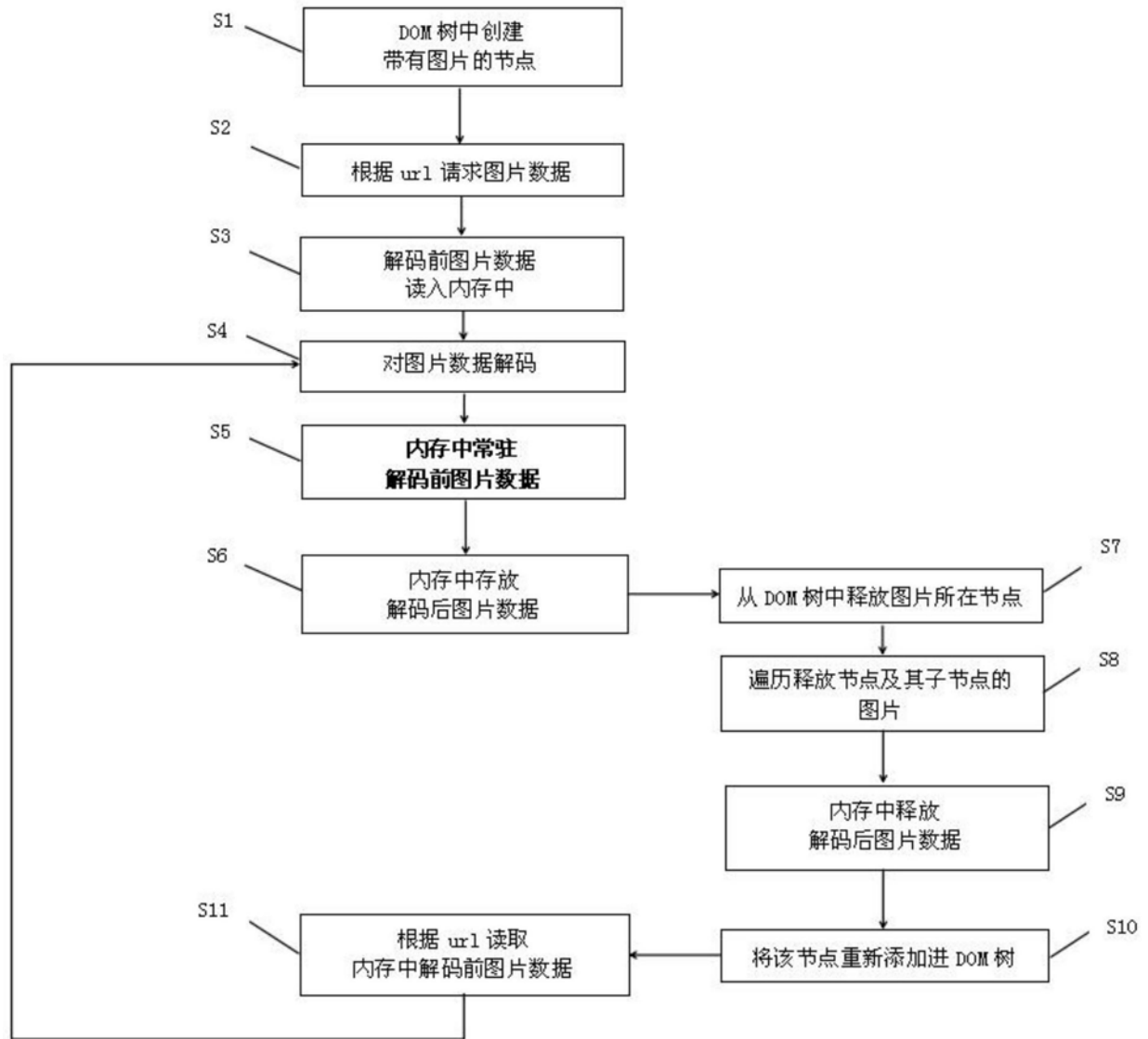


图2