



(12) 发明专利

(10) 授权公告号 CN 108197160 B

(45) 授权公告日 2022. 11. 25

(21) 申请号 201711317391.X

(22) 申请日 2017.12.12

(65) 同一申请的已公布的文献号
申请公布号 CN 108197160 A

(43) 申请公布日 2018.06.22

(73) 专利权人 腾讯科技(深圳)有限公司
地址 518044 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 柯灵杰 赵桢阳 夏胜飞 刘小龙

(74) 专利代理机构 北京同达信恒知识产权代理
有限公司 11291

专利代理师 郭润湘

(51) Int. Cl.

G06F 16/957 (2019.01)

G06F 16/955 (2019.01)

(56) 对比文件

CN 105118020 A, 2015.12.02

CN 106897030 A, 2017.06.27

qq_20889581.Android中图片的三级缓存策略.《CSDNhttps://blog.csdn.net/qq_20889581/article/details/51034472》.2016,
flychao88.缓存淘汰算法—LRU算法.《<https://www.pudn.com/news/628f82f6bf399b7f351e5c06.html>》.2013,
江海细流.Android之网络下载与图片解码-----网络图片浏览器.《<https://blog.csdn.net/fengliang191/article/details/31032089>》.2014,

审查员 何华

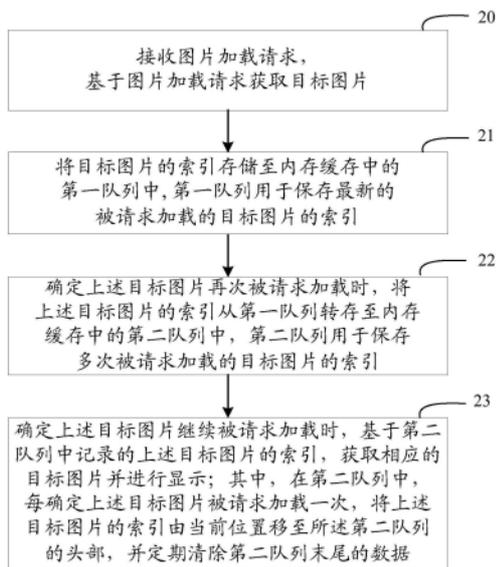
权利要求书3页 说明书12页 附图6页

(54) 发明名称

一种图片加载方法及装置

(57) 摘要

本发明涉及智能终端技术,特别涉及一种图片加载方法及装置,用以避免发生缓存污染,以及提高图片加载效率。该方法为:先将目标图片的索引保存在第一队列中,若目标图片被再次请求加载,则将目标图片的索引移至第二队列中,以及确定目标图片继续被请求加载时,基于第二队列中记录的目标图片的索引,获取相应的目标图片并进行显示,其中,每确定上述目标图片被请求加载一次,则将目标图片的索引由当前位置转存移至第二队列的头部,并定期清除第二队列末尾的数据。这样,可以避免由于偶发性批量加载操作而导致第一队列被清空,从而避免了发生缓存污染,同时又可以迅速从第一队列或第二队列获取目标图片并进行加载,进而有效提高了图片加载及显示效率。



1. 一种图片加载方法,其特征在于,包括:

接收图片加载请求,基于所述图片加载请求获取目标图片;

将所述目标图片的索引存储至内存缓存中的第一队列中;所述第一队列用于保存最新的被请求加载的目标图片的索引;

确定所述目标图片再次被请求加载时,将所述目标图片的索引从所述第一队列转存至内存缓存中的第二队列中,所述第二队列用于保存多次被请求加载的目标图片的索引;

确定所述目标图片继续被请求加载时,基于所述第二队列中记录的所述目标图片的索引,获取相应的目标图片并进行显示;

其中,在所述第二队列中,每确定所述目标图片被请求加载一次,将所述目标图片的索引由当前位置移至所述第二队列的头部,并定期清除所述第二队列末尾的数据;

其中,所述方法进一步包括:

对应所述目标图片绑定相应的释放标记,以及对应所述目标图片设置相应的弱指针容器;所述弱指针容器用于对所述释放标记进行监测;

通过所述弱指针容器获知所述释放标记被系统释放时,确定所述目标图片被系统释放,则释放所述弱指针容器;

其中,所述方法进一步包括:

确定所述目标图片再次被请求加载时,判断是否存在对应所述目标图片设置的弱指针容器;

若是,则通过所述弱指针容器获取所述目标图片并进行加载;

否则,在所述第一队列和第二队列中查找是否存在所述目标图片的索引,确定存在时,在所述第一队列或所述第二队列中基于所述目标图片的索引获取所述目标图片并进行加载。

2. 如权利要求1所述的方法,其特征在于,接收图片加载请求后,基于所述图片加载请求获取目标图片,包括:

基于所述图片加载请求,获得所述目标图片的统一资源定位URL地址;

基于所述URL地址,确定所述目标图片未保存于本地的缓存数据中时,基于所述URL地址从网络侧下载图片原始数据;

对所述图片原始数据进行解析,获得所述目标图片。

3. 如权利要求2所述的方法,其特征在于,基于所述URL地址从网络侧下载所述图片原始数据,包括:

获取所述URL地址包含的域名;

判断所述域名是否记录在本地;

若是,则基于已记录的域名和服务器IP地址之间的映射关系,获取所述域名对应的服务器IP地址,并基于所述服务器IP地址访问相应的服务器下载所述图片原始数据;

否则,将所述域名发往域名系统DNS服务器进行DNS解析,确定所述域名对应的服务器IP地址,并基于所述服务器IP地址访问相应的服务器下载所述图片原始数据,以及对所述服务器IP地址和所述域名之间的映射关系进行记录。

4. 如权利要求2所述的方法,其特征在于,对所述图片原始数据进行解析,获得所述目标图片,包括:

在下载所述图片原始数据的过程中,确定已下载数据量达到设定阈值后,开始对已下载的数据进行解码,直到所述图片原始数据下载完毕以及解码完毕;

按照指定的图片组件的显示区域的大小,对所述图片原始数据的解码数据进行调整,获得所述目标图片。

5.如权利要求2所述的方法,其特征在于,进一步包括:

将所述图片原始数据保存在磁盘缓存中;

以及将所述目标图片保存在内存中,并生成所述目标图片的索引。

6.如权利要求5所述的方法,其特征在于,将所述目标图片保存在内存中,包括:

根据中央处理器CPU的位数,采用字节对齐方式将所述目标图片保存在内存中。

7.一种图片加载装置,其特征在于,包括:

获取单元,用于接收图片加载请求后,基于所述图片加载请求获取目标图片;

处理单元,用于将所述目标图片的索引存储至内存缓存中的第一队列中;所述第一队列用于保存最新的被请求加载的目标图片的索引;以及用于确定所述目标图片再次被请求加载时,将所述目标图片的索引转存至内存缓存中的第二队列中,所述第二队列用于保存多次被请求加载的目标图片的索引;以及还用于确定所述目标图片继续被请求加载时,基于所述第二队列中记录的所述目标图片的索引,获取相应的目标图片并进行显示;

其中,在所述第二队列中,每确定所述目标图片被请求加载一次,将所述目标图片的索引由当前位置移至所述第二队列的头部,并定期清除所述第二队列末尾的数据;

所述处理单元,还用于对应所述目标图片绑定相应的释放标记,以及对应所述目标图片设置相应的弱指针容器;所述弱指针容器用于对所述释放标记进行监测;通过所述弱指针容器获知所述释放标记被系统释放时,确定所述目标图片被系统释放,则释放所述弱指针容器;

所述处理单元,还用于确定所述目标图片再次被请求加载时,判断是否存在对应所述目标图片设置的弱指针容器;若是,则通过所述弱指针容器获取所述目标图片并进行加载;否则,在所述第一队列和第二队列中查找是否存在所述目标图片的索引,确定存在时,在所述第一队列或所述第二队列中基于所述目标图片的索引获取所述目标图片并进行加载。

8.一种存储介质,其特征在于,存储用于实现图片存储的程序,所述程序被处理器运行时,执行以下步骤:

接收图片加载请求后,基于所述图片加载请求获取目标图片;

将所述目标图片的索引存储至内存缓存中的第一队列中;所述第一队列用于保存最新的被请求加载的目标图片的索引;

确定所述目标图片再次被请求加载时,将所述目标图片的索引转存至内存缓存中的第二队列中,所述第二队列用于保存多次被请求加载的目标图片的索引;

确定所述目标图片继续被请求加载时,基于所述第二队列中记录的所述目标图片的索引,获取相应的目标图片并进行显示;

其中,在所述第二队列中,每确定所述目标图片被请求加载一次,将所述目标图片的索引由当前位置移至所述第二队列的头部,并定期清除所述第二队列末尾的数据;

其中,所述程序被处理器运行时,进一步执行以下步骤:

对应所述目标图片绑定相应的释放标记,以及对应所述目标图片设置相应的弱指针容

器;所述弱指针容器用于对所述释放标记进行监测;

通过所述弱指针 容器获知所述释放标记被系统释放时,确定所述目标图片被系统释放,则释放所述弱指针 容器;

其中,所述程序被处理器运行时,进一步执行以下步骤:

确定所述目标图片再次被请求加载时,判断是否存在对应所述目标图片设置的弱指针容器;

若是,则通过所述弱指针容器获取所述目标图片并进行加载;

否则,在所述第一队列和第二队列中查找是否存在所述目标图片的索引,确定存在时,在所述第一队列或所述第二队列中基于所述目标图片的索引获取所述目标图片并进行加载。

9. 一种通信装置,其特征在于,包括一个或多个处理器;以及一个或多个计算机可读介质,所述可读介质上存储有指令,所述指令被所述一个或多个处理器执行时,使得所述装置执行如权利要求1至6中任一项所述的方法。

一种图片加载方法及装置

技术领域

[0001] 本发明涉及智能终端技术,特别涉及一种图片加载方法及装置。

背景技术

[0002] 目前,智能终端上的各类应用程序通常采用SDWebImage作为图片加载工具,SDWebImage用于为应用程序中嵌入的图像组件(UITableView)加载各类图片并且对图片的缓存进行管理。

[0003] 已有技术中,SDWebImage采用最近最少使用(Least Recently Used,LRU)队列作为内存缓存的基本数据结构。在LRU队列中,访问次数最少的数据会被末位淘汰,因此,现有的图片保存方式会存在以下问题:当存在偶发性批量操作时,会导致LRU队列清空,从而导致缓存数据意外丢失,进而在下一次访问的造成显示效率不高。

[0004] 例如,假设智能终端在运行某一应用程序并打开页面X后立即退出,从而造成了页面X的一次性访问,然而,即使是一次性访问,智能终端也会在下载页面X包含的各类图片,并加载至相应的图片组件内。

[0005] 通常情况下,智能终端会对首次访问的页面包含的各类图片进行全部下载及缓存,因此,智能终端会将从页面X下载的各类图片全部保存在LRU队列中,如果图片数量过多,则会造成LRU队列中的原始数据的清空,从而造成缓存污染;其中,若原始数据中保存有智能终端经常访问的网页Y包含的各类图片的缓存数据,那么,智能终端在下次再访问网页Y时,就需要重新下载图片,这会造成网页Y的显示效率不高,缓存图片命中率,并且智能终端需要根据用户的访问习惯,再经过相当长一段时间,才能重新收集恢复上述原始数据,进而造成了用户的使用不便,降低了用户体验。

发明内容

[0006] 本发明实施例提供一种图片加载方法及装置,用以在通过图片组件加载图片的过程中,避免发生缓存污染,以及提高图片加载效率。

[0007] 本发明实施例提供的具体技术方案如下:

[0008] 第一方面,一种图片加载方法,包括:

[0009] 接收图片加载请求后,基于所述图片加载请求获取目标图片;

[0010] 将所述目标图片的索引存储至内存缓存中的第一队列中;所述第一队列用于保存最新的被请求加载的目标图片的索引;

[0011] 确定所述目标图片再次被请求加载时,将所述目标图片的索引转存至内存缓存中的第二队列中,所述第二队列用于保存多次被请求加载的目标图片的索引;

[0012] 确定所述目标图片继续被请求加载时,基于所述第二队列中记录的所述目标图片的索引,获取相应的目标图片并进行显示;

[0013] 其中,在所述第二队列中,每确定所述目标图片被请求加载一次,将所述目标图片的索引由当前位置移至所述第二队列的头部,并定期清除所述第二队列末尾的数据。

- [0014] 可选的,接收图片加载请求后,基于所述图片加载请求获取目标图片,包括:
- [0015] 基于所述图片加载请求,获得所述目标图片的URL地址;
- [0016] 基于所述URL地址,确定所述目标图片未保存于本地的缓存数据中时,基于所述URL地址从网络侧下载图片原始数据;
- [0017] 对所述图片原始数据进行解析,获得所述目标图片。
- [0018] 可选的,基于所述URL地址从网络侧下载所述图片原始数据,包括:
- [0019] 获取所述URL地址包含的域名;
- [0020] 判断所述域名是否记录在本地;
- [0021] 若是,则基于已记录的域名和服务器IP地址之间的映射关系,获取所述域名对应的服务器IP地址,并基于所述服务器IP地址访问相应的服务器下载所述图片原始数据;
- [0022] 否则,将所述域名发往域名系统DNS服务器进行DNS解析,确定所述域名对应的服务器IP地址,并基于所述服务器IP地址访问相应的服务器下载所述图片原始数据,以及对所述服务器IP地址和所述域名之间的映射关系进行记录。
- [0023] 可选的,对所述图片原始数据进行解析,获得所述目标图片,包括:
- [0024] 在下载所述图片原始数据的过程中,确定已下载数据量达到设定阈值后,开始对已下载的数据进行解码,直到所述图片原始数据下载完毕以及解码完毕;
- [0025] 按照指定的图片组件的显示区域的大小,对所述图片原始数据的解码数据进行调整,获得所述目标图片。
- [0026] 可选的,进一步包括:
- [0027] 将所述图片原始数据保存在磁盘缓存中;
- [0028] 以及将所述目标图片保存在内存中,并生成所述目标图片的索引。
- [0029] 可选的,将所述目标图片保存在内存中,包括:
- [0030] 根据中央处理器CPU的位数,采用字节对齐方式将所述目标图片保存在内存中。
- [0031] 可选的,进一步包括:
- [0032] 对应所述目标图片绑定相应的释放标记,以及对应所述目标图片设置相应的弱指针容器;所述弱指针容器用于对所述释放标记进行监测;
- [0033] 通过所述弱指针容器获知所述释放标记被系统释放时,确定所述目标图片被系统释放,则释放所述弱指针容器。
- [0034] 可选的,进一步包括:
- [0035] 确定所述目标图片再次被请求加载时,判断是否存在对应所述目标图片设置的弱指针容器;
- [0036] 若是,则通过所述弱指针容器获取所述目标图片并进行加载;
- [0037] 否则,在所述第一队列和第二队列中查找是否存在所述目标图片的索引,确定存在时,在所述第一队列或所述第二队列中基于所述目标图片的索引获取所述目标图片并进行加载。
- [0038] 第二方面,一种图片加载装置,包括:
- [0039] 获取单元,用于接收图片加载请求后,基于所述图片加载请求获取目标图片;
- [0040] 处理单元,用于将所述目标图片的索引存储至内存缓存中的第一队列中;所述第一队列用于保存最新的被请求加载的目标图片的索引;以及用于确定所述目标图片再次被

请求加载时,将所述目标图片的索引转存至内存缓存中的第二队列中,所述第二队列用于保存多次被请求加载的目标图片的索引;以及还用于确定所述目标图片继续被请求加载时,基于所述第二队列中记录的所述目标图片的索引,获取相应的目标图片并进行显示;

[0041] 其中,在所述第二队列中,每确定所述目标图片被请求加载一次,将所述目标图片的索引由当前位置移至所述第二队列的头部,并定期清除所述第二队列末尾的数据。

[0042] 可选的,接收图片加载请求后,基于所述图片加载请求获取目标图片时,所述获取单元用于:

[0043] 基于所述图片加载请求,获得所述目标图片的URL地址;

[0044] 基于所述URL地址,确定所述目标图片未保存于本地的缓存数据中时,基于所述URL地址从网络侧下载图片原始数据;

[0045] 对所述图片原始数据进行解析,获得所述目标图片。

[0046] 可选的,基于所述URL地址从网络侧下载所述图片原始数据时,所述获取单元用于:

[0047] 获取所述URL地址包含的域名;

[0048] 判断所述域名是否记录在本地;

[0049] 若是,则基于已记录的域名和服务器IP地址之间的映射关系,获取所述域名对应的服务器IP地址,并基于所述服务器IP地址访问相应的服务器下载所述图片原始数据;

[0050] 否则,将所述域名发往域名系统DNS服务器进行DNS解析,确定所述域名对应的服务器IP地址,并基于所述服务器IP地址访问相应的服务器下载所述图片原始数据,以及对所述服务器IP地址和所述域名之间的映射关系进行记录。

[0051] 可选的,对所述图片原始数据进行解析,获得所述目标图片时,所述获取单元用于:

[0052] 在下载所述图片原始数据的过程中,确定已下载数据量达到设定阈值后,开始对已下载的数据进行解码,直到所述图片原始数据下载完毕以及解码完毕;

[0053] 按照指定的图片组件的显示区域的大小,对所述图片原始数据的解码数据进行调整,获得所述目标图片。

[0054] 可选的,所述处理单元进一步用于:

[0055] 将所述图片原始数据保存在磁盘缓存中;

[0056] 以及将所述目标图片保存在内存中,并生成所述目标图片的索引。

[0057] 可选的,将所述目标图片保存在内存中时,所述处理单元用于:

[0058] 根据中央处理器CPU的位数,采用字节对齐方式将所述目标图片保存在内存中。

[0059] 可选的,所述处理单元进一步用于:

[0060] 对应所述目标图片绑定相应的释放标记,以及对应所述目标图片设置相应的弱指针容器;所述弱指针容器用于对所述释放标记进行监测;

[0061] 通过所述弱指针容器获知所述释放标记被系统释放时,确定所述目标图片被系统释放,则释放所述弱指针容器。

[0062] 可选的,所述处理单元进一步用于:

[0063] 确定所述目标图片再次被请求加载时,判断是否存在对应所述目标图片设置的弱指针容器;

[0064] 若是,则通过所述弱指针容器获取所述目标图片并进行加载;

[0065] 否则,在所述第一队列和第二队列中查找是否存在所述目标图片的索引,确定存在时,在所述第一队列或所述第二队列中基于所述目标图片的索引获取所述目标图片并进行加载。

[0066] 第三方面,一种存储介质,存储用于实现图片存储的程序,所述程序被处理器运行时,执行以下步骤:

[0067] 接收图片加载请求后,基于所述图片加载请求获取目标图片;

[0068] 将所述目标图片的索引存储至内存缓存中的第一队列中;所述第一队列用于保存最新的被请求加载的目标图片的索引;

[0069] 确定所述目标图片再次被请求加载时,将所述目标图片的索引转存至内存缓存中的第二队列中,所述第二队列用于保存多次被请求加载的目标图片的索引;

[0070] 确定所述目标图片继续被请求加载时,基于所述第二队列中记录的所述目标图片的索引,获取相应的目标图片并进行显示;

[0071] 其中,在所述第二队列中,每确定所述目标图片被请求加载一次,将所述目标图片的索引由当前位置移至所述第二队列的头部,并定期清除所述第二队列末尾的数据。

[0072] 第四方面,一种通信装置,包括一个或多个处理器;以及一个或多个计算机可读介质,所述可读介质上存储有指令,所述指令被所述一个或多个处理器执行时,使得所述装置执行第一方面中提及的任一项所述的方法。

[0073] 本发明实施例中,在对目标图片存储时,采用双队列缓存方式保存目标图片,即先将目标图片的索引保存在第一队列中,若目标图片被再次请求加载,则将目标图片的索引移至第二队列中,以及在确定目标图片继续被请求加载时,基于第二队列中记录的目标图片的索引,获取相应的目标图片并进行显示,其中,在第二队列中,每确定上述目标图片被请求加载一次,则将目标图片的索引由当前位置转存至第二队列的头部,并定期清除第二队列末尾的数据。这样,可以避免由于偶发性批量加载操作而导致第一队列被清空,从而避免了发生缓存污染,当目标图片被再次请求访问时,可以迅速从第一队列或第二队列获取目标图片并进行加载,进而有效提高了图片加载及显示效率。

附图说明

[0074] 图1为本发明实施例中图片加载界面示意图;

[0075] 图2A为本发明实施例中图片加载方法基本流程示意图;

[0076] 图2B为本发明实施例中图片加载方法详细流程示意图;

[0077] 图3为本发明实施例中双队列算法示意图;

[0078] 图4为本发明实施例中智能缓存工作原理示意图;

[0079] 图5和图6为本发明实施例中智能终端功能结构示意图。

具体实施方式

[0080] 为了避免发生缓存污染,以及提高图片加载效率,本发明实施例中,采用双队列对下载的图片进行缓存,从网页上下载的各类图片均先保存至第一队列中,而只有在第一队列中被再次访问的图片,才会被存入第二队列,而第二队列采用基于访问次数的末位淘汰

制,将不经常访问的图片从第二队列中删除。

[0081] 下面结合附图对本发明优选的实施例作出进一步详细说明。

[0082] 参阅图1所示,在一个应用程序的操作界面中,运行着各种图片组件,智能终端便需要从网络、磁盘缓存、内存缓存等存储位置加载相关的图片加载至各类图片组件。

[0083] 参阅图2A所示,本发明实施例中,对目标图片进行存储以及加载的基本流程如下:

[0084] 步骤20:接收图片加载请求,基于图片加载请求获取目标图片;

[0085] 步骤21:将目标图片的索引存储至内存缓存中的第一队列中;第一队列用于保存最新的被请求加载的目标图片的索引。

[0086] 可选的,第一队列可以采用先入先出(First Input First Output,FIFO)队列,后续实施例中将沿用这一称谓,将不再赘述。

[0087] 另一方面,所谓最新的被请求加载的目标图片,是指智能终端最新接收到的图片加载请求所关联的图片,即每接收一次图片加载请求,就需要将相应的目标图片的索引在第一队列中保存一次。

[0088] 例如,假设FIFO队列中最多可保存3张图片可保存,那么:

[0089] 假设智能终端启动后,依次接收到针对图片1、图片2和图片3的图片加载请求,那么,智能终端就会将图片1、图片2和图片3的索引保存在FIFO队列中。

[0090] 接着,假设智能终端又接收到图片4的图片加载请求,那么,智能终端就会采用图片4的索引对最先接收的图片1的索引进行覆盖,此时,FIFO队列中保存了图片2、图片3和图片4的索引。

[0091] 之后,假设智能终端又在同一时期接收到图片5、图片6和图片7的图片加载请求,

[0092] 那么,智能终端就会采用图片5、图片6和图片7的索引对图片2、图片3和图片4的索引进行覆盖。

[0093] ……

[0094] 以类类推,FIFO中一直保存的是最新的被请求加载的目标图片的索引。

[0095] 步骤22:确定上述目标图片再次被请求加载时,将上述目标图片的索引从第一队列转存至内存缓存中的第二队列中,第二队列用于保存多次被请求加载的目标图片的索引。

[0096] 可选的,第二队列可以采用LRU队列,后续实施例中将沿用这一称谓,将不再赘述。

[0097] 另一方面,所谓多次被请求加载的目标图片,是指至少两次作为目标图片被请求加载的图片,即一张图片只有被请求加载至少两次,智能终端才会将这一张图片由FIFO队列转移保存至LRU队列中。

[0098] 例如,假设FIFO队列中当前保存了图片5、图片6和图片7的索引,而此时智能终端再次接收到了针对图5的图片加载请求,那么,智能终端会将图5的索引由FIFO队列转移至LRU队列中。

[0099] 步骤23:确定上述目标图片继续被请求加载时,基于第二队列中记录的上述目标图片的索引,获取相应的目标图片并进行显示;其中,在第二队列中,每确定上述目标图片被请求加载一次,将上述目标图片的索引由当前位置移至所述第二队列的头部,并定期清除第二队列末尾的数据。

[0100] 具体的,在将目标图片的索引由FIFO队列转存至LRU队列之后,目标图片还有可能

继续被请求加载,那么,智能终端每确定目标图片被请求加载一次,就会将目标图片的索引重新存储至LRU队列的头部,这样,可以保证被请求加载次数相对较多的目标图片的索引被保留在LRU队列的头部,而被请求加载次数相对较少的目标图片的索引则被保留在LRU队列的尾部,被请求加载次数最少的目标图片将会被定期清空,以避免占用LRU队列资源的时间过长。

[0101] 本发明实施例中,上述FIFO队列和LRU队列均是设置在内存缓存中的队列。

[0102] 进一步地,在图片批量请求加载的场景下,内存缓存可能会由于资源过少而无法支持批量保存目标图片的索引,针对这一情况,可选的,智能终端可以将目标图片的原始数据保存在磁盘缓存中,后续流程中可以按照目标图片的索引直接大磁盘缓存中调用原始数据进行解码处理后进行显示。

[0103] 更进一步地,进一步地,还可以通过在内存中配置智能缓存,针对目标图片设置弱指针容器,用于对目标图片的资源占用情况进行监测,只要通过弱指针容器确定目标图片未被系统释放,即可以迅速获取到目标图片并进行加载及显示。

[0104] 参阅图2B所示,本发明实施例中,对图片进行加载及存储的详细流程如下:

[0105] 步骤200:智能终端接收图片加载请求,该图片加载请求中携带有目标图片的URL地址。

[0106] 具体的,智能终端启动某一应用程序后,应用程序中的各类图片组件便开始运行,运行过程中需要开始加载图片,待加载的各个图片的统一资源定位符(Uniform Resource Locator,URL)地址可以是默认设置的,也可以是由用户自定义设置的,那么,应用程序便可以将携带有图片的URL地址的图片加载请求发送给智能终端。

[0107] 步骤201:智能终端在缓存数据中查找是否已存在目标图片,若是,则执行步骤202;否则,执行步骤203。

[0108] 本发明实施例中,智能终端会按照以下顺序在各类缓存中查找目标图片:智能缓存->内存缓存->磁盘缓存;智能缓存和内存缓存均为内存的一部分,智能缓存用于对目标图片的使用情况进行监测,具体作用将在后续实施例中进行详细介绍。

[0109] 步骤202:智能终端在缓存数据中获取目标图片,接着,执行步骤213。

[0110] 可选的,智能缓存中保存的是目标图片的已解码及已经图片处理的数据,可以直接作为目标图片使用。

[0111] 同理,内存缓存中保存的也是目标图片的已解码及已经图片处理的数据,亦可以直接作为目标图片使用。

[0112] 而磁盘缓存中保存的是目标图片的原始数据,因此,需要对原始数据进行文件解码及进行图片处理后,获得目标图片再使用。

[0113] 另一方面,保存在智能缓存、内存缓存及磁盘缓存中的目标图片的文件名,可以是目标图片的URL地址的MD5值。

[0114] 步骤203:智能终端生成向网络侧发送的图片下载请求,该图片下载请求中携带有目标图片的URL地址。

[0115] 步骤204:智能终端判断图片下载请求中的URL地址包含的域名对应的IP地址是否已保存在本地,若是,则执行步骤206;否则,执行步骤205。

[0116] 通常情况下,智能终端在首次接收到某一图片的图片下载请求时,会将图片下载

请求中携带的目标图片的URL地址中包含的域名发往域名系统(Domain Name Server,DNS)服务器进行DNS解析,获得上述域名对应的服务器IP地址,再按照该服务器IP地址发起http请求或https请求,从相应的服务器下载所需的目标图片。

[0117] 步骤205:智能终端将图片下载请求中URL地址包含的域名发往DNS服务器进行域名解析,获得相应的服务器IP地址。

[0118] 步骤206:智能终端将图片下载请求中URL地址包含的域名替换为对应的服务器IP地址。

[0119] 智能终端通常会在本地维护一个域名和服务器IP地址之间的查询映射表,智能终端再次接收到图片下载请求时,会先判断图片下载请求中携带的URL地址包含的域名是否记载在本地的查询映射表中,确定存在时,则会直接采用域名对应的服务器IP地址对URL地址中的域名进行替换,并按照该服务器IP地址发起http请求或https请求,从相应的服务器下载所需的目标图片。

[0120] 这样,可以避免再次进行DNS解析,令智能终端可以直接访问服务器进行图片下载,从而达到了加快图片下载速度的目的。

[0121] 步骤207:智能终端向服务器发送图片下载请求,从服务器下载目标图片的原始数据。

[0122] 步骤208:智能终端将目标图片的原始数据保存在磁盘缓存中。

[0123] 本发明实施例中,在磁盘缓存中保存目标图片的原始数据,是为了防止在短时间内突发批量下载,而内存缓存无力承担的情况发生。

[0124] 在磁盘缓存中保存目标图片的原始数据后,其文件名可以记录为相应的URL地址的MD5值,这样,智能终端在下次接收到应用程序发出的图片加载请求时,便可以通过图片加载请求中携带的URL地址的MD5值,来判断本地是否缓存数据中是否保存有相应的目标图片(对应于步骤101),若存在则可以直接提取后进行加载,从而能大幅度提升图片的二次加载速度。

[0125] 步骤209:智能终端对目标图片的原始数据进行文件解码。

[0126] 本发明实施例中,智能终端会维护一个任务队列和一个线程池,在目标图片的原始数据需要被解码时,智能终端会将任务加入任务队列中,而任务队列中正在运行的任务数量,小于当前CPU的核心数的时候,智能终端会从线程池中取出一个未被占用的线程,分配给下一个任务,保证智能模块的解码硬件总是以最高效率在运行。

[0127] 进一步地,已有技术中,通知是将目标图片的原始数据全部下载后再开始进行解码,这会造成解码效率低,速度慢等问题。可选的,本发明实施例中,智能终端在下载目标图片的原始数据的过程中,可以一边保存至磁盘缓存,一边进行文件解码,如,已下载数据量达到设定阈值后,便可以开始执行文件解码操作,这样,可以有效提升文件解码效率。

[0128] 步骤210:智能终端对目标图片的解码数据进行图片处理,获得目标图片,并保存在内存中。

[0129] 可选的,所谓图片处理,即是指按照图片控件的显示区域大小对目标图片的解码数据进行调整,转变为显示区域的大小,从而获得目标图片。

[0130] 当然,若下载的目标图片的数量不多,或者,下载的目标图片的图片数据无需解码及进行图片调整,则也可以不将目标图片保存至磁盘缓存中,而是直接执行步骤211保存至

内存缓存中,以方便后续的快速操作,在此不再赘述。

[0131] 步骤211:智能终端将目标图片的索引作为缓存数据保存在内存缓存中。

[0132] 具体的,内存缓存为内存中的一部分,目标图片保存在内存中,而目标图片的索引则保存在内存缓存中,这样,在后续流程中,当目标图片再次被请求加载时,智能终端可以直接在内存缓存中查找是否存在目标图片的索引,并链接至相应的存储位置提取目标图片。

[0133] 进一步地,在内存中保存目标图片时,需要根据CPU的位数,采用字节对齐方式保存目标图片,即将目标图片的数据绘制拷贝到字节对齐的区域中。

[0134] 字节(Byte)是计算机信息技术用于计量存储容量和传输容量的一种计量单位,一个字节等于8位二进制数,而字节按照设定规则在空间上排列就是字节对齐。

[0135] 例如,假设CPU位数为32位,且预设规则为:以32位为单位进行字节对齐;那么,假设目标图片的数据大小为18位,就需要进行位数补充至32位后再进行保存,又假设目标图片为40位,就需要进行位数补充致64位后再进行保存。

[0136] 另一方面,在内存缓存中保存目标图片的索引时,可选的,可以采用双队列模式。具体如下:

[0137] 参阅图3所示,设置一个FIFO队列及一个LRU队列以及一个哈希地图(hashmap)组成,使用双队列缓存淘汰算法解决缓存污染问题,其中,hashmap采用key-value形式构建,用于保存目标图片的索引和存储位置之间的映射关系。

[0138] 具体的,智能终端会将初次获得的各个目标图片插入至FIFO队列中,本发明实施例中,FIFO队列用于保存最新的被请求加载的目标图片的索引;

[0139] 在FIFO队列中,若智能终端确定一张目标图片被再次请求加载,则将会上述一张目标图片的索引移至LRU队列中,可选的,移至LRU队列的头部;本发明实施例中,LRU队列用于保存多次(至少两次)被请求加载的目标图片的索引;

[0140] 按照这种方式,在FIFO队列中一直没有被再次请求加载的目标图片的索引,会最终按照FIFO规则被淘汰;

[0141] 而在LRU队列中,智能终端每确定一张目标图片被再次请求加载一次,则会将上述一张目标图片的索引由当前位置移至LRU队列的头部,即LRU队列头部的索引对应的目标图片,均是最新的被再次请求加载的目标图片。

[0142] 进一步地,智能终端会按照设定周期淘汰LRU队列末尾的目标图片的索引。

[0143] 当目标图片再次被请求加载时,智能终端会在FIFO队列或LRU队列中查找是否存在对应的索引,如果存在,则基于索引和上述hashmap,查找到目标图片在内存中的存储位置,以便读取目标图片。

[0144] 基于上述方法,在FIFO队列中,只有最少被访问两次的目标图片的索引才会由FIFO队列移到LRU队列,而在LRU队列中,只有被访问次数最多的目标图片的索引才会保持在LRU队列的头部,从而避免被淘汰。这样,即使存在偶发性批量操作时,也只会导致FIFO队列被清空,而FIFO队列中经常被访问的目标图片的索引早已移致LRU队列中,从而避免了LRU队列中缓存数据的意外丢失,令LRU队列始终保持缓存清洁,进而在下一次访问的有效保证了目标图片的显示效率。

[0145] 步骤212:智能终端将用于监测目标图片的弱指针容器作为缓存数据保存在智能

缓存中。

[0146] 本发明实施例中,智能缓存也是内存的一部分,之所以要建立智能缓存,是为了采用资源弱引用的方式对内存资源的释放进行监视维护,使得智能缓存几乎不再占用额外的存储空间,且能够在合适的时机自动被清除,达到节约资源,以及提高图片显示效率的作用。

[0147] 参阅图4所示,智能缓存由弱指针索引表、弱指针容器、目标图片(内存中的缓存数据)、释放通知器(又可称为释放标记)四个部分组成。在智能缓存中维护了一个弱指针索引表,这是一个Hash表,在弱指针索引表中每一个元素都是一个弱指针容器,一个弱指针容器可以弱引用了一个目标图片,并将一个释放通知器绑定在目标图片上。

[0148] 所谓弱引用即是指,对目标图片的引用关系不会导致目标图片无法被系统释放。由于无法对目标图片直接进行监测(会导致目标图片无法被释放),因此,弱指针容器可以通过对释放通知器的资源使用情况进行监测,从而实现对与释放通知器绑定的目标图片的监测,即通过对已知对象的监测实现对未知对象的监测。

[0149] 当目标图片被系统释放时,与目标图片绑定的释放通知器将通知相应的弱指针容器,而弱指针容器会将自身从弱指针索引表中移除,若目标图片始终未被系统释放,那么,当目标图片再次被请求加载时,智能终端会直接通过智能缓存中对目标图片进行弱引用的弱指针容器,在系统中找到相应的目标图片并进行加载。

[0150] 简言之,智能终端会对应目标图片绑定相应的释放标记(即释放通知器),以及对目标图片设置相应的弱指针容器,并通过弱指针容器用于对释放标记进行监测,以及在通过弱指针容器获知释放标记被系统释放时,确定目标图片被系统释放,进而释放相应的弱指针容器。

[0151] 实际应用中,由于在内存缓存中的FIFO队列和LRU队列会按照设定规则删除队列中的数据,因此,FIFO队列和LRU队列中保存的数据量有限,那么,某些目标图片即使没有被系统释放,其相应的索引也有可能已不在FIFO队列和LRU队列中保存了,因此,本发明实施例中,采用智能缓存这种无额外内存占用的弱引用缓存,可以直接找到未被系统释放的目标图片,令目标图片再次被请求加载时,无需重新下载和解码,极大地提升了图片显示速率。

[0152] 另一方面,实际应用中,若系统中展示的目标图片数量不多或者目标图片的被访问次数不频繁,内存缓存足以支撑目标图片的缓存操作时,也可以不设置智能缓存,在此不再赘述。

[0153] 步骤213:智能终端在图片组件中显示目标图片。

[0154] 基于上述实施例,实际应用中,当步骤213中提及的目标图片再次被请求加载时(即再次被访问)时,智能终端会执行以下操作:

[0155] 先在智能缓存中判断是否存在对应目标图片设置的弱指针容器;

[0156] 若是,则通过弱指针容器获取目标图片并进行加载;

[0157] 此时,说明系统并未释放目标图片,可以通过弱指针容器直接获取。

[0158] 否则,在上述FIFO队列或LRU队列中查找是否存在目标图片的索引,确定存在时,基于目标图片的索引在相应的存储位置获取目标图片并进行加载。进一步地,若FIFO队列或LRU队列中也没有保存目标图片的索引,则智能终端会按照目标图片的文件名(如,URL地

址的MD5值)在磁盘缓存中查找是否保存有目标图片的原始数据,若存在,则再次对目标图片的原始数据进行解码和图片处理,获得目标图片后进行加载,若不存在,则再次向网络侧发送图片下载请求,重新下载并加载目标图片,在此不再赘述。

[0159] 参阅图5所示,本发明实施例中,图片加载装置(如,智能终端)至少包括获取单元50和处理单元51,其中,

[0160] 获取单元50,用于接收图片加载请求后,基于所述图片加载请求获取目标图片;

[0161] 处理单元51,用于将所述目标图片的索引存储至内存缓存中的第一队列中;所述第一队列用于保存最新的被请求加载的目标图片的索引;以及用于确定所述目标图片再次被请求加载时,将所述目标图片的索引转存至内存缓存中的第二队列中,所述第二队列用于保存多次被请求加载的目标图片的索引;以及还用于确定所述目标图片继续被请求加载时,基于所述第二队列中记录的所述目标图片的索引,获取相应的目标图片并进行显示;

[0162] 其中,在所述第二队列中,每确定所述目标图片被请求加载一次,将所述目标图片的索引由当前位置移至所述第二队列的头部,并定期清除所述第二队列末尾的数据。

[0163] 可选的,接收图片加载请求后,基于所述图片加载请求获取目标图片时,获取单元50用于:

[0164] 基于所述图片加载请求,获得所述目标图片的URL地址;

[0165] 基于所述URL地址,确定所述目标图片未保存于本地的缓存数据中时,基于所述URL地址从网络侧下载图片原始数据;

[0166] 对所述图片原始数据进行解析,获得所述目标图片。

[0167] 可选的,基于所述URL地址从网络侧下载所述图片原始数据时,获取单元50用于:

[0168] 获取所述URL地址包含的域名;

[0169] 判断所述域名是否记录在本地;

[0170] 若是,则基于已记录的域名和服务器IP地址之间的映射关系,获取所述域名对应的服务器IP地址,并基于所述服务器IP地址访问相应的服务器下载所述图片原始数据;

[0171] 否则,将所述域名发往域名系统DNS服务器进行DNS解析,确定所述域名对应的服务器IP地址,并基于所述服务器IP地址访问相应的服务器下载所述图片原始数据,以及对所述服务器IP地址和所述域名之间的映射关系进行记录。

[0172] 可选的,对所述图片原始数据进行解析,获得所述目标图片时,获取单元50用于:

[0173] 在下载所述图片原始数据的过程中,确定已下载数据量达到设定阈值后,开始对已下载的数据进行解码,直到所述图片原始数据下载完毕以及解码完毕;

[0174] 按照指定的图片组件的显示区域的大小,对所述图片原始数据的解码数据进行调整,获得所述目标图片。

[0175] 可选的,处理单元51进一步用于:

[0176] 将所述图片原始数据保存在磁盘缓存中;

[0177] 以及将所述目标图片保存在内存中,并生成所述目标图片的索引。

[0178] 可选的,将所述目标图片保存在内存中时,处理单元51用于:

[0179] 根据中央处理器CPU的位数,采用字节对齐方式将所述目标图片保存在内存中。

[0180] 可选的,处理单元51进一步用于:

[0181] 对应所述目标图片绑定相应的释放标记,以及对应所述目标图片设置相应的弱指

针容器;所述弱指针容器用于对所述释放标记进行监测;

[0182] 通过所述弱指针容器获知所述释放标记被系统释放时,确定所述目标图片被系统释放,则释放所述弱指针容器。

[0183] 可选的,处理单元51进一步用于:

[0184] 确定所述目标图片再次被请求加载时,判断是否存在对应所述目标图片设置的弱指针容器;

[0185] 若是,则通过所述弱指针容器获取所述目标图片并进行加载;

[0186] 否则,在所述第一队列和第二队列中查找是否存在所述目标图片的索引,确定存在时,在所述第一队列或所述第二队列中基于所述目标图片的索引获取所述目标图片并进行加载。

[0187] 基于同一发明构思,本发明一个实施例中,提供一种存储介质,存储用于实现图片存储的程序,所述程序被处理器运行时,执行以下步骤:

[0188] 接收图片加载请求后,基于所述图片加载请求获取目标图片;

[0189] 将所述目标图片的索引存储至内存缓存中的第一队列中;所述第一队列用于保存最新的被请求加载的目标图片的索引;

[0190] 确定所述目标图片再次被请求加载时,将所述目标图片的索引移至内存缓存中的第二队列中,所述第二队列用于保存多次被请求加载的目标图片的索引;

[0191] 其中,在所述第二队列中,每确定所述目标图片被请求加载一次,将所述目标图片的索引由当前位置转存至所述第二队列的头部,并定期清除所述第二队列末尾的数据。

[0192] 基于同一发明构思,参阅图6所示,本发明一个实施例中,提供一种通信装置,包括一个或多个处理器;以及一个或多个计算机可读介质,所述可读介质上存储有指令,所述指令被所述一个或多个处理器执行时,使得所述装置执行上述实施例中提及的任一种方法。

[0193] 具体的,可读介质可以是磁盘缓存、内存缓存以及智能缓存。

[0194] 本发明实施例中,在对目标图片存储时,采用双队列缓存方式保存目标图片,即先将目标图片的索引保存在第一队列中,若目标图片被再次请求加载,则将目标图片的索引移至第二队列中,以及确定目标图片继续被请求加载时,基于第二队列中记录的目标图片的索引,获取相应的目标图片并进行显示,其中,在第二队列中,每确定上述目标图片被请求加载一次,则将目标图片的索引由当前位置移至第二队列的头部,并定期清除第二队列末尾的数据。这样,可以避免由于偶发性批量加载操作而导致第一队列被清空,从而避免了发生缓存污染,当目标图片被再次请求访问时,可以迅速从第一队列或第二队列获取目标图片并进行加载,进而有效提高了图片加载及显示效率。

[0195] 进一步地,还可以通过配置智能缓存,针对目标图片设置弱指针容器,用于对目标图片的资源占用情况进行监测,当弱指针容器确定与目标图片绑定的释放标记被释放进,即判定目标图片被系统释放,进而将自身占用的内存资源也进行释放,这样,当目标图片被再次请求访问时,只要通过弱指针容器确定目标图片未被系统释放,即可以迅速获取到目标图片并进行加载及显示,从而即减少了目标图片在内存中占用的空间,又进一步提升了图片的加载及显示效率。

[0196] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实

施例的形式。而且,本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0197] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0198] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0199] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0200] 尽管已描述了本发明的优选实施例,但本领域内的技术人员一旦得知了基本创造性概念,则可对这些实施例作出另外的变更和修改。所以,所附权利要求意欲解释为包括优选实施例以及落入本发明范围的所有变更和修改。

[0201] 显然,本领域的技术人员可以对本发明实施例进行各种改动和变型而不脱离本发明实施例的精神和范围。这样,倘若本发明实施例的这些修改和变型属于本发明权利要求及其等同技术的范围之内,则本发明也意图包含这些改动和变型在内。



图1

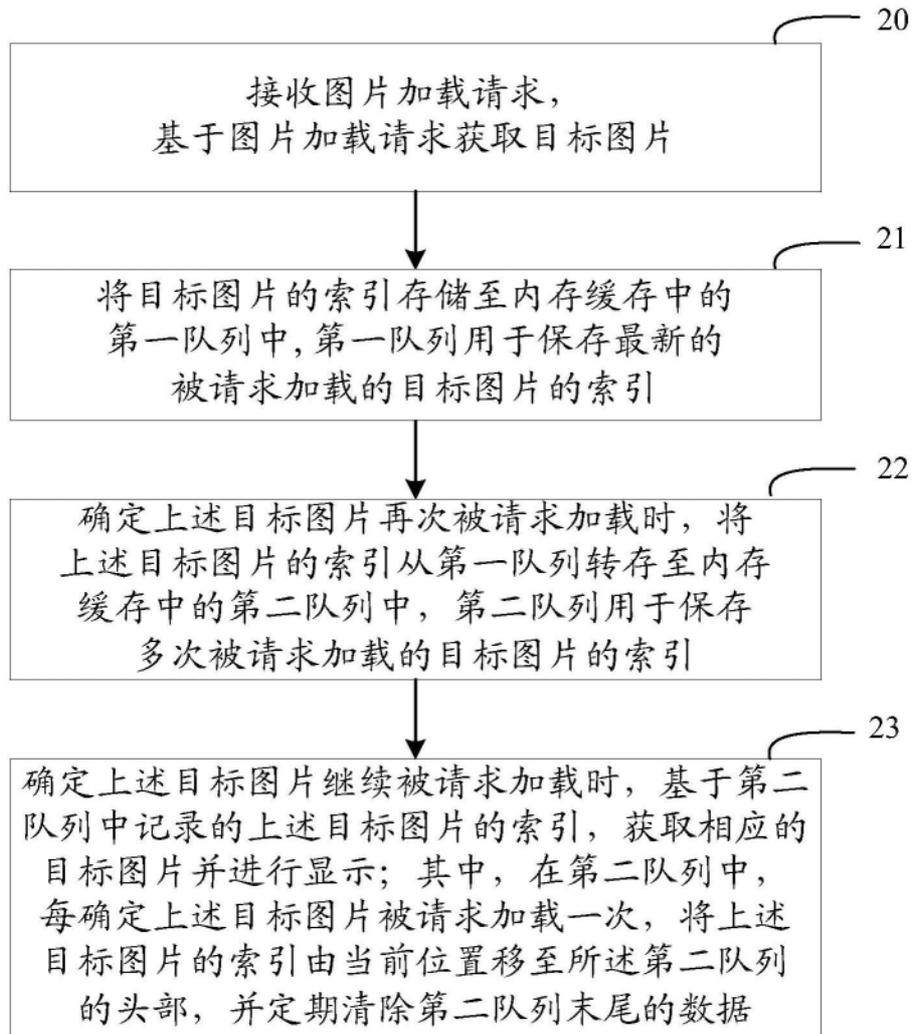


图2A

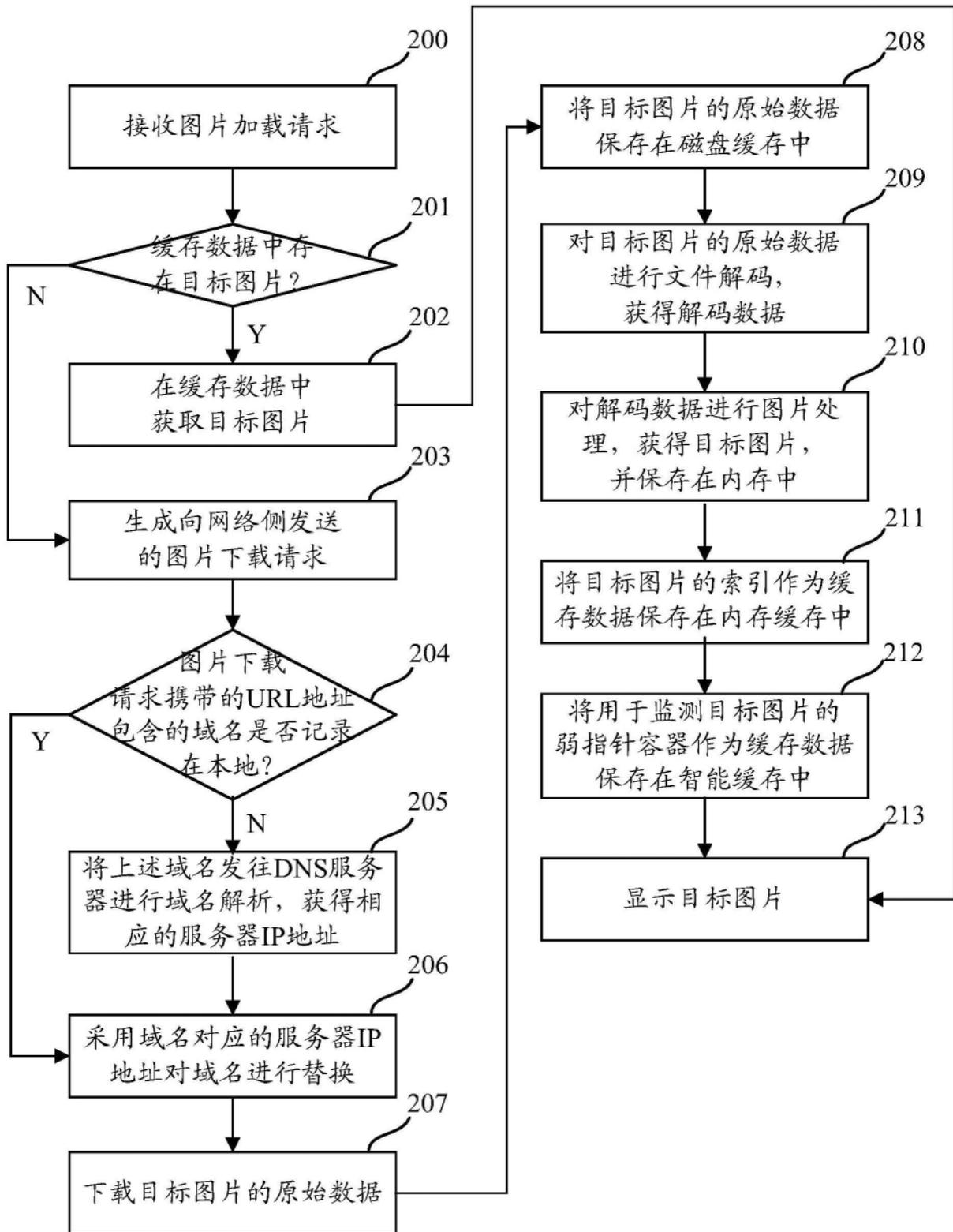


图2B

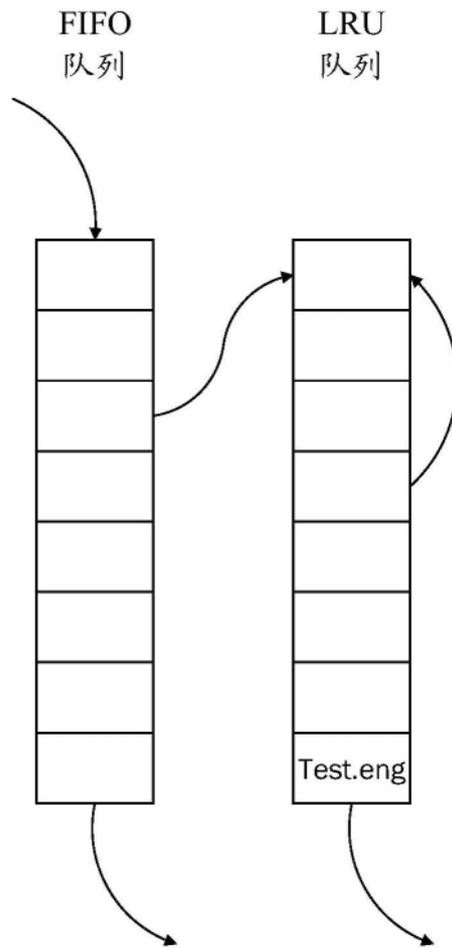


图3

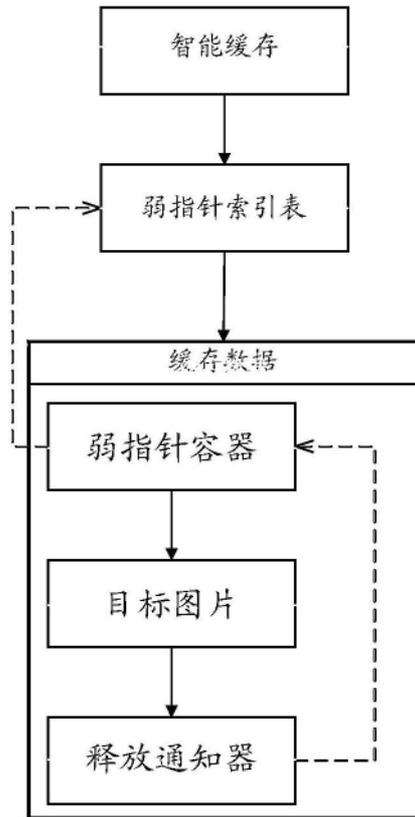


图4

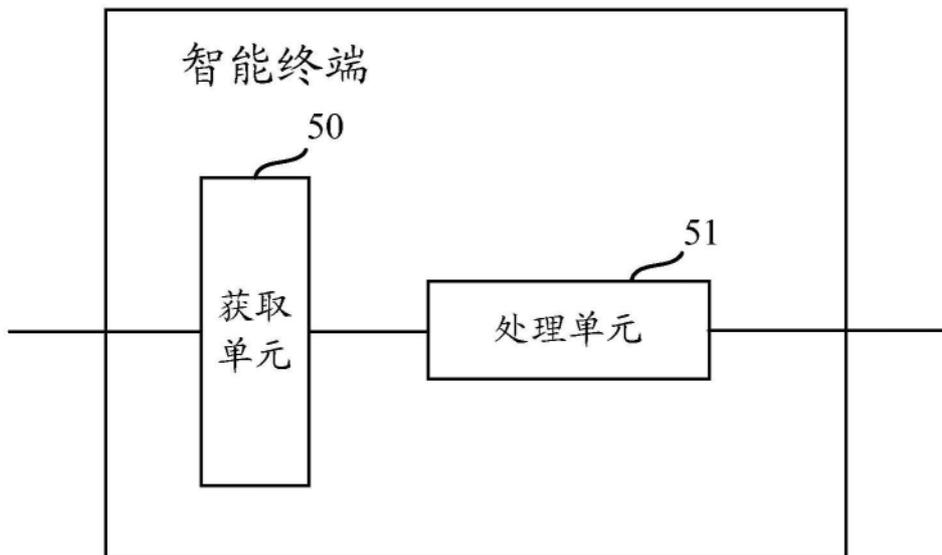


图5

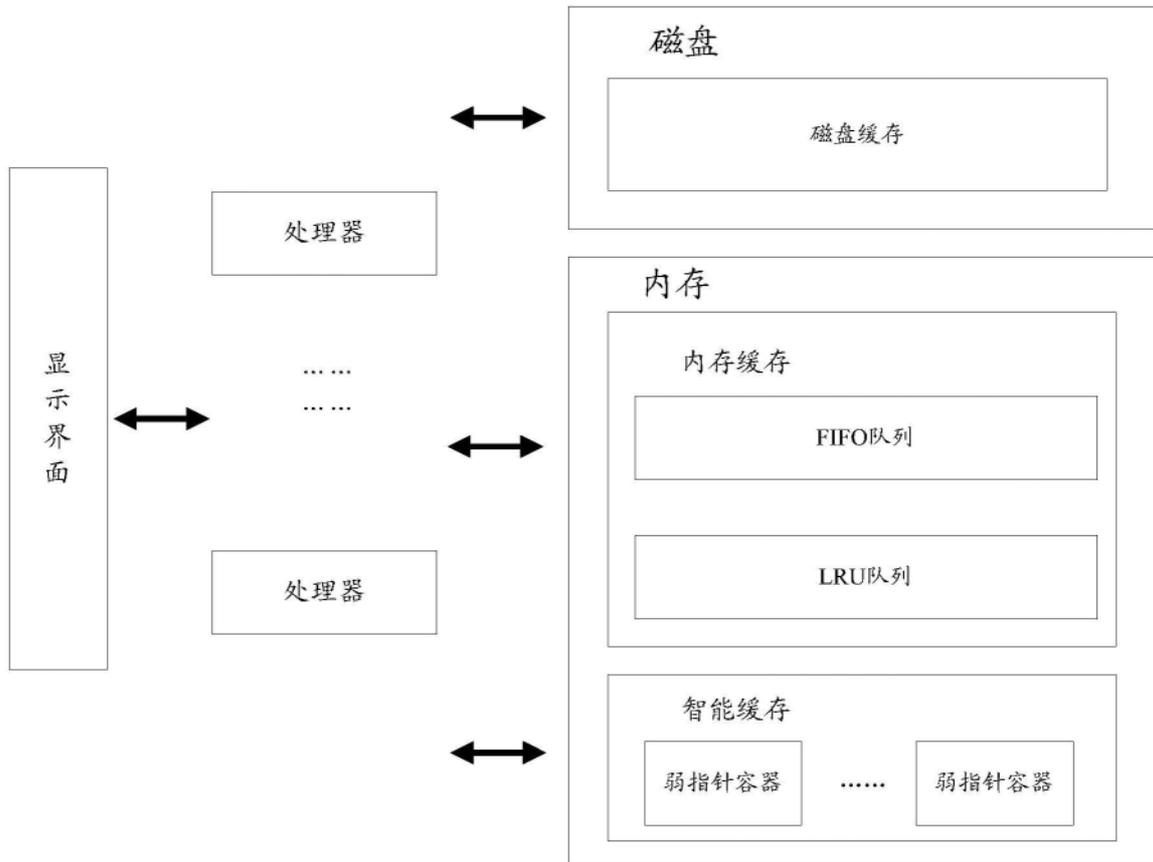


图6