



(12) 发明专利

(10) 授权公告号 CN 113536168 B

(45) 授权公告日 2023. 11. 14

(21) 申请号 202110618041.7
 (22) 申请日 2021.06.03
 (65) 同一申请的已公布的文献号
 申请公布号 CN 113536168 A
 (43) 申请公布日 2021.10.22
 (73) 专利权人 中国联合网络通信集团有限公司
 地址 100033 北京市西城区金融大街21号
 (72) 发明人 王尚志
 (74) 专利代理机构 北京同立钧成知识产权代理有限公司 11205
 专利代理师 张娜 臧建明
 (51) Int. Cl.
 G06F 16/957 (2019.01)
 G06F 16/958 (2019.01)
 G06F 9/445 (2018.01)
 (56) 对比文件
 CN 104461632 A, 2015.03.25
 CN 110020388 A, 2019.07.16

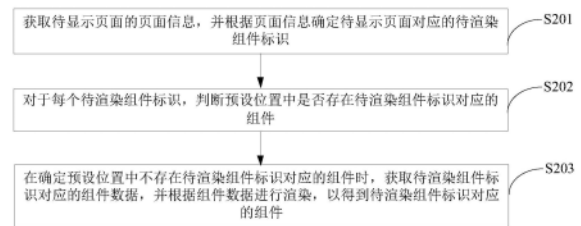
CN 111443912 A, 2020.07.24
 US 2018107632 A1, 2018.04.19
 CN 109241469 A, 2019.01.18
 CN 111581563 A, 2020.08.25
 CN 111708600 A, 2020.09.25
 知乎.vue组件缓存之keep-alive正确使用姿势.《https://www.zhihu.com/tardis/bd/art/352059246?source_id=1001》, 2021, 第1-6页.
 Banatus soiraya 等.An analysis of visual and presentation factors influencing the design of e-commerce web sites.《2008 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology》.2009, 第1-2页.
 潘文文.云平台下Web前端的若干优化技术的研究与实现.《中国优秀硕士学位论文全文数据库信息科技辑》.2020, 第30、36-48页.

审查员 乔晋

权利要求书2页 说明书11页 附图3页

(54) 发明名称
 组件处理方法及设备

(57) 摘要
 本发明提供一种组件处理方法及设备,所述方法包括:获取待显示页面的页面信息,并根据所述页面信息确定所述待显示页面对应的待渲染组件标识;对于每个待渲染组件标识,判断预设位置中是否存在所述待渲染组件标识对应的组件;在确定预设位置中不存在所述待渲染组件标识对应的组件时,获取所述待渲染组件标识对应的组件数据,并根据所述组件数据进行渲染,以得到所述待渲染组件标识对应的组件,无需对已存在的组件重复加载渲染,减少组件的重复加载,从而可以减少资源的浪费,且还可以缩短页面显示所需的时间,提高页面的访问速度,用户体验高。



CN 113536168 B

1. 一种组件处理方法,其特征在于,包括:

获取待显示页面的页面信息,并根据所述页面信息确定所述待显示页面对应的待渲染组件标识;

对于每个待渲染组件标识,判断预设位置中是否存在所述待渲染组件标识对应的组件;

在确定预设位置中不存在所述待渲染组件标识对应的组件时,获取所述待渲染组件标识对应的组件数据,并根据所述组件数据进行渲染,以得到所述待渲染组件标识对应的组件,并将所述待渲染组件标识对应的组件存储至所述预设位置。

2. 根据权利要求1所述的方法,其特征在于,所述根据所述页面信息确定所述待显示页面对应的待渲染组件标识,包括:

根据所述页面信息确定所述待显示页面对应的控制器,并通过所述控制器,确定所述待显示页面对应的待渲染组件标识。

3. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

在接收到页面访问请求时,获取所述页面访问请求中的目标网络地址,并获取当前页面的页面信息;

根据所述目标网络地址和当前页面的页面信息确定页面是否变化;

若页面变化,则根据所述目标网络地址确定所述待显示页面。

4. 根据权利要求3所述的方法,其特征在于,所述当前页面的页面信息包括当前页面的网络地址对应的哈希值;

所述根据所述目标网络地址和当前页面的页面信息确定页面是否变化,包括:

获取所述目标网络地址对应的哈希值;

判断所述目标网络地址对应的哈希值是否与所述当前页面的网络地址对应的哈希值相同;

若相同,则确定页面不变化;

若不相同,则确定页面变化。

5. 根据权利要求3所述的方法,其特征在于,所述方法还包括:

在接收到页面访问请求时,获取各节点对应的权重值;

根据各节点对应的权重值确定目标节点,并将所述页面访问请求发送至所述目标节点。

6. 根据权利要求1至5任一项所述的方法,其特征在于,所述预设位置包括缓存。

7. 一种组件处理设备,其特征在于,包括:

信息获取模块,用于获取待显示页面的页面信息,并根据所述页面信息确定所述待显示页面对应的待渲染组件标识;

处理模块,用于对于每个待渲染组件标识,判断预设位置中是否存在所述待渲染组件标识对应的组件;

所述处理模块,还用于在确定预设位置中不存在所述待渲染组件标识对应的组件时,获取所述待渲染组件标识对应的组件数据,并根据所述组件数据进行渲染,以得到所述待渲染组件标识对应的组件,并将所述待渲染组件标识对应的组件存储至所述预设位置。

8. 一种电子设备,其特征在于,包括:至少一个处理器和存储器;

所述存储器存储计算机执行指令；

所述至少一个处理器执行所述存储器存储的计算机执行指令，使得所述至少一个处理器执行如权利要求1至6任一项所述的组件处理方法。

9. 一种计算机可读存储介质，其特征在于，所述计算机可读存储介质中存储有计算机执行指令，当处理器执行所述计算机执行指令时，实现如权利要求1至6任一项所述的组件处理方法。

组件处理方法及设备

技术领域

[0001] 本发明实施例涉及计算机技术领域,尤其涉及一种组件处理方法及设备。

背景技术

[0002] 网页组件,又叫web(World Wide Web,万维网)组件,其包括各种常用的界面组件。在进行云平台系统页面开发时,经常需要利用到组件。目前,当用户访问云平台系统的一个页面时,需要加载该页面对应的所有组件,即需要对页面对应的组件数据进行渲染,以在页面上显示相应的组件。

[0003] 然而,在进行页面跳转时,即在由当前页面跳转至目标页面时,对目标页面对应的所有组件数据进行渲染,在从目标页面跳转回该当前页面时,需要重新加载该当前页面对应的所有组件,即需要重新对该当前页面对应的组件数据进行渲染,导致组件的重复加载,从而造成资源的浪费。

发明内容

[0004] 本发明实施例提供一种组件处理方法及设备,以解决现有技术中资源浪费的技术问题。

[0005] 第一方面,本发明实施例提供一种组件处理方法,包括:

[0006] 获取待显示页面的页面信息,并根据所述页面信息确定所述待显示页面对应的待渲染组件标识;

[0007] 对于每个待渲染组件标识,判断预设位置中是否存在所述待渲染组件标识对应的组件;

[0008] 在确定预设位置中不存在所述待渲染组件标识对应的组件时,获取所述待渲染组件标识对应的组件数据,并根据所述组件数据进行渲染,以得到所述待渲染组件标识对应的组件。

[0009] 在一种可能的设计中,所述根据所述页面信息确定所述待显示页面对应的待渲染组件标识,包括:

[0010] 根据所述页面信息确定所述待显示页面对应的控制器,并通过所述控制器,确定所述待显示页面对应的待渲染组件标识。

[0011] 在一种可能的设计中,所述方法还包括:

[0012] 在接收到页面访问请求时,获取所述页面访问请求中的目标网络地址,并获取当前页面的页面信息;

[0013] 根据所述目标网络地址和当前页面的页面信息确定页面是否变化;

[0014] 若页面变化,则根据所述目标网络地址确定所述待显示页面。

[0015] 在一种可能的设计中,所述当前页面的页面信息包括当前页面的网络地址对应的哈希值;

[0016] 所述根据所述目标网络地址和当前页面的页面信息确定页面是否变化,包括:

- [0017] 获取所述目标网络地址对应的哈希值；
- [0018] 判断所述目标网络地址对应的哈希值是否与所述当前页面的网络地址对应的哈希值相同；
- [0019] 若相同，则确定页面不变化；
- [0020] 若不相同，则确定页面变化。
- [0021] 在一种可能的设计中，所述方法还包括：
- [0022] 在接收到页面访问请求时，获取各节点对应的权重值；
- [0023] 根据各节点对应的权重值确定目标节点，并将所述页面访问请求发送至所述目标节点。
- [0024] 在一种可能的设计中，所述预设位置包括缓存。
- [0025] 第二方面，本发明实施例提供一种组件处理设备，包括：
- [0026] 信息获取模块，用于获取待显示页面的页面信息，并根据所述页面信息确定所述待显示页面对应的待渲染组件标识；
- [0027] 处理模块，用于对于每个待渲染组件标识，判断预设位置中是否存在所述待渲染组件标识对应的组件；
- [0028] 所述处理模块，还用于在确定预设位置中不存在所述待渲染组件标识对应的组件时，获取所述待渲染组件标识对应的组件数据，并根据所述组件数据进行渲染，以得到所述待渲染组件标识对应的组件。
- [0029] 在一种可能的设计中，所述信息获取模块还用于：
- [0030] 根据所述页面信息确定所述待显示页面对应的控制器，并通过所述控制器，确定所述待显示页面对应的待渲染组件标识。
- [0031] 在一种可能的设计中，所述处理模块还用于：
- [0032] 在接收到页面访问请求时，获取所述页面访问请求中的目标网络地址，并获取当前页面的页面信息；
- [0033] 根据所述目标网络地址和当前页面的页面信息确定页面是否变化；
- [0034] 若页面变化，则根据所述目标网络地址确定所述待显示页面。
- [0035] 在一种可能的设计中，所述当前页面的页面信息包括当前页面的网络地址对应的哈希值；
- [0036] 所述处理模块还用于：
- [0037] 获取所述目标网络地址对应的哈希值；
- [0038] 判断所述目标网络地址对应的哈希值是否与所述当前页面的网络地址对应的哈希值相同；
- [0039] 若相同，则确定页面不变化；
- [0040] 若不相同，则确定页面变化。
- [0041] 在一种可能的设计中，所述处理模块还用于：
- [0042] 在接收到页面访问请求时，获取各节点对应的权重值；
- [0043] 根据各节点对应的权重值确定目标节点，并将所述页面访问请求发送至所述目标节点。
- [0044] 在一种可能的设计中，所述预设位置包括缓存。

[0045] 第三方面,本发明实施例提供一种电子设备,包括:至少一个处理器和存储器;

[0046] 所述存储器存储计算机执行指令;

[0047] 所述至少一个处理器执行所述存储器存储的计算机执行指令,使得所述至少一个处理器执行如上第一方面以及第一方面各种可能的设计所述的组件处理方法。

[0048] 第四方面,本发明实施例提供一种计算机可读存储介质,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如上第一方面以及第一方面各种可能的设计所述的组件处理方法。

[0049] 第五方面,本发明实施例提供一种计算机程序产品,包括计算机程序,所述计算机程序被处理器执行时,实现如上第一方面以及第一方面各种可能的设计所述的组件处理方法。

[0050] 本发明提供一种组件处理方法及设备,通过在需要显示待显示页面时,确定该待显示页面对应的所有组件,即待渲染组件标识,并分别判断预设位置中是否存在各待渲染组件标识对应的组件,在确定预设位置中不存在待渲染标识对应的组件时,表明需要加载该组件,则获取该组件对应的组件数据,以供利用该组件对应的组件数据进行加载渲染,在确定预设位置中存在待渲染标识对应的组件时,表明无需重复加载该组件,即无需重新渲染该组件,可以直接利用该组件,即将该组件显示在待显示页面上,由于在显示待显示页面时,仅需对待显示页面对应的所有组件中的未在预设位置中组件进行渲染,无需对已存在的组件重复加载渲染,减少组件的重复加载,从而可以减少资源的浪费,且还可以缩短页面显示所需的时间,提高页面的访问速度,用户体验高。

附图说明

[0051] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0052] 图1为本发明实施例提供的云平台系统的结构示意图;

[0053] 图2为本发明实施例提供的组件处理方法的流程示意图一;

[0054] 图3为本发明实施例提供的组件处理方法的流程示意图二;

[0055] 图4为本发明实施例提供的组件处理设备的结构示意图;

[0056] 图5为本发明实施例提供的电子设备的硬件结构示意图。

具体实施方式

[0057] 为使本发明实施例的目的、技术方案和优点更加清楚,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0058] 现有技术中,云平台系统通过重新打开新窗口的形式来组织页面,当用户访问系统的一个页面时,需要加载该页面对应的所有组件,在进行页面跳转时,原本的资源都会被销毁,即该页面对应的组件实例会被销毁,因此,当用户跳转回访问过的页面时,需要重新

渲染相应的组件,出现组件需要重复加载的问题,损耗性能,造成资源的浪费。

[0059] 因此,针对上述问题,本发明的技术构思是提供一种基于Vue框架的云平台系统,其采用前后端分离的方式,并结合Node.js技术引入中间层概念,提升了首页加载速度。前端在进行渲染页面时,将渲染得到的组件,即组件实例存储至缓存中,以便于在重新显示该页面时,可以直接从缓存中获取该页面对应的组件,并将其引入至该页面,无需重新获取相应的组件数据以加载组件,避免组件的重复加载,从而可以避免性能的不必要损耗,减少资源的浪费,且提高页面渲染速度,缩短用户访问的时间,用户体验高。

[0060] 下面以具体地示例对本公开的技术方案以及本公开的技术方案如何解决上述技术问题进行详细说明。下面这几个具体的示例可以相互结合,对于相同或相似的概念或过程可能在某些示例中不再赘述。下面将结合附图,对本公开的示例进行描述。

[0061] 图1为本发明实施例提供的云平台系统的结构示意图,如图1所示,云平台系统100包括前端设备101和后台设备102,前端设备101在接收到用户输入的访问请求后,将该访问请求发送至后台设备102,以获取相应的组件数据,并根据组件数据进行渲染,以将组件渲染至页面上。

[0062] 其中,前端设备101可以是服务器、电脑等设备。前端设备包括由页面组件模块和路由管理模块组成的视图层。

[0063] 其中,后端设备102可以是服务器、电脑等设备。

[0064] 另外,云平台系统100还包括接口层,前端设备101和后台设备102通过接口层进行通信。

[0065] 图2为本发明实施例提供的组件处理方法的流程示意图一,本实施例的执行主体可以为电子设备,具体的,可以为图1中的前端设备。如图2所示,该方法包括:

[0066] S201、获取待显示页面的页面信息,并根据页面信息确定待显示页面对应的待渲染组件标识。

[0067] 在本实施例中,在需要显示某个页面时,将该页面作为待显示页面。获取待显示页面的页面信息,以供利用该页面信息获取待显示页面对应的所有待渲染组件的标识,即得到待渲染组件标识。

[0068] 其中,待显示页面对应的待渲染组件为待显示页面所需显示的组件,即待显示页面需加载的组件。

[0069] 可选的,待显示页面的页面信息包括待显示页面的标识,该标识包括以下中的一个或多个:待显示页面的网络地址、待显示页面的网络地址对应的哈希值和待显示页面的编号。

[0070] 其中,网络地址包括IP(Internet Protocol,网际互连协议)地址、url(Uniform Resource Locator,统一资源定位符)地址。

[0071] 进一步的,可选的,在确定待显示页面的网络地址对应的哈希值时,基于预设哈希函数,对待显示页面的网络地址进行哈希计算,得到待显示页面的网络地址对应的哈希值。

[0072] 可选的,在利用待显示页面的页面信息,即待显示页面的标识确定待显示页面对应的待渲染组件标识时,从预设页面组件标识映射表中查找该待显示页面的标识对应的组件标识,并将其确定为待渲染组件标识。

[0073] S202、对于每个待渲染组件标识,判断预设位置中是否存在待渲染组件标识对应

的组件。

[0074] 在本实施例中,在得到待显示页面对应的所有待渲染组件标识中的每个待渲染组件标识,判断预设位置中存在该待渲染组件标识对应的组件,即组件实例。

[0075] 在确定预设位置中存在该待渲染组件标识对应的组件时,表明无需重新利用该待渲染组件标识获取相应的组件数据,以供利用该组件数据重新渲染得到该组价,则可以在待显示页面上直接引入该待渲染组件标识对应的组件,即将该待渲染组件标识对应的组件直接部署在待显示页面上,无需重新渲染该组件,避免组件的重复渲染,实现组件的重复利用,从而避免资源的浪费。

[0076] 可选的,预设位置包括缓存。

[0077] S203、在确定预设位置中不存在待渲染组件标识对应的组件时,获取待渲染组件标识对应的组件数据,并根据组件数据进行渲染,以得到待渲染组件标识对应的组件。

[0078] 在本实施例中,对于每个待渲染组件标识,在确定预设位置不存在该待渲染组件标识对应的组件时,表明未存在该待渲染组件标识对应的已经渲染完成的组件,需要渲染加载该组件,则获取待渲染组件标识对应的组件数据,以供利用该组件数据在待显示页面上进行渲染,从而得到相应的组件,即组件实例,实现组件的加载。

[0079] 其中,组件数据包括类、函数、图像以及视频等内容。

[0080] 可选的,在获取待渲染组件标识对应的组件数据时,可以从待显示页面对应的页面文件中获取,也可以从其它存储位置中进行获取,还可以发送相应的请求至后台设备,以使后台设备将待显示页面对应的相关数据发送至电子设备,从而使电子设备从该相关数据中获取相应的组件数据。

[0081] 在本实施例中,在进行页面显示,即页面加载时,判断预设位置中是否存在页面需加载的组件,即确定是否存在页面对应的组件,若存在,则可以直接利用相应的组件,无需重新渲染,若不存在,则获取该组件对应的组件数据,以利用该组件数据渲染得到相应的组件,从而可以避免组件的重复渲染。

[0082] 从上述描述可知,在需要显示待显示页面时,确定该待显示页面对应的所有组件,即待渲染组件标识,并分别判断预设位置中是否存在各待渲染组件标识对应的组件,在确定预设位置中不存在待渲染标识对应的组件时,表明需要加载该组件,则获取该组件对应的组件数据,以供利用该组件对应的组件数据进行加载渲染,在确定预设位置中存在待渲染标识对应的组件时,表明无需重复加载该组件,即无需重新渲染该组件,可以直接利用该组件,即将该组件显示在待显示页面上,由于在显示待显示页面时,仅需对待显示页面对应的所有组件中的未在预设位置中组件进行渲染,无需对已存在的组件重复加载渲染,减少组件的重复加载,从而可以减少资源的浪费,且还可以缩短页面显示所需的时间,提高页面的访问速度,用户体验高。

[0083] 图3为本发明实施例提供的组件处理方法的流程示意图二,本实施例在图2实施例的基础上,在显示待显示页面时,根据页面是否发生变化确定是否需要进行页面渲染,下面将结合一个具体实施例对此过程进行描述。如图3所示,该方法包括:

[0084] S301、在接收到页面访问请求时,获取页面访问请求中的目标网络地址,并获取当前页面的页面信息。

[0085] 在本实施例中,在接收到用户输入的页面访问请求,则获取该页面访问请求中的

目标网络地址,该目标网络地址为所需访问页面的网络地址,并获取电子设备当前显示的页面,即当前页面的页面信息,以供利用当前页面的页面信息和目标网络地址确定是否需要跳转页面,即确定页面是否需要发生变化。

[0086] 可选的,当前页面的页面信息包括当前页面的网络地址和/或当前页面的网络地址对应的哈希值。

[0087] 可选的,当用户进行点击超链接或者切换Tab页等操作时,生成相应的页面访问请求。例如,当用户点击当前页面中的超链接时,剩下相应的页面访问请求,该超链接对应的网络地址便为目标网络地址。

[0088] S302、根据目标网络地址和当前页面的页面信息确定页面是否变化。

[0089] 在本实施例中,在得到所需访问页面的目标网络地址以及当前页面的页面信息后,利用该目标网络地址和当前页面的页面信息确定所需访问的页面是否为当前页面,即判断页面是否需要发生变化,以确定是否需要跳转至新的页面,从而确定是否需要进行页面加载。

[0090] 在本实施例中,在利用该目标网络地址和当前页面的页面信息确定页面是否发生变化时,可以通过以下两种方式进行确定。

[0091] 一种方式为,判断目标网络地址与当前页面的网络地址是否相同,若相同,表明目标网络地址对应的页面为当前页面,即无需进行页面跳转,则确定页面无需发生变化,即页面不变化。若不同,表明目标网络地址对应的页面不为当前页面,需要进行页面跳转,则确定页面发生变化。

[0092] 另一种方式为,获取目标网络地址对应的哈希值。判断目标网络地址对应的哈希值是否与当前页面的网络地址对应的哈希值相同。若相同,则确定页面不变化。若不相同,则确定页面变化。

[0093] 具体的,通过预设哈希函数,对目标网络地址进行哈希计算,以得到目标网络地址对应的哈希值。判断目标网络地址对应的哈希值与当前页面的网络地址对应的哈希值是否相同,若相同,表明目标网络地址对应的页面为当前页面,即无需进行页面跳转,则确定页面无需发生变化,即页面不变化。若不同,表明目标网络地址对应的页面不为当前页面,需要进行页面跳转,则确定页面发生变化。

[0094] 其中,计算目标网络地址对应的哈希值所利用的哈希函数与计算当前页面的网络地址对应的哈希值所利用的哈希函数相同。

[0095] S303、若页面变化,则根据目标网络地址确定待显示页面。

[0096] 在本实施例中,在确定需要发生变化时,表明需要进行页面跳转,即需要跳转到目标网络地址对应的页面,则将目标网络地址对应的页面作为待显示页面,以进行待显示页面的加载。

[0097] S304、获取待显示页面的页面信息,并根据页面信息确定待显示页面对应的待渲染组件标识。

[0098] 在本实施例中,在进行页面显示,即页面加载时,可以利用控制器进行加载,即获取与页面信息,即待显示页面的网络地址对应的控制器,并将其确定为待显示页面对应的控制器,以利用该控制器进行页面渲染操作,例如,根据页面信息确定待显示页面对应的待渲染组件标识,以供利用该待渲染组件标识确定相应的组件。

[0099] 其中,路由管理模块通过urlCreate类来建立网络地址(例如,URL地址)和控制器之间对应关系,主要使用regToPath和historyManager两个方法来进行路由管理,并保持网络地址和系统UI保持同步,通过使用historyManager函数来根据用户浏览的页面地址维护浏览器历史记录信息并更新history对象,通过regToPath函数,匹配URL地址对应的控制器,以通过控制器来实现UI的更新。

[0100] S305、对于每个待渲染组件标识,判断预设位置中是否存在待渲染组件标识对应的组件。

[0101] S306、在确定预设位置中不存在待渲染组件标识对应的组件时,获取待渲染组件标识对应的组件数据,并根据组件数据进行渲染,以得到待渲染组件标识对应的组件。

[0102] 在本实施例中,利用待渲染组件标识对应的组件数据进行渲染,以得到待渲染组件标识对应的组件,即组件实例,并将该待渲染组件标识对应的组件存储至预设位置,以便于后续可以从预设位置中直接利用该组件,而不是直接摧毁还组件实例,从而可以避免组件的重复渲染。

[0103] 另外,为了提高访问效率以及成功率,可以基于负载均衡规则,确定出目标节点,以利用目标节点进行页面加载,即在接收到页面访问请求时,获取各节点对应的权重值;根据各节点对应的权重值确定目标节点,并将所述页面访问请求发送至所述目标节点。

[0104] 在本实施例中,在接收到页面访问请求时,即在确定需要显示页面访问请求中的目标网络地址所对应的待显示页面时,选取权重值最高的服务器节点,即节点,并将其确定为目标节点,将该页面访问请求发送至该目标节点,以使目标节点将与该待显示页面对应的页面文件和/或组件数据发送至电子设备,从而使电子设备获取其所需要的数据,并利用该数据进行渲染。

[0105] 另外,在确定预设位置中不存在待渲染组件标识对应的组件时,将该待渲染组件标识确定为目标组件标识,获取所有目标组件标识,并生成包括该目标组件标识的数据请求,将该数据请求发送至目标节点,以使目标节点将该目标组件标识对应的组件数据发送至电子设备,从而使电子设备可以仅获取所需的数据。

[0106] 其中,节点为服务器,具体的,可以为Nginx服务器,其为后台设备。

[0107] 另外,在需要发送请求(例如,页面访问请求、数据请求)至一个节点时,可以将其发送至目标Nginx服务器,以使目标Nginx服务器获取各个节点的权重值,并根据各个节点的权重值确定出目标节点。

[0108] 其中,节点对应的权重值可以是预设不变的,也可以是根据节点的负载情况实时调整的,例如,在将请求分配至一个节点后,将该节点对应的权重值减去预设数值(例如,1)。

[0109] 在本实施例中,节点的权重决定节点被分配到请求的概率,权值配置可以根据节点的性能进行按需配置,高性能服务器节点的权重值大,低性能服务器节点的权重值小,这样能更合理利用服务器资源,更快的响应用户请求,避免低性能服务器因分配请求过多而导致性能下降。

[0110] 本实施例中,采用前后端分离架构,前端负责页面渲染,并通过与后端协商好的统一数据格式来进行数据处理和传递,对于前端可能出现的单机处理性能瓶颈,通过横向扩展增加实例节点,并对外统一暴露一个负载均衡Nginx服务器,通过负载均衡策略保证系统

面对高并发情况下仍能最大程度的保证系统的可用性,最终能够提供给用户一个快速、稳定的使用体验。

[0111] 在本实施例中,云平台系统包括由页面组件模块和路由管理模块组成的视图层、对外提供统一接口服务的接口层以及响应客户端请求,并将结果交付给接口层进一步处理的服务层,视图层组件采用模块化设计,通过路由管理模块实时监测当前页面url变化信息,选择对应组件渲染至页面,接口层通过Ajax与Node.js进行通信,传递用户请求至所述服务层,服务层配合Nginx负载均衡策略获取相应服务器数据。

[0112] 在本实施例中,视图层按照业务逻辑进行划分,将同一类需求或者功能模块划分成为同一组件,保证了业务逻辑的高度统一,对于后续维护提供了便利,当需求发生变化时,更容易针对性的去解决和重构。

[0113] 在本实施例中,接口层接收前端请求,并根据页面访问请求中的目标网络地址,通过regToPath函数获取最终匹配的组件,通过预获取方法向后台服务器请求所需要的数据,并将该数据写回到HTML(Hyper Text Markup Language,超文本标记语言)文档,最后将拥有完整数据的HTML文档返回给浏览器,由浏览器进行渲染。

[0114] S307、若页面不变化,则无需获取待显示页面的页面信息。

[0115] 在本实施例中,在确定页面不变化,即在确定页面无需发生变化时,表明无需进行页面跳转,即用户所需访问的页面为当前页面,则无需获取待显示页面的页面信息,即无需页面加载。

[0116] 在本实施例中,检测页面的网络地址的哈希值是否发生变化,在捕获到页面的网络地址的哈希值发生变化时,选择相应的组件渲染至页面,以避免进行不必要的页面渲染,减少资源的浪费。

[0117] 在本实施例中,路由管理模块监测页面网络地址是否发生变化,并在发生变化时,选择对应的组件渲染至页面上。

[0118] 在本实施例中,UtillsAjax方法内置url和options参数配置后台设备对应接口地址以及当前请求的配置信息,并返回一个Promiss对象,从而屏蔽同步和异步代码之间的一致性,能够进一步减少后续开发和维护的压力。利用UtillsAjax向服务层,即后台设备发送数据请求,并通过调用NativeAPP的方式注入API和拦截url-Scheme。该拦截url-Scheme主要利用Web端通过iframe.src发送url-Scheme请求,根据Native拦截到相应请求并根据url-Scheme自身所带参数信息进行拦截操作。

[0119] 在本实施例中,当后台设备无法正常工作,目标Nginx服务器会将所有机器,即节点的标识位进行重置,避免出现由于后台设备均处于timeout情况,导致前端一直处于未响应状态的问题。

[0120] 在本实施例中,针对单台Nginx服务器在实际运行过程中可能会出现宕机或者停止服务等异常情况的问题,本申请结合keepalive以及Nginx配置出高可用集群,保证负载均衡服务器节点能够应对异常情况,持续提供正常服务。

[0121] 图4为本发明实施例提供的组件处理设备的结构示意图,如图4所示,该组件处理设备400包括:信息获取模块401和处理模块402。

[0122] 其中,信息获取模块401,用于获取待显示页面的页面信息,并根据页面信息确定待显示页面对应的待渲染组件标识。

[0123] 处理模块402,用于对于每个待渲染组件标识,判断预设位置中是否存在待渲染组件标识对应的组件。

[0124] 处理模块402,还用于在确定预设位置中不存在待渲染组件标识对应的组件时,获取待渲染组件标识对应的组件数据,并根据组件数据进行渲染,以得到待渲染组件标识对应的组件。

[0125] 在任意实施例中,可选的,预设位置包括缓存。

[0126] 在本发明的另一个实施例中,在上述图4所示的实施例的基础上,处理模块402还用于:

[0127] 在接收到页面访问请求时,获取页面访问请求中的目标网络地址,并获取当前页面的页面信息。

[0128] 根据目标网络地址和当前页面的页面信息确定页面是否变化。

[0129] 若页面变化,则根据目标网络地址确定待显示页面。

[0130] 在本实施例中,可选的,当前页面的页面信息包括当前页面的网络地址对应的哈希值。

[0131] 处理模块402还用于:

[0132] 获取目标网络地址对应的哈希值。

[0133] 判断目标网络地址对应的哈希值是否与当前页面的网络地址对应的哈希值相同。

[0134] 若相同,则确定页面不变化。

[0135] 若不相同,则确定页面变化。

[0136] 在任意实施例中,可选的,处理模块402还用于:

[0137] 在接收到页面访问请求时,获取各节点对应的权重值。

[0138] 根据各节点对应的权重值确定目标节点,并将页面访问请求发送至目标节点。

[0139] 在任意实施例中,可选的,信息获取模块401还用于:

[0140] 根据页面信息确定待显示页面对应的控制器,并通过控制器,确定待显示页面对应的待渲染组件标识。

[0141] 图5为本发明实施例提供的电子设备的硬件结构示意图。如图5所示,本实施例的电子设备500包括:处理器501以及存储器502;

[0142] 其中,存储器502,用于存储计算机执行指令;

[0143] 处理器501,用于执行存储器存储的计算机执行指令,以实现上述实施例中接收设备所执行的各个步骤。具体可以参见前述方法实施例中的相关描述。

[0144] 可选地,存储器502既可以是独立的,也可以跟处理器501集成在一起。

[0145] 当存储器502独立设置时,该电子设备还包括总线503,用于连接所述存储器502和处理器501。

[0146] 本发明实施例还提供一种计算机可读存储介质,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如上所述的组件处理方法。

[0147] 本发明实施例还提供一种计算机程序产品,包括计算机程序,所述计算机程序被处理器执行时,实现如上所述的组件处理方法。

[0148] 在本发明所提供的几个实施例中,应该理解到,所揭露的设备和方法,可以通过其它的方式实现。例如,以上所描述的设备实施例仅仅是示意性的,例如,所述模块的划分,仅

仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个模块可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或模块的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0149] 所述作为分离部件说明的模块可以是或者也可以不是物理上分开的,作为模块显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。

[0150] 另外,在本发明各个实施例中的各功能模块可以集成在一个处理单元中,也可以是各个模块单独物理存在,也可以两个或两个以上模块集成在一个单元中。上述模块成的单元既可以采用硬件的形式实现,也可以采用硬件加软件功能单元的形式实现。

[0151] 上述以软件功能模块的形式实现的集成的模块,可以存储在一个计算机可读取存储介质中。上述软件功能模块存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)或处理器(英文:processor)执行本申请各个实施例所述方法的部分步骤。

[0152] 应理解,上述处理器可以是中央处理单元(英文:Central Processing Unit,简称:CPU),还可以是其他通用处理器、数字信号处理器(英文:Digital Signal Processor,简称:DSP)、专用集成电路(英文:Application Specific Integrated Circuit,简称:ASIC)等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合发明所公开的方法的步骤可以直接体现为硬件处理器执行完成,或者用处理器中的硬件及软件模块组合执行完成。

[0153] 存储器可能包含高速RAM存储器,也可能还包括非易失性存储NVM,例如至少一个磁盘存储器,还可以为U盘、移动硬盘、只读存储器、磁盘或光盘等。

[0154] 总线可以是工业标准体系结构(Industry Standard Architecture,ISA)总线、外部设备互连(Peripheral Component,PCI)总线或扩展工业标准体系结构(Extended Industry Standard Architecture,EISA)总线等。总线可以分为地址总线、数据总线、控制总线等。为便于表示,本申请附图中的总线并不限定仅有一根总线或一种类型的总线。

[0155] 上述存储介质可以由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器(SRAM),电可擦除可编程只读存储器(EEPROM),可擦除可编程只读存储器(EPROM),可编程只读存储器(PROM),只读存储器(ROM),磁存储器,快闪存储器,磁盘或光盘。存储介质可以是通用或专用计算机能够存取的任何可用介质。

[0156] 一种示例性的存储介质耦合至处理器,从而使处理器能够从该存储介质读取信息,且可向该存储介质写入信息。当然,存储介质也可以是处理器的组成部分。处理器和存储介质可以位于专用集成电路(Application Specific Integrated Circuits,简称:ASIC)中。当然,处理器和存储介质也可以作为分立组件存在于电子设备或主控设备中。

[0157] 本领域普通技术人员可以理解:实现上述各方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成。前述的程序可以存储于一计算机可读取存储介质中。该程序在执行时,执行包括上述各方法实施例的步骤;而前述的存储介质包括:ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0158] 最后应说明的是：以上各实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述各实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分或者全部技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的范围。

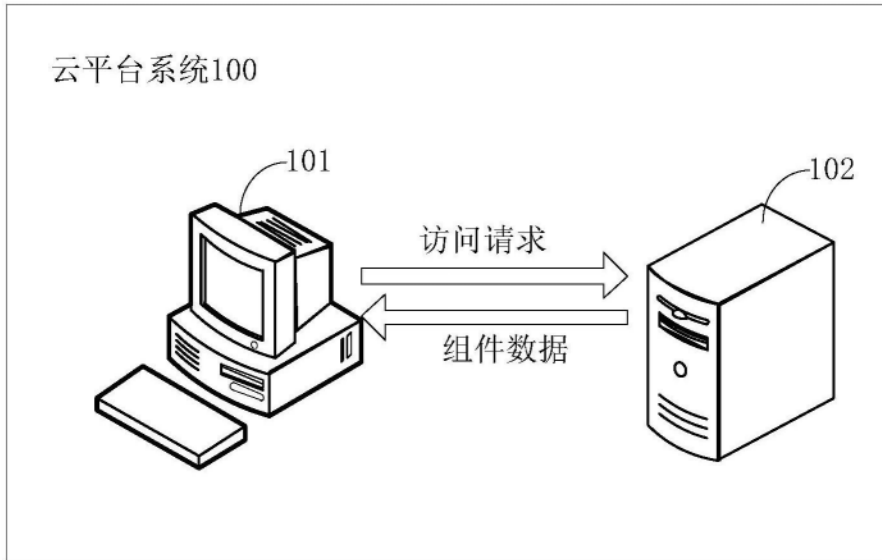


图1

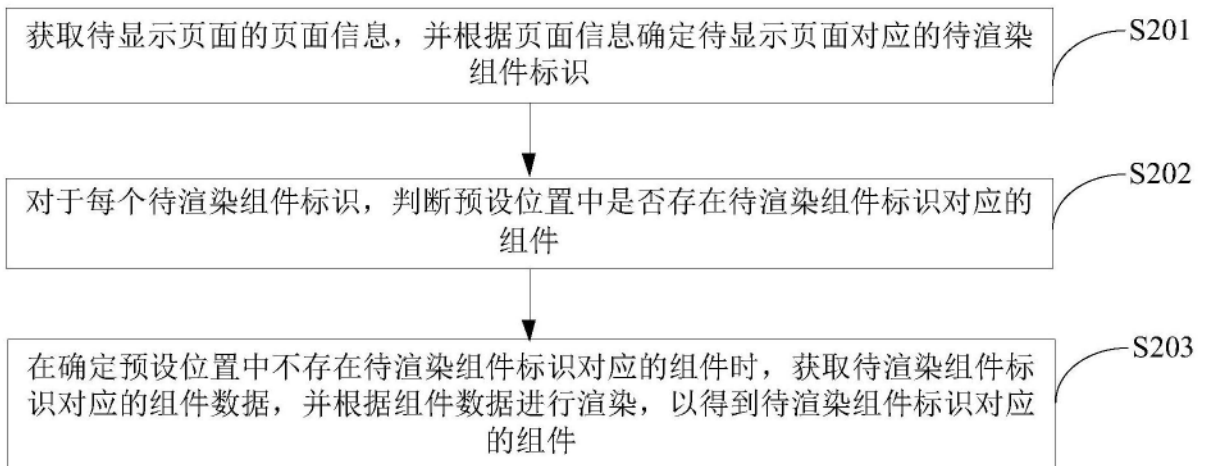


图2

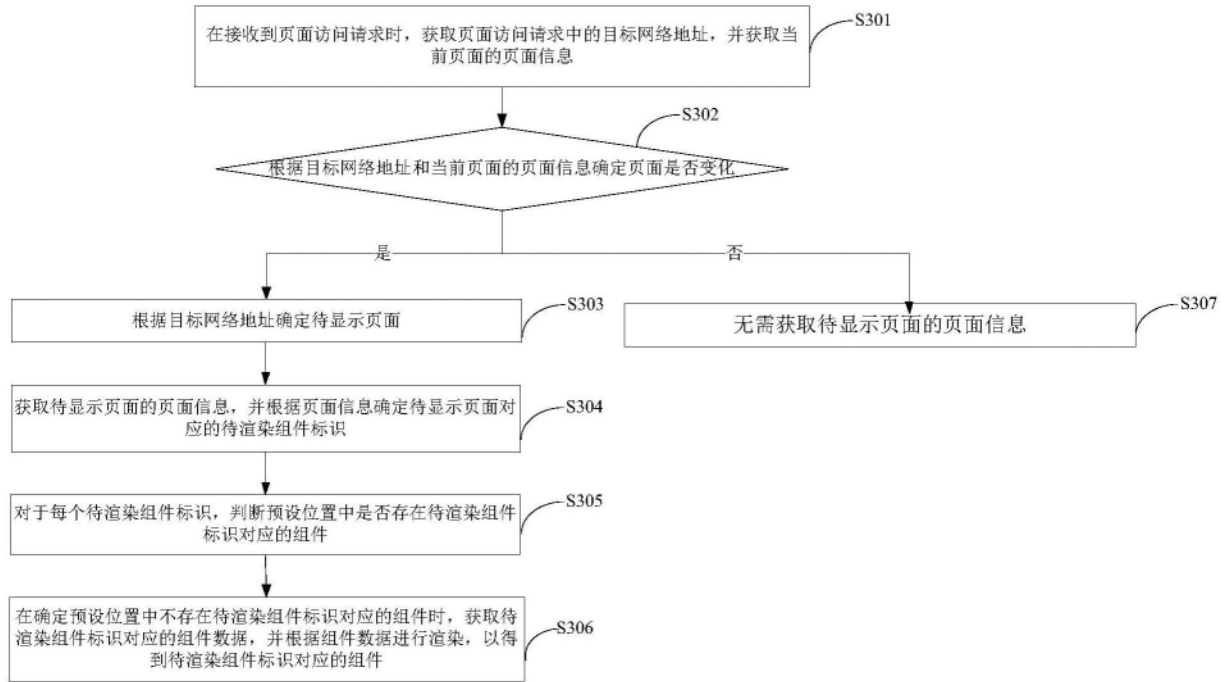


图3

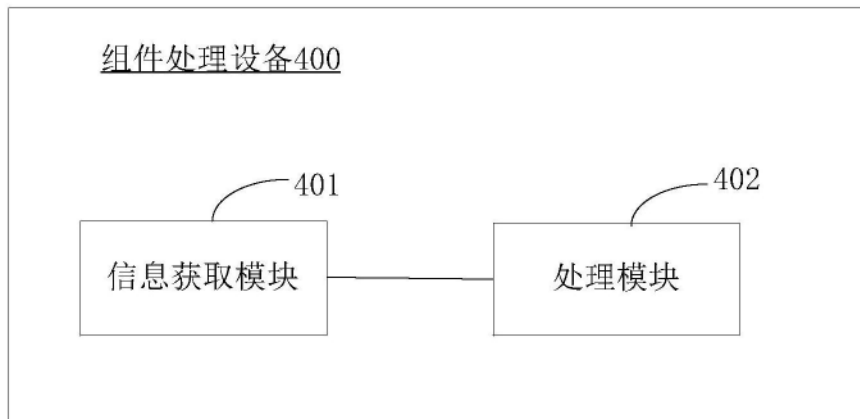


图4

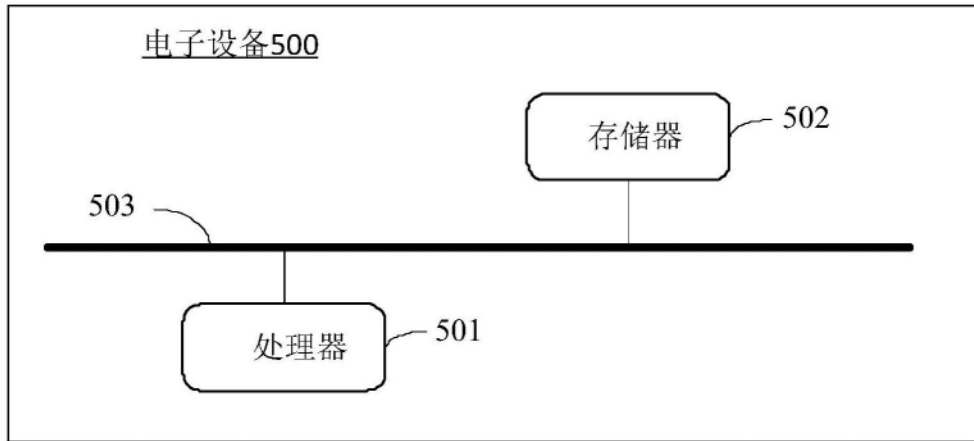


图5