(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 110442811 A
(43)申请公布日 2019.11.12

(21)申请号 201910750332.4

(22)申请日 2019.08.14

(71)申请人 广州虎牙科技有限公司
地址 511400 广东省广州市番禺区钟村街(汉溪商业中心)泽溪街13号1301

(72)发明人 纳勇林

(74)专利代理机构 北京品源专利代理有限公司 11332
代理人 孟金喆

(51)Int.Cl.
G06F 16/957(2019.01)

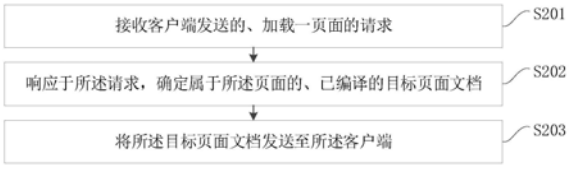
权利要求书2页 说明书17页 附图6页

(54)发明名称

一种页面的处理方法、装置、计算机设备和存储介质

(57)摘要

本发明实施例公开了一种页面的处理方法、装置、计算机设备和存储介质。该方法包括:接收客户端发送的、加载一页面的请求;响应于所述请求,确定属于所述页面的、已编译的目标页面文档,所述目标页面文档已渲染位于所述页面首屏区域中的动态资源;将所述目标页面文档发送至所述客户端,所述客户端用于从资源服务器获取属于所述页面的、已编译的静态资源,将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。在服务端中拉取并渲染页面首屏的动态资源,可提高页面首屏的渲染速度、减少页面首屏的渲染时间,从而提高在客户端显示页面首屏的速度,由于首屏为用户所见,可给用户带来页面显示快的效果。



CN 110442811 A

1.一种页面的处理方法,其特征在于,包括:

接收客户端发送的、加载一页面的请求;

响应于所述请求,确定属于所述页面的、已编译的目标页面文档,所述目标页面文档已渲染有位于所述页面首屏区域中的动态资源;

将所述目标页面文档发送至所述客户端,所述客户端用于从资源服务器获取属于所述页面的、已编译的静态资源,将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

2.根据权利要求1所述的方法,其特征在于,所述确定属于所述页面的、已编译的目标页面文档,包括:

查找属于所述页面的、已编译的原始页面文档;

获取位于所述页面首屏区域的动态资源;

将所述动态资源渲染至所述原始页面文档中,获得目标页面文档。

3.根据权利要求2所述的方法,其特征在于,所述获取位于所述页面非首屏区域的动态资源,包括:

查找属于所述页面的、已编译的首屏脚本;

运行所述首屏脚本,所述首屏脚本用于获取位于所述页面首屏区域中的动态资源。

4.根据权利要求2所述的方法,其特征在于,所述确定属于所述页面的、已编译的目标页面文档,还包括:

将所述目标页面文档存储至缓存中;

对所述目标页面文档设置有效时间,所述有效时间用于在超时时,清除所述目标页面文档。

5.根据权利要求1-4任一项所述的方法,其特征在于,所述确定属于所述页面的、已编译的目标页面文档,包括:

在缓存中查找属于所述页面的、已编译的目标页面文档;

若查找到,则从所述缓存中提取所述目标页面文档。

6.一种页面的处理方法,其特征在于,包括:

接收显示页面的操作;

响应于所述操作,向页面服务器请求确定属于所述页面的、已编译的目标页面文档,所述目标页面文档已渲染位于所述页面首屏区域中的动态资源;

从资源服务器获取属于所述页面的、已编译的静态资源;

将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

7.根据权利要求6所述的方法,其特征在于,还包括:

获取位于所述页面非首屏区域中的动态资源;

将所述动态资源渲染至所述目标页面文档、以加载所述页面的非首屏区域。

8.根据权利要求7所述的方法,其特征在于,所述获取位于所述页面非首屏区域中的动态资源,包括:

从资源服务器获取属于所述页面的非首屏脚本;

运行所述非首屏脚本,所述脚本用于获取位于所述页面非首屏区域中的动态资源。

9.一种页面的处理方法,其特征在于,包括:

获取属于页面的源文件；

将所述源文件编译为目标文件，所述目标文件包括原始页面文档、首屏脚本、非首屏脚本与静态资源，所述首屏脚本用于获取位于所述页面首屏区域中的动态资源、所述非首屏脚本用于获取位于所述页面非首屏区域中的动态资源；

将所述原始页面文档与所述首屏脚本分发至页面服务器；

将所述非首屏脚本与所述静态资源分发至资源服务器。

10.根据权利要求9所述的方法，其特征在于，在所述将所述源文件编译为目标文件之后，还包括：

对所述目标文件进行压缩。

11.一种页面的处理装置，其特征在于，包括：

请求接收模块，用于接收客户端发送的、加载一页面的请求；

目标页面文档确定模块，用于响应于所述请求，确定属于所述页面的、已编译的目标页面文档，所述目标页面文档已渲染位于所述页面首屏区域中的动态资源；

目标页面文档发模块，用于将所述目标页面文档发送至所述客户端，所述客户端用于从资源服务器获取属于所述页面的、已编译的静态资源，将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

12.一种页面的处理装置，其特征在于，包括：

操作接收模块，用于接收显示页面的操作；

目标页面文档请求模块，用于响应于所述操作，向页面服务器请求确定属于所述页面的、已编译的目标页面文档，所述目标页面文档已渲染有位于所述页面首屏区域中的动态资源；

静态资源获取模块，用于从资源服务器获取属于所述页面的、已编译的静态资源；

静态资源渲染模块，用于将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

13.一种页面的处理装置，其特征在于，包括：

源文件获取模块，用于获取属于页面的源文件；

文件编译模块，用于将所述源文件编译为目标文件，所述目标文件包括原始页面文档、首屏脚本、非首屏脚本与静态资源，所述首屏脚本用于获取位于所述页面首屏区域中的动态资源、所述非首屏脚本用于获取位于所述页面非首屏区域中的动态资源；

第一资源分发模块，用于将所述原始页面文档与所述首屏脚本分发至页面服务器；

第二资源分发模块，用于将所述非首屏脚本与所述静态资源分发至资源服务器。

14.一种计算机设备，其特征在于，所述计算机设备包括：

一个或多个处理器；

存储器，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，使得所述一个或多个处理器实现如权利要求1-10中任一所述的页面的处理方法。

15.一种计算机可读存储介质，其上存储有计算机程序，其特征在于，该计算机程序被处理器执行时实现如权利要求1-10中任一所述的页面的处理方法。

一种页面的处理方法、装置、计算机设备和存储介质

技术领域

[0001] 本发明实施例涉及页面的技术，尤其涉及一种页面的处理方法、装置、计算机设备和存储介质。

背景技术

[0002] 随着WEB、H5等页面技术的发展，基于页面的应用越来越多，应用的功能也日趋增多。

[0003] 目前，客户端向服务器请求页面之后，接收服务器返回的HMTL(HyperText Markup Language，超文本标记语言)文档，该HMTL文档为页面的框架，没有内容，需要调用接口拉取数据，再将拉取的数据填充到HMTL文档中。

[0004] 在拉取数据期间，页面只有框架而没有具体的内容，呈现刷新状态，而拉取数据的时间较长，导致用户等待页面的时间较长。

发明内容

[0005] 本发明实施例提供一种页面的处理方法、装置、计算机设备和存储介质，以解决拉取页面中的数据的时间较长，导致用户等待页面的时间较长的问题。

[0006] 第一方面，本发明实施例提供了一种页面的处理方法，包括：

[0007] 接收客户端发送的、加载一页面的请求；

[0008] 响应于所述请求，确定属于所述页面的、已编译的目标页面文档，所述目标页面文档已渲染有位于所述页面首屏区域中的动态资源；

[0009] 将所述目标页面文档发送至所述客户端，所述客户端用于从资源服务器获取属于所述页面的、已编译的静态资源，将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

[0010] 第二方面，本发明实施例还提供了一种页面的处理方法，包括：

[0011] 接收显示页面的操作；

[0012] 响应于所述操作，向页面服务器请求确定属于所述页面的、已编译的目标页面文档，所述目标页面文档已渲染位于所述页面首屏区域中的动态资源；

[0013] 从资源服务器获取属于所述页面的、已编译的静态资源；

[0014] 将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

[0015] 第三方面，本发明实施例还提供了一种页面的处理方法，包括：

[0016] 获取属于页面的源文件；

[0017] 将所述源文件编译为目标文件，所述目标文件包括原始页面文档、首屏脚本、非首屏脚本与静态资源，所述首屏脚本用于获取位于所述页面首屏区域中的动态资源、所述非首屏脚本用于获取位于所述页面非首屏区域中的动态资源；

[0018] 将所述原始页面文档与所述首屏脚本分发至页面服务器；

[0019] 将所述非首屏脚本与所述静态资源分发至资源服务器。

[0020] 第四方面，本发明实施例还提供了一种页面的处理装置，包括：

[0021] 请求接收模块，用于接收客户端发送的、加载一页面的请求；

[0022] 目标页面文档确定模块，用于响应于所述请求，确定属于所述页面的、已编译的目标页面文档，所述目标页面文档已渲染有位于所述页面首屏区域中的动态资源；

[0023] 目标页面文档发模块，用于将所述目标页面文档发送至所述客户端，所述客户端用于从资源服务器获取属于所述页面的、已编译的静态资源，将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

[0024] 第五方面，本发明实施例还提供了一种页面的处理装置，包括：

[0025] 操作接收模块，用于接收显示页面的操作；

[0026] 目标页面文档请求模块，用于响应于所述操作，向页面服务器请求确定属于所述页面的、已编译的目标页面文档，所述目标页面文档已渲染位于所述页面首屏区域中的动态资源；

[0027] 静态资源获取模块，用于从资源服务器获取属于所述页面的、已编译的静态资源；

[0028] 静态资源渲染模块，用于将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

[0029] 第六方面，本发明实施例还提供了一种页面的处理装置，包括：

[0030] 源文件获取模块，用于获取属于页面的源文件；

[0031] 文件编译模块，用于将所述源文件编译为目标文件，所述目标文件包括原始页面文档、首屏脚本、非首屏脚本与静态资源，所述首屏脚本用于获取位于所述页面首屏区域中的动态资源、所述非首屏脚本用于获取位于所述页面非首屏区域中的动态资源；

[0032] 第一资源分发模块，用于将所述原始页面文档与所述首屏脚本分发至页面服务器；

[0033] 第二资源分发模块，用于将所述非首屏脚本与所述静态资源分发至资源服务器。

[0034] 第七方面，本发明实施例还提供了一种计算机设备，所述计算机设备包括：

[0035] 一个或多个处理器；

[0036] 存储器，用于存储一个或多个程序；

[0037] 当所述一个或多个程序被所述一个或多个处理器执行，使得所述一个或多个处理器实现如第一方面、第二方面、第三方面中任一所述的页面的处理方法。

[0038] 第八方面，本发明实施例还提供了一种计算机可读存储介质，其上存储有计算机程序，该计算机程序被处理器执行时实现如第一方面、第二方面、第三方面中任一所述的页面的处理方法。

[0039] 在本实施例中，接收客户端发送的、加载一页面的请求，响应于该请求，确定属于页面的、已编译的目标页面文档，目标页面文档已渲染位于页面首屏区域中的动态资源，将目标页面文档发送至客户端，客户端用于从资源服务器获取属于所述页面的、已编译的静态资源，将静态资源渲染至目标页面文档、以显示页面的首屏区域。对于页面渲染的过程而言，拉取并渲染动态资源占用的设备资源较高，耗时较多，而服务端所在的设备的性能一般比客户端所在的设备要强、网络环境一般比客户端所在的设备要好，因此，在服务端中拉取并渲染页面首屏的动态资源，可提高页面首屏的渲染速度、减少页面首屏的渲染时间，从而提高在客户端显示页面首屏的速度，由于首屏为用户所见，可给用户带来页面显示快的效

果。

附图说明

[0040] 图1为本发明实施例一提供的一种页面的处理方法的流程图；

[0041] 图2是本发明实施例二提供的一种页面的处理方法的流程图；

[0042] 图3是本发明实施例三提供的一种页面的处理方法的流程图；

[0043] 图4是本发明实施例四提供的一种页面的处理方法的流程图；

[0044] 图5A至图5C是本发明实施例提供的一种页面的处理的示例图；

[0045] 图6为本发明实施例五提供的一种页面的处理装置的结构示意图；

[0046] 图7为本发明实施例六提供的一种页面的处理装置的结构示意图；

[0047] 图8为本发明实施例七提供的一种页面的处理装置的结构示意图；

[0048] 图9为本发明实施例八提供的一种计算机设备的结构示意图。

具体实施方式

[0049] 下面结合附图和实施例对本发明作进一步的详细说明。可以理解的是，此处所描述的具体实施例仅仅用于解释本发明，而非对本发明的限定。另外还需要说明的是，为了便于描述，附图中仅示出了与本发明相关的部分而非全部结构。

[0050] 实施例一

[0051] 图1为本发明实施例四提供的一种页面的处理方法的流程图，本实施例提供一个框架，用于开发项目(即页面)，该项目整体进行编译并分发至页面服务器、资源服务器。

[0052] 对于一个客户端而言，由于功能较多，其首页可按照原生的框架进行开发，如MWC(Model View Controller，模型-视图-控制器)，而二级或二级以下的页面，功能相对简单，可按照本实施例所提供的框架进行开发。

[0053] 本实施例可适用于将该页面的资源分发至页面服务器、资源服务器的情况，该方法可以由页面的处理装置来执行，该页面的处理装置可以由软件和/或硬件实现，可配置在计算机设备中，如SVN(Subversion，版本控制系统)等，该方法具体包括如下步骤：

[0054] S101、获取属于页面的源文件。

[0055] 页面的源文件，可以指使用源语言编写该页面所形成的源程序，该源文件包括未编译的原始页面文档、未编译的首屏脚本、未编译的非首屏脚本与未编译的静态资源。

[0056] S102、将所述源文件编译为目标文件。

[0057] 在具体实现中，页面的源文件用户编写完成之后，则可以调用编译器，将项目整体(源文件)编译为目标程序，作为目标文件，在编译之后，该目标文件可直接运行。

[0058] 以Android系统为例，编译环境为Ubuntu，源文件编译的流程如下：

[0059] 1、源码下载

[0060] 下载及安装repo工具，repo工具由一系列python脚本组成，通过调用Git命令实现对AOSP项目的管理。

[0061] 为项目在本地创建对应的仓库，同样，这里为了方便对代码进行管理，为其创建一个文件夹，后面所有的下载的源文件和编译出的目标文件也都放在这个文件夹中。

[0062] 将该文件夹作为仓库，通过执行初始化仓库命令获取AOSP项目master上最新的代

码并初始化该仓库。

[0063] 初始化仓库之后,就可以开始正式同步源文件的代码到本地了。

[0064] 2、构建编译环境

[0065] 通过source build/envsetup.sh等命令初始化编译环境。

[0066] 3、编译源码

[0067] 初始化编译环境之后,则进入源码编译阶段,这个阶段又包括两个阶段:选择编译目标和执行编译。

[0068] 通过lunch指令设置编译目标,所谓的编译目标就是生成的镜像要运行在什么样的设备上。

[0069] 通过make指令进行代码编译,该指令通过-j参数来设置参与编译的线程数量,以提高编译速度。

[0070] 4、运行模拟器

[0071] 在编译完成之后,就可以通过以下命令运行Android虚拟机了。

[0072] 当然,上述编译方法只是作为示例,在实施本实施例时,可以根据实际情况(如操作系统的类型、版本)设置其他编译方法,本发明实施例对此不加以限制。另外,除了上述编译方法外,本领域技术人员还可以根据实际需要采用其它编译方法,本发明实施例对此也不加以限制。

[0073] 在本实施例中,目标文件包括已编译的原始页面文档、已编译的首屏脚本、已编译的非首屏脚本与已编译的静态资源。

[0074] 其中,原始页面文档是页面的主体,用于传递页面的各类信息,如HTML文档。

[0075] 首屏脚本,属于描述页面行为的脚本,如JavaScript文件,在本实施例中,首屏脚本可用于获取位于页面首屏区域中的动态资源,以便将该动态资源渲染至原始页面文档中,作为目标文档,提高该页面首屏区域在客户端的显示速度。

[0076] 非首屏脚本,属于描述页面行为的脚本,如JavaScript文件,在本实施例中,非首屏脚本用于获取位于页面非首屏区域中的动态资源,以便将该动态资源渲染至目标页面文档中,加载该页面的非首屏区域。

[0077] 静态资源,可以指在一段时间内稳定的资源,如图片、CSS(Cascading Style Sheets,层叠样式表单)文件等,该静态资源用于渲染至目标页面文档。

[0078] 在将源文件编译为目标文件之后,可采用Uglifyjs等方式,对目标文件进行压缩,从而减小文件的大小,减少网络开销。

[0079] S103、将所述原始页面文档与所述首屏脚本分发至页面服务器。

[0080] 在本实施例中,编译之后的原始页面文档与首屏脚本可分发至页面服务器。

[0081] 页面服务器,可以指为客户端提供页面访问的服务器,即页面服务器面向客户端,如Node服务器、Vue服务器、Jquery服务器等。

[0082] 以Node服务器为例,Node.js应用如下几部分:

[0083] 1、引入required模块:使用require指令来载入Node.js模块。

[0084] 2、创建服务器:使用http.createServer()方法创建服务器,服务器可以通过request参数监听客户端的请求。

[0085] 3、接收请求与响应请求:客户端可以发送请求,服务器接收请求后通过response

参数返回响应数据。

[0086] 相应地,配合Node服务器,页面可以指由React开发的单页面或多页面。

[0087] Reactk以声明式编写UI(User Interface,用户界面),可创建交互式UI,为应用的每一个状态设计视图,当数据改变时,React能有效地更新并正确地渲染组件。

[0088] 对于简单组件,React组件使用一个名为render()的方法,接收输入的数据并返回需要展示的内容,这种写法被称为JSX。被传入的数据可在组件中通过this.props在render()访问。

[0089] 对于有状态组件,除了使用外部数据(通过this.props访问)以外,组件还可以维护其内部的状态数据(通过this.state访问)。当组件的状态数据改变时,组件会再次调用render()方法重新渲染对应的标记。

[0090] 对于应用,使用props和state,可以创建一个简易的Todo应用。

[0091] 在存储页面的原始目标文档、首屏脚本之后,对于页面服务器而言,可提供该页面的访问服务。

[0092] 在具体实现中,页面服务器可接收客户端发送的、加载一页面的请求,响应于该请求,确定属于页面的、已编译的目标页面文档,以及,将目标页面文档发送至客户端,其中,目标页面文档已渲染位于该页面首屏区域中的动态资源。

[0093] 进一步地,确定属于页面的、已编译的目标页面文档,包括:

[0094] 查找属于页面的、已编译的原始页面文档;

[0095] 获取位于页面首屏区域的动态资源;

[0096] 将动态资源渲染至原始页面文档中,获得目标页面文档。

[0097] 进一步地,获取位于页面非首屏区域的动态资源,包括:

[0098] 查找属于页面的、已编译的首屏脚本;

[0099] 运行首屏脚本,首屏脚本用于获取位于页面首屏区域中的动态资源。

[0100] 进一步地,确定属于页面的、已编译的目标页面文档,还包括:

[0101] 将目标页面文档存储至缓存中;

[0102] 对目标页面文档设置有效时间,有效时间用于在超时时,清除目标页面文档。

[0103] 进一步地,确定属于所述页面的、已编译的目标页面文档,包括:

[0104] 在缓存中查找属于页面的、已编译的目标页面文档;

[0105] 若查找到,则从缓存中提取所述目标页面文档。

[0106] 在本实施例中,由于客户端、页面服务器的操作与实施例二、三、四的应用基本相似,所以描述的比较简单,相关之处参见实施例二、三、四的部分说明即可,本实施例在此不加以详述。

[0107] 在本实施例中,页面服务器存储有原始目标文档与首屏脚本,首屏脚本可用于获取位于页面首屏区域中的动态资源,以便将该动态资源渲染至原始页面文档中,作为目标文档,对于页面渲染的过程而言,拉取并渲染动态资源占用的设备资源较高,耗时较多,而页面服务器的性能一般比客户端所在的设备要强、网络环境一般比客户端所在的设备要好,因此,在页面服务器中拉取并渲染页面首屏的动态资源,可提高页面首屏的渲染速度、减少页面首屏的渲染时间,从而提高在客户端显示页面首屏的速度,由于首屏为用户所见,可给用户带来页面显示快的效果。

[0108] S104、将所述非首屏脚本与所述静态资源分发至资源服务器。

[0109] 在本实施例中，编译之后的非首屏脚本与静态资源可分发至资源服务器。

[0110] 资源服务器与页面服务器相互独立，用于存储、分发两个或两个以上页面中的非首屏脚本、静态资源，如CDN(Content Delivery Network，内容分发网络)、Redis服务器等。

[0111] 以CDN为例，通过在网络各处放置节点服务器所构成的在现有的网络基础之上的一层智能虚拟网络，CDN可以实时地根据网络流量和各节点的连接、负载状况以及到客户端的距离和响应时间等综合信息将客户端的请求重新导向离客户端最近的服务节点上，使得客户端户可就近取得所需内容(如非首屏脚本、页面的静态资源)，解决网络拥挤的状况，提高访问的响应速度，从而减少获取内容的时间。

[0112] 在存储页面的非首屏脚本、静态资源之后，对于资源服务器而言，可提供该页面的访问服务。

[0113] 在具体实现中，页面服务器可接收客户端发送的、加载一页面的请求，响应于该请求，确定属于页面的、已编译的非首屏脚本与已编译的静态资源，以及，将非首屏脚本与静态资源发送至客户端。

[0114] 对于客户端而言，将静态资源渲染至所述目标页面文档、以显示页面的首屏区域，以及，将动态资源渲染至所述目标页面文档、以加载页面的非首屏区域。

[0115] 进一步地，获取位于页面非首屏区域中的动态资源，包括：

[0116] 从资源服务器获取属于页面的非首屏脚本；

[0117] 运行非首屏脚本，脚本用于获取位于页面非首屏区域中的动态资源。

[0118] 在本实施例中，由于客户端、页面服务器的操作与实施例二、三、四的应用基本相似，所以描述的比较简单，相关之处参见实施例二、三、四的部分说明即可，本实施例在此不加以详述。

[0119] 在本实施例中，资源服务器存储有非首屏脚本、静态资源，非首屏脚本可用于获取位于页面非首屏区域中的动态资源，以便将该动态资源渲染至目标页面文档中，对于页面渲染的过程而言，渲染静态资源占用的资源较低，耗时较少，因此，在客户端渲染静态资源，对于渲染页面首屏区域的耗时影响很少，页面服务器面向多个客户端，相比于页面服务器渲染静态资源、非首屏区域的动态资源，在客户端渲染静态资源、非首屏区域的动态资源，可较少页面服务器的负载，并且，由于非首屏区域初始不可见，在用户浏览首屏区域的时间内可完成非首屏区域的渲染，可保证页面的完整显示。

[0120] 实施例二

[0121] 图2为本发明实施例二提供的一种页面的处理方法的流程图，本实施例提供一个框架，用于开发项目(即页面)，该项目整体进行编译以及分发到页面服务器、资源服务器，所谓编译，可以用编译程序将页面从源语言编写的源程序产生目标程序的过程。

[0122] 对于一个客户端而言，由于功能较多，其首页可按照原生的框架进行开发，如MWC，而二级或二级以下的页面，功能相对简单，可按照本实施例所提供的框架进行开发。

[0123] 在页面服务器部署该页面的、已编译的页面文档(如HMTL文档)、在资源服务器部署该页面的、已编译的静态资源。

[0124] 所谓静态资源，可以指在一段时间内固定的资源，如图片、CSS文件等。

[0125] 本实施例可适用于页面服务器在页面文档渲染页面的首屏区域的动态数据的情

况。

[0126] 在一页面中,相对于首屏区域,即为非首屏区域。

[0127] 所谓首屏区域,可以指加载页面时,该页面默认处于屏幕的可视区域中的部分。

[0128] 所谓非首屏区域,可以指加载页面时,该页面默认处于屏幕的可视区域外的部分,该非首屏区域可以通过滑动、拖动等方式移动至屏幕的可视区域显示。

[0129] 该方法可以由页面的处理装置来执行,该页面的处理装置可以由软件和/或硬件实现,可配置在计算机设备中,如页面服务器等,该方法具体包括如下步骤:

[0130] S201、接收客户端发送的、加载一页面的请求。

[0131] 在具体实现中,计算机设备(如页面服务器)除了提供页面加载服务之外,还提供页面渲染服务,包括Node服务器、Vue服务器、Jquery服务器,等等。

[0132] 以Node服务器为例,Node.js应用如下几部分:

[0133] 1、引入required模块:使用require指令来载入Node.js模块。

[0134] 2、创建服务器:使用http.createServer()方法创建服务器,服务器可以通过request参数监听客户端的请求。

[0135] 3、接收请求与响应请求:客户端可以发送请求,服务器接收请求后通过response参数返回响应数据。

[0136] 相应地,配合Node服务器,页面可以指由React开发的单页面或多页面。

[0137] Reactk以声明式编写UI(User Interface,用户界面),可创建交互式UI,为应用的每一个状态设计视图,当数据改变时,React能有效地更新并正确地渲染组件。

[0138] 对于简单组件,React组件使用一个名为render()的方法,接收输入的数据并返回需要展示的内容,这种写法被称为JSX。被传入的数据可在组件中通过this.props在render()访问。

[0139] 对于有状态组件,除了使用外部数据(通过this.props访问)以外,组件还可以维护其内部的状态数据(通过this.state访问)。当组件的状态数据改变时,组件会再次调用render()方法重新渲染对应的标记。

[0140] 对于应用,使用props和state,可以创建一个简易的Todo应用。

[0141] 在本实施例中,客户端可以通过HTTP(HyperText Transfer Protocol,超文本传输协议)、HTTPS(Hyper Text Transfer Protocol over Secure Socket Layer或Hypertext Transfer Protocol Secure,超文本传输安全协议)等协议,将加载页面的请求发送至计算机设备(如页面服务器)。

[0142] 其中,请求的内容可以包括客户端请求加载的页面的标识和/或页面的特征。页面标识可以是能够代表一个唯一确定的页面的信息,例如统一资源标识符(Uniform Resource Identifier,URI),统一资源标识符又具体可以包括统一资源定位符(Uniform Resource Locator,URL),或者统一资源名称(Uniform Resource Name,URN)等等。

[0143] 需要说明的是,对于Node服务器等计算机设备,可以并发地处理两个或两个以上的加载页面的请求。

[0144] S202、响应于所述请求,确定属于所述页面的、已编译的目标页面文档。

[0145] 在计算机设备(如页面服务器)中,可预先存储两个或两个以上页面的、已编译的原始页面文档。

[0146] 原始页面文档是页面的主体，用于传递页面的各类信息。

[0147] 以HTML文档作为页面文档的示例，HTML文档属于页面的框架，可以指示应用如何显示Web文档(即页面)的信息，如何链接各种信息。使用HTML文本可以在其生成的文档中含有其它文档，或者含有图像、音频、视频等，从而形成超文本。

[0148] 在该原始页面文档中渲染位于页面首屏区域中的动态资源，可生成目标页面文档，即目标页面文档已渲染有位于页面首屏区域中的动态资源。

[0149] 其中，动态资源可以指实时发生变化的资源，该动态资源可以由本领域技术人员根据实际情况设定，本实施例对此不加以限制。

[0150] 例如，对于直播网站的页面，该动态资源可以包括排行榜中的数据，如某个用户的排名及其排行数值(如主播用户的人气值、观众用户的贡献值、陪玩用户的订单数等)。

[0151] 在本实施例中，计算机设备(如页面服务器)以已编译的项目作为路由，响应于客户端的请求，在本地查找该请求(以URL表示)对应的项目。

[0152] 如果未查找到，则向客户端返回错误页面，如404页面。

[0153] 如果查找到，则可获取属于该请求对应页面的、已编译的目标页面文档。

[0154] 对于实时性要求较高的项目、实时性要求较低的项目且未缓存或缓存失效等情况，可实时生成目标页面文档。

[0155] 在此情况中，可查找属于页面的、已编译的原始页面文档，如HTML文档，获取位于页面首屏区域的动态资源，将动态资源渲染至原始页面文档中，获得目标页面文档。

[0156] 该项目包括属于页面的、已编译的首屏脚本，如JavaScript文件，存储在计算机设备(如页面服务器)。

[0157] 计算机设备(如页面服务器)可查找属于页面的、已编译的首屏脚本，运行首屏脚本，该首屏脚本用于从业务服务器获取位于页面首屏区域中的动态资源。

[0158] 进一步而言，若该项目属于实时性要求较低的项目，则可以将目标页面文档存储至缓存中，以及，对目标页面文档设置有效时间，有效时间用于在超时时，清除目标页面文档，以在有效时间内直接提取目标页面文档。

[0159] 对于实时性要求较低的项目且缓存生效等情况，可在缓存中查找属于页面的、已编译的目标页面文档，若查找到，则从缓存中提取目标页面文档。

[0160] S203、将所述目标页面文档发送至所述客户端。

[0161] 在本实施例中，计算机设备(如页面服务器)可以通过HTTP、HTTPS等协议，将客户端所请求页面的目标页面文档发送至该客户端。

[0162] 客户端用于从资源服务器获取属于页面的、已编译的静态资源，将静态资源渲染至目标页面文档、以显示页面的首屏区域。

[0163] 在本实施例中，接收客户端发送的、加载一页面的请求，响应于该请求，确定属于页面的、已编译的目标页面文档，目标页面文档已渲染位于页面首屏区域中的动态资源，将目标页面文档发送至客户端，客户端用于从资源服务器获取属于所述页面的、已编译的静态资源，将静态资源渲染至目标页面文档、以显示页面的首屏区域。对于页面渲染的过程而言，拉取并渲染动态资源占用的设备资源较高，耗时较多，而服务端所在的设备的性能一般比客户端所在的设备要强、网络环境一般比客户端所在的设备要好，因此，在服务端中拉取并渲染页面首屏的动态资源，可提高页面首屏的渲染速度、减少页面首屏的渲染时间，从而

提高在客户端显示页面首屏的速度，由于首屏为用户所见，可给用户带来页面显示快的效果。

[0164] 此外，出于安全性的考虑，搜索引擎一般是拉取HTML文档，不会对里面的脚本文件进行解析，将导致网站无法获得有效的搜索引擎排名；而本实施例中，在服务端提供渲染动态数据的目标页面文档，有利用搜索引擎获取到包含数据的页面，增加项目的曝光度，从而提高项目的搜索引擎排名。

[0165] 实施例三

[0166] 图3为本发明实施例三提供的一种页面的处理方法的流程图，本实施例提供一个框架，用于开发项目(即页面)，该项目整体进行编译以及分发到页面服务器、资源服务器，所谓编译，可以用编译程序将页面从源语言编写的源程序产生目标程序的过程。

[0167] 对于一个客户端而言，由于功能较多，其首页可按照原生的框架进行开发，如MWC，而二级或二级以下的页面，功能相对简单，可按照本实施例所提供的框架进行开发。

[0168] 在页面服务器部署该页面的、已编译的页面文档(如HMTL文档)、在资源服务器部署该页面的、已编译的静态资源。

[0169] 所谓静态资源，可以指在一段时间内固定的资源，如图片、CSS文件等。

[0170] 本实施例可适用于客户端从页面服务器获取已渲染动态数据的目标页面文档、从资源服务器获取静态资源，在目标页面文档渲染页面的首屏区域的情况。

[0171] 在一页面中，相对于首屏区域，即为非首屏区域。

[0172] 所谓首屏区域，可以指加载页面时，该页面默认处于屏幕的可视区域中的部分。

[0173] 所谓非首屏区域，可以指加载页面时，该页面默认处于屏幕的可视区域外的部分，该非首屏区域可以通过滑动、拖动等方式移动至屏幕的可视区域显示。

[0174] 该方法可以由页面的处理装置来执行，该页面的处理装置可以由软件和/或硬件实现，可配置在计算机设备中，如移动终端(如手机、平板电脑、智能穿戴设备(如智能手表、智能眼镜等))等，该方法具体包括如下步骤：

[0175] S301、接收显示页面的操作。

[0176] 在本实施例中，在计算机设备中，操作系统包括Android(安卓)、iOS、windows等等，在该操作系统中可以安装、运行可加载页面的客户端，例如，直播应用、浏览器、即时通讯工具、搜索应用，等等。

[0177] 以直播应用为客户端的示例，该直播应用可播放实时的直播视频数据，也可以点播在先的直播视频数据，也可以提供其他服务，如游戏陪玩服务、游戏等等。

[0178] 对于直播应用等非浏览器的客户端，可以配置页面容器，以显示页面，例如，在Android系统中，该页面容器可以为WebView，WebView是一个基于webkit引擎、展现页面的控件。

[0179] 一般情况下，实现WebView有以下两种不同的方法：

[0180] 第一种方法：

[0181] 1、在要Activity中实例化WebView组件：WebView webView＝new WebView(this)；

[0182] 2、调用WebView的loadUrl()方法，设置WevView要显示的页面：

[0183] 互联网用：webView.loadUrl("http://www.***.com")；

[0184] 本地文件用：webView.loadUrl("file:///android_asset/XX.html")；本地文件

存放在:assets文件中;

[0185] 3、调用Activity的setContentView()方法来显示页面视图;

[0186] 4、用WebView显示了多个页面以后为了让WebView支持回退功能,覆盖Activity类的onKeyDown()方法;

[0187] 5、在AndroidManifest.xml(根目录)文件中添加权限。

[0188] 第二种方法:

[0189] 1、在布局文件中声明WebView;

[0190] 2、在Activity中实例化WebView;

[0191] 3、调用WebView的loadUrl()方法,设置WevView要显示的网页;

[0192] 4、为了让WebView能够响应超链接功能,调用setWebViewClient()方法,设置WebView视图;

[0193] 5、用WebView显示了多个页面以后为了让WebView支持回退功能,覆盖Activity类的onKeyDown()方法;

[0194] 6、在AndroidManifest.xml文件中添加权限。

[0195] 当然,上述非浏览器的客户端显示页面的方式只是作为示例,在实施本发明实施例时,可以根据实际情况(如操作系统的类型、版本等)设置其他非浏览器的客户端显示页面的方式,本实施例对此不加以限制。另外,除了上述非浏览器的客户端显示页面的方式外,本领域技术人员还可以根据实际需要采用其它非浏览器的客户端显示页面的方式,本实施例对此也不加以限制。

[0196] 在本实施例中,客户端可以在UI(User Interface,用户界面)上显示操作元素,如输入栏、控件等,用户在UI上通过该操作元素触发个浏览页面的操作。

[0197] 例如,用户可以通过点击控件来打开一个页面,或者在地址栏输入网址来打开一个页面等等。当用户点击控件时,相当于接收到了显示页面的操作;同样,当用户在地址栏输入网址并点击确定按钮或按下回车键时,相当于接收到了显示页面的操作。

[0198] S302、响应于所述操作,向页面服务器请求确定属于所述页面的、已编译的目标页面文档。

[0199] 在本实施例中,客户端响应于用户发起的操作,通过HTTP、HTTPS等协议,将加载相应页面的请求发送至页面服务器。

[0200] 页面服务器响应于客户端的请求,返回相应页面的、已编译的目标页面文档(如HTML文档),该目标页面文档已渲染位于页面首屏区域中的动态资源。

[0201] 在一种渲染的方式中,页面服务器查找属于页面的、已编译的原始页面文档;获取位于页面首屏区域的动态资源,将动态资源渲染至原始页面文档中,获得目标页面文档。

[0202] 在获取位于页面非首屏区域的动态资源时,可查找属于页面的、已编译的首屏脚本;运行首屏脚本,首屏脚本用于获取位于页面首屏区域中的动态资源。

[0203] 进一步地,页面服务器可将目标页面文档存储至缓存中,对目标页面文档设置有效时间,该有效时间用于在超时时,清除目标页面文档。

[0204] 在另一种渲染的方式中,页面服务器在缓存中查找属于页面的、已编译的目标页面文档;若查找到,则从缓存中提取目标页面文档。

[0205] 在本实施例中,由于页面服务器返回目标页面文档的方式与实施例一的应用基本

相似，所以描述的比较简单，相关之处参见实施例一的部分说明即可，本实施例在此不加以详述。

[0206] S303、从资源服务器获取属于所述页面的、已编译的静态资源。

[0207] 资源服务器，与页面服务器相互独立，用于存储两个或两个以上页面中的静态资源，如CDN、Redis服务器等。

[0208] 以CDN为例，通过在网络各处放置节点服务器所构成的在现有的网络基础之上的一层智能虚拟网络，CDN可以实时地根据网络流量和各节点的连接、负载状况以及到客户端的距离和响应时间等综合信息将客户端的请求重新导向离客户端最近的服务节点上，使得客户端户可就近取得所需内容(如页面的静态资源)，解决网络拥挤的状况，提高访问的响应速度，从而减少获取内容的时间。

[0209] 在本实施例中，客户端通过HTTP、HTTPS等协议，向资源服务器发送访问页面的请求，资源服务器响应于该请求，通过HTTP、HTTPS等协议，将属于该页面的、已编译的静态资源发送至客户端。

[0210] 需要说明的是，对于页面的静态资源而言，并不区分首屏区域与非首屏区域，即该静态资源既包括位于首屏区域的静态资源，也包括属于非首屏区域的静态资源。

[0211] S304、将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

[0212] 客户端在接收到目标页面文档、静态资源之后，则可以将静态资源渲染至目标页面文档，从而在计算机设备(如移动设备)的屏幕的可视区域显示页面的首屏区域。

[0213] 在一个示例中，渲染页面中静态资源的过程如下：

[0214] 1、客户端开始载入HTML文本的源代码，当检测到＜head＞标签内有一个＜link＞标签时，引用CSS文件；

[0215] 2、客户端继续载入HTML文本中＜body＞部分的代码，此时CSS文件已经加载到，可以开始渲染页面；

[0216] 3、当客户端在HTML文本中检测到一个＜img＞标签时，引用了一张图片；

[0217] 4、由于图片占用了一定面积，影响了后面段落的排布，因此客户端可能重新渲染这部分代码。

[0218] 在本实施例中，接收显示页面的操作，响应于该操作，向页面服务器请求确定属于页面的、已编译的目标页面文档，目标页面文档已渲染位于页面首屏区域中的动态资源，从资源服务器获取属于页面的、已编译的静态资源，将静态资源渲染至目标页面文档、以显示页面的首屏区域。对于页面渲染的过程而言，渲染静态资源占用的资源较低，耗时较少，因此，在客户端渲染静态资源，对于渲染页面的首屏区域的时间影响很少，并且，非首屏区域初始不可见，页面服务器面向多个客户端，相比于页面服务器渲染静态资源、非首屏区域的动态资源，在客户端渲染静态资源、非首屏区域的动态资源，在页面服务器中渲染页面首屏的动态资源，可提高页面首屏的渲染速度、减少页面首屏的渲染时间，从而提高在客户端显示页面首屏的速度，由于首屏为用户所见，可给用户带来页面显示快的效果。

[0219] 再者，对于页面渲染的过程而言，渲染静态资源占用的资源较低，耗时较少，因此，在客户端渲染静态资源，对于渲染页面首屏区域的耗时影响很少，页面服务器面向多个客户端，相比于页面服务器渲染静态资源、非首屏区域的动态资源，在客户端渲染静态资源、非首屏区域的动态资源，可较少页面服务器的负载。

[0220] 此外，出于安全性的考虑，搜索引擎一般是拉取HTML文档，不会对里面的脚本文件进行解析，将导致网站无法获得有效的搜索引擎排名；而本实施例中，在服务端提供渲染动态数据的目标页面文档，有利用搜索引擎获取到包含数据的页面，增加项目的曝光度，从而提高项目的搜索引擎排名。

[0221] 实施例四

[0222] 图4为本发明实施例四提供的一种页面的处理方法的流程图，本实施例以前述实施例为基础，进一步增加渲染非首屏区域的处理操作。该方法具体包括如下步骤：

[0223] S401、接收显示页面的操作。

[0224] S402、响应于所述操作，向页面服务器请求确定属于所述页面的、已编译的目标页面文档。

[0225] 其中，目标页面文档已渲染位于页面首屏区域中的动态资源。

[0226] S403、从资源服务器获取属于所述页面的、已编译的静态资源。

[0227] S404、将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

[0228] S405、获取位于所述页面非首屏区域中的动态资源。

[0229] 在具体实现中，动态资源可以指实时发生变化的资源，该动态资源可以由本领域技术人员根据实际情况设定，本实施例对此不加以限制。

[0230] 例如，对于直播网站的页面，该动态资源可以包括排行榜中的数据，如某个用户的排名及其排行数值(如主播用户的人气值、观众用户的贡献值、陪玩用户的订单数等)。

[0231] 在一种获取方式中，该项目包括属于页面的、已编译的非首屏脚本，如JavaScript文件，存储在资源服务器中。

[0232] 在此方式中，客户端可从资源服务器获取属于页面的非首屏脚本，以及，运行该非首屏脚本，该脚本用于获取位于页面非首屏区域中的动态资源。

[0233] 为了提高页面加载的速度，客户端向资源服务器发送访问页面的请求之后，资源服务器可一同返回静态资源和非首屏脚本。

[0234] 当然，客户端也可以分别独立从资源服务器获取静态资源和非首屏脚本，本实施例对此不加以限制。

[0235] S406、将所述动态资源渲染至所述目标页面文档、以加载所述页面的非首屏区域。

[0236] 客户端将属于非首屏区域的动态资源渲染至目标页面文档(如HTML文档)，则可以完成页面的非首屏区域的加载。

[0237] 在用户浏览页面的过程中，可以通过拖动、滑动等操作，将非首屏区域移入计算机设备(如移动终端)的屏幕的可视区域，此时非首屏区域处于可视状态。

[0238] 在本实施例中，客户端获取位于页面非首屏区域中的动态资源，将动态资源渲染至目标页面文档、以加载页面的非首屏区域，页面服务器面向多个客户端，相比于页面服务器非首屏区域的动态资源，在客户端渲染非首屏的动态资源，可较少页面服务器的负载，并且，由于非首屏区域初始不可见，在用户浏览首屏区域的时间内可完成非首屏区域的渲染，可保证页面的完整显示。

[0239] 为使本领域技术人员更好地理解本实施例，以下通过具体的示例来说明本实施例中页面的处理方法。

[0240] 对于一个直播应用的排行榜(页面)，名称为“陪玩大神榜”，用对提供陪玩业务的

用户进行排名。

[0241] 如图5A所示，页面服务器510存储该排行榜的HMTL文档、首屏区域的JavaScript文件，CDN520存储该排行榜的CSS文件、图片(如背景、下单、用户徽章、排名名次、榜单说明、页数等)、非首屏区域的JavaScript文件。

[0242] 在移动终端530中安装直播应用。

[0243] 如图5B所示，用户启动该直播应用，并点击“陪玩”栏目，此时，显示“陪玩大神榜”的入口对象。

[0244] 用户点击该入口对象，触发显示排行榜的操作。

[0245] 响应于该操作，直播应用发送加载排行榜的请求至页面服务器510。

[0246] 页面服务器510查找是否存在该排行榜的项目。

[0247] 若无相应的项目，则页面服务器510返回404页面至直播应用。

[0248] 若有相应的项目，则页面服务器510判断该项目是否使用缓存。

[0249] 如果未使用缓存，则运行首屏区域的JavaScript文件，从业务服务器540拉取该排行榜中位于首屏区域的排名数据，如排名第1-5名的用户的用户头像、用户昵称、接单量等，将该位于首屏区域的排名数据渲染至HTML文档。

[0250] 如果已使用缓存，则进一步判断缓存是否过期。

[0251] 如果未过期，则提取已渲染位于首屏区域的排名数据HTML文档。

[0252] 如果已过期，则运行首屏区域的JavaScript文件，从业务服务器540拉取该排行榜中位于首屏区域的排名数据，如排名第1-5名的用户的用户头像、用户昵称、接单量等，将该位于首屏区域的排名数据渲染至HTML文档。

[0253] 对于未使用缓存、已使用缓存但已过期的情况，将已渲染位于首屏区域的排名数据HTML文档写入缓存，并设置有效时间。

[0254] 页面服务器510将已渲染位于首屏区域的排名数据HTML文档发送至直播应用。

[0255] 直播应用从CDN520拉取CSS文件、图片以及非首屏区域的JavaScript文件。

[0256] 如图5C所示，直播应用将CSS文件、图片渲染至HTML文档，可显示首屏区域551。

[0257] 如图5C所示，直播应用运行非首屏区域的JavaScript文件，从业务服务器540拉取该排行榜中位于非首屏区域的排名数据，如排名第6-10名的用户的用户头像、用户昵称、接单量等，将该位于非首屏区域的排名数据渲染至HTML文档，可加载非首屏区域552，待用户向下滑动时，则可显示非首屏区域552。

[0258] 实施例五

[0259] 图6为本发明实施例五提供的一种页面的处理装置的结构示意图，该装置具体可以包括如下模块：

[0260] 请求接收模块601，用于接收客户端发送的、加载一页面的请求；

[0261] 目标页面文档确定模块602，用于响应于所述请求，确定属于所述页面的、已编译的目标页面文档，所述目标页面文档已渲染有位于所述页面首屏区域中的动态资源；

[0262] 目标页面文档发模块603，用于将所述目标页面文档发送至所述客户端，所述客户端用于从资源服务器获取属于所述页面的、已编译的静态资源，将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

[0263] 在本发明的一个实施例中，所述目标页面文档确定模块602包括：

[0264] 原始页面文档查找子模块，用于查找属于所述页面的、已编译的原始页面文档；

[0265] 首屏动态资源获取子模块，用于获取位于所述页面首屏区域的动态资源；

[0266] 动态资源渲染子模块，用于将所述动态资源渲染至所述原始页面文档中，获得目标页面文档。

[0267] 在本发明的一个实施例中，所述首屏动态资源获取子模块包括：

[0268] 首屏脚本查找单元，用于查找属于所述页面的、已编译的首屏脚本；

[0269] 首屏脚本运行单元，用于运行所述首屏脚本，所述首屏脚本用于获取位于所述页面首屏区域中的动态资源。

[0270] 在本发明的一个实施例中，所述目标页面文档确定模块602还包括：

[0271] 目标页面文档缓存子模块，用于将所述目标页面文档存储至缓存中；

[0272] 有效时间设置子模块，用于对所述目标页面文档设置有效时间，所述有效时间用于在超时时，清除所述目标页面文档。

[0273] 在本发明的一个实施例中，所述目标页面文档确定模块602包括：

[0274] 目标页面文档查找子模块，用于在缓存中查找属于所述页面的、已编译的目标页面文档；

[0275] 目标页面文档提取子模块，用于若查找到，则从所述缓存中提取所述目标页面文档。

[0276] 本发明实施例所提供的页面的处理装置可执行本发明任意实施例所提供的页面的处理方法，具备执行方法相应的功能模块和有益效果。

[0277] 实施例六

[0278] 图7为本发明实施例六提供的一种页面的处理装置的结构示意图，该装置具体可以包括如下模块：

[0279] 操作接收模块701，用于接收显示页面的操作；

[0280] 目标页面文档请求模块702，用于响应于所述操作，向页面服务器请求确定属于所述页面的、已编译的目标页面文档，所述目标页面文档已渲染位于所述页面首屏区域中的动态资源；

[0281] 静态资源获取模块703，用于从资源服务器获取属于所述页面的、已编译的静态资源；

[0282] 静态资源渲染模块704，用于将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

[0283] 在本发明的一个实施例中，还包括：

[0284] 非首屏动态资源获取模块，用于获取位于所述页面非首屏区域中的动态资源；

[0285] 动态资源渲染模块，用于将所述动态资源渲染至所述目标页面文档、以加载所述页面的非首屏区域。

[0286] 在本发明的一个实施例中，所述非首屏动态资源获取模块包括：

[0287] 非首屏脚本获取子模块，用于从资源服务器获取属于所述页面的非首屏脚本；

[0288] 非首屏脚本运行子模块，用于运行所述非首屏脚本，所述脚本用于获取位于所述页面非首屏区域中的动态资源。

[0289] 本发明实施例所提供的页面的处理装置可执行本发明任意实施例所提供的页面

的处理方法，具备执行方法相应的功能模块和有益效果。

[0290] 实施例七

[0291] 图8为本发明实施例七提供的一种页面的处理装置的结构示意图，该装置具体可以包括如下模块：

[0292] 源文件获取模块801，用于获取属于页面的源文件；

[0293] 文件编译模块802，用于将所述源文件编译为目标文件，所述目标文件包括原始页面文档、首屏脚本、非首屏脚本与静态资源，所述首屏脚本用于获取位于所述页面首屏区域中的动态资源、所述非首屏脚本用于获取位于所述页面非首屏区域中的动态资源；

[0294] 第一资源分发模块803，用于将所述原始页面文档与所述首屏脚本分发至页面服务器；

[0295] 第二资源分发模块804，用于将所述非首屏脚本与所述静态资源分发至资源服务器。

[0296] 在本发明的一个实施例中，还包括：

[0297] 目标文件压缩模块，用于对所述目标文件进行压缩。

[0298] 本发明实施例所提供的页面的处理装置可执行本发明任意实施例所提供的页面的处理方法，具备执行方法相应的功能模块和有益效果。

[0299] 实施例八

[0300] 图9为本发明实施例八提供的一种计算机设备的结构示意图。如图9所示，该计算机设备包括处理器900、存储器901、通信模块902、输入装置903和输出装置904；计算机设备中处理器900的数量可以是一个或多个，图9中以一个处理器900为例；计算机设备中的处理器900、存储器901、通信模块902、输入装置903和输出装置904可以通过总线或其他方式连接，图9中以通过总线连接为例。

[0301] 存储器901作为一种计算机可读存储介质，可用于存储软件程序、计算机可执行程序以及模块，如本实施例中的页面的处理方法对应的模块(例如，如图6所示的页面的处理装置中的请求接收模块601、目标页面文档确定模块602和目标页面文档发模块603；或者，如图7所示的页面的处理装置中的操作接收模块701、目标页面文档请求模块702、静态资源获取模块703和静态资源渲染模块704；或者，如图8所示的页面的处理装置中的源文件获取模块801、文件编译模块802、第一资源分发模块803和第二资源分发模块804)。处理器900通过运行存储在存储器901中的软件程序、指令以及模块，从而执行计算机设备的各种功能应用以及数据处理，即实现上述的页面的处理方法。

[0302] 存储器901可主要包括存储程序区和存储数据区，其中，存储程序区可存储操作系统、至少一个功能所需的应用程序；存储数据区可存储根据计算机设备的使用所创建的数据等。此外，存储器901可以包括高速随机存取存储器，还可以包括非易失性存储器，例如至少一个磁盘存储器件、闪存器件、或其他非易失性固态存储器件。在一些实例中，存储器901可进一步包括相对于处理器900远程设置的存储器，这些远程存储器可以通过网络连接至计算机设备。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0303] 通信模块902，用于与显示屏建立连接，并实现与显示屏的数据交互。输入装置903可用于接收输入的数字或字符信息，以及产生与计算机设备的用户设置以及功能控制有关

的键信号输入。

[0304] 本实施例提供的计算机设备，可执行本发明任一实施例提供的页面的处理方法，具体相应的功能和有益效果。

[0305] 实施例九

[0306] 本发明实施例九还提供一种计算机可读存储介质，其上存储有计算机程序，该计算机程序被处理器执行时实现一种页面的处理方法。

[0307] 在一种情况中，该方法包括：

[0308] 接收客户端发送的、加载一页面的请求；

[0309] 响应于所述请求，确定属于所述页面的、已编译的目标页面文档，所述目标页面文档已渲染有位于所述页面首屏区域中的动态资源；

[0310] 将所述目标页面文档发送至所述客户端，所述客户端用于从资源服务器获取属于所述页面的、已编译的静态资源，将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

[0311] 在另一种情况中，该方法包括：

[0312] 接收显示页面的操作；

[0313] 响应于所述操作，向页面服务器请求确定属于所述页面的、已编译的目标页面文档，所述目标页面文档已渲染位于所述页面首屏区域中的动态资源；

[0314] 从资源服务器获取属于所述页面的、已编译的静态资源；

[0315] 将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域。

[0316] 在又一种情况中，该方法包括：

[0317] 获取属于页面的源文件；

[0318] 将所述源文件编译为目标文件，所述目标文件包括原始页面文档、首屏脚本、非首屏脚本与静态资源，所述首屏脚本用于获取位于所述页面首屏区域中的动态资源、所述非首屏脚本用于获取位于所述页面非首屏区域中的动态资源；

[0319] 将所述原始页面文档与所述首屏脚本分发至页面服务器；

[0320] 将所述非首屏脚本与所述静态资源分发至资源服务器。

[0321] 当然，本发明实施例所提供的计算机可读存储介质，其计算机程序不限于如上所述的方法操作，还可以执行本发明任意实施例所提供的页面的处理方法中的相关操作。

[0322] 通过以上关于实施方式的描述，所属领域的技术人员可以清楚地了解到，本发明可借助软件及必需的通用硬件来实现，当然也可以通过硬件实现，但很多情况下前者是更佳的实施方式。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品可以存储在计算机可读存储介质中，如计算机的软盘、只读存储器(Read-Only Memory，ROM)、随机存取存储器(Random Access Memory，RAM)、闪存(FLASH)、硬盘或光盘等，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)执行本发明各个实施例所述的方法。

[0323] 值得注意的是，上述页面的处理装置的实施例中，所包括的各个单元和模块只是按照功能逻辑进行划分的，但并不局限于上述的划分，只要能够实现相应的功能即可；另外，各功能单元的具体名称也只是为了便于相互区分，并不用于限制本发明的保护范围。

[0324] 注意，上述仅为本发明的较佳实施例及所运用技术原理。本领域技术人员会理解，

本发明不限于这里所述的特定实施例，对本领域技术人员来说能够进行各种明显的变化、重新调整和替代而不会脱离本发明的保护范围。因此，虽然通过以上实施例对本发明进行了较为详细的说明，但是本发明不仅仅限于以上实施例，在不脱离本发明构思的情况下，还可以包括更多其他等效实施例，而本发明的范围由所附的权利要求范围决定。

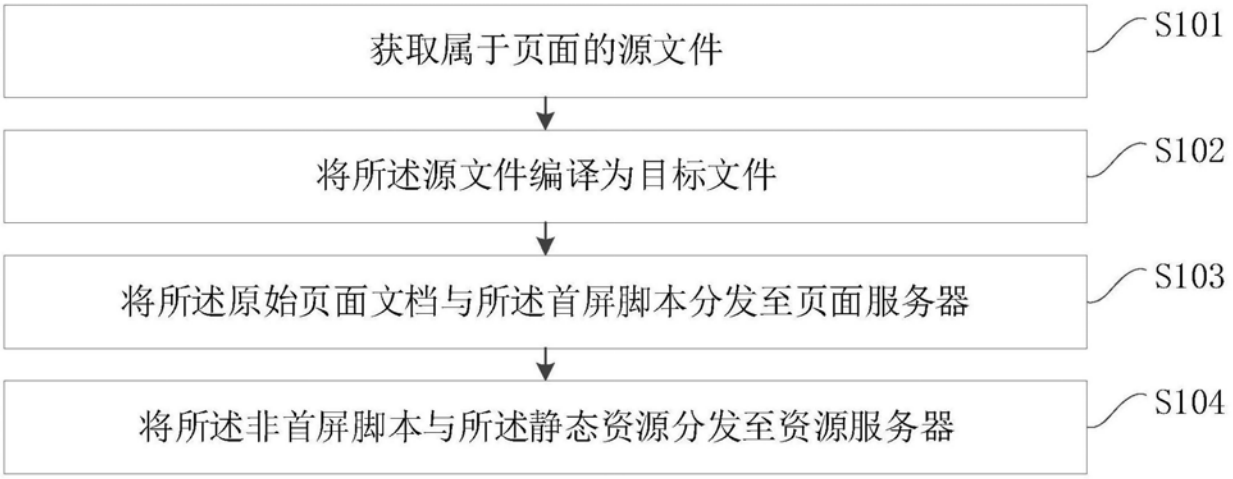

获取属于页面的源文件
S101
将所述源文件编译为目标文件
S102
将所述原始页面文档与所述首屏脚本分发至页面服务器
S103
将所述非首屏脚本与所述静态资源分发至资源服务器
S104

图1

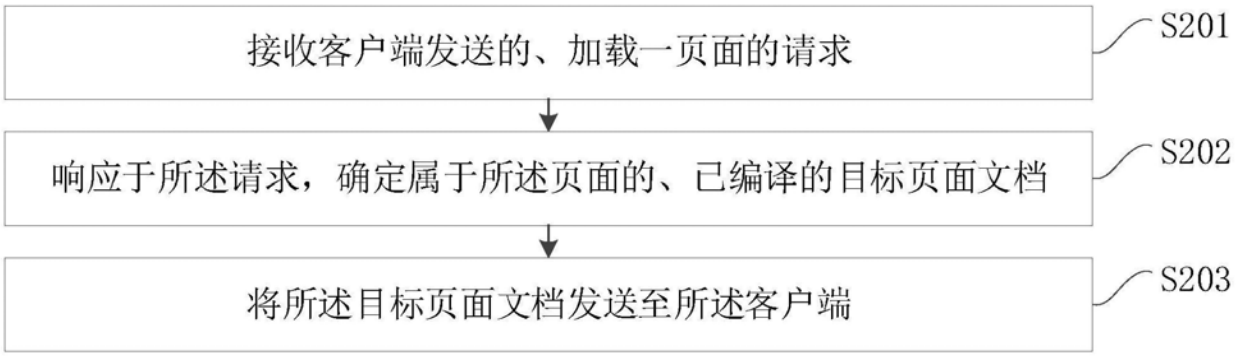

接收客户端发送的、加载一页面的请求
S201
响应于所述请求，确定属于所述页面的、已编译的目标页面文档
S202
将所述目标页面文档发送至所述客户端
S203

图2

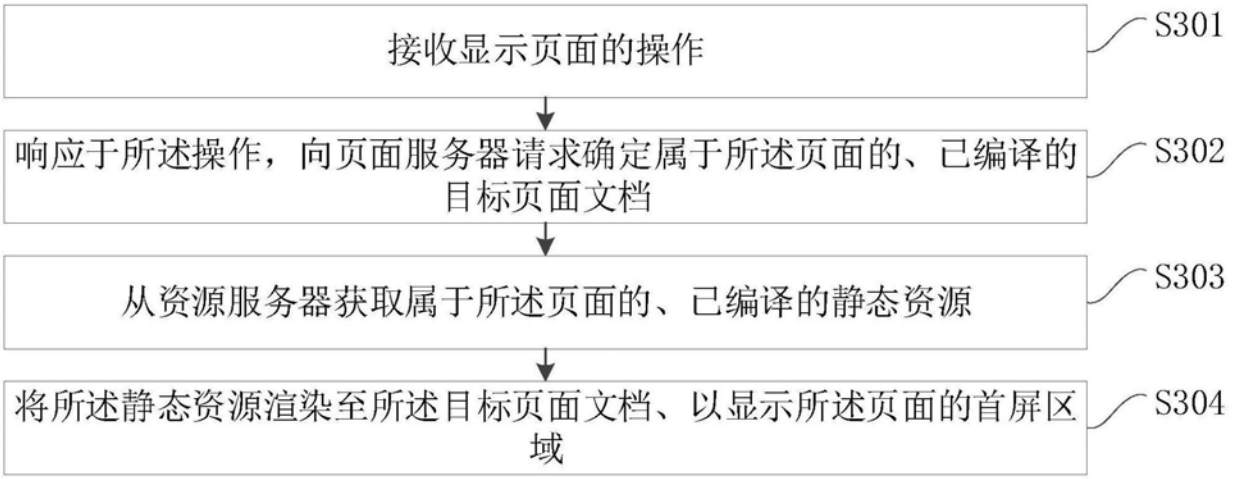

接收显示页面的操作
S301
响应于所述操作，向页面服务器请求确定属于所述页面的、已编译的目标页面文档
S302
从资源服务器获取属于所述页面的、已编译的静态资源
S303
将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域
S304

图3

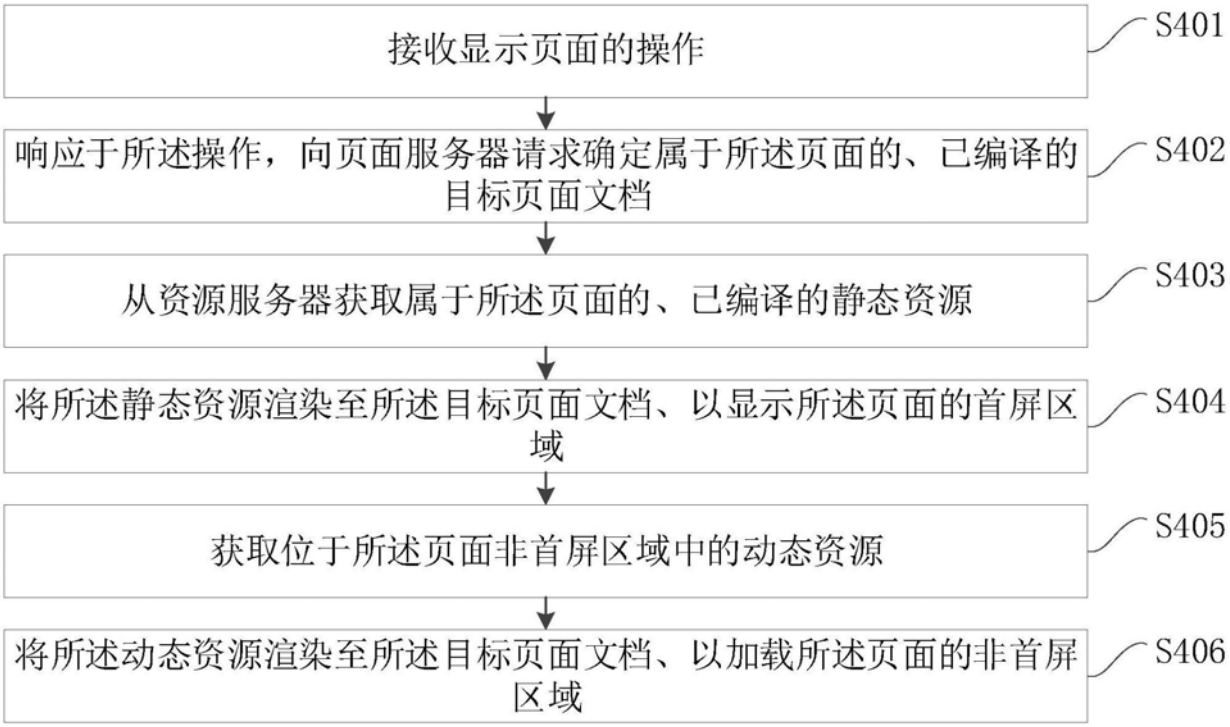

接收显示页面的操作
S401
响应于所述操作，向页面服务器请求确定属于所述页面的、已编译的目标页面文档
S402
从资源服务器获取属于所述页面的、已编译的静态资源
S403
将所述静态资源渲染至所述目标页面文档、以显示所述页面的首屏区域
S404
获取位于所述页面非首屏区域中的动态资源
S405
将所述动态资源渲染至所述目标页面文档、以加载所述页面的非首屏区域
S406

图4

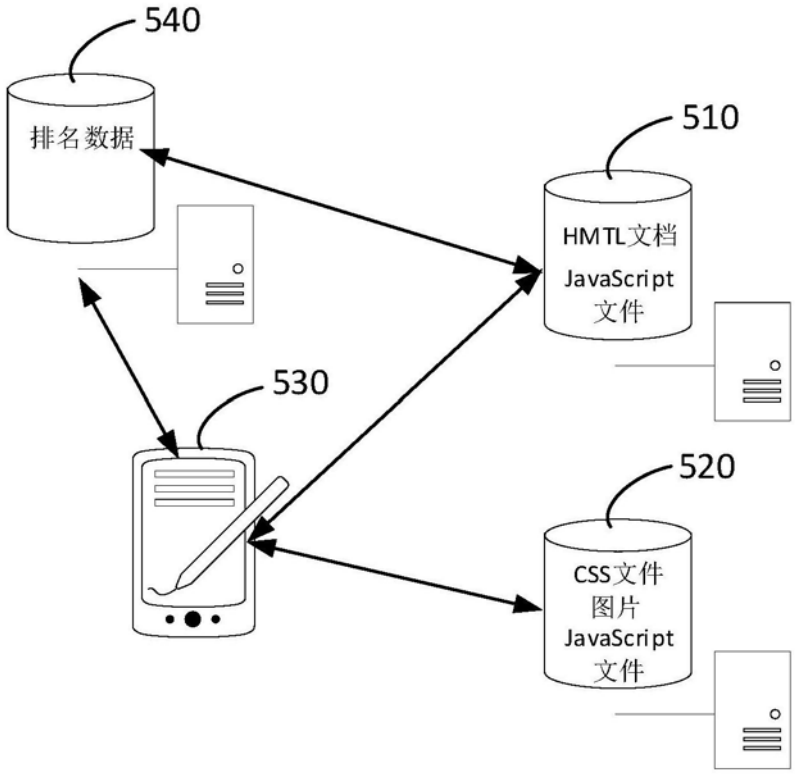

540
排名数据
510
HMTL文档
JavaScript
文件
530
520
CSS文件
图片
JavaScript
文件

图5A



图5B

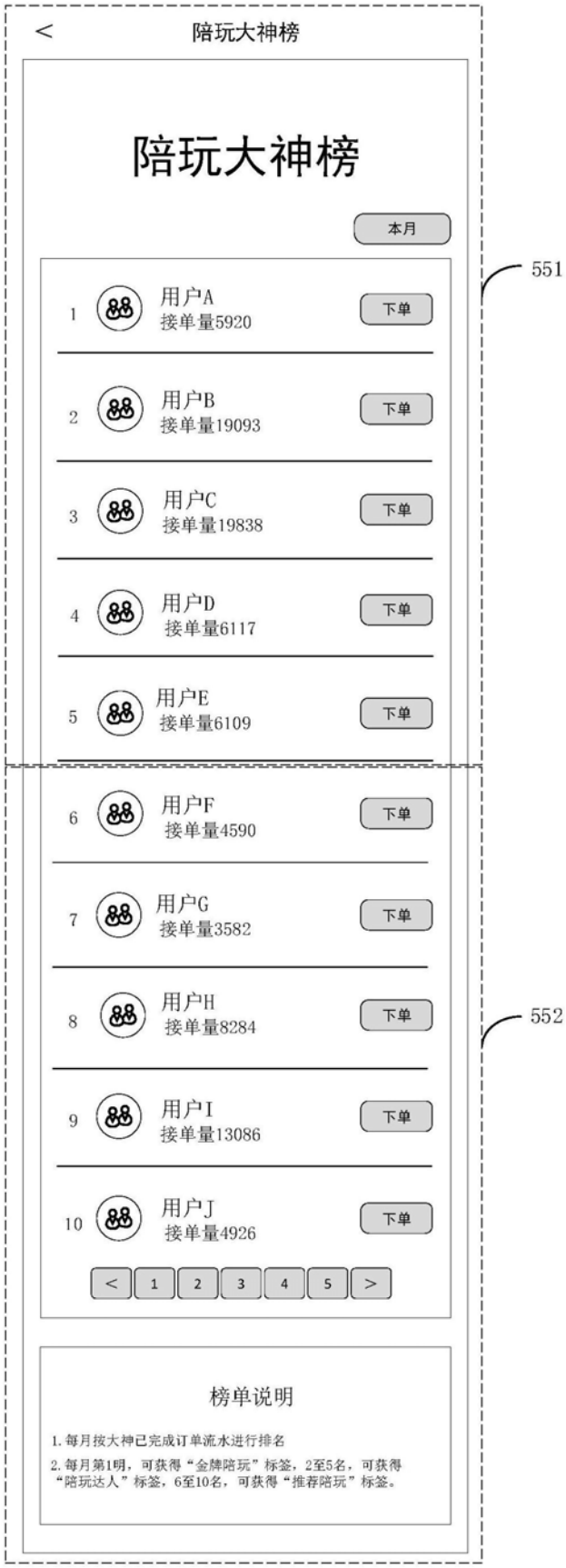

< 陪玩大神榜
陪玩大神榜
本月
551
1 用户A 接单量5920 下单
2 用户B 接单量19093 下单
3 用户C 接单量19838 下单
4 用户D 接单量6117 下单
5 用户E 接单量6109 下单
552
6 用户F 接单量4590 下单
7 用户G 接单量3582 下单
8 用户H 接单量8284 下单
9 用户I 接单量13086 下单
10 用户J 接单量4926 下单
< 1 2 3 4 5 >
榜单说明
1. 每月按大神已完成订单流水进行排名
2. 每月第1明，可获得“金牌陪玩”标签，2至5名，可获得“陪玩达人”标签，6至10名，可获得“推荐陪玩”标签。

图5C


请求接收模块
601
目标页面文档确定模块
602
目标页面文档发模块
603

图6

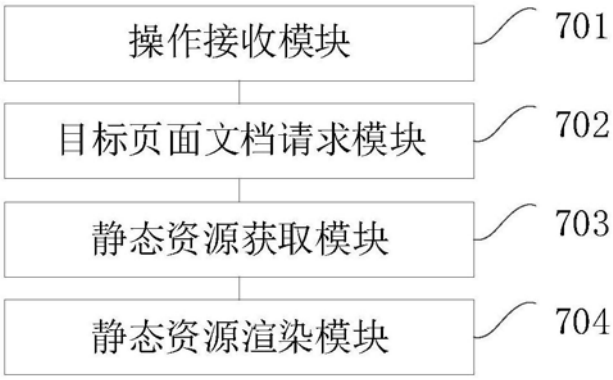

操作接收模块
701
目标页面文档请求模块
702
静态资源获取模块
703
静态资源渲染模块
704

图7

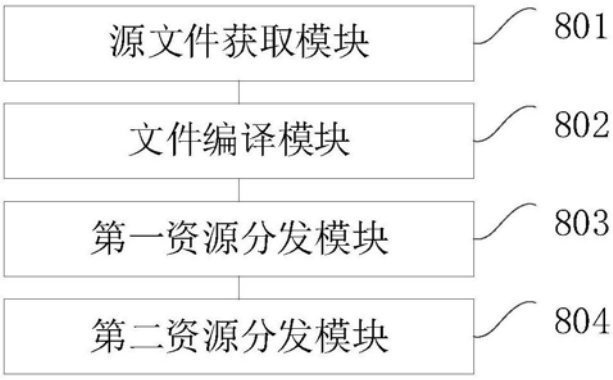

源文件获取模块
801
文件编译模块
802
第一资源分发模块
803
第二资源分发模块
804

图8

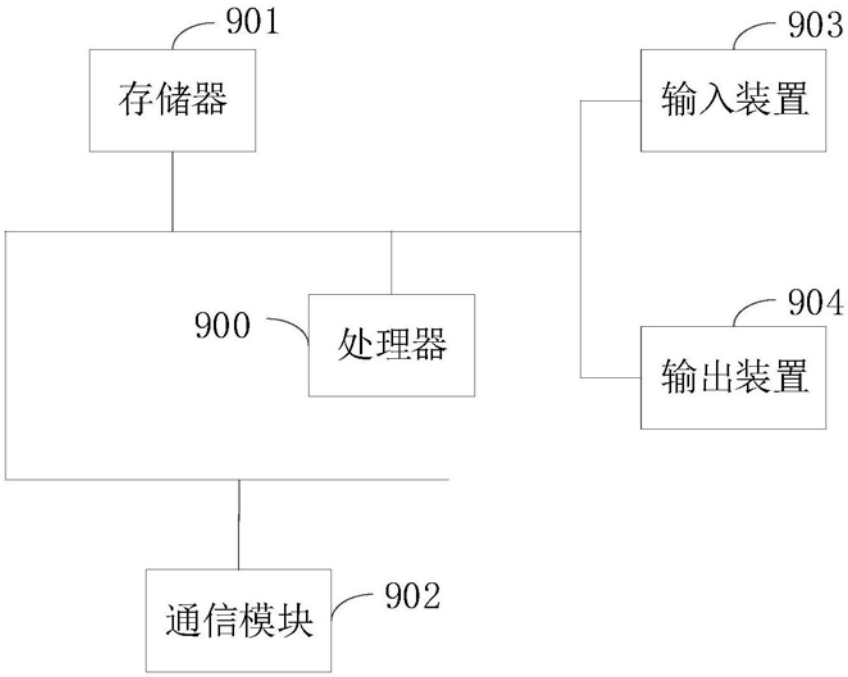


图9