



(12) 发明专利申请

(10) 申请公布号 CN 105045645 A

(43) 申请公布日 2015. 11. 11

(21) 申请号 201510608661. 7

(22) 申请日 2015. 09. 22

(71) 申请人 广州神马移动信息科技有限公司

地址 510627 广东省广州市天河区黄埔大道
西平云路 163 号广电平云广场 B 塔 12
层自编 01 单元

(72) 发明人 侯鹏 孟杰 邵明海 陈一宁

(74) 专利代理机构 北京集佳知识产权代理有限
公司 11227

代理人 王宝筠

(51) Int. Cl.

G06F 9/445(2006. 01)

G06F 17/30(2006. 01)

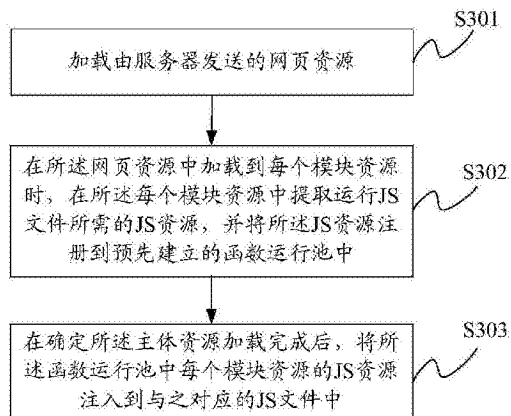
权利要求书2页 说明书9页 附图3页

(54) 发明名称

网页加载方法、装置及系统

(57) 摘要

本申请提供了一种网页加载方法、装置及系统,其中一种方法包括:加载由服务器发送的网页资源;其中,每个模块资源的 JS 文件均设置于所述主体资源之后;在所述网页资源中加载到每个模块时,在所述每个模块资源中提取运行 JS 文件所需的 JS 资源,并将所述 JS 资源注册到预先建立的函数运行池中;在确定所述主体资源加载完成后,将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中。本申请将原本在加载主体资源过程中需要执行的业务逻辑,设置在主体资源加载完成之后再执行;即提前执行主体资源显示首屏页面的程序,延后执行各个模块运行 JS 文件的程序,所以本申请可以提高首屏首屏页面展现效率。



1. 一种网页加载方法,其特征在于,包括:

加载由服务器发送的网页资源;其中,所述网页资源包括头部资源、主体资源和尾部资源,所述主体资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;

在所述网页资源中加载到每个模块资源时,在所述每个模块资源中提取运行 JS 文件所需的 JS 资源,并将所述 JS 资源注册到预先建立的函数运行池中;

在确定所述主体资源加载完成后,将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中。

2. 如权利要求 1 所述的方法,其特征在于,所述确定所述主体资源加载完成,包括:

在所述主体资源加载完成后,触发一个加载完成事件;

在检测到所述加载完成事件后,确定所述主体资源加载完成。

3. 如权利要求 1 所述的方法,其特征在于,在将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中之后,还包括:

清空所述函数运行池。

4. 如权利要求 1 所述的方法,其特征在于,在所述加载由服务器发送的网页资源之前,还包括:

接收用户输入的网页地址;其中,所述网页地址与所述网页资源对应;

向所述服务器发送所述网页地址。

5. 一种网页加载方法,其特征在于,包括:

接收浏览器发送的网页地址;

计算所述网页地址所需的网页资源;所述网页资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;

将所述网页资源发送至所述浏览器。

6. 一种网页加载装置,其特征在于,包括:

加载单元,用于加载由服务器发送的网页资源;其中,所述网页资源包括头部资源、主体资源和尾部资源,所述主体资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;

注册单元,用于在所述网页资源中加载到每个模块资源时,在所述每个模块资源中提取运行 JS 文件所需的 JS 资源,并将所述 JS 资源注册到预先建立的函数运行池中;

确定单元,用于确定所述主体资源加载完成;

注入单元,用于在所述主体资源加载完成后,将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中。

7. 如权利要求 6 所述的装置,其特征在于,所述确定单元具体用于:

在所述主体资源加载完成后,触发一个加载完成事件;在检测到所述加载完成事件后,确定所述主体资源加载完成。

8. 如权利要求 6 所述的装置,其特征在于,还包括:

清空单元,用于在将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中之后,清空所述函数运行池。

9. 如权利要求 6 所述的装置,其特征在于,还包括:

第一接收单元,用于接收用户输入的网页地址;其中,所述网页地址与所述网页资源对应;

第一发送单元,用于向所述服务器发送所述网页地址。

10. 一种网页加载装置,其特征在于,包括:

第二接收单元,用于接收浏览器发送的网页地址;

计算单元,用于计算所述网页地址所需的网页资源;所述网页资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;

第二发送单元,用于将所述网页资源发送至所述浏览器。

11. 一种网页加载系统,其特征在于,包括:

浏览器和服务器;

其中,所述浏览器,用于加载由服务器发送的网页资源;其中,所述网页资源包括头部资源、主体资源和尾部资源,所述主体资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;在所述网页资源中加载到每个模块资源时,在所述每个模块资源中提取运行 JS 文件所需的 JS 资源,并将所述 JS 资源注册到预先建立的函数运行池中;在确定所述主体资源加载完成后,将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中;

所述服务器,用于接收浏览器发送的网页地址;计算所述网页地址所需的网页资源;所述网页资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;将所述网页资源发送至所述浏览器。

网页加载方法、装置及系统

技术领域

[0001] 本申请涉及网络技术领域,尤其涉及网页加载方法、装置及系统。

背景技术

[0002] 随着互联网技术的飞速发展,互联网的网页内容越来越多。随着网页内容的增多,对开发人员的开发效率也提出更高的要求。目前,一个网页基本包括三个部分,头部、主体和尾部;可以理解的是,与网页对应的网页资源也分为三个部分:头部资源、主体资源和尾部资源。其中,主体为网页的主要部分,该部分也是技术人员主要开发的部分;即,技术人员需要对与主体对应的主体资源进行开发。

[0003] 为了提高网页主体的开发效率,业内通常采用的技术手段为将主体模块化,即将主体划分为多个模块,例如,网页上的天气模块、搜索模块、音乐模块等;这使得每个技术人员可以单独负责一个模块,从而可以提高主体的开发效率,这使得一个主体上具有一个或多个模块。在主体包含多个模块的情况下,可以理解的是,与主体对应的主体资源中同样包括各个模块的模块资源。

[0004] 在实际应用中,浏览器会加载服务器发送的网页资源,从而显示网页内容。目前,浏览器会采用由上至下的方式加载网页资源,即先加载网页资源的头部资源,在加载网页资源的主体资源,最后加载网页资源的尾部资源。在加载主体资源的过程中,仍然按照由上至下的加载方式,先加载排列靠前的资源再加载排列靠后的资源,直到所有的主体资源均加载完毕。

[0005] 浏览器在加载主体资源中的模块资源时,会加载模块资源所包括的三种类型的文件:独立功能模板文件、CSS 文件以及 JS 文件;在三种类型的文件均加载完成之后,再继续加载该模块资源后面的其它主体资源。

[0006] 但是,在一个模块资源中,在独立功能模板文件和 CSS 文件加载完成之后,即可显示出模块资源对应的网页内容;而运行 JS 文件仅仅是完善该模块所能实现的功能而已。即各个模块资源的 JS 文件是否执行完成与显示网页内容无关,并且运行各个模块资源的 JS 文件的速率较慢;这导致浏览器需要花费资源和时间先执行各个模块资源的 JS 文件,才能够显示网页内容,即各个模块资源的 JS 文件会阻碍网页内容的渲染。

[0007] 针对网页的首屏页面而言,最重要的是首屏页面的展现效率,即浏览器可以快速显示网页的全部内容。但是,在根据浏览器目前的加载网页资源的方式来看,浏览器需要花费较长的时间才能显示出网页的首屏页面,这降低了首屏页面的展现效率。

[0008] 因此现在需要一种网页加载方式,来提高网页首屏页面的展现效率。

发明内容

[0009] 本申请提供了一种网页加载方法、装置及系统,使用本申请可以能够提高网页的首屏展现效率。

[0010] 为了实现上述目的,本申请采用以下技术手段:

[0011] 一种网页加载方法,包括:

[0012] 加载由服务器发送的网页资源;其中,所述网页资源包括头部资源、主体资源和尾部资源,所述主体资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;

[0013] 在所述网页资源中加载到每个模块资源时,在所述每个模块资源中提取运行 JS 文件所需的 JS 资源,并将所述 JS 资源注册到预先建立的函数运行池中;

[0014] 在确定所述主体资源加载完成后,将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中。

[0015] 优选的,所述确定所述主体资源加载完成,包括:

[0016] 在所述主体资源加载完成后,触发一个加载完成事件;

[0017] 在检测到所述加载完成事件后,确定所述主体资源加载完成。

[0018] 优选的,在将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中之后,还包括:

[0019] 清空所述函数运行池。

[0020] 优选的,在所述加载由服务器发送的网页资源之前,还包括:

[0021] 接收用户输入的网页地址;其中,所述网页地址与所述网页资源对应;

[0022] 向所述服务器发送所述网页地址。

[0023] 一种网页加载方法,包括:

[0024] 接收浏览器发送的网页地址;

[0025] 计算所述网页地址所需的网页资源;所述网页资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;

[0026] 将所述网页资源发送至所述浏览器。

[0027] 一种网页加载装置,包括:

[0028] 加载单元,用于加载由服务器发送的网页资源;其中,所述网页资源包括头部资源、主体资源和尾部资源,所述主体资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;

[0029] 注册单元,用于在所述网页资源中加载到每个模块资源时,在所述每个模块资源中提取运行 JS 文件所需的 JS 资源,并将所述 JS 资源注册到预先建立的函数运行池中;

[0030] 确定单元,用于确定所述主体资源加载完成;

[0031] 注入单元,用于在所述主体资源加载完成后,将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中。

[0032] 优选的,所述确定单元具体用于:

[0033] 在所述主体资源加载完成后,触发一个加载完成事件;在检测到所述加载完成事件后,确定所述主体资源加载完成。

[0034] 优选的,还包括:

[0035] 清空单元,用于在将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中之后,清空所述函数运行池。

[0036] 优选的,还包括:

[0037] 第一接收单元,用于接收用户输入的网页地址;其中,所述网页地址与所述网页资

源对应；

[0038] 第一发送单元,用于向所述服务器发送所述网页地址。

[0039] 一种网页加载装置,包括：

[0040] 第二接收单元,用于接收浏览器发送的网页地址；

[0041] 计算单元,用于计算所述网页地址所需的网页资源；所述网页资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后；

[0042] 第二发送单元,用于将所述网页资源发送至所述浏览器。

[0043] 一种网页加载系统,包括：

[0044] 浏览器和服务器；

[0045] 其中,所述浏览器,用于加载由服务器发送的网页资源；其中,所述网页资源包括头部资源、主体资源和尾部资源,所述主体资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后；在所述网页资源中加载到每个模块资源时,在所述每个模块资源中提取运行 JS 文件所需的 JS 资源,并将所述 JS 资源注册到预先建立的函数运行池中；在确定所述主体资源加载完成后,将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中；

[0046] 所述服务器,用于接收浏览器发送的网页地址；计算所述网页地址所需的网页资源；所述网页资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后；将所述网页资源发送至所述浏览器。由以上内容可以看出,本申请具有以下有益效果：

[0047] 本申请将原本设置每个模块资源内部的 JS 文件,改为设置在网页资源的主体资源之后。由于网页资源采用由上至下的加载方式,所以,浏览器在加载网页资源的过程为,先加载头部资源,再加载主体资源,在主体资源加载完成之后,再运行每个模块内部的 JS 文件,最后加载尾部资源。

[0048] 由于主体资源包含首屏页面的主要显示内容,因此在主体资源加载完毕之后,即可显示出网页的首屏页面。然后,浏览器在再加载各个模块资源的 JS 文件,以完善各个模块的具体功能。

[0049] 即,本申请将原本在加载主体资源过程中需要执行的业务逻辑,设置在主体资源加载完成之后再执行；即提前执行主体资源显示首屏页面的程序,延后执行各个模块运行 JS 文件的程序,所以本申请可以提高首屏页面展现效率。并且,本申请没有增加额外的资源请求开销,便可加速首屏页面的展现,从而可以提高网页性能。

附图说明

[0050] 为了更清楚地说明本申请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简单地介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,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。

[0051] 图 1 为本申请实施例公开的一种网页加载系统的结构示意图；

[0052] 图 2 为本申请实施例公开的一种网页加载方法的流程图；

[0053] 图 3 为本申请实施例公开的又一种网页加载方法的流程图；

- [0054] 图 4 为现有技术中主体资源的示意图；
- [0055] 图 5 为本申请实施例公开的一种网页加载方法中主体资源的示意图；
- [0056] 图 6 为本申请实施例公开的一种网页加载方法中主体资源的示意图；
- [0057] 图 7 为本申请实施例公开的一种网页加载装置的结构图；
- [0058] 图 8 为本申请实施例公开的又一种网页加载装置的结构图；
- [0059] 图 9 为本申请实施例公开的又一种网页加载装置的结构图。

具体实施方式

[0060] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0061] 在介绍本申请的具体实施例之前,首先介绍网页的相关情况：

[0062] 一个网页分可以为三个部分：头部、主体和尾部。服务器在获得网页地址后,可以在数据库中查询与网页地址对应的网页资源,即头部资源、主体资源和尾部资源。其中,头部资源主要作用为显示公司或网站标志性的内容,比如,企业 LOGO、企业名称等信息；尾部资源主要作用为显示企业的联系方式、ICP 备案等信息；主体资源的主要作用为显示网页的主要内容。

[0063] 由于主体资源的作用为显示网页的主要内容,所以,在网页资源中主体资源占有重要作用。在浏览器加载网页资源的过程中,如果主体资源加载完成,则一个网页的大体内容则显示完成。

[0064] 为了提高技术人员开发网页的效率,在网页的主体一般有多个模块,每个模块均有自己的模块资源。每个模块资源主要包括有独立功能模板、CSS 文件以及 JS 文件。其中,本申请所指的 CSS 文件是模块中以 .css 为后缀的文件；JS 文件是以 .js 后缀的文件。

[0065] 在每个模块中的 CSS (Cascading Style Sheet, 层叠样式表单) 文件,主要用来定义一个 HTML 或一类 HTML 标记的显示样式。JS 文件为用 javascript 脚本语言编写的、用于执行模块主要功能的软件程序。例如,一个超链接字符串；在实现这个超链接字符串的过程中, CSS 文件主要显示字符串的字体 (宋体、楷体或仿宋体等), JS 文件的作用为字符串添加超链接。

[0066] 如图 1 所示,本申请提供了一种网页加载系统,所述系统包括：

[0067] 服务器 100 和浏览器 200；服务器 100 和浏览器 200 之间采用无线方式通信。

[0068] 浏览器 200 即为安装在终端上实现上网冲浪的客户端,用户可以使用浏览器 200 查询自己所需的信息。用户可以在浏览器 200 的地址栏中输入网页地址,以便浏览器 200 将该网页地址发送至服务器 100,请求获取该网页地址对应的网页资源。

[0069] 服务器 100 可以接收浏览器 200 发送的网页地址,并抓取该网页地址对应的网页资源,并将网页资源发送至浏览器 200,以供浏览器 200 加载网页资源,从而可以显示网页资源至用户。

[0070] 如图 2 所示,本申请提供了一种网页加载方法,应用于图 1 所示的服务器。所述方法包括以下步骤：

[0071] 步骤 S201 :接收浏览器发送的网页地址。

[0072] 服务器实时接收浏览器发送的网页地址,网页地址可以表示用户希望获取的网页资源。例如,用户输入 www.taobao.com,则表示用户现在希望查看淘宝网的首屏页面。

[0073] 步骤 S202 :计算所述网页地址所需的网页资源;所述网页资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后。

[0074] 网页资源中可以包含有多个模块资源,各个模块资源的 JS 文件会阻碍首屏页面的渲染。为了解决这个问题,服务器在构建网页资源时,在确定主体资源之后,将主体资源中每个模块资源的 JS 文件提取出来,放置在主体资源之后;以便浏览器在加载网页资源时,可以先加载主体资源,在主体资源加载完成之后,再加载每个模块资源的 JS 文件。

[0075] 步骤 S203 :将所述网页资源发送至所述浏览器。

[0076] 服务器将按步骤 S202 计算得到的网页资源发送至浏览器,供浏览器进行加载并显示至用户。

[0077] 图 2 所示的实施例为服务器端执行的过程,下面介绍浏览器端执行的过程。如图 3 所示,本申请提供一种网页加载方法,应用于图 1 所示的浏览器。所述方法具体包括以下步骤:

[0078] 步骤 S301 :加载由服务器发送的网页资源;其中,所述网页资源包括头部资源、主体资源和尾部资源,所述主体资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后。

[0079] 浏览器可以接收服务器发送的网页资源,网页资源的主体资源中除了与模块对应的模块资源之外还有其它资源;模块资源穿插在其它资源中间,所有资源均按照由上至下的方式排列。并且,各个模块资源的 JS 文件,设置在主体资源之后。

[0080] 浏览器在加载网页资源时,采用由上至下的加载方式。即,先加载头部资源、再加载主体资源、再加载排列在主体资源之后的各个模块资源的 JS 文件,最后加载尾部资源。被加载资源可以显示在网页上,未被加载的资源在网页上显示空白网页。

[0081] 以主体资源为例,如图 4 所示,主体资源包括:模块资源 A、模块资源 B 和模块资源 C,其它资源包括资源 1、资源 2 和资源 3;模块资源 A、模块资源 B 和模块资源 C 的 JS 文件分别为 JS 文件 A、JS 文件 B 和 JS 文件 C。由于主体资源中资源 1 排列靠前,则首先加载资源 1,资源 1 在加载完成后,则会显示在网页上。其它未加载的内容,则未显示在网页上。

[0082] 在浏览器加载主体资源的过程中,当浏览器加载到模块资源时则执行步骤 S302。在步骤 S302 执行完毕之后,继续按顺序加载未加载的资源,直到主体资源被加载完毕为止,然后进入步骤 S303。

[0083] 例如,沿用上述举例,浏览器在加载资源 1 之后,则会加载模块 A,此时则执行步骤 S302 的相关内容。在步骤 S302 执行完毕之后,继续加载资源 2,在网页上显示资源 2 对应的内容。资源 2 下方为模块资源 B,则重复执行步骤 S302,在步骤 S302 执行结束之后,继续加载模块资源 C,重复执行步骤 S302;最后加载资源 3 从而完成主体资源的加载,进入步骤 S303。

[0084] 步骤 S302 :在所述网页资源中加载到每个模块资源时,在所述每个模块资源中提取运行 JS 文件所需的 JS 资源,并将所述 JS 资源注册到预先建立的函数运行池中。

[0085] 本步骤针对每个模块的执行过程均为一致的,所以仅对一个模块的执行过程进行

详细说明,其它模块的执行过程不再赘述。

[0086] 每个模块对应的模块资源可以分为两部分:一部分资源为后续运行 JS 文件所需的资源,这部分资源可以称为 JS 资源;一部分为运行 JS 文件不需要使用的资源,这部分资源称为其它资源。

[0087] 由于每个模块资源的 JS 文件设置在主体资源之后,所以在加载主体资源时,不执行各个模块资源的 JS 文件。但是,运行各个模块 JS 文件还是存储在各个模块对应的模块资源中。为了在主体资源加载完毕之后,可以运行模块资源的 JS 文件;所以,在浏览器加载到模块时,在模块对应的模块资源中提取运行 JS 文件所需要的 JS 资源;并将 JS 资源注册到预先建立的函数运行池中。其中,函数运行池为浏览器预先建立的,该函数运行池可以是对象、数组等各种可以存储函数变量的数据结构。

[0088] 例如,在浏览器加载到需要运行的模块时,该模块向函数运行池中注册该模块需要运行的函数名、参数、上下文等。比如,网页中有模块 A 和模块 B 两个模块。当浏览器加载到模块 A 和模块 B 时,该模块 A 和模块 B 依照网页加载的顺序依次提取 JS 参数,JS 参数可以为 JS 文件的函数名称、参数、上下文和其他参数等。

[0089]

JS 文件函数名	参数	上下文	其他参数
functionA	paramsA	contextA	otherA
functionB	paramsB	contextB	otherB

[0090] 由于每个模块均需要本步骤,所以,在加载网页资源的过程中,每个模块的 JS 资源均被注册到函数运行池中,从而使得函数运行池包括每个模块的 JS 资源。

[0091] 由于模块的 JS 资源已注册到函数运行池中,加载该模块时,不需要运行 JS 资源,仅需加载该模块对应模块资源中的其它资源(运行 JS 文件不需要的资源)。这部分资源可以是 CSS 文件对应的资源,在加载 CSS 文件对应资源之后,可以显示出模块的界面,但是具体功能无法使用,需等待运行 JS 文件的 JS 资源之后,才能够完善模块的功能。

[0092] 例如,一个超链接字符串;在实现这个超链接字符串的过程中,CSS 文件主要显示字符串的字体(宋体、楷体或仿宋体等),JS 文件的作用为字符串添加超链接。如果仅运行 CSS 文件,则仅显示规定字体的字符串,而字符串还不具有超链接功能。即本申请的技术效果为先显示出字符串,以便让用户看到有一个直观感受,再为字符串添加超链接,以完善字符串的具体功能。

[0093] 步骤 S303:在确定所述主体资源加载完成后,将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中。

[0094] 在所述主体资源加载完成后,触发一个加载完成事件;当浏览器检测到所述加载完成事件后,确定所述主体资源加载完成。如果检测到主体资源加载完毕,则说明首屏页面已经展现完毕。此时可以将函数运行池中每个模块的 JS 资源,注入到每个 JS 文件中,那么每个 JS 文件则可以加载已注入的模块资源。

[0095] 即,向真正的 JS 文件注入 JS 资源,使原本应当在网页中间执行的业务逻辑,在首屏页面加载完毕之后再执行。此时,运行各个模块资源的 JS 文件时,虽然依旧会阻塞浏览

器的渲染,但是由于网页的首屏页面(即网页的主要可见内容)已经渲染完毕,所以,可以方便用户对网页整体内容的感知。而 JS 文件延后执行,即完善模块具体功能的操作(执行人机交互操作的功能)延后执行;从而可以提高首屏页面的展现效率。

[0096] 本申请在浏览器加载到模块时,为了保证首屏页面的展现效率,暂时不执行模块内部的 JS 文件。而是,将运行 JS 文件所需的模块资源注册到预先建立的函数运行池中,以便在主体资源加载完毕之后,再统一运行各个模块资源的 JS 文件。

[0097] 换言之,本申请将原本在加载主体资源过程中执行的程序,延后到主体资源加载完毕之后再执行。将主体资源中原本在模块资源之后运行的资源,提前到模块资源的 JS 文件之前运行。由于主体资源包含网页的主要显示内容,在主体资源加载完毕之后,网页的首屏页面也就展现完毕。本申请由于延后模块资源的 JS 文件的加载过程,提前主体资源的加载过程,所以本申请可以提高网页的首屏展现效率。

[0098] 可以理解的是,在函数运行池中的 JS 资源被 JS 模块加载完毕之后,可以清空函数运行池,以便保证之前加载过的模块资源不被重复运行。

[0099] 从以上内容可以看出本申请具有以下有益效果:

[0100] 本申请将原本设置每个模块资源内部的 JS 文件,改为设置在网页资源的主体资源之后。由于网页资源采用由上至下的加载方式,所以,浏览器在加载网页资源的过程为,先加载头部资源,再加载主体资源,在主体资源加载完成之后,再运行每个模块内部的 JS 文件,最后加载尾部资源。

[0101] 由于主体资源包含首屏页面的主要显示内容,因此在主体资源加载完毕之后,即可显示出网页的首屏页面。然后,浏览器在再加载各个模块资源的 JS 文件,以完善各个模块的具体功能。

[0102] 即,本申请将原本在加载主体资源过程中需要执行的业务逻辑,设置在主体资源加载完成之后再执行;即提前执行主体资源显示首屏页面的程序,延后执行各个模块运行 JS 文件的程序,所以本申请可以提高首屏页面展现效率。并且,本申请没有增加额外的资源请求开销,便可加速首屏页面的展现,从而可以提高网页性能。

[0103] 为了更加清楚表示本申请和现有技术区别,下面介绍现有技术和本申请中网页资源的示意图。以主体资源中包括模块资源 A 和模块资源 B 为例,参见图 5 为目前主体资源的示意图,从图示中可以看出主体资源包含一个公共 CSS 文件、框架 JS 文件以及模块资源 A 和模块资源 B。

[0104] 以模块资源 A 为例,模块资源 A 中包含模块资源 A 自身的 CSS 文件 A、JS 文件 A、模块资源 1(运行 JS 文件所不需要的其它资源)以及 JS 资源 A(运行 JS 文件所需要的资源)。模块资源 B 的构造与模块资源 A 一样,在此不再赘述。在图 5 所示的实施例中,开发人员仅需要关注模块资源 A 和模块资源 B 即可,从而可以提高开发效率。

[0105] 如图 6 所示,为本申请中主体资源的示意图。从图示中可以看出主体资源包含一个公共 CSS 文件、框架 JS 文件以及模块资源 A 和模块资源 B。模块资源 A 和模块资源 B 中的 JS 文件 A 和 JS 文件 B 设置在主体资源之后,以便可以在执行完成主体资源之后,再运行模块资源 A 的 JS 文件 A 和模块资源 B 的 JS 文件 B。

[0106] 从图 5 和图 6 可以看出:本申请将原本设置于模块资源内部的 JS 文件放置于主体资源之后,从而实现提高首屏页面展现效率的目的。

- [0107] 与图 2 所示的实施例相对应,如图 7 所示,本申请提供一种网页加载装置,包括:
- [0108] 第二接收单元 71,用于接收浏览器发送的网页地址;
- [0109] 计算单元 72,用于计算所述网页地址所需的网页资源;所述网页资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;
- [0110] 第二发送单元 73,用于将所述网页资源发送至所述浏览器。
- [0111] 与图 3 所示的实施例相对应,如图 8 所示,本申请提供一种网页加载装置,包括:
- [0112] 加载单元 81,用于加载由服务器发送的网页资源;其中,所述网页资源包括头部资源、主体资源和尾部资源,所述主体资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;
- [0113] 注册单元 82,用于在所述网页资源中加载到每个模块资源时,在所述模块资源中提取运行 JS 文件所需的 JS 资源,并将所述 JS 资源注册到预先建立的函数运行池中;
- [0114] 确定单元 83,用于在确定所述主体资源加载完成;
- [0115] 其中,所述确定单元 83 具体用于:在所述主体资源加载完成后,触发一个加载完成事件;在检测到所述加载完成事件后,确定所述主体资源加载完成。
- [0116] 注入单元 84,用于在所述主体资源加载完成后,将所述函数运行池中每个模块的 JS 资源注入到与之对应的 JS 文件中。
- [0117] 如图 9 所示,本申请提供网页加载装置,还包括:
- [0118] 清空单元 91,用于在将所述函数运行池中每个模块资源的 JS 资源注入到与之对应的 JS 文件中之后,清空所述函数运行池。
- [0119] 第一接收单元 92,用于接收用户输入的网页地址;其中,所述网页地址与所述网页资源对应;
- [0120] 第一发送单元 93,用于向所述服务器发送所述网页地址。
- [0121] 从以上内容可以看出本申请具有以下有益效果:
- [0122] 本申请应用于浏览器加载具有模块的网页资源的过程,在模块资源中 JS 文件的运行效率较慢,所以本申请将原本设置每个模块内部的 JS 文件,改为设置在网页资源的主体资源之后。由于网页资源采用由上至下的加载方式,所以,浏览器在加载网页资源的过程为,在主体资源加载完成之后,再运行每个模块内部的 JS 文件。在网页中主体资源对应网页的首屏页面;所以本申请可以达到的技术效果为:在显示网页的首屏页面之后(在主体资源加载完成之后),再运行各个模块资源的 JS 文件(完善各个模块的具体功能)。
- [0123] 本申请将原本在加载主体资源过程中需要执行的业务逻辑,设置在主体资源加载完成之后再执行;即提前执行主体资源显示首屏页面的程序,延后执行各个模块运行 JS 文件的程序,所以本申请可以提高首屏页面展现效率。并且,本申请没有增加额外的资源请求开销,便可加速首屏页面展现,从而可以提高网页性能。
- [0124] 参见图 1,本申请还提供了一种网页加载系统,包括:
- [0125] 浏览器 200 和服务器 100;
- [0126] 其中,所述浏览器 200,用于加载由服务器 100 发送的网页资源;其中,所述网页资源包括头部资源、主体资源和尾部资源,所述主体资源包括至少一个模块资源,并且,每个模块资源的 JS 文件均设置于所述主体资源之后;在所述网页资源中加载到每个模块资源时,在所述每个模块资源中提取运行 JS 文件所需的 JS 资源,并将所述 JS 资源注册到预先

建立的函数运行池中；在确定所述主体资源加载完成后，将所述函数运行池中每个模块的 JS 资源注入到与之对应的 JS 文件中。

[0127] 所述服务器 100，用于接收浏览器 200 发送的网页地址；计算所述网页地址所需的网页资源；所述网页资源包括至少一个模块资源，并且，每个模块资源的 JS 文件均设置于所述主体资源之后；将所述网页资源发送至所述浏览器 200。

[0128] 本实施例方法所述的功能如果以软件功能单元的形式实现并作为独立的产品销售或使用，可以存储在一个计算设备可读取存储介质中。基于这样的理解，本申请实施例对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来，该软件产品存储在一个存储介质中，包括若干指令用以使得一台计算设备（可以是个人计算机，服务器，移动计算设备或者网络设备）执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（ROM, Read-Only Memory）、随机存取存储器（RAM, Random Access Memory）、磁碟或者光盘等各种可以存储程序代码的介质。

[0129] 本说明书中各个实施例采用递进的方式描述，每个实施例重点说明的都是与其它实施例的不同之处，各个实施例之间相同或相似部分互相参见即可。

[0130] 对所公开的实施例的上述说明，使本领域专业技术人员能够实现或使用本申请。对这些实施例的多种修改对本领域的专业技术人员来说将是显而易见的，本文中所定义的一般原理可以在不脱离本申请的精神或范围的情况下，在其它实施例中实现。因此，本申请将不会被限制于本文所示的这些实施例，而是要符合与本文所公开的原理和新颖特点相一致的最宽的范围。

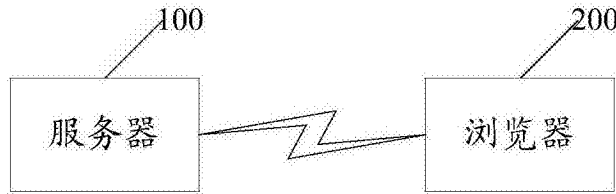


图 1

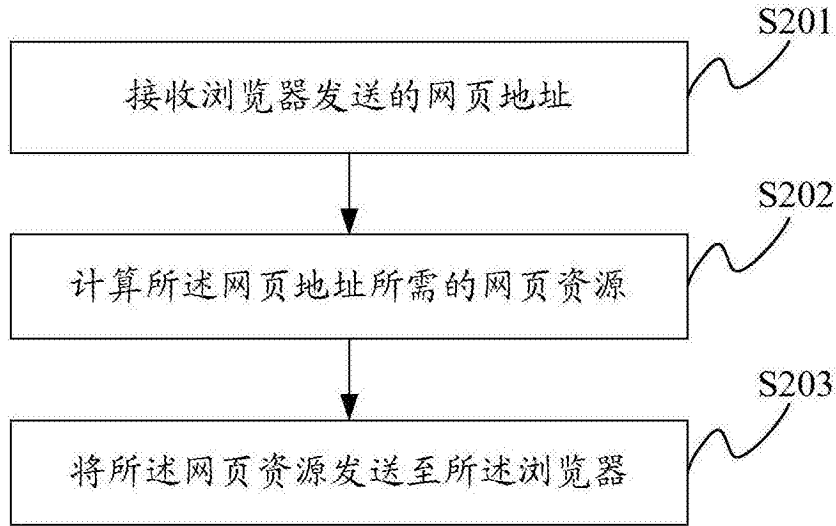


图 2

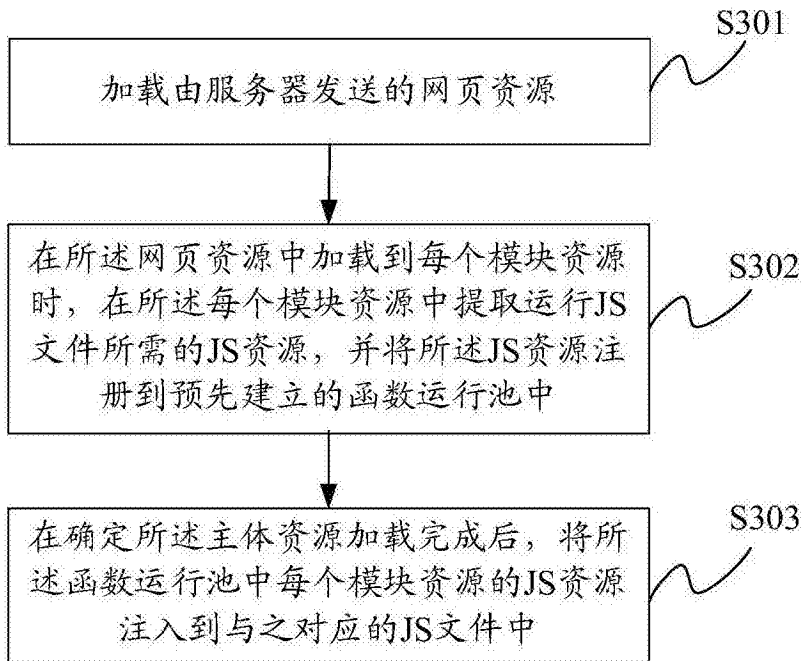


图 3



图 4



图 5

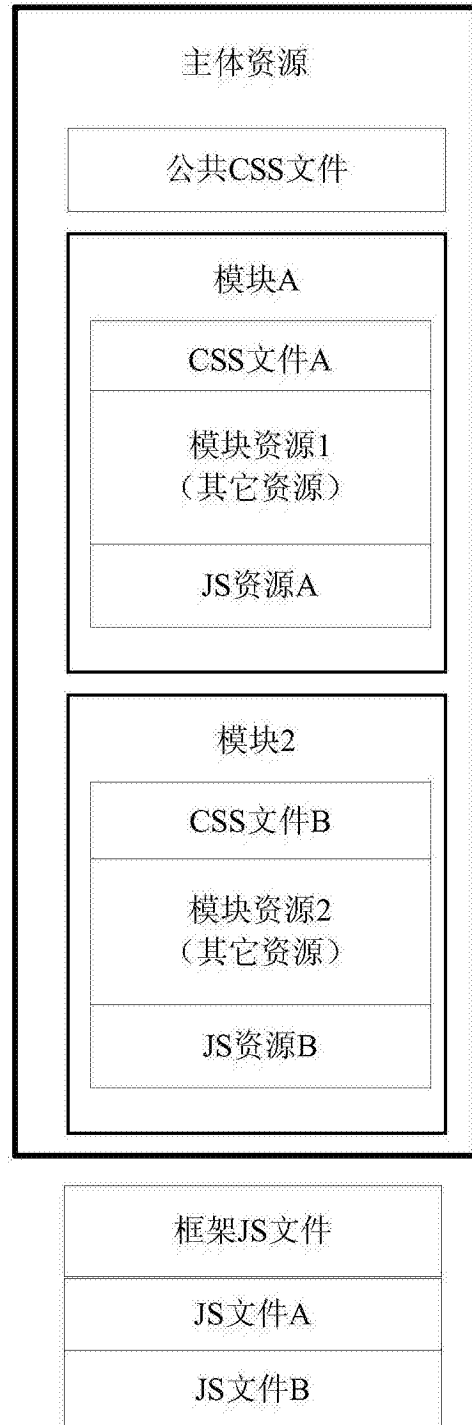


图 6

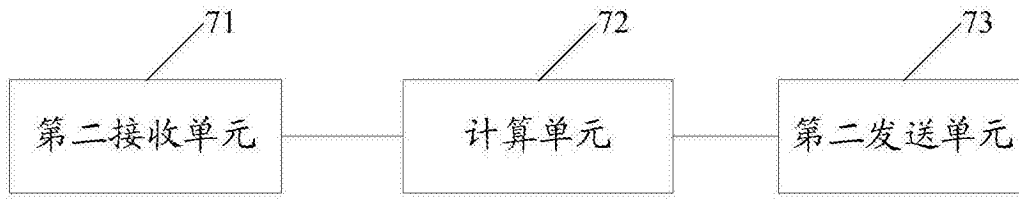


图 7

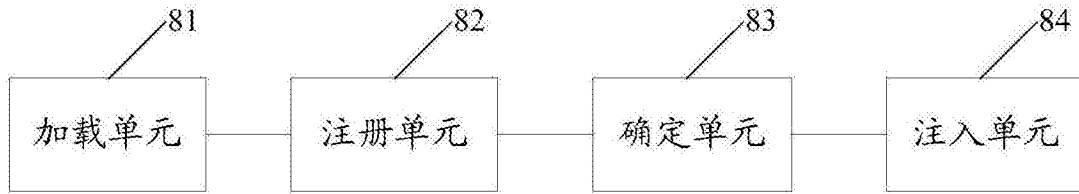


图 8

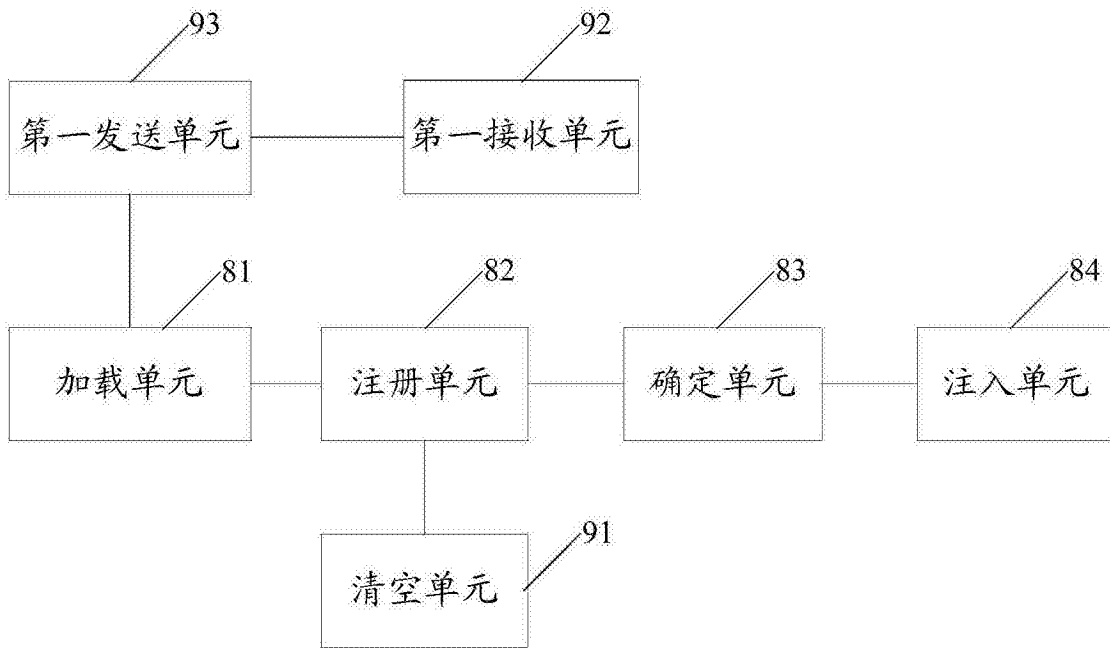


图 9