



(12)发明专利申请

(10)申请公布号 CN 107346309 A

(43)申请公布日 2017. 11. 14

(21)申请号 201610290676.8

(22)申请日 2016.05.04

(71)申请人 北京京东尚科信息技术有限公司
地址 100080 北京市海淀区杏石口路65号
西杉创意园西区11C楼东段1-4层西段
1-4层
申请人 北京京东世纪贸易有限公司

(72)发明人 陈福洪 彭定邦

(74)专利代理机构 北京英赛嘉华知识产权代理
有限责任公司 11204
代理人 王达佐 马晓亚

(51)Int. Cl.
G06F 17/30(2006.01)

权利要求书2页 说明书8页 附图5页

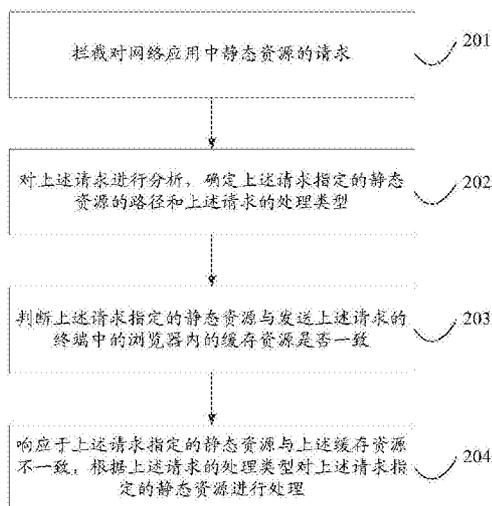
(54)发明名称

一种网络应用中静态资源的处理方法及装置

(57)摘要

本申请公开了网络应用中静态资源的处理方法及装置。所述方法的一具体实施方式包括：拦截对网络应用中静态资源的请求；对所述请求进行分析，确定所述请求指定的静态资源的路径和所述请求的处理类型；判断所述请求指定的静态资源与发送所述请求的终端中的浏览器内的缓存资源是否一致；响应于所述请求指定的静态资源与上述缓存资源不一致，根据所述请求的处理类型对所述请求指定的静态资源进行处理。该实施方式能够提高对网络应用中静态资源的处理效率。

200



1. 一种网络应用中静态资源的处理方法,其特征在于,所述方法包括:
拦截对网络应用中静态资源的请求;
对所述请求进行分析,确定所述请求指定的静态资源的路径和所述请求的处理类型;
判断所述请求指定的静态资源与发送所述请求的终端中的浏览器内的缓存资源是否一致;
响应于所述请求指定的静态资源与所述缓存资源不一致,根据所述请求的处理类型对所述请求指定的静态资源进行处理。
2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:
将处理后的静态资源返回给所述终端。
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:
响应于所述请求指定的静态资源与所述缓存资源一致,将所述缓存资源返回给所述终端。
4. 根据权利要求1所述的方法,其特征在于,所述判断所述请求指定的静态资源与发送所述请求的终端中的浏览器内的缓存资源是否一致,包括:
提取所述请求指定的静态资源的第一时间戳,所述第一时间戳为所述请求指定的静态资源的最近修改时间;
提取所述发送所述请求的终端中的浏览器内的缓存资源的第二时间戳,所述第二时间戳为所述缓存资源的修改时间;
在所述第一时间戳与所述第二时间戳一致时,确定所述请求指定的静态资源与所述缓存资源一致;
在所述第一时间戳与所述第二时间戳不一致时,确定所述请求指定的静态资源与所述缓存资源不一致。
5. 根据权利要求4所述的方法,其特征在于,所述方法还包括:
将所述处理后的静态资源的最近修改时间返回给所述终端,以供所述终端将所述处理后的静态资源的最近修改时间作为更新的第二时间戳。
6. 根据权利要求1-5任一项所述的方法,其特征在于,所述请求的处理类型至少包括以下中的一项:压缩处理、合并处理。
7. 一种网络应用中静态资源的处理装置,其特征在于,所述装置包括:
拦截单元,用于拦截对网络应用中静态资源的请求;
分析单元,用于对所述请求进行分析,确定所述请求指定的静态资源的路径和所述请求的处理类型;
判断单元,用于判断所述请求指定的静态资源与发送所述请求的终端中的浏览器内的缓存资源是否一致;
处理单元,用于响应于所述请求指定的静态资源与所述缓存资源不一致,根据所述请求的处理类型对所述请求指定的静态资源进行处理。
8. 根据权利要求7所述的装置,其特征在于,所述装置还包括:
第一返回单元,用于将处理后的静态资源返回给所述终端。
9. 根据权利要求7所述的装置,其特征在于,所述装置还包括:
第二返回单元,用于响应于所述请求指定的静态资源与所述缓存资源一致,将所述缓

存资源返回给所述终端。

10. 根据权利要求7所述的装置,其特征在于,所述判断单元包括:

第一提取模块,用于提取所述请求指定的静态资源的第一时间戳,所述第一时间戳为所述请求指定的静态资源的最近修改时间;

第二提取模块,用于提取所述发送所述请求的终端中的浏览器内的缓存资源的时间戳,所述第二时间戳为所述缓存资源的修改时间;

第一确定模块,用于在所述第一时间戳与所述第二时间戳一致时,确定所述请求指定的静态资源与所述缓存资源一致;

第二确定模块,用于在所述第一时间戳与所述第二时间戳不一致时,确定所述请求指定的静态资源与所述缓存资源不一致。

11. 根据权利要求10所述的装置,其特征在于,所述装置还包括:

返回模块,用于将所述处理后的静态资源的最近修改时间返回给所述终端,以供所述终端将所述处理后的静态资源的最近修改时间作为更新的第二时间戳。

12. 根据权利要求7-11任一项所述的方法,其特征在于,所述请求的处理类型至少包括以下中的一项:压缩处理、合并处理。

一种网络应用中静态资源的处理方法及装置

技术领域

[0001] 本申请涉及计算机技术领域,具体涉及网络技术领域,尤其涉及一种网络应用中静态资源的处理方法及装置。

背景技术

[0002] 目前,很多需要前端页面的应用程序都需要JS(Javascript,一种脚本语言)和CSS(Cascading Style Sheets,层叠样式表)的支撑,从而实现良好的交互体验与视觉展现。网络应用中的静态资源就包括前端页面上加载的脚本和样式文件,对静态资源进行压缩能够有效地减少页面呈现所需的时间。

[0003] 现有的很多文本压缩工具(如YUI compressor)都可以实现对JS和CSS的压缩,但这些压缩方式需要人工线下对压缩文件进行转化和合并后替换原始压缩文件放入工程中或者嵌入请求响应代码才能实现,存在不方便项目的开发调试和运行的技术问题。

发明内容

[0004] 本申请的目的在于提供一种网络应用中静态资源的处理方法及装置,来解决以上背景技术部分提到的技术问题。

[0005] 第一方面,本申请提供了一种网络应用中静态资源的处理方法,所述方法包括:拦截对网络应用中静态资源的请求;对所述请求进行分析,确定所述请求指定的静态资源的路径和所述请求的处理类型;判断所述请求指定的静态资源与发送所述请求的终端中的浏览器内的缓存资源是否一致;响应于所述请求指定的静态资源与所述缓存资源不一致,根据所述请求的处理类型对所述请求指定的静态资源进行处理。

[0006] 在一些实施例中,所述方法还包括:将处理后的静态资源返回给所述终端。

[0007] 在一些实施例中,所述方法还包括:响应于所述请求指定的静态资源与所述缓存资源一致,将所述缓存资源返回给所述终端。

[0008] 在一些实施例中,所述判断所述请求指定的静态资源与发送所述请求的终端中的浏览器内的缓存资源是否一致,包括:提取所述请求指定的静态资源的第一时间戳,所述第一时间戳为所述请求指定的静态资源的最近修改时间;提取所述发送所述请求的终端中的浏览器内的缓存资源的第二时间戳,所述第二时间戳为所述缓存资源的修改时间;在所述第一时间戳与所述第二时间戳一致时,确定所述请求指定的静态资源与所述缓存资源一致;在所述第一时间戳与所述第二时间戳不一致时,确定所述请求指定的静态资源与所述缓存资源不一致。

[0009] 在一些实施例中,所述方法还包括:将所述处理后的静态资源的最近修改时间返回给所述终端,以供所述终端将所述处理后的静态资源的最近修改时间作为更新的第二时间戳。

[0010] 在一些实施例中,所述请求的处理类型至少包括以下中的一项:压缩处理、合并处理。

[0011] 第二方面,本申请提供一种网络应用中静态资源的处理装置,所述装置包括:拦截单元,用于拦截对网络应用中静态资源的请求;分析单元,用于对所述请求进行分析,确定所述请求指定的静态资源的路径和所述请求的处理类型;判断单元,用于判断所述请求指定的静态资源与发送所述请求的终端中的浏览器内的缓存资源是否一致;处理单元,用于响应于所述请求指定的静态资源与所述缓存资源不一致,根据所述请求的处理类型对所述请求指定的静态资源进行处理。

[0012] 在一些实施例中,所述装置还包括:第一返回单元,用于将处理后的静态资源返回给所述终端。

[0013] 在一些实施例中,所述装置还包括:第二返回单元,用于响应于所述请求指定的静态资源与所述缓存资源一致,将所述缓存资源返回给所述终端。

[0014] 在一些实施例中,所述判断单元包括:第一提取模块,用于提取所述请求指定的静态资源的第一时间戳,所述第一时间戳为所述请求指定的静态资源的最近修改时间;第二提取模块,用于提取所述发送所述请求的终端中的浏览器内的缓存资源的第二时间戳,所述第二时间戳为所述缓存资源的修改时间;第一确定模块,用于在所述第一时间戳与所述第二时间戳一致时,确定所述请求指定的静态资源与所述缓存资源一致;第二确定模块,用于在所述第一时间戳与所述第二时间戳不一致时,确定所述请求指定的静态资源与所述缓存资源不一致。

[0015] 在一些实施例中,所述装置还包括:返回模块,用于将所述处理后的静态资源的最近修改时间返回给所述终端,以供所述终端将所述处理后的静态资源的最近修改时间作为更新的第二时间戳。

[0016] 在一些实施例中,所述请求的处理类型至少包括以下中的一项:压缩处理、合并处理。

[0017] 本申请提供的网络应用中静态资源的处理方法及装置,通过拦截对静态资源的请求,并在缓存资源不可用时,对静态资源进行相应的处理,能够提高对网络应用中静态资源的处理效率,不需人工线下对压缩文件进行转换,方便项目的开发调试和运行。

附图说明

[0018] 通过阅读参照以下附图所作的对非限制性实施例所作的详细描述,本申请的其它特征、目的和优点将会变得更明显:

[0019] 图1是本申请可以应用于其中的示例性系统架构图;

[0020] 图2是根据本申请的网络应用中静态资源的处理方法的一个实施例的流程图;

[0021] 图3是根据本申请的网络应用中静态资源的处理方法的又一个实施例的流程图;

[0022] 图4是根据本申请的网络应用中静态资源的处理装置的一个实施例的结构示意图;

[0023] 图5是适于用来实现本申请实施例的终端设备或服务器的计算机系统的结构示意图。

具体实施方式

[0024] 下面结合附图和实施例对本申请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相关发明,而非对该发明的限定。另外还需要说明的是,为了便于描述,附图中仅示出了与有关发明相关的部分。

[0025] 需要说明的是,在不冲突的情况下,本申请中的实施例及实施例中的特征可以相互组合。下面将参考附图并结合实施例来详细说明本申请。

[0026] 图1示出了可以应用本申请的网络应用中静态资源的处理方法或网络应用中静态资源的处理装置的实施例的示例性系统架构100。

[0027] 如图1所示,系统架构100可以包括终端设备101、102、103,网络104和服务器105。网络104用以在终端设备101、102、103和服务器105之间提供通信链路的介质。网络104可以包括各种连接类型,例如有线、无线通信链路或者光纤电缆等等。

[0028] 用户可以使用终端设备101、102、103通过网络104与服务器105交互,以接收或发送消息等。终端设备101、102、103上可以安装有各种通讯客户端应用,例如网页浏览器应用等。

[0029] 终端设备101、102、103可以是具有显示屏的各种电子设备,包括但不限于智能手机、平板电脑、电子书阅读器、膝上型便携计算机和台式计算机等等。

[0030] 服务器105可以是提供各种服务的服务器,例如对终端设备101、102、103上显示的内容提供支持的静态资源处理服务器。静态资源处理服务器可以对接收到的静态资源请求等数据进行分析等处理,并将处理结果反馈给终端设备。

[0031] 需要说明的是,本申请实施例所提供的网络应用中静态资源的处理方法一般由服务器105执行,相应地,网络应用中静态资源的处理装置一般设置于服务器105中。

[0032] 应该理解,图1中的终端设备、网络和服务器的数目仅仅是示意性的。根据实现需要,可以具有任意数目的终端设备、网络和服务器。

[0033] 继续参考图2,示出了根据本申请的网络应用中静态资源的处理方法的一个实施例的流程图200。本实施例的网络应用中静态资源的处理方法,包括以下步骤:

[0034] 步骤201,拦截对网络应用中静态资源的请求。

[0035] 在本实施例中,网络应用中静态资源的处理方法运行于其上的电子设备(例如图1所示的服务器105)可以通过设置拦截器来拦截对网络应用中静态资源的请求。其中,拦截器的作用是拦截终端设备的请求并进行相应的处理。

[0036] 可以理解的是,上述请求中可以包括:待处理静态资源的标识、对这些待处理静态资源的处理类型、上述待处理静态资源的路径等。其中,待处理的静态资源可以是.js文件(Javascript代码文件),也可以是.cs文件(C#语言源代码文件)。

[0037] 步骤202,对上述请求进行分析,确定上述请求指定的静态资源的路径和上述请求的处理类型。

[0038] 对上述请求进行分析,可以解析出上述请求中包含的待处理静态资源的路径和上述请求的处理类型。

[0039] 在本实施例的一个可选的实现方式中,对网络应用中静态资源的请求URI(Uniform Resource Identifier,统一资源定位符)可以为URL(Uniform Resource Locator,统一资源定位器)格式,根据上述URL中携带的信息即可确定上述请求的处理类型。可以理解的是,对网络应用中静态资源的请求还可以是其它可实施的格式。

[0040] 在本实施例的一些可选的实现方式中,上述请求的处理类型至少包括以下中的一

项:压缩处理、合并处理。

[0041] 以静态资源a.js和b.js为例来说,a.js和b.js分别为名称为a和b的Javascript代码文件,压缩处理请求的URL如下:

[0042] http://ip:port/resource/compress?a.js,b.js;

[0043] 合并处理请求的URL如下:

[0044] http://ip:port/resource/merge?a.js,b.js。

[0045] 步骤203,判断上述请求指定的静态资源与发送上述请求的终端中的浏览器内的缓存资源是否一致。

[0046] 当发送上述请求的终端中的浏览器内的缓存资源可用时,可有效地节约带宽和时间,因此,在处理前要判断浏览器中的缓存资源是否可用,即判断上述请求指定的静态资源与浏览器中的缓存资源是否一致。

[0047] 步骤204,响应于上述请求指定的静态资源与上述缓存资源不一致,根据上述请求的处理类型对上述请求指定的静态资源进行处理。

[0048] 当发送上述请求的终端中的浏览器内的缓存资源不可用时,需要对上述请求指定的静态资源进行相应的处理,即当上述请求为压缩处理请求时,要对上述压缩处理请求指定的静态资源进行压缩;当上述请求为合并处理请求时,要对上述合并处理请求指定的静态资源进行合并。

[0049] 以步骤202中压缩处理请求的URL为例,其指定的静态资源为a.js和b.js,当浏览器中的缓存资源不可用时,服务器对a.js和b.js进行压缩处理。在实践中,对上述文件进行压缩时,采用正则表达式对文件中的注释、空白字符、换行等不影响代码逻辑的字符进行删除。对静态资源进行合并处理时,将多个文件合并成一个文件。对文件合并后,可有效地减少请求数目,从而节约网络带宽。

[0050] 本申请的上述实施例提供的网络应用中静态资源的处理方法,能够提高对网络应用中静态资源的处理效率,不需人工线下对压缩文件进行转换,方便项目的开发调试和运行。

[0051] 继续参考图3,示出了根据本申请的网络应用中静态资源的处理方法的又一个实施例的流程图300。本实施例的网络应用中静态资源的处理方法包括以下步骤:

[0052] 步骤301,拦截对网络应用中静态资源的请求。

[0053] 通过各种拦截方式拦截对网络应用中静态资源的请求。

[0054] 在一些可选的实现方式中,拦截器可以是SpringMVC拦截器。其中,SpringMVC是一种基于Java的轻量级Web框架,作用是简化Web开发。SpringMVC拦截器是可根据需求,通过代码设置对特定的请求进行拦截的拦截器,比如本申请中可以通过代码配置SpringMVC拦截器,使其拦截对网络应用中静态资源的请求。

[0055] 步骤302,确定上述请求指定的静态资源的路径。

[0056] 在本实施例的一个可选的实现方式中,对请求进行分析时,可通过在上述SpringMVC拦截器中分析ResourceHttpRequestHandler中的locations属性,得到上述请求指定的静态资源的路径。主要原理是利用SpringMVC中的<mvc:resources/>元素将对静态资源的访问请求转移到ResourceHttpRequestHandler中,从而可以通过对其分析来解析对静态资源的请求。其中,ResourceHttpRequestHandler是SpringMVC的一个资源路径请求句

柄,它的locations属性指定了静态资源的位置,即静态资源的路径。

[0057] 步骤303,判断浏览器中的缓存资源是否可用。

[0058] 可以理解的是,此处浏览器中的缓存资源是否可用是指服务器判断发送上述请求的终端中的浏览器内的缓存资源是否可用。

[0059] 在本实施例的一些可选的实现方式中,步骤303进一步包括300中未示出的以下子步骤:

[0060] 提取所述请求指定的静态资源的第一时间戳,所述第一时间戳为所述请求指定的静态资源的最近修改时间。

[0061] 如果静态资源经过多次修改,那么认定静态资源的最后一次修改的时间,即最近修改时间为第一时间戳。可以理解的是,当上述请求指定的静态资源有多个文件时,有的文件可能进行了修改,有的文件可能没有修改,那么此时静态资源的第一时间戳可以为所有文件中修改时间最近的时间,也可以是所有文件的最后一次修改时间的平均值。

[0062] 本实施例中,可将静态资源的最近修改时间作为第一时间戳。

[0063] 提取所述发送所述请求的终端中的浏览器内的缓存资源的第二时间戳,所述第二时间戳为所述缓存资源的修改时间。

[0064] 浏览器内的缓存资源是服务器在将终端请求的静态资源返回给终端时缓存在浏览器中的,因此,缓存资源不存在多次修改,其第二时间戳为缓存资源的修改时间。

[0065] 在所述第一时间戳与所述第二时间戳一致时,确定所述请求指定的静态资源与所述缓存资源一致。

[0066] 如果上述请求指定的静态资源的第一时间戳与缓存资源的第二时间戳相同,则说明在客户端此次向服务器发送对上述静态资源的请求前,上述静态资源都没有进行过修改,因此,终端设备中浏览器内的缓存资源与上述请求指定静态资源是一致的。此时,浏览器内的缓存资源是可用的。

[0067] 在所述第一时间戳与所述第二时间戳不一致时,确定所述请求指定的静态资源与所述缓存资源不一致。

[0068] 如果上述请求指定的静态资源的第一时间戳与缓存资源的第二时间戳不相同,则说明在客户端此次向服务器发送对上述静态资源的请求前,上述静态资源进行了一次或多次修改,修改后的静态资源与浏览器内的缓存资源有可能不同,因此,上述请求指定的静态资源与终端设备中的浏览器内的缓存资源不一致。此时,浏览器内的缓存资源是不可用的。

[0069] 当终端设备中的浏览器内的缓存资源不可用时,执行步骤304;当终端设备中的浏览器内的缓存资源可用时,执行步骤304'。

[0070] 步骤304,判断上述请求是否为压缩处理请求。

[0071] 本实施例中,认定终端设备对静态资源的处理只包括压缩处理和合并处理,因此,认定终端设备发送的对网络应用中静态资源的请求为压缩处理请求或合并处理请求。当上述请求不是压缩处理请求时,默认上述请求为合并处理请求。

[0072] 当上述请求为压缩处理请求时,执行步骤305;当上述请求不是压缩处理请求时,执行步骤305'。

[0073] 步骤305,压缩上述请求指定的静态资源。

[0074] 步骤305',合并上述请求指定的静态资源。

- [0075] 当对上述请求指定的静态资源处理完成后,执行步骤306。
- [0076] 步骤306,返回处理后的静态资源。
- [0077] 将处理后的静态资源返回给发送上述请求的终端设备。
- [0078] 步骤304',返回缓存资源。
- [0079] 当终端设备中浏览器内的缓存资源可用时,直接返回缓存资源,可有效节约带宽。
- [0080] 步骤307,结束。
- [0081] 在本实施例一些可选的实现方式中,上述方法还包括300中未示出的以下步骤:将所述处理后的静态资源的最近修改时间返回给所述终端,以供所述终端将所述处理后的静态资源的最近修改时间作为更新的第二时间戳。
- [0082] 及时更新缓存资源的第二时间戳,以便于下次请求能够高效的利用缓存,节约网络带宽。
- [0083] 本申请的上述实施例提供的网络应用中静态资源的处理方法,在高效利用缓存的同时,可实现对静态资源的实时压缩和合并,可节约服务器接收的TCP请求数,同时也节约带宽,并且对代码的调试方便,便于项目的运行。
- [0084] 继续参考图4,示出了根据本申请的网络应用中静态资源的处理装置的一个实施例的结构示意图。本实施例的网络应用中静态资源的处理装置包括:拦截单元401、分析单元402、判断单元403以及处理单元404。其中,拦截单元401,用于拦截对网络应用中静态资源的请求;分析单元402,用于对上述请求进行分析,确定上述请求指定的静态资源的路径和上述请求的处理类型;判断单元403,用于判断上述请求指定的静态资源与发送上述请求的终端中的浏览器内的缓存资源是否一致;处理单元404,用于响应于上述请求指定的静态资源与上述缓存资源不一致,根据上述请求的处理类型对上述请求指定的静态资源进行处理。
- [0085] 本实施例中,拦截单元401可通过各种拦截方式拦截对网络应用中静态资源的请求。例如可通过拦截器来拦截上述请求。
- [0086] 本实施例中,分析单元402对拦截单元拦截的请求进行分析,确定上述请求指定的静态资源的路径以及对上述静态资源的处理类型。可以理解的是,上述请求中可以包括:待处理静态资源的标识、上述待处理静态资源的路径、对上述待处理静态资源的处理类型等信息。
- [0087] 判断单元403根据分析单元402分析的待处理资源的标识及路径,判断发送上述请求的终端设备中的浏览器内的缓存资源是否可用,如果缓存资源可用,则直接将缓存资源返回给终端设备。
- [0088] 处理单元404在缓存资源不可用时,根据上述请求的处理类型对待处理静态资源进行处理。
- [0089] 在本实施例一些可选的实现方式中,上述装置还包括400中未示出的第一返回单元,用于将处理后的静态资源返回给终端。
- [0090] 第一返回单元在处理单元404对待处理静态资源处理后,将处理后的静态资源返回给发送上述请求的终端。
- [0091] 在本实施例一些可选的实现方式中,上述装置还包括400中未示出的第二返回单元,用于响应于上述请求指定的静态资源与缓存资源一致,将缓存资源返回给终端。

[0092] 在本实施例一些可选的实现方式中,上述判断单元进一步包括400中未示出的以下部分:第一提取模块,用于提取上述请求指定的静态资源的第一时间戳,第一时间戳为上述请求指定的静态资源的最近修改时间;第二提取模块,用于提取发送上述请求的终端中的浏览器内的缓存资源的第二时间戳,第二时间戳为缓存资源的修改时间;第一确定模块,用于在第一时间戳与第二时间戳一致时,确定上述请求指定的静态资源与缓存资源一致;第二确定模块,用于在第一时间戳与第二时间戳不一致时,确定上述请求指定的静态资源与缓存资源不一致。

[0093] 判断单元403通过比较缓存资源的第二时间戳和上述请求指定的静态资源的第一时间戳,来判断缓存资源是否与上述请求指定的静态资源一致。在一致的情况下,表明缓存资源是可用的;在不一致的情况下,表明缓存资源是不可用的。

[0094] 在本实施例一些可选的实现方式中,上述装置还包括400中未示出的返回模块,用于将处理后的静态资源的最近修改时间返回给终端,以供终端将所述处理后的静态资源的最近修改时间作为更新的第二时间戳。

[0095] 及时更新缓存资源的第二时间戳,有利于充分利用缓存资源,从而实现节省带宽和流量。

[0096] 在本实施例一些可选的实现方式中,上述请求的处理类型至少包括以下中的一项:压缩处理、合并处理。

[0097] 本申请的上述实施例提供的网络应用中静态资源的处理装置,能够提高对网络应用中静态资源的处理效率,不需人工线下对压缩文件进行转换,方便项目的开发调试和运行。

[0098] 下面参考图5,其示出了适于用来实现本申请实施例的服务器的计算机系统500的结构示意图。

[0099] 如图5所示,计算机系统500包括中央处理单元(CPU)501,其可以根据存储在只读存储器(ROM)502中的程序或者从存储部分508加载到随机访问存储器(RAM)503中的程序而执行各种适当的动作和处理。在RAM 503中,还存储有系统500操作所需的各种程序和数据。CPU 501、ROM 502以及RAM 503通过总线504彼此相连。输入/输出(I/O)接口505也连接至总线504。

[0100] 以下部件连接至I/O接口505:包括键盘、鼠标等的输入部分506;包括诸如阴极射线管(CRT)、液晶显示器(LCD)等以及扬声器等的输出部分507;包括硬盘等的存储部分508;以及包括诸如LAN卡、调制解调器等的网络接口卡的通信部分509。通信部分509经由诸如因特网的网络执行通信处理。驱动器510也根据需要连接至I/O接口505。可拆卸介质511,诸如磁盘、光盘、磁光盘、半导体存储器等等,根据需要安装在驱动器510上,以便于从其上读出的计算机程序根据需要被安装入存储部分508。

[0101] 特别地,根据本申请公开的实施例,上文参考流程图描述的过程可以被实现为计算机软件程序。例如,本申请公开的实施例包括一种计算机程序产品,其包括有形地包含在机器可读介质上的计算机程序,所述计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中,该计算机程序可以通过通信部分509从网络上被下载和安装,和/或从可拆卸介质511被安装。在该计算机程序被中央处理单元(CPU)501执行时,执行本申请的方法中限定的上述功能。

[0102] 附图中的流程图和框图,示出了按照本申请各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,所述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0103] 描述于本申请实施例中所涉及到的单元可以通过软件的方式实现,也可以通过硬件的方式来实现。所描述的单元也可以设置在处理器中,例如,可以描述为:一种网络应用中的静态资源处理装置包括拦截单元、分析单元、判断单元和处理单元。其中,这些单元的名称在某种情况下并不构成对该单元本身的限定,例如,拦截单元还可以被描述为“拦截对网络应用中静态资源的请求的单元”。

[0104] 作为另一方面,本申请还提供了一种非易失性计算机存储介质,该非易失性计算机存储介质可以是上述实施例中所述装置中所包含的非易失性计算机存储介质;也可以是单独存在,未装配入终端中的非易失性计算机存储介质。上述非易失性计算机存储介质存储有一个或者多个程序,当所述一个或者多个程序被一个设备执行时,使得所述设备:拦截对网络应用中静态资源的请求;对所述请求进行分析,确定所述请求指定的静态资源的路径和所述请求的处理类型;判断所述请求指定的静态资源与发送所述请求的终端中的浏览器内的缓存资源是否一致;响应于所述请求指定的静态资源与所述缓存资源不一致,根据所述请求的处理类型对所述请求指定的静态资源进行处理。

[0105] 以上描述仅为本申请的较佳实施例以及对所运用技术原理的说明。本领域技术人员应当理解,本申请中所涉及的发明范围,并不限于上述技术特征的特定组合而成的技术方案,同时也应涵盖在不脱离所述发明构思的情况下,由上述技术特征或其等同特征进行任意组合而形成的其它技术方案。例如上述特征与本申请中公开的(但不限于)具有类似功能的技术特征进行互相替换而形成的技术方案。

100

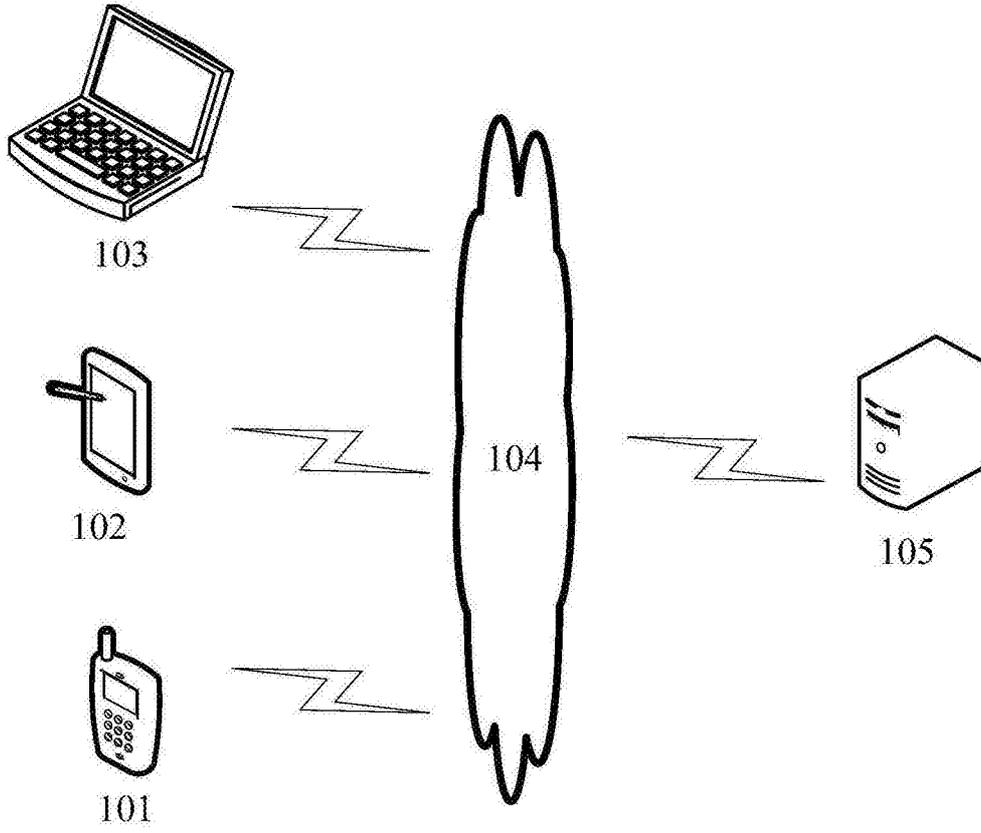


图1

200

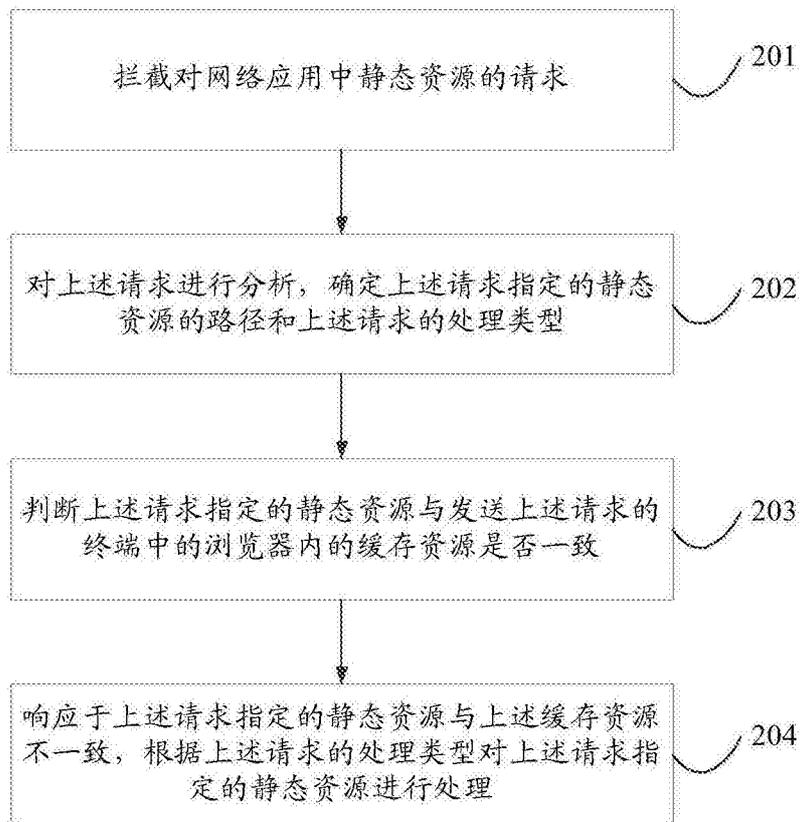


图2

300

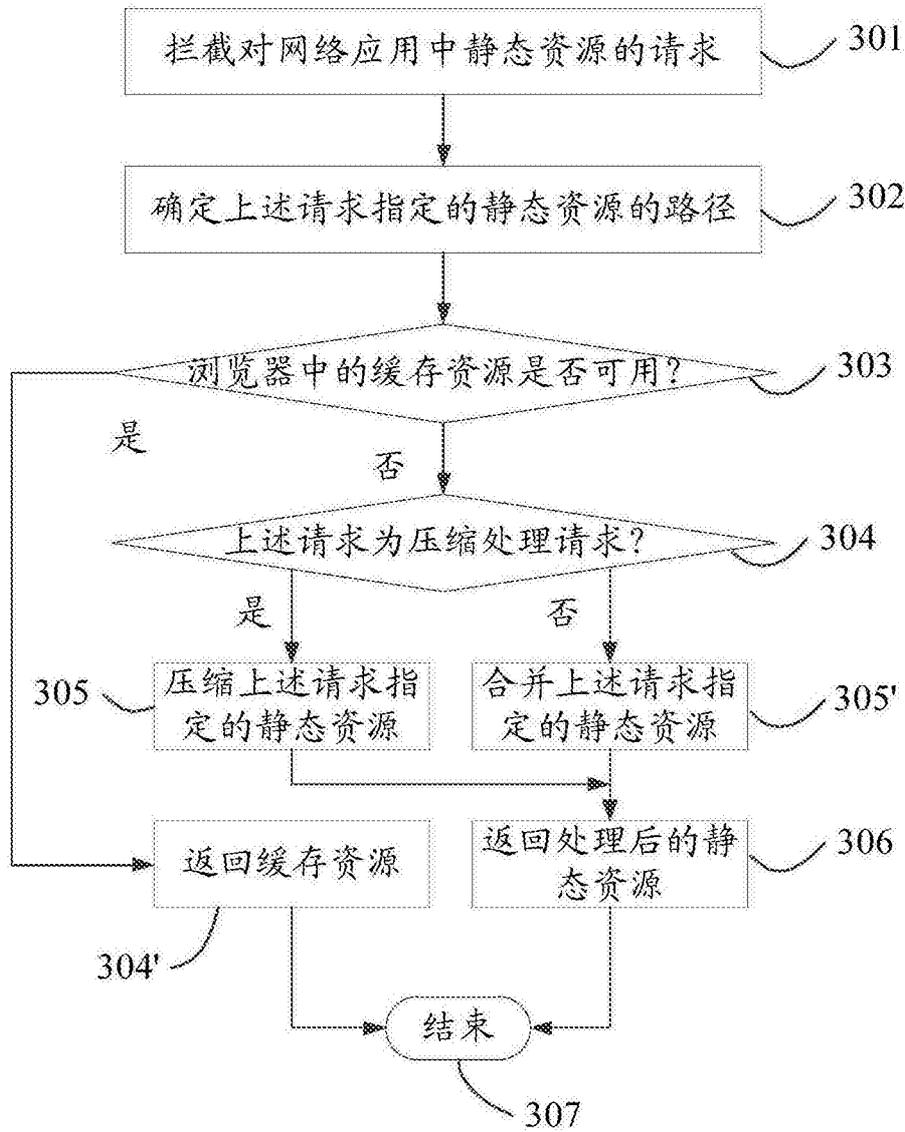


图3

400

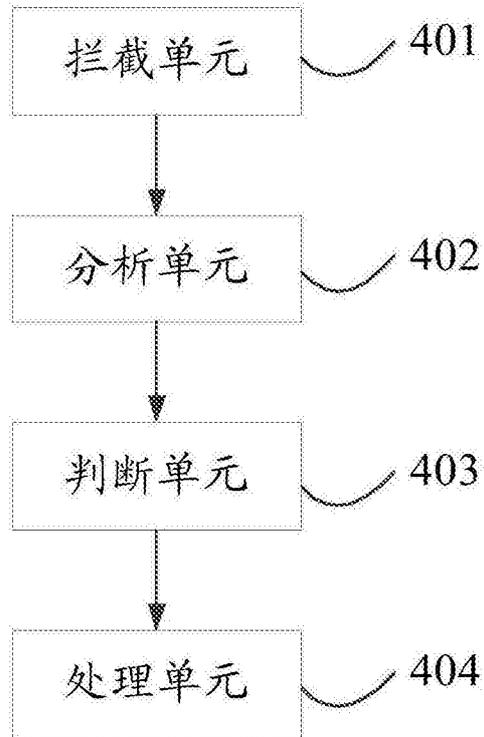


图4

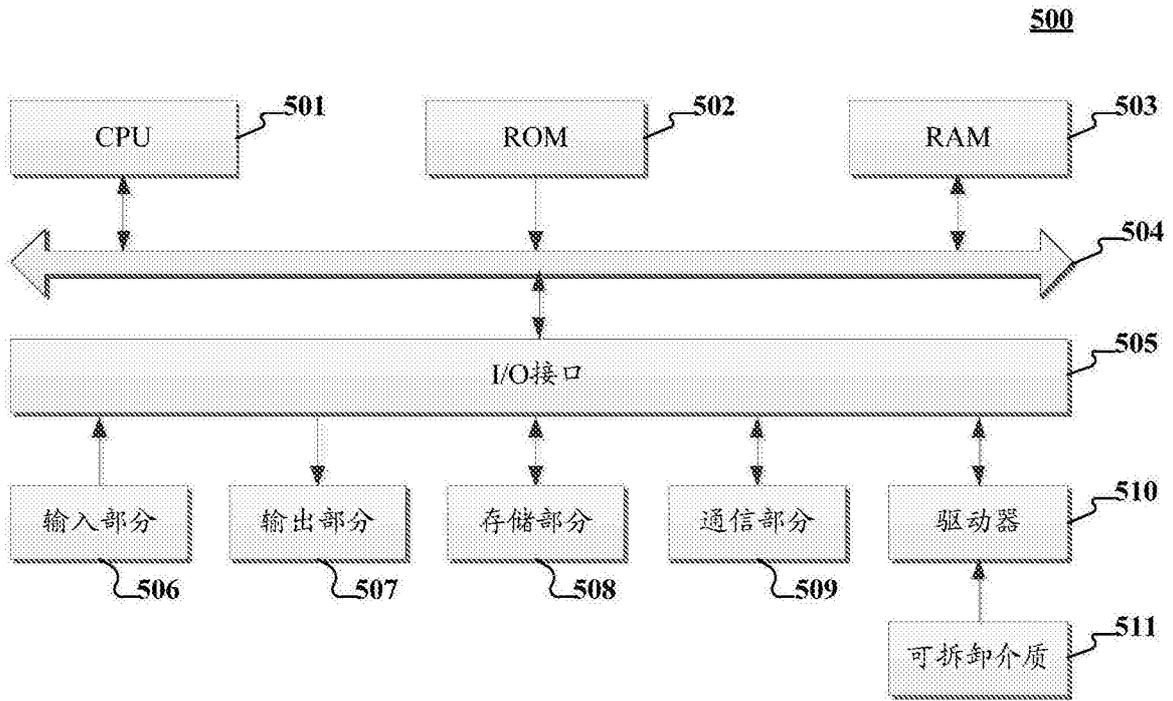


图5