(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 109388766 A
(43)申请公布日 2019.02.26

(21)申请号 201710666009.X

(22)申请日 2017.08.07

(71)申请人 阿里巴巴集团控股有限公司
地址 英属开曼群岛大开曼资本大厦一座四层847号邮箱

(72)发明人 张舒迪

(74)专利代理机构 北京清源汇知识产权代理事务所(特殊普通合伙) 11644
代理人 冯德魁 窦晓慧

(51)Int.Cl.
G06F 16/958(2019.01)
G06F 16/957(2019.01)

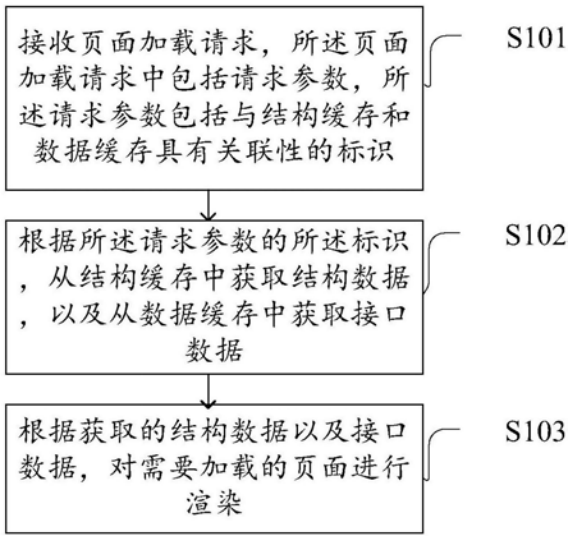
权利要求书2页 说明书11页 附图2页

(54)发明名称

页面加载的方法和装置

(57)摘要

本申请公开页面加载的方法和装置，该方法具有结构缓存和数据缓存的缓存模式，包括：接收页面加载请求，所述页面加载请求中包括请求参数，所述请求参数包括与结构缓存和数据缓存具有关联性的标识；根据所述请求参数的所述标识，从结构缓存中获取结构数据，以及从数据缓存中获取接口数据；根据获取的结构数据以及接口数据，对需要加载的页面进行渲染。该方法是基于接口数据与DOM结构的双缓存方案，基于DOM结构缓存的秒出方案，可以以最快速度完成首屏渲染。同时，对于依赖接口数据的场景，极大减少二次加载的页面可见时间及页面可用时间，加快了Html页面首屏的加载时间。



CN 109388766 A

1.一种页面加载的方法，其特征在于，具有结构缓存和数据缓存的缓存模式，该方法包括：

接收页面加载请求，所述页面加载请求中包括请求参数，所述请求参数包括与结构缓存和数据缓存具有关联性的标识；

根据所述请求参数的所述标识，从结构缓存中获取结构数据，以及从数据缓存中获取接口数据；

根据获取的结构数据以及接口数据，对需要加载的页面进行渲染。

2.根据权利要求1所述的页面加载的方法，其特征在于，在所述接收页面加载请求之后，包括：

判断所述页面加载请求是否为初次请求；

若是，根据所述页面加载请求获取结构数据以及与接口相关的接口数据；

将所述结构数据、请求参数和接口名称存储在结构缓存中，将所述接口数据、请求参数和接口名称存储在数据缓存中；

接收第二次页面加载请求；

所述根据所述请求中的请求参数，基于所述标识从结构缓存中获取结构数据，以及从数据缓存中获取接口数据，包括：

根据所述第二次页面加载请求中的请求参数，以所述请求参数为依据，分别从结构缓存中获取结构数据，从数据缓存中获取接口数据。

3.根据权利要求2所述的页面加载的方法，其特征在于，在所述将所述结构数据、请求参数和接口名称存储在结构缓存中，将所述接口数据、请求参数和接口名称存储在数据缓存中之后，包括：

将所述结构缓存与所述数据缓存之间设置对应关系；

所述结构缓存与所述数据缓存采用请求参数和接口名称作为键值进行标记。

4.根据权利要求3所述的页面加载的方法，其特征在于，所述将所述结构缓存与所述数据缓存之间设置对应关系，包括：

若请求参数相同，则所述结构缓存与所述数据缓存之间的对应关系为一一对应关系。

5.根据权利要求2所述的页面加载的方法，其特征在于，所述将所述结构数据、请求参数和接口名称存储在结构缓存中，包括：

预先设置结构缓存容器；

根据所述请求参数获取相应的结构数据；

在所述结构缓存容器中将所述结构数据转换为字符串格式；

将字符串格式的结构数据、请求参数以及接口名称存储在所述结构缓存中。

6.根据权利要求5所述的页面加载的方法，其特征在于，所述根据获取的结构数据以及接口数据对需要加载的页面进行渲染中，根据获取的结构数据对需要加载的页面进行渲染，包括：

根据所述请求参数和接口名称获取相对应的字符串；

根据所述字符串生成相应的页面结构；

将所述结构通过所述结构缓存容器进行页面渲染。

7.根据权利要求1所述的页面加载的方法，其特征在于，所述根据所述请求中的请求参

数，基于所述标识从结构缓存中获取结构数据，以及从数据缓存中获取接口数据，包括：

先执行根据所述请求中的请求参数，从结构缓存中获取结构数据；

再执行从数据缓存中获取接口数据。

8.根据权利要求1所述的页面加载的方法，其特征在于，在所述根据获取的结构数据以及接口数据对需要加载的页面进行渲染之后，执行以下操作：

接收缓存更新请求；

根据所述缓存更新请求，将所述结构数据和结构数据中缓存的数据进行更新。

9.根据权利要求8所述的页面加载的方法，其特征在于，所述根据所述缓存更新请求，将所述结构数据和结构数据中缓存的数据进行更新，包括：

根据所述缓存更新请求中的请求参数和接口名称，获取更新的结构数据和更新的接口数据；

将所述更新的结构数据存储在结构缓存中，将所述更新的接口数据存储在接口缓存中。

10.根据权利要求1所述的页面加载的方法，其特征在于，所述根据所述请求中的请求参数，基于所述标识从结构缓存中获取结构数据，以及从数据缓存中获取接口数据，包括：

根据接收的页面加载请求，获取所述结构缓存中的页面渲染；

根据所述页面加载请求，获取该次页面渲染与之前的页面渲染之间的差异性；

根据获取的所述差异性，获得此次页面加载请求的页面渲染。

11.根据权利要求10所述的页面加载的方法，其特征在于，所述获取该次页面渲染与之前的页面渲染之间的差异性，包括：

采用虚拟DOM方式获取该次页面渲染与之前的页面渲染之间的差异性。

12.根据权利要求10所述的页面加载的方法，其特征在于，预先设置缓存过期时间；

在所述根据所述页面加载请求，获取该次页面渲染与之前的页面渲染之间的差异性之前，执行以下操作：

判断是否在所述缓存过期时间之前；

若是，则执行根据所述页面加载请求，获取该次页面渲染与之前的页面渲染之间的差异性。

13.一种页面加载的装置，其特征在于，包括：

存储单元，用于设置具有结构缓存和数据缓存的缓存模式；

接收单元，用于接收页面加载请求；所述页面加载请求中包括请求参数；所述请求参数包括与结构缓存和数据缓存具有关联性的标识；

获取单元，用于根据所述请求中的请求参数，基于所述标识从结构缓存中获取结构数据，以及从数据缓存中获取接口数据；

页面渲染单元，用于根据获取的结构数据以及接口数据对需要加载的页面进行渲染。

14.一种页面加载的方法，其特征在于，该方法包括：

接收页面加载请求，所述页面加载请求中包括请求参数，所述请求参数与结构缓存和数据缓存具有关联性；

根据所述请求参数，从结构缓存中获取结构数据，以及从数据缓存中获取接口数据；

根据获取的结构数据以及接口数据，对需要加载的页面进行渲染。

页面加载的方法和装置

技术领域

[0001] 本申请涉及网页中页面加载的技术领域，具体涉及一种页面加载的方法。本申请同时涉及一种页面加载的装置。

背景技术

[0002] 针对现有的各种无线客户端，如手机淘宝、支付宝、阿里旅行等，且客户端内大量使用了Html5(以下可简称为H5)作为页面展示容器的混合开发方案。

[0003] 传统的H5页面加载及渲染需要经历HTML加载、CSS/JS加载、数据请求、页面渲染四个步骤，但是，在网络环境不理想的情况下，过多的资源加载与初始化会大大延长页面加载及渲染完成时间。

[0004] 对于用户体验来说，点击跳转后的页面展示时间是十分关键的指标，为此，各容器都实现了静态资源离线化方案(将HTML/JS/CSS文件离线化到本地)，减少资源加载时间，但数据接口访问依然受制于网络状况，如商品列表页等场景页面依然无法实现秒出。

发明内容

[0005] 本申请提供一种页面加载的方法，以解决现有技术中存在的上述技术问题。

[0006] 本申请另外提供页面加载的装置。

[0007] 本申请提供一种页面加载的方法，具有结构缓存和数据缓存的缓存模式，该方法包括：

[0008] 接收页面加载请求，所述页面加载请求中包括请求参数，所述请求参数包括与结构缓存和数据缓存具有关联性的标识；

[0009] 根据所述请求参数的所述标识，从结构缓存中获取结构数据，以及从数据缓存中获取接口数据；

[0010] 根据获取的结构数据以及接口数据，对需要加载的页面进行渲染。

[0011] 可选的，在所述接收页面加载请求之后，包括：

[0012] 判断所述页面加载请求是否为初次请求；

[0013] 若是，根据所述页面加载请求获取结构数据以及与接口相关的接口数据；

[0014] 将所述结构数据、请求参数和接口名称存储在结构缓存中，将所述接口数据、请求参数和接口名称存储在数据缓存中；

[0015] 接收第二次页面加载请求；

[0016] 所述根据所述请求中的请求参数，基于所述标识从结构缓存中获取结构数据，以及从数据缓存中获取接口数据，包括：

[0017] 根据所述第二次页面加载请求中的请求参数，以所述请求参数为依据，分别从结构缓存中获取结构数据，从数据缓存中获取接口数据。

[0018] 可选的，在所述将所述结构数据、请求参数和接口名称存储在结构缓存中，将所述接口数据、请求参数和接口名称存储在数据缓存中之后，包括：

[0019] 将所述结构缓存与所述数据缓存之间设置对应关系；

[0020] 所述结构缓存与所述数据缓存采用请求参数和接口名称作为键值进行标记。

[0021] 可选的，所述将所述结构缓存与所述数据缓存之间设置对应关系，包括：

[0022] 若请求参数相同，则所述结构缓存与所述数据缓存之间的对应关系为一一对应关系。

[0023] 可选的，所述将所述结构数据、请求参数和接口名称存储在结构缓存中，包括：

[0024] 预先设置结构缓存容器；

[0025] 根据所述请求参数获取相应的结构数据；

[0026] 在所述结构缓存容器中将所述结构数据转换为字符串格式；

[0027] 将字符串格式的结构数据、请求参数以及接口名称存储在所述结构缓存中。

[0028] 可选的，所述根据获取的结构数据以及接口数据对需要加载的页面进行渲染中，根据获取的结构数据对需要加载的页面进行渲染，包括：

[0029] 根据所述请求参数和接口名称获取相对应的字符串；

[0030] 根据所述字符串生成相应的页面结构；

[0031] 将所述结构通过所述结构缓存容器进行页面渲染。

[0032] 可选的，所述根据所述请求中的请求参数，基于所述标识从结构缓存中获取结构数据，以及从数据缓存中获取接口数据，包括：

[0033] 先执行根据所述请求中的请求参数，从结构缓存中获取结构数据；

[0034] 再执行从数据缓存中获取接口数据。

[0035] 可选的，在所述根据获取的结构数据以及接口数据对需要加载的页面进行渲染之后，执行以下操作：

[0036] 接收缓存更新请求；

[0037] 根据所述缓存更新请求，将所述结构数据和结构数据中缓存的数据进行更新。

[0038] 可选的，所述根据所述缓存更新请求，将所述结构数据和结构数据中缓存的数据进行更新，包括：

[0039] 根据所述缓存更新请求中的请求参数和接口名称，获取更新的结构数据和更新的接口数据；

[0040] 将所述更新的结构数据存储在结构缓存中，将所述更新的接口数据存储在接口缓存中。

[0041] 可选的，所述根据所述请求中的请求参数，基于所述标识从结构缓存中获取结构数据，以及从数据缓存中获取接口数据，包括：

[0042] 根据接收的页面加载请求，获取所述结构缓存中的页面渲染；

[0043] 根据所述页面加载请求，获取该次页面渲染与之前的页面渲染之间的差异性；

[0044] 根据获取的所述差异性，获得此次页面加载请求的页面渲染。

[0045] 可选的，所述获取该次页面渲染与之前的页面渲染之间的差异性，包括：

[0046] 采用虚拟DOM方式获取该次页面渲染与之前的页面渲染之间的差异性。

[0047] 可选的，预先设置缓存过期时间；

[0048] 在所述根据所述页面加载请求，获取该次页面渲染与之前的页面渲染之间的差异性之前，执行以下操作：

[0049] 判断是否在所述缓存过期时间之前；

[0050] 若是，则执行根据所述页面加载请求，获取该次页面渲染与之前的页面渲染之间的差异性。

[0051] 本申请还提供一种页面加载的装置，该装置包括：

[0052] 存储单元，用于设置具有结构缓存和数据缓存的缓存模式；

[0053] 接收单元，用于接收页面加载请求；所述页面加载请求中包括请求参数；所述请求参数包括与结构缓存和数据缓存具有关联性的标识；

[0054] 获取单元，用于根据所述请求中的请求参数，基于所述标识从结构缓存中获取结构数据，以及从数据缓存中获取接口数据；

[0055] 页面渲染单元，用于根据获取的结构数据以及接口数据对需要加载的页面进行渲染。

[0056] 可选的，该装置还包括：

[0057] 判断单元，用于在所述接收页面加载请求之后，判断所述页面加载请求是否为初次请求；

[0058] 数据获取单元，当判断单元的结果为是时，根据所述页面加载请求获取结构数据以及与接口相关的接口数据；

[0059] 缓存单元，用于将所述结构数据、请求参数和接口名称存储在结构缓存中，将所述接口数据、请求参数和接口名称存储在数据缓存中；

[0060] 第二次请求接收单元，用于接收第二次页面加载请求；

[0061] 所述获取单元具体用于根据所述第二次页面加载请求中的请求参数，以所述请求参数为依据，分别从结构缓存中获取结构数据，从数据缓存中获取接口数据。

[0062] 可选的，该装置还包括：

[0063] 关系设置单元，用于在所述将所述结构数据、请求参数和接口名称存储在结构缓存中，将所述接口数据、请求参数和接口名称存储在数据缓存中之后，将所述结构缓存与所述数据缓存之间设置对应关系；

[0064] 标记单元，用于所述结构缓存与所述数据缓存采用请求参数和接口名称作为键值进行标记。

[0065] 可选的，所述缓存单元包括：

[0066] 预设子单元，用于预先设置结构缓存容器；

[0067] 数据获取子单元，用于根据所述请求参数获取相应的结构数据；

[0068] 转换子单元，用于在所述结构缓存容器中将所述结构数据转换为字符串格式；

[0069] 缓存子单元，用于将字符串格式的结构数据、请求参数以及接口名称存储在所述结构缓存中。

[0070] 可选的，所述页面渲染单元包括：

[0071] 字符串获取子单元，用于根据所述请求参数和接口名称获取相对应的字符串；

[0072] 结构生成子单元，用于根据所述字符串生成相应的页面结构；

[0073] 渲染子单元，用于将所述结构通过所述结构缓存容器进行页面渲染。

[0074] 本申请还提供一种页面加载的方法，该方法包括：

[0075] 接收页面加载请求，所述页面加载请求中包括请求参数，所述请求参数与结构缓

存和数据缓存具有关联性；

[0076] 根据所述请求参数，从结构缓存中获取结构数据，以及从数据缓存中获取接口数据；

[0077] 根据获取的结构数据以及接口数据，对需要加载的页面进行渲染。

[0078] 与现有技术相比，本申请具有以下优点：

[0079] 本申请提供一种页面加载的方法，具有结构缓存和数据缓存的缓存模式，该方法包括：接收页面加载请求，所述页面加载请求中包括请求参数，所述请求参数包括与结构缓存和数据缓存具有关联性的标识；根据所述请求参数的所述标识，从结构缓存中获取结构数据，以及从数据缓存中获取接口数据；根据获取的结构数据以及接口数据，对需要加载的页面进行渲染。

[0080] 本申请所提供的上述方法是基于接口数据与DOM结构的双缓存方案，即同时缓存DOM结构与接口数据。基于DOM结构缓存的秒出方案，可以以最快速度完成首屏渲染。同时，对于依赖接口数据的场景，极大减少二次加载的页面可见时间及页面可用时间。该方法同时缓存DOM结构与接口数据，优势互补，同时将页面的可视时间与可用时间均提前，达到点击即可见、可见即可用的技术效果，在一定程度上加快了Html页面首屏的加载时间。

附图说明

[0081] 图1是本申请第一实施例提供的页面加载的方法的流程图。

[0082] 图2是本申请第一实施例提供的页面渲染流程的总体架构的示意图。

[0083] 图3是本申请第一实施例提供的页面加载和渲染简化的流程图。

[0084] 图4是本申请第二实施例提供的页面加载的装置的结构示意图。

[0085] 图5是本申请第三实施例提供的页面加载的方法的流程图。

具体实施方式

[0086] 本申请第一实施例提供一种页面加载的方法，该方法主要应用于移动客户端使用Html页面的场景，在移动客户端中很多情况下会使用的Html作为页面信息的展示方式。本申请提供的页面加载的方法主要应用于上述场景中。另外，采用本申请提供的页面加载的方法可以缩短加载页面信息的时间，在最短时间内加载出Html页面所需要加载信息，以对Html页面进行渲染。

[0087] 采用该方法是基于接口数据与DOM结构的双缓存方案，即同时缓存DOM结构与接口数据。基于DOM缓存的秒出方案，可以以最快速度完成首屏渲染。同时，对于依赖接口数据的场景，极大减少二次加载的页面可见时间及页面可用时间。该方法同时缓存DOM结构与接口数据，优势互补，同时将页面的可视时间与可用时间均提前，达到点击即可见、可见即可用的技术效果，在一定程度上加快了Html页面的加载时间。

[0088] 以下通过具体的实施例对该方法进行详细的介绍和说明。

[0089] 图1是本申请第一实施例提供的页面加载的方法的流程图。请参照图1，该方法包括以下步骤：

[0090] 步骤S101，接收页面加载请求，所述页面加载请求中包括请求参数，所述请求参数包括与结构缓存和数据缓存具有关联性的标识。

[0091] 该步骤是客户端接收页面加载请求的过程，本实施例中的页面加载可以是纯Html页面，也可以是APP应用中嵌入的Html页面，该方法适用于任何Html页面首页的加载。本实施例着重介绍APP应用中嵌入的Html页面的首页的加载，因为针对APP应用的用户来说，点击跳转后的Html页面的展示时间是十分重要的，可以影响用户体验，所以，本实施例着重介绍APP应用中嵌入的Html页面的首页的加载。

[0092] 需要说明的是，本实施例提供的方法是具有结构缓存和数据缓存的双缓存模式的，也就是该方法是采用双缓存策略加快页面加载的时间。

[0093] 所述结构缓存是将DOM结构转换为字符串进行缓存，在二次加载时可以优先读取缓存数据；所述数据缓存是页面异步接口数据返回后，将数据存入缓存中，在静态资源加载完成后，以缓存数据模拟数据返回情况提前渲染。以下会对双缓冲缓存存模式进行详细的介绍和说明。

[0094] 所述页面加载请求中包括请求参数，所述请求参数包括与结构缓存和数据缓存具有关联性的标识。

[0095] 所述结构缓存和数据缓存之间具有关联性，而该关联性通过所述标识实现，所述标识包括Html页面加载的数据接口的接口名称，根据所述接口名称以及请求参数可以将所述结构缓存和数据缓存之间建立关联。

[0096] 关于所述结构缓存与数据缓存之间是通过什么方式建立关联的，在后续步骤中有详细的介绍和说明。

[0097] 请继续参照图1，步骤S102，根据所述请求参数的所述标识，从结构缓存中获取结构数据，以及从数据缓存中获取接口数据。

[0098] 首先介绍该步骤的背景知识，结构缓存可以是DOM结构数据的缓存，文档对象模型(Document Object Model，简称DOM)，是W3C组织推荐的处理可扩展标志语言的标准编程接口。在网页上，组织页面(或文档)的对象被组织在一个树形结构中，用来表示文档中对象的标准模型就称为DOM。

[0099] DOM认为是页面上数据和结构的一个树形表示，通过Java Script，可以重构整个Html文档。可以添加、移除、改变或重排页面上的项目。要改变页面的某个信息，JavaScript就需要获得对Html文档中所有元素进行访问的入口。该入口连同对Html元素进行添加、移动、改变或移除的方法和属性，都是通过文档对象模型(DOM)来获得的。

[0100] 但是，基于DOM缓存的秒出方案存在以下问题：缓存页面因无JS(java script)逻辑，无法实现基于JS的交互逻辑，如滚屏广告、跑马灯不能滚动，下拉框等组件点击无效，实际可操作仍需等到数据接口返回后。

[0101] 另外，基于接口数据缓存的秒出方案存在以下问题：在工程化Html5(可简称H5)前端解决方案中，JS一般包括基础库与业务库，即使离线化到本地，磁盘IO(读写)加载效率依然较低，而该方案需要依赖JS执行渲染逻辑，因此无法实现DOM缓存方案达到的秒出效果。

[0102] 但是，本实施例所提供的方法是基于接口数据与DOM结构的双缓存方案，通过有效的方式将两个缓存方案结合，创造出新的可加快页面加载的方法。同时缓存DOM结构与接口数据，优势互补，同时提前页面的可视时间与可用时间，达到点击即可见、可见即可用的技术效果。

[0103] 具体的，该步骤是基于接口数据缓存的秒出的技术方案。页面异步接口数据返回

后，将数据存入缓存中，在静态资源加载完成后，以缓存数据模拟数据返回情况提前渲染。该步骤可产生的技术效果是基于缓存数据秒出后，页面即是可用状态。

[0104] 该步骤是根据所述请求参数和标识，分别从结构缓存中获取结构数据和从数据缓存中获取结构数据的过程。由于结构缓存和数据缓存对应请求参数，通过请求参数可以获取到响应的数据和信息。

[0105] 所述结构数据和接口数据的读取过程是针对页面的第二次加载的情况，首次加载时是需要建立结构缓存和接口数据的过程，在首次加载完成之后，将相应的数据进行缓存，在第二次或后续加载访问时，可以直接将已经加载的数据进行读取。

[0106] 因此，为了更清楚的描述上述数据读取的过程，首先介绍首次加载页面的过程。

[0107] 首次加载页面的步骤在该步骤之前，具体的，在所述接收页面加载请求之后，包括：判断所述页面加载请求是为初次请求；若是，根据所述页面加载请求获取结构数据以及与接口相关的接口数据；将所述结构数据、请求参数和接口名称存储在结构缓存中，将所述接口数据、请求参数和接口名称存储在数据缓存中。

[0108] 之后，接收第二次页面加载请求；相应的，该步骤所述根据所述请求中的请求参数，基于所述标识从结构缓存中获取结构数据，以及从数据缓存中获取接口数据，所执行的操作具体为：根据所述第二次页面加载请求中的请求参数，以所述请求参数为依据，分别从结构缓存中获取结构数据，从数据缓存中获取接口数据。

[0109] 首先介绍首次加载页面的情况，在接收页面加载请求之后，需要判断该请求是否为初次请求，若是，则说明该页面加载为第一次加载，这种情况下，需要对结构数据和接口数据进行缓存；若上述判断结果为否，则说明该页面加载并不是首次加载，而是二次加载或多次加载的情况，在该情况下，可直接执行该步骤S102。

[0110] 因此，针对首次加载的情况，根据所述页面加载请求获取结构数据以及与接口相关的接口数据，将所述结构数据、请求参数和接口名称作为标识存储在结构缓存中，将所述接口数据、请求参数和接口名称存储在数据缓存中。后续数据读取的过程可通过请求参数和接口名称作为识别的标识进行对应。

[0111] 在所述将所述结构数据、请求参数和接口名称存储在结构缓存中，将所述接口数据、请求参数和接口名称存储在数据缓存中之后，包括：将所述结构缓存与所述数据缓存之间设置对应关系；所述结构缓存与所述数据缓存采用请求参数和接口名称作为键值进行标记。需要说明的是，若请求参数相同，则所述结构缓存与所述数据缓存之间的对应关系为一一对应关系。

[0112] 所述将所述结构数据、请求参数和接口名称存储在结构缓存中，包括：预先设置结构缓存容器；根据所述请求参数获取相应的结构数据；在所述结构缓存容器中将所述结构数据转换为字符串格式；将字符串格式的结构数据、请求参数以及接口名称存储在所述结构缓存中。

[0113] 具体的，根据所述请求数据获取该请求数据对应的DOM容器；获取所述DOM容器中的信息转换为字符串，将所述字符串存入所述DOM缓存中。其中，所述字符串是以请求参数和接口名称为参考标识存入所述DOM缓存中。

[0114] 而针对数据缓存，将请求参数和接口名称同时作为数据缓存的参考标识中的键值，获取相应的接口数据，并将接口数据存入数据缓存中。根据所述相同的请求参数和接口

名称作为键值，将所述DOM缓存与所述数据缓存之间确定为一一对应关系。

[0115] 图2是本申请第一实施例提供的页面渲染流程的总体架构的示意图。请参照图2，见架构图中上半侧部分。数据请求接口返回后，首先将返回数据以接口名称、请求参数为键值存入缓存中。假设请求数据渲染目标DOM容器id定义为J_Container，在渲染完成后，取J_Container中Html片段内容转换为字符串，同样以接口名称、请求参数为键值存入数据缓存中，保证数据缓存与DOM的结构缓存一一对应。

[0116] 以上是对首次加载的情况的介绍，在首次加载完成之后，用户第二次访问该页面时，需要再次加载该页面，而此时，只需要根据所述第二次页面加载请求中的请求参数，以所述请求参数为依据，分别从结构缓存中获取结构数据，从数据缓存中获取接口数据。

[0117] 具体的，可继续参照图2，见架构图中下半侧部分。

[0118] 二次访问Html/CSS加载完成后，使用原生JS(javascript)(此处主要是为了提高效率)将对应接口名/参数缓存字符串取出、生成对应DOM结构并填入容器J_Container渲染，此时页面达到可视状态，用户可滑动页面查看，但仍不可交互。

[0119] 当JS加载完成并初始化后，优先读取对应接口名称和请求参数缓存数据，并基于数据执行处理逻辑二次渲染J_Container，此时页面达到可用状态，页面自有逻辑开始执行，用户也可以与页面进行正常交互。

[0120] 另外，关于结构数据和接口数据的读取顺序，可以先执行根据所述请求中的请求参数，从结构缓存中获取结构数据；再执行从数据缓存中获取接口数据。采用这样的顺序符合数据加载的规则。

[0121] 请继续参照图1，步骤S103，根据获取的结构数据以及接口数据，对需要加载的页面进行渲染。

[0122] 在上一步骤中已经获取到结构数据和接口数据，通过结构数据和接口数据就可以完成页面的渲染。不管是首次页面渲染还是二次页面渲染，都是依据获取的结构数据和接口数据对页面进行渲染的。

[0123] 页面渲染，就是浏览器将请求返回的页面资源(Html文本，图像，动画，视频，音频等)基于一定的规则(CSS语句，JS语句，浏览器本身的一些规则等)完成页面布局及绘制的过程。

[0124] 对加载到的资源(Html、JS、CSS等)进行语法解析，相应的内部数据结构(比如Html的DOM树，JS的(对象)属性表，CSS的样式规则等等)。

[0125] 对页面加载和渲染流程进行简单的介绍和说明。图3是本申请第一实施例提供的页面加载和渲染简化的流程图。请参照图3，整体方案基于静态资源离线化技术，默认Html/CSS/JS资源文件加载仅包含磁盘IO读写时间，不包含网络传输时间；JS初始化主要指H5基础库、业务库的初始化逻辑执行，在航旅技术体系下，约需要400～600ms；数据请求需要视网络情况时长区间较大，在航旅技术体系下，平均需要约为480ms。

[0126] 其中，页面可见以及可用均需要时间，相应的，可见时间是指：用户能够看到页面结构、内容的时间，包含基于接口数据渲染得到的列表等。

[0127] 可用时间是指：用户能够与渲染完成页面进行正常交互的时间，包括但不限于基于JS逻辑的页面自动滚动、内容变化及用户交互产生的页面反馈。

[0128] 具体的，所述根据获取的结构数据以及接口数据对需要加载的页面进行渲染中，

根据获取的结构数据对需要加载的页面进行渲染，包括：根据所述请求参数和接口名称获取相对应的字符串；根据所述字符串生成相应的页面结构；将所述结构通过所述结构缓存容器进行页面渲染。

[0129] 其中，所述根据所述请求参数和接口名称获取相对应的字符串，包括：将请求参数和接口名称作为参考标识，采用java script根据所述请求参数和接口名称确定对应的DOM结构缓存，从该确定的DOM结构缓存中获取字符串。

[0130] 另外，针对页面数据的更新的情况，在所述根据获取的结构数据以及接口数据对需要加载的页面进行渲染之后，执行以下操作：接收缓存更新请求；根据所述缓存更新请求，将所述结构数据和结构数据中缓存的数据进行更新。

[0131] 具体的，所述根据所述缓存更新请求，将所述结构数据和结构数据中缓存的数据进行更新，包括：根据所述缓存更新请求中的请求参数和接口名称，获取更新的结构数据和更新的接口数据；将所述更新的结构数据存储在结构缓存中，将所述更新的接口数据存储在接口缓存中。

[0132] 缓存数据渲染完成后，发送正常的数据请求，获取返回数据后再次更新页面，此时页面状态与请求线上数据一致。同时，将最新的数据请求结果及渲染结果以相应键值存入缓存中，更新缓存结果。

[0133] 上述是对缓存数据更新情况的介绍和说明，除了通过缓存数据更新的方式更新缓存之后，还可以通过其他方式解决缓存数据的差异性。

[0134] 具体的，所述根据所述请求中的请求参数，基于所述标识从结构缓存中获取结构数据，以及从数据缓存中获取接口数据，包括：根据接收的页面加载请求，获取所述结构缓存中的页面渲染；根据所述页面加载请求，获取该次页面渲染与之前的页面渲染之间的差异性；根据获取的所述差异性，获得此次页面加载请求的页面渲染。

[0135] 需要说明的是，所述获取该次页面渲染与之前的页面渲染之间的差异性，包括：采用虚拟DOM方式获取该次页面渲染与之前的页面渲染之间的差异性。

[0136] 另外，还需要设置一个前提条件，就是需要预先设置缓存过期时间。

[0137] 相应的，在所述根据所述页面加载请求，获取该次页面渲染与之前的页面渲染之间的差异性之前，执行以下操作：判断是否在所述缓存过期时间之前；若是，则执行根据所述页面加载请求，获取该次页面渲染与之前的页面渲染之间的差异性。

[0138] 具体的操作方式如下：缓存数据需要根据业务场景设置过期时间，且必须与当前业务代码版本匹配，确保数据时效性及渲染正确性。

[0139] 对于缓存DOM、缓存接口数据、异步接口数据三次渲染带来的性能开销，主要是在H5DOM渲染部分。本发明采用开源Virtual DOM(虚拟DOM)技术解决，即在第一次渲染完成后，后续渲染仅对比两次渲染中的差异部分并进行渲染，一方面减少了多次渲染造成的页面闪烁、抖动问题；另一方面在缓存过期时间设置合理的前提下，缓存数据与异步数据差异一般不大，diff后生成的待更新部分一般较少，因此渲染性能完全能够达到使用需求。

[0140] 其中，diff命令比较文本文件。它能比较单个文件或者目录内容。diff命令只有当输入为文本文件时才有效。

[0141] 因此，该方法的核心是加快H5页面的首屏加载，因此针对性地设计了基于接口数据与DOM结构的双缓存方案与实施细节，同时引入Virtual DOM解决多次渲染带来的性能开

销，达到同时缩短页面可见时间与页面可用时间的目的。

[0142] 另外，对使用的缓存策略、Virtual DOM实现均还可使用替代技术解决。替代方案主要考虑首屏渲染提速，如首次渲染采用客户端渲染方式，H5容器直接展示；更快获取到业务数据以执行渲染，如采用服务端推技术定时更新本地数据，保证客户端页面访问即可拿到最新数据；降低渲染开销，如使用服务端渲染等。

[0143] 需要说明的是，该方法的应用场景可以是：阿里旅行iOS/Android客户端，及阿里旅行Html5页面投放的手机淘宝、支付宝、天猫等客户端及纯Html5页等等。

[0144] 总之，本申请实施例所提供的上述方法是基于接口数据与DOM结构的双缓存方案，即同时缓存DOM结构与接口数据。基于DOM结构缓存的秒出方案，可以以最快速度完成首屏渲染。同时，对于依赖接口数据的场景，极大减少二次加载的页面可见时间及页面可用时间。该方法同时缓存DOM结构与接口数据，优势互补，同时将页面的可视时间与可用时间均提前，达到点击即可见、可见即可用的技术效果，在一定程度上加快了Html页面首屏的加载时间。

[0145] 通过本专利提供的双缓存页面秒出方案，首次加载完成后缓存DOM结构及接口数据，二次加载优先走缓存数据，能够极大减少加载时间，达到提升用户体验的目的。

[0146] 在上述的实施例中，提供了一种页面加载的方法，与之相对应的，本申请还提供一种页面加载的装置。请参看图4，其为本申请的一种页面加载的装置的实施例的示意图。由于装置实施例基本相似于方法实施例，所以描述得比较简单，相关之处参见方法实施例的部分说明即可。下述描述的装置实施例仅仅是示意性的。

[0147] 本实施例的一种页面加载的装置，包括：

[0148] 存储单元401，用于设置具有结构缓存和数据缓存的缓存模式；

[0149] 接收单元402，用于接收页面加载请求；所述页面加载请求中包括请求参数；所述请求参数包括与结构缓存和数据缓存具有关联性的标识；

[0150] 获取单元403，用于根据所述请求中的请求参数，基于所述标识从结构缓存中获取结构数据，以及从数据缓存中获取接口数据；

[0151] 页面渲染单元404，用于根据获取的结构数据以及接口数据对需要加载的页面进行渲染。

[0152] 可选的，该装置还包括：

[0153] 判断单元，用于在所述接收页面加载请求之后，判断所述页面加载请求是否为初次请求；

[0154] 数据获取单元，当判断单元的结果为是时，根据所述页面加载请求获取结构数据以及与接口相关的接口数据；

[0155] 缓存单元，用于将所述结构数据、请求参数和接口名称存储在结构缓存中，将所述接口数据、请求参数和接口名称存储在数据缓存中；

[0156] 第二次请求接收单元，用于接收第二次页面加载请求；

[0157] 所述获取单元具体用于根据所述第二次页面加载请求中的请求参数，以所述请求参数为依据，分别从结构缓存中获取结构数据，从数据缓存中获取接口数据。

[0158] 可选的，该装置还包括：

[0159] 关系设置单元，用于在所述将所述结构数据、请求参数和接口名称存储在结构缓

存中，将所述接口数据、请求参数和接口名称存储在数据缓存中之后，将所述结构缓存与所述数据缓存之间设置对应关系；

[0160] 标记单元，用于所述结构缓存与所述数据缓存采用请求参数和接口名称作为键值进行标记。

[0161] 可选的，所述缓存单元包括：

[0162] 预设子单元，用于预先设置结构缓存容器；

[0163] 数据获取子单元，用于根据所述请求参数获取相应的结构数据；

[0164] 转换子单元，用于在所述结构缓存容器中将所述结构数据转换为字符串格式；

[0165] 缓存子单元，用于将字符串格式的结构数据、请求参数以及接口名称存储在所述结构缓存中。

[0166] 可选的，所述页面渲染单元包括：

[0167] 字符串获取子单元，用于根据所述请求参数和接口名称获取相对应的字符串；

[0168] 结构生成子单元，用于根据所述字符串生成相应的页面结构；

[0169] 渲染子单元，用于将所述结构通过所述结构缓存容器进行页面渲染。

[0170] 另外，本申请第三实施例还提供一种页面加载的方法，该方法与第一实施例相类似，但存在不同之处。

[0171] 具体的，图5是本申请第三实施例提供的页面加载的方法的流程图，请参照图5，该方法包括以下步骤：

[0172] 步骤S501，接收页面加载请求，所述页面加载请求中包括请求参数，所述请求参数与结构缓存和数据缓存具有关联性；

[0173] 步骤S502，根据所述请求参数，从结构缓存中获取结构数据，以及从数据缓存中获取接口数据；

[0174] 步骤S503，根据获取的结构数据以及接口数据，对需要加载的页面进行渲染。

[0175] 在该实施例中，并不强制要求有具有结构缓存和数据缓存的缓存模式，只要能够在方法实施的过程中能够实现DOM结构数据的获取以及接口数据的获取即可达到第一实施例可达到的技术效果。

[0176] 另外，只要能够保证请求参数与结构缓存和数据缓存之间具有关联即可，采用任何可以标识两者关联的指标均可，不管采用何种指标，将结构缓存和数据缓存通过请求参数关联即可。

[0177] 本实施例提供的方法的其他步骤和操作方式与第一实施例的方式相同，相同之处可参考第一实施例的说明。并且，本实施例提供的方法同样可以产生第一实施例的效果，通过本实施例提供的双缓存页面秒出方案，首次加载完成后缓存DOM结构及接口数据，二次加载优先走缓存数据，能够极大减少加载时间，达到提升用户体验的目的。

[0178] 本申请虽然以较佳实施例公开如上，但其并不是用来限定本申请，任何本领域技术人员在不脱离本申请的精神和范围内，都可以做出可能的变动和修改，因此本申请的保护范围应当以本申请权利要求所界定的范围为准。

[0179] 在一个典型的配置中，计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器(RAM)和/或非易失性内存等形式，如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介

质的示例。

[0180] 计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括非暂存电脑可读媒体(transitory media)，如调制的数据信号和载波。

[0181] 本领域技术人员应明白，本申请的实施例可提供为方法、系统或计算机程序产品。因此，本申请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，本申请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。

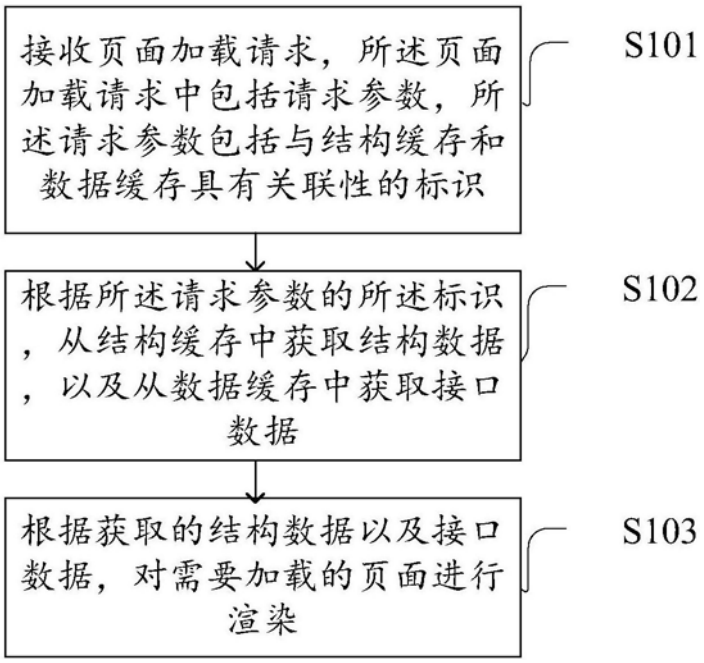

接收页面加载请求，所述页面加载请求中包括请求参数，所述请求参数包括与结构缓存和数据缓存具有关联性的标识
S101
根据所述请求参数的所述标识，从结构缓存中获取结构数据，以及从数据缓存中获取接口数据
S102
根据获取的结构数据以及接口数据，对需要加载的页面进行渲染
S103

图1

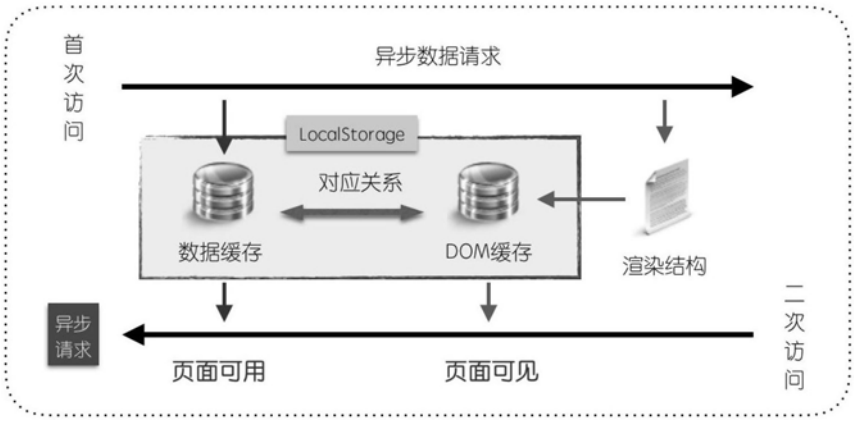

首次访问
异步数据请求
LocalStorage
对应关系
数据缓存
DOM缓存
渲染结构
异步请求
页面可用
页面可见
二次访问

图2

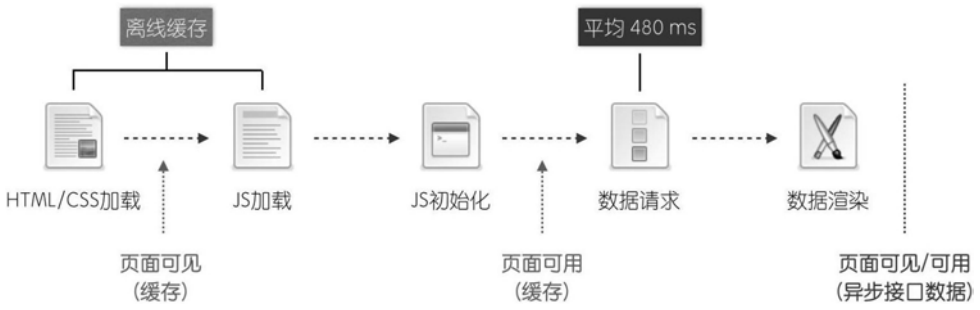

离线缓存
平均 480 ms
HTML/CSS加载
JS加载
JS初始化
数据请求
数据渲染
页面可见
(缓存)
页面可用
(缓存)
页面可见/可用
(异步接口数据)

图3

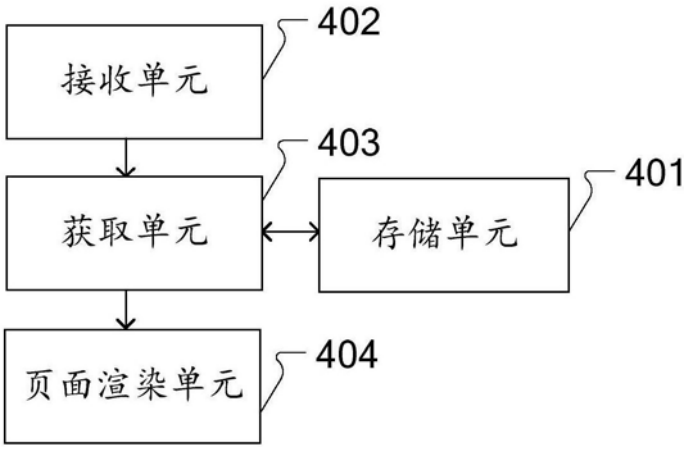

402
接收单元
403
获取单元
401
存储单元
404
页面渲染单元

图4

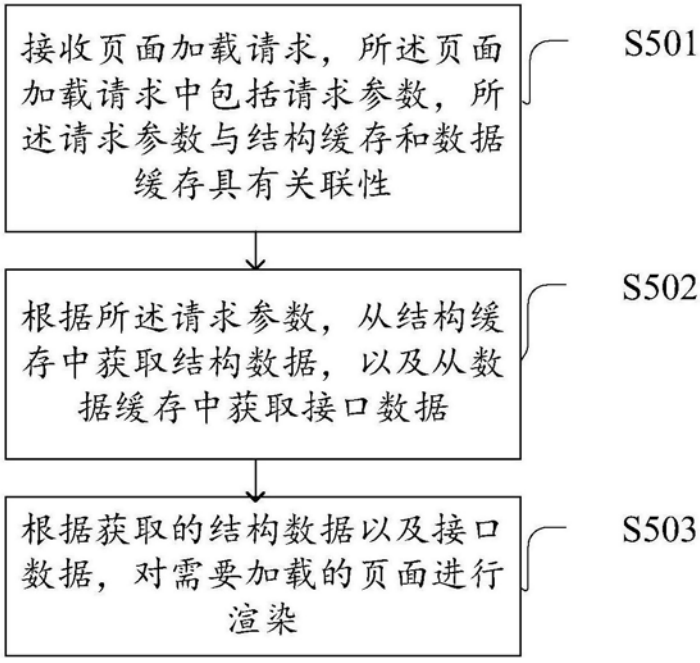

接收页面加载请求，所述页面加载请求中包括请求参数，所述请求参数与结构缓存和数据缓存具有关联性
S501
根据所述请求参数，从结构缓存中获取结构数据，以及从数据缓存中获取接口数据
S502
根据获取的结构数据以及接口数据，对需要加载的页面进行渲染
S503

图5