(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 109669720 B
(45)授权公告日 2022.05.10

(21)申请号 201811394545.X

(22)申请日 2018.11.22

(65)同一申请的已公布的文献号
申请公布号 CN 109669720 A

(43)申请公布日 2019.04.23

(73)专利权人 北京字节跳动网络技术有限公司
地址 100041 北京市石景山区实兴大街30号院3号楼2层B-0035房间

(72)发明人 姚成昊

(74)专利代理机构 北京风雅颂专利代理有限公司 11403
专利代理师 鲍胜如

(51)Int.Cl.
G06F 8/71(2018.01)
G06F 9/46(2006.01)

(56)对比文件
CN 105389209 A,2016.03.09
US 2015113122 A1,2015.04.23
CN 106911784 A,2017.06.30
Superman.使用Promise链式调用解决多个异步回调的问题.《https://m.jb51.net/article/103068.htm》.2017,第1-3页.

审查员 郑金珠

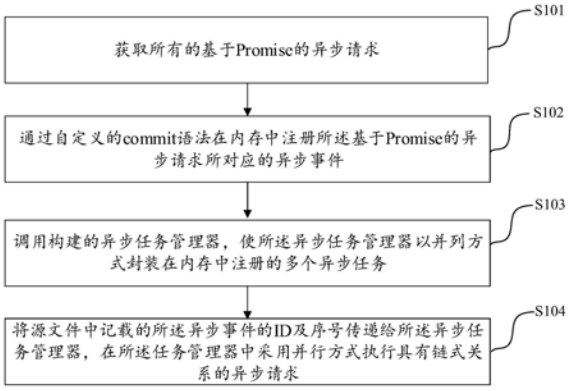
权利要求书2页 说明书10页 附图4页

(54)发明名称

基于Promise的链式异步请求处理方法、装置及电子设备

(57)摘要

本发明实施例中提供了一种基于Promise的链式异步请求处理方法、装置及电子设备，属于数据处理技术领域，该方法包括：获取所有的基于Promise的异步请求；通过自定义的commit语法在内存中注册所述基于Promise的异步请求所对应的异步事件；调用构建的异步任务管理器，使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务；将源文件中记载的所述异步事件的ID及序号传递给所述异步任务管理器，在所述任务管理器中采用并行方式执行具有链式关系的异步请求。通过本申请的处理方案，使得异步任务更急利于执行和维护。



CN 109669720 B

1.一种基于Promise的链式异步请求处理方法,其特征在于,包括:

获取所有的基于Promise的异步请求;

通过自定义的commit语法在内存中注册所述基于Promise的异步请求所对应的异步事件,包括:通过设置commit函数接收字符串参数;通过字符串匹配的方式在寻址内存中查找与所述字符串参数相同的运行任务;在与所述字符串参数相同的运行任务中选取符合条件的异步任务进行注册;

调用构建的异步任务管理器,使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务;

将源文件中记载的所述异步事件的ID及序号传递给所述异步任务管理器,在所述任务管理器中采用并行方式执行具有链式关系的异步请求。

2.根据权利要求1所述的方法,其特征在于,所述获取所有的基于Promise的异步请求,包括:

获取当前页面上的显示内容;

判断所述显示内容是否有多个内容源;

若是,则基于多个内容源来确定基于Promise的异步请求。

3.根据权利要求2所述的方法,其特征在于,所述基于多个内容源来确定基于Promise的异步请求,包括:

查找与所述多个内容源对应的多个内容请求;

从多个内容请求中选择基于具有链式异步请求关系的内容请求作为基于Promise的异步请求。

4.根据权利要求1所述的方法,其特征在于,所述在与所述字符串参数相同的运行任务中选取符合条件的异步任务进行注册,包括:

判断与所述字符串参数相同的运行任务是否是异步任务;

若是,则在当前没有调用中的异步任务的情况下执行当前任务,并在执行结束后寻找内存中是否有暂存未调用的任务集合;

若执行结束后寻找内存中存有暂存未调用的任务集合,则调用并重复当前过程,若当前有调用中的任务,则将未调用的任务集合存入异步任务调用集合,等待当前执行任务结束后再调用。

5.根据权利要求4所述的方法,其特征在于,所述在与所述字符串参数相同的运行任务中选取符合条件的异步任务进行注册,还包括:

当所有的异步任务都调用结束后,进一步判断当前调用结束时是否有全局的错误处理与成功处理;

若存在,则将结果返回给相应的处理,否则,直接返回处理结果。

6.根据权利要求1所述的方法,其特征在于,所述调用构建的异步任务管理器,使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务,包括:

在内存中将所需要用到的异步任务先行注册;

通过框架对所述异步任务进行封装,保证在执行过程中的任务符合框架内Promise异步规范的格式;

在所述异步任务在框架初始化时完成注册。

7.根据权利要求5所述的方法,其特征在于,所述调用构建的异步任务管理器,使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务,还包括:

在所述异步任务管理器被调用过程中,接收调用异步任务的名称、顺序、状态、是否并发信息;

动态根据当前的执行状态来决策是执行下一个任务、或者返回结果。

8.根据权利要求1所述的方法,其特征在于,所述在所述任务管理器中采用并行方式执行具有链式关系的异步请求,包括:

通过commit语法同步注册所有要执行的异步事件;

在所述任务管理器的框架内部设置用于执行每个异步事件的异步事件执行器,所述异步事件执行器根据异步事件的状态执行Promise的回调。

9.一种基于Promise的链式异步请求处理装置,其特征在于,包括:

获取模块,用于获取所有的基于Promise的异步请求;

注册模块,用于通过自定义的commit语法在内存中注册所述基于Promise的异步请求所对应的异步事件:通过设置commit函数接收字符串参数;通过字符串匹配的方式在寻址内存中查找与所述字符串参数相同的运行任务;在与所述字符串参数相同的运行任务中选取符合条件的异步任务进行注册;

调用模块,用于调用构建的异步任务管理器,使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务;

传递模块,用于将源文件中记载的所述异步事件的ID及序号传递给所述异步任务管理器,在所述任务管理器中采用并行方式执行具有链式关系的异步请求。

10.一种电子设备,其特征在于,所述电子设备包括:

至少一个处理器;以及,

与所述至少一个处理器通信连接的存储器;其中,

所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行前述权利要求1-8中任一项所述的基于Promise的链式异步请求处理方法。

11.一种非暂态计算机可读存储介质,该非暂态计算机可读存储介质存储计算机指令,该计算机指令用于使该计算机执行前述权利要求1-8中任一项所述的基于Promise的链式异步请求处理方法。

基于Promise的链式异步请求处理方法、装置及电子设备

技术领域

[0001] 本发明涉及数据处理技术领域，尤其涉及一种基于Promise的链式异步请求处理方法、装置及电子设备。

背景技术

[0002] 近年来，随着互联网技术的广泛应用，越来越多的用户在计算机、智能设备之类的计算机设备上浏览由诸如HTML之类的标记语言描述的内容的需求不断增加。HTTP(Hyper Text Transfer Protocol)，即超文本传输协议，是一种无状态、单向的协议。一个HTTP请求被发送到服务器，服务器接受请求并进行处理，完成后发回一个响应，在服务器的处理过程中，客户端和服务器需要保持这个HTTP连接。

[0003] 为了能够适应网页上存在多个HTTP请求，同时提高网页的响应，可以采用HTTP异步请求的方式来进行处理，异步请求可以通过短轮询的方式返回HTTP异步请求响应结果。采用以短轮询方式返回HTTP异步请求响应结果，采用定时向服务器提交新的请求来实现数据的加载，实时效果较差。特别是对于存在多个依赖关系的异步任务，通常多个异步任务之间存在循环依赖关系，即后一个异步任务的执行依赖于前一个异步任务的执行结果。由此导致异步任务的执行效率较慢。

[0004] 在JavaScript中，代码都是单线程方式执行的。由于JavaScript的这个特性，导致JavaScript的所有网络操作，浏览器事件，都必须是异步执行。异步执行都是通过回调函数实现。有时候可能出现非常多次的回调嵌套，这样就使代码的层级特别深，不利于代码的维护和移植。

发明内容

[0005] 有鉴于此，本发明实施例提供一种基于Promise的链式异步请求处理方法、装置及电子设备，至少部分解决现有技术中存在的问题。

[0006] 第一方面，本发明实施例提供了一种基于Promise的链式异步请求处理方法，包括：

[0007] 获取所有的基于Promise的异步请求；

[0008] 通过自定义的commit语法在内存中注册所述基于Promise的异步请求所对应的异步事件；

[0009] 调用构建的异步任务管理器，使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务；

[0010] 将源文件中记载的所述异步事件的ID及序号传递给所述异步任务管理器，在所述任务管理器中采用并行方式执行具有链式关系的异步请求。

[0011] 根据本发明实施例的一种具体实现方式，所述获取所有的基于Promise的异步请求，包括：

[0012] 获取当前页面上的显示内容；

[0013] 判断所述显示内容是否有多个内容源；

[0014] 若是，则基于多个内容源来确定基于Promise的异步请求。

[0015] 根据本发明实施例的一种具体实现方式，所述基于多个内容源来确定基于Promise的异步请求，包括：

[0016] 查找与所述多个内容源对应的多个内容请求；

[0017] 从多个内容请求中选择基于具有链式异步请求关系的内容请求作为基于Promise的异步请求。

[0018] 根据本发明实施例的一种具体实现方式，所述通过自定义的commit语法在内存中注册所述基于Promise的异步请求所对应的异步事件，包括：

[0019] 通过设置commit函数接收字符串参数；

[0020] 通过字符串匹配的方式在寻址内存中查找与所述字符串参数相同的运行任务；

[0021] 在与所述字符串参数相同的运行任务中选取符合条件的异步任务进行注册。

[0022] 根据本发明实施例的一种具体实现方式，所述在与所述字符串参数相同的运行任务中选取符合条件的异步任务进行注册，包括：

[0023] 判断与所述字符串参数相同的运行任务是否是是异步任务；

[0024] 若是，则在当前没有调用中的异步任务的情况下执行当前任务，并在执行结束后寻找内存中是否有暂存未调用的任务集合；

[0025] 若执行结束后寻找内存中存有暂存未调用的任务集合，则调用并重复当前过程，若当前有调用中的任务，则将未调用的任务集合存入异步任务调用集合，等待当前执行任务结束后再调用。

[0026] 根据本发明实施例的一种具体实现方式，所述在与所述字符串参数相同的运行任务中选取符合条件的异步任务进行注册，还包括：

[0027] 当所有的异步任务都调用结束后，进一步判断当前调用结束时是否有全局的错误处理或成功处理；

[0028] 若存在，则返回错误结果，否则，直接返回处理结果。

[0029] 根据本发明实施例的一种具体实现方式，所述构建异步任务管理器，使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务，包括：

[0030] 在内存中将所需要用到的异步任务先行注册；

[0031] 通过框架对所述异步任务进行封装，保证在执行过程中的任务符合框架内Promise异步规范的格式；

[0032] 在所述异步任务在框架初始化的时完成注册。

[0033] 根据本发明实施例的一种具体实现方式，所述构建异步任务管理器，使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务，还包括：

[0034] 在所述异步任务管理器被调用过程中，接收调用异步任务的名称、顺序、状态、是否并发信息；

[0035] 动态根据当前的执行状态来决策是执行下一个任务、或者返回结果。

[0036] 根据本发明实施例的一种具体实现方式，所述在所述任务管理器中采用并行方式执行具有链式关系的异步请求，包括：

[0037] 通过commit语法同步注册所有要执行的异步事件；

[0038] 在所述任务管理器的框架内部设置用于执行每个异步事件的异步事件执行器，所述异步事件执行器根据异步事件的状态执行Promise的回调。

[0039] 第二方面，本发明实施例提供了一种基于Promise的链式异步请求处理装置，包括：

[0040] 获取模块，用于获取所有的基于Promise的异步请求；

[0041] 注册模块，用于通过自定义的commit语法在内存中注册所述基于Promise的异步请求所对应的异步事件；

[0042] 调用模块，用于调用构建的异步任务管理器，使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务；

[0043] 传递模块，用于将源文件中记载的所述异步事件的ID及序号传递给所述异步任务管理器，在所述任务管理器中采用并行方式执行具有链式关系的异步请求。

[0044] 第三方面，本发明实施例还提供了一种电子设备，该电子设备包括：

[0045] 至少一个处理器；以及，

[0046] 与该至少一个处理器通信连接的存储器；其中，

[0047] 该存储器存储有可被该至少一个处理器执行的指令，该指令被该至少一个处理器执行，以使该至少一个处理器能够执行前述任第一方面或第一方面的任一实现方式中的基于Promise的链式异步请求处理方法。

[0048] 第四方面，本发明实施例还提供了一种非暂态计算机可读存储介质，该非暂态计算机可读存储介质存储计算机指令，该计算机指令用于使该计算机执行前述第一方面或第一方面的任一实现方式中的基于Promise的链式异步请求处理方法。

[0049] 第五方面，本发明实施例还提供了一种计算机程序产品，该计算机程序产品包括存储在非暂态计算机可读存储介质上的计算程序，该计算机程序包括程序指令，当该程序指令被计算机执行时，使该计算机执行前述第一方面或第一方面的任一实现方式中的基于Promise的链式异步请求处理方法。

[0050] 本发明实施例中的基于Promise的链式异步请求处理方案，包括获取所有的基于Promise的异步请求；通过自定义的commit语法在内存中注册所述基于Promise的异步请求所对应的异步事件；调用构建的异步任务管理器，使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务；将源文件中记载的所述异步事件的ID及序号传递给所述异步任务管理器，在所述任务管理器中采用并行方式执行具有链式关系的异步请求。通过本申请的处理方案，使得异步任务执行代码的结构更清晰，异步任务之间的依赖更明确，并且对于每一个异步任务不需要增加额外的代码，不需要手动增加Promise封装，使得异步任务迁移成本较低更加便于执行和维护。

附图说明

[0051] 为了更清楚地说明本发明实施例的技术方案，下面将对实施例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的前提下，还可以根据这些附图获得其它的附图。

[0052] 图1为本发明实施例提供的一种基于Promise的链式异步请求处理流程示意图；

[0053] 图2为本发明实施例提供的另一种基于Promise的链式异步请求处理流程示意图；

[0054] 图3为本发明实施例提供的另一种基于Promise的链式异步请求处理流程示意图；

[0055] 图4为本发明实施例提供的另一种基于Promise的链式异步请求处理流程示意图；

[0056] 图5为本发明实施例提供的基于Promise的链式异步请求处理装置结构示意图；

[0057] 图6为本发明实施例提供的电子设备示意图。

具体实施方式

[0058] 下面结合附图对本发明实施例进行详细描述。

[0059] 以下通过特定的具体实例说明本公开的实施方式，本领域技术人员可由本说明书所揭露的内容轻易地了解本公开的其他优点与功效。显然，所描述的实施例仅仅是本公开一部分实施例，而不是全部的实施例。本公开还可以通过另外不同的具体实施方式加以实施或应用，本说明书中的各项细节也可以基于不同观点与应用，在没有背离本公开的精神下进行各种修饰或改变。需说明的是，在不冲突的情况下，以下实施例及实施例中的特征可以相互组合。基于本公开中的实施例，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，都属于本公开保护的范围。

[0060] 需要说明的是，下文描述在所附权利要求书的范围内的实施例的各种方面。应显而易见，本文中所描述的方面可体现于广泛多种形式中，且本文中所描述的任何特定结构及/或功能仅为说明性的。基于本公开，所属领域的技术人员应了解，本文中所描述的一个方面可与任何其它方面独立地实施，且可以各种方式组合这些方面中的两者或两者以上。举例来说，可使用本文中所阐述的任何数目个方面来实施设备及/或实践方法。另外，可使用除了本文中所阐述的方面中的一或多者之外的其它结构及/或功能性实施此设备及/或实践此方法。

[0061] 还需要说明的是，以下实施例中所提供的图示仅以示意方式说明本公开的基本构想，图式中仅显示与本公开中有关的组件而非按照实际实施时的组件数目、形状及尺寸绘制，其实际实施时各组件的型态、数量及比例可为一种随意的改变，且其组件布局型态也可能更为复杂。

[0062] 另外，在以下描述中，提供具体细节是为了便于透彻理解实例。然而，所属领域的技术人员将理解，可在没有这些特定细节的情况下实践所述方面。

[0063] 本公开实施例提供一种基于Promise的链式异步请求处理方法。本实施例提供的基于Promise的链式异步请求处理方法可以由一计算装置来执行，该计算装置可以实现为软件，或者实现为软件和硬件的组合，该计算装置可以集成设置在服务器、终端设备等中。

[0064] 参见图1，本发明实施例提供的一种基于Promise的链式异步请求处理方法，包括如下步骤：

[0065] S101，获取所有的基于Promise的异步请求。

[0066] 在同一个页面中存在多种依赖关系的调用对象，而通过promise能够对异步编程进行抽象，通过采用Promise作为代理对象，能够代表一个需要进行异步处理的函数返回的值或抛出的异常。Promise对象能够代表一个异步操作，可以将异步对象和回调函数脱离开来。

[0067] 以含有JavaScript语言的网页为例，对于网页来讲，通常含有较多的显示内容(例如，文字，图片，表格等)，为了获得当前页面上的异步请求，可以首先获取当前页面上的所

有显示内容,不同的内容对应与不同的来源地址,基于页面上的内容可以判断该显示内容是否有多个内容源。

[0068] 当显示内容来自于多个内容源时,便可以基于多个内容源来确定基于Promise的异步请求。具体的,可以查找与所述多个内容源对应的多个内容请求,通过判断多个内容请求之间是否存在依赖关系,若多个请求之间存在依赖关系,则从多个内容请求中选择基于具有链式异步请求关系的内容请求作为基于Promise的异步请求。

[0069] S102,通过自定义的commit语法在内存中注册所述基于Promise的异步请求所对应的异步事件。

[0070] 传统的基于Promise的异步调用仍然会存在代码调用层次较深的问题。为此,利用自定义的commit语法来解决这个问题。通过commit语法定义commit函数,commit函数能够接收与异步调用关系相关的字符串参数,接收到字符串参数之后,通过字符串匹配的方式,在当前设备的寻址内存中基于相同命名的方法,选择与字符串匹配的任务,并进一步判断该任务是否是一个异步任务。

[0071] 如果该任务非异步任务,则直接调用并返回结果。如果该任务是异步任务,并且当前没有调用中的异步任务,则执行当前任务。

[0072] 当一个异步任务执行结束后,继续寻找内存中是否有暂存未调用的任务集合,若有则调用并重复执行判断任务是否是异步任务的过程。如果当前有调用中的任务,则存入异步任务调用集合,等待当前执行任务结束后调用。当所有的异步任务都调用结束后,判断当前调用结束时是否有全局的错误处理与成功处理,有则将结果返回给相应的处理,无则将结果直接返回。

[0073] S103,调用构建的异步任务管理器,使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务。

[0074] 为了使Promise对应的异步任务更加规范,设定异步任务管理器,通过异步任务管理器管理异步任务,能使异步任务之间的调用关系更加的平行化。

[0075] 具体的,首先内存中将所需要用到的异步任务先行注册,并且通过自定义的异步任务器框架对异步任务进行封装,这样可以保证在执行过程中的任务符合框架内的基本格式,即符合Promise异步规范的格式,使得任务本身在框架初始化的时候已经注册成功。在异步任务被调用的过程中,接收指令为调用异步任务的名称、顺序、状态、是否并发等条件,从而在框架内按照要求对任务进行执行,并动态根据当前的执行状态来决策是执行下一个任务、或者返回结果。

[0076] S104,将源文件中记载的所述异步事件的ID及序号传递给所述异步任务管理器,在所述任务管理器中采用并行方式执行具有链式关系的异步请求。

[0077] 在设置完成异步任务管理器之后,便可以通过异步任务管理器接收和管理针对异步任务的调用。具体的,可以异步事件的ID及序号传递给所述异步任务管理器,异步任务管理器通过异步事件的ID及序号来并行执行具有链式关系的异步请求。从而避免了由于异步任务之间由于存在较深的调用关系,导致任务执行中出现的回调层次加深问题。

[0078] 参见图2,根据本发明实施例的一种具体实现方式,步骤S101获取所有的基于Promise的异步请求,可以包括如下步骤:

[0079] S201,获取当前页面上的显示内容。

[0080] 以含有JavaScript语言的网页为例,对于网页来讲,通常含有较多的显示内容(例如,文字,图片,表格等),为了获得当前页面上的异步请求,可以首先获取当前页面上的所有显示内容。

[0081] S202,判断所述显示内容是否有多个内容源。

[0082] 不同的内容对应与不同的来源地址,基于页面上的内容的来源地址便可以判断该显示内容是否有多个内容源。

[0083] S203,若是,则基于多个内容源来确定基于Promise的异步请求。

[0084] 步骤S203在实现的过程中,具体可以包括:

[0085] S2031,查找与所述多个内容源对应的多个内容请求;

[0086] S2032,从多个内容请求中选择基于具有链式异步请求关系的内容请求作为基于Promise的异步请求。

[0087] 当显示内容来自于多个内容源时,便可以基于多个内容源来确定基于Promise的异步请求。具体的,可以查找与所述多个内容源对应的多个内容请求,通过判断多个内容请求之间是否存在依赖关系,若多个请求之间存在依赖关系,则从多个内容请求中选择基于具有链式异步请求关系的内容请求作为基于Promise的异步请求。

[0088] 参见图3,根据本发明实施例的一种具体实现方式,步骤S102通过自定义的commit语法在内存中注册所述基于Promise的异步请求所对应的异步事件,可以包括如下步骤:

[0089] S301,通过设置commit函数接收字符串参数。

[0090] 传统的基于Promise的异步调用仍然会存在代码调用层次较深的问题。为此,利用自定义的commit语法来解决这个问题。通过commit语法定义commit函数,commit函数能够接收与异步调用关系相关的字符串参数。

[0091] S302,通过字符串匹配的方式在寻址内存中查找与所述字符串参数相同的运行任务。

[0092] 接收到字符串参数之后,通过字符串匹配的方式,在当前设备的寻址内存中基于相同命名的方法,选择与字符串匹配的任务,并进一步判断该任务是否是一个异步任务。

[0093] S303,在与所述字符串参数相同的运行任务中选取符合条件的异步任务进行注册。

[0094] 如果该任务非异步任务,则直接调用并返回结果。如果该任务是异步任务,并且当前没有调用中的异步任务,则执行当前任务。

[0095] 当一个异步任务执行结束后,继续寻找内存中是否有暂存未调用的任务集合,若有则调用并重复执行判断任务是否是异步任务的过程。如果当前有调用中的任务,则存入异步任务调用集合,等待当前执行任务结束后调用。当所有的异步任务都调用结束后,判断当前调用结束时是否有全局的错误处理与成功处理,有则将结果返回给相应的处理,无则将结果直接返回。

[0096] 参见图4,根据本发明实施例的一种具体实现方式,步骤S103构建异步任务管理器,使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务,包括:

[0097] S401,在内存中将所需要用到的异步任务先行注册。

[0098] 具体的,可以将异步任务所对应的对象设置为key-value格式的对象,构建链表散列的数据结构,该数据结构中包含数组和链表,通过该数据结构在内存中存储异步任务的

对象。

[0099] S402,通过框架对所述异步任务进行封装,保证在执行过程中的任务符合框架内Promise异步规范的格式。

[0100] 通过所述框架对异步任务相关的函数、类的封装起来,对外只提供一个接口。通过将异步任务按照Promise异步规范进行封装之后,当需要调用不同的异步任务时,只需要调用异步任务相关的参数即可。

[0101] S403,在所述异步任务在框架初始化的时完成注册。

[0102] 处理步骤S401～S403之外,根据本发明实施例的一种具体实现方式,所述构建异步任务管理器,使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务,还可以包括:在所述异步任务管理器被调用过程中,接收调用异步任务的名称、顺序、状态、是否并发信息;动态根据当前的执行状态来决策是执行下一个任务、或者返回结果。

[0103] 与上面的方法实施例相对应,参见图5,本发明实施例还公开了一种基于Promise的链式异步请求处理装置50,包括:

[0104] 获取模块501,用于获取所有的基于Promise的异步请求。

[0105] 在同一个页面中存在多种依赖关系的调用对象,而通过promise能够对异步编程进行抽象,通过采用Promise作为代理对象,能够代表一个需要进行异步处理的函数返回的值或抛出的异常。Promise对象能够代表一个异步操作,可以将异步对象和回调函数脱离开来。

[0106] 以含有JavaScript语言的网页为例,对于网页来讲,通常含有较多的显示内容(例如,文字,图片,表格等),为了获得当前页面上的异步请求,可以首先获取当前页面上的所有显示内容,不同的内容对应与不同的来源地址,基于页面上的内容可以判断该显示内容是否有多个内容源。

[0107] 当显示内容来自于多个内容源时,便可以基于多个内容源来确定基于Promise的异步请求。具体的,可以查找与所述多个内容源对应的多个内容请求,通过判断多个内容请求之间是否存在依赖关系,若多个请求之间存在依赖关系,则从多个内容请求中选择基于具有链式异步请求关系的内容请求作为基于Promise的异步请求。

[0108] 注册模块502,用于通过自定义的commit语法在内存中注册所述基于Promise的异步请求所对应的异步事件。

[0109] 传统的基于Promise的异步调用仍然会存在代码调用层次较深的问题。为此,利用自定义的commit语法来解决这个问题。通过commit语法定义commit函数,commit函数能够接收与异步调用关系相关的字符串参数,接收到字符串参数之后,通过字符串匹配的方式,在当前设备的寻址内存中基于相同命名的方法,选择与字符串匹配的任务,并进一步判断该任务是否是一个异步任务。

[0110] 如果该任务非异步任务,则直接调用并返回结果。如果该任务是异步任务,并且当前没有调用中的异步任务,则执行当前任务。

[0111] 当一个异步任务执行结束后,继续寻找内存中是否有暂存未调用的任务集合,若有则调用并重复执行判断任务是否是异步任务的过程。如果当前有调用中的任务,则存入异步任务调用集合,等待当前执行任务结束后调用。当所有的异步任务都调用结束后,判断当前调用结束时是否有全局的错误处理与成功处理,有则将结果返回给相应的处理,无则

将结果直接返回。

[0112] 调用模块503,用于调用构建的异步任务管理器,使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务。

[0113] 为了使Promise对应的异步任务更加规范,设定异步任务管理器,通过异步任务管理器管理异步任务,能使异步任务之间的调用关系更加的平行化。

[0114] 具体的,首先内存中将所需要用到的异步任务先行注册,并且通过自定义的异步任务器框架对异步任务进行封装,这样可以保证在执行过程中的任务符合框架内的基本格式,即符合Promise异步规范的格式,使得任务本身在框架初始化的时候已经注册成功。在异步任务被调用的过程中,接收指令为调用异步任务的名称、顺序、状态、是否并发等条件,从而在框架内按照要求对任务进行执行,并动态根据当前的执行状态来决策是执行下一个任务、或者返回结果。

[0115] 传递模块504,用于将源文件中记载的所述异步事件的ID及序号传递给所述异步任务管理器,在所述任务管理器中采用并行方式执行具有链式关系的异步请求。

[0116] 在设置完成异步任务管理器之后,便可以通过异步任务管理器接收和管理针对异步任务的调用。具体的,可以异步事件的ID及序号传递给所述异步任务管理器,异步任务管理器通过异步事件的ID及序号来并行执行具有链式关系的异步请求。从而避免了由于异步任务之间由于存在较深的调用关系,导致任务执行中出现的回调层次加深问题。

[0117] 图5所示装置可以对应的执行上述方法实施例中的内容,本实施例未详细描述的部分,参照上述方法实施例中记载的内容,在此不再赘述。

[0118] 参见图6,本发明实施例还提供了一种电子设备60,该电子设备包括:

[0119] 至少一个处理器;以及,

[0120] 与该至少一个处理器通信连接的存储器;其中,

[0121] 该存储器存储有可被该至少一个处理器执行的指令,该指令被该至少一个处理器执行,以使该至少一个处理器能够执行前述方法实施例中基于Promise的链式异步请求处理方法。

[0122] 本发明实施例还提供了一种非暂态计算机可读存储介质,该非暂态计算机可读存储介质存储计算机指令,该计算机指令用于使该计算机执行前述方法实施例中。

[0123] 本发明实施例还提供了一种计算机程序产品,该计算机程序产品包括存储在非暂态计算机可读存储介质上的计算程序,该计算机程序包括程序指令,当该程序指令被计算机执行时,使该计算机执行前述方法实施例中的基于Promise的链式异步请求处理方法。

[0124] 下面参考图6,其示出了适于用来实现本公开实施例的电子设备60的结构示意图。本公开实施例中的电子设备可以包括但不限于诸如移动电话、笔记本电脑、数字广播接收器、PDA(个人数字助理)、PAD(平板电脑)、PMP(便携式多媒体播放器)、车载终端(例如车载导航终端)等等的移动终端以及诸如数字TV、台式计算机等等的固定终端。图6示出的电子设备仅仅是一个示例,不应对本公开实施例的功能和使用范围带来任何限制。

[0125] 如图6所示,电子设备60可以包括处理装置(例如中央处理器、图形处理器等)601,其可以根据存储在只读存储器(ROM)602中的程序或者从存储装置608加载到随机访问存储器(RAM)603中的程序而执行各种适当的动作和处理。在RAM 603中,还存储有电子设备60操作所需的各种程序和数据。处理装置601、ROM 602以及RAM 603通过总线604彼此相连。输

入/输出(I/O)接口605也连接至总线604。

[0126] 通常,以下装置可以连接至I/O接口605:包括例如触摸屏、触摸板、键盘、鼠标、图像传感器、麦克风、加速度计、陀螺仪等的输入装置606;包括例如液晶显示器(LCD)、扬声器、振动器等的输出装置607;包括例如磁带、硬盘等的存储装置608;以及通信装置609。通信装置609可以允许电子设备60与其他设备进行无线或有线通信以交换数据。虽然图中示出了具有各种装置的电子设备60,但是应理解的是,并不要求实施或具备所有示出的装置。可以替代地实施或具备更多或更少的装置。

[0127] 特别地,根据本公开的实施例,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,本公开的实施例包括一种计算机程序产品,其包括承载在计算机可读介质上的计算机程序,该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装置609从网络上被下载和安装,或者从存储装置608被安装,或者从ROM 602被安装。在该计算机程序被处理装置601执行时,执行本公开实施例的方法中限定的上述功能。

[0128] 需要说明的是,本公开上述的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本公开中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本公开中,计算机可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读信号介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:电线、光缆、RF(射频)等等,或者上述的任意合适的组合。

[0129] 上述计算机可读介质可以是上述电子设备中所包含的;也可以是单独存在,而未装配入该电子设备中。

[0130] 上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被该电子设备执行时,使得该电子设备:获取至少两个网际协议地址;向节点评价设备发送包括所述至少两个网际协议地址的节点评价请求,其中,所述节点评价设备从所述至少两个网际协议地址中,选取网际协议地址并返回;接收所述节点评价设备返回的网际协议地址;其中,所获取的网际协议地址指示内容分发网络中的边缘节点。

[0131] 或者,上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被该电子设备执行时,使得该电子设备:接收包括至少两个网际协议地址的节点评价请求;从所述至少两个网际协议地址中,选取网际协议地址;返回选取出的网际协议地址;其中,接收到的网际协议地址指示内容分发网络中的边缘节点。

[0132] 可以以一种或多种程序设计语言或其组合来编写用于执行本公开的操作的计算机程序代码，上述程序设计语言包括面向对象的程序设计语言—诸如Java、Smalltalk、C++，还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络——包括局域网(LAN)或广域网(WAN)—连接到用户计算机，或者，可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0133] 附图中的流程图和框图，图示了按照本公开各种实施例的系统、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，该模块、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。也要注意的是，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。

[0134] 描述于本公开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。其中，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，第一获取单元还可以被描述为“获取至少两个网际协议地址的单元”。

[0135] 应当理解，本发明的各部分可以用硬件、软件、固件或它们的组合来实现。

[0136] 以上所述，仅为本发明的具体实施方式，但本发明的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本发明揭露的技术范围内，可轻易想到的变化或替换，都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应以权利要求的保护范围为准。

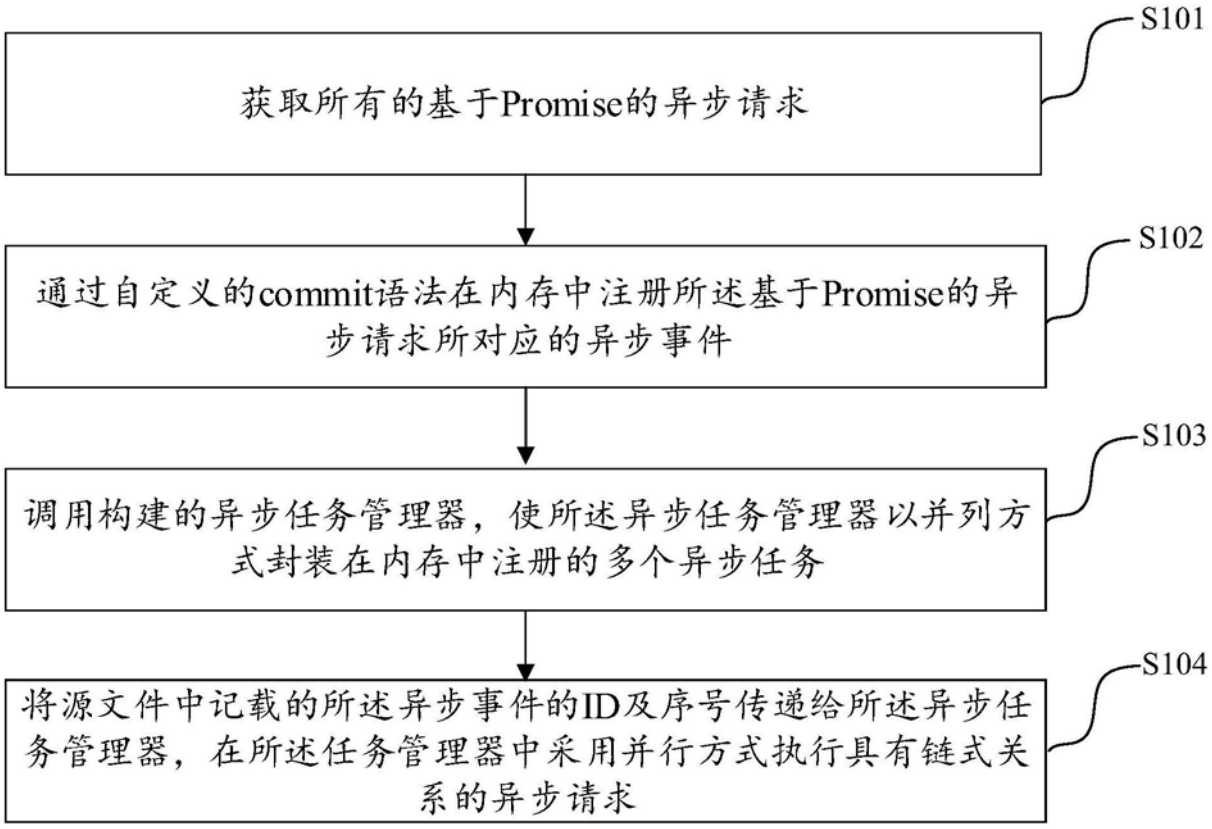

S101
获取所有的基于Promise的异步请求
S102
通过自定义的commit语法在内存中注册所述基于Promise的异步请求所对应的异步事件
S103
调用构建的异步任务管理器，使所述异步任务管理器以并列方式封装在内存中注册的多个异步任务
S104
将源文件中记载的所述异步事件的ID及序号传递给所述异步任务管理器，在所述任务管理器中采用并行方式执行具有链式关系的异步请求

图1

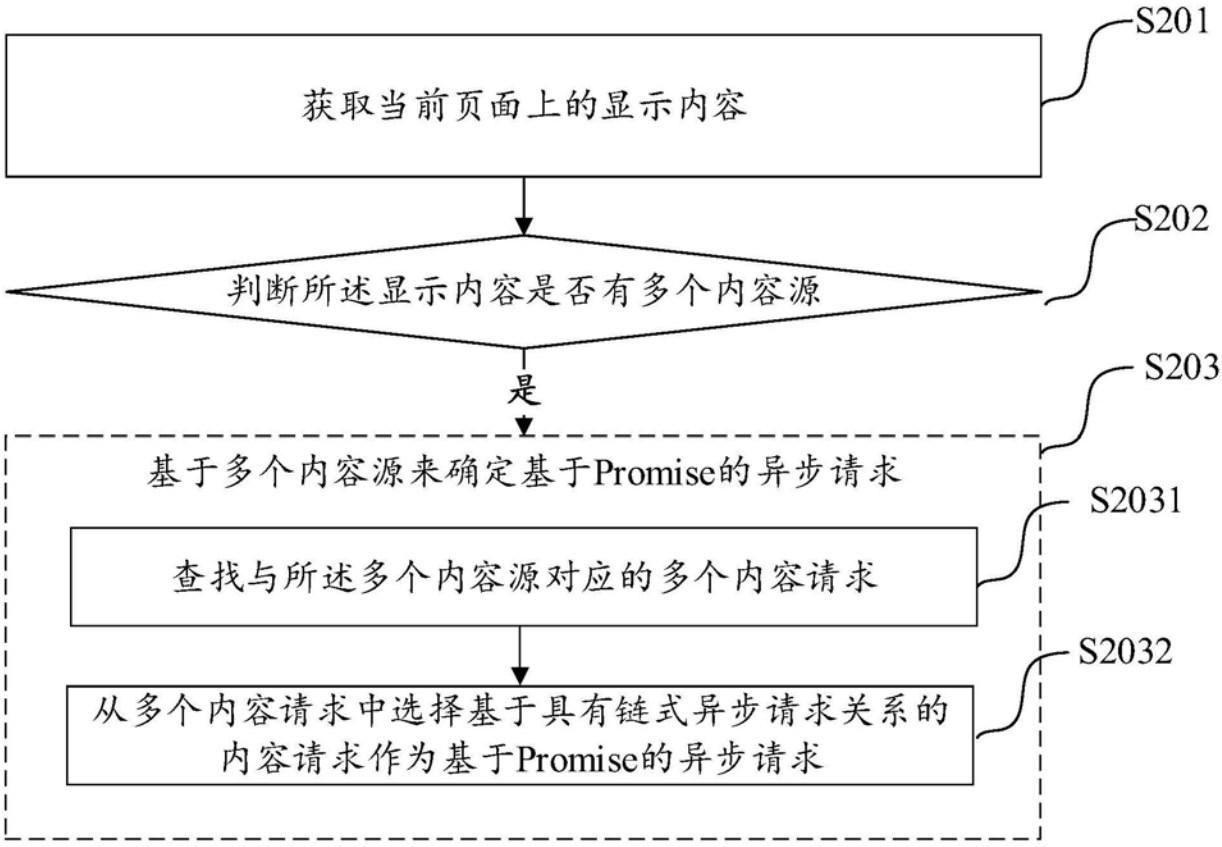

S201
获取当前页面上的显示内容
S202
判断所述显示内容是否有多个内容源
是
S203
基于多个内容源来确定基于Promise的异步请求
S2031
查找与所述多个内容源对应的多个内容请求
S2032
从多个内容请求中选择基于具有链式异步请求关系的内容请求作为基于Promise的异步请求

图2

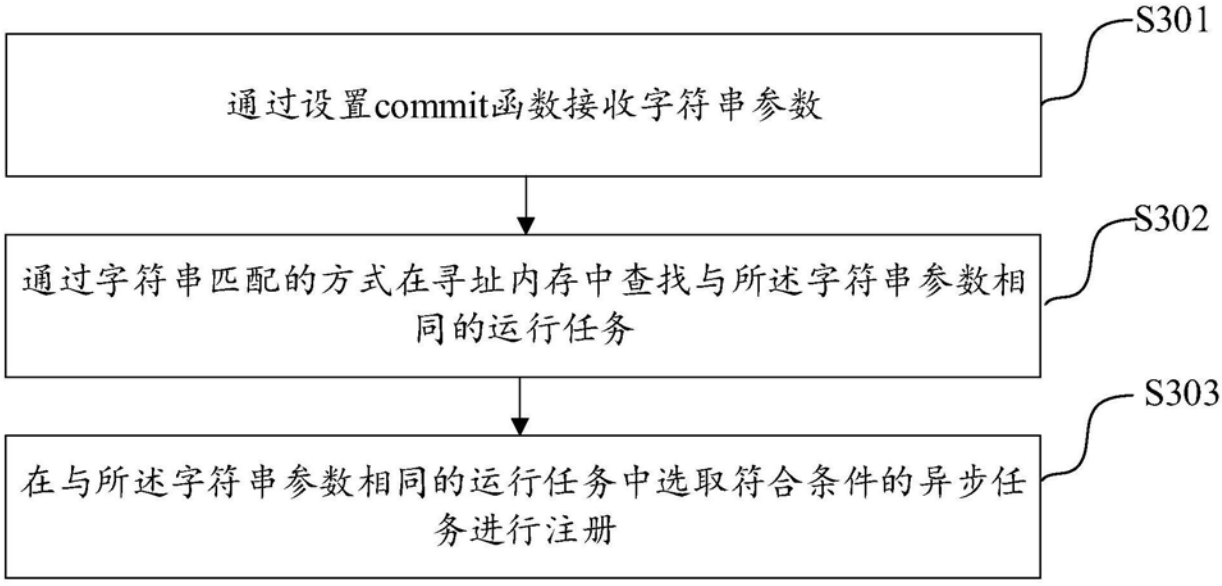

S301
通过设置commit函数接收字符串参数
S302
通过字符串匹配的方式在寻址内存中查找与所述字符串参数相同的运行任务
S303
在与所述字符串参数相同的运行任务中选取符合条件的异步任务进行注册

图3

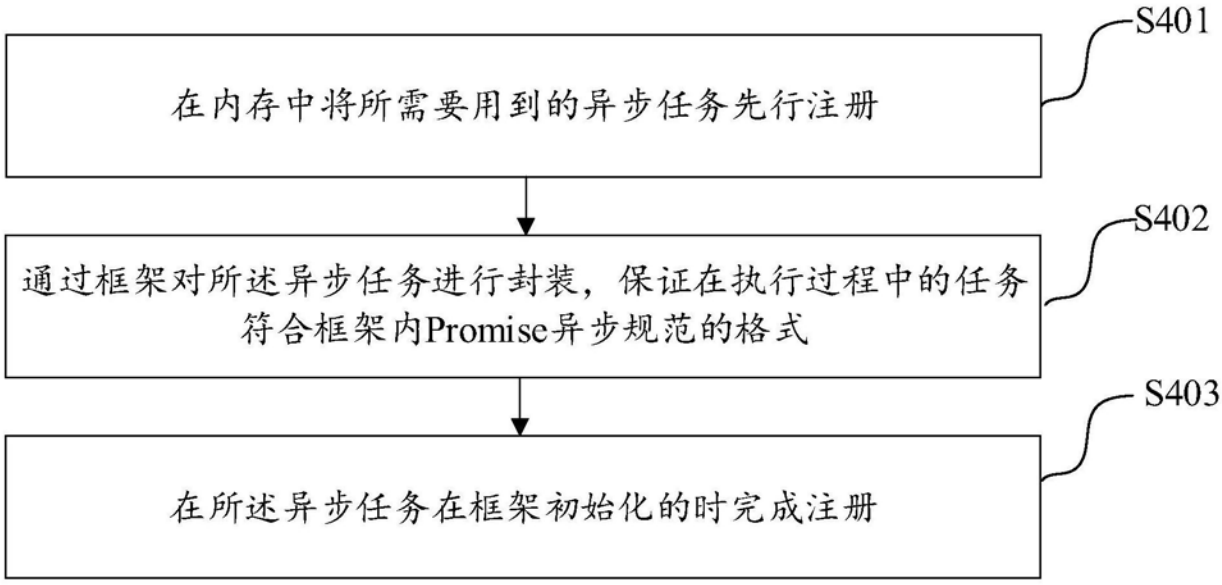

S401
在内存中将所需要用到的异步任务先行注册
S402
通过框架对所述异步任务进行封装，保证在执行过程中的任务符合框架内Promise异步规范的格式
S403
在所述异步任务在框架初始化的时完成注册

图4

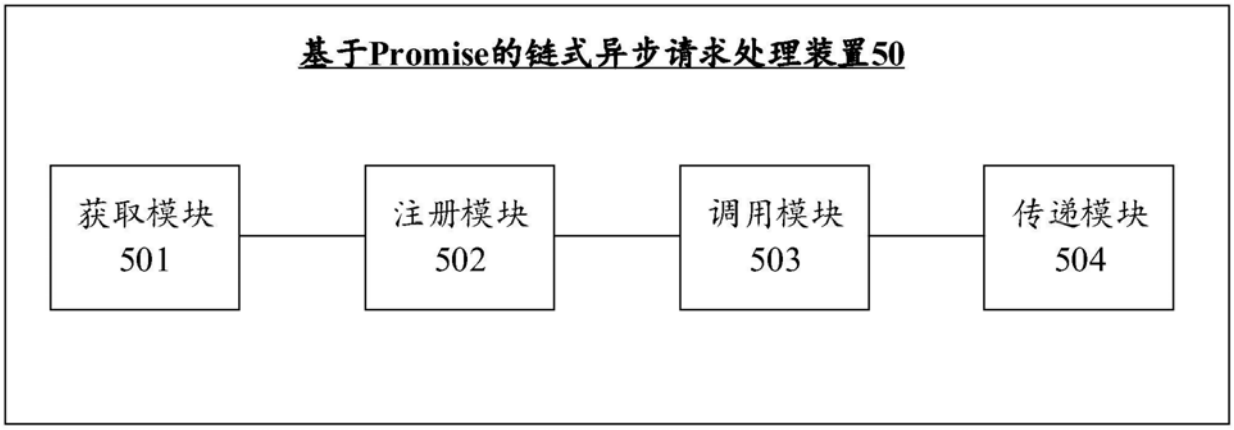

基于Promise的链式异步请求处理装置50
获取模块 501
注册模块 502
调用模块 503
传递模块 504

图5

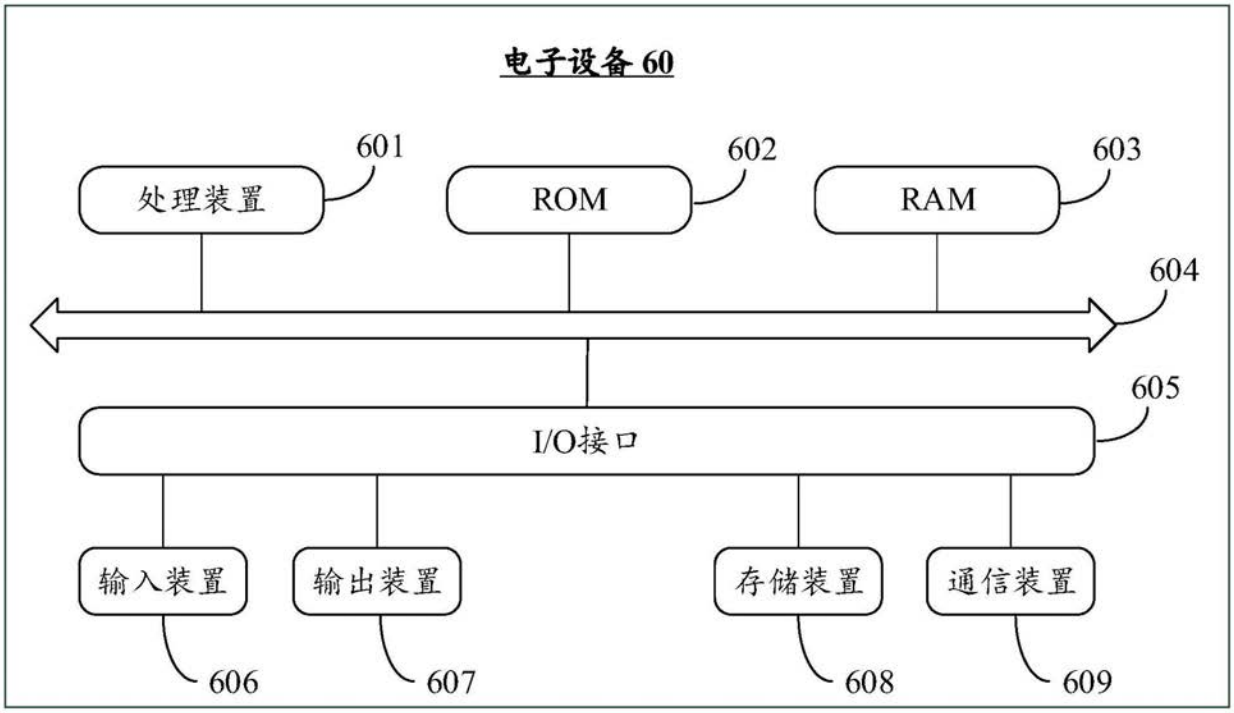

电子设备 60
601
处理装置
602
ROM
603
RAM
604
605
I/O接口
输入装置
输出装置
存储装置
通信装置
606
607
608
609

图6