



(12) 发明专利

(10) 授权公告号 CN 114079653 B

(45) 授权公告日 2023. 10. 20

(21) 申请号 202111365213.0

H04L 67/56 (2022.01)

(22) 申请日 2021.11.17

H04L 69/16 (2022.01)

(65) 同一申请的已公布的文献号

申请公布号 CN 114079653 A

(56) 对比文件

CN 113312168 A, 2021.08.27

CN 111382381 A, 2020.07.07

(43) 申请公布日 2022.02.22

CN 111813385 A, 2020.10.23

(73) 专利权人 抖音视界有限公司

CN 112527284 A, 2021.03.19

地址 100041 北京市石景山区实兴大街30

US 2017242665 A1, 2017.08.24

号院3号楼2层B-0035房间

解定东.Web版VS Code的自主搭建与应用. 科学技术创新.2020,(第29期),第1-2页.

(72) 发明人 杨吉明

审查员 陈露

(74) 专利代理机构 北京中知法苑知识产权代理

有限公司 11226

专利代理师 李明

(51) Int. Cl.

H04L 67/02 (2022.01)

H04L 67/141 (2022.01)

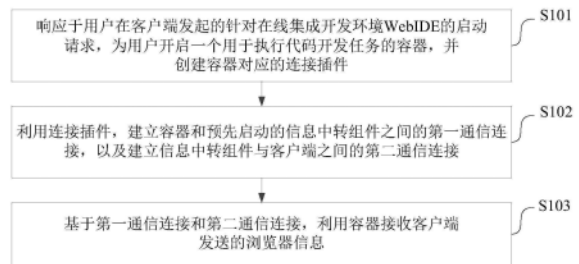
权利要求书3页 说明书15页 附图3页

(54) 发明名称

一种通信方法、装置、计算机设备和存储介质

(57) 摘要

本公开提供了一种通信方法、装置、计算机设备和存储介质,其中,该方法包括:响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,为用户开启一个用于执行代码开发任务的容器,并创建容器对应的连接插件;利用连接插件,建立容器和预先启动的信息中转组件之间的第一通信连接,以及建立信息中转组件与客户端之间的第二通信连接;基于第一通信连接和第二通信连接,利用容器接收客户端发送的浏览器信息。本公开实施例能够实现服务端容器和客户端之间的通信。



1. 一种通信方法,其特征在于,应用于服务端,包括:

响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件;

将所述容器的容器信息存储到目标局域网址对应的存储位置;所述容器信息中包括所述用户的用户身份信息;

利用所述连接插件,建立所述容器和预先启动的信息中转组件之间的第一通信连接,以及建立所述信息中转组件与所述客户端之间的第二通信连接;

基于所述第一通信连接和第二通信连接,利用所述容器接收所述客户端发送的浏览器信息;以及

利用所述第一通信连接,通过所述容器向所述信息中转组件发送通信信息;

利用所述信息中转组件基于所述容器对应的目标局域网址,获取所述容器的容器信息;不同的容器对应于不同的目标局域网址,信息中转组件对应于一个目标局域网址;

在确定所述容器信息中存储有用户身份信息的情况下,确定所述容器身份验证通过;并在确定所述容器身份验证通过的情况下,基于所述第二通信连接,将所述通信信息发送给所述客户端。

2. 根据权利要求1所述的方法,其特征在于,在所述为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件和信息中转组件之前,还包括:

响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,对所述用户进行权限验证;

在所述用户通过权限验证的情况下,将所述WebIDE对应的启动代码发送给所述客户端,以使所述客户端基于所述启动代码创建消息连接组件;

所述建立所述信息中转组件与所述客户端之间的第二通信连接,包括:

响应于客户端发起的建立所述消息连接组件和所述信息中转组件之间的连接请求,建立所述消息连接组件和所述信息中转组件之间的第二通信连接。

3. 根据权利要求1所述的方法,其特征在于,所述建立所述信息中转组件与所述客户端之间的第二通信连接,包括:

在所述客户端中打开的所述WebIDE对应的窗口包括多个的情况下,建立所述信息中转组件与所述客户端的每个窗口之间的第二通信连接;

所述基于所述第二通信连接,将所述通信信息发送给所述客户端,包括:

基于与每个所述窗口分别对应的第二通信连接,将所述通信信息发送给所述客户端中的每个所述窗口。

4. 根据权利要求1所述的方法,其特征在于,所述基于所述第一通信连接和第二通信连接,利用所述容器接收客户端发送的浏览器信息,包括:

利用所述信息中转组件,通过第二通信连接获取所述客户端发送的浏览器信息;

基于所述浏览器信息中用户的用户身份信息,从预先启动的多个容器中筛选出所述用户对应的容器;

基于筛选出的容器对应的第一通信连接,将浏览器信息中除所述用户身份信息以外的其他浏览器信息发送给所述用户对应的容器。

5. 一种通信方法,其特征在于,应用于客户端,包括:

获取服务端发送的WebIDE对应的启动代码,并基于所述启动代码,创建消息连接组件;

向服务端发起建立所述消息连接组件和信息中转组件之间的连接连接请求,以使所述服务端基于所述连接请求,建立所述消息连接组件和所述信息中转组件之间的第二通信连接;

利用所述第二通信连接,向所述服务端中的信息中转组件发送浏览器信息,以使所述信息中转组件将浏览器信息发送至所述服务端对应的容器;以及

利用所述第二通信连接,获取所述服务端在确定所述容器身份验证通过的情况下,通过所述信息中转组件发送的通信信息;所述通信信息为所述服务端利用所述容器和所述信息中转组件之间的第一通信连接,通过所述容器向所述信息中转组件发送的;所述容器身份验证通过是所述服务端在确定所述容器的容器信息中存储有用户身份信息的情况下确定的;所述容器信息为利用所述信息中转组件基于所述容器对应的目标局域网址获取的,所述容器信息为所述服务端在创建所述容器对应的连接插件后,存储到目标局域网址对应的存储位置的;不同的容器对应于不同的目标局域网址,信息中转组件对应于一个目标局域网址。

6. 一种通信装置,其特征在于,应用于服务端,包括:

第一创建模块,用于响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件;

存储模块,用于将所述容器的容器信息存储到目标局域网址对应的存储位置;所述容器信息中包括所述用户的用户身份信息;

连接建立模块,用于利用所述连接插件,建立所述容器和预先启动的信息中转组件之间的第一通信连接,以及建立所述信息中转组件与所述客户端之间的第二通信连接;

通信模块,用于基于所述第一通信连接和第二通信连接,利用所述容器接收所述客户端发送的浏览器信息;以及

利用所述第一通信连接,通过所述容器向所述信息中转组件发送通信信息;

利用所述信息中转组件基于所述容器对应的目标局域网址,获取所述容器的容器信息;不同的容器对应于不同的目标局域网址,信息中转组件对应于一个目标局域网址;

在确定所述容器信息中存储有用户身份信息的情况下,确定所述容器身份验证通过;并在确定所述容器身份验证通过的情况下,基于所述第二通信连接,将所述通信信息发送给所述客户端。

7. 一种通信装置,其特征在于,应用于客户端,包括:

第二创建模块,用于获取服务端发送的WebIDE对应的启动代码,并基于所述启动代码,创建消息连接组件;

第一发送模块,用于向服务端发起建立所述消息连接组件和信息中转组件之间的连接连接请求,以使所述服务端基于所述连接请求,建立所述消息连接组件和所述信息中转组件之间的第二通信连接;

第二发送模块,用于利用所述第二通信连接,向所述服务端中的信息中转组件发送浏览器信息,以使所述信息中转组件将浏览器信息发送至所述服务端对应的容器;

所述第二创建模块,还用于利用所述第二通信连接,获取所述服务端在确定所述容器

身份验证通过的情况下,通过所述信息中转组件发送的通信信息;所述通信信息为所述服务端利用所述容器和所述信息中转组件之间的第一通信连接,通过所述容器向所述信息中转组件发送的;所述容器身份验证通过是所述服务端在确定所述容器的容器信息中存储有用户身份信息的情况下确定的;所述容器信息为利用所述信息中转组件基于所述容器对应的目标局域网址获取的,所述容器信息为所述服务端在创建所述容器对应的连接插件后,存储到目标局域网址对应的存储位置的;不同的容器对应于不同的目标局域网址,信息中转组件对应于一个目标局域网址。

8. 一种计算机设备,其特征在于,包括:处理器、存储器,所述存储器存储有所述处理器可执行的机器可读指令,所述处理器用于执行所述存储器中存储的机器可读指令,所述机器可读指令被所述处理器执行时,所述处理器执行如权利要求1至5任意一项所述的通信方法的步骤。

9. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被计算机设备运行时,所述计算机设备执行如权利要求1至5任意一项所述的通信方法的步骤。

一种通信方法、装置、计算机设备和存储介质

技术领域

[0001] 本公开涉及计算机技术领域,具体而言,涉及一种通信方法、装置、计算机设备和存储介质。

背景技术

[0002] WebIDE作为一种针对编程开发人员的在线集成开发环境,其具有编程人员无需本地安装开发环境,只需打开浏览器就能立即开发的优越特性。WebIDE对应的页面中展示有各种服务端对应的插件程序,这些插件程序通常需要运行在服务端的容器中形成插件。

[0003] 在一些情况下,插件在服务端的容器中运行时,需要获取一些特定的浏览器信息,例如,cookie信息、本地存储信息等等。但现有技术中,由于插件无法与浏览器通信,因而无法及时的获取需要的浏览器信息。

发明内容

[0004] 本公开实施例至少提供一种通信方法、装置、计算机设备和存储介质。

[0005] 第一方面,本公开实施例提供了一种通信方法,包括:

[0006] 响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件;

[0007] 利用所述连接插件,建立所述容器和预先启动的信息中转组件之间的第一通信连接,以及建立所述信息中转组件与所述客户端之间的第二通信连接;

[0008] 基于所述第一通信连接和第二通信连接,利用所述容器接收所述客户端发送的浏览器信息。

[0009] 在一种可能的实施方式中,所述方法还包括:

[0010] 利用所述第一通信连接,通过所述容器向所述信息中转组件发送通信信息;

[0011] 基于所述第二通信连接,将所述通信信息通过所述信息中转组件发送给所述客户端,以使所述客户端基于所述通信信息进行响应。

[0012] 在一种可能的实施方式中,在所述为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件和信息中转组件之前,还包括:

[0013] 响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,对所述用户进行权限验证;

[0014] 在所述用户通过权限验证的情况下,将所述WebIDE对应的启动代码发送给所述客户端,以使所述客户端基于所述启动代码创建消息连接组件;

[0015] 所述建立所述信息中转组件与所述客户端之间的第二通信连接,包括:

[0016] 响应于客户端发起的建立所述消息连接组件和所述信息中转组件之间的连接请求,建立所述消息连接组件和所述信息中转组件之间的第二通信连接。

[0017] 在一种可能的实施方式中,在为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件之后,还包括:

[0018] 将所述容器的容器信息存储到目标局域网址对应的存储位置;所述容器信息中包括所述用户的用户身份信息;

[0019] 所述基于所述第二通信连接,将所述通信信息通过所述信息中转组件发送给所述客户端,包括:

[0020] 基于所述容器对应的目标局域网址,获取所述容器的容器信息;

[0021] 在确定所述容器信息中存储有用户身份信息的情况下,确定所述容器身份验证通过;并在确定所述容器身份验证通过的情况下,基于所述第二通信连接,将所述通信信息发送给所述客户端。

[0022] 在一种可能的实施方式中,所述建立所述信息中转组件与所述客户端之间的第二通信连接,包括:

[0023] 在所述客户端中打开的所述WebIDE对应的窗口包括多个的情况下,建立所述信息中转组件与所述客户端的每个窗口之间的第二通信连接;

[0024] 所述基于所述第二通信连接,将所述通信信息通过所述信息中转组件发送给所述客户端,包括:

[0025] 基于与每个所述窗口分别对应的第二通信连接,将所述通信信息发送给所述客户端中的每个所述窗口。

[0026] 在一种可能的实施方式中,所述基于所述第一通信连接和第二通信连接,利用所述容器接收客户端发送的浏览器信息,包括:

[0027] 利用所述信息中转组件,通过第二通信连接获取所述客户端发送的浏览器信息;

[0028] 基于所述浏览器信息中用户的用户身份信息,从预先启动的多个容器中筛选出所述用户对应的容器;

[0029] 基于筛选出的容器对应的第一通信连接,将浏览器信息中除所述用户身份信息以外的其他浏览器信息发送给所述用户对应的容器。

[0030] 第二方面,本公开实施例还提供一种通信方法,包括:

[0031] 获取服务端发送的WebIDE对应的启动代码,并基于所述启动代码,创建消息连接组件;

[0032] 向服务端发起建立所述消息连接组件和信息中转组件之间的连接请求,以使所述服务端基于所述连接请求,建立所述消息连接组件和所述信息中转组件之间的第二通信连接;

[0033] 利用所述第二通信连接,向所述服务端中的信息中转组件发送浏览器信息,以使所述信息中转组件将浏览器信息发送至所述服务端对应的容器。

[0034] 第三方面,本公开实施例提供一种通信装置,包括:

[0035] 第一创建模块,用于响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件;

[0036] 连接建立模块,用于利用所述连接插件,建立所述容器和预先启动的信息中转组件之间的第一通信连接,以及建立所述信息中转组件与所述客户端之间的第二通信连接;

[0037] 通信模块,用于基于所述第一通信连接和第二通信连接,利用所述容器接收所述客户端发送的浏览器信息。

[0038] 在一种可能的实施方式中,所述通信模块,还用于利用所述第一通信连接,通过所述容器向所述信息中转组件发送通信信息;

[0039] 基于所述第二通信连接,将所述通信信息通过所述信息中转组件发送给所述客户端,以使所述客户端基于所述通信信息进行响应。

[0040] 在一种可能的实施方式中,所述装置还包括验证模块:

[0041] 所述验证模块,用于在所述为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件和信息中转组件之前,响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,对所述用户进行权限验证;

[0042] 在所述用户通过权限验证的情况下,将所述WebIDE对应的启动代码发送给所述客户端,以使所述客户端基于所述启动代码创建消息连接组件;

[0043] 所述连接建立模块,用于响应于客户端发起的建立所述消息连接组件和所述信息中转组件之间的连接连接请求,建立所述消息连接组件和所述信息中转组件之间的第二通信连接。

[0044] 在一种可能的实施方式中,所述装置还包括存储模块:

[0045] 所述存储模块,用于在为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件之后,将所述容器的容器信息存储到目标局域网址对应的存储位置;所述容器信息中包括所述用户的用户身份信息;

[0046] 所述通信模块,用于基于所述容器对应的目标局域网址,获取所述容器的容器信息;

[0047] 在确定所述容器信息中存储有用户身份信息的情况下,确定所述容器身份验证通过;并在确定所述容器身份验证通过的情况下,基于所述第二通信连接,将所述通信信息发送给所述客户端。

[0048] 在一种可能的实施方式中,所述连接建立模块,用于在所述客户端中打开的所述WebIDE对应的窗口包括多个的情况下,建立所述信息中转组件与所述客户端的每个窗口之间的第二通信连接;

[0049] 所述通信模块,用于基于与每个所述窗口分别对应的第二通信连接,将所述通信信息发送给所述客户端中的每个所述窗口。

[0050] 在一种可能的实施方式中,所述通信模块,用于利用所述信息中转组件,通过第二通信连接获取所述客户端发送的浏览器信息;

[0051] 基于所述浏览器信息中用户的用户身份信息,从预先启动的多个容器中筛选出所述用户对应的容器;

[0052] 基于筛选出的容器对应的第一通信连接,将浏览器信息中除所述用户身份信息以外的其他浏览器信息发送给所述用户对应的容器。

[0053] 第四方面,本公开实施例还提供了一种通信,包括:

[0054] 第二创建模块,用于获取服务端发送的WebIDE对应的启动代码,并基于所述启动代码,创建消息连接组件;

[0055] 第一发送模块,用于向服务端发起建立所述消息连接组件和信息中转组件之间的连接连接请求,以使所述服务端基于所述连接请求,建立所述消息连接组件和所述信息中转组件之间的第二通信连接;

[0056] 第二发送模块,用于利用所述第二通信连接,向所述服务端中的信息中转组件发送浏览器信息,以使所述信息中转组件将浏览器信息发送至所述服务端对应的容器。

[0057] 第五方面,本公开可选实现方式还提供一种计算机设备,处理器、存储器,所述存储器存储有所述处理器可执行的机器可读指令,所述处理器用于执行所述存储器中存储的机器可读指令,所述机器可读指令被所述处理器执行时,所述机器可读指令被所述处理器执行时执行上述第一方面,或第一方面中任一种可能的实施方式中的步骤;或者,所述机器可读指令被所述处理器执行时执行上述第二方面。

[0058] 第六方面,本公开可选实现方式还提供一种计算机可读存储介质,该计算机可读存储介质上存储有计算机程序,该计算机程序被运行时执行上述第一方面,或第一方面中任一种可能的实施方式中的步骤,或者,该计算机程序被运行时执行上述第二方面。

[0059] 关于上述通信装置、计算机设备、及计算机可读存储介质的效果描述参见上述通信方法的说明,这里不再赘述。

[0060] 本公开实施例提供的通信方法、装置、计算机设备和存储介质,基于创建的信息中转组件和容器之间的第一通信连接,可以实现容器和信息中转组件之间的通信;基于创建的信息中转组件和客户端之间的第二通信连接,可以实现信息中转组件和客户端之间的通信;进而,利用预先创建的信息中转组件进行信息中转,能够实现服务端容器和客户端的通信,也即能够实现运行在容器中的插件和客户端的通信,从而能够及时获取客户端对应的本地信息,有效解决了现有技术中插件无法和客户端通信的问题。

[0061] 为使本公开的上述目的、特征和优点能更明显易懂,下文特举较佳实施例,并配合所附附图,作详细说明如下。

附图说明

[0062] 为了更清楚地说明本公开实施例的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,此处的附图被并入说明书中并构成本说明书中的一部分,这些附图示出了符合本公开的实施例,并与说明书一起用于说明本公开的技术方案。应当理解,以下附图仅示出了本公开的某些实施例,因此不应被看作是对范围的限定,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相关的附图。

[0063] 图1示出了本公开实施例所提供的一种通信方法的流程图;

[0064] 图2示出了本公开实施例所提供的另一种通信方法的流程图;

[0065] 图3示出了本公开实施例所提供的一种服务端和客户端的交互示意图;

[0066] 图4示出了本公开实施例所提供的一种通信装置的示意图;

[0067] 图5示出了本公开实施例所提供的另一种通信装置的示意图;

[0068] 图6示出了本公开实施例所提供的一种计算机设备结构示意图。

具体实施方式

[0069] 为使本公开实施例的目的、技术方案和优点更加清楚,下面将结合本公开实施例中附图,对本公开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公开一部分实施例,而不是全部的实施例。通常在此处描述和示出的本公开实施例的组件可以以各种不同的配置来布置和设计。因此,以下对本公开的实施例的详细描述并非

旨在限制要求保护的公开的公开的范围,而是仅仅表示公开的选定实施例。基于公开的实施例,本领域技术人员在没有做出创造性劳动的前提下所获得的所有其他实施例,都属于公开保护的范围。

[0070] 另外,公开实施例中的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的实施例能够以除了在这里图示或描述的内容以外的顺序实施。

[0071] 在本文中提及的“多个或者若干个”是指两个或两个以上。“和/或”,描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。字符“/”一般表示前后关联对象是一种“或”的关系。

[0072] 经研究发现,WebIDE作为一种针对编程开发人员的在线集成开发环境,其具有编程人员无需本地安装开发环境,只需打开浏览器就能立即开发的优越特性。针对WebIDE开发,需要在服务端为用户创建的容器中进行,并且大多时候还需要利用服务端对应的插件进行进一步的开发。但是在现有技术中,WebIDE中的各种核心插件对应的API(Application Programming Interface,应用程序接口)均没有提供与客户端进行通信的方法,也即无法实现客户端与运行在服务端中的插件的通信。因此,运行在服务端的插件也无法及时获取一些特定的浏览器信息。

[0073] 基于上述研究,公开提供了一种通信方法、装置、计算机设备和存储介质,基于创建的信息中转组件和容器之间的第一通信连接,可以实现容器和信息中转组件之间的通信;基于创建的信息中转组件和客户端之间的第二通信连接,可以实现信息中转组件和客户端之间的通信;进而,利用预先创建的信息中转组件进行信息中转,能够实现服务端容器和客户端的通信,也即能够实现运行在容器中的插件和客户端的通信,从而能够及时获取客户端对应的本地信息,有效解决了现有技术中插件无法和客户端通信的问题。

[0074] 针对以上方案所存在的缺陷,均是发明人在经过实践并仔细研究后得出的结果,因此,上述问题的发现过程以及下文中公开针对上述问题所提出的解决方案,都应该是发明人在本公开过程中对本公开做出的贡献。

[0075] 应注意到:相似的标号和字母在下面的附图中表示类似项,因此,一旦某一项在一个附图中被定义,则在随后的附图中不需要对其进行进一步定义和解释。

[0076] 需要说明的是,公开实施例中所提到的特定名词包括:

[0077] WebIDE:一种针对编程开发人员的在线集成开发环境,其具有编程人员无需本地安装开发环境,只需打开浏览器就能立即开发的优越特性;

[0078] WebSocket是一种在单个TCP连接上进行全双工通信的协议;

[0079] Cookie:储存在用户本地终端上的数据,在本公开实施例中,可以为存储在浏览器中的数据;

[0080] B/S结构:Browser/Server,浏览器/服务器结构,是WEB兴起后的一种网络结构模式,WEB浏览器是客户端最主要的应用软件;

[0081] WEB:World Wide Web,即全球广域网,也称为万维网;

[0082] IP:Internet Protocol,网际互连协议,是TCP/IP体系中的网络层协议;

[0083] IPC:Inter-Process Communication,进程间通信,进程间通信是指两个进程的数

据之间产生交互；

[0084] Unix Domain Socket:称为Unix域套接字,简称UDS,是基于套接字Socket API的基础上发展而来的,Socket API适用于不同机器(如客户端和服务端)上进程间的通讯;

[0085] HTTP:Hyper Text Transfer Protocol,超文本传输协议,是一个简单的请求-响应协议;

[0086] message queue:消息队列,是一种消息处理技术,它可以在任何安装了Microsoft Windows的计算机组合中,为任何应用程序提供消息处理和消息队列功能,无论这些计算机是否在同一网络上或者是否同时联机;

[0087] 文件共享方式:是指主动地在网络上共享自己的计算机文件的方式。

[0088] 为便于对本实施例进行理解,首先对本公开实施例所公开的一种通信方法进行详细介绍,本公开实施例所提供的通信方法的执行主体一般为具有一定计算能力的计算机设备,在一些可能的实现方式中,该通信方法可以通过处理器调用存储器中存储的计算机可读指令的方式来实现。

[0089] 下面以执行主体为服务端为例对本公开实施例提供的通信方法加以说明。

[0090] 如图1所示,为本公开实施例提供的一种通信方法的流程图,可以包括以下步骤:

[0091] S101:响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,为用户开启一个用于执行代码开发任务的容器,并创建容器对应的连接插件。

[0092] 这里,WebIDE是一种针对编程开发人员的在线集成开发环境,其具有编程人员无需本地安装开发环境,只需打开浏览器就能立即开发的优越特性。其中,本公开实施例所提及的用户的均可以为编程开发人员,本公开实施例所提供的客户端均可以为浏览器,本公开实施例所提及的服务端均可以为WebIDE对应的服务端。

[0093] 容器为服务端创建的,用于执行代码开发任务,一个代码开发任务可以对应于一个容器,一个容器一旦启动将一直存在,且一直用于其对应的代码开发任务的开发,直至其对应的代码开发任务结束,该容器将会被摧毁,该容器对应的第一通信连接也将删除。具体的,容器可以为Docker容器。

[0094] 代码开发任务为用户指定的用于开发目标代码的任务,目标代码可以为目标程序、目标应用对应的代码,为具有特定功能的代码。具体的,如果一个代码开发任务已经对应存在一个已经开启好的容器,那么在再次进行该代码任务的开发时可以直接利用该开启好的容器继续进行开发,不需要新开启一个容器;反之,如果一个代码开发任务为新的任务,不存在已经开启好的容器,则需要为该代码开发任务创建一个新的容器,以用于代码开发。

[0095] 启动请求为用户在客户端发起的用于启动WebIDE的请求,具体实施时,当用户在客户端中登录WebIDE时,客户端可以基于用户的登录请求,生成启动请求并发给WebIDE对应的服务端。

[0096] 连接插件为一种能够和预先创建好的消息中转组件进行通信连接的插件,能够运行在服务端开启的容器中。预先创建好的消息中转组件在服务端中运行,能够用于对来自客户端的信息和来自服务端的信息进行中转,例如,将来自客户端的信息发送至服务端。

[0097] 本步骤在具体实施时,当用户需要在WebIDE进行代码开发时,也即需要在WebIDE中执行代码开发任务时,可以针对该代码开发任务,在浏览器中登录WebIDE,之后,浏览器

可以生成针对WebIDE的启动请求并发送给WebIDE对应的服务端。服务端在接收到启动请求之后,可以响应于该启动请求,为该用户开启一个用于执行代码开发任务的容器。

[0098] 并且,服务端在成功开启容器之后,可以在该容器中创建一个连接插件,以用于后续和浏览器进行通信。另外,可以在该容器中运行服务端对应的至少一个插件程序,其中,运行的插件程序可以是代码开发任务相关的插件程序。

[0099] S102:利用连接插件,建立容器和预先启动的信息中转组件之间的第一通信连接,以及建立信息中转组件与客户端之间的第二通信连接。

[0100] 这里,第一通信连接可以用于代码开发任务对应的容器和预先创建的信息中转组件之间的通信。具体的,服务端中可以只包括一个信息中转组件,服务端创建的任一个容器对应的连接插件,都可以与该信息中转组件建立连接,并利用与该信息中转组件之间的连接,实现和信息中转组件之间的通信。

[0101] 具体的,信息中转组件可以为一个添加有消息中转服务的、预先开启的WebIDE调度器,基于该消息中转服务,WebIDE调度器可以实现对来自客户端的信息和来自服务端的信息的中转。

[0102] 示例性的,在利用S101得到的代码开发任务对应的容器以及容器对应的连接插件之后,可以利用连接插件,建立容器和信息中转组件之间第一通信连接,具体的,第一通信连接可以为任一具有双向通信功能的进程间通信连接,例如,http连接。

[0103] 并且,还可以建立信息中转组件和客户端之间的第二通信连接,其中,第二通信连接也可以为任一具有双向通信功能的进程间通信连接,具体的,第二通信连接可以为WebSocket连接。针对建立第二通信连接的操作和建立第一通信连接的操作,在具体实施时,两个操作之间不存在严格的先后顺序,此处不进行严格的限定。例如,可以先建立第二通信连接,再建立第一通信连接,或者,可以同时建立第一通信连接和第二通信连接。

[0104] S103:基于第一通信连接和第二通信连接,利用容器接收客户端发送的浏览器信息。

[0105] 这里,浏览器信息可以为浏览器的本地信息,包括但不限于cookies、本地存储localStorage等信息。

[0106] 具体实施时,客户端(即浏览器)可以通过建立的第二通信连接,将服务端需要的本地存储的浏览器信息发送给信息中转组件;或者,客户端也可以通过第二通信连接,主动向信息中转组件发送本地存储的浏览器信息。

[0107] 然后,在信息中转组件接收到客户端发送的浏览器信息之后,可以基于建立的第一通信连接,将浏览器信息发送给容器,由于容器运行在服务端,所以将浏览器信息发送给容器也即将浏览器信息发送至服务端,这样,可以实现客户端和浏览器端的通信。进而,服务端可以基于获取的浏览器信息进行响应,例如,基于获取的浏览器信息对插件进行更新、复制等等。

[0108] 另外,服务端还可以基于已经建立的任一第一通信连接以及与该第一通信连接相匹配的第二通信连接,利用该第一通信连接对应的容器,获取与该第一通信连接相匹配的第二通信连接对应的客户端、利用消息连接组件发送的浏览器信息。

[0109] 这样,基于创建的信息中转组件和容器之间的第一通信连接,可以实现容器和信息中转组件之间的通信;基于创建的信息中转组件和客户端之间的第二通信连接,可以实

现信息中转组件和客户端之间的通信;进而,利用预先创建的信息中转组件进行信息中转,能够实现服务端容器和客户端的通信,也即能够实现运行在容器中的插件和客户端的通信,从而能够及时获取客户端对应的本地信息,有效解决了现有技术中插件无法和客户端通信的问题。

[0110] 在一种实施例中,在创建容器对应的连接插件和信息中转组件之前,还包括:

[0111] 响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,对用户进行权限验证。

[0112] 这里,在响应于用户在客户端发起的启动请求时,还可以对用户进行权限验证。具体的,可以利用运行在服务端的用户身份鉴权服务,对用户进行权限验证。

[0113] 具体实施时,用户在登录WebIDE时,可以输入用户身份信息,例如,用户名信息和用户密码信息等等,之后,客户端可以将用户身份信息发送至服务端,服务端可以利用鉴定模块中添加的用户身份鉴权服务,基于用户身份信息,对用户进行权限验证。例如,可以基于用户身份信息中的用户名信息,确定在服务端预先存储的、已授权的用户身份信息对应的用户名信息中,是否存在与该用户名信息相匹配的已授权的用户名信息;如果有,则可以确定该用户认证通过权限验证,反之,则可以确定该用户未通过权限验证。又例如,还可以在确定存在与该用户名信息相匹配的已授权的用户名信息的情况下,确定用户输入的用户身份信息中的用户密码信息、和与该用户名信息相匹配的已授权的用户名信息对应的已授权的用户身份信息中的用户密码信息是否一致,如果是,确定该用户未通过权限验证,反之,也确定该用户未通过权限验证。

[0114] 进一步的,可以在用户通过权限验证的情况下,将WebIDE对应的启动代码发送给客户端,以使客户端基于启动代码创建消息连接组件。

[0115] 这里,启动代码可以为启动WebIDE以及加载WebIDE对应的WebIDE页面的代码。消息连接组件为客户端创建的、可运行在客户端中的、用于和服务端进行通信的组件,任一客户端均可以在用户通过权限验证的情况下,创建对应的消息连接组件。

[0116] 具体实施时,在确定用户通过权限验证的情况下,服务端可以将WebIDE对应的启动代码发送给客户端。之后,客户端可以在接收到启动代码之后,执行启动代码并对代码执行结果进行渲染,从而得到WebIDE页面。

[0117] 并且,在启动代码中添加有用于创建消息连接组件的子代码,客户端在执行启动代码的过程中,将执行该子代码,从而完成对消息连接组件的创建。

[0118] 之后,针对S102中的建立信息中转组件与客户端之间的第二通信连接的步骤,客户端可以在创建得到消息连接组件之后,向服务端发起来连接请求,具体的,客户端可以向运行在服务端中的信息中转组件发起建立消息连接组件和信息中转组件之间的连接连接请求。

[0119] 然后,信息中转组件可以在接收到连接请求之后,响应于该连接请求,建立和消息连接组件之间的第二通信连接。其中,建立的第二通信连接可以为支持B/S的双向通信连接,具体的,第二通信连接可以为如上所述的WebSocket连接。

[0120] 另外,在用户未通过权限验证的情况下,服务端可以直接停止对用户发起的启动请求的响应,拒绝此次对WebIDE的启动。

[0121] 在一种实施例中,在建立第一通信连接和第二通信连接之后,服务端还可以主动

向客户端发送通信信息。其中,服务端主动发送的通信信息可以包括查找客户端对应的至少部分浏览器信息的信息,例如,查找浏览器信息中存储的各个授权用户名信息、存储的各个任务ID信息、数据库信息,又例如,改变客户端对应的WebIDE页面的页面高度、页面位置等等,以及通信信息还可以包括服务端主动发送的通知信息,例如,容器异常信息、服务端异常信息、页面刷新信息等等。

[0122] 具体实施时,服务端可以利用容器和信息中转组件之间的第一通信连接,将通信信息发送至信息中转组件。然后,服务端可以进一步地基于该第一通信连接对应的第二通信连接,通过信息中转组件将该通信信息发送至该第二通信连接对应的消息连接组件,由于消息连接组件运行在客户端中,将通信信息发送至消息连接组件,也即将通知信息发送至客户端。

[0123] 之后,客户端在接收到通信信息之后,可以对该通信信息进行响应。例如,在通信信息为服务端发送的改变客户端对应的WebIDE页面的页面高度的情况下,客户端可以在接收到通信信息之后,基于该通信信息对应的页面高度和WebIDE页面当前的页面高度,对WebIDE页面的页面高度进行调整。

[0124] 在一种实施例中,在创建了容器对应的连接插件之后,还可以将创建的容器对应的容器信息存储到目标局域网址对应的存储位置。其中,容器信息可以包括但不限于容器对应的用户的用户身份信息,例如,还可以包括容器对应的创建时间信息、容器对应的任务ID信息等等。目标局域网址具体可以为一个局域网IP,不同的容器对应于不同的目标局域网址,信息中转组件也对应于一个目标局域网址。这里,服务端可以将创建的每个容器和信息中转组件组成一个局域网络,信息中转组件和各个容器分别对应于一个局域网IP。

[0125] 具体实施时,在确定创建的容器对应的容器信息之后,可以确定该容器对应的目标局域网址,进而,可以确定该目标局域网址对应的存储位置并将该容器信息存储在该存储位置。

[0126] 进一步的,针对基于第二通信连接,将通信信息通过信息中转组件发送给客户端的步骤,服务端可以利用容器和第一通信连接,将通信信息发送至信息中转组件,并且,还可以基于信息中转组件对应的目标局域网址,调度信息中转组件的API,以实现通过信息中转组件将通信信息发送至客户端。但信息中转组件在接收到通信信息之后,需要先确定发送通信信息的容器对应的目标局域网址;之后,基于该目标局域网址,确定容器信息的存储位置。然后,可以先判断确定的该存储位置处是否存储有容器信息,如果是,则获取该容器信息,如果否,则确定该通知信息为无效信息,不对该通知信息进行中转。

[0127] 在获取到存储的容器信息之后,还可以基于容器信息对发送通信信息的容器进行容器身份的验证。具体的,可以确定容器信息中是否存储有用户身份信息,如果是,则确定容器身份验证通过;进而,可以确定与该用户身份信息相匹配的第二通信连接,也即确定该用户身份信息对应的用户打开的每个客户端;最后,可以利用确定的第二通信连接,通过信息中转组件将通信信息发送给客户端对应的消息连接组件,也即实现将通信信息发送给客户端。

[0128] 另外,在确定容器身份验证未通过的情况下,也不对该通知信息进行中转。

[0129] 这样,由于容器一旦建立,其对应的目标局域网址将被确定,通过目标局域网址进行容器身份验证,相比利用密钥等方式进行容器身份验证而言,不会存在密码泄漏的风险,

提高了身份验证的安全性和准确性,并且,目标局域网址作为容器的一个常规属性,利用目标局域网址进行身份验证,降低了身份验证的复杂性。

[0130] 在一种实施例中,针对建立信息中转组件和客户端之间的第二通信连接的步骤,可能存在用户在客户端中同时打开多个WebIDE对应的窗口的情况,也即,可能存在用户同时打开多个浏览器窗口的情况。其中,WebIDE对应的每个窗口中展示的WebIDE页面一致,且可在客户端进行独立编辑,每个窗口对应的用户身份信息相同,但窗口标识信息不同。

[0131] 这样,在客户端中打开的WebIDE对应的窗口包括多个的情况下,可以建立信息中转组件和每个窗口之间的第二通信连接,以实现服务端通过信息中转组件和客户端对应的各个窗口进行独立通信。

[0132] 进一步的,在建立了信息中转组件和每个窗口之间的第二通信连接之后,服务端可以通过信息中转组件,利用与每个窗口对应的第二通信连接,具体的,可以利用与每个窗口对应的WebSocket连接,将通信信息发送至客户端中的每个窗口对应的消息连接器,也即实现将通信信息发送至客户端中的每个窗口。这样,将服务端发送的每个通信信息,同步发送至客户端中的每个窗口,保证了每个窗口都可以及时获取服务端发送的通信信息。

[0133] 例如,服务端可以将目标插件对应的插件信息作为通信信息,利用第一通信连接和第二通信连接,发送至客户端中的每个窗口,进而,客户端中的每个窗口均可以基于通信信息,加载得到目标插件并展示。

[0134] 在一种实施例中,针对S103,可以按照以下步骤实施:

[0135] 步骤一、利用信息中转组件,通过第二通信连接获取客户端发送的浏览器信息。

[0136] 其中,浏览器信息可以包括用户身份信息、cookies、本地存储localStorage等等。另外,发送的浏览器信息还可以包括客户端基于用户对展示的WebIDE页面的调整操作和/或用户对展示的WebIDE页面中的插件的更新操作,生成的更新信息。

[0137] 示例性的,用户对展示的WebIDE页面中的插件的更新操作例如可以为在插件上添加信息的操作,比如添加文字信息和/或添加图片信息等,进而,客户端生成的更新信息可以为插件文字更新信息和/或插件图片更新信息。

[0138] 具体实施时,客户端在确定需要发送的浏览器信息之后,可以利用消息连接组件,通过第二通信连接,将浏览器信息主动发送给信息中转组件。进而,服务端可以利用信息中转组件,通过第二通信连接获取客户端发送的浏览器信息。

[0139] 这样,如果客户端对应的窗口包括多个,针对任一个窗口,可以利用该窗口对应的消息连接组件以及该窗口对应的第二通信连接,将该窗口对应的浏览器信息发送至信息中转组件,其中,这里发送的浏览器信息中还可以包括窗口标识信息,例如,窗口标识号、窗口标识符等等。

[0140] 步骤二、基于浏览器信息中用户的用户身份信息,从预先启动的多个容器中筛选出用户对应的容器。

[0141] 这里,服务端中可以包括多个已经启动的容器,由上述实施例可知,一个容器一旦启动将一直存在,且一直用于其对应的代码开发任务的开发,直至其对应的代码开发任务结束,该容器将会被摧毁,该容器对应的第一通信连接也将删除。而由于代码开发通常需要较长的时间,所以服务端中可以存在多个预先启动的容器,又由于服务端中只存在一个信息中转组件,所以,信息中转组件对应于多个第一通信连接。

[0142] 因此,本步骤在具体实施时,服务端在利用信息中转组件获取到浏览器信息之后,可以确定出浏览器信息中用户的用户身份信息,基于用户身份信息,以及预先启动的每个容器对应的用户身份信息,从预先启动的多个容器中,筛选出发送该浏览器信息的客户端对应的容器,也即,筛选出与该浏览器信息中用户身份信息对应的用户相匹配容器。

[0143] 步骤三、基于筛选出的容器对应的第一通信连接,将浏览器信息中除用户身份信息以外的其他浏览器信息发送给用户对应的容器。

[0144] 具体实施时,可以利用信息中转组件,基于筛选出的容器对应的第一通信连接,将浏览器信息中除用户身份信息以外的其他浏览器信息发送给用户对应的容器,也即实现将浏览器信息中除用户身份信息以外的其他浏览器信息发送给服务端。

[0145] 例如,将浏览器信息中的cookies、本地存储localStorage、更新信息发送给用户对应的容器。

[0146] 在一种实施方式中,将浏览器信息中除用户身份信息以外的其他浏览器信息发送给用户对应的容器的方式可以是任一在Linux环境下的IPC方式。例如,可以为unix domain socket,http,message queue,文件共享等方式中的任一种。

[0147] 另外,在客户端对应的窗口包括多个的情况下,还可以将该窗口对应的窗口标识信息发送给用户对应的容器,这样,可以实现一对一的信息中转,也即,一个窗口利用消息连接组件和第二通信连接发送的浏览器信息对应于一个容器。

[0148] 在一种实施例中,本公开实施例中还提供了一种与运行在服务端的通信方法相对应的、可以运行在客户端的通信方法,如图2所示,为本公开实例所提供的另一种通信方法的流程图,可以包括以下步骤:

[0149] S201:获取服务端发送的WebIDE对应的启动代码,并基于启动代码,创建消息连接组件。

[0150] 这里,由上述实施例可知,服务端在确定用户通过权限验证的情况下,可以将WebIDE对应的启动代码发送给客户端。进而,客户端可以获取到服务端发送的启动代码。

[0151] 之后,可以在接收到启动代码之后,执行启动代码并对代码执行结果进行渲染,从而得到WebIDE页面。

[0152] 并且,在启动代码中添加有用于创建消息连接组件的子代码,客户端在执行启动代码的过程中,将执行该子代码,从而完成对消息连接组件的创建。

[0153] S202:向服务端发起建立消息连接组件和信息中转组件之间的连接连接请求,以使服务端基于连接请求,建立消息连接组件和信息中转组件之间的第二通信连接。

[0154] 这里,客户端可以在创建得到消息连接组件之后,向服务端发起来连接请求,具体的,客户端可以向运行在服务端中的信息中转组件发起建立消息连接组件和信息中转组件之间的连接连接请求。

[0155] 进一步的,服务端在通过信息中转组件可以在接收到连接请求之后,可以响应于该连接请求,建立和消息连接组件之间的第二通信连接。

[0156] 这里,在客户端中包括多个窗口的情况下,服务端可以建立信息中转组件和每个窗口对应的第二通信连接。其中,客户端中包括的每个窗口均可以对应于一个消息连接组件,这样,服务端可以基于接收到的任一窗口对应的消息连接组件发送的连接请求,建立和该窗口对应的第二通信连接。

[0157] 或者,一个客户端可以只对应于一个消息连接组件,在客户端中包括多个窗口的情况下,客户端利用消息连接组件发送的连接请求中可以包括每个窗口对应的窗口标识信息,进而,服务端可以根据连接请求中的每个窗口标识信息,建立和每个窗口对应的第二通信连接。

[0158] S203:利用第二通信连接,向服务端中的信息中转组件发送浏览器信息,以使信息中转组件将浏览器信息发送至服务端对应的容器。

[0159] 这里,浏览器信息可以包括用户身份信息、cookies、本地存储localStorage等等。另外,发送的浏览器信息还可以包括客户端基于用户对展示的WebIDE页面的调整操作和/或用户对展示的WebIDE页面中的插件的更新操作,生成的更新信息。

[0160] 具体实施时,客户端在确定需要发送的浏览器信息之后,可以利用消息连接组件,通过第二通信连接,将浏览器信息主动发送给信息中转组件。进而,信息中转组件可以通过第一通信连接将浏览器信息发送至服务端对应的容器。这里,关于S203的具体实施步骤,可以参照上述实施例,此处不再赘述。

[0161] 如图3所示,为本公开实例所提供的一种服务端和客户端的交互示意图,其中,服务端中可以包括用户身份鉴权服务的鉴定模块,预先创建的信息中转组件,信息中转组件中添加有消息中转服务,以及,服务端还包括容器及容器对应的连接插件。具体的,当用户需要在WebIDE中执行代码开发任务时,可以针对该代码开发任务,在浏览器中登录WebIDE,并在登录时输入用户身份信息。服务端可以利用鉴定模块中的用户身份鉴权服务,基于用户输入的用户身份信息,对用户进行权限验证。在用户通过权限验证的情况下,将WebIDE对应的启动代码发送给客户端(即浏览器),之后,客户端可以执行启动代码,得到消息连接组件。并且,服务端可以在用户通过权限验证的情况下,为用户开启一个用于执行代码开发任务的容器,并创建容器对应的连接插件。

[0162] 之后,服务端可以利用连接插件,建立容器和预先启动的信息中转组件之间的第一通信连接,以及响应于客户端发起的建立消息连接组件和信息中转组件之间的连接的第一通信连接,建立消息连接组件和信息中转组件之间的第二通信连接。

[0163] 然后,在容器通过第一通信连接向信息中转组件发送通信信息时,信息中转组件可以通过容器对应的目标局域网址进行容器身份验证;在确定容器身份验证通过的情况下,通过容器对应的客户端与信息中转组件之间的第二通信连接(具体的,可以利用容器对应的客户端与信息中转组件之间的WebSocket连接),将通信信息发送至客户端中的消息连接组件。

[0164] 客户端可以利用消息连接组件以及对应的第二通信连接(具体可以为WebSocket连接),向信息中转组件发送浏览器信息;进而,信息中转组件可以基于浏览器信息中用户的用户身份信息,从预先启动的多个容器中筛选出用户对应的容器;之后,利用该容器对应的第一通信连接,将浏览器信息中除用户身份信息以外的其他浏览器信息发送给用户对应的容器。

[0165] 本领域技术人员可以理解,在具体实施方式的上述方法中,各步骤的撰写顺序并不意味着严格的执行顺序而对实施过程构成任何限定,各步骤的具体执行顺序应当以其功能和可能的内在逻辑确定。

[0166] 基于同一发明构思,本公开实施例中还提供了与应用在服务端的通信方法对应的

通信装置,由于本公开实施例中的装置解决问题的原理与本公开实施例上述应用在服务端的通信方法相似,因此装置的实施可以参见方法的实施,重复之处不再赘述。

[0167] 如图4所示,为本公开实施例提供的一种通信装置的示意图,包括:

[0168] 第一创建模块401,用于响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件;

[0169] 连接建立模块402,用于利用所述连接插件,建立所述容器和预先启动的信息中转组件之间的第一通信连接,以及建立所述信息中转组件与所述客户端之间的第二通信连接;

[0170] 通信模块403,用于基于所述第一通信连接和第二通信连接,利用所述容器接收所述客户端发送的浏览器信息。

[0171] 在一种可能的实施方式中,所述通信模块403,还用于利用所述第一通信连接,通过所述容器向所述信息中转组件发送通信信息;

[0172] 基于所述第二通信连接,将所述通信信息通过所述信息中转组件发送给所述客户端,以使所述客户端基于所述通信信息进行响应。

[0173] 在一种可能的实施方式中,所述装置还包括验证模块404:

[0174] 所述验证模块404,用于在所述为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件和信息中转组件之前,响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求,对所述用户进行权限验证;

[0175] 在所述用户通过权限验证的情况下,将所述WebIDE对应的启动代码发送给所述客户端,以使所述客户端基于所述启动代码创建消息连接组件;

[0176] 所述连接建立模块402,用于响应于客户端发起的建立所述消息连接组件和所述信息中转组件之间的连接请求,建立所述消息连接组件和所述信息中转组件之间的第二通信连接。

[0177] 在一种可能的实施方式中,所述装置还包括存储模块405:

[0178] 所述存储模块405,用于在为所述用户开启一个用于执行代码开发任务的容器,并创建所述容器对应的连接插件之后,将所述容器的容器信息存储到目标局域网址对应的存储位置;所述容器信息中包括所述用户的用户身份信息;

[0179] 所述通信模块403,用于基于所述容器对应的目标局域网址,获取所述容器的容器信息;

[0180] 在确定所述容器信息中存储有用户身份信息的情况下,确定所述容器身份验证通过;并在确定所述容器身份验证通过的情况下,基于所述第二通信连接,将所述通信信息发送给所述客户端。

[0181] 在一种可能的实施方式中,所述连接建立模块402,用于在所述客户端中打开的所述WebIDE对应的窗口包括多个的情况下,建立所述信息中转组件与所述客户端的每个窗口之间的第二通信连接;

[0182] 所述通信模块403,用于基于与每个所述窗口分别对应的第二通信连接,将所述通信信息发送给所述客户端中的每个所述窗口。

[0183] 在一种可能的实施方式中,所述通信模块403,用于利用所述信息中转组件,通过

第二通信连接获取所述客户端发送的浏览器信息；

[0184] 基于所述浏览器信息中用户的用户身份信息，从预先启动的多个容器中筛选出所述用户对应的容器；

[0185] 基于筛选出的容器对应的第一通信连接，将浏览器信息中除所述用户身份信息以外的其他浏览器信息发送给所述用户对应的容器。

[0186] 基于同一发明构思，本公开实施例中还提供了与应用在客户端的通信方法对应的通信装置，由于本公开实施例中的装置解决问题的原理与本公开实施例上述应用在客户端的通信方法相似，因此装置的实施可以参见方法的实施，重复之处不再赘述。

[0187] 如图5所示，为本公开实施例提供的另一种通信装置的示意图，包括

[0188] 第二创建模块501，用于获取服务端发送的WebIDE对应的启动代码，并基于所述启动代码，创建消息连接组件；

[0189] 第一发送模块502，用于向服务端发起建立所述消息连接组件和信息中转组件之间的连接请求，以使所述服务端基于所述连接请求，建立所述消息连接组件和所述信息中转组件之间的第二通信连接；

[0190] 第二发送模块503，用于利用所述第二通信连接，向所述服务端中的信息中转组件发送浏览器信息，以使所述信息中转组件将浏览器信息发送至所述服务端对应的容器。

[0191] 关于装置中的各模块的处理流程、以及各模块之间的交互流程的描述可以参照上述方法实施例中的相关说明，这里不再详述。

[0192] 本公开实施例还提供了一种计算机设备，如图6所示，为本公开实施例提供的一种计算机设备结构示意图，包括：

[0193] 处理器61和存储器62；所述存储器62存储有处理器61可执行的机器可读指令，处理器61用于执行存储器62中存储的机器可读指令，所述机器可读指令被处理器61执行时，处理器61执行下述步骤：S101：响应于用户在客户端发起的针对在线集成开发环境WebIDE的启动请求，为用户开启一个用于执行代码开发任务的容器，并创建容器对应的连接插件；S102：利用连接插件，建立容器和预先启动的信息中转组件之间的第一通信连接，以及建立信息中转组件与客户端之间的第二通信连接以及S103：基于第一通信连接和第二通信连接，利用容器接收客户端发送的浏览器信息；或者，所述机器可读指令被处理器61执行时，处理器61执行下述步骤：S201：获取服务端发送的WebIDE对应的启动代码，并基于启动代码，创建消息连接组件；S202：向服务端发起建立消息连接组件和信息中转组件之间的连接请求，以使服务端基于连接请求，建立消息连接组件和信息中转组件之间的第二通信连接以及S203：利用第二通信连接，向服务端中的信息中转组件发送浏览器信息，以使信息中转组件将浏览器信息发送至服务端对应的容器。

[0194] 上述存储器62包括内存621和外部存储器622；这里的内存621也称内存储器，用于暂时存放处理器61中的运算数据，以及与硬盘等外部存储器622交换的数据，处理器61通过内存621与外部存储器622进行数据交换。

[0195] 上述指令的具体执行过程可以参考本公开实施例中所述的通信方法的步骤，此处不再赘述。

[0196] 本公开实施例还提供一种计算机可读存储介质，该计算机可读存储介质上存储有计算机程序，该计算机程序被处理器运行时执行上述方法实施例中所述的通信方法的步

骤。其中,该存储介质可以是易失性或非易失的计算机可读取存储介质。

[0197] 本公开实施例所提供的通信方法的计算机程序产品,包括存储了程序代码的计算机可读存储介质,所述程序代码包括的指令可用于执行上述方法实施例中所述的通信方法的步骤,具体可参见上述方法实施例,在此不再赘述。

[0198] 该计算机程序产品可以具体通过硬件、软件或其结合的方式实现。在一个可选实施例中,所述计算机程序产品具体体现为计算机存储介质,在另一个可选实施例中,计算机程序产品具体体现为软件产品,例如软件开发包(Software Development Kit,SDK)等等。

[0199] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统 and 装置的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。在本公开所提供的几个实施例中,应该理解到,所揭露的系统、装置和方法,可以通过其它的方式实现。以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,又例如,多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些通信接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0200] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0201] 另外,在本公开各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。

[0202] 所述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,可以存储在一个处理器可执行的非易失的计算机可读取存储介质中。基于这样的理解,本公开的技术方案本质上或者说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,该计算机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机设备(可以是个人计算机,服务器,或者网络设备等)执行本公开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(Read-Only Memory,ROM)、随机存取存储器(Random Access Memory,RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0203] 最后应说明的是:以上所述实施例,仅为本公开的具体实施方式,用以说明本公开的技术方案,而非对其限制,本公开的保护范围并不局限于此,尽管参照前述实施例对本公开进行了详细的说明,本领域的普通技术人员应当理解:任何熟悉本技术领域的技术人员在本公开揭露的技术范围内,其依然可以对前述实施例所记载的技术方案进行修改或可轻易想到变化,或者对其中部分技术特征进行等同替换;而这些修改、变化或者替换,并不使相应技术方案的本质脱离本公开实施例技术方案的精神和范围,都应涵盖在本公开的保护范围之内。因此,本公开的保护范围应所述以权利要求的保护范围为准。

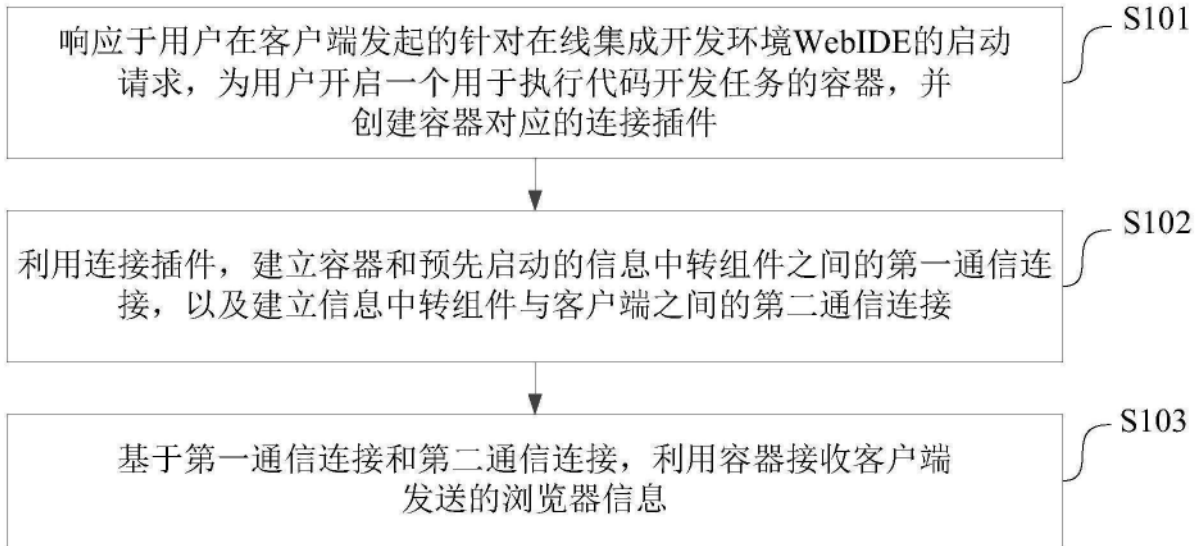


图1

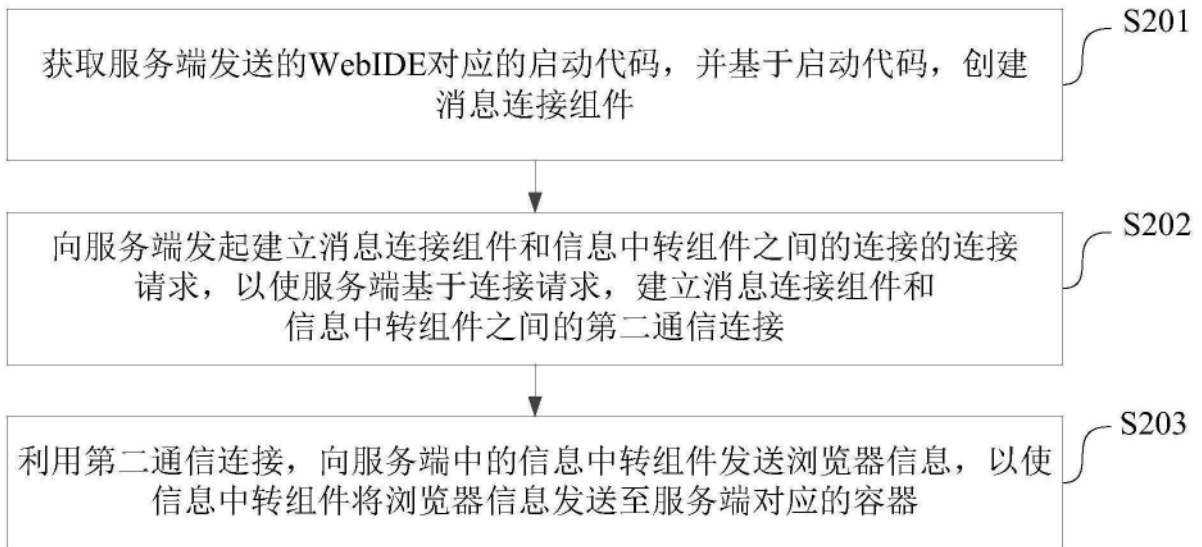


图2

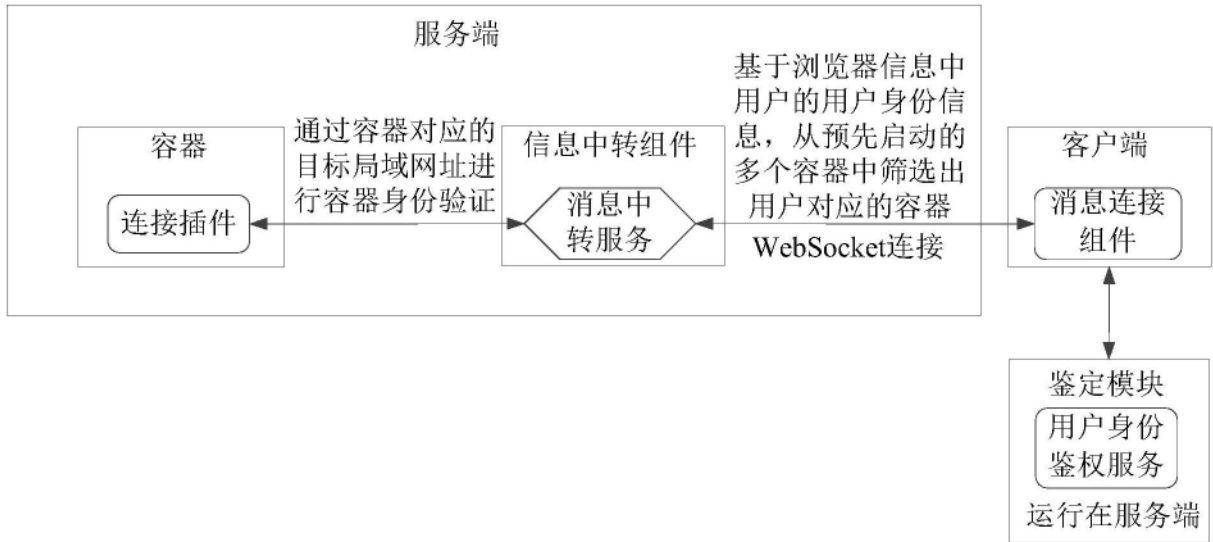


图3

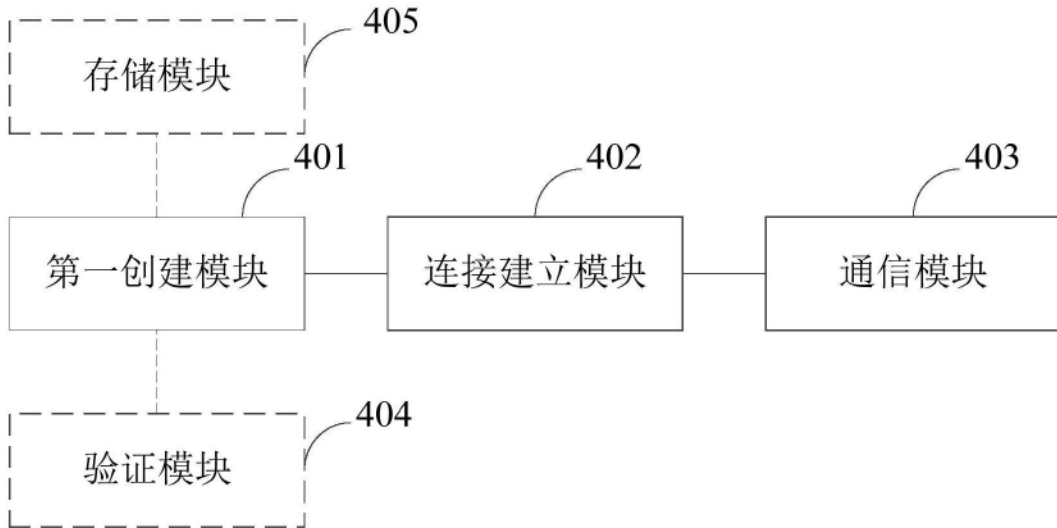


图4

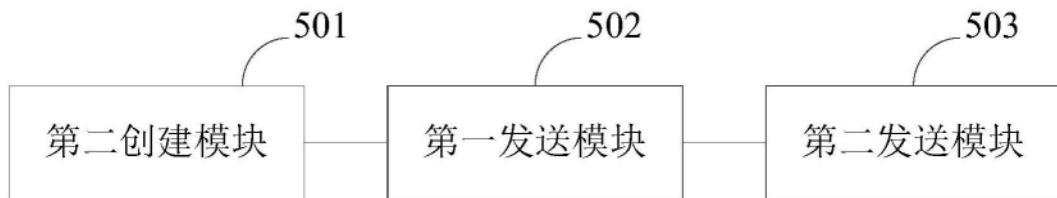


图5

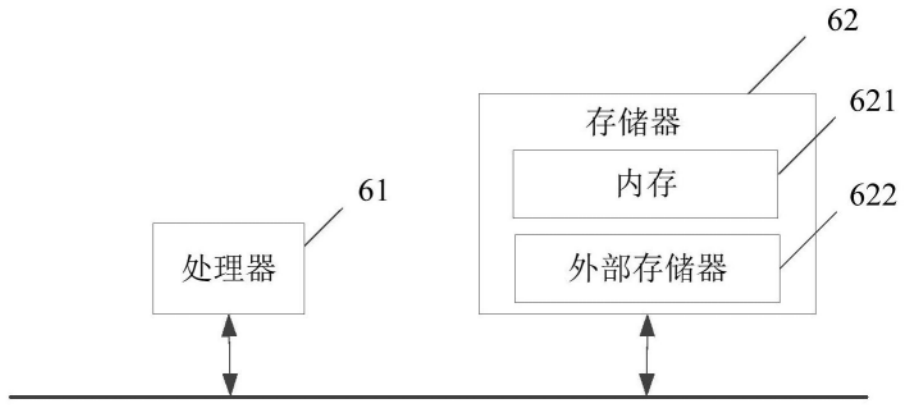


图6