



(12) 发明专利申请

(10) 申请公布号 CN 104126176 A

(43) 申请公布日 2014. 10. 29

(21) 申请号 201380009889. 3

(74) 专利代理机构 上海专利商标事务所有限公司 31100

(22) 申请日 2013. 02. 11

代理人 罗婷婷

(30) 优先权数据

13/398, 851 2012. 02. 17 US

(51) Int. Cl.

G06F 11/36(2006. 01)

(85) PCT国际申请进入国家阶段日

2014. 08. 18

(86) PCT国际申请的申请数据

PCT/US2013/025483 2013. 02. 11

(87) PCT国际申请的公布数据

W02013/122844 EN 2013. 08. 22

(71) 申请人 微软公司

地址 美国华盛顿州

(72) 发明人 S·H·吴 L·比罗夫 A·克赖德

J·杨

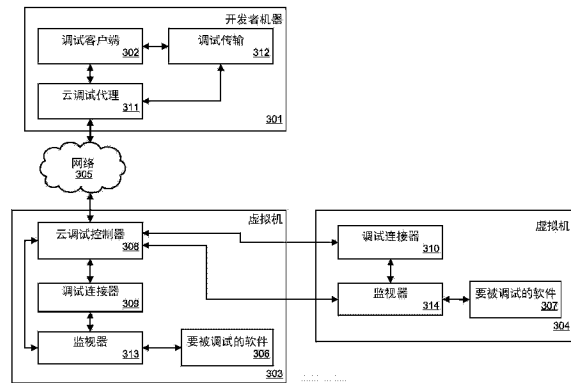
权利要求书1页 说明书6页 附图4页

(54) 发明名称

作为服务进行远程调试

(57) 摘要

本发明公开了针对在云计算环境中运行的软件的调试能力。控制器标识云中的哪些机器正在运行要被调试的软件实例。代理被部署在云中的机器上以促成与开发者的机器的通信。当开发者想要调试该代理的机器上的软件时,代理下载监视器并将监视器安装在该机器上。代理配置该机器以经由该监视器来进行远程调试。安全机制确保仅经认证的开发者可访问监视器和供调试的远程机器。控制器自动确定哪些机器可被调试、更新这些机器上可用于调试的过程列表、以及标识如何将开发者的调试客户端连接到这些机器。该控制器准许仅在来自经认证开发者的请求时仅对准许该开发者调试的那些过程进行远程调试。



1. 一种方法,包括:  
标识在分布式网络中操作的正运行软件应用的实例的一个或多个机器;  
接收来自调试客户端的对所述软件应用的所选实例进行调试的请求;  
指令代理配置与所述软件应用的所选实例相关联的机器以供远程调试;  
在所述调试客户端和在与所述软件应用的所选实例相关联的所述机器上的监视应用之间建立连接;以及  
使用所述调试客户端来调试所述软件应用的所选实例。
2. 如权利要求 1 所述的方法,其特征在于,调试控制器应用标识正在运行软件应用的实例的一个或多个机器。
3. 如权利要求 2 所述的方法,其特征在于,所述调试控制器应用正在所述分布式网络中操作的机器之一上运行。
4. 如权利要求 1 所述的方法,其特征在于,所述代理通过向机器添加对开发者的许可来配置所述机器以供调试。
5. 如权利要求 1 所述的方法,其特征在于,所述代理通过下载所述监视应用并将所述监视应用安装在机器上来配置所述机器以供调试。
6. 如权利要求 1 所述的方法,其特征在于,所述调试客户端和所述监视应用之间的连接由连接器应用建立。
7. 如权利要求 6 所述的方法,其特征在于,所述连接器提供并管理公共网络上的开发者机器和在私有网络上运行所述软件应用的机器之间的接口。
8. 如权利要求 6 所述的方法,其特征在于,所述连接器在连接所述调试客户端和机器之前认证所述调试客户端。
9. 一种计算机系统,包括:  
一个或多个处理器;  
系统存储器;  
其上存储有计算机可执行指令的一种或多种计算机可读存储介质,所述指令在由一个或多个处理器执行时,致使所述处理器执行一种用于调试软件应用的远程实例的方法,所述处理器用于:  
标识在分布式网络中操作的正运行所述软件应用的所述实例的一个或多个机器;  
接收来自调试客户端的对所述软件应用的所选实例进行调试的请求;  
指令代理配置与所述软件应用的所选实例相关联的机器以供远程调试;以及  
在所述调试客户端和在与所述软件应用的所选实例相关联的所述机器上的监视应用之间建立连接。
10. 如权利要求 9 所述的计算机系统,其特征在于,所述处理器还用于:使用连接器应用在所述调试客户端和所述监视应用之间建立所述连接。

## 作为服务进行远程调试

### [0001] 背景

[0002] 调试是发现和锁定软件中的错误或“隐错”的过程。一般来说,开发者调试计算机程序的过程或实例。被称为“调试器”的软件工具通常用于帮助软件开发者调试其软件。远程调试是调试正在与开发者正使用来写入、分析或调试软件的机器不同的机器上运行的软件的过程。“远程调试器”工具被开发者用来进行远程调试。远程调试器通常具有两个部分。监视器软件在该远程机器上运行,并使得开发者的机器能够附连到该远程机器并调试该远程机器上的软件。客户端软件在开发者的机器上运行并连接到监视器,由此向开发者提供接口来调试远程机器上的软件。

[0003] 最近,在云计算环境中执行软件开发已变得普遍。在本上下文中,开发者可能想要调试的软件正在并非由该开发者拥有或控制的服务器上运行。相反,运行该软件的服务器通常位于第三方所拥有和 / 或管理的远程数据中心中。此外,开发者和第三方通常不知道云环境中的哪些机器的确在运行要被调试的软件。云计算环境中的管理实体评估对该软件的需求及其他因素(诸如负载共享要求和机器可用性),并动态地选择哪些机器(及多少机器)应运行该软件。结果,开发者难以确定云中的哪些服务器正在运行该软件。在许多情况下,云环境中的服务器的所有者或管理员限制开发者对服务器和软件的访问,以使得仅某些过程可被调试。

[0004] 云计算环境中的服务器和其他机器可经由分布式公共和 / 或私有计算机网络(诸如,因特网或内联网)来访问。为了连接到云计算环境中的各机器,开发者通常需要将公共的因特网用于该连接的至少部分。即使开发者能够标识哪些机器正在运行该软件,云环境中的一些服务器可能难以从公共因特网进行访问。结果,可能难以在开发者的机器和云计算环境中运行要被调试的软件的机器之间实现直接的网络 TCP/IP 连接。

[0005] 软件中的错误、隐错和其他缺陷可能直到该软件已运行了较长时间段后才被注意到。因此,运行该软件的机器不需要在所有时间都是可调试的,但仅需被配置为在开发者想要或需要调试该软件时进行调试。

### [0006] 概述

[0007] 提供本概要从而以简要形式引入将在下面具体实施例中进一步描述的概念的选择。本概要旨在标识所要求保护的的主题的关键特征或必要特征,也不旨在用来限制所要求保护的的主题的范围。

[0008] 各实施例提供针对在云计算环境中运行的软件的调试能力。控制器标识云中的哪些机器正在运行要被调试的软件实例。代理被部署在云中的各机器上。代理促成与开发者的机器的通信。当开发者想要调试代理的机器上的软件时,该代理下载监视器并将该监视器安装在该机器上。该代理配置该机器以供经由监视器来进行远程调试。安全机制确保仅经认证的开发者可访问监视器以及远程机器来进行调试。

[0009] 在一个实施例中,云中的软件应用自动确定哪些机器可被调试、更新这些机器上可被调试的过程列表(诸如,过程标识符列表)、并标识如何将开发者的调试客户端连接到这些机器。

[0010] 在另一实施例中,在云中运行的软件应用准许仅在来自经认证的开发者的请求时才进行远程调试。应用在开发者完成了调试后移除对该远程过程的访问。该应用仅列出了准许开发者调试的那些过程。所有其他过程都被过滤掉,以使得开发者不会看见他没有许可的过程。

[0011] 在其他实施例中,软件应用使得远程调试器能够被下载,并被配置在运行云的远程机器上。

[0012] 附图

[0013] 为了进一步阐明本发明的各实施例的以上和其他优点和特征,将参考附图来呈现本发明的各实施例的更具体的描述。可以理解,这些附图只描绘本发明的典型实施例,因此将不被认为是对其范围的限制。本发明将通过使用附图用附加特征和细节来描述和解释,附图中:

[0014] 图 1 是根据一个实施例的用于在云环境中进行远程调试的系统的框图;

[0015] 图 2 是示出用于对云环境中的远程机器上的软件进行调试的过程或方法的流程图;

[0016] 图 3 是根据一个实施例的用于在云环境中进行远程调试的系统的框图;以及

[0017] 图 4 示出合适的计算和联网环境 400(诸如,在一个实施例中为开发者机器和/或基于云的远程服务器)的示例。

[0018] 详细描述

[0019] 图 1 是根据一个实施例的用于在云环境中对软件进行远程调试的系统的框图。调试器客户端 101 被开发者用来调试软件应用。调试器客户端 101 可以是独立的应用,或者是集成开发环境(IDE)中允许开发者设计并调试软件应用的组件。控制器 102 是标识并监视云环境的配置细节的软件应用。控制器 102 具有询问云环境以得到关于哪些机器正在运行要被开发者调试的软件的实例 103 的细节的能力。任何数目的软件实例 103 可同时运行。多个软件实例 103 可在相同的机器上运行和/或可散布在多个不同的机器上。

[0020] 控制器 102 向调试器客户端 101 提供机器身份、过程身份和任何其他必要信息,以使得开发者可对在各远程机器上运行的软件实例 103 进行调试。控制器 102 可在持续的基础上为有兴趣调试的软件实例 103 更新云环境细节(诸如机器身份和过程身份),使得调试器客户端 101 和开发者总是具有最正确的信息用于调试。在一个实施例中,控制器 12 仅返回针对在机器上运行的调试器客户端 101 具有访问许可的过程的信息。如果机器的所有者尚未将用户许可给予开发者或调试器客户端,则控制器 102 不标识那些软件实例以防止尝试调试开发者不具有许可的过程。

[0021] 为每一软件实例 103 或为运行一个或多个软件实例 103 的每一机器部署代理 104。代理 104 经由控制器 102 与调试器客户端 101 通信。代理 104 侦听指示开发者想要调试机器上的软件 103 的命令。在一个实施例中,控制器 102 可命令代理 104 配置或设置供调试的机器。代理 104 可以是具有下载并安装监视器 105 的能力的软件应用。代理 104 随后诸如通过打开防火墙端口、设置合适的许可等来配置该机器,以使得监视器 105 能够在该机器上运行。

[0022] 连接器 106 是在云环境中运行的软件应用。连接器 106 充当从公共因特网上的调试器客户端 101 到不可公共地访问的机器上运行的监视器 105 的连接桥(bridge)。连

接器 106 可以是例如认证这些连接并将来自调试器客户端 101 的请求路由到合适的监视器 105 的路由器。

[0023] 代理 104 软件可连同要被调试的软件代码一起部署,或者代理 104 可以在较后的时间部署。

[0024] 图 2 是示出用于调试云环境中的远程机器上的软件的过程或方法的流程图。在步骤 201,开发者打开调试器客户端并选择要被调试的软件代码。在步骤 202,调试器客户端与控制器通信以标识出要被调试的软件的正在云环境中运行的实例。在步骤 203,控制器通过例如向远程机器上的代理应用请求软件状态信息来标识与要被调试的软件实例相关联的机器和过程。代理应用将软件状态信息提供给控制器,该控制器随后发送机器和过程列表以供在步骤 204 向开发者显示。在步骤 205 使用调试器客户端,开发者选择要被调试的一个或多个软件实例。软件实例可被标识成例如在这些机器之一上运行的特定过程。

[0025] 在步骤 206,调试器客户端向控制器标识要被调试的软件实例。在步骤 207,控制器随后指令与该软件实例相关联的机器上的代理配置该机器以进行远程调试。在步骤 208,在监视器尚不可用的情况下,代理下载监视器软件,并将该监视器软件安装在该机器上。在步骤 209,代理通过例如添加合适的许可和用户账户、打开防火墙端口和启动监视器软件来配置该机器。

[0026] 在步骤 210,调试器客户端与连接器通话,连接器将连接路由到合适的监视器。在一些实施例中,调试器客户端可能无法直接抵达在该机器上运行的软件。然而,连接器在云环境中的公共因特网和私有网络连接之间提供接口并管理这些接口。在步骤 211,连接器认证调试器客户端并随后将客户端和机器连接起来。在步骤 212,开发者可随后开始调试所选的远程软件实例。

[0027] 可以理解,在图 2 中所示的过程的步骤 201-212 可以同时和 / 或按序执行。还可以理解,每个步骤可以按任何顺序执行,且在其他实施例中可以被执行一次或被重复执行。

[0028] 图 3 是根据另一实施例的用于在云环境中进行远程调试的系统的框图。软件开发者使用正运行调试器客户端 302(诸如,IDE 应用)的开发者机器 301。开发者可能想要调试远程机器或服务器上的软件。如图 3 所示,这些远程机器可能是仅可经由公共或私有网络 305(诸如,因特网或内联网)来访问的云环境中的虚拟机 303、304。

[0029] 要被调试的软件的多个实例 306、307 可被部署在虚拟机 303、304 上。云调试控制器 308 用于标识软件的实例 306、307。在一个实施例中,云调试控制器 308 的单个实例被部署到虚拟机 303、304 之一。此外,调试连接器 309、310 被部署到每一虚拟机 303、304。与云调试控制器 308 一起工作,调试连接器 309、310 充当用于配置要被调试的机器 303、304 的代理。云调试连接器 308 和调试连接器 309、310 可以与软件实例 306、307 一起部署,或者在调试是在开发者机器 301 上发起的时在较后的时间再加载。

[0030] 云调试控制器 308 使用来自云环境运行时和框架的信息来发现云环境的拓扑。云调试器控制器 308 发现调试控制器 309、310,并建立到它们的连接。云调试控制器 308 充当网关,并为开发者机器 301 提供在云环境中的可见连接。云调试代理 311 在开发者机器 301 的调试传输 312 和云调试控制器 308 之间提供桥。在一个实施例中,调试传输 312 为调试客户端 302 创建云拓扑的合成视图。在云计算环境中,由于工作负载被动态平衡,因此各软件实例的机器名、IP 地址、端口数和过程名可改变。调试传输 312 将这些软件实例作为供

调试的单个应用呈现给调试客户端 302。通过这种方式,调试客户端 302 和开发者不必持续跟踪每一软件实例的信息。

[0031] 云调试控制器 308 和调试连接器 309、310 标识要被调试的软件的实例。可对软件实例列表进行过滤,以仅包括可被开发者机器 301 调试的那些过程。例如,如果开发者不具有访问虚拟机或过程的许可,或者如果机器的所有者以其他方式限制了调试能力,则不向调试客户端 302 列出那些机器上的软件实例。

[0032] 监视器软件 313、314 被加载到用于调试软件 306、307 的虚拟机 303、304 上。监视器软件 313、314 可以与软件实例或调试连接器 309 一起部署,或者可以在稍后由云调试控制器 308 来部署。

[0033] 为了调试软件,调试客户端 302 向云调试控制器 308 发送发现网络拓扑的请求。云调试控制器 308 与连接器 309、310 通信并请求关于要被调试的软件的实例的信息。云调试控制器 308 将该信息传递至调试客户端 302,调试客户端 302 显示已发现了多少机器、过程和 / 或软件实例。开发者可随后选择要调试的机器、过程和 / 或实例中的一个或全部。在一个实施例中,提供允许用户选择对软件应用的在不同机器上运行的所有实例同时进行远程调试的界面或可视化。用户可以用一个动作(诸如,一次点击或选择)来选择此跨所有机器的同时调试。随后该系统自动为所选的软件实例在调试客户端 302 和监视器 313、314 之间建立连接。开发者可随后经由监视器 313、314 来访问所选的软件实例。

[0034] 图 4 示出了计算和联网环境 400(诸如开发者机器和 / 或基于云的远程服务器)的示例,在一个实施例中,该计算和联网环境 400 使用本文中所描述的控制器、连接器、代理和监视器来支持调试器客户端和 / 或被调试的远程软件实例并促成开发者机器和远程服务器之间的连接。计算系统环境 400 只是合适计算环境的一个示例,而非意在暗示对本发明使用范围或功能有任何限制。本发明可用各种其他通用或专用计算系统环境或配置来操作。适用于本发明的公知计算系统、环境、和 / 或配置的示例包括但不限于:个人计算机、服务器计算机、手持式或膝上型设备、平板设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费电子产品、网络 PC、微型计算机、大型计算机、包括任何以上系统或设备的分布式计算环境等等。

[0035] 调试客户端、控制器、连接器、代理、监视器和被调试的软件实例可以在诸如程序模块等由计算机执行的计算机可执行指令的通用上下文中描述。一般而言,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。本发明也可在其中由通过通信网络链接的远程处理设备执行任务的分布式计算环境中实现。在分布式计算环境中,程序模块可以位于包括存储器存储设备在内的本地和 / 或远程计算机存储介质中。

[0036] 参考图 4,用于实现本发明的各个方面的示例性系统可以包括计算机 400 形式的通用计算设备。组件可包括但不限于各种硬件组件(诸如处理单元 401)、数据存储 402(诸如系统存储器)、以及将包括数据存储 402 在内的各种系统组件耦合到处理单元 401 的系统总线 403。系统总线 403 可以是若干类型的总线结构中的任一种,包括使用各种总线体系结构中的任一种的存储器总线或存储器控制器、外围总线、以及局域总线。作为示例而非限制,这样的体系结构包括工业标准体系结构 (ISA) 总线、微通道体系结构 (MCA) 总线、增强型 ISA (EISA) 总线、视频电子技术标准协会 (VESA) 局部总线和外围部件互连 (PCI) 总线

(也称为夹层 (Mezzanine) 总线)。

[0037] 计算机 400 通常包括各种计算机可读介质 404。计算机可读介质 404 可以是能由计算机 401 访问的任何可用介质,并同时包含易失性和非易失性介质以及可移动、不可移动介质,但不包括传播信号。作为示例而非限制,计算机可读介质 404 可包括计算机存储介质和通信介质。计算机存储介质包括以存储诸如计算机可读的指令、数据结构、程序模块或其他数据之类的信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括,但不仅限于, RAM、ROM、EEPROM、闪存或其他存储器技术、CD-ROM、数字多功能盘 (DVD) 或其他光盘存储、磁带盒、磁带、磁盘存储或其他磁存储设备,或可以用来存储所需信息并可以被计算机 400 访问的任何其他介质。通信介质通常以诸如载波或其他传输机制之类的已调制数据信号来体现计算机可读指令、数据结构、程序模块或其他数据,并且包括任何信息传送介质。术语已调制数据信号是指使得以在信号中编码信息的方式来设置或改变其一个或多个特征的信号。作为示例而非限制,通信介质包括诸如有线网络或直接线连接之类的有线介质,以及诸如声学、RF、红外及其他无线介质之类的无线介质。上面各项中的任何项的组合也包括在计算机可读介质的范围内。计算机可读介质可被实现为计算机程序产品,诸如存储在计算机存储介质上的软件。

[0038] 数据存储或系统存储器 402 包括诸如只读存储器 (ROM) 和 / 或随机存取存储器 (RAM) 之类的易失性和 / 或非易失性存储器形式的计算机存储介质。基本输入 / 输出系统 (BIOS) 包含有助于诸如启动时在计算机 400 中元件之间传递信息的基本例程,它通常被存储在 ROM 中。RAM 通常包含处理单元 401 可立即访问和 / 或当前正在操作的数据和 / 或程序模块。作为示例而非限制性,数据存储 402 保存操作系统、应用程序、其他程序模块、和程序数据。

[0039] 数据存储 402 还可以包括其它可移动 / 不可移动、易失性 / 非易失性计算机存储介质。仅作为示例,数据存储 402 可以是对不可移动、非易失性磁介质进行读写的硬盘驱动器,对可移动、非易失性磁盘进行读写的磁盘驱动器,以及对诸如 CD ROM 或其它光学介质等可移动、非易失性光盘进行读写的光盘驱动器。可在示例性操作环境中使用的其它可移动 / 不可移动、易失性 / 非易失性计算机存储介质包括但不限于,磁带盒、闪存卡、数字多功能盘、数字录像带、固态 RAM、固态 ROM 等。上文所描述的并且在图 4 中所显示的驱动器以及它们的关联的计算机存储介质,为计算机 400 提供对计算机可读的指令、数据结构、程序模块及其他数据的存储。

[0040] 用户可通过用户接口 405 或诸如平板、电子数字化仪、话筒、键盘和 / 或定点设备 (通常指的是鼠标、跟踪球或触摸垫) 等其它输入设备输入命令和信息。其他输入设备可以包括操纵杆、游戏垫、圆盘式卫星天线、扫描仪等等。另外,语音输入、使用手或手指的手势输入、或其它自然用户接口 (NUI) 也可与适当的输入设备 (诸如话筒、相机、平板、触摸垫、手套、或其它传感器) 一起使用。这些及其他输入设备常常通过耦合到系统总线 403 的用户输入接口 405 连接到处理单元 401,但是,也可以通过其他接口和总线结构,如并行端口、游戏端口或通用串行总线 (USB),来进行连接。监视器 406 或其他类型的显示设备也通过诸如视频接口之类的接口连接至系统总线 403。监视器 406 也可以与触摸屏面板等集成。注意到监视器和 / 或触摸屏面板可以在物理上耦合至其中包括计算设备 400 的外壳,诸如在平板型个人计算机中。此外,诸如计算设备 400 等计算机还可以包括其他外围输出设备,诸

如扬声器和打印机,它们可以通过输出外围接口等连接。

[0041] 计算机 400 可使用至一个或多个远程设备(诸如远程计算机)的逻辑连接 407 在网络化或云计算环境中操作。远程计算机可以是个人计算机、服务器、路由器、网络 PC、对等设备或其它常见的网络节点,并且一般包括上面相对于计算机 400 所述的许多或全部元件。图 4 中所描述的逻辑连接包括一个或多个局域网(LAN)和一个或多个广域网(WAN),但是,也可以包括其他网络。此类联网环境在办公室、企业范围的计算机网络、内联网和因特网中是常见的。

[0042] 当在联网或云计算环境中使用时,计算机 400 可通过网络接口或适配器 407 连接至公共或私有网络。在一些实施例中,使用调制解调器或用于在网络上建立通信的其它装置。调制解调器可以是内置或外置的,它经由网络接口 407 或其它适当的机制连接至系统总线 403。诸如包括接口和天线的无线联网组件可通过诸如接入点或对等计算机之类的合适的设备耦合到网络。在联网环境中,相关于计算机 400 所示的程序模块或其部分可被存储在远程存储器存储设备中。可以理解,所示的网络连接是示例性的,也可以使用在计算机之间建立通信链路的其他手段。

[0043] 尽管用结构特征和 / 或方法动作专用的语言描述了本主题,但可以理解,所附权利要求书中定义的主题不必限于上述具体特征或动作。更确切而言,上述具体特征和动作是作为实现权利要求的示例形式公开的。



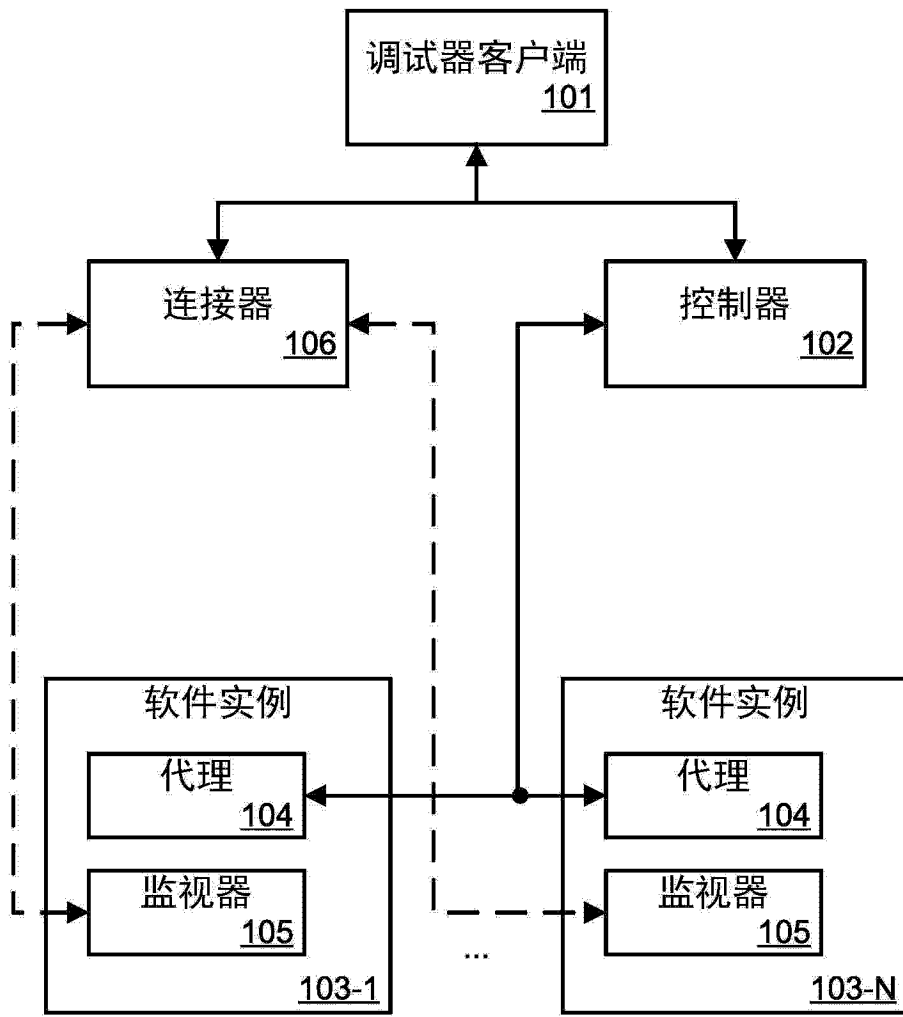


图 1

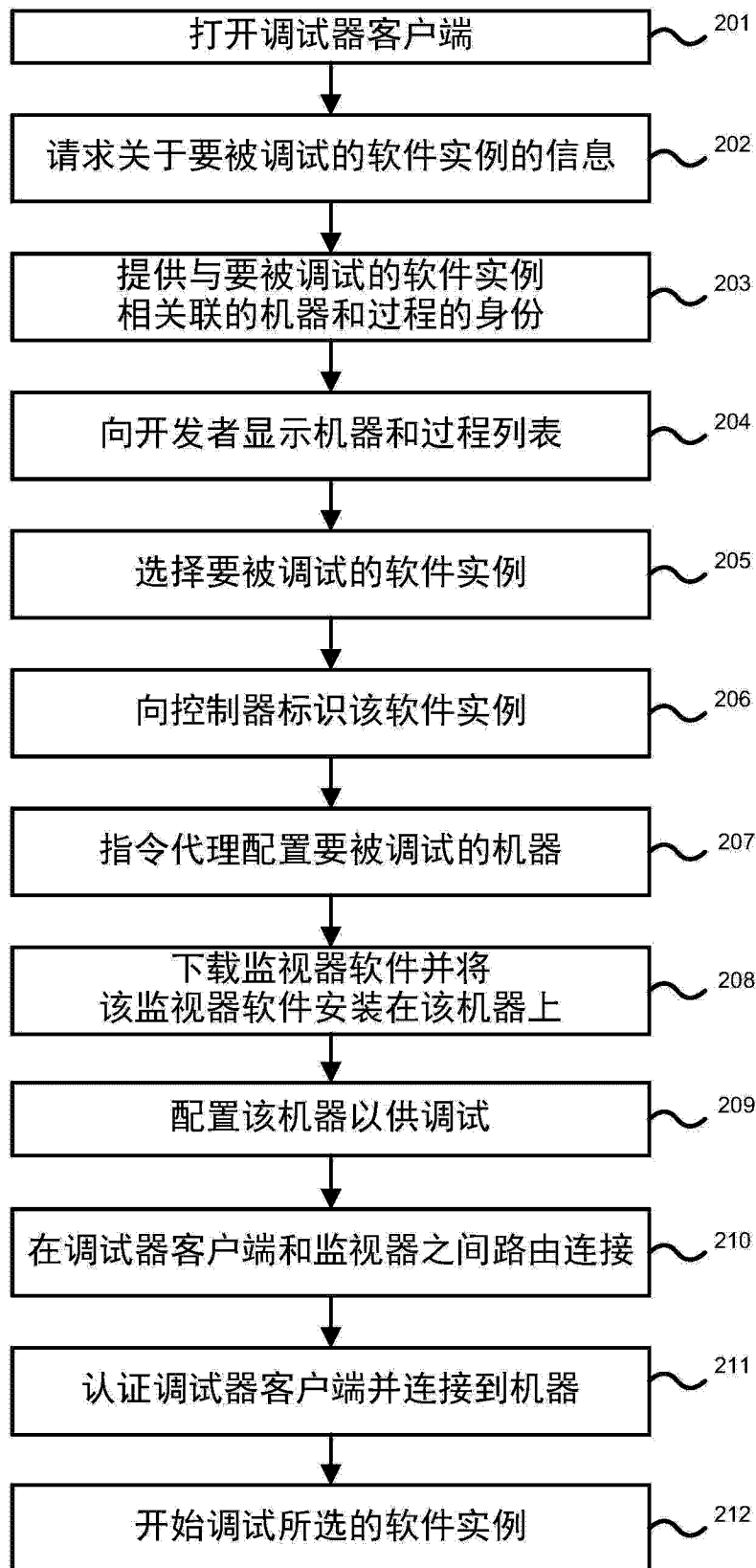


图 2

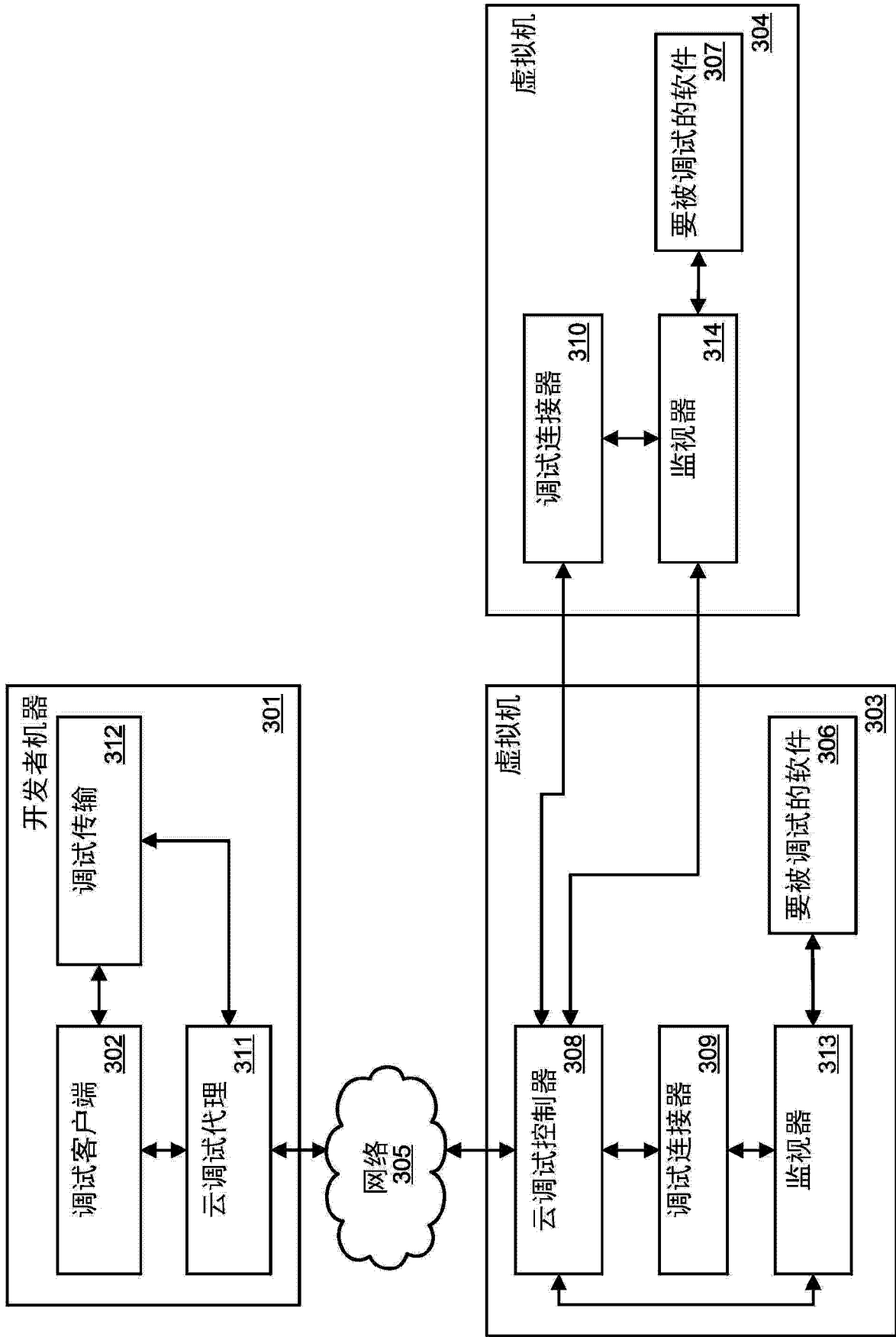


图 3

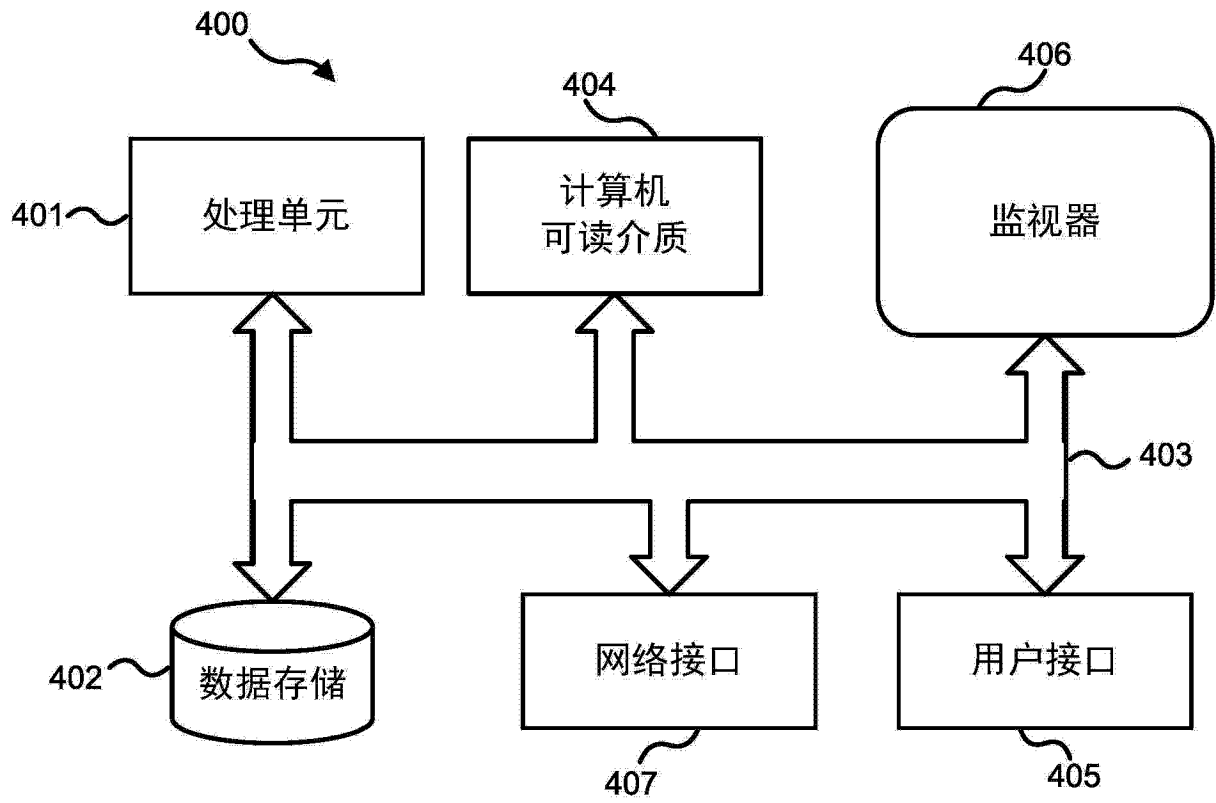


图 4