



(12) 发明专利

(10) 授权公告号 CN 101689165 B

(45) 授权公告日 2013. 07. 24

(21) 申请号 200880022209. 0

(22) 申请日 2008. 06. 16

(30) 优先权数据

11/769, 022 2007. 06. 27 US

(85) PCT申请进入国家阶段日

2009. 12. 28

(86) PCT申请的申请数据

PCT/US2008/067141 2008. 06. 16

(87) PCT申请的公布数据

W02009/002751 EN 2008. 12. 31

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 N·A·雅格布森 J·M·希恩

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 蔡悦 钱静芳

(51) Int. Cl.

G06F 15/16 (2006. 01)

(56) 对比文件

CN 1342290 A, 2002. 03. 27, 全文.

US 2005/0246718 A1, 2005. 11. 03, 说明书第 5-12 栏, 图 2.

审查员 吴鑫

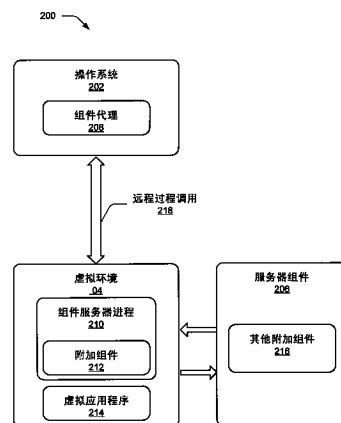
权利要求书2页 说明书7页 附图4页

(54) 发明名称

在虚拟环境中运行附加组件

(57) 摘要

描述了用于在虚拟环境中运行附加组件的系统和方法。可以通过在主操作系统中的组件代理和在虚拟环境中运行的服务器进程之间建立通信链路来在该虚拟环境中执行附加组件。该服务器进程基于转发来自操作系统的命令的组件代理所转发的命令来执行附加组件。



1. 一种在虚拟环境中运行附加组件的系统,包括:

用于调用(302)对应于附加组件的、包括在操作系统中的组件代理的装置,其中所述附加组件提供功能并与所述虚拟环境中的虚拟应用程序相关联;

用于将所述组件代理与所述虚拟环境相关联(304)的装置;

用于基于所述附加组件在所述虚拟环境中是否可用来确定(306)所述虚拟环境是否是正确的虚拟环境的装置;

用于如果确定所述虚拟环境是正确的,则启动(308)虚拟环境中的组件服务器进程的装置,其中所述组件服务器进程使用 API 接收由所述组件代理转发的命令并执行所述附加组件;以及

用于如果所述确定发现所述虚拟环境是不正确的,则查找正确的虚拟环境的装置,以及如果所述组件代理是使用足够的信息注册的,则连接到在适当的虚拟环境中运行的服务器。

2. 如权利要求 1 所述的系统,其特征在于,针对所述附加组件中的一个或多个向所述操作系统注册所述组件代理。

3. 如权利要求 2 所述的系统,其特征在于,所述注册是在一个或多个计算设备上模拟所述虚拟环境时执行的。

4. 如权利要求 1 所述的系统,其特征在于,所述操作系统调用所述组件代理来访问所述附加组件。

5. 如权利要求 1 所述的系统,其特征在于,所述组件代理访问虚拟环境的清单以确定在不启动服务器的情况下所述组件代理所支持的组件。

6. 如权利要求 1 所述的系统,其特征在于,所述附加组件包括外壳扩展。

7. 一种在虚拟环境中运行附加组件的方法(300),包括:

调用(302)对应于附加组件的、包括在操作系统中的组件代理,其中所述附加组件提供功能并与所述虚拟环境中的虚拟应用程序相关联;

将所述组件代理与所述虚拟环境相关联(304);

基于所述附加组件在所述虚拟环境中是否可用来确定(306)所述虚拟环境是否是正确的虚拟环境;

如果确定所述虚拟环境是正确的,则启动(308)虚拟环境中的组件服务器进程,其中所述组件服务器进程使用 API 接收由所述组件代理转发的命令并执行所述附加组件;以及

如果所述确定发现所述虚拟环境是不正确的,则查找正确的虚拟环境,以及如果所述组件代理是使用足够的信息注册的,则连接到在适当的虚拟环境中运行的服务器。

8. 如权利要求 7 所述的方法,其特征在于,在进行所述关联时使用应用程序接口。

9. 如权利要求 7 所述的方法,其特征在于,所述确定包括检查所述虚拟环境是否包括所述对应的附加组件。

10. 如权利要求 7 所述的方法,其特征在于,所述启动是在确认所述附加组件主存在所述虚拟环境中之后执行的。

11. 如权利要求 7 所述的方法,其特征在于,所述组件代理访问虚拟环境的清单以确定在不启动服务器的情况下所述组件代理所支持的组件。

12. 如权利要求 7 所述的方法,其特征在于,还包括在所述虚拟环境中创建对应于组件

代理的所述附加组件的实例。

13. 如权利要求 9 所述的方法,其特征在于,还包括从所述组件代理将命令转发给所述组件服务器进程,进而转发给所述附加组件。

在虚拟环境中运行附加组件

[0001] 背景

[0002] 在应用程序虚拟化中,主操作系统在计算设备上创建模拟计算机环境和虚拟环境,以执行虚拟应用程序而无需在本地安装虚拟应用程序且无需改变主操作系统。虽然虚拟应用程序使用本地硬件资源本地地执行,并且可以与主操作系统交互,但是可以向主操作系统隐藏虚拟环境的各方面(例如注册表项、文件等)。

[0003] 这是可能的,因为虚拟环境包含在计算设备上执行虚拟应用程序所需的各组件,诸如注册表项、文件、环境变量、用户界面元素、以及全局对象等。虚拟环境中的这些组件对于主操作系统而言是不可见的。诸如外壳扩展、插件等与虚拟应用程序相关联的附加组件对于主操作系统而言也是不可见的,因为主操作系统不能访问附加组件的注册表项或文件。结果,主操作系统一直不知道虚拟环境中可用的附加组件,并且不能使用虚拟环境中可用的附加组件所提供的功能。

[0004] 概述

[0005] 提供本概述以便以介绍与在虚拟环境中运行应用程序的附加组件相关的概念。这些概念在以下详细描述中进一步描述。本概述并不旨在标识出所要求保护的主题的必要特征,也不旨在用于确定所要求保护的题目的范围。

[0006] 在一个实施例中,包括在操作系统中的组件代理与包括一个或多个虚拟应用程序的虚拟环境通信。附加组件与该一个或多个虚拟应用程序相关联,并且组件服务器进程接收由该组件代理转发的命令并且执行附加组件。

[0007] 附图简述

[0008] 参考附图描述了详细描述。在附图中,参考标号的最左位标识了该参考标号首次出现在其中的附图。在所有附图中使用相同的标号来指示相同的特征和组件。

[0009] 图 1 示出了在一个实施例中用于在虚拟环境中运行附加组件的示例性系统。

[0010] 图 2 示出了在另一实施例中用于在虚拟环境中运行附加组件的示例性系统。

[0011] 图 3 示出了用于在虚拟环境中运行附加组件的示例性过程流程图。

[0012] 图 4 示出了示例性的基于计算的设备。

[0013] 详细描述

[0014] 以下发明描述了用于在虚拟环境中运行附加组件的系统和方法。附加组件可以通过在主操作系统(即操作系统)和在虚拟环境中运行的服务器进程之间建立通信链路来在虚拟环境中执行。操作系统包括与在服务器进程中执行的附加组件进行通信的组件代理。所描述的系统和方法还允许通过网络按需访问和使用虚拟化的附加组件。

[0015] 虽然所描述的用于在虚拟环境中运行附加组件的系统和方法的各方面可以在任何数量的不同计算系统、环境、和 / 或配置中实现,但是在以下示例性系统体系结构的上下文中描述各实施例。

[0016] 图 1 示出了用于在虚拟环境中运行附加组件的示例性系统 100。系统 100 可以是诸如个人计算机等独立计算设备的一部分或被包括在其中,或者可以作为若干计算设备的一部分来包括。附加组件可以是外壳(shell)扩展、插件、或可以向现有应用程序提供附加

功能的任何其他类似的实体。为此,系统 100 包括计算设备上的操作系统 102 和虚拟环境 104。在一个实现中,系统 100 可以在单个计算设备上实现,以便于在该计算设备上运行的一个或多个虚拟环境之间的资源共享。计算设备可以被实现为诸如台式计算设备、膝上型计算机、移动计算设备等任何已知的计算设备。

[0017] 此外,操作系统 102 包括与虚拟环境 104 通信的组件代理 106。虚拟环境 104 包括组件服务器进程 108 和诸如虚拟应用程序 110 等一个或多个虚拟应用程序。组件服务器进程 108 包括与虚拟应用程序相关联的一个或多个附加组件,诸如与虚拟应用程序 110 相关联的附加组件 112。

[0018] 为了访问由驻留在虚拟环境 104 中的附加组件 112 所提供的功能,组件代理 106 被配置成与组件服务器进程 108 通信以便执行附加组件 112。为此,组件代理 106 可以使用诸如远程过程调用 114 等应用程序接口 (API) 来与组件服务器进程 108 通信。

[0019] 在示例性操作中,对于虚拟环境 104 中可用的每个附加组件 112,向操作系统 102 注册组件代理 106。注册可以例如在计算设备上模拟虚拟环境 104 时执行。为了访问附加组件 112,操作系统 102 调用对应的组件代理 106。如果操作系统 102 首次调用组件代理 106,则组件代理 106 在虚拟环境 104 中启动组件服务器进程 108,并且使用远程过程调用 114 与组件服务器进程 108 通信。远程过程调用 114 允许在虚拟环境 (例如虚拟环境 104) 中执行来自组件代理 106 的过程或命令,而无需为该交互提供显式的代码细节。

[0020] 一个示例实施包括其中有多于一个虚拟环境在系统 100 中运行的情况,组件代理 106 检查在其中启动组件服务器进程 108 的虚拟环境 (例如虚拟环境 104) 是否是正确的虚拟环境。为此,组件代理 106 确定对应的附加组件 112 在虚拟环境 104 中是否可用。一旦标识了正确的虚拟环境 (例如虚拟环境 104),则组件代理 106 在所标识的虚拟环境 (例如虚拟环境 104) 中启动组件服务器进程 108。在另一示例性实施中,组件代理 106 使用足够的信息来注册以便直接连接到在适当的虚拟环境中运行的服务器。如果服务器没有在正确的虚拟环境中运行,则组件代理可以检测到这一点并且启动服务器。在又一示例性实施中,组件代理 106 能够访问每个虚拟环境的清单,以确定在不启动虚拟环境内的服务器并且不与该服务器通信的情况下支持什么组件。

[0021] 组件服务器进程 108 接收由组件代理 106 转发的命令并且相应地执行附加组件 112。该命令可以包括用于在虚拟环境 104 中实现共同界面的指令,例如更新上下文菜单、共享菜单等。

[0022] 如果操作系统 102 先前调用了组件代理 106,并启动了组件服务器进程 108,则组件服务器进程 108 直接将所有命令转发给附加组件 112 以供进一步的处理。例如,操作系统 102 可以是来自微软®公司的Windows®操作系统,而附加组件 112 可以是在虚拟环境 104 中可用的Windows®操作系统外壳扩展。Windows®操作系统通过外壳扩展代理 (例如组件代理 106) 与虚拟环境 014 交互,以便在服务器进程 (例如组件服务器进程 108) 中创建外壳扩展 (例如附加组件 112) 的对应实例。一旦创建了外壳扩展的实例,则Windows®操作系统通过经由外壳扩展代理 (例如组件代理 106) 将所有命令转发给服务器进程 (例如组件服务器进程 108)、服务器进程进而将命令转发给外壳扩展来访问外壳扩展。

[0023] 可以理解,当有若干虚拟环境在计算设备上运行时,可以使用类似的方法来运行附加组件。此外,每个虚拟环境可以具有在其上运行的若干虚拟应用程序以及附加组件。此

外,可以有同一应用程序和附加组件的多个版本在计算设备上的不同虚拟环境中运行。

[0024] 图 2 示出了在另一实施例中用于在虚拟环境中运行附加组件的示例性系统 200。系统 200 可以被实现为诸如个人计算机等独立计算设备或在包括多个计算设备的服务器-客户机体系结构中实现。附加组件可以是外壳扩展、插件、或可以向现有应用程序提供附加功能的任何相似的实体。为此,系统 200 包括客户机设备上的操作系统 202、虚拟环境 204、和服务器组件 206。在一示例中,虚拟环境 204 可以由主存在服务器组件 206 上的应用程序来模拟。

[0025] 操作系统 202 包括与虚拟环境 204 通信的组件代理 208。虚拟环境 204 包括组件服务器进程 210、附加组件 212、和诸如虚拟应用程序 214 等一个或多个虚拟化的应用程序。

[0026] 此外,其他附加组件 216 可以被主存在服务器组件 206 上或主存在虚拟环境 204 中。同样,诸如在应用程序安装和执行期间其他附加组件 216 如何与操作系统(例如操作系统 202)进行交互等信息可以被主存在服务器组件 206 上或主存在虚拟环境 204 中。服务器组件 206 也可以保存注册表设置的日志;文件;以及其他应用程序和操作系统相关变量。

[0027] 操作系统 202 通过经由远程过程调用 218 通信来访问虚拟环境 204 中的附加组件 212。为此,操作系统 202 可以使用各种引擎和技术来在组件代理 208 和虚拟环境 204 之间建立通信链路。

[0028] 如参考图 1 所解释的,对于可以在虚拟环境 204 中执行的每个附加组件 212,在操作系统 202 中注册相对应的组件代理 208。为了使用由附加组件 212 所提供的功能,操作系统 202 调用相对应的组件代理 208。如果操作系统 202 首次调用组件代理 208,则组件代理 208 在虚拟环境 204 中启动组件服务器进程 210 并且使用远程过程调用 218 来与该组件服务器进程通信。否则,组件代理 208 直接将所有命令转发给组件服务器进程 210 以供进一步处理。

[0029] 在存在多个虚拟环境的情况下,组件代理 208 检查组件服务器进程 210 是否在正确的虚拟环境,例如虚拟环境 204 中启动。为此,组件代理 208 确定虚拟环境是否包括相对应的附加组件 212。组件服务器进程 210 在虚拟环境 204 中执行附加组件 212,并且将来自组件代理 208 的所有命令转发给附加组件 212。

[0030] 此外,如果附加组件 212 不与由操作系统 202 所调用的组件代理相对应,则组件代理 208 可以在服务器组件 206 上的其它附加组件 216 中和/或在系统 200 可以连接或联网到的其他虚拟环境和系统中查找相对应的附加组件。

[0031] 可以理解,上述实现也可以被扩展到在其他场景中在虚拟环境中运行附加组件。例如,在计算机网络上可以存在若干虚拟环境的场景中,其可以具有若干虚拟应用程序和在其上运行的相关联的附加组件。可以有同一应用程序和附加组件的多个版本在网络上的不同服务器设备中运行。此外,同一虚拟应用程序的具有不同配置的实例可以在同一计算设备上运行。

[0032] 在另一示例性场景中,客户机设备可以访问与在实现服务器-客户机体系结构的网络上的服务器或客户机设备中所主存的虚拟应用程序相关联的附加组件。例如,客户机设备的操作系统 202 可以调用对应于包括在驻留在服务器组件 206 上的其它附加组件 216 中的附加组件的组件代理,其中服务器组件 206 驻留在分开的计算设备中。在这种情况下,

操作系统 202 建立与查找服务器组件 206 中的附加组件的虚拟环境 204 的通信链路。如果该附加组件位于其他附加组件 216 中,则客户机设备可以经由虚拟环境 204 访问该附加组件 216,而无需将该附加组件下载到客户机设备自身中。

[0033] 在另一实现中,运行多个虚拟环境的客户机计算设备还可以访问来自存在于该客户机计算设备所连接到的网络上的其它虚拟环境的附加组件,以便于共享附加组件。

[0034] 图 3 是示出在虚拟环境中运行附加组件的实现的示例性过程图 300。过程图 300 被示为逻辑流程图中的表示可以用硬件、软件、固件或其组合实现的操作序列的步骤集合。描述该方法的次序并非旨在被解释为限制,并且任何数目的所描述的方法框可以按任何次序组合以实现该方法或替换方法。此外,可以从该方法中删除各个框而不背离本文中所描述的主题的精神和范围。

[0035] 在软件上下文中,各框可以表示当由一个或多个处理器执行时完成所述操作的计算机指令。出于讨论的目的,参考图 1 中所示的系统 100 和图 2 中所示的系统 200 来描述过程 300。

[0036] 在框 302 处,操作系统调用对应于附加组件的组件代理以便使用该附加组件的功能。例如,操作系统 102 调用组件代理 106。在另一示例中,操作系统 202 调用组件代理 208。

[0037] 在框 304 处,组件代理使用 API 来与虚拟环境相关联。例如,组件代理 106 可以使用 RPC 114 来与系统 100 中的虚拟环境 104 相关联;或者组件代理 208 可以使用 RPC 218 来与虚拟环境 204 相关联。

[0038] 在框 306 处,组件代理确定相关联的虚拟环境是否是正确的虚拟环境。为此,组件代理检查虚拟环境是否包括操作系统所请求的相对应的附加组件。如果确定虚拟环境是正确的虚拟环境(即沿着自框 306 的“是”分支),则过程流程控制继续前进到框 308。否则(即沿着自框 306 的“否”分支),过程流程控制继续前进到框 304 以便建立与另一虚拟环境的关联,直到找到正确的虚拟环境。例如,如果在虚拟环境 104 中未检测到附加组件 112,则组件代理 106 在其他虚拟环境中查找附加组件 112。在另一实现中,在系统 200 上遵循确定正确的虚拟环境的同一过程。

[0039] 在框 308 处,一旦找到正确的虚拟环境,则组件代理在正确的虚拟环境中启动组件服务器进程。例如,组件代理 106 在确认附加组件 112 主存在虚拟环境 104 中之后,在虚拟环境 104 中启动组件服务器进程 108。在另一实现中,在系统 200 上,组件代理 208 在虚拟环境 204 中启动组件服务器进程 210。

[0040] 对于某些实现,组件代理将使用足够的信息来注册以直接连接到在适当的虚拟环境中运行的服务器。如果服务器没有在正确的虚拟环境中运行,则组件代理可以检测到这一点并且启动该服务器。在其他实现中,组件代理能够访问每个虚拟环境的清单,以确定在不启动虚拟环境内的服务器并且不与该服务器通信情况下支持什么组件。

[0041] 在框 310 处,在操作系统中的组件代理和虚拟环境中的组件服务器进程之间建立(即设立)通信链路,这样使得可以在操作系统和虚拟环境之间传递数据和命令。该通信链路可以使用 RPC 或任何其他公知技术来建立。例如,操作系统 102 可以在激活组件服务器进程 210 之后,建立通信链路以便与虚拟环境 104 交换数据和命令。在另一实现中,可以在系统 200 上使用相同的方法来建立通信链路。

[0042] 在框 312 处,组件服务器进程在虚拟环境中创建对应于组件代理的附加组件的实例。此外,可以在一个或多个虚拟环境上创建附加组件的多于一个的实例。例如,组件服务器进程 108 响应于从组件代理 106 处接收到的通信来在虚拟环境 104 中创建附加组件 112 的实例。在另一实现中,可以使用针对系统 100 所描述的同一种方法来在系统 200 中创建附加组件的实例。

[0043] 在框 314 处,一旦创建了附加组件,则组件代理将来自操作系统的命令转发给组件服务器进程,组件服务器进程进而将这些命令转发给附加组件。由此,操作系统可以访问附加组件并且可以由附加组件来扩展。例如,一旦操作系统 102 在虚拟环境 104 中定位了附加组件 112,则操作系统 102 可以使用由附加组件 112 所提供的功能。

[0044] 示例性计算机环境

[0045] 图 4 示出了可用于实现本文中所描述的技术或者可以整体或部分地代表本文中所描述的元素示例性通用计算机环境 400。例如,通用计算机环境可用于图 1 的系统 100 和图 2 的系统 200,并且实现图 3 的过程 300 中所描述的步骤。

[0046] 计算机环境 400 仅是计算环境的一个示例,而非旨在对计算机和网络体系结构的使用范围或功能提出任何限制。计算机环境 400 也不应被解释成对于在示例计算机环境 400 中所示出的任一组件或其组合有任何依赖或要求。

[0047] 计算机环境 400 包括计算机 402 形式的通用的基于计算的设备。计算机 402 可以例如是台式计算机、手持式计算机、笔记本或膝上型计算机、服务器计算机、游戏控制台等。计算机 402 的组件可以包括但不限于一个或多个处理器或处理单元 404、系统存储器 406、和将包括处理器 404 在内的各系统组件耦合到系统存储器 406 的系统总线 408。

[0048] 系统总线 408 表示任何若干种类型的总线结构中的一个或多个,包括存储器总线或存储器控制器、外围总线、加速图形端口、和使用各种总线体系结构中的任意一种的处理器或局部总线。作为示例,这样的体系结构可以包括工业标准体系结构 (ISA) 总线、微通道体系结构 (MSA) 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)。

[0049] 计算机 402 通常包括各种计算机可读介质。这种介质可以是能由计算机 402 访问的任何可用介质,而且包含易失性和非易失性介质、可移动和不可移动介质。

[0050] 系统存储器 406 包括诸如随机存取存储器 (RAM) 410 等易失性形式的计算机存储介质,和 / 或诸如只读存储器 (ROM) 412 等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) 414 包含有助于诸如启动时在计算机 410 中的元件之间传递信息的基本例程,它通常被存储在 ROM 412 中。RAM 410 通常包含处理单元 404 可以立即访问和 / 或目前正在操作的数据和 / 或程序模块。

[0051] 计算机 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。作为示例,图 4 示出了对不可移动、非易失性磁介质 (未示出) 进行读写的硬盘驱动器 416,对可移动、非易失性磁盘 420 (例如“软盘”) 进行读写的磁盘驱动器 418,以及对诸如 CD-ROM、DVD-ROM 或其它光学介质等可移动、非易失性光盘 424 进行读写的光盘驱动器 422。硬盘驱动器 416、磁盘驱动器 418 和光盘驱动器 422 各自通过一个或多个数据介质接口 426 连接到系统总线 408。另选地,硬盘驱动器 416、磁盘驱动器 418 和光盘驱动器 422 可以通过一个或多个接口 (未示出) 连接到系统总线 408。

[0052] 盘驱动器及其相关联的计算机可读介质向计算机 402 提供对计算机可读指令、数据结构、程序模块和其它数据的非易失性存储。尽管该示例示出了硬盘 416、可移动磁盘 420 和可移动光盘 424, 然而可以理解, 可储存可由计算机访问的数据的其它类型的计算机可读介质也可用来实现过示例性计算系统和环境, 这些介质诸如磁带盒或其它磁存储设备、闪存卡、CD-ROM、数字多功能盘 (DVD) 或其它光存储、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) 等。

[0053] 多个程序模块可储存在硬盘 416、磁盘 420、光盘 424、ROM 412 和 / 或 RAM410 中, 作为示例, 包括操作系统 427、一个或多个应用程序 428、其它程序模块 430 以及程序数据 432。这些操作系统 427、一个或多个应用程序 428、其它程序模块 430 和程序数据 432 (或其某一组合) 中的每一个可实现支持分布式文件系统的所有或部分常驻组件。

[0054] 用户可以通过诸如键盘 434 和定点设备 436 (如, “鼠标”) 等输入设备向计算机 402 输入命令和信息。其它输入设备 438 (未具体示出) 可包括话筒、操纵杆、游戏手柄、圆盘式卫星天线、扫描仪等等。这些和其它输入设备通过耦合至系统总线 408 的输入 / 输出接口 440 连接到处理单元 404, 但也可以通过其它接口和总线结构连接, 如并行端口、游戏端口、或通用串行总线 (USB)。

[0055] 监视器 442 或其它类型的显示设备也通过接口, 如视频适配器 444 连接到系统总线 408。除监视器 442 之外, 其它输出外围设备可包括诸如扬声器 (未示出) 和打印机 446 等组件, 它们可通过输入 / 输出接口 440 连接到计算机 402。

[0056] 计算机 402 可以使用到一个或多个远程计算机, 如远程基于计算的设备 448 的逻辑连接在网络化环境中操作。作为示例, 远程基于计算的设备 448 可以是个人计算机、便携式计算机、服务器、路由器、网络计算机、对等设备或其它常见的网络节点等等。远程基于计算的设备 448 被示为可包括此处相对于计算机 402 所描述的许多或所有元件和特征的便携式计算机。

[0057] 计算机 402 和远程计算机 448 之间的逻辑连接被描述为局域网 (LAN) 440 和通用广域网 (WAN) 452。这类网络环境常见于办公室、企业范围计算机网络、内联网以及因特网。

[0058] 当在 LAN 网络环境中实现时, 计算机 402 通过网络接口或适配器 444 连接至局域网 440。当在 WAN 网络环境中使用时, 计算机 402 通常包括调制解调器 446 或用于通过广域网 442 建立通信的其它装置。调制解调器 446 可以对计算机 102 是内置或外置的, 它可通过输入 / 输出接口 440 或其它适当的机制连接至系统总线 408。可以理解, 所示出的网络连接是示例性的, 可以使用在计算机 402 和 448 建立通信链路的其它手段。

[0059] 在诸如对计算环境 400 示出的网络化环境中, 相对于计算机 402 描述的程序模块或其部分可储存在远程存储器存储设备中。作为示例, 远程应用程序驻留在远程计算机 448 的存储器设备中。出于说明的目的, 应用程序和诸如操作系统等其它可执行程序组件在此处被示出为离散的框, 但可以认识到, 这些程序和组件在不同的时刻驻留在基于计算的设备 402 的不同存储组件中, 并由计算机的 (诸) 数据处理器来执行。

[0060] 各个模块和技术可以在诸如程序模块等由一个或多个计算机或其它设备执行的计算机可执行指令的一般上下文中描述。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。通常, 程序模块的功能可以在各实施例中按需进行组合或分布。

[0061] 这些模块和技术的实现可以被存储在某种形式的计算机可读介质中或者通过某种形式的计算机可读介质传输。计算机可读介质可以是可由计算机访问的任何可用介质。作为示例而非限制,计算机可读介质可以包括计算机存储介质和通信介质。

[0062] 计算机存储介质包括以用于存储诸如计算机可读指令、数据结构、程序模块或其它数据等信息的任何方法或技术来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限于, RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘 (DVD) 或其它光学存储、磁带盒、磁带、磁盘存储或其它磁存储设备,或者可用于存储所需信息并且可由计算设备访问的任何其它介质。

[0063] 另选地,框架的各部分可以用硬件或硬件、软件和 / 或固件的组合来实现。例如可以设计或编程一个或多个专用集成电路 (ASIC) 或可编程逻辑器件 (PLD) 来实现该框架的一个或多个部分。

[0064] 结论

[0065] 尽管用对结构特征和 / 或方法专用的语言描述了用于在虚拟环境中运行附加组件的各实施例,但可以理解,所附权利要求书的主题不必限于所描述的具体特征或动作。相反,这些具体特征和动作是作为用于在虚拟环境中运行附加组件的示例性实现来公开的。

100

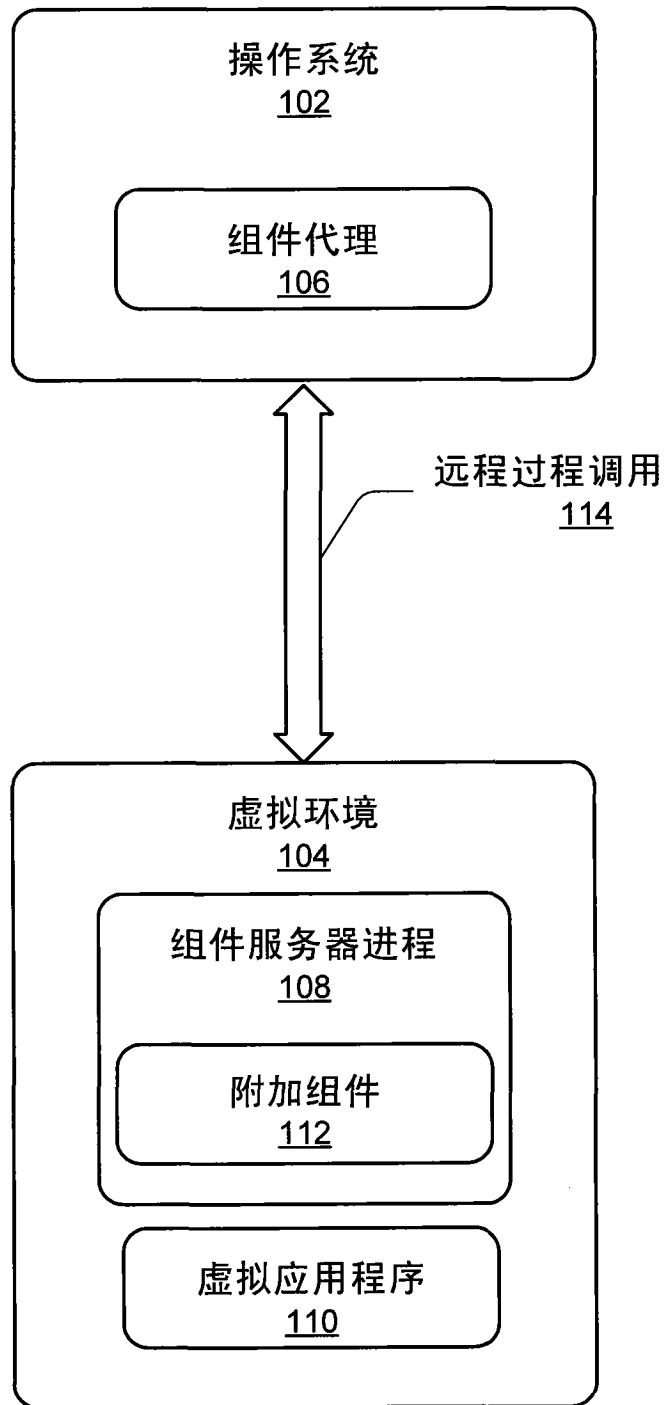


图 1

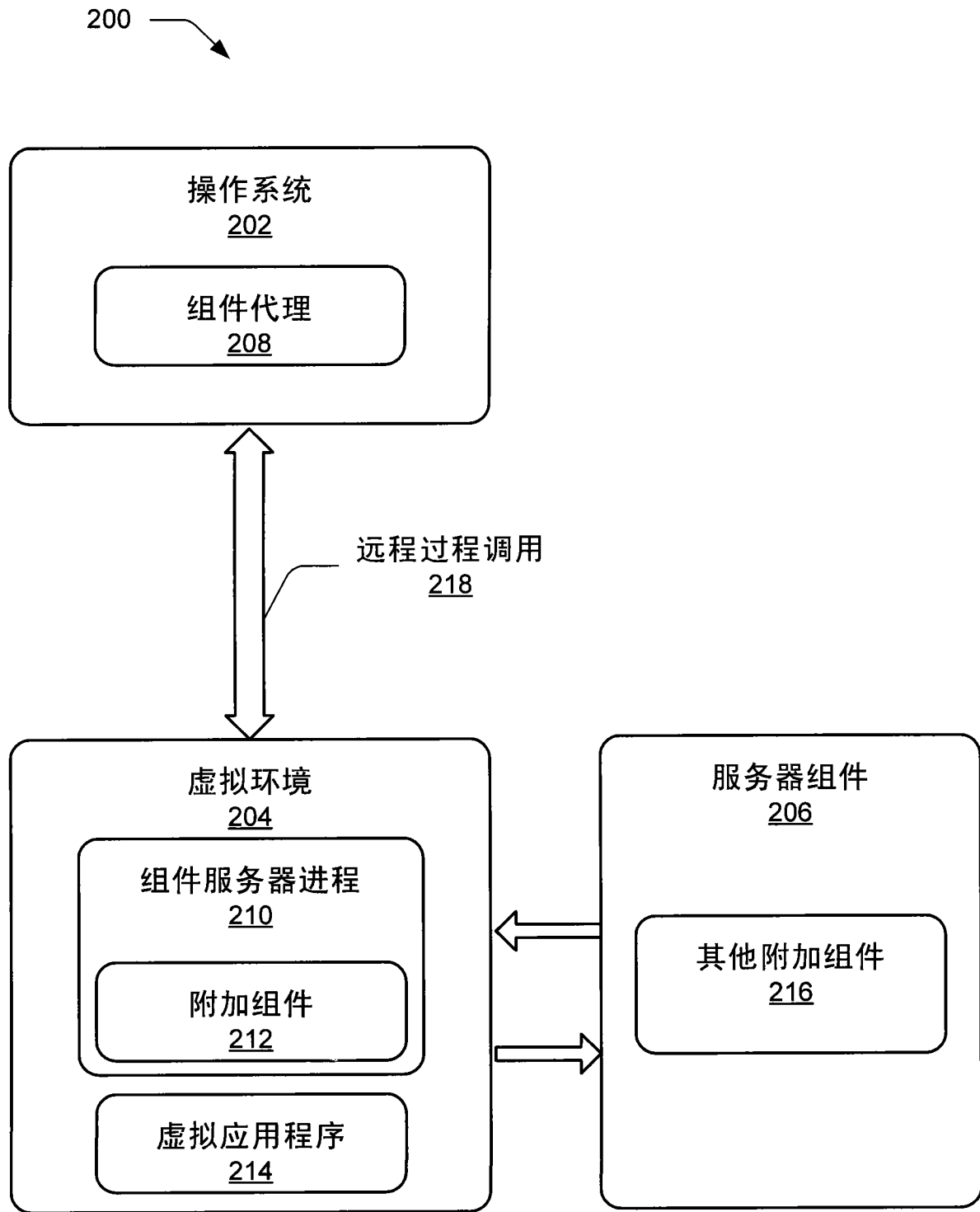


图 2

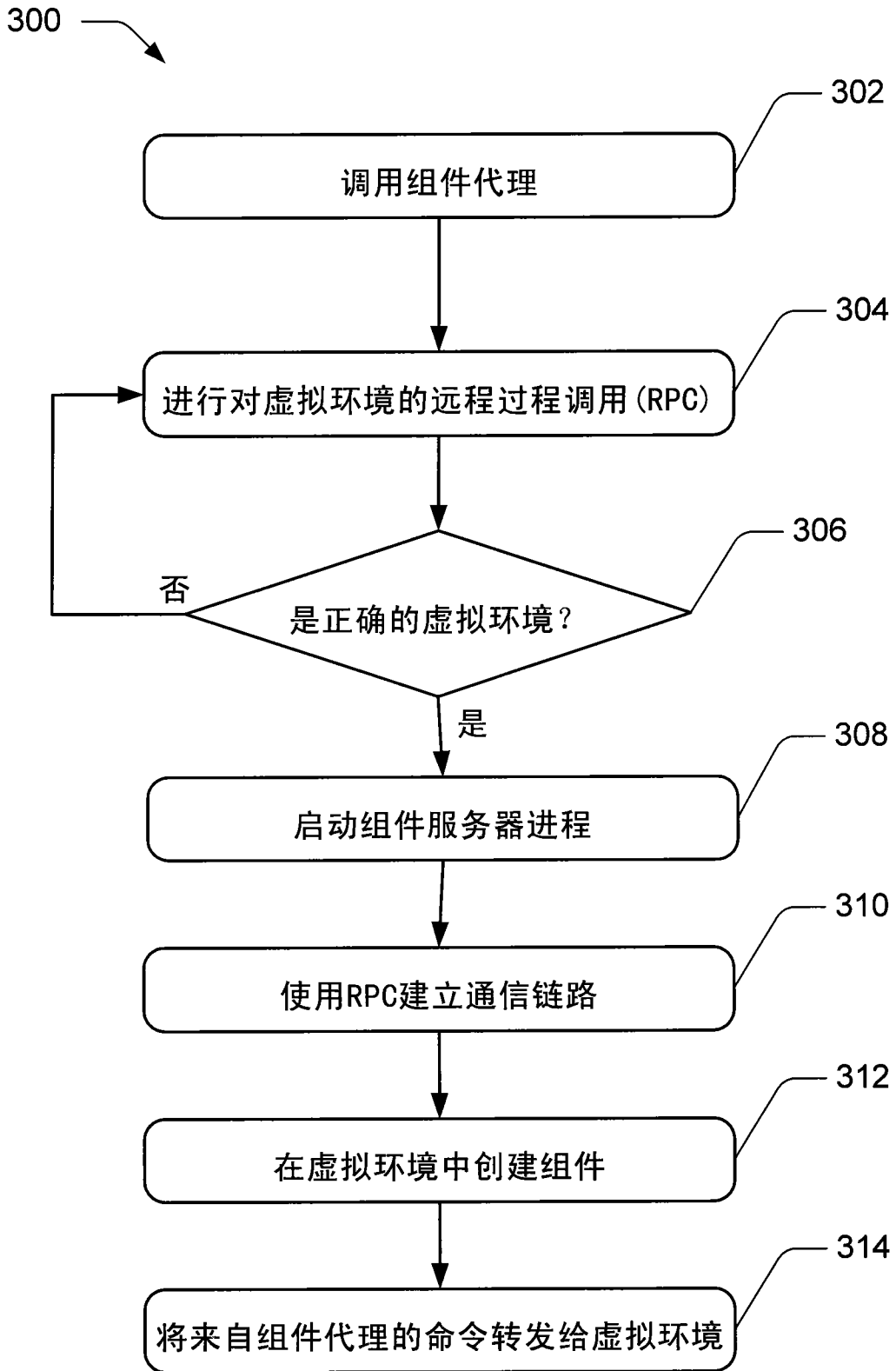


图 3

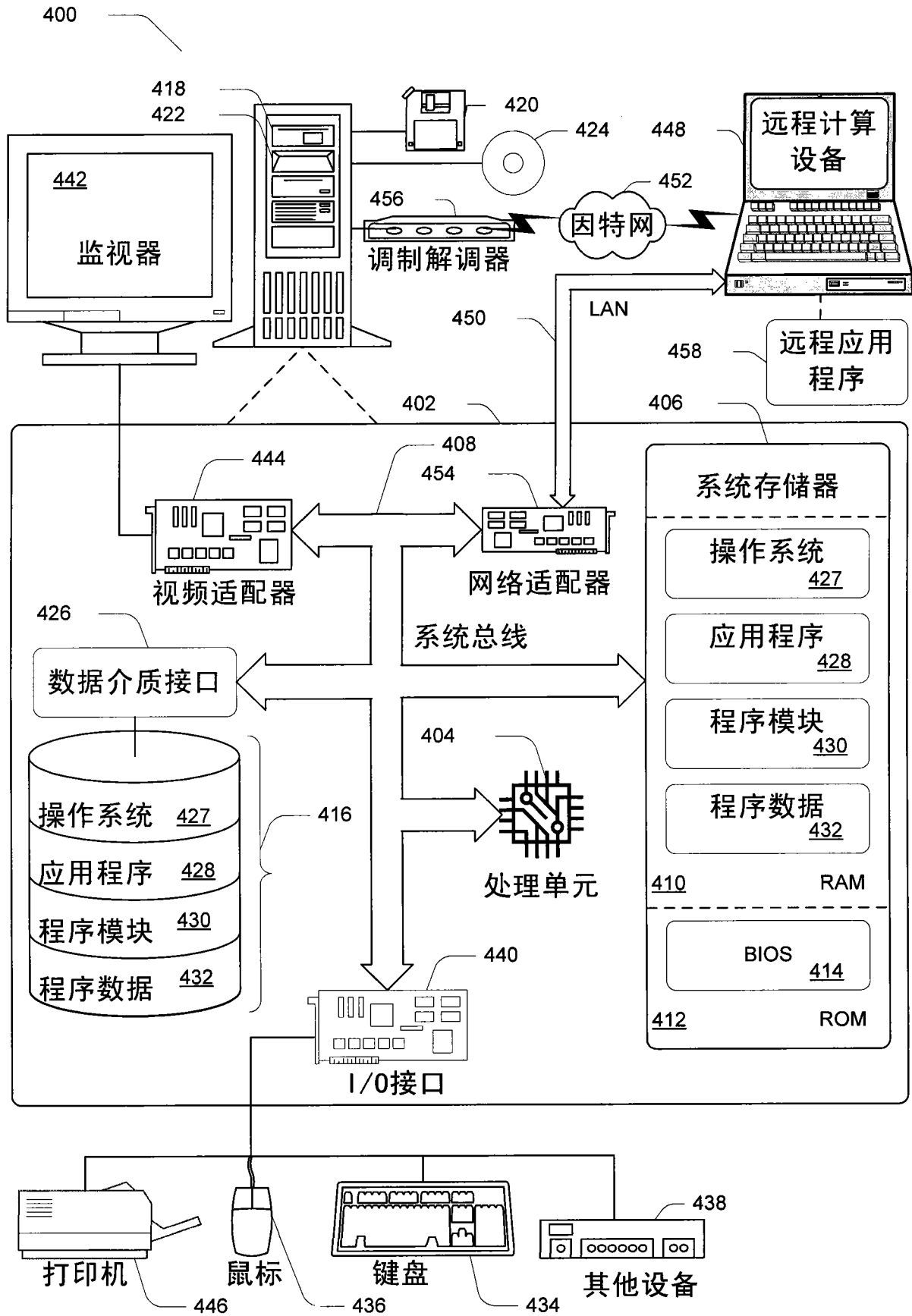


图 4