



(12) 发明专利申请

(10) 申请公布号 CN 103493016 A

(43) 申请公布日 2014. 01. 01

(21) 申请号 201180070307. 3

(22) 申请日 2011. 04. 21

(85) PCT国际申请进入国家阶段日
2013. 10. 21

(86) PCT国际申请的申请数据
PCT/US2011/033367 2011. 04. 21

(87) PCT国际申请的公布数据
W02012/145002 EN 2012. 10. 26

(71) 申请人 惠普发展公司, 有限合伙企业
地址 美国德克萨斯州

(72) 发明人 詹姆斯·M·曼

(74) 专利代理机构 北京德琦知识产权代理有限公司 11018
代理人 于未茗 康泉

(51) Int. Cl.
G06F 9/44 (2006. 01)

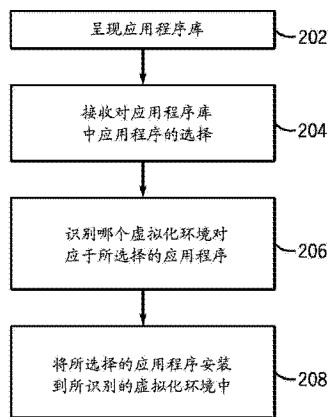
权利要求书2页 说明书5页 附图3页

(54) 发明名称

将应用程序安装到虚拟化环境中

(57) 摘要

一种电子设备具有多个虚拟化环境。对应用程序库中的应用程序的选择被接收, 并且虚拟化环境被识别以安装从所述应用程序库中选择的应用程序。



1. 一种电子设备的方法,包括:
在所述电子设备处呈现应用程序库的表示,其中所述电子设备具有各自被配置有各个不同类型操作系统的多个虚拟化环境;
接收对所述应用程序库中的应用程序的选择;
识别所述多个虚拟化环境中的哪个虚拟化环境对应于所述应用程序;以及
将所述应用程序安装到所识别的虚拟化环境中。
2. 根据权利要求1所述的方法,其中所述识别包括访问与所述应用程序相关联的信息以确定所述应用程序与所述操作系统中的哪个操作系统相关联。
3. 根据权利要求2所述的方法,其中访问所述信息包括访问所述应用程序的进程程序或安装程序中的元数据。
4. 根据权利要求2所述的方法,其中访问所述信息包括访问与所述应用程序的源相关的信息。
5. 根据权利要求4所述的方法,其中访问与所述源相关的信息包括访问指示所述应用程序与多个单个应用程序库中的哪个应用程序库相关联的信息。
6. 根据权利要求2所述的方法,其中访问所述信息包括访问所述应用程序的扩展名。
7. 根据权利要求1所述的方法,其中所述接收、识别和安装通过所述电子设备的虚拟化基础架构中的逻辑执行。
8. 根据权利要求7所述的方法,其中所述虚拟化基础架构包括控制域或虚拟机。
9. 一种包括存储指令的至少一个机器可读存储介质的物品,所述指令一旦执行就促使具有处理器的电子设备执行根据权利要求1所述的方法。
10. 一种电子设备包括:
至少一个处理器;
具有各个客户机操作系统的多个虚拟化环境;以及
安装程序逻辑,用于:
接收对通过具有多个应用程序的应用程序库呈现的应用程序的选择;
访问与所选择的应用程序相关联的信息以识别所选择的应用程序要安装到所述虚拟化环境中的哪个虚拟化环境中;并且
将所选择的应用程序安装在所识别的虚拟化环境中。
11. 根据权利要求10所述的电子设备,其中所述安装程序逻辑是所述电子设备的虚拟化基础架构的一部分。
12. 根据权利要求10所述的电子设备,其中所述安装程序逻辑通过下列方式识别所选择的应用程序要安装到所述虚拟化环境中的哪个虚拟化环境中:
将所述多个虚拟化环境中的多个虚拟化环境识别为适合于安装所选择的应用程序;以及
执行用于从所述多个虚拟化环境中进行选择以安装所选择的应用程序的動作。
13. 根据权利要求12所述的电子设备,其中所述動作包括从下列動作中选择的至少一个:
提示用户选择所述多个虚拟化环境中的一个虚拟化环境;
使用至少一个标准来从所述多个虚拟化环境中自动选择;以及

使用至少一个标准来从所述多个虚拟化环境中选择,并且在给用户的提示中呈现所述多个虚拟化环境中选择的一个虚拟化环境,作为用于安装所选择应用程序的建议虚拟化环境。

14. 根据权利要求 10 所述的电子设备,其中所访问的信息包括从下列选择的至少一个:

所选择应用程序的进程程序或安装程序中的元数据;

所选择应用程序的源;以及

所选择应用程序的扩展名。

15. 根据权利要求 10 所述的电子设备,其中所述应用程序库为具有与不同类型操作系统相关联的应用程序的共享应用程序库,所述电子设备进一步包括:

各自包含所述应用程序的各个子集的单个应用程序库,其中所述单个应用程序库中的每个与所述不同类型操作系统中的相应一个相关联。

将应用程序安装到虚拟化环境中

背景技术

[0001] 诸如计算机或其它类型电子设备之类的物理机器能够被配置为包括能运行各自不同客户机操作系统的多个虚拟化环境。这种虚拟化环境有时也被称作虚拟机。多个虚拟化环境在物理机器中的存在允许针对不同操作系统所写的应用程序在相同的物理机器上的不同虚拟化环境中运行。

附图说明

[0002] 关于下图描述一些实施例：

[0003] 图 1A 至图 1B 为根据各种实施方式的示例系统的框图；

[0004] 图 2 为根据一些实施方式的从应用程序库加载应用程序的过程的流程图；以及

[0005] 图 3 为根据另外的实施方式的系统组件的框图。

具体实施方式

[0006] 在电子设备中，能够呈现包含表示各自不同应用程序的项目的应用程序库的图形表示。用户能够在应用程序库中选择一个或多个项目，以使各个应用程序安装到电子设备中。电子设备的示例包括计算机（桌上型计算机、笔记本电脑、平板电脑等）、个人数字助理（PDA）、移动电话、电子电器或其它类型的电子设备。

[0007] 应用程序库中的能够被安装到电子设备中的应用程序能够位于各种远程站点，例如网站或者为应用程序提供销售或免费下载的供应商的应用程序服务器。当用户选择应用程序库中与给定应用程序相对应的项目时，电子设备通过网络与远程站点通信以下载与给定应用程序相关联的代码用于安装在电子设备中。与给定应用程序相关联的“代码”指机器可读指令，一旦机器可读指令安装在电子设备中，机器可读指令就可在电子设备上运行。机器可读指令能够采用编译后代码、脚本或其它形式代码的形式。将应用程序“安装”到电子设备中指从源（例如远程站点）获得应用程序的代码，并且将该代码放置于在例如通过用户或通过应用程序调用时允许代码运行的状态下的电子设备中。代码被放置到电子设备的合适位置，以允许代码在被调用时运行。在安装代码之前，代码先前不存在于在响应于代码的调用允许代码运行的状态下的电子设备中。

[0008] 在一些实施方式中，电子设备能够被配置有多个虚拟化环境（也被称作“虚拟机”）。“虚拟化环境”或“虚拟机”指用于虚拟化或仿真实际机器的组件的布置。在随后的讨论中，术语“虚拟化环境”和“虚拟机”可互换使用。如下面进一步讨论的那样，当用户期望将应用程序安装到电子设备中时，多个虚拟化环境在电子设备中的存在给用户提出挑战。例如，应用程序可能针对特定操作系统编写，使得应用程序必须安装在包含特定操作系统的虚拟化环境中。如果用户手动确定多个虚拟化环境中的哪个是适合于应用程序安装的虚拟化环境，则这对用户来说可能是挑战。

[0009] 每个虚拟化环境的组件能够包括客户机操作系统和能够在客户机操作系统内操作的一个或多个应用程序。电子设备中的不同虚拟化环境能够包括不同类型的客户机操作

系统(例如, Unix 操作系统、Linux 操作系统、Windows[®]操作系统、WebOS 或其它类型的操作系统)。多个虚拟化环境在电子设备中的存在允许在电子设备中并发运行针对不同操作系统设计的应用程序。

[0010] 在具有多个虚拟化环境的电子设备中,由这种电子设备呈现的应用程序库可以包括用于不同类型操作系统的应用程序。与为不同类型操作系统提供不同单个应用程序库的布置相比,在公共应用程序库内呈现用于不同类型操作系统的应用程序提升了用户体验。用户能够容易地通过公共应用程序库进行浏览以查找用户可能感兴趣的应用程序。

[0011] 根据一些实施方式,提供技术或机制以允许所选择应用程序(通过用户在包含用于不同类型操作系统的应用程序的应用程序库中选择)自动安装到包括与所选择应用程序兼容的客户机操作系统的相应虚拟化环境中。通过使用根据一些实施方式的技术或机制,用户不必手动执行分析来确定多个虚拟化环境中的哪个是适合于所选择应用程序的安装的虚拟化环境,而手动执行分析对没有正确训练或引导的用户来说可能是挑战。

[0012] 图 1A 为包括多个虚拟化环境(或虚拟机)102A、102B 的示例电子设备 100 的框图。尽管在图 1A 中只是示出两个虚拟化环境,但应注意,在可替代实施方式中,可以在电子设备 100 中提供附加的虚拟化环境。如图 1A 中所示,虚拟化环境 102A 包括客户机操作系统 104A 和被设计为在客户机操作系统 104A 上运行的应用程序 106A。类似地,虚拟化环境 102B 包括客户机操作系统 104B 和被设计为在客户机操作系统 104B 上运行的应用程序 106B。客户机操作系统 104B 是与客户机操作系统 104A 不同类型的客户机操作系统。

[0013] 在一些示例中,电子设备 100 的硬件资源 108 包括处理器 112、存储器 114、持久性存储介质 116 和输入/输出(I/O)设备 118。注意,尽管从单数意义上对处理器 112、存储器 114、持久性存储介质 116 和 I/O 设备 118 进行引用,但硬件资源 108 也能够包括多个处理器、存储器、持久性存储介质和/或 I/O 设备。

[0014] 电子设备的硬件资源 108 的虚拟化通过虚拟机监控器(VMM)110 提供。VMM 也被称作管理程序。VMM110 管理硬件资源 108 的共享(通过虚拟化环境 102A、102B)。VMM110 虚拟化硬件资源 108,并且拦截来自客户机操作系统 104A、104B 的对这种硬件资源的请求。

[0015] 有效地,VMM110 在每个虚拟化环境的客户机操作系统与底层硬件资源 108 之间提供接口。在一些示例中,由 VMM110 提供的到客户机操作系统的接口被设计为仿效由实际硬件资源 108 提供的接口。

[0016] 在一些实施方式中,VMM110 为类型 1 (或本地、裸机)VMM 或被配置为在硬件资源 108 上直接运行以控制硬件资源 108 并与客户机操作系统交互的管理程序。在其它实施方式中,VMM110 可以为类型 2 (或托管的)VMM 或者在主机操作系统上运行的管理程序。例如,主机操作系统 120 在图 1A 中以虚线轮廓被描绘。如果 VMM110 为托管的 VMM,则将存在主机操作系统 120 以托管 VMM110。另一方面,如果 VMM110 为类型 1VMM,则将省略主机操作系统 120。

[0017] 电子设备 100 还包括显示设备 122。显示设备 122 与硬件资源 108 的图形控制器(其可以是 I/O 设备 118 之一)交互。显示设备 122 可以被控制以呈现应用程序库 124 的图形表示。应用程序库 124 包括表示可由用户选择的各个应用程序的项目 126 (例如图标)以引起各个所选择应用程序的安装。

[0018] 注意,I/O 设备 118 也可以包括用于允许电子设备 100 通过网络 128 与各个远程

站点 130 (例如,网站、应用程序服务器等)通信的网络接口控制器。每个远程站点 130 可以存储与表示在应用程序库 124 中的一个或多个应用程序相对应的代码。响应于在应用程序库 124 中对应用程序的选择,电子设备 100 通过网络 128 与各个远程站点 130 通信,以促使与所选择应用程序相对应的代码被下载到电子设备 100 用于在电子设备 100 中安装。

[0019] 从应用程序库 124 中选择的应用程序的安装可以由在电子设备 100 中运行的安装程序逻辑 132 控制。安装程序逻辑 132 可以被考虑为是应用程序库代理(用于应用程序库 124 的代理)。在一些实施方式中,应用程序库连接通过电子设备 100 的虚拟化基础架构代理。代理应用程序库连接指其中与应用程序库相关联的安装任务(例如响应于用户对应用程序库 124 中应用程序的选择的安装任务)被转发给作为电子设备 100 的虚拟化基础架构的一部分的安装程序逻辑 132 的操作。

[0020] 例如,安装程序逻辑 132 可以是位于虚拟化基础架构中的控制域 134 的一部分。控制域可以是域 0,域 0 是电子设备 100 初始启动时发起的第一域。注意,域 0 概念可应用于一些类型的 VMM110。域 0 包括特殊管理权限,并且可以具有对硬件资源 108 的直接访问权。域 0 可以被看作管理虚拟机。

[0021] 在其它示例中,安装程序逻辑 132 可以是服务虚拟机 136(其被配置为在电子设备 100 中执行各种管理任务)的一部分。作为又一替代,安装程序逻辑 132 可以是虚拟化环境 102A、102B 之一的一部分。

[0022] 图 1B 示出根据可替代实施方式的简化电子设备 100A。电子设备 100A 中的、存在于图 1A 的电子设备 100 中的元件利用相同的附图标记标出。如图 1B 中所示,电子设备 100A 包括虚拟化环境 102A 和 102B 以及能够显示应用程序库 124 的表示的显示设备 122。

[0023] 电子设备 100A 的硬件资源 108A 包括机器可读介质 150 和处理器 112。如所描绘的,在一些实施方式中,根据一些实施方式的安装程序逻辑 132 初始存储在机器可读存储介质 150 中并且被加载用于在处理器 112 上运行。

[0024] 图 2 为根据一些实施方式的由电子设备 100 执行的过程。应用程序库 124 的图形表示(图 1A 或图 1B)被呈现(在 202 处)用于显示。在一些示例中,应用程序库 124 可以由电子设备 100 中的应用程序库应用程序(未示出)呈现。应用程序库应用程序可以为虚拟化环境 102A、102B 之一的一部分,或者可以为服务虚拟模块 136 的一部分,或者可以为电子设备 100 中其它模块的一部分。

[0025] 安装程序逻辑 132 接收(在 204 处)对应用程序库 124 中应用程序的选择。响应于该选择,安装程序逻辑 132 识别(在 206 处)多个虚拟化环境 102A、102B 中的哪个对应于所选择的应用程序。所选择的应用程序然后被安装(在 208 处)到所识别的虚拟化环境中。

[0026] 注意,在一些情况下,在 206 处执行的识别可以包括识别适合于所选择应用程序的安装的多个虚拟化环境(换句话说,所选择应用程序与所识别多个虚拟化环境中的每个客户机操作系统兼容)。在这种情况下,安装程序逻辑 132 可以提示用户选择所识别多个虚拟化环境中的哪个以安装所选择应用程序。可替代地,安装程序逻辑 132 可以使用一个或多个预定标准以自动从所识别多个虚拟化环境中选择。例如,安装程序逻辑 132 可以选择:(1)所识别多个虚拟化环境中最近使用的虚拟化环境,或者(2)所识别多个虚拟化环境中的、目前在前景中或活动的一个虚拟化环境。由于存在多个虚拟化环境,因此虚拟化环境中的一些可以在电子设备中活动地执行任务(这种虚拟化环境中的应用程序和/或操作系

统活动地运行,因此使用电子设备的硬件资源)。这种虚拟化环境被考虑为在前景中或活动的。相比之下,其它虚拟化环境可以是休眠的(应用程序和/或操作系统怠速并且不使用电子设备的硬件资源)。对于从所识别多个虚拟化环境中进行选择的其它标准可以用于其它示例中。

[0027] 一旦安装程序逻辑 132 采用预定的标准(或那些标准)以从所识别多个虚拟化环境中选择,安装程序逻辑 132 就能够自动将所选择的应用程序安装到所选择的虚拟化环境中,而不从用户进一步输入。可替代地,连同所识别的多个虚拟化环境一起,安装程序逻辑 132 可以首先向用户呈现所选择的虚拟化环境作为建议或意见。用户可以选择接受或忽视该建议或意见。

[0028] 所选择应用程序待安装入的虚拟化环境的识别(图 2 中的 206)可以基于若干不同机制之一。例如,元数据可以与应用程序库中的应用程序相关联。元数据可以包括在应用程序的进程程序(wrapper)或安装程序中。进程程序指与应用程序相关联的、被配置为在进程程序运行时调用应用程序的逻辑。应用程序的安装程序指与应用程序相关联的、被配置为在安装程序运行时将应用程序安装在系统中的逻辑。元数据可以识别特定应用程序与给定操作系统相关联,使得安装程序逻辑 132 能够基于元数据判定哪个虚拟化环境 102A、102B 是适合于安装所选择应用程序的虚拟化环境(通过将元数据指示的操作系统与每个虚拟化环境中的客户机操作系统进行匹配)。

[0029] 用于允许将应用程序与各个虚拟化环境 102A、102B 相关联的另一机制是将各个不同的应用程序(针对不同类型操作系统设计的)包括在各个唯一的、单个应用程序库中,例如图 3 中描绘的单个应用程序库 302、304 和 306。尽管图 3 中示出三个单个应用程序库 302、304 和 306,但注意,不同数目的单个应用程序库可以用于其它示例中。

[0030] 单个应用程序库 302 包含用于第一类型操作系统的应用程序,单个应用程序库 304 包含用于第二类型操作系统的应用程序,并且单个应用程序库 306 包含用于第三类型操作系统的应用程序。单个应用程序库 302、304 和 306 的内容可以被组合以呈现在所显示应用程序库 124 中。在根据图 3 的实现中,所显示应用程序库 124 可以看作是组合单个应用程序库 302、304 和 306 的内容的虚拟或摘要应用程序库。

[0031] 当在所显示应用程序库 124 中选择应用程序时,安装程序逻辑 132 能够将所选择应用程序的源识别为单个应用程序库 302、304 和 306 中之一。基于将各个单个应用程序库识别为源,安装程序逻辑 132 能够确定哪个虚拟化环境是安装所选择应用程序的正确虚拟化环境。

[0032] 作为又一替代,应用程序库 124 中应用程序的扩展名可以用于识别合适的虚拟化环境以安装所选择的应用程序。例如,具有 .exe 扩展的应用程序名指示应用程序用于 Windows[®]操作系统,因此应当安装在包括 Windows[®]操作系统的虚拟化环境中。作为另一示例,具有 .rpm 或 .yum 扩展的应用程序名能够指示应用程序与 Linux 操作系统相关联,因此应当安装在包含 Linux 操作系统的虚拟化环境中。

[0033] 尽管许多扩展是唯一的并且是明确的(关于应用程序针对哪个操作系统设计),但存在可能潜在地模糊的其它扩展。这种模糊扩展的示例包括 .xml 或 .html。具有这些扩展的应用程序可以是潜在地能够安装在具有不同操作系统的若干不同虚拟化环境之一中的插件。为了解决该模糊,可以使用前述技术之一,例如采用与应用程序相关联的元数据的技

术或其中用于不同操作系统的应用程序存储在不同单个应用程序库中的技术。

[0034] 可替代地,在扩展模糊的情况下,可以向用户呈现提示以允许用户选择哪个虚拟化环境适合于安装所选择的应用程序。

[0035] 通过使用根据一些实施方式的技术或机制,针对不同类型操作系统设计的应用程序可以呈现在公共或共享应用程序库中,同时仍然允许能够将所选择应用程序安装到具有多个虚拟化环境的电子设备中的合适虚拟化环境中。

[0036] 上述模块(包括图 1A 或图 1B 中示出的模块)的机器可读指令在处理器(例如图 1A 或图 1B 中的 112)上运行。处理器可以包括微处理器、微控制器、处理器模块或子系统、可编程集成电路、可编程门阵列或者其它控制或计算设备。

[0037] 数据和指令存储在为实现为一个或多个计算机可读或机器可读存储介质的各个存储设备中。存储介质包括不同形式的存储器,包括:诸如动态或静态随机存取存储器(DRAM 或 SRAM)、可擦除和可编程只读存储器(EPROM)、电可擦除和可编程只读存储器(EEPROM)和闪存之类的半导体存储设备;诸如固定磁盘、软盘和可移动磁盘之类的磁盘;包括磁带的其它磁介质;诸如光盘(CD)或数字视频盘(DVD)之类的光学介质;或其它类型的存储设备。注意,上面讨论的指令可以提供在一个计算机可读或机器可读存储介质上,或者可替代地,可以提供在分布于具有可能多个节点的大系统中的多个计算机可读或机器可读存储介质上。这种或这些计算机可读或机器可读存储介质被考虑为是物品(或制品)的一部分。物品或制品可以指制造的任意单个组件或多个组件。存储介质可以位于运行机器可读指令的机器中,或者位于通过网络从其下载机器可读指令以运行的远程站点处。

[0038] 在前面的描述中,大量细节被阐述以提供对这里公开的主题的理解。然而,可以在不具有这些细节中的一些或全部细节的情况下实践这些实现。其它实现可以包括对上面讨论的细节的修改和改变。目的是,所附权利要求涵盖这种修改和改变。

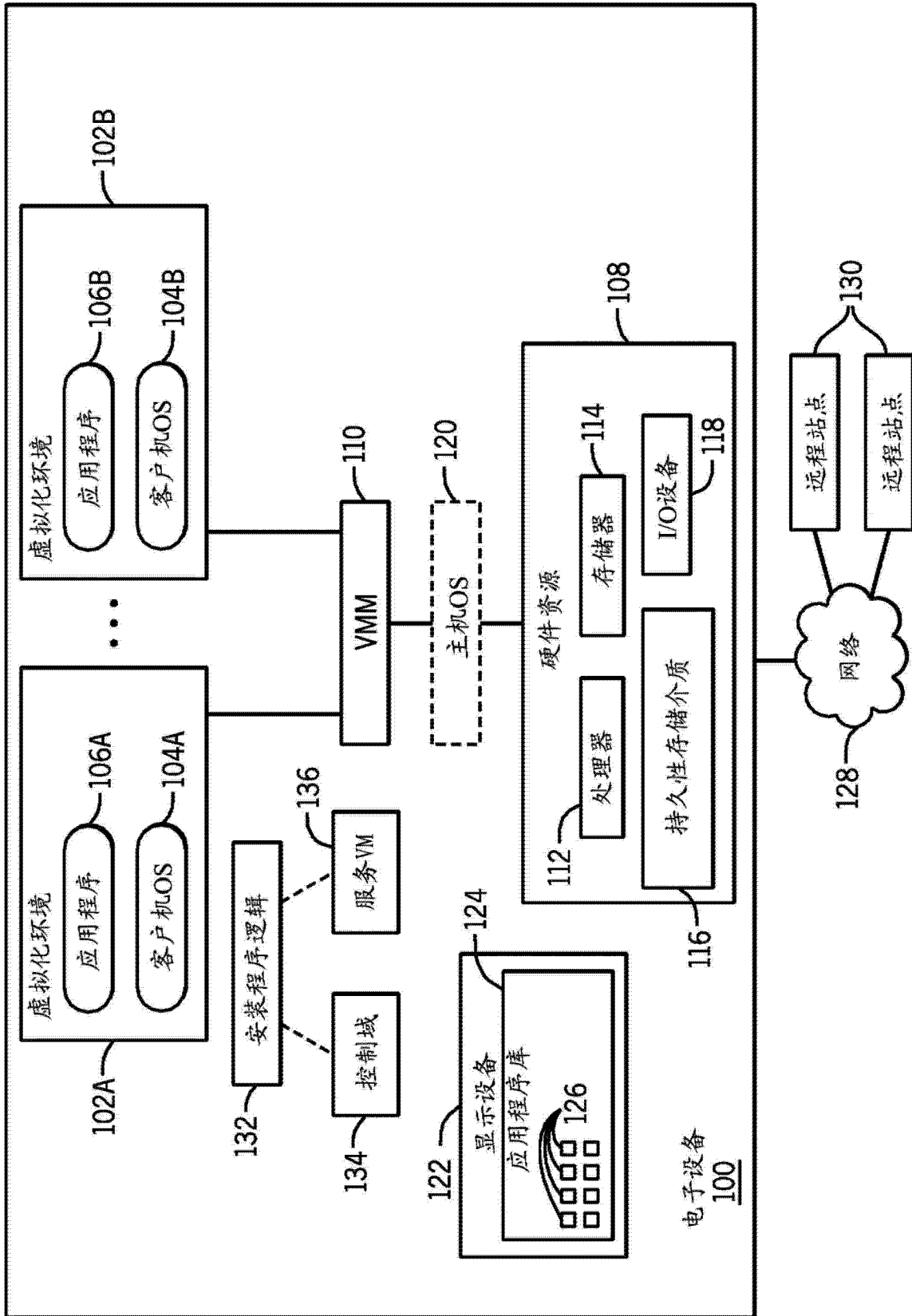


图 1A

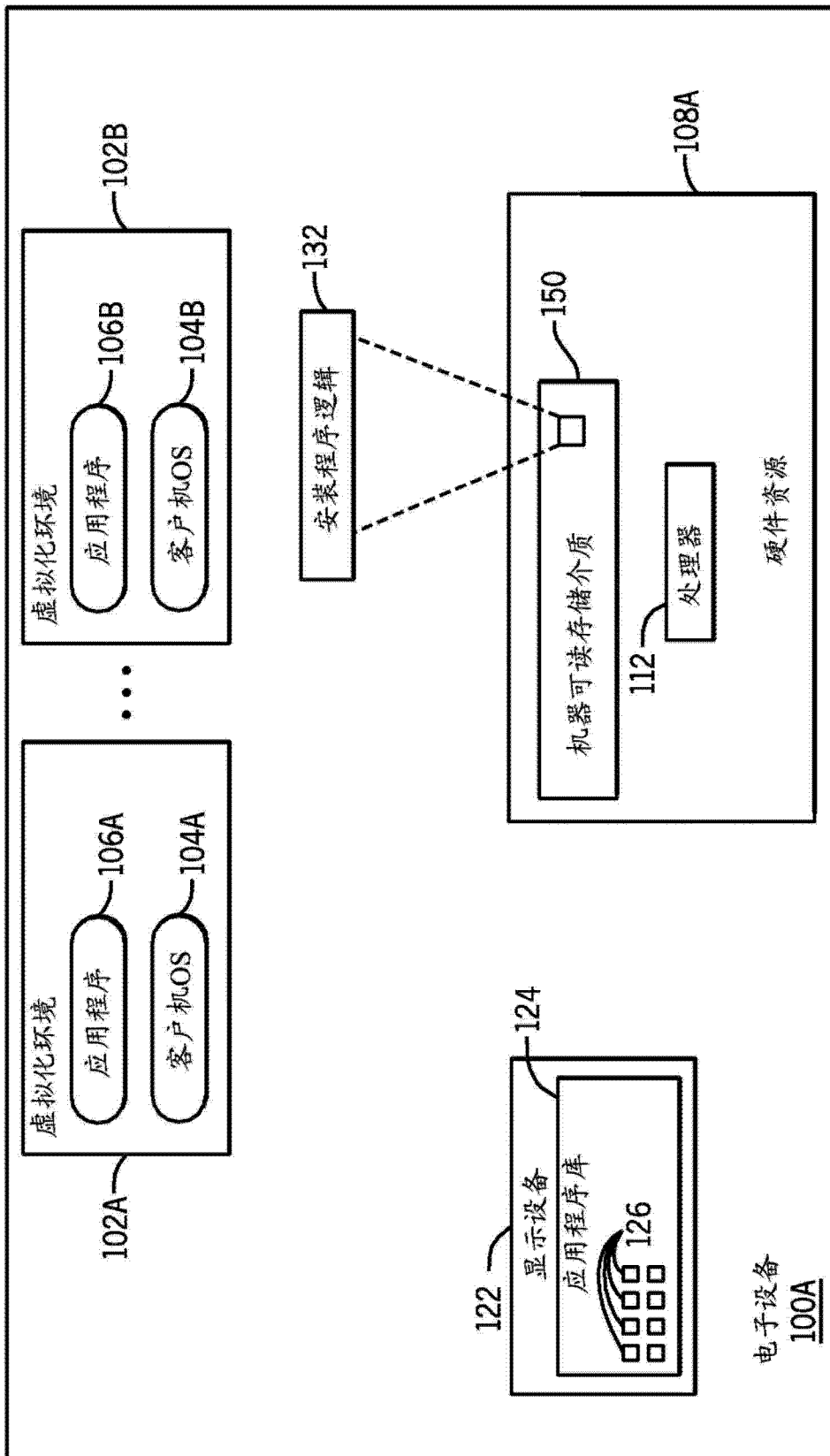


图 1B

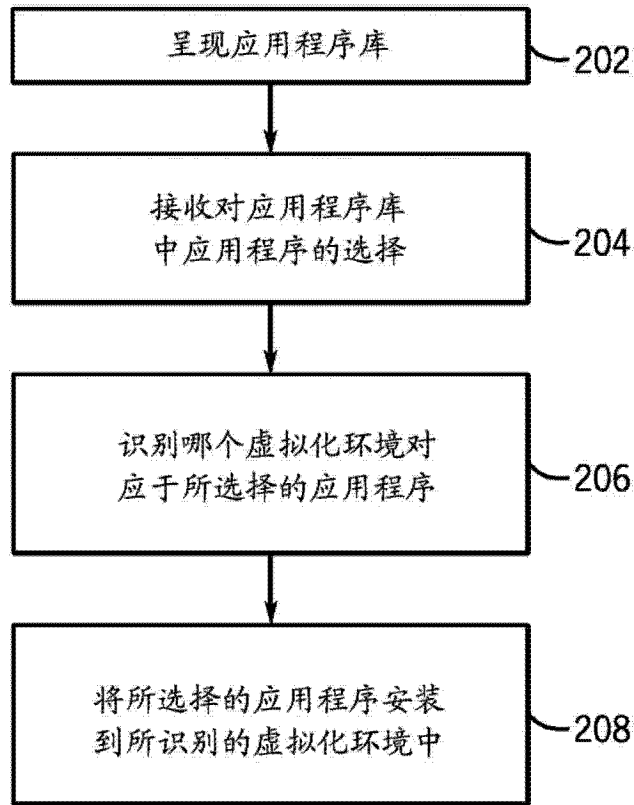


图 2

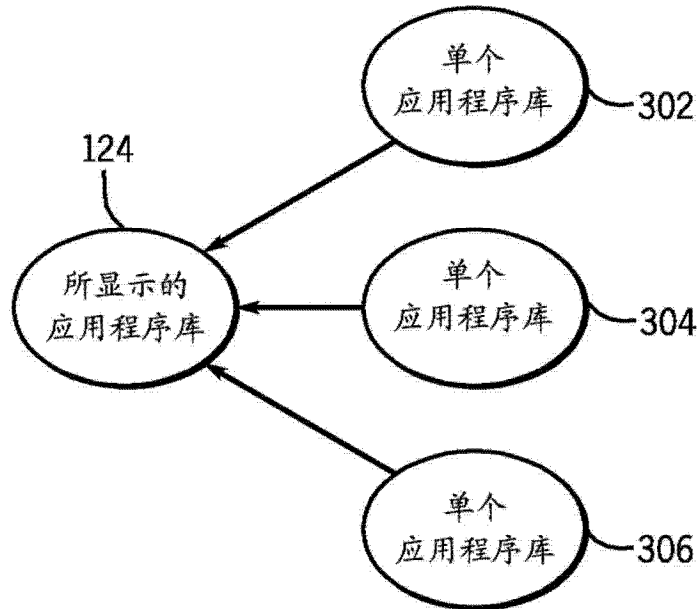


图 3