



(12) 发明专利申请

(10) 申请公布号 CN 103201719 A

(43) 申请公布日 2013. 07. 10

(21) 申请号 201180053754. 8

(74) 专利代理机构 北京市中咨律师事务所  
11247

(22) 申请日 2011. 10. 28

代理人 于静 张亚非

(30) 优先权数据

12/953, 483 2010. 11. 24 US

(51) Int. Cl.

G06F 9/455(2006. 01)

(85) PCT申请进入国家阶段日

2013. 05. 08

(86) PCT申请的申请数据

PCT/EP2011/068987 2011. 10. 28

(87) PCT申请的公布数据

W02012/069276 EN 2012. 05. 31

(71) 申请人 国际商业机器公司

地址 美国纽约

(72) 发明人 A·特雷格 S·本耶胡达

B·A·亚苏尔 M·法克特

E·博罗维克 Z·马舒尔斯基

K·梅特

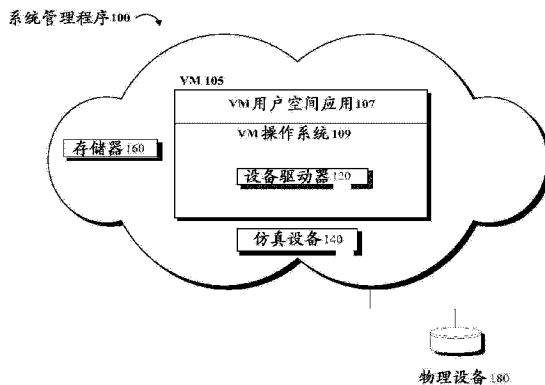
权利要求书2页 说明书7页 附图6页

(54) 发明名称

虚拟化计算环境中的设备仿真

(57) 摘要

提供了用于优化虚拟化计算环境中的物理设备的操作的系统和方法。所述方法包括：监视在虚拟机（VM）的操作系统上运行的应用发出的指令，其中所述 VM 由系统管理程序托管，所述系统管理程序提供对连接到虚拟化计算环境的物理设备的访问；其中所述操作系统所支持的设备驱动器向所述系统管理程序所支持的仿真设备发出一个或多个指令，以便根据所发出的指令控制所述物理设备，其中所述仿真设备的寄存器在所述系统管理程序的存储器中实现，并且所述设备驱动器发出的指令存储在所述寄存器中，其中响应于获知所述指令存储在所述寄存器中，所述系统管理程序处理所述指令。



1. 一种用于优化虚拟化计算环境中的物理设备的操作的计算机实现的方法,所述方法包括:

监视在虚拟机(VM)的操作系统上运行的应用发出的指令,其中所述 VM 由系统管理程序托管,所述系统管理程序提供对连接到虚拟化计算环境的物理设备的访问;

其中所述操作系统所支持的设备驱动器向所述系统管理程序所支持的仿真设备发出一个或多个指令,以便根据所发出的指令控制所述物理设备,

其中所述仿真设备的寄存器在所述系统管理程序的存储器中实现,并且所述设备驱动器发出的指令存储在所述寄存器中,

其中响应于获知所述指令存储在所述寄存器中,所述系统管理程序处理所述指令。

2. 根据权利要求 1 的方法,其中所述操作系统控制所发出的指令是否存储在所述寄存器中。

3. 根据权利要求 1 的方法,其中所述仿真设备监视所发出的指令是否存储在所述寄存器中。

4. 根据权利要求 1 的方法,其中所述系统管理程序监视所发出的指令是否存储在所述寄存器中,并且向所述仿真设备通知所述指令存储在所述寄存器中。

5. 一种处理用于管理虚拟化计算环境中的物理设备的指令的计算机实现的方法,所述方法包括:

监视设备驱动器是否向仿真设备发出指令,其中所述设备驱动器在虚拟机(VM)操作系统上运行,并且所述仿真设备在托管所述 VM 的系统管理程序上运行;

响应于判定将指令写入到所述仿真设备的寄存器,触发捕获以便向所述仿真设备通知所述设备驱动器发出指令;

其中所述系统管理程序将对处理所述指令的 CPU 的控制从所述 VM 切换到所述系统管理程序。

6. 根据权利要求 5 的方法,其中所述系统管理程序处理所述指令。

7. 根据权利要求 5 的方法,其中所述仿真设备处理所述指令。

8. 根据权利要求 5 的方法,其中所述 VM 操作系统监视所述设备驱动器是否发出指令。

9. 根据权利要求 5 的方法,其中所述仿真设备的寄存器在所述系统管理程序的存储器中实现。

10. 一种处理用于管理虚拟化计算环境中的物理设备的指令的计算机实现的方法,所述方法包括:

监视由设备驱动器向仿真设备发出的指令是否被写入到所述仿真设备的寄存器,其中所述设备驱动器在虚拟机(VM)操作系统上运行,并且所述仿真设备在托管所述 VM 的系统管理程序上运行;

处理所发出的指令。

11. 根据权利要求 10 的方法,其中所述仿真设备的寄存器在所述系统管理程序的存储器中实现。

12. 根据权利要求 10 的方法,其中所述系统管理程序执行所述监视。

13. 根据权利要求 10 的方法,其中所述仿真设备执行所述监视。

14. 根据权利要求 10 的方法,其中所述系统管理程序处理所发出的指令。

15. 根据权利要求 10 的方法,其中所述仿真设备处理所发出的指令。

16. 一种用于优化虚拟化计算环境中的物理设备的操作的计算机实现的系统,所述系统包括:

逻辑单元,其用于监视在虚拟机(VM)的操作系统上运行的应用发出的指令,其中所述 VM 由系统管理程序托管,所述系统管理程序提供对连接到虚拟化计算环境的物理设备的访问;

设备驱动器,其由所述操作系统支持并用于向所述系统管理程序所支持的仿真设备发出一个或多个指令,以便根据所发出的指令控制所述物理设备,

其中所述仿真设备的寄存器在所述系统管理程序的存储器中实现,并且所述设备驱动器发出的指令存储在所述寄存器中,

其中响应于获知所述指令存储在所述寄存器中,所述系统管理程序处理所述指令。

17. 根据权利要求 16 的系统,其中所述操作系统控制所发出的指令是否存储在所述寄存器中。

18. 根据权利要求 16 的系统,其中所述仿真设备监视所发出的指令是否存储在所述寄存器中。

19. 根据权利要求 16 的系统,其中所述系统管理程序监视所发出的指令是否存储在所述寄存器中,并且向所述仿真设备通知所述指令存储在所述寄存器中。

## 虚拟化计算环境中的设备仿真

### 技术领域

[0001] 本发明一般地涉及虚拟机,更具体地说,涉及一种用于优化在虚拟化环境中仿真的设备的数据操作的系统和方法。

### 背景技术

[0002] 虚拟机(VM)是像物理机器那样执行程序的机器(即,计算机)的软件实现。通常,对虚拟化环境中的资源的访问与高开销水平关联,这是因为 VM 通常无法直接与虚拟化资源通信。托管 VM 的系统管理程序通常负责代表 VM 仿真所需的接口或驱动器。换言之,系统管理程序为 VM 提供虚拟平台,并监视 VM 的执行以及 VM 如何访问可用的硬件资源和设备。

[0003] 例如,为使 VM 从虚拟化存储设备读取数据或将数据写入虚拟化存储设备,必须在 VM 与系统管理程序之间转移对读取或写入操作的控制。在完全仿真的情况下,VM 并不知道仿真设备不是实际物理设备,因为系统管理程序以软件仿真接口。此类仿真将产生高开销。当访问设备时,如果需要在 VM 与系统管理程序之间间歇性地转移控制,则操作开销甚至更昂贵。

### 发明内容

[0004] 出于概述目的,在此描述了某些方面、优点和新颖特性。将理解,根据任何一个特定实施例,可能未实现所有这些优点。因此,可以以如下方式包含或执行所公开的主题:实现或优化一个优点或一组优点,而未实现可能在此教导或提出的所有优点。

[0005] 提供了用于优化虚拟化计算环境中的物理设备的操作的系统和方法。所述方法包括:监视在虚拟机(VM)的操作系统上运行的应用发出的指令,其中所述 VM 由系统管理程序托管,所述系统管理程序提供对连接到虚拟化计算环境的物理设备的访问;其中所述操作系统所支持的设备驱动器向所述系统管理程序所支持的仿真设备发出一个或多个指令,以便根据所发出的指令控制所述物理设备,其中所述仿真设备的寄存器在所述系统管理程序的存储器中实现,并且所述设备驱动器发出的指令存储在所述寄存器中,其中响应于获知所述指令存储在所述寄存器中,所述系统管理程序处理所述指令。

[0006] 根据一个或多个实施例,提供了一种包括一个或多个逻辑单元的系统。所述一个或多个逻辑单元被配置为执行与上面公开的方法关联的功能和操作。在另一个实施例中,提供了一种计算机程序产品,其包括具有计算机可读程序的计算机可读存储介质。当在计算机上执行时,所述计算机可读程序导致计算机执行与上面公开的方法关联的功能和操作。

[0007] 下面参考附图,进一步详细地提供了上面公开的一个或多个实施例以及某些备选实施例。但是,所公开的主题并不限于所公开的任何特定实施例。

### 附图说明

[0008] 通过参考下面提供的附图中的各图,可以更好地理解所公开的实施例,这些附图

是：

[0009] 图 1 和 2 示出了根据一个或多个实施例的示例性计算环境，其中实现系统管理程序托管的 VM 以便与可通过所述系统管理程序访问的设备通信；

[0010] 图 3A 和 3B 是根据一个实施例的用于处理指令以便控制图 1 和 2 中示出的设备的方法的示例性流程图；

[0011] 图 4A 和 4B 是根据一个或多个实施例的其中可以运行所公开的系统和方法的硬件和软件环境的框图。

[0012] 根据一个或多个实施例，不同图中的相同标号引用的特性、元素和方面表示相同、等效或类似的特性、元素或方面。

### 具体实施方式

[0013] 以下说明了许多具体细节以便提供对各种实施例的详尽描述。某些实施例可以在没有这些具体细节的情况下实现，或者可以具有细节的变化。在某些情况下，并未详细描述某些特性以免使其它方面模糊不清。与每个元素或特性关联的细节级别不应被解释为限定某个特性较之其它特性的新颖性或重要性。

[0014] 参考图 1，根据一个实施例，在计算环境中，可以由系统管理程序 100 托管 VM 105。可以实现系统管理程序 100 以支持物理设备（例如，磁盘存储设备）的仿真，以便允许 VM 105 与资源（例如，物理设备 180）通信。仿真设备 140 用作设备驱动器 120 和物理设备 180 之间的接口。例如，仿真设备 140 可以支持标准协议以便与设备驱动器 120 连接，其中所述协议定义设备寄存器和相应 I/O 功能的使用方式。

[0015] 仿真设备 140 可以以软件或固件实现以仿真某一物理设备，以便 VM 可以与物理设备 180 通信，而无需任何有关物理设备 180 的特定接口要求的确切结构或性质的知识。取决于实施方式，可以通过使用半虚拟化设备作为设备驱动器 120 和物理设备 180 之间的接口，减少与仿真关联的开销，并减少对在 VM 105 和系统管理程序 100 之间转移控制的需要。在一个实施例中，可以使用虚拟化组件（例如，定制驱动器）实现设备的半虚拟化，以便例如通过分批处理多个指令或 I/O 请求并使用共享存储器以避免多个数据副本而提高系统的性能。

[0016] 如图 1 中所示，可以在 VM 105 上执行的 VM 操作系统 109 上运行诸如软件代码之类的计算机实现的应用（例如，用户空间应用 107）。VM 操作系统 109 可以支持设备驱动器 120，并且系统管理程序 100 可以支持仿真设备 140，仿真设备 140 能够与设备驱动器 120 和物理设备 180 通信。在一种实施方式中，可以使用设备分配选项以便允许 VM 105 访问物理设备 180。设备分配如所期望地为 VM 105 授予对物理设备 180 的完全访问，以便 VM 105 可以直接向物理设备 180 发送 I/O 请求，其中可以通过系统管理程序 100 路由中断。如果物理设备 180 旨在在若干 VM 和系统管理程序之间共享并且不支持 VM 迁移，则可能不需要设备分配选项，这是因为 VM 将接口直接分配给特定设备。

[0017] 在一种实施方式中，仿真设备 140 具有寄存器，它们实际上在系统管理程序 100 的存储器 160 中实现。即，可以针对仿真设备 140 共享或保留系统管理程序 100 的存储器 160 的一部分，其中设备驱动器 120 发出的任何指令都写入到所共享的存储器区域。以此方式，设备驱动器 120 和仿真设备 140 可以被配置为允许系统管理程序 100 监视或控制物理设备

180 上的对应操作。

[0018] 在一个实施例中,写入到仿真设备 140 的寄存器(即,共享的系统管理程序 100 的存储器 160)的指令可以由 VM 操作系统 109 控制,或者由系统管理程序 100 或仿真设备 140 监视。要指出的是,仿真设备 140 的寄存器可以完全从系统管理程序 100 中访问,因为所述寄存器在系统管理程序 100 的存储器 160 中实现。根据一个实施例,当发出相应指令时,设备驱动器 120 并不知道它正在与仿真设备 140 通信而不是与物理设备 180 通信。

[0019] 如在下面进一步详细提供的那样,仿真设备 140 或系统管理程序 100 可以监视共享存储器 160,以便确定何时将指令从设备驱动器 120 发送到仿真设备 140。当检测到此类指令时,系统管理程序 100 通过仿真物理设备 180 的适当行为而参与处理所述指令。以此方式,提高物理设备 180 的整体操作性能,因为 VM 105 和系统管理程序 100 之间的控制切换次数减到最少。在一个实施例中,通过从 VM 105 执行单独的执行线程来监视共享存储器 160,以便允许仿真设备 140 或系统管理程序 100 确定指令是否被写入到存储器 160,同时 VM 105 继续运行。

[0020] 因此,在上面的示例性情景中,当设备驱动器 120 向在共享存储器 160 中实现的设备寄存器中写入时,VM 105 继续运行,同时在单独的执行线程中,仿真设备 140 或系统管理程序 100 继续监视写入到共享存储器 160 中的设备寄存器的数据。响应于注意到寄存器值的更改(即,由来自设备驱动器 120 的写入操作产生),仿真设备 140 或系统管理程序 100 执行任务以处理所述指令,而无需 VM 105 退出并显式地向系统管理程序通知寄存器值的更改(即,导致将 CPU 控制从 VM 切换到系统管理程序)。

[0021] 参考图 2 和 3A,在一个实施例中,VM 用户空间应用 107 可以发出系统调用。VM 操作系统 109 针对输入/输出(I/O)操作或其它发出的指令监视所述系统调用,并且判定是否已将指令从设备驱动器 120 发送到仿真设备 140(S310)。例如,VM 105 中的设备驱动器 120 可以使用程控输入/输出(PIO)或存储器映射 I/O(MMIO)方法,将指令发送到仿真设备 140。

[0022] PIO 和 MMIO 是在 CPU 和一个或多个物理设备 180(例如,网络适配器、存储设备等)之间传输数据并执行 I/O 的方法。例如,当 VM 用户空间应用 107 发出访问 I/O 地址空间的 CPU 指令以执行去往或来自物理设备 180 的数据传输时可发生 PIO。MMIO 可以使用地址接口(例如,总线)对存储器 160 和物理设备 180 进行寻址。用于访问存储器 160 的 CPU 指令也可以用于访问物理设备 180。

[0023] 在一个实施例中,可以针对 I/O 保留 CPU 的可寻址空间区域。所述保留例如可以是临时的或永久的。取决于实施方式,物理设备 180 可以监视 CPU 的地址总线,并响应 CPU 对设备分配的地址空间的访问,从而将数据总线连接到物理设备 180 的硬件寄存器。

[0024] 在图 2 中示出的虚拟化系统中,VM 操作系统 109 可以检测将 CPU 指令从设备驱动器 120 发出到仿真设备 140(S320)。如果发送,则 VM 操作系统 109 生成信号(例如,触发捕获)以便向仿真设备 140 通知所述指令的到达(S330)。

[0025] 要指出的是,由于 VM 用户空间应用 107 在 VM 105 的上下文中执行,因此 VM 105 负责处理相应的 CPU 指令。在一个实施例中,响应于上述被触发的捕获,系统管理程序 100 导致将 CPU 控制从 VM 105 切换到系统管理程序 100(S340)。因此,取决于实施方式,仿真设备 140 或系统管理程序 100 可以直接处理所述指令(S350)。

[0026] 参考图 2 和 3B, 在一种备选实施方式中, 仿真设备 140 或系统管理程序 100 针对仿真设备 140 的设备驱动器 120 发出的指令, 主动监视(例如, 轮询)仿真设备 140 的寄存器(即, 与系统管理程序共享的存储器 160 的区域)(S410)。如果向所述寄存器中写入(S420), 则仿真设备 140 或系统管理程序 100 处理所述指令(S430)。

[0027] 在不同的实施例中, 所要求保护的主体可以实现为硬件和软件元素两者的组合, 或者备选地完全以硬件形式或完全以软件形式实现。此外, 在此公开的计算机系统 and 程序软件可以包括受控计算环境, 其可以根据执行的硬件组件或逻辑代码提供, 这些硬件组件或逻辑代码用于执行获得在此构想的结果的方法和过程。当由通用计算机系统或机器执行时, 所述方法和过程将通用机器转换为专用机器。

[0028] 参考图 4A 和 4B, 根据一个示例性实施例的计算系统环境可以包括硬件环境 1110 和软件环境 1120。硬件环境 1110 可以包括逻辑单元、电路或其它机器和设备, 它们为软件环境 1120 的组件提供执行环境。反过来, 软件环境 1120 可以为硬件环境 1110 的各种组件提供执行指令, 包括底层操作设置和配置。

[0029] 参考图 4A, 在此公开的应用软件和逻辑代码可以以计算机可读代码的形式实现, 所述计算机可读代码在示例性硬件环境 1110 表示的一个或多个计算系统上执行。如图所示, 硬件环境 1110 可以包括处理器 1101, 其通过系统总线 1100 耦合到一个或多个存储元件。所述存储元件例如可以包括本地存储器 1102、存储介质 1106、高速缓冲存储器 1104 或其它计算机可用或计算机可读介质。在本公开的上下文中, 计算机可用或计算机可读存储介质可以包括任何可以用于包含、存储、发送、传播或传输程序代码的可记录制品。

[0030] 计算机可读存储介质可以是电、磁、光、电磁、红外线或半导体介质、系统、装置或设备。所述计算机可读存储介质还可以以传播介质实现(没有限制), 程度为此类实现被视为法定主题。计算机可读存储介质的实例可以包括半导体或固态存储器、磁带、可移动计算机盘、随机存取存储器(RAM)、只读存储器(ROM)、硬盘、光盘或载波(如果适用)。光盘的当前实例包括光盘 - 只读存储器(CD-ROM)、光盘 - 读 / 写(CD-R/W)、数字视频盘(DVD)、高清晰度视频盘(HD-DVD) 或 Blue-ray™ 盘。

[0031] 在一个实施例中, 处理器 1101 将可执行代码从存储介质 1106 加载到本地存储器 1102。高速缓冲存储器 1104 通过提供临时存储优化处理时间, 此类临时存储有助于减少加载代码以便执行的次数。一个或多个用户接口设备 1105 (例如, 键盘、指点设备等) 和显示屏 1107 例如可以直接或通过中间 I/O 控制器 1103 耦合到硬件环境 1110 中的其它元件。可以提供通信接口单元 1108 (例如网络适配器), 以使硬件环境 1110 能够通过中间专用或公共网络(例如, 因特网)与本地或远程计算系统、打印机和存储设备通信。有线或无线调制解调器和以太网卡是网络适配器的几种示例性类型。

[0032] 注意, 在某些实施方式中, 硬件环境 1110 可能不包括上面的部分或全部组件, 或者可能包括其它组件以提供补充功能或效用。取决于构想的使用和配置, 硬件环境 1110 可以是台式或膝上型计算机, 或者可选地包含在嵌入式系统中的其它计算设备, 例如机顶盒、个人数字助理(PDA)、个人媒体播放器、移动通信单元(例如, 无线电话), 或者具有信息处理或数据存储能力的其它类似的硬件平台。

[0033] 在某些实施例中, 通信接口 1108 用作数据通信端口, 以便通过发送和接收数字、电、电磁或光信号提供与一个或多个计算系统通信的手段, 这些信号承载表示各种类型信

息(包括程序代码)的模拟或数字数据流。所述通信可以通过本地或远程网络建立,或者备选地通过空气传输或其它介质传输(包括但不限于载波传播)建立。

[0034] 如在此所提供的,根据本质上是示例性的逻辑或功能关系定义在所示硬件元件上执行的所公开的软件元素。但是,应当指出,通过所述示例性软件元素实现的相应方法还可以例如通过配置和编程的处理器、专用集成电路(ASIC)、现场可编程门阵列(FPGA)和数字信号处理器(DSP)被编码在所述硬件元件中。

[0035] 参考图 4B,软件环境 1120 通常可以分为两类,包括在一个或多个硬件环境 1110 上执行的系统软件 1121 和应用软件 1122。在一个实施例中,在此公开的方法和过程可以实现为系统软件 1121、应用软件 1122 或它们的组合。系统软件 1121 可以包括诸如操作系统(OS)和信息管理系统之类的控制程序,它们指示硬件环境 1110 中的一个或多个处理器 1101 (例如,微处理器)如何运行和处理信息。应用软件 1122 可以包括但不限于程序代码、数据结构、固件、驻留软件、微代码,或者可以由处理器 1101 读取、分析或执行的任何其它形式的信息或例程。

[0036] 换言之,应用软件 1122 可以实现为程序代码,其以计算机可用或计算机可读存储介质的形式包含在计算机程序产品中,所述计算机程序产品提供程序代码以便由计算机或任何指令执行系统使用或者与其结合使用。此外,应用软件 1122 可以包括一个或多个计算机程序,这些计算机程序在从存储介质 1106 加载到本地存储器 1102 之后,在系统软件 1121 之上执行。在客户端-服务器体系结构中,应用软件 1122 可以包括客户端软件和服务器软件。例如,在一个实施例中,客户端软件可以在客户端计算系统上执行,该客户端计算系统不同于并独立于执行服务器软件的服务器计算系统。

[0037] 软件环境 1120 还可以包括浏览器软件 1126 以便访问通过本地或远程计算网络提供的数据。此外,软件环境 1120 可以包括用户界面 1124 (例如,图形用户界面(GUI))以便接收用户命令和数据。有必要重申,上面描述的硬件和软件体系结构和环境用于实例目的。因此,可以在任何类型的系统体系结构、功能或逻辑平台或处理环境中实现一个或多个实施例。

[0038] 还应理解,所述逻辑代码、程序、模块、过程、方法,以及每种方法的相应过程的执行顺序完全是示例性的。取决于实施方式,所述过程或任何底层子过程和方法可以以任何顺序执行或同时执行,除非本公开中另有所指。此外,除非明确地另有所指,否则本公开的上下文中的逻辑代码定义并不与任何特定程序设计语言相关或限于任何特定程序设计语言,并且可以包括一个或多个模块,它们可以在分布式、非分布式、单处理或多处理环境中的一个或多个处理器上执行。

[0039] 所属技术领域的技术人员知道,软件实施例可以包括固件、驻留软件、微代码等。包括软件或硬件或者组合软件和硬件方面的某些组件在此通常可以统称为“电路”、“模块”或“系统”。此外,所公开的主题可以实现为包含在一个或多个计算机可读存储介质中的计算机程序产品,这些介质在其中包含计算机可读程序代码。可以使用一个或多个计算机可读存储介质的任意组合。所述计算机可读存储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。计算机可读存储介质例如可以是一但不限于一电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者上述的任意合适的组合。

[0040] 在此文档的上下文中,计算机可读存储介质可以是任何能够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。计算机可读信号介质可以包括例如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程序代码。此类传播的信号可以采用多种形式,包括—但不限于—电磁信号、光信号或它们的任意合适的组合。计算机可读信号介质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。

[0041] 计算机可读存储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—无线、有线、光缆、RF 等等,或者上述的任意合适的组合。可以以一种或多种程序设计语言的任意组合来编写用于执行所公开的操作的计算机程序代码,所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++ 等,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。

[0042] 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)。

[0043] 将参考根据各实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图公开某些实施例。应当理解,流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合,都可以由计算机程序指令实现。这些计算机程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器,从而生产出一种机器,使得这些计算机程序指令在通过计算机或其它可编程数据处理装置的处理器执行时,产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。

[0044] 也可以把这些计算机程序指令存储在计算机可读存储介质中,这些指令使得计算机、其它可编程数据处理装置、或其它设备以特定方式工作,从而,存储在计算机可读存储介质中的指令就产生出包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的指令的制品(article of manufacture)。

[0045] 也可以把计算机程序指令加载到计算机、其它可编程数据处理装置、或其它设备上,使得在计算机、其它可编程装置或其它设备上执行一系列操作步骤,以产生计算机实现的过程,从而使得在计算机或其它可编程装置上执行的指令提供实现流程图和/或框图中的一个或多个方框中规定的功能/动作的过程。

[0046] 附图中的流程图和框图显示了根据多个实施例的系统、方法和计算机程序产品的可能实现的体系结构、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、组件段或代码的一部分,所述模块、组件段或代码的一部分包括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可以以不同于附图中所标注的顺序发生。

[0047] 例如,两个连续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,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动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。

[0048] 在此参考一个或多个特性或实施例提供了所要求保护的主体。所属技术领域的技术人员将认识到并理解, 尽管在此提供了示例性实施例的细节, 但是可以向所述实施例应用更改和修改而不会限制或偏离预期的范围。在此提供的实施例的这些和各种其它改变和组合, 均在权利要求及其全部等效物限定的所公开的主体的范围之内。

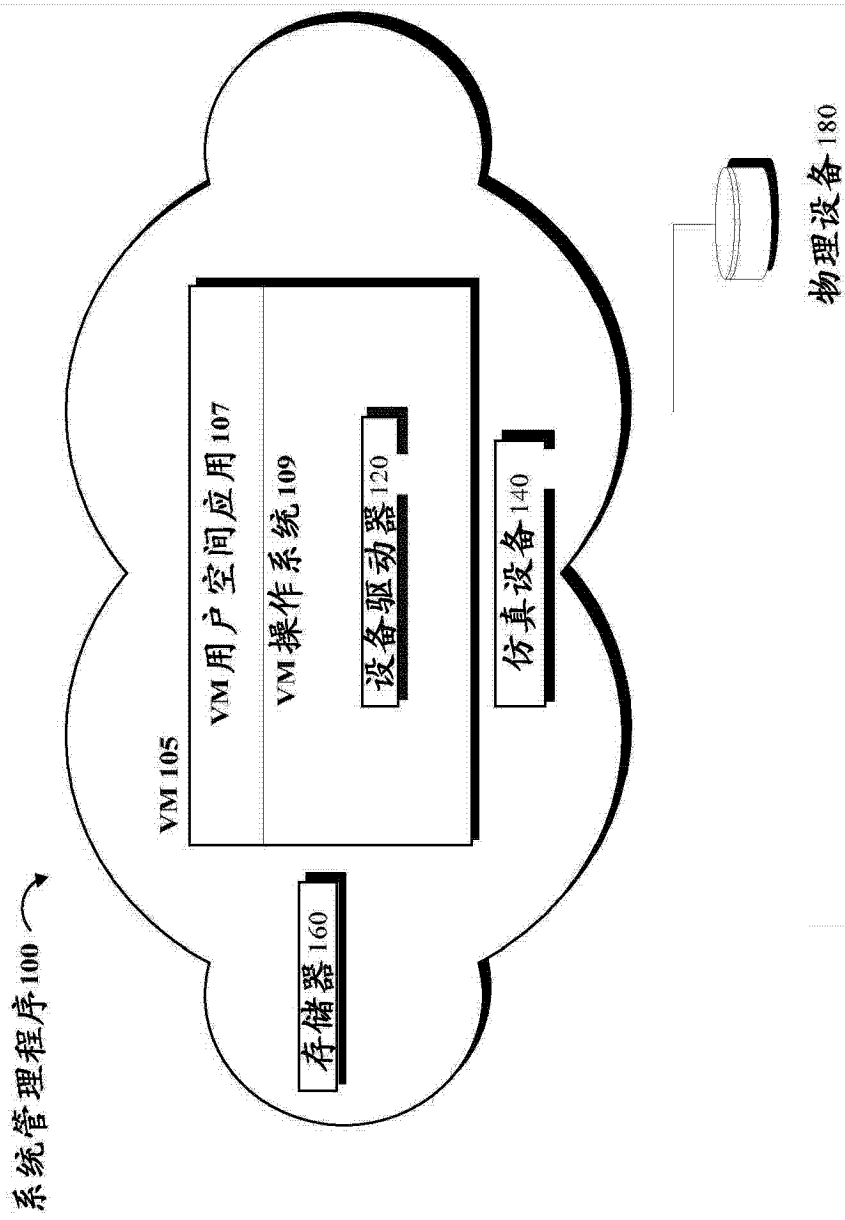


图 1

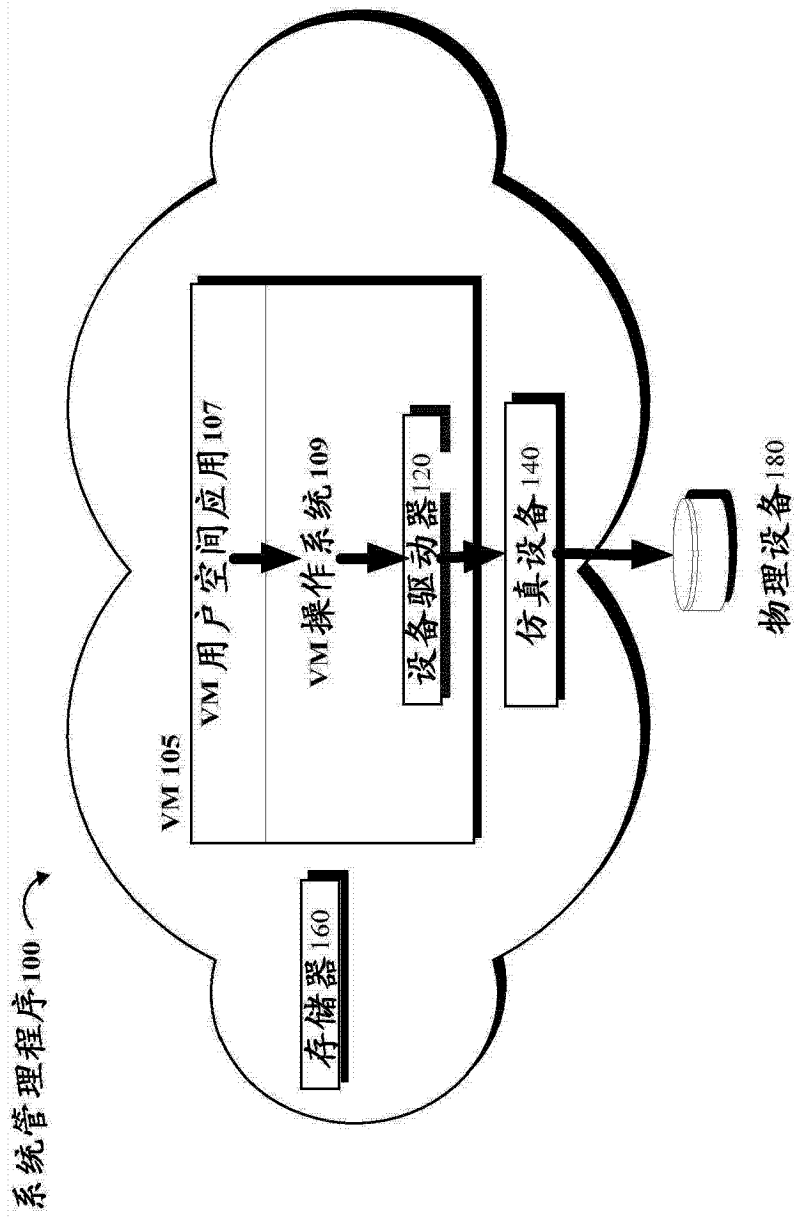


图 2

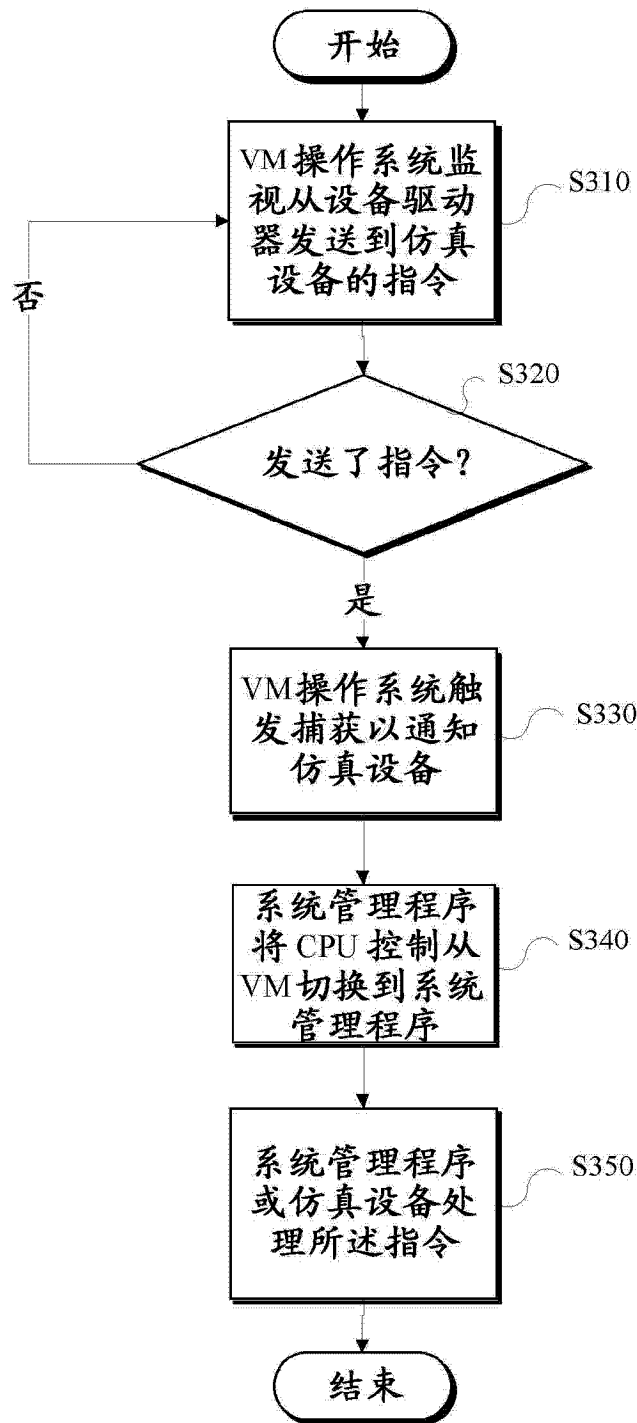


图 3A

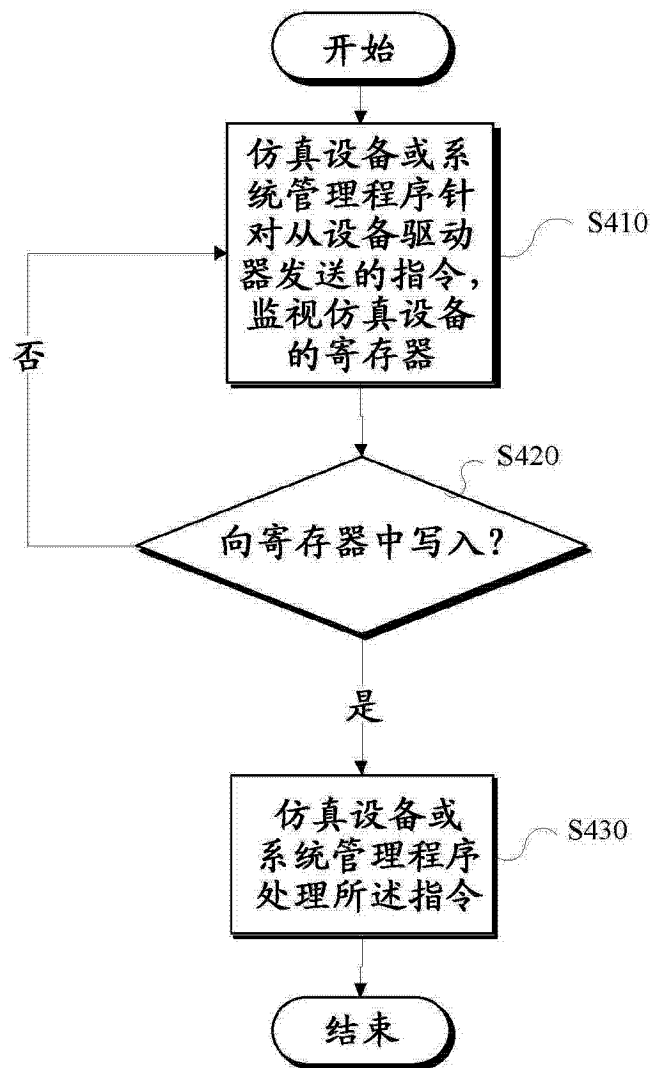


图 3B

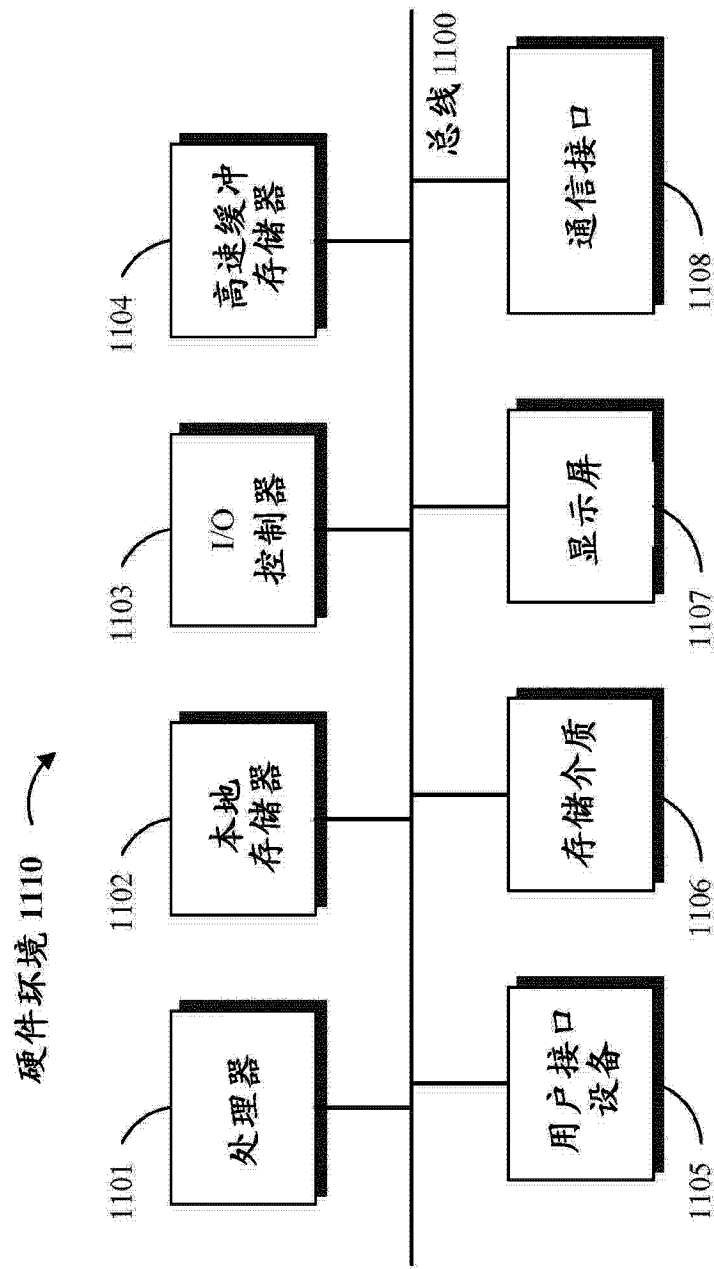


图 4A

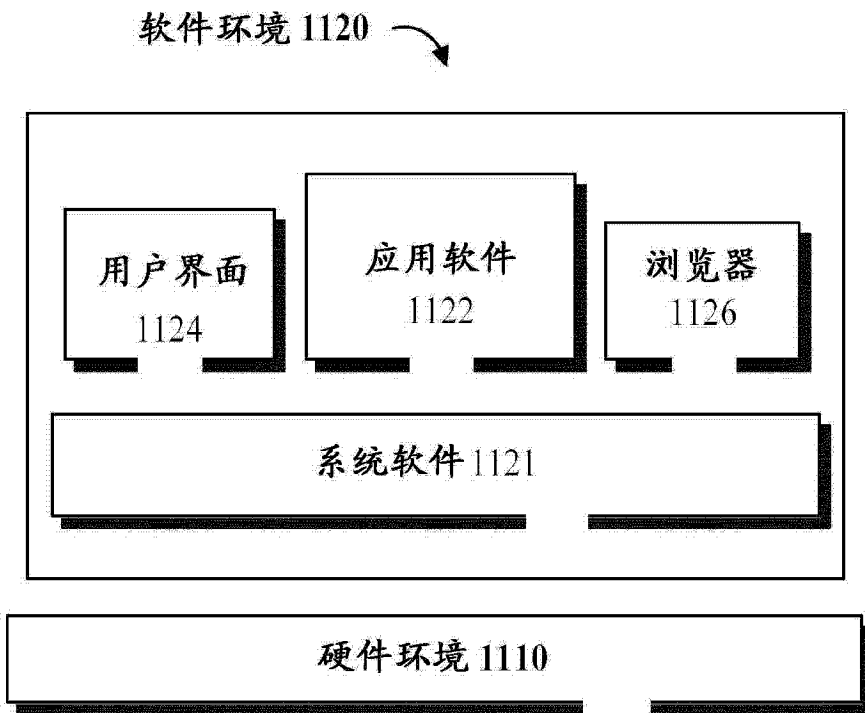


图 4B