(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 105511941 B
(45)授权公告日 2019.04.19

(21)申请号 201510870274.0

(22)申请日 2015.12.02

(65)同一申请的已公布的文献号
申请公布号 CN 105511941 A

(43)申请公布日 2016.04.20

(30)优先权数据
14/659,049 2015.03.16 US

(73)专利权人 卡巴斯基实验室股份公司
地址 俄罗斯莫斯科

(72)发明人 维亚切斯拉夫·I·列夫琴科
艾戈尔·Y·库马金

(74)专利代理机构 北京市磐华律师事务所 11336
代理人 董巍 谢栒

(51)Int.Cl.
G06F 9/455(2006.01)

(56)对比文件
TW 201120752 A,2011.06.16,
CN 101490645 A,2009.07.22,
CN 101523351 A,2009.09.02,
US 2011/0066787 A,2011.03.17,
US 2014/0181805 A1,2014.06.26,

审查员 张文全

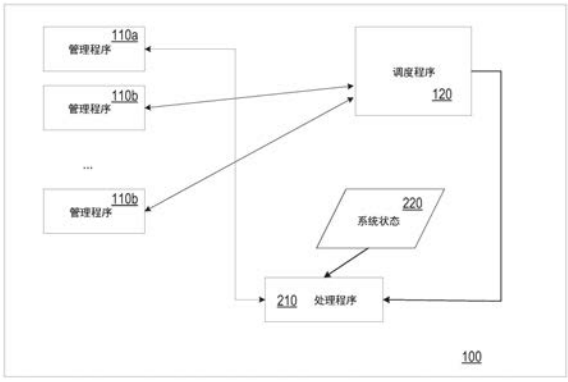
权利要求书3页 说明书9页 附图5页

(54)发明名称

辅助计算机系统中多个管理程序的联合操作的系统和方法

(57)摘要

公开了辅助计算机系统中多个管理程序的联合操作的系统和方法。公开了用于协调多个管理程序的联合操作的系统和方法。在计算机系统中协调多个管理程序的联合操作。执行持续管理程序和非持续管理程序。非持续管理程序根据由调度程序引擎控制的操作规程在监管员程序模式下执行，并且持续管理程序在处理程序引擎的控制下在管理程序模式下执行。处理程序引擎监控并响应处理器在管理程序模式和监管员程序模式之间的企图模式转换，并且视情况协调持续管理程序的挂起和恢复。



CN 105511941 B

1.一种用于协调多个管理程序的联合操作的系统,所述系统包括:

具有处理器、数据存储和输入/输出设备的计算平台,所述处理器可在管理程序模式和监管员程序模式之间切换,所述管理程序模式相比所述监管员程序模式提供更高的优先权级别,所述计算平台包含指令,当所述指令由所述计算平台执行时使得所述计算平台实现:

持续管理程序和非持续管理程序;

调度程序引擎,其配置为协调所述监管员程序模式下所述非持续管理程序的操作;

处理程序引擎,其配置为协调所述管理程序模式下所述持续管理程序的操作,使得:

所述处理程序引擎监控并响应所述处理器在所述管理程序模式和所述监管员程序模式之间的企图模式转换;

响应于从所述管理程序模式到所述监管员程序模式的企图模式转换,所述处理程序引擎挂起对所述持续管理程序的执行,包括保存所述处理器的状态,并且将所述处理器转换为在所述监管员程序模式下执行所述非持续管理程序;并且

其中响应于监管员程序模式指令的执行的终结,所述处理程序引擎挂起对所述非持续管理程序的执行,包括保存所述处理器状态,并且将所述处理器转换为在所述管理程序模式下执行所述持续管理程序,其中所述处理程序引擎包括拦截器引擎,其配置为响应于检测到企图进行处理器模式改变而挂起对这样的模式改变的执行。

2.根据权利要求1所述的系统,其中所述处理程序引擎使用操作系统驱动程序实现。

3.根据权利要求1所述的系统,其中所述处理程序引擎配置为监控所述计算平台的状态信息,所述状态信息指示在所述管理程序模式和所述监管员程序模式之间的处理器模式转换。

4.根据权利要求1所述的系统,其中所述处理程序引擎配置为监控与在所述管理程序模式和所述监管员程序模式之间的处理器模式转换相关联的至少一个命令。

5.根据权利要求4所述的系统,其中所述至少一个命令包括vmrun命令。

6.根据权利要求1所述的系统,其中所述处理程序引擎配置为检测与在所述管理程序模式和所述监管员程序模式之间的处理器模式转换相关联的至少一个参数状态改变。

7.根据权利要求6所述的系统,其中所述参数状态改变包括重新加载页目录。

8.根据权利要求6所述的系统,所述参数状态改变包括重新加载CR3注册表。

9.根据权利要求1所述的系统,其中所述处理程序引擎配置为使得,响应于从所述管理程序模式到所述监管员程序模式的企图模式转换,所述处理程序引擎确定从中发起模式转换的处理器模式并且基于该处理器模式的确定来确定允许还是不允许对所述持续管理程序的挂起。

10.根据权利要求1所述的系统,其中所述处理程序引擎配置为针对条件来监控所述调度程序引擎,所述条件指示对非持续管理程序的停止执行的调用,并且所述处理程序引擎响应于所述条件而恢复经挂起的持续管理程序的操作。

11.根据权利要求1所述的系统,其中所述拦截器引擎响应于检测到企图进行处理器模式改变而挂起对这样的模式改变的执行,直到完成用于控制所述持续管理程序的操作的一系列动作。

12.一种用于协调计算系统中多个管理程序的联合操作的方法,所述计算系统具有可在管理程序模式和监管员程序模式执行切换的处理器,所述管理程序模式相比所述监管员

程序模式提供更高的优先权级别，所述方法包括：

由所述计算系统执行持续管理程序和非持续管理程序，所述非持续管理程序根据由调度程序引擎控制的操作规程在所述监管员程序模式下执行；

由所述计算系统执行处理程序引擎，以协调所述管理程序模式下的所述持续管理程序的操作，使得：

所述处理程序引擎监控并响应所述处理器在所述管理程序模式和所述监管员程序模式之间的企图模式转换；

响应于从所述管理程序模式到所述监管员程序模式的企图模式转换，所述处理程序引擎挂起对所述持续管理程序的执行，包括保存所述处理器的状态，并且将所述处理器转换为在所述监管员程序模式下执行所述非持续管理程序；并且

其中响应于监管员程序模式指令的执行的终结，所述处理程序引擎挂起对所述非持续管理程序的执行，包括保存所述处理器状态，并且将所述处理器转换为在所述管理程序模式下执行所述持续管理程序，其中所述处理程序引擎响应于检测到企图进行处理器模式改变而挂起对这样的模式改变的执行。

13.根据权利要求12所述的方法，其中所述处理程序引擎监控所述计算平台的状态信息，所述状态信息指示在所述管理程序模式和所述监管员程序模式之间的处理器模式转换。

14.根据权利要求12所述的方法，其中所述处理程序引擎监控与在所述管理程序模式和所述监管员程序模式之间的处理器模式转换相关联的至少一个命令。

15.根据权利要求14所述的方法，其中所述至少一个命令包括vmrun命令。

16.根据权利要求12所述的方法，其中所述处理程序引擎检测与在所述管理程序模式和所述监管员程序模式之间的处理器模式转换相关联的至少一个参数状态改变。

17.根据权利要求16所述的方法，其中所述参数状态改变包括重新加载页目录或CR3注册表。

18.根据权利要求12所述的方法，其中所述处理程序引擎响应于从所述管理程序模式到所述监管员程序模式的企图模式转换，确定从中发起模式转换的处理器模式并且基于该处理器模式的确定来确定允许还是不允许对所述持续管理程序的挂起。

19.根据权利要求12所述的方法，其中所述处理程序引擎配置为针对条件来监控所述调度程序引擎，所述条件指示对非持续管理程序的停止执行的调用，并且所述处理程序引擎响应于该条件而恢复经挂起的持续管理程序的操作。

20.根据权利要求12所述的方法，其中所述处理程序引擎响应于检测到企图进行处理器模式改变而挂起对这样的模式改变的执行，直到完成用于控制所述持续管理程序的操作的一系列动作。

21.一种用于协调计算系统中多个管理程序的联合操作的系统，所述计算系统具有可在管理程序模式和监管员程序模式执行切换的处理器，所述管理程序模式相比所述监管员程序模式提供更高的优先权级别，所述系统包括：

用于执行持续管理程序和非持续管理程序的装置，所述非持续管理程序根据由调度程序引擎控制的操作规程在所述监管员程序模式下执行；

用于执行处理程序引擎以协调所述管理程序模式下所述持续管理程序的操作的装置，

使得：

所述处理程序引擎监控并响应所述处理器在所述管理程序模式和所述监管员程序模式之间的企图模式转换；

响应于从所述管理程序模式到所述监管员程序模式的企图模式转换，所述处理程序引擎挂起对所述持续管理程序的执行，包括保存所述处理器的状态，并且将所述处理器转换为在所述监管员程序模式下执行所述非持续管理程序；并且

其中响应于监管员程序模式指令的执行的终结，所述处理程序引擎挂起对所述非持续管理程序的执行，包括保存所述处理器状态，并且将所述处理器转换为在所述管理程序模式下执行所述持续管理程序，其中用于执行所述处理程序引擎的装置响应于检测到企图进行处理器模式改变而挂起对这样的模式改变的执行。

辅助计算机系统中多个管理程序的联合操作的系统和方法

技术领域

[0001] 本发明的一般地涉及信息处理，具体地，涉及计算虚拟化技术。更具体地涉及计算机系统中多个管理程序(hypervisor)的联合操作。

背景技术

[0002] 现今的计算机系统看到了日益增加的各种应用程序中的虚拟化技术的使用—无论在大数据中心还是在个人计算中。虚拟化支持如在单个计算机系统上运行多个操作系统、最大化利用可用硬件资源这样的配置。例如，个人计算机的用户可以利用多个虚拟机，用于在其单个机器上的不同的操作系统中同步执行各种应用程序。更多的时候，在大容量的服务器上发现虚拟化，诸如当提供主机托管服务时的那些服务器。

[0003] 虚拟机的操作一般需要使用管理程序，其协调虚拟机的执行，充当虚拟机的管理器或分派器。通常，管理程序还称为虚拟机监控程序(VMM)。

[0004] 如果计算机系统有多个管理程序，有必要恰当地协调它们的操作。一般由其设计提供虚拟机管理程序的联合操作：当管理程序控制了计算平台时，处理器的状态(即保护环)临时改变，并且在管理程序的动作完成之后，处理器返回到其初始状态。例如，在windows操作系统中，如果同步操作来自不同制造商的数个虚拟机，那么调度程序会以针对操作系统中的通常的线程同样的方式来分配时间用于管理程序的执行(例如考虑各个线程的优先级)。管理程序代码本身根据某一时间量的到期来正确完成其执行；然而应该注意到该时间量并不是必须和由OS的调度程序分配的时间量一致。

[0005] 在将要同时使用多个管理程序的系统中，常规的多任务技术可能不能有效工作，并且在最坏情况下它们甚至不会有效工作。当在多个管理程序中存在这样的管理程序：其以比另一个管理程序的优先级高的优先级执行时，那些问题进一步恶化。例如，在持续管理程序用于执行非常重要的虚拟机而伴随有一个或多个非持续管理程序的情况下，常规的多任务技术保证持续管理程序的正确持续操作存在困难。另一个问题在于，其他的常规管理程序可能没有意识到持续管理程序的存在，并且可能在管理程序模式下试图抢占的处理器上的执行时使操作系统崩溃。因此需要辅助多个管理程序的有效和高效的同步执行的解决方案。

发明内容

[0006] 本发明的一个方面涉及用于协调多个管理程序的联合操作的专用系统。系统是包括计算平台的专用机器，所述计算平台具有处理器、数据存储和输入/输出设备，所述处理器可在管理程序模式和监管员程序模式之间切换，管理程序模式提供比监管员程序模式更高的优先权级别。计算平台包含指令，当所述指令被执行时使得计算平台实现持续管理程序和非持续管理程序、配置为协调监管员程序模式下非持续管理程序的操作的调度程序引擎、配置为协调管理程序模式下持续管理程序的操作的处理程序引擎。

[0007] 处理程序引擎监控并响应处理器在管理程序模式和监管员程序模式之间的企图

模式转换。响应于从管理程序模式到监管员程序模式的企图模式转换，处理程序引擎挂起对持续管理程序的执行，包括保存处理器的状态，并且将处理器转换为在监管员程序模式下执行非持续管理程序。响应于监管员程序模式指令的执行的终结，处理程序引擎挂起对非持续管理程序的执行，包括保存处理器状态，并且将处理器转换为在管理程序模式下执行持续管理程序。

附图说明

[0008] 结合附图参考本发明的各种实施例的以下详细描述可更完整地理解本发明，其中：

[0009] 图1是示出了多个管理程序在其上操作的常规操作系统的一部分的示图，包括持续管理程序和一个或多个非持续管理程序。

[0010] 图2A是示出了根据一个实施例的、保证多个管理程序在系统中的执行的系统的示图。

[0011] 图2B是示出了根据一个实施例的、处理程序引擎的示例性体系架构的示图。

[0012] 图3是示出了根据一个实施例的、图2A-图2B中所示系统的操作过程。

[0013] 图4是示出了成为专用机器的计算机系统的示图，所述专用机器具有根据本发明的方面的实现的改善功能。

[0014] 虽然本发明可修正为各种修改和替代形式，但其细节已在图中通过示例的方式示出，并将进行详细描述。然而，应予以理解的是，不旨在将本发明限于所描述的特定实施例。相反，旨在覆盖落入由附加权利要求所定义的本发明的精神和范围内的所有修改、等同物以及替代物。

具体实施方式

[0015] I术语词汇

[0016] 以下术语词汇阐述了本文中使用的术语的定义。该术语词汇不仅用于本申请。

[0017] “计算平台”、“计算机”和“计算机系统”—可互操作的电子器件的电子设备或系统，包含硬件，所述硬件包括一个或多个处理器、数据存储器、输入-输出设备；以及可根据由硬件实施的软件指令存储和操纵的信息。它可以是一个物理机器，或者分布在多个物理机器中，例如通过角色或功能，或者在云计算分布式模型的情况下通过处理线程。示例包括台式机或移动个人计算机(PC)、智能手机和平板电脑，以及网络设备诸如路由器、交换机等。计算平台可以是独立的设备或是作为较大的设备或系统的一部分的嵌入式设备。

[0018] “数据存储”—在物理存储介质上存储数据的一个或多个电子硬件设备。示例包括易失性存储(例如随机存取存储器(RAM)，无论静态或动态)、非易失性存储(例如电子可擦除可编程只读存储器、磁盘等)。

[0019] “驱动程序”—引擎或组件，其就像诸如磁盘驱动器的设备与诸如操作系统壳(shell)的使用设备的程序之间的翻译器。驱动程序通常接受来自程序的通用命令并且随后将其翻译为用于设备的专用命令。

[0020] “引擎”—使用硬件、或者作为硬件或软件的组合实现的真实的设备、组件或组件的布置，例如通过微处理器系统和适配引擎以实现特定功能的程序指令集，其(被执行时)

将微处理器转换为专用设备。引擎也可以实现为两者的组合，具有某些由硬件单独实现的功能，以及由软件控制的硬件的组合辅助的其他功能。在某些实现方案中，至少部分引擎或者在一些情况下全部引擎可以在执行操作系统、系统程序和应用程序的一个或多个计算机的处理器上执行，同时也使用多任务、多线程、分布式(例如集群、点对点、云等)处理、合适的或其他这样的技术来实现引擎。此外，引擎本身可以包括一个以上的子引擎，其每一个可以看作独立的引擎。

[0021] “管理程序”—也称为虚拟机监控程序(VMM)，是可由处理器执行的程序，其协调虚拟机的执行并充当虚拟机的管理器或虚拟机的分派器。管理程序呈现采用虚拟操作平台的访客操作系统或系统程序，并且管理访客操作系统或系统程序的执行。程序系统或其他系统软件的多个实例可以共享虚拟化硬件资源。

[0022] “管理程序模式”—与甚至高于监管员模式的优先权级别相关联的处理器模式。管理程序模式一般提供专用指令，其辅助一个或多个管理程序防止每个均在监管员模式下执行的不同的操作系统或其他系统程序彼此影响的能力。一般，在系统中监管员程序模式指定为保护环0，管理程序模式指定为保护环-1

[0023] “输入/输出设备”或“输入/输出设施”—计算机系统的电子硬件部分，其辅助信息流入和流出计算机系统。示例包括网络接口设备、监视器、键盘、鼠标、打印机、串行端口等。

[0024] “持续管理程序”—在其操作的常规过程中被期望执行为操作系统或其他基本程序的活跃的不间断的进程的管理程序。该持续的操作规程不同于周期性的或非持续的规程，在周期性的或非持续的规程中管理程序偶尔用于特定任务，在任务结束后周期性管理程序可以关闭，而持续管理程序却保持运行。持续管理程序的示例包括要求不间断执行特别重要的虚拟机的管理程序，或者要求保证和安全相关的操作诸如反病毒进程的执行的管理程序。

[0025] “处理器”—计算机系统的电子硬件部分，其通过实施系统的基本的算术的、逻辑的、临时的存储和输入/输出操作来实施计算机程序的指令。典型地，处理器实现为微处理器(即，集成在单芯片上)，虽然该定义包括在多个互连的集成电路上实现的处理器电路。如今的处理器典型地包括多个处理核并且可以在多个处理核之间分布工作负载。

[0026] “处理器模式”—也称为“权限级别”，是用于一些计算机体系架构的处理器的操作规程，其选择性地对由处理器运行的某些进程所执行的操作的类型和方面设置一个或多个限制。例如，高信任度的内核代码以及在某些情况下高信任度的驱动程序或其他系统程序被允许在称为监管员程序模式的不受限制(或相对较少限制)的模式下执行；而其他进程(包括操作系统的非监管员部分)在称为用户模式的相对较多限制的模式下执行，并且必须使用系统调用以请求更受信任的内核代表其实施受限制的操作。处理器模式支持保护环体系架构。

[0027] “保护环”—在计算机系统的体系架构中的两个或两个以上的权限的分层级别或层次。这通常通过在硬件或微代码级别提供不同的处理器模式的一些架构体系硬件强制实施。例如，传统上已经在从最高权限(最受信任，通常编号为0)到最小权限(最不受信任，通常具有最高的环号)的层级中布置环。典型地，环0是操作系统的内核在其上执行的级别，并且较高编号的环与逐渐增加限制的权限级别相关联。某些现代的处理器支持甚至高于环0的权限级别，称为环-1。

[0028] “监管员程序模式”—与较高权限级别的进程诸如高信任度的操作系统内核进程和某些高信任度的驱动程序相关联的处理器模式。监管员程序模式允许敏感的机器代码操作的执行，诸如针对各种描述符表修改寄存器或者执行诸如禁用中断的操作。监管员程序模式还可以准许访问受限制的地址空间、存储器管理硬件和否则难以通过用户模式进程访问的某些外围设备。

[0029] “系统虚拟机”—提供完整的系统平台的虚拟机，所述完整的系统平台支持完整的操作系统的执行。这些通常仿真现有的体系架构，并且构建为提供虚拟机的多个实例，所述实例的每一个提供隔离的计算环境。这种类型的体系架构在支持云计算服务、网页主机托管和很多其他服务的服务器上是普遍存在的。系统虚拟机可以在称为主机机器的物理计算机系统的硬件上直接运行，或者在直接在主机机器上运行的操作系统之上运行(称为操作系统级虚拟化)。示例包括Windows Virtual PC、VMware、Oracle VM等。

[0030] “用户模式”—与诸如应用程序的较低优先权级别的进程相关联的处理器模式。用户模式禁止执行能够改变或破坏数据、访问输入/输出设施和外围设备、协调各种程序的执行的敏感操作。在用户模式下执行的进程必须请求较高优先权级别的进程代表其实施这些类型的动作。

[0031] “虚拟机”是计算机系统的基于软件的实现，其像使用主机机器的硬件的物理机器一样执行程序。基于它们的使用和与任何真实机器的对应程度虚拟机被分为两个主要的分类：系统虚拟机和进程虚拟机。

[0032] “用于执行持续管理程序和非持续管理程序的装置”—根据下述它们的各种实施例的任何一个(或者其组合)或它们的等同物的任何一个的、一个或多个处理单元404、系统存储器406和用于执行持续管理程序110a和非持续管理程序110b的代码。

[0033] “用于执行处理程序引擎的装置”—根据下述它们的各种实施例的任何一个(或者其组合)或它们的结构等同物的任何一个的、一个或多个处理单元404、系统存储器406和用于执行处理程序引擎210的代码。

[0034] Ⅱ.优选实施例描述

[0035] 发明的方面涉及如下计算方案：如果还存在其他管理程序，其中那些其他管理程序在同一个处理器核中周期性地操作，则辅助管理程序之一在计算机系统中持续操作。在本上下文中，管理程序的持续性意味着管理程序在其操作的常规过程中被期望执行为操作系统或其他基础程序的活跃的不间断的进程。该持续的操作规程不同于周期性的或非持续的规程，在该周期性的或非持续的规程中管理程序偶尔用于特定任务，在完成该任务之后周期性的管理程序可以被关闭而持续管理程序却保持运行。

[0036] 持续管理程序110a的示例可以是要求非常重要的虚拟机的不间断的执行的管理程序，或者是要求确保诸如反病毒进程的数个与安全相关的操作的执行的管理程序。非持续管理程序110b的示例可以是虚拟机分派器，也称为虚拟机监控器(VMM)诸如由加利福利亚州帕洛阿尔托的VMWare公司生产的那些、加利福利亚州红杉市Oracle公司的VirtualBox、以及佛罗里达州劳德代尔堡的Citrix Systems公司的Xen。

[0037] 持续管理程序110a相比任何非持续管理程序110b将通常以更高的优先权级别执行。例如，在支持具有甚至比操作系统更高的优先权级别的保护环的现代的处理器中，管理程序110a将会以较高的优先权级别(例如，保护环-1，即“管理程序模式”)执行，而非持续管

理程序110b将会以操作系统的优先权级别(例如,保护环0,即“监管员程序模式”)执行。

[0038] 根据本发明的方面的计算系统包括各种引擎,其每一个被构造、编程、配置或另外适配为自主地实施功能或功能集。文本所使用的术语引擎指真实的设备、组件,或者使用硬件或作为硬件和软件的结合所实现的组件布置,所述硬件诸如通过专用集成电路(ASIC)或现场可编程门阵列(FPGA),所述硬件和软件的结合诸如通过微处理器系统和一组调整模块实现特定功能的程序指令,这些指令(在执行时)将该微处理器系统转化为特殊目的装置。引擎也可以实现为两者的结合,由硬件单独帮助实现某些功能,以及由硬件和软件的结合来帮助实现其他功能。在某些实现方案中,可以在执行操作系统、系统程序和应用程序的一个或多个计算机的处理器上执行引擎的至少一部分以及在一些情况下执行引擎的全部,同时也实现引擎(并且从而成为专用机器)。每个引擎均可以在各种适合的物理和逻辑配置中实现,并且,一般不应限定于任何本文例示的特定实现方案,除非这种限定被明确要求。此外,引擎自身可以包括一个以上的子引擎,其每一个可以看作单独的引擎。另外,在本文所述的实施例中,各种引擎的每一个对应于限定的功能;然而,应该理解在其他预期的实施例中,每个功能可以分布到一个以上的引擎。同样,相比在本文的示例中所具体示出的,在其他预期的实施例中,多个限定的功能可以由单个引擎实现,其实施可能与其他函数一起的或者不同地分布在引擎组中的那些多个函数。

[0039] 图1是示出了常规操作系统(OS)10的一部分的示图,多个管理程序在所述操作系统10上操作—例如,持续运行的管理程序110a,以及其他管理程序110b(可以存在一个以上)。在管理程序110a和110b的常规操作的情况下,常规任务调度程序120将分配每个管理程序用于执行的时间。例如,在Windows OS中,调度程序120通过优先权管理多任务控制,其意味着较高优先权线程的第一优先权执行。结果,在使用调度程序120的OS 10中,即使较高优先权被指派给管理程序110a,也不可能保证管理程序110a的持续操作,因为调度程序120迟早将分配时间量用于管理程序110b的代码的执行。在本上下文中,管理程序110b中的一个的执行基本上是指在那些管理程序下操作的虚拟机的执行。

[0040] 在各种实施例中,管理程序110a、110b以及任务调度程序120中的每一个均被实现为包括计算平台的相关部分的引擎,即与所有必要的固件或软件组件相结合的硬件,例如计算机系统的基本输入/输出系统、操作系统、设备驱动程序、函数库和存储在计算平台的存储介质中并且能够在计算机系统的处理器上执行的其他程序指令的相关部分。

[0041] 应该注意由于数个原因持续管理程序110a在反病毒操作的执行期间可能成为必须的。首先,管理程序可以允许检测OS核(监管员程序模式,环0)级别上恶意代码的执行。其次,管理程序可能需要旁路OS核保护,诸如PatchGuard。在多内核处理器中,管理程序的分开的拷贝将在每个核中操作,但是具有不同的对应上下文。

[0042] 图2A示出了根据一个实施例的,用于保证系统中多个管理程序的执行的系统。如图示,操作系统100宿主持续管理程序110a,但是在该实施例中,持续管理程序110a不由调度程序120控制(不像管理程序110b),而是由处理程序210控制,其跟踪系统的状态信息220。处理程序210在各个实施例中物理地实现为包括计算硬件和对应软件组件的引擎。在相关实施例中,处理程序210进一步使用在计算硬件上执行的专门的OS驱动程序来实现。

[0043] 通常,当处理器在管理程序模式(保护环-1)下任何管理程序110b试图开始执行其代码时,处理器的操作模式首先发生改变,因为在该时间期间负责准备虚拟机的后续启动

的命令将被执行。例如，页目录和CR3注册表被整体重新加载。虚拟机110b的启动的另一个示例是vmrun命令(在VMWare的情况下)的执行。任何的这些参数和命令在图2A中全体表现为系统的状态信息220。

[0044] 处理程序210跟踪对规定参数的任何修改的进行，或者命令的执行(通过拦截它们)以检测执行对应的管理程序110b的需要。相应地，处理程序210将停止持续管理程序110a的执行。

[0045] 图2B是示出了根据一个实施例的处理程序引擎210的示例性体系架构的框图。命令检测器引擎250和参数改变监控器引擎252被各自编程或者另外配置为监控系统的状态信息220。命令检测器引擎检测命令，而参数改变监控器引擎252检测操作系统中某些参数的改变。管理程序执行检查器256将检测到的命令和参数中的改变与指示对管理程序110b的执行的调用的标准或者恢复持续管理程序110a的操作的适合性进行比较。这样的标准包括在处理器的状态中的改变、页目录或CR3注册表的重新加载、或者vmrun命令的发生。在相关实施例中，拦截器引擎258被编程或者另外配置为响应于检测到指示企图模式转换的命令或参数改变的发生，阻止该命令或模式转换的执行直到采取特定的系列动作。持续管理程序执行控制器引擎260被编程或另外配置为采取如下详细描述的动作。

[0046] 下面是根据一个实施例的、由持续管理程序执行控制器引擎260结合组成处理程序210的其他引擎激活的和去激活持续管理程序110a的进程的过程的更详细的讨论。该过程包括以下动作：

[0047] a.保留处理器的先前状态(即保护环-1的管理程序模式中的管理程序110b之一的代码的执行的最后时刻)；

[0048] b.在不同的模式例如监管员程序模式(保护环0)中执行持续管理程序110a的代码；

[0049] c.确定企图执行任何管理程序110b的代码；

[0050] d.恢复处理器的先前状态用于由管理程序110b正确处理用于虚拟机的激活的指令；

[0051] e.在管理程序110b的代码的执行之前，从管理程序模式退出并且在监管员程序模式下转移对的管理程序110a的最后的指令的控制。该操作确保持续管理程序110a被退出管理程序模式并且进入监管员程序模式，从而当管理程序110b在管理程序模式下执行时管理程序110b意识不到持续管理程序110a的存在。

[0052] 因此，因为继续执行已经在不同的保护环的权限内实施(通常，它会是对应于内核的优先权级别的环0)，所以持续管理程序110a的线程的下一个指令已经在管理程序模式之外执行，。

[0053] 图3示出了根据一个实施例的操作的过程。在310，处理程序210(使用命令检测器250和参数改变监控器252)跟踪系统的状态信息220。如果在320管理程序执行检测器引擎256确定系统的状态已经改变(例如页目录被覆写或者vmrun命令已经运行)，那么在330持续管理程序110a的执行由持续管理程序执行控制器260停止。

[0054] 在相关的实施例中，在320由管理程序执行检测器引擎256实施附加的检查，即确定CR3注册表的重新加载被从中调用的处理器模式。如果注册表重新加载被从内核模式调用，则被认为仅是上下文的切换，并且如果是从用户模式调用，则有可能另外检查这是例如

来自虚拟机的进程的调用。这样的检测允许系统最小化在停止持续管理程序110a的情况下“错误的激活”的数目，以便避免无绝对的必要而停止它。

[0055] 在340，管理程序执行监测器引擎256针对恢复持续管理程序110a的适合性进行检查。该动作可以响应于以下一个或多个条件：

[0056] • 由调度程序120分配用于管理程序110b之一的执行的执行时间量到期；

[0057] • 系统220的状态的改变已经发生；

[0058] • 处理程序210跟踪管理程序110b的执行并且接收关于其操作结束的通知(例如，通过由管理程序的开发者提供的正式的应用程序可编程接口或者API)。

[0059] 如果在350确定了管理程序110b之一已经完成执行(通过检查条件诸如以上标识的那些)，那么在360持续管理程序110a由持续管理程序执行控制器260恢复。在相关实施例中，存在多个不同的管理程序110b的情况下，在紧接着第一管理程序110b的执行完成之后、持续管理程序110a恢复之前，持续管理程序110a在恢复执行前等待所有管理程序110b完成它们的执行。

[0060] 在一个实施例中，持续管理程序110a的操作的恢复使用定时器过程来实施(例如，在Windows OS中使用KeSetTimer调用来进行)。例如，一旦指定的时间到期(当执行时间量已经到期)，那么调度程序120确定哪个代码必须被执行(例如，持续管理程序110a的开始)。定时器可以被重新设置以便在下一次管理程序110b完成其操作并且持续管理程序110a可以启动时允许后续的启动。

[0061] 持续管理程序110a何时应该停止的另一个示例与系统的状态信息220中的电源相关的改变有关，例如，当将要进入“休眠”模式时。为此，跟踪与电源相关的函数(Windows OS中的电源管理事件回调函数)或系统变量(Windows OS中的SYSTEM_POWER_STATE)。

[0062] 需要停止管理程序110a的另一个示例是也导致处理器的状态的显著改变的一段代码的启动，例如，仿真器(诸如QEMU(快速仿真器)和其他类似的程序，包括在反病毒应用程序中使用的那些)。但是如果仿真器和持续管理程序110a包括在反病毒产品中，那么在仿真器的操作之后，后者可以自动启动持续管理程序110a。

[0063] 图4为更详细地示出了计算机系统400的图示，其被制成具有根据本文所描述的发明的方面的实现的改善功能的专用机器。计算机系统400可以包括诸如个人计算机402的计算设备。个人计算机402包括一个或多个处理单元404、系统存储器406、视频接口408、输出外围设备接口410、网络接口412、用户输入接口414、可移除存储器接口416和非可移除存储器接口418以及耦合到各个组件的系统总线或高速通信通道420。在各个实施例中，处理单元404可以具有多个逻辑核，其能够处理存储在计算机可读介质诸如系统存储器406或附接到可移除存储器接口416和非可移除存储器接口418的存储器上的信息。计算机402系统存储器406可以包括诸如只读存储器(ROM)的非易失性存储器422或者诸如随机存取存储器(RAM)的易失性存储器424。ROM 422可以包括级别输入/输出系统(BIOS)426以帮助与计算机402的其他部分通信。RAM 424可以存储各种软件应用程序部分诸如操作系统428、应用程序430和其他程序模块432。此外，RAM 424可以存储诸如程序或应用程序数据434的其他信息。在各个实施例中，RAM 424存储要求低延时和高效率访问的信息诸如程序和对其进行操纵或操作的数据。在各个实施例中，RAM 424包括双倍数据速率(DDR)存储器、错误纠正存储器(ECC)或具有改变的延迟和配置的其他存储器技术诸如RAMBUS或DDR2和DDR3。这样，在各

个实施例中，系统存储器406可以存储输入数据存储、访问证书数据存储、操作存储器数据存储、指令集数据存储、分析结果数据存储和操作存储器数据存储。此外，在各个实施例中，处理单元404可以配置为通过要求在对信息的访问被授予之前访问证书来执行限制对上述数据存储的访问的指令。

[0064] 可移除存储器接口416和非可移除存储器接口418可以将计算机402耦合到硬盘驱动器436诸如SSD或旋转硬盘驱动器。这些硬盘驱动器436可以提供针对各种软件应用程序的进一步存储，诸如操作系统438、应用程序440和其他程序引擎442。此外，硬盘驱动器436可以存储诸如程序或应用程序数据444的其他信息。在各个实施例中，硬盘驱动器436存储不要求如在其他存储介质中的同样的低延迟的信息。此外，操作系统438、应用程序440数据、程序引擎442和程序或应用程序数据444可以是如在上述的各个实施例中的RAM 424中存储的同样的信息或者它可以是RAM 424存储的数据的不同的数据可能衍生。

[0065] 此外，可移除非易失性存储器接口416可以耦合计算机402到利用磁介质诸如软盘448，Iomega®Zip or Jazz的磁便携硬盘驱动器446，或耦合到利用诸如Blu-Ray®，DVD-R/RW，CD-R/RW以及其他类似格式的用于计算机可读介质的存储的光盘驱动器450。另外其他实施例利用SSD或封装在便携外壳54中的旋转硬盘以增加可以移除存储器的容量。

[0066] 计算机402可以利用网络接口412通过本地局域网(LAN)458或广域网(WAN)460和一个或多个远程计算机456通信。网络接口412可以利用网络接口卡(NIC)或其他接口诸如调制解调器462以使能通信。调制解调器462可以使能通过电话线、同轴线、光纤、电力线或无线的通信。远程计算机456可以包含类似的硬件和软件配置或者可以具有存储器464，其包含可以提供附加的计算机可读指令给计算机402的远程应用程序466。在各个实施例中，远程计算机存储器464可以用于存储信息诸如识别的文件信息，其可以随后下载到本地系统存储器406。此外，在各个实施例中，远程计算机456可以是应用程序服务器、管理服务器、客户端计算机或网络装置。

[0067] 用户可以使用连接到用户输入接口414输入设备诸如鼠标468和键盘470来输入信息到计算机402。此外，输入设备可以是触控板、指纹扫描仪、操纵杆、条形码扫描仪、媒体扫描仪等。视频接口408可以提供可视信息到诸如监控器的显示器472。视频接口408可以是嵌入式接口或者可以是分离式接口。而且，为了增加计算机402操作中的灵活性，计算机可以利用多个视频接口408、网络接口412以及可移除接口416和非可移除接口418。而且，各个实施例利用若干监视器472和若干视频接口408改变计算机402的性能和容量。计算机402中可以包括其他计算机接口，例如输出外围接口410。该接口可以耦合至打印机474或扬声器476或其他给计算机402提供附加功能的外围接口。

[0068] 计算机402的不同的可替代的配置和实现方式都落入本发明的精神。这些改变可以包括但不限于，耦合至系统总线420的附加接口，例如通用串行总线(USB)、打印机端口、游戏端口、PCI总线、PCI Express或上面描述的多个元件结合成的芯片集元件，例如北桥和南桥。例如，在各个实施例中，处理单元404可以包括嵌入式存储器控制器(未示出)，使得能够相比系统总线420可以提供的，从系统存储器406进行更有效的数据转移。

[0069] 上面的实施例意图说明并且不意图限制。附加的实施例落入权利要求的保护范围。此外，虽然本发明的各方面已经参考具体实施例进行了描述，但本领域技术人员将意识到，可以进行形式以及细节上的改变而不脱离由权利要求限定的本发明的保护范围。

[0070] 相关领域的一般技术人员将意识到，本发明可以包括比上面描述的任何个别实施例中描述的更少的特征。此处描述的实施例并不意味着可能组合了发明的多个特征的详尽介绍方式。因此，各实施例并不是相互排斥的特征组合；而是，本发明可以包括选自不同个别实施例的不同个别特征的组合，如本领域普通技术人员将理解的。

[0071] 限制引用上面的文档的任何合并，以便没有与此处明确公开的相反的主题名称并合并。引用上面文档的任何合并进一步被限制，以便包括在文档中的权利要求不会通过引用到本申请的权利要求中被合并。然而，任何文件的权利要求作为此公开的一部分被合并，除非特别排除。引用上面文件的任何合并仍然进一步被限制，以便文件中提供的任何定义不会在此处被引用合并，除非此处明确包括。

[0072] 为了理解本发明的权利要求的目的，明确意图不借助35U.S.C.第6段第112部分的条款，除非权利要求中叙述了特定术语“装置是”或“步骤为”。

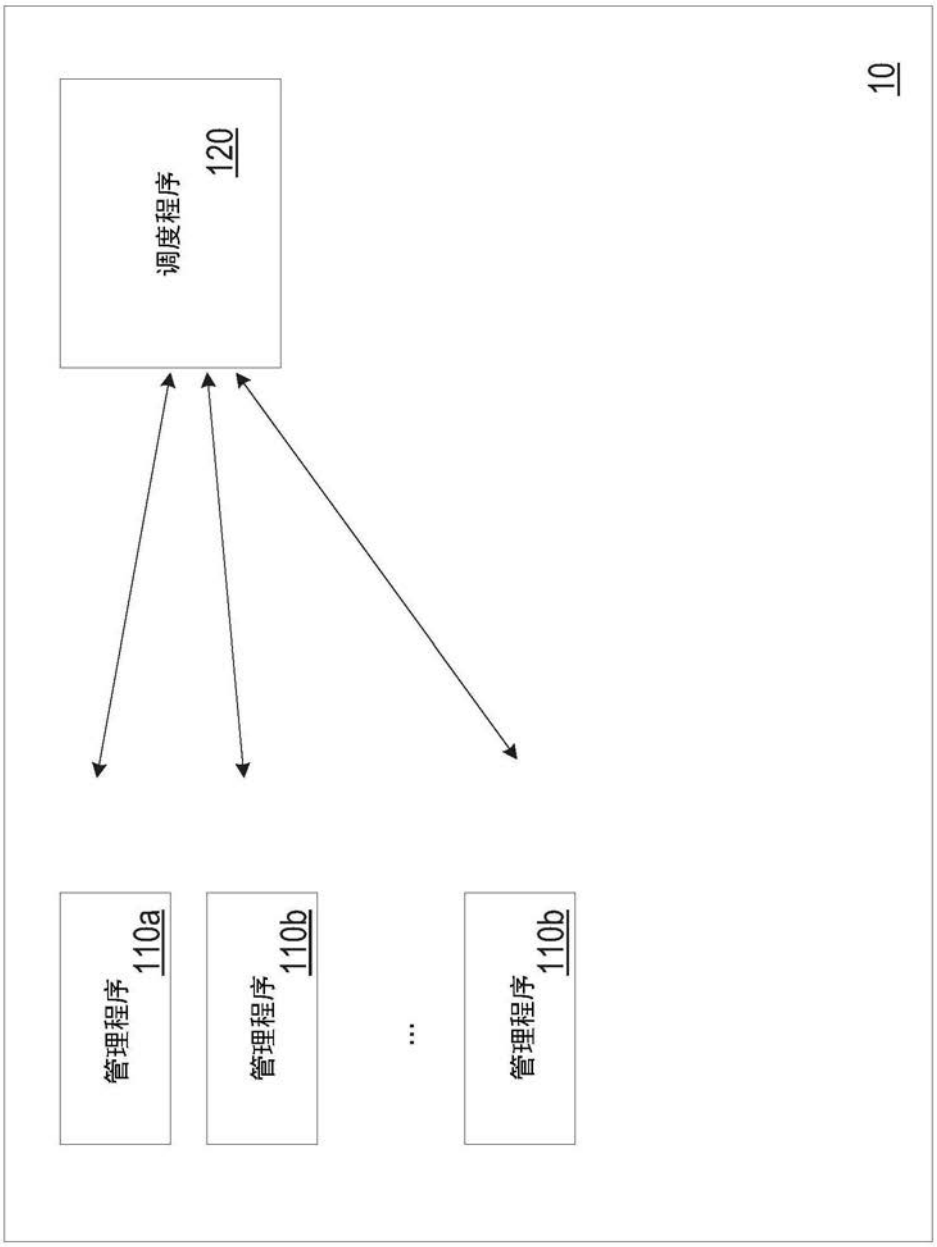


图1(现有技术)

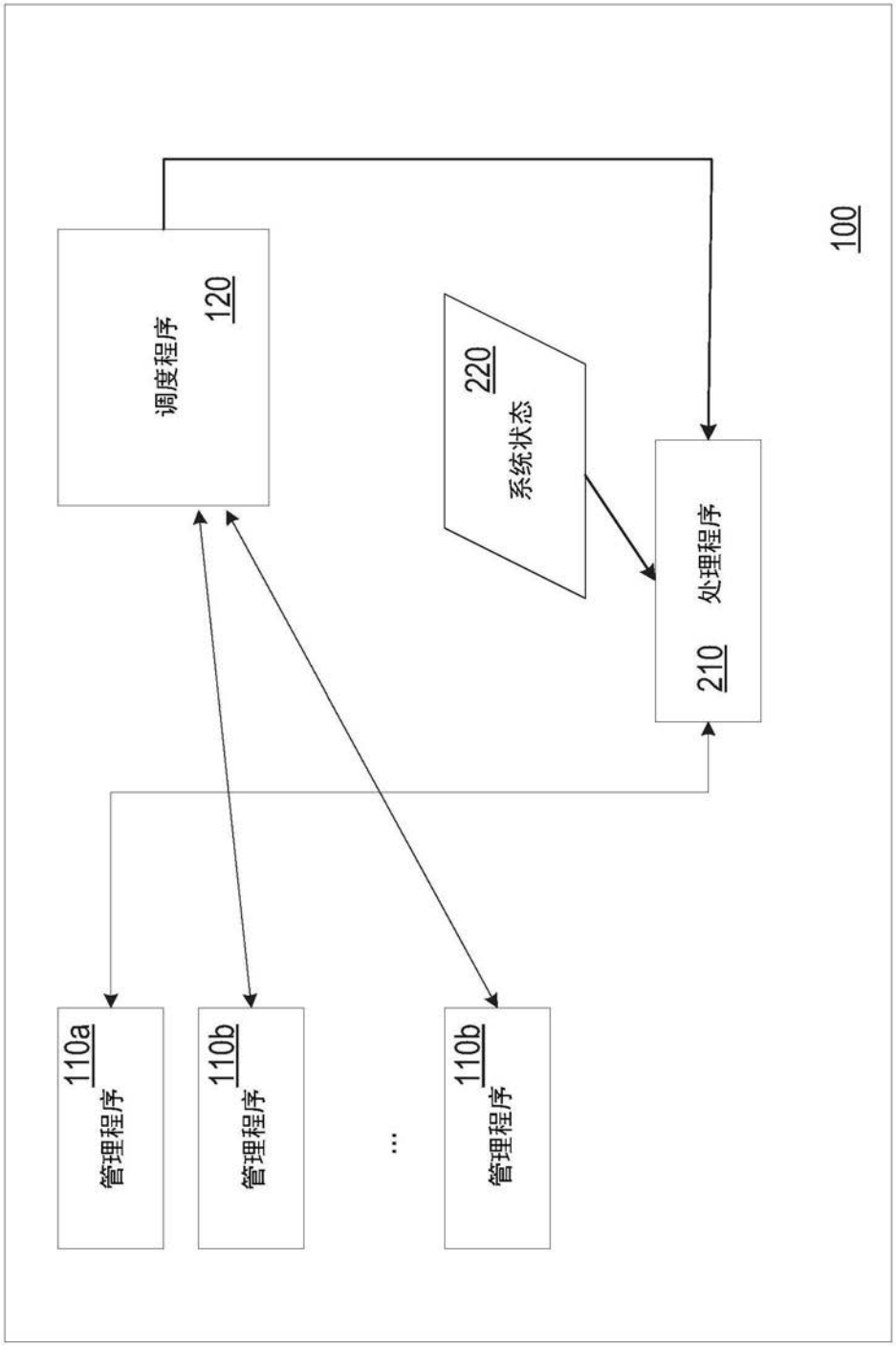


图2A

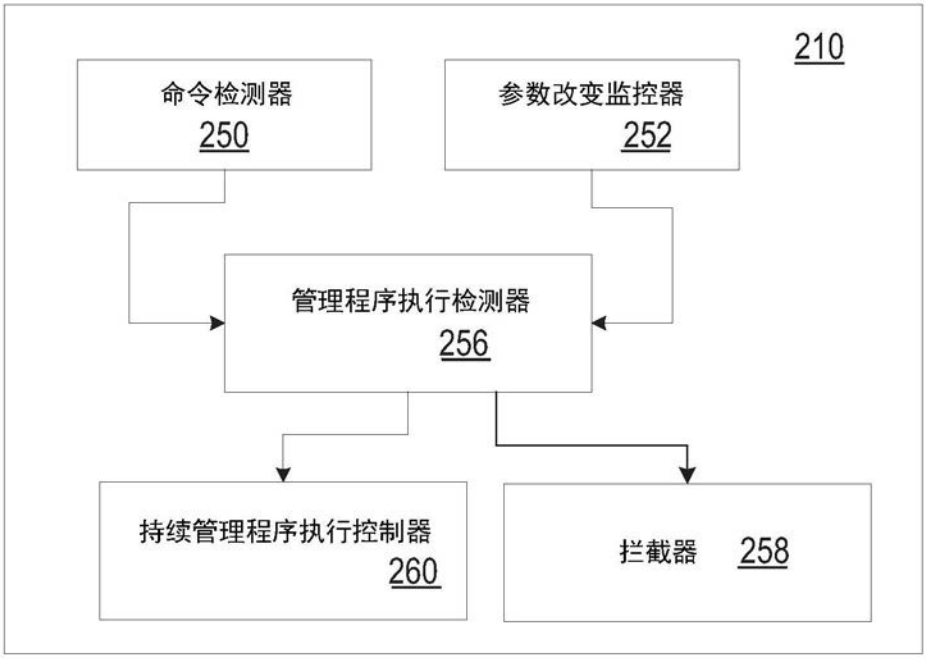

210
命令检测器
250
参数改变监控器
252
管理程序执行检测器
256
持续管理程序执行控制器
260
拦截器 258

图2B

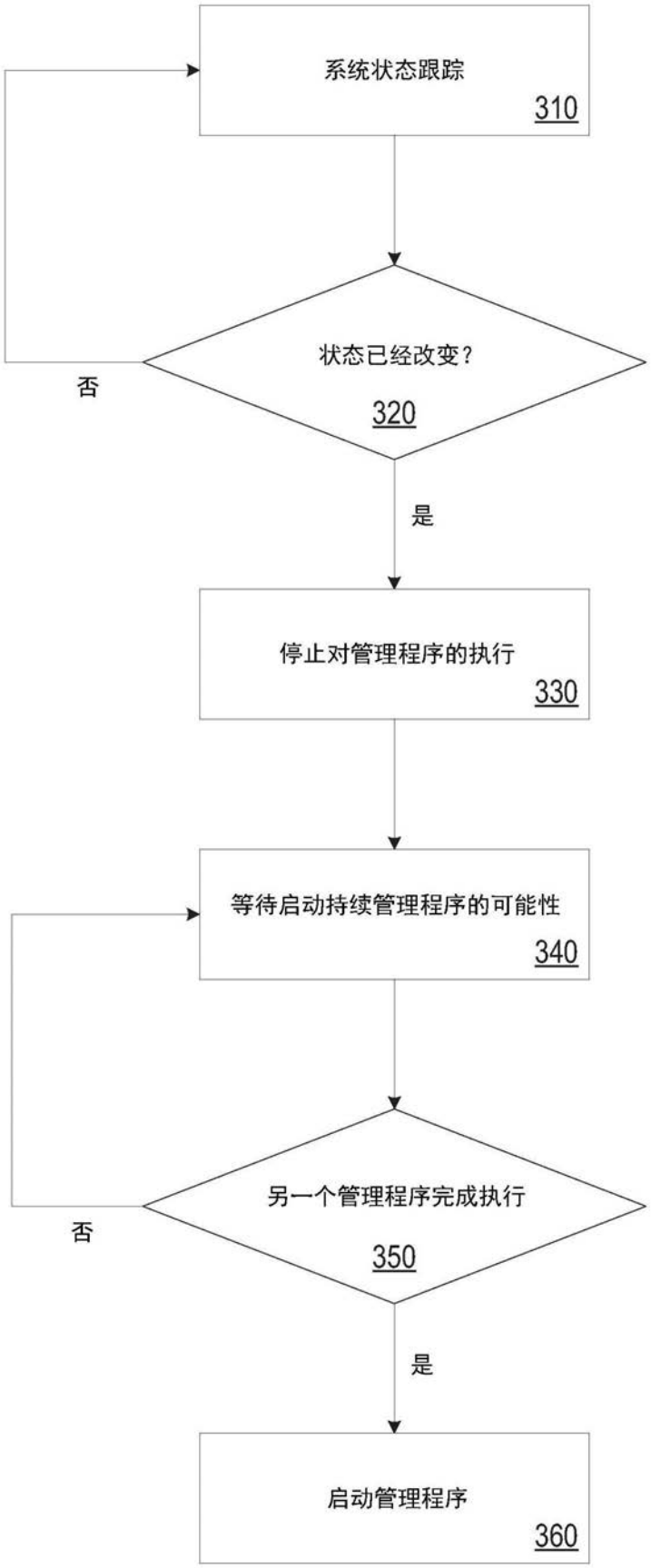

系统状态跟踪
310
状态已经改变？
320
否
是
停止对管理程序的执行
330
等待启动持续管理程序的可能性
340
另一个管理程序完成执行
350
否
是
启动管理程序
360

图3

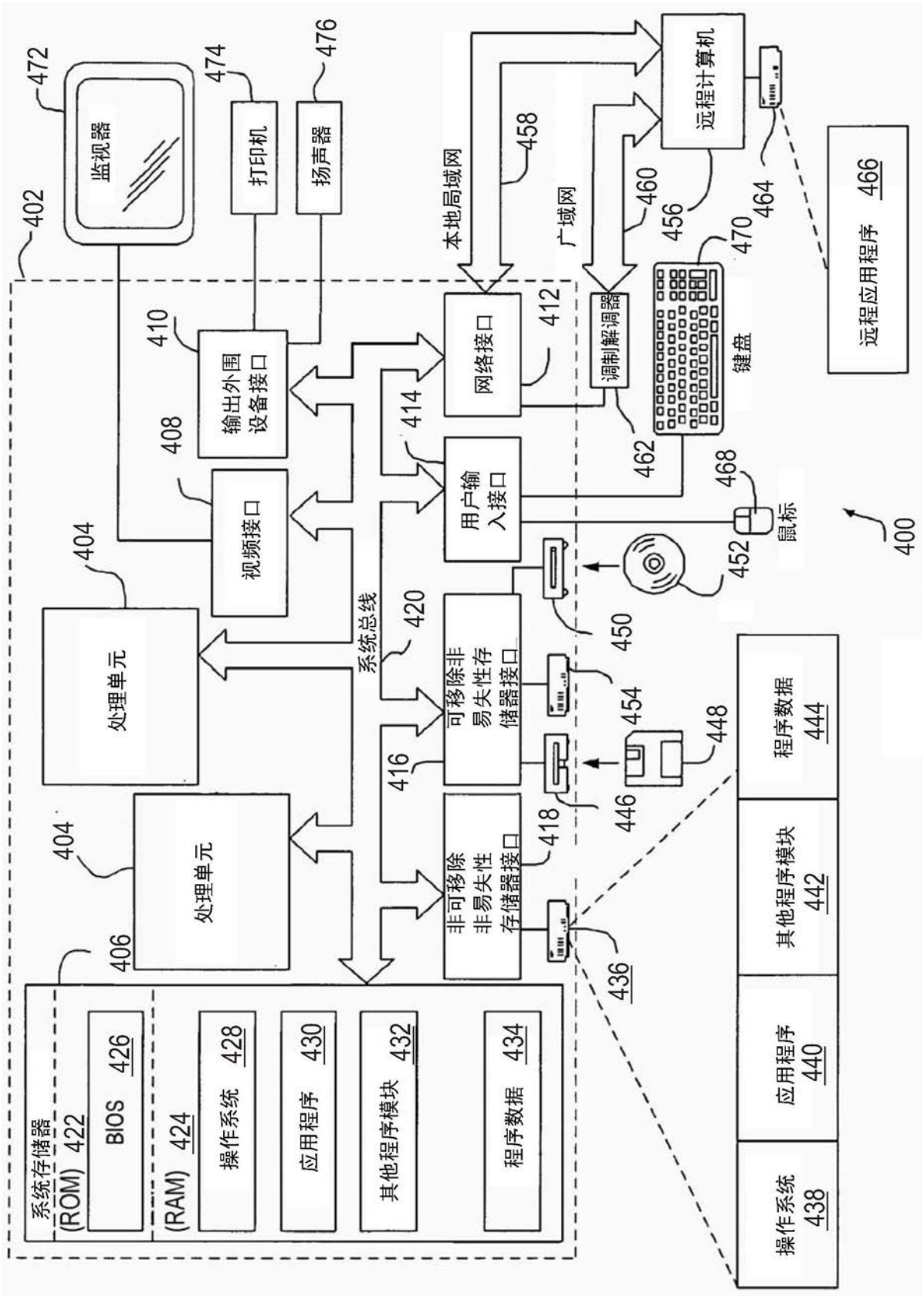

监视器
472
打印机
474
扬声器
476
402
本地局域网
458
广域网
460
远程计算机
456
464
远程应用程序 466
470
键盘
462
调制解调器
输出外围设备接口
410
网络接口
412
408
414
视频接口
用户输入接口
468
鼠标
452
400
404
处理单元
系统总线
420
450
可移除非易失性存储器接口
454
416
448
处理单元
404
406
非可移除非易失性存储器接口
418
446
436
程序数据 444
其他程序模块 442
应用程序 440
操作系统 438
系统存储器 (ROM) 422
BIOS 426
(RAM) 424
操作系统 428
应用程序 430
其他程序模块 432
程序数据 434

图4