(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113544678 B
(45)授权公告日 2025.06.20

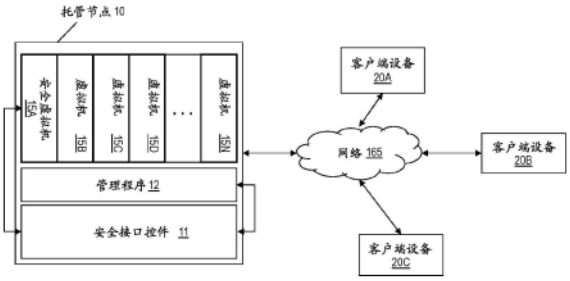
(21)申请号 202080019330.9
(22)申请日 2020.02.27
(65)同一申请的已公布的文献号
申请公布号 CN 113544678 A
(43)申请公布日 2021.10.22
(30)优先权数据
16/296,316 2019.03.08 US
(85)PCT国际申请进入国家阶段日
2021.09.07
(86)PCT国际申请的申请数据
PCT/IB2020/051667 2020.02.27
(87)PCT国际申请的公布数据
WO2020/183278 EN 2020.09.17
(73)专利权人 国际商业机器公司
地址 美国纽约
(72)发明人 F·布萨巴 L·海勒
J·布拉德伯里 C·博恩特雷格
C·英布伦达
(74)专利代理机构 北京市中咨律师事务所 11247
专利代理师 李永敏 于静
(51)Int.Cl.
G06F 21/57(2006.01)
G06F 21/53(2006.01)
G06F 21/56(2006.01)
G06F 9/455(2006.01)
(56)对比文件
US 2017177392 A1,2017.06.22
US 2017177398 A1,2017.06.22
审查员 董立波
权利要求书4页 说明书16页 附图5页

(54)发明名称

安全虚拟机环境中的客户机指令的透明解释

(57)摘要

公开了一种计算机实现的方法。该方法包括由在主机服务器上执行的虚拟机执行指令流,其中来自指令流的指令将要被拦截到管理程序。该方法进一步包括基于确定虚拟机是安全虚拟机,阻止管理程序直接访问安全虚拟机的任何数据。该方法还包括由主机服务器的安全接口控件基于确定指令不能被安全接口控件本身解释来执行:从安全虚拟机提取与指令相关联的一个或多个参数数据,以及将参数数据存储到管理程序可访问的缓冲器中。指令随后被拦截到管理程序中。



CN 113544678 B

1.一种计算机实现的方法,包括:

由在主机服务器上执行的虚拟机执行指令流,其中,来自所述指令流的指令将要被拦截到管理程序;

基于确定所述虚拟机是安全虚拟机,阻止所述管理程序直接访问所述安全虚拟机的任何数据;以及

由所述主机服务器的安全接口控件执行:

基于确定所述指令不能够由所述安全接口控件本身来解释:

由所述安全接口控件从所述安全虚拟机提取与所述指令相关联的一个或多个参数数据;

由所述安全接口控件将所述一个或多个参数数据存储到能够由所述管理程序访问的缓冲器中;以及

由所述安全接口控件拦截所述指令到所述管理程序中。

2.根据权利要求1所述的计算机实现的方法,其中,基于确定所述指令能够由所述安全接口控件本身解释:

由所述安全接口控件执行所述指令;以及

将执行控制返回到所述安全虚拟机以继续执行所述指令流。

3.根据权利要求1所述的计算机实现的方法,其中,拦截所述指令还包括:

由所述安全接口控件设置第一标志,所述第一标志指示所述指令被部分完成;以及

由所述安全接口控件设置第二标志,所述第二标志指示所述安全虚拟机处于执行的锁住状态。

4.根据权利要求3所述的计算机实现的方法,其中,所述安全接口控件阻止分派处于所述锁住状态的所述安全虚拟机,所述锁住状态指示所述安全虚拟机正在等待来自所述管理程序的响应。

5.根据权利要求1至4中的任一项所述的计算机实现的方法,还包括:

基于所述安全接口控件确定所述指令在被执行时导致程序异常,将所述异常呈现给所述安全虚拟机。

6.根据权利要求1至4中的任一项所述的计算机实现的方法,还包括:

在完成所述管理程序对所述指令的执行时,由所述安全接口控件使用来自所述管理程序的响应来更新所述安全虚拟机的状态,所述状态的至少一部分被存储在存储器的不能由所述管理程序访问的安全部分中。

7.根据权利要求6所述的计算机实现的方法,其中,所述管理程序将对所述指令的所述响应存储在专用缓冲器中。

8.根据权利要求6所述的计算机实现的方法,还包括:

在所述执行完成时,基于确定由所述管理程序生成的所述响应是无效的,由所述安全接口控件拦截错误状况到所述管理程序。

9.一种计算机系统,包括:

存储器;

安全接口控件;以及

与所述存储器和所述安全接口控件耦接的处理单元,所述处理单元被配置为执行托管

多个虚拟机的管理程序,所述管理程序被禁止直接访问安全虚拟机的任何数据,并且其中,所述管理程序被配置为执行用于解释来自所述虚拟机的一个或多个客户机指令的方法,所述方法包括:

由虚拟机执行指令流,其中,来自所述指令流的指令将要被拦截到管理程序;

确定所述虚拟机是安全虚拟机;以及

由所述安全接口控件执行:

基于确定所述指令不能够由所述安全接口控件本身来解释:

由所述安全接口控件从所述安全虚拟机提取与所述指令相关联的一个或多个参数数据;

由所述安全接口控件将所述一个或多个参数数据存储到能够由所述管理程序访问的缓冲器中;以及

由所述安全接口控件拦截所述指令到所述管理程序中。

10.根据权利要求9所述的系统,其中,基于确定所述指令能够由所述安全接口控件本身来解释:

由所述安全接口控件执行所述指令;以及

将执行控制返回到所述安全虚拟机以继续执行所述指令流。

11.根据权利要求9所述的系统,其中,拦截所述指令还包括:

由所述安全接口控件设置第一标志,所述第一标志指示所述指令被部分完成;以及

由所述安全接口控件设置第二标志,所述第二标志指示所述安全虚拟机对于执行被锁住。

12.根据权利要求9至11中的任一项所述的系统,其中,所述方法还包括:

基于所述安全接口控件确定所述指令在被执行时导致程序异常,将所述异常呈现给所述安全虚拟机。

13.根据权利要求9至11中的任一项所述的系统,其中,所述方法还包括:

在完成所述管理程序对所述指令的执行时,由所述安全接口控件使用来自所述管理程序的响应来更新所述安全虚拟机的状态,所述状态的至少一部分被存储在存储器的不能由所述管理程序访问的安全部分中。

14.根据权利要求13所述的系统,其中,所述管理程序将对所述指令的所述响应存储在专用缓冲器中。

15.根据权利要求13所述的系统,其中,所述方法还包括:

在所述执行完成时,基于确定由所述管理程序生成的所述响应是无效的,由所述安全接口控件拦截错误状况到所述管理程序。

16.一种计算机程序产品,包括计算机可执行指令,所述计算机可执行指令在由处理单元执行时使所述处理单元执行一种方法,所述方法包括:

由在主机服务器上执行的虚拟机执行指令流,其中,来自所述指令流的指令将要被拦截到管理程序;

确定所述虚拟机是安全虚拟机,所述管理程序被阻止直接访问所述安全虚拟机的任何数据;以及

由所述主机服务器的安全接口控件执行:

基于确定所述指令不能够由所述安全接口控件本身来解释：

由所述安全接口控件从所述安全虚拟机提取与所述指令相关联的一个或多个参数数据；

由所述安全接口控件将所述一个或多个参数数据存储到能够由所述管理程序访问的缓冲器中；以及

由所述安全接口控件拦截所述指令到所述管理程序中。

17.根据权利要求16所述的计算机程序产品，其中，基于确定所述指令能够由所述安全接口控件本身来解释：

由所述安全接口控件执行所述指令；以及

将执行控制返回到所述安全虚拟机以继续执行所述指令流。

18.根据权利要求16或17所述的计算机程序产品，其中，所述方法还包括：

在完成所述管理程序对所述指令的执行时，由所述安全接口控件使用来自所述管理程序的响应来更新所述安全虚拟机的状态，所述状态的至少一部分被存储在存储器的不能由所述管理程序访问的安全部分中。

19.根据权利要求18所述的计算机程序产品，其中，所述管理程序将对所述指令的所述响应存储在专用缓冲器中。

20.根据权利要求18所述的计算机程序产品，其中，所述方法还包括：

在所述执行完成时，基于确定由所述管理程序生成的所述响应是无效的，由所述安全接口控件拦截错误状况到所述管理程序。

21.一种计算机实现的方法，包括：

由主机机器上的管理程序执行来自安全虚拟机的客户机指令，所述管理程序被禁止直接访问所述安全虚拟机的任何数据；

由所述管理程序将对所述客户机指令的响应存储在预定缓冲器中；以及

由所述主机机器的安全接口控件通过将所述响应复制到所述安全虚拟机的状态描述符中的指定寄存器中来用所述响应更新所述状态描述符，所述状态描述符被存储在存储器的不能由所述管理程序访问的安全部分中。

22.根据权利要求21所述的计算机实现的方法，还包括：

基于确定由所述管理程序生成的所述响应是无效的，由所述安全接口控件拦截错误状况到所述管理程序。

23.一种计算机系统，包括：

存储器；

安全接口控件；以及

与所述存储器和所述安全接口控件耦接的处理单元，所述处理单元被配置为执行托管多个虚拟机的管理程序，所述管理程序被禁止直接访问安全虚拟机的任何数据，并且其中，所述管理程序被配置为执行用于解释来自所述虚拟机的一个或多个客户机指令的方法，所述方法包括：

由所述管理程序执行来自安全虚拟机的客户机指令，所述管理程序被禁止直接访问安全虚拟机的任何数据；

由所述管理程序将对所述客户机指令的响应存储在预定缓冲器中；以及

由安全接口控件通过将所述响应复制到所述安全虚拟机的状态描述符中的指定寄存器中来用所述响应更新所述状态描述符,所述状态描述符被存储在存储器的不能由所述管理程序访问的安全部分中。

24.根据权利要求23所述的系统,其中,所述方法还包括:

基于确定由所述管理程序生成的所述响应是无效的,由所述安全接口控件拦截错误状况到所述管理程序。

25.根据权利要求23或24所述的系统,其中,所述方法还包括:

由所述安全接口控件重置与所述安全虚拟机相关联的第一标志,所述重置指示所述安全虚拟机能够被分派以继续指令流的执行。

安全虚拟机环境中的客户机指令的透明解释

背景技术

[0001] 本申请涉及计算机技术,并且更具体地,涉及虚拟机。

[0002] 云计算促进了快速且容易地为客户供应虚拟机的能力,而不需要客户购买硬件或为物理服务器提供地面空间。客户可以根据改变的偏好来扩展或收缩虚拟机。通常,云计算提供商供应物理上驻留在提供商的数据中心处的虚拟机。在这种环境中,客户的虚拟机作为客户机运行,并且云提供商使用作为主机运行的管理程序代码来虚拟化可能属于不同客户的多个虚拟机之间的服务器资源。

[0003] 客户通常关心虚拟机中的数据的安全性。云操作者可能不是受信的,并且客户可能想要在没有被恶意或不完善的代码(如管理程序)和/或具有恶意操作数据中心的意图的系统管理员连累的风险的情况下部署他们的工作。例如,客户可以请求云提供商不能访问他们的数据,以便减少或避免诸如美国(U.S.)公司的云计算提供商可能通过传票被迫移交机密或专有文档的可能性。

发明内容

[0004] 根据本发明的一个或多个实施例,一种计算机实现的方法包括由在主机服务器上执行的虚拟机执行指令流,其中,来自指令流的指令将要被拦截到管理程序。该方法还包括基于确定虚拟机是安全虚拟机,阻止管理程序直接访问安全虚拟机的任何数据。该方法还包括由主机服务器的安全接口控件基于确定指令不能被安全接口控件本身解释来执行:从安全虚拟机提取与指令相关联的一个或多个参数数据,以及将参数数据存储到管理程序可访问的缓冲器中。指令随后被拦截到管理程序中。

[0005] 根据本发明的一个或多个实施例,该方法还包括由主机服务器的安全接口控件基于确定该指令可由安全接口控件本身解释来执行:由安全接口控件执行该指令,以及将执行控制返回给安全虚拟机以便继续执行指令流。

[0006] 根据本发明的一个或多个实施例,拦截指令包括由安全接口控件设置第一标志,该第一标志指示指令被部分完成,以及由安全接口控件设置第二标志,该第二标志指示安全虚拟机对于执行被锁住。

[0007] 根据本发明的一个或多个实施例,该方法还包括基于安全接口控件确定指令在被执行时导致程序异常,将异常呈现给安全虚拟机而不是将指令拦截到管理程序中。

[0008] 根据本发明的一个或多个实施例,该方法还包括,在管理程序完成指令的执行时,由安全接口控件用来自管理程序的响应来更新安全虚拟机的状态,该状态的至少一部分被存储在存储器的不能由管理程序访问的安全部分中。

[0009] 根据本发明的一个或多个实施例,管理程序将对指令的响应存储在专用缓冲器中。

[0010] 根据本发明的一个或多个实施例,该方法还包括,在执行完成时,基于确定由管理程序生成的响应是无效的,由安全接口控件拦截错误状况到管理程序。

[0011] 根据本发明的一个或多个实施例,安全接口控件包括毫码。

[0012] 此外,根据本发明的一个或多个实施例,上述特征至少由系统、计算机程序产品和机器提供。

[0013] 根据本发明的一个或多个实施例,一种计算机实现的方法包括由主机机器上的管理程序执行来自安全虚拟机的客户机指令,所述管理程序被禁止直接访问所述安全虚拟机的任何数据。所述方法还包括由所述管理程序将对所述客户机指令的响应存储在预定缓冲器中。该方法还包括由主机机器的安全接口控件通过将响应复制到安全虚拟机的状态描述符中的指定寄存器中来用响应更新状态描述符,状态描述符被存储在存储器的不能由管理程序访问的安全部分中。

[0014] 根据本发明的一个或多个实施例,该方法还包括基于确定由管理程序生成的响应是无效的,由安全接口控件拦截错误状况到管理程序。

[0015] 根据本发明的一个或多个实施例,该方法还包括由安全接口控件重置与安全虚拟机相关联的第一标志,该重置指示安全虚拟机可被分派以继续指令流的执行。

[0016] 上述方法的特征还可以至少由系统、计算机程序产品和机器提供。

[0017] 本文描述的特征提供了对计算机技术的改进,尤其是通过促进计算机服务器托管安全VM来托管虚拟机(VM)的计算机服务器。在安全VM的情况下,管理程序在VM数据的控制下不再受信任,并且甚至禁止管理程序访问存储器、寄存器和与安全VM相关联的其它此类数据。此外,本文描述的技术特征促进主机计算机服务器促进安全VM和管理程序的分离,从而维护由计算服务器托管的VM的安全性。

[0018] 通过本发明的技术实现了额外的技术特征和益处。本发明的实施例和各方面在本文中详细描述,并且被认为是所要求保护的主题的一部分。为了更好地理解,参考详细描述和附图。

附图说明

[0019] 在说明书的结尾处的权利要求中特别指出并清楚地要求了本文描述的专有权的细节。从下面结合附图的详细描述中,本发明的实施例的前述和其它特征和优点将变得显而易见,其中:

[0020] 图1示出了根据本发明实施例的云计算环境;

[0021] 图2示出了根据本发明实施例的抽象模型层;

[0022] 图3示出了根据实施例的用于托管系统的示例系统;

[0023] 图4示出了根据实施例的托管系统的示例框图;

[0024] 图5示出了根据本发明的一个或多个实施例的用于在安全虚拟机环境中透明解释客户机指令的示例方法的流程图;以及

[0025] 图6示出了根据本发明的一个或多个实施例的用于在安全虚拟机环境中透明解释客户机指令的示例方法的流程图。

具体实施方式

[0026] 在此参考相关附图描述本发明的各种实施例。在不偏离本发明的范围的情况下,可以设计本发明的替代实施例。在以下描述和附图中,在元件之间阐述了各种连接和位置关系(例如,上方、下方、相邻等)。除非另有说明,这些连接和/或位置关系可以是直接的或

间接的,并且本发明并不旨在在这方面进行限制。因此,实体的耦接可以指直接或间接耦接,并且实体之间的位置关系可以是直接或间接位置关系。此外,本文所述的各种任务和过程步骤可被并入具有本文未详细描述的额外步骤或功能性的更综合的程序或过程中。

[0027] 以下定义和缩写用于解释权利要求和说明书。如本文所用,术语“包含”、“包括”、“具有”、“含有”或其任何其它变型旨在涵盖非排他性的包括。例如,包括一系列元素的组合物、混合物、工艺、方法、制品或装置不一定仅限于那些元素,而是可以包括未明确列出的或此类组合物、混合物、工艺、方法、制品或装置固有的其他元素。

[0028] 另外,术语“示例性”在本文中用于表示“用作示例、实例或说明”。在此描述为“示例性”的任何实施例或设计不一定被解释为比其它实施例或设计更优选或有利。术语“至少一个”和“一个或多个”可以被理解为包括大于或等于一的任何整数,即,一、二、三、四等。术语“多个”可以被理解为包括大于或等于二的任何整数,即二、三、四、五等。术语“连接”可以包括间接“连接”和直接“连接”两者。

[0029] 术语“约”、“基本上”、“大约”及其变体旨在包括与基于提交本申请时可用的设备的特定量的测量相关联的误差度。例如,“约”可以包括给定值的±8%或5%或2%的范围。

[0030] 为了简洁起见,与制造和使用本发明的各方面相关的常规技术可以或可以不在本文中详细描述。特别地,用于实现本文描述的各种技术特征的计算系统和特定计算机程序的各个方面是公知的。因此,为了简洁起见,许多常规实现细节在本文中仅简要提及或被完全省略,而不提供众所周知的系统和/或过程细节。

[0031] 关于典型云环境的技术挑战是对数据和算法的潜在非安全和不想要的访问(例如,通过云提供商或云管理员)。云提供商通常将管理程序代码作为主机运行,而客户的VM作为客户机运行。该管理程序代码提供允许多个VM在单个物理机上运行所需的虚拟化功能。在现有系统中,管理程序(并且通常通过扩展,云管理员)具有对客户的数据和算法的访问权,以用于它必须访问该数据的有限部分以提供虚拟化功能的情形。例如,云提供商可以转储(dump)存储器以分析系统中的性能和功能问题。经转储的存储器可以包括客户不希望暴露的客户秘密。系统操作员还能够显示内部处理器寄存器状态,该状态也暴露秘密。管理程序被要求访问客户机数据,例如,以解释客户机指令,以及代表客户机进行I/O操作,以及其它原因。需要对客户机存储器的管理程序访问以便提供客户机功能正确性。在安全执行中,如本发明的一个或多个实施例所使用的,管理程序不再受信,但仍可参与客户机指令解释。因此,本发明的一个或多个实施例提供了一种用于非受信管理程序安全地仿真客户机指令的方式。

[0032] 在主机管理程序的控制下作为客户机运行的虚拟机依赖于该管理程序为该客户机透明地提供虚拟化服务。这些服务可以包括但不限于存储器管理、指令仿真和中断处理。本发明的一个或多个实施例可以应用于安全实体和另一个非受信实体之间的任何接口,该接口传统上允许该另一个实体访问安全资源。例如,对于中断和异常解释,管理程序通常读取和/或写入客户机的前缀区域(低核)。如本文所使用的术语“虚拟机”或“VM”指的是物理机(计算设备、处理器等)及其处理环境(操作系统(OS)、软件资源等)的逻辑表示。虚拟机状态由在底层主机(物理处理器或处理器组)上执行的管理程序来维护。从用户或软件资源的角度来看,VM看起来是它自己的独立物理机器。这里使用的术语“管理程序”和“VM监视器(VMM)”是指管理和允许多个VM在同一主机上使用多个(有时是不同的)OS来执行的处理环

境或平台服务。应当理解,部署VM包括:VM的安装过程和VM的激活(或启动)过程。在另一个示例中,部署VM包括:VM的激活(或启动)过程(例如,在VM被事先安装或已经存在的情况下)。

[0033] 然而,为了促进安全客户机,在诸如托管节点的计算服务器必须在管理程序和安全客户机之间提供附加的安全性的情况下存在技术挑战,使得管理程序不能从VM访问数据,并且因此不能以如上述的方式提供服务。

[0034] 在当前可用的技术方案中,管理程序(例如,IBM®的z/VM®或基于开源软件内核的虚拟机(KVM))通过发出使得开始解释执行(SIE)进入毫码被调用的SIE指令来分派物理处理单元或主机服务器上的新VM虚拟CPU(vCPU)。SIE指令的操作数是控制块,被称为状态描述(SD),其包含客户机状态。在现有的实现中,该状态描述驻留在管理程序存储中。在SIE进入期间,该客户机状态(包括通用寄存器和控制寄存器、客户机指令地址和客户机程序状态字(PSW))由毫码加载到硬件中。这允许客户机vCPU在物理处理器上运行。当vCPU在硬件上运行时,客户机状态被维持在硬件中。在某一点,硬件/毫码必须将控制返回给管理程序。这通常被称为SIE退出。例如,如果该vCPU执行需要由管理程序仿真的指令,或者如果vCPU时间片(即,分配给该vCPU在物理处理器上运行的时间)到期,则可能需要这种处理。在SIE退出期间,由于硬件在任何给定时间具有支持仅单个vCPU的资源,并且它现在必须将管理程序状态加载到硬件中,所以毫码将当前客户机状态保存在状态描述中。当该vCPU未被分派时,其状态被保持在状态描述中。由于该状态描述位于管理程序存储内,因此在这种情况下,管理程序具有对VM的数据的控制,并且在一些情况下,需要这种控制来解释在VM上执行的指令。现有的管理程序依赖于使用此类接口通过SIE指令来分派vCPU。

[0035] 此外,管理多个VM的管理程序必须解释客户机指令,即,在VM上执行的指令。作为这种解释的一部分,在现有解决方案中,管理程序通常具有对SD和VM的其他数据的完全访问,SD和数据被存储在作为正在执行管理程序的主机机器的一部分的存储器中。应当注意,并非在VM上执行的所有指令都需要由管理程序解释。通常,需要系统级访问(例如I/O访问)的指令需要此类管理程序解释。此类需要管理程序解释的客户机指令导致SIE退出拦截(客户机指令流的中断)从客户机模式出来到管理程序。管理程序然后检查哪个指令引起拦截,并且通过读取和/或写入存在于客户机寄存器或客户机存储器中的对应客户机数据来解释它。然后,重新分派所拦截的VM以继续执行其指令流。

[0036] 应当注意,通常,“指令拦截”的情况包括VM的操作系统发出CPUID指令以标识硬件特性、访问机器专用寄存器(MSR)或直接访问I/O端口。当检测到需要拦截的这种指令时,控制立即被转移到管理程序12以进行解析。例如,如果VM发出指令以读取或更新控制寄存器(CR)或机器专用寄存器(MSR)值,则在一些情况下,该操作被拦截,并且控制被转移到管理程序12,其中VM的行为由管理程序模拟,管理程序需要从VM的状态描述符访问一个或多个数据值。此外,所拦截的操作还可以包括主机程序异常,其由管理程序12处理。

[0037] 在安全VM的情况下,当管理程序在安全VM的数据的控制下不再受信时,客户机指令的传统解释不再是可能的。存在这种技术挑战是因为管理程序不能访问客户机数据(寄存器及其存储器)。即使管理程序被给予受限的客户机状态访问,管理程序也不能观察安全VM的行为以便阻止或限制副信道攻击。因此,对于托管节点的技术挑战是当VM具有访问托管节点的系统资源(例如I/O设备等)的有限“功率”或“权限”时,促进安全接口控件和/或管

理程序解释和执行安全VM中的客户机指令。

[0038] 根据本发明的一个或多个实施例,描述了技术方案,以解释安全VM中的指令,而不将客户机数据暴露给管理程序,并且不需要重写客户机应用或操作系统。换言之,毫码和管理程序对客户机指令的解释对安全VM本身是透明的。因此,可以照原样使用现有应用和操作系统,并且提供这种安全VM的托管节点和管理程序与现有产品兼容。换言之,本发明的一个或多个实施例通过促进现有VM、操作系统和计算机应用代码不必改变并且该代码可在托管节点和/或管理程序中使用来解决技术挑战,所述托管节点和/或管理程序也以安全方式分派VM,从而禁止管理程序访问安全VM的状态和/或数据。本发明的一个或多个实施例通过在将要解释安全VM的客户机指令的安全接口控件(例如毫码)中使用客户机指令解释的改进的底层实现,来促进此类技术方案。

[0039] 在一个或多个示例中,此类功能可以通过使用毫码和/或其他硬件模块来提供,并且在本描述中,硬件模块和毫码被统称为“安全接口控件”。因此,本发明的一个或多个实施例促进管理程序通过使用安全接口控件(毫码和内部受信硬件和固件)由非受信管理程序对安全VM中的指令进行安全解释。安全接口控件协调安全VM和非受信管理程序之间的指令的执行。客户机指令像在非安全VM中一样被拦截,但是控制不立即被给回管理程序,而是控制首先被给到安全接口控件,安全接口控件提取必要的客户机数据并将其连同客户机指令的拦截条件和附加上下文信息一起传递给管理程序。管理程序处理该请求并将响应返回给安全接口控件。安全接口控件然后将根据需要取决于所拦截的指令来更新客户机状态。因此,安全接口控件将管理程序与安全VM的数据隔离,从而提高与安全VM相关联的数据的安全性。

[0040] 现在,以下简要描述背景技术,之后描述由本发明的一个或多个实施例用于由管理程序注入中断和/或异常到安全VM的特定特征。应预先理解,尽管本公开包括关于云计算的详细描述,但本文中所述的教导的实现不限于云计算环境。相反,本发明的实施例能够结合现在已知或以后开发的任何其它类型的计算环境来实现。

[0041] 云计算是服务交付的模型,用于对共享的可配置计算资源池进行方便、按需的网络访问。可配置计算资源(例如,网络、网络带宽、服务器、处理、存储器、存储、应用、虚拟机和服务)能够以最小的管理成本或与服务提供商进行最少的交互来快速部署和释放。这种云模型可以包括至少五个特征,至少三个服务模型和至少四个部署模型。

[0042] 特征如下:

[0043] 按需自助式服务:云的消费者在无需与服务提供商进行人为交互的情况下,能够单方面自动地按需部署计算能力,诸如服务器时间和网络存储。

[0044] 广泛的网络接入:计算能力通过网络获得,并通过标准机制访问,该标准机制促进了通过不同种类的瘦客户机平台或厚客户机平台(例如,移动电话,膝上型电脑和PDA)的使用。

[0045] 资源池:提供商的计算资源被归入资源池,以使用多租户模式为多个消费者提供服务,其中根据需求动态分配和重新分配不同的实体资源和虚拟资源。通常消费者不能控制或者并不知晓所提供的资源的确切位置,但是可能能够在更高抽象级别(例如,国家,州或数据中心)指定位置,因此具有位置无关性。

[0046] 迅速弹性:可以迅速、有弹性地(有时是自动地)部署计算能力,以快速扩展,并且

能迅速释放来快速缩小。对于消费者来说,用于部署的可用计算能力通常显得是无限的,并能在任意时候都能获取任意数量的计算能力。

[0047] 可测量的服务:云系统通过利用适于服务类型(例如,存储,处理,带宽和活跃用户帐户)的某种抽象级别的计量能力来自动控制和优化资源使用。可以监视,控制和报告资源使用,从而为所使用的服务的提供者和消费者提供透明度。

[0048] 服务模型如下:

[0049] 软件即服务(SaaS):提供给消费者的能力是使用提供商在云基础架构上运行的应用。可以通过诸如网络浏览器(例如,基于网络的电子邮件)的瘦客户机接口从各种客户机设备访问应用。消费者既不管理也不控制底层云基础架构,包括网络、服务器、操作系统、存储、甚至单个应用能力,可能的例外是有限的特定于用户的应用配置设置。

[0050] 平台即服务(PaaS):提供给消费者的能力是在云基础架构上部署消费者创建或获得的应用,这些应用利用由提供商支持的编程语言和工具创建。消费者既不管理也不控制底层云基础架构,包括网络、服务器、操作系统或存储,但可以控制已部署的应用以及可能的应用托管环境配置。

[0051] 基础架构即服务(IaaS):提供给消费者的能力是部署处理、存储、网络和其它基础计算资源,其中消费者能够部署和运行任意软件,该软件可以包括操作系统和应用。消费者既不管理也不控制底层云基础设施,而是具有对操作系统,存储,部署的应用的控制,以及具有可能的对选择的网络组件(例如,主机防火墙)的有限控制。

[0052] 部署模型如下:

[0053] 私有云:云基础架构单独为组织运行。它可以由组织或第三方管理,可以存在于该组织内部或外部。

[0054] 共同体云:云基础架构由多个组织共享,并支持具有共同利害关系(例如,任务使命,安全要求,策略和合规考虑)的特定共同体。它可能由组织或第三方管理,并且可能存在于该共同体内部或外部。

[0055] 公共云:云基础架构向公众或大型产业群提供,并由销售云服务的组织所有。

[0056] 混合云:云基础架构由两个或多个云(私有云,共同体云或公共云)组成,这些云仍然是独特的实体,但通过使数据和应用能够移植的标准化或私有技术(例如,用于云之间的负载平衡的云突发流量分担技术)绑定在一起。

[0057] 云计算环境是面向服务的,特点集中在无状态,低耦合性,模块性和语意的互操作性。云计算的核心是包含互连节点网络的基础设施。

[0058] 现在参考图1,描绘了示意性的云计算环境50。如图所示,云计算环境50包括云的消费者使用本地计算设备可以与其通信的一个或多个云计算节点10,本地计算设备例如是个人数字助理(PDA)或蜂窝电话54A,台式计算机54B,膝上型计算机54C和/或汽车计算机系统54N。节点10可以彼此通信。它们可以在一个或多个网络中物理地或虚拟地分组(未示出),例如如上所述的私有云,共同体云,公共云或混合云,或其组合。这样,云的消费者无需维护本地计算设备上的资源就能够允许云计算环境50提供基础架构即服务、平台即服务和/或软件即服务。应该理解,图1中所示的计算设备54A-N的类型仅仅是示意性的,而计算节点10和云计算环境50可以(例如,使用网络浏览器)通过任何类型的网络和/或网络可寻址连接与任何类型的计算设备通信。

[0059] 现在参考图2,示出了由云计算环境50(图6)提供的一组功能抽象层。应该事先理解图2中所示的组件、层和功能仅仅是示意性的,并且本发明的实施例不限于此。如图所示,提供了以下层和相应的功能:

[0060] 硬件和软件层60包括硬件和软件组件。硬件组件的示例包括主机61;基于RISC(精简指令集计算机)体系结构的服务器62;服务器63;刀片服务器64;存储设备65;网络和网络组件66。在一些实施例中,软件组件包括网络应用服务器软件67和数据库软件68。

[0061] 虚拟层70提供抽象层,从该抽象层可以提供以下虚拟实体的示例:虚拟机71;虚拟存储72;虚拟网络73(包括虚拟私有网络);虚拟应用和操作系统74;和虚拟客户端75。

[0062] 在一个示例中,管理层80可以提供下面描述的功能。资源供应功能81提供用于在云计算环境内执行任务的计算资源和其它资源的动态获取。计量和定价功能82在云计算环境内对资源的使用进行成本跟踪,并且提供用于消费这些资源的帐单或发票。在一个示例中,这些资源可以包括应用软件许可。安全功能为云的消费者和任务提供身份认证,以及为数据和其它资源提供保护。用户门户功能83为消费者和系统管理员提供对云计算环境的访问。服务水平管理功能84提供云计算资源的分配和管理,以满足所需的服务水平。服务水平协议(SLA)计划和履行功能85为根据SLA预测的对云计算资源未来需求提供预先安排和供应。

[0063] 工作负载层90提供可以利用云计算环境的功能的示例。可以从该层提供的工作负载和功能的示例包括:地图绘制与导航91;软件开发和生命周期管理92;虚拟教室的教学提供93;数据分析处理94;交易处理95;和源代码版本化96。可以理解,这些仅仅是一些示例,并且在其它实施例中,这些层可包括不同的服务。

[0064] 图3示出根据本发明的一个或多个实施例的示例托管节点10。托管节点10经由网络165与一个或多个客户端设备20A-20C通信。此外,客户端设备20A-20C可以与托管节点10直接通信,托管节点10可以是云计算提供商的数据中心或主机服务器。托管节点10执行管理程序12,其促进部署一个或多个虚拟机15(15A-15N)。托管节点10还包括安全接口控件11,该安全接口控件11包括一个或多个硬件模块和毫码,其促进管理程序12向虚拟机15提供一个或多个服务,在现有技术方案中,在管理程序12与安全接口控件11之间存在通信;在安全接口控件11和一个或多个VM 15之间存在通信;在管理程序12和一个或多个VM 15之间存在通信;以及存在通过安全接口控件11从管理程序12到VM 15的通信。为了促进安全VM环境,根据本发明的一个或多个实施例的托管节点10不包括管理程序12与一个或多个VM 15之间的任何直接通信。

[0065] 例如,托管节点10可以促进客户端设备20A部署虚拟机15A-15N中的一个或多个。虚拟机15A-15N可以响应于来自不同客户端设备20A-20C的相应请求而被部署。例如,虚拟机15A可由客户端设备20A部署,虚拟机15B可由客户端设备20B部署,并且虚拟机15C可由客户端设备20C部署。托管节点10还可以促进客户端提供物理服务器(而不作为虚拟机运行)。这里描述的示例将托管节点10中的资源供应具体化为‘虚拟机’的一部分,然而,所描述的技术方案可以应用于将资源供应为物理服务器的一部分。

[0066] 在一个示例中,客户端设备20A-20C可以属于同一实体,诸如个人、企业、政府机构、公司内的部门或任何其他实体,并且托管节点10可以作为实体的私有云来操作。在这种情况下,托管节点10单独地托管由属于该实体的客户端设备20A-20C部署的虚拟机15A-

15N。在另一个示例中,客户端设备20A-20C可以属于不同的实体。例如,第一实体可以拥有客户端设备20A,而第二实体可以拥有客户端设备20B。在这种情况下,托管节点10可以作为托管来自不同实体的虚拟机的公共云来操作。例如,虚拟机15A-15N可以以其中虚拟机15A不促进对虚拟机15B的访问的受遮蔽方式来部署。例如,托管节点10可以使用IBM z系统®处理器资源/系统管理器(PR/SM)逻辑分区(LPAR)特征来覆盖虚拟机15A-15N。这些特征,例如PR/SM LPAR提供分区之间的隔离,从而促进托管节点10在不同的逻辑分区中针对同一物理托管节点10上的不同实体部署两个或更多个虚拟机15A-15N。

[0067] 来自客户端设备20A-20C的客户端设备20A是通信设备,例如计算机、智能电话、平板计算机、台式计算机、膝上型计算机、服务器计算机或请求由托管节点10的管理程序12部署虚拟机的任何其它通信装置。客户端设备20A可以经由网络165或直接发送请求以由管理程序接收。虚拟机15A-15N中的虚拟机15A是管理程序12响应于来自客户端设备20A-20C中的客户端设备20A的请求而部署的虚拟机镜像。管理程序12是虚拟机监视器(VMM),其可以是创建并运行虚拟机的软件、固件或硬件。管理程序12促进虚拟机15A使用托管节点10的硬件组件来执行程序和/或存储数据。通过适当的特征和修改,管理程序12可以是IBM z系统®、ORACLE VM SERVER™、CITRIX XENSERVER™、VMWARE ESX™、MICROSOFT HYPER-V™或任何其它管理程序。管理程序12可以是直接在托管节点10上执行的本机管理程序,或者是在另一管理程序上执行的被托管的管理程序。

[0068] 图4示出根据本发明的一个或多个实施例的示例托管节点的组件。托管节点10可以是计算机,诸如服务器计算机、台式计算机、平板计算机、智能电话或执行管理程序12的任何其它计算机,其继而部署虚拟机15A-15N。托管节点10包括包含硬件(例如电子电路)的组件。托管节点10包括处理器105、耦接到存储器控制器115的存储器110以及一个或多个输入装置145和/或输出装置140,例如经由本地I/O控制器135通信地耦接的外围或控制装置,以及其它组件。这些装置140和145可包括例如电池传感器、位置传感器(高度计40、加速度计42、GPS 44)、指示器/识别灯等。诸如传统键盘150和鼠标155的输入设备可以被耦接到I/O控制器135。I/O控制器135可以是例如如本领域已知的一个或多个总线或其它有线或无线连接。I/O控制器135可以具有附加的元件,例如控制器、缓冲器(高速缓存)、驱动器、中继器和接收器,以实现通信,为了简单起见省略了这些元件。

[0069] I/O设备140、145还可以包括与输入端和输出端二者通信的设备,例如磁盘和磁带存储器、网络接口卡(NIC)或调制器/解调器(用于访问其他文件、设备、系统或网络)、射频(RF)或其他收发器、电话接口、桥接器、路由器等。

[0070] 处理器105是用于执行硬件指令或软件的硬件设备,特别是那些存储在存储器110中的硬件指令或软件。处理器105可以是定制的或商业上可获得的处理器、中央处理单元(CPU)、与托管节点10相关联的多个处理器中的辅助处理器、基于半导体的微处理器(以微芯片或芯片组的形式)、宏处理器或用于执行指令的其它装置。处理器105包括高速缓存170,其可以包括但不限于加速可执行指令提取的指令高速缓存、加速数据提取和存储的数据高速缓存、以及用于加速可执行指令和数据二者的虚拟到物理地址转换的转换后备缓冲器(TLB)。高速缓存170可以被组织为更多高速缓存级别(L1、L2等)的层级。

[0071] 存储器110可包括以下中的一个或其组合:易失性存储器元件(例如,随机存取存

储器RAM,诸如DRAM、SRAM、SDRAM)和非易失性存储器元件(例如,闪存、ROM、可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)、可编程只读存储器(PROM)、磁带、光盘只读存储器(CD-ROM)、磁盘、软磁盘、磁带盒、盒式磁带等)。此外,存储器110可以合并电、磁、光或其它类型的存储介质。注意,存储器110可以具有分布式架构,其中各种组件彼此远离,但是可以由处理器105访问。

[0072] 存储器110中的指令可包括一个或多个单独的程序,每个程序包括用于实现逻辑功能的可执行指令的有序列表。在图2的示例中,存储器110中的指令包括执行管理程序12的适当的操作系统(OS)。该操作系统可以控制其他计算机程序的执行,并且提供调度、输入-输出控制、文件和数据管理、存储器管理、以及通信控制和相关服务。在诸如z系统™的示例中,托管节点10的制造商可以提供管理程序12,在系统具有与z系统的结构不同的结构的情况下,其中管理程序12不是由硬件制造商提供的,所提供的云计算可以使用诸如来自VMWARE™的管理程序12或其他管理程序提供商。在示例中,物理托管节点10的管理员不能修改管理程序12,除非在需要时以便应用由制造商提供的服务。例如,管理程序12可以被提供为用于托管节点10的“许可内部码(LIC)”和/或微码的一部分。

[0073] 附加数据(包括例如用于处理器105的指令或其他可取得信息)可以被存储在存储装置120中,其可以是诸如硬盘驱动器或固态驱动器的存储设备。存储器110或存储装置120中的所存储指令可包括使得处理器能够执行本公开的系统和方法的一个或多个方面的指令。

[0074] 托管节点10还可以包括耦接到用户接口或显示器130的显示控制器125。在一些实施例中,显示器130可以是LCD屏幕。在其它实施例中,显示器130可以包括多个LED状态灯。在一些实施例中,托管节点10还可以包括用于耦接到网络165的网络接口160。网络165可以是用于经由宽带连接在主机节点10与外部服务器、客户端等之间通信的基于IP的网络。在一个实施例中,网络165可以是卫星网络。网络165在托管节点10和外部系统之间发送和接收数据。在一些实施例中,网络165可以是由服务提供商管理的受管理IP网络。网络165可以例如使用无线协议和技术(诸如WiFi、WiMax、卫星或任何其他)以无线方式实现。网络165还可以是分组交换网络,例如局域网、广域网、城域网、因特网或其他类似类型的网络环境。网络165可以是固定无线网络、无线局域网(LAN)、无线广域网(WAN)、个人区域网(PAN)、虚拟专用网(VPN)、内联网或其它合适的网络系统,并且可以包括用于接收和发送信号的设备。

[0075] 客户端设备20A可请求管理程序12部署具有对托管节点10的特定硬件和/或软件组件的访问权的对应虚拟机15A,例如,客户端设备20A可请求虚拟机15A具有对预定数量的处理器、预定量的易失性存储器(例如随机存取存储器(RAM))、预定量的非易失性存储器(例如存储空间)或任何其它硬件组件的访问权。可替换地或另外地,客户端设备20A可以请求虚拟机15A具有对特定硬件组件(例如由相应的唯一标识符标识的电子电路)的访问权。例如,客户端设备20A可以请求虚拟机15A具有对特定类型的处理器、协处理器、网卡、或任何其他芯片或电子电路的访问权。在一个示例中,客户端设备20A可以使用由电子电路的制造商提供的标识符来识别电子电路。在一个示例中,标识符可以与版本标识符结合使用。可替换地或另外地,客户端设备20A可请求虚拟机15A具有对特定的软件组件(例如操作系统、应用、基本输入/输出系统(BIOS)、启动镜像或任何其他软件组件)的访问权。所请求的软件组件可包括固件和托管节点10的硬件组件中的嵌入式程序。客户端设备20A可以使用由相

应软件组件的开发者/制造商提供的相应唯一标识符来识别所请求的软件组件。在一个示例中,标识符可以与软件组件的版本标识符结合使用。

[0076] 如前所述,对于将要成为安全VM的虚拟机15A,禁止所有非安全客户机和管理程序12对存储器110、存储装置120、寄存器的一个或多个部分以及与虚拟机15A相关联的任何其它数据的访问。在一个或多个示例中,管理程序12保证对于任何给定的驻留安全客户机页,仅可通过单个管理程序(主机)DAT映射来访问相关联的主机绝对地址。也就是说,存在映射到被分配给安全VM 15A的任何给定主机绝对地址的单个主机虚拟地址。此外,与任何给定安全客户机页相关联的管理程序DAT映射(主机虚拟到主机绝对)在其被页调入时不改变。此外,与任何安全客户机页相关联的主机绝对页仅被映射用于单个安全客户机。另外,在虚拟机15之间不存在存储器/寄存器的共享,尤其是在安全VM 15A的情况下。此外,在一个或多个示例中,管理程序12将存储装置120的安全部分分配给安全接口控件11。存储装置120的该安全部分仅在被分配后才可由安全接口控件11访问。在安全部分的内容被分配给安全接口控件11之后,虚拟机15和/或管理程序12都不能访问安全部分的内容。

[0077] 对这些规则的任何尝试违反都被安全接口控件11和托管节点10禁止,并且可以发出警报。可以通过向一个或多个人员发送通知、阻止托管节点10的操作、阻止来自一个或多个客户端设备20的请求、阻止安全VM 15(以及任何其他安全VM)的操作等来引发警报。

[0078] 图5示出了根据本发明的一个或多个实施例的用于管理程序执行对来自安全VM的客户机指令的透明解释的示例方法的流程图。该方法可包括在501处从客户端设备20A接收用于启动安全VM 15A的请求。在一个或多个示例中,可以从任何其他源接收请求,诸如另一VM 15B-15N、由管理程序12执行的计算机应用、管理员等。

[0079] 该方法包括为安全VM 15A创建非安全状态描述符(SD)。在一个或多个示例中,非安全SD包括对安全SD的引用,安全SD存储在存储器的不可由管理程序12或其他非安全实体访问的一部分中。例如,安全接口控件11创建安全SD并在非安全SD中添加对安全SD的引用。安全SD可包括:VM通用寄存器(GR)、访问寄存器(AR)、控制寄存器(CR)、VM计时器(包括时钟比较器和CPU计时器)、VM前缀寄存器、虚拟CPU号(VCN)、程序状态字(PSW)和指令地址(IA)。另外,SD可以包括控制信息,例如拦截控制(IC)位,以指示某些指令(例如,加载程序状态字(LPSW)、无效页表条目(IPTE)等)是否需要拦截主机,或者在VM指令执行可以开始之前是否需要清除VM转换后备缓冲器(TLB)。上述SD字段是示例性的,并且在本发明的一个或多个实施例中可以不同,例如包括用于其它VM状态的各种其它字段。在非安全VM的情况下,非安全SD本身包括在安全VM的情况下被描述为安全SD的一部分的上述字段/参数。

[0080] 管理程序12发出SIE指令,其以非安全SD作为操作数,指令的执行由安全接口控件11处理,SIE(开始解释执行)指令用于分派VM 15并将处理器105和其它计算资源分配给VM 15。SIE指令将处理器105置于非安全SD中定义的仿真状态。在安全VM的情况下,访问安全SD以启动安全VM 15A的分派。在一个或多个示例中,非安全SD包括标识符以指示正在被分派的VM是否是安全/非安全的,并且因此,安全接口控件11从存储器的适当安全/非安全部分访问字段/参数。

[0081] 该方法进一步包括在505处执行安全VM 15A中的客户机指令。在510处,安全VM 15A中的指令流可以继续执行,直到遇到导致拦截条件的指令。在指令导致拦截条件之后,在510处和515处,具有拦截指令的VM确定VM是否已被分派为安全VM。

[0082] 如果VM被分派为非安全VM,其中管理程序12可以访问与VM相关联的数据和/或存储器,则在520处,拦截指令的执行如现有方案中那样继续。安全接口控件基于存储在VM的SD中的状态标志来确定它是否已被分派为安全VM或非安全VM。

[0083] 在VM是非安全VM的情况下,暂停非安全VM中的指令流的执行,并且将控制转移到管理程序12,其通过直接访问非安全VM的SD来访问非安全VM的数据从而解释拦截指令。例如,管理程序12使用的数据可以包括被存储在非安全VM的GR、CR、客户机存储器和/或其他寄存器中的值。在所拦截的指令解释被完成之后,管理程序12通过存储一个或多个值作为响应来适当地更新客户机状态。例如,管理程序12通过更新在非安全VM的SD被维护的存储器位置中的值,将这些值直接存储到非安全VM的一个或多个寄存器中。管理程序12还可更新客户机存储器作为指令解释的一部分。

[0084] 或者,如果VM是安全VM 15A,则在525处调用安全接口控件11以执行拦截指令。在530处,安全接口控件11确定该指令是否应当导致由于执行而引起的抑制或无效(nullifying)程序异常。安全接口控件11模拟拦截指令的执行以确定该执行是否将导致程序异常。例如,指令可以请求在托管节点10中不可用的计算资源,替代地或附加地,拦截指令可以是在安全VM 15A正在执行的特权级别上不被允许的指令。替代地或附加地,拦截指令可在存储器存取期间违反边界条件。安全接口控件11可以检查将由拦截指令的执行引起的任何其他类型的程序异常。如果由于执行拦截指令而发生异常,则在535处,安全接口控件11将异常呈现给安全VM 15A。

[0085] 在一个或多个示例中,取决于程序中断的类型来抑制或无效指令。如果抑制或无效客户机异常适用,则指令拦截没有到达管理程序12,相反,控制被直接给到安全VM 15A的中断处理机以处理程序异常。没有理由拦截指令到管理程序,因为必须首先处理程序异常。例如,如果LCTLG指令将被拦截并且提供给该指令的存储操作数具有页错误,则将程序异常给到安全VM 15A以首先解决页错误。在解决了页错误之后,再次执行LCTLG,这次没有程序异常。因此,在后续执行中,使用本文描述的特征来将指令拦截到管理程序12。

[0086] 一些程序异常是主机相关的。在这种情况下,SIE退出被调用,并且程序异常被提供给管理程序12。例如,在主机页故障的情况下,管理程序12将客户机页映射到主机绝对页,然后重新分派VM。

[0087] 应当注意,程序异常可以是无效、抑制和完成。抑制或无效之间的差别在于指令地址(IA)被更新的方式。在无效异常中,IA指向引起异常的指令,而在抑制异常中,IA指向在具有异常的指令之后的下一顺序指令。在这两种情况下,都不对客户机状态或存储器进行其他更新。

[0088] 或者,如果该拦截指令导致完成异常,则在537处安全接口控件11检测并标记该拦截指令的完成异常标志。完成异常(如程序事件记录(PER))通常在相关联的指令完成之后被呈现。在非安全VM环境中,管理程序12检测此类PER异常,并且在管理程序12完成解释客户机指令之后,将它们呈现给VM。在安全VM环境中,安全接口控件11检测此类PER程序异常,在客户机指令完成执行之后,将它们标记为要呈现给安全VM。

[0089] 此外,在540处,安全接口控件11检查哪个指令正在被拦截。例如,安全接口控件11基于将要被拦截的指令,并且在某些情况下,指令的多个操作数之一确定指令是否可以被安全接口控件11本身解释。诸如写入可以使能客户机异步中断(例如,LCTLG指令)的客户机

控制寄存器的一些操作通常由安全接口控件完成,然后被拦截,向管理程序12提供最低限度需要的客户机状态,使得管理程序12可以检查任何先前活动但被禁用的客户机中断是否被使能。管理程序12然后对未处理且被启用的客户机中断划分优先级,并通过安全接口控件将最高优先级中断安全地呈现给客户机。根据本发明的一个或多个实施例,在安全VM环境中,安全接口控件11代表安全VM解释LCTLG解释。类似地,对于拦截回到管理程序12的用于非安全VM的设置前缀(SPX)指令,在本发明的一个或多个实施例中,将由安全接口控件11来解释。如果操作数可由安全接口控件11访问,则该指令可被认为可由安全接口控件11解释。如果该指令可由安全接口控件11解释,则指令执行由安全接口控件11本身完成,并且在545处恢复安全VM 15A的指令流。

[0090] 如果指令不可由安全接口控件11本身解释,则在550处,安全接口控件11将与客户机指令相关联的部分-完成标志设置为第一状态,例如,部分-完成=1。应当理解,尽管在本申请文件中部分-完成标记被描述为具有第一状态=1和第二状态=0,但是在本发明的一个或多个实施例中,可以针对这两个状态使用其他值。

[0091] 此外,在552处,安全接口控件11将与包括拦截指令的安全VM 15A相关联的锁住-VM标志设置为第一状态,例如锁住-VM=1。可以理解,尽管在本申请文件中锁住-VM标志被描述为具有第一状态(例如,=1)和第二状态(例如,=0),但是在本发明的一个或多个实施例中可以针对这两个状态使用其他值。锁住-VM标志指示安全VM 15A是否被“锁住”,即,指令流的执行是否被暂停,等待指令的解释完成。锁住-VM标志还指示是否已经验证了对由安全控制接口转发到安全VM 15A的被拦截的指令的管理程序响应。在这里描述的示例中,考虑在第一状态(即,锁住-VM=1)中,对安全VM 15A的响应将要被验证,否则(即,锁住-VM=0),此时不必验证该响应。锁住-VM还阻止管理程序12在没有响应的情况下意外或恶意地分派安全VM,并且还阻止管理程序在完成指令解释之前试图将中断注入到安全VM 15A中。因此,锁住-VM锁住安全VM 15A,直到它接收到正在被解释的客户机指令所需的响应。

[0092] 当安全VM 15A处于锁住状态时,向安全接口控件11指示安全VM 15A正在等待来自管理程序12的与指令解释有关的响应。在这种情况下,安全接口控件11阻止安全VM 15A被分派,直到安全接口控件接收到有效响应,在这种情况下,锁住-VM标志的状态被改变。

[0093] 在555处,安全接口控件11进一步拦截指令供管理程序12执行。到管理程序12的拦截包括安全接口控件11从存储器的安全部分提取与指令相关联的一个或多个参数数据,并使其可由管理程序12访问。安全接口控件11可以通过检查指令、其操作数及其数据来从安全VM 15A提取参数数据。操作数和数据可以被存储在与安全VM 15A相关联的硬件寄存器和存储器中。在SIE退出中,客户机状态被保存在安全SD中,并且拦截原因(和其它管理程序相关信息)被保存在非安全SD中。VM正在运行的物理处理器现在是空闲的,并且管理程序可以重新分派同一VM或者将新的VM分派到同一物理处理器。锁住-VM和与所拦截的客户机指令相关的其它信息被保存在安全SD中,从而允许将VM分派到任何可用的处理器。

[0094] 安全接口控件11可以通过创建用于执行该特定指令的专用缓冲器或使用用于执行此类所拦截的客户机指令的专用缓冲器,来使得所提取的参数可被管理程序12访问。安全接口控件11将来自安全VM 15A的数据(管理程序为指令解释所需)存储在专用缓冲器中,并将所存储的数据作为将要用于解释正被拦截的客户机指令的操作数/数据传递到管理程序12。

[0095] 管理程序12通常不知道来自安全VM 15A中的指令流的哪个指令引起拦截-仅知道在它们的部分上所需的特定动作。在拦截时,管理程序12被调度为执行指令解释,而没有在解释所需的上下文之外的任何其它上下文,诸如指令被执行的原因。因此,在560处,管理程序12在不知道指令的源/原因的情况下解释指令。

[0096] 图6描绘了根据本发明的一个或多个实施例的在完成对所拦截的安全VM指令的解释之后从管理程序向安全VM返回响应的方法的流程图。在完成指令解释时,管理程序12分派安全VM,而不进行任何中断注入(562)或进行中断注入(564)。可以使用如这里所述的SIE指令来执行分派。作为安全VM分派操作的一部分,从与VM相关联的安全SD读取客户机状态,并将其加载到处理器硬件(寄存器、存储器等)。在一个或多个示例中,管理程序12通过将中断参数相关信息(诸如标识符和参数)存储在专用缓冲器中,经由安全接口控件11将中断注入到VM中。安全接口控件11接收SIE调用的执行控制。

[0097] 在570处,安全接口控件11检查锁住-VM标志是否被设置为1。如果锁住-VM=0(或不等于1),则,安全接口控件11认为不同的VM(没有拦截指令)正由被调用的SIE指令分派,并且因此,在575处进入不同的VM并且发起用于该VM的指令流的执行。

[0098] 或者,如果锁住-VM=1,即管理程序12正将执行控制返回到发出拦截指令的安全VM 15A,则在580处,安全接口控件11检查是否从管理程序12接收到对指令的响应。安全接口控件11通过检查专用缓冲器中的指定位置来进行确定,该指定位置是为管理程序12设置的,以提供指令响应。如果没有找到响应,则在582处,安全接口控件11用错误状况拦截管理程序12。如果在专用缓冲器中提供了响应,则在585处,安全接口控件11确保该响应是有效的。如果响应是无效的,则在582处,安全接口控件11用错误状况拦截管理程序12。

[0099] 在一个或多个示例中,安全接口控件11基于从管理程序接收的响应类型来确保响应的有效性。例如所提供的响应对于所拦截的指令类型是有效的。因此,对于所允许的每个拦截指令,安全接口控件11具有所允许的一个或多个可能类型的响应。例如,安全接口控件11从针对拦截指令的预期响应类型的列表中检查响应的数据类型是有效的。例如,如果拦截指令是对I/O操作的请求,则安全接口控件11执行检查以验证响应是适合于I/O操作的数据类型。

[0100] 如果确定响应是有效的,则该方法继续进行,在590处,安全接口控件11基于来自管理程序12的响应来更新安全VM 15A的状态。更新状态可以包括更新与安全VM 15A相关联的寄存器和存储器。只有安全接口控件11可以访问安全VM 15A的SD,因为它被存储在存储器的仅可由安全接口控件11访问的安全部分中。因此,响应保持安全并且不可从其它VM以及管理程序12访问。

[0101] 此外,在592处,安全接口控件11完成拦截客户机指令的执行。完成包括安全接口控件11将标志部分-完成和锁住-VM更新为第二状态,即,针对安全VM 15A设置部分-完成=0和锁住-VM=0。在595处,安全接口控件11确定该拦截指令是否具有与其相关联的对应的完成异常。如果异常存在,则安全接口控件11将完成异常呈现给安全VM 15A,其指示安全VM 15A在处理异常之后在597处恢复其正在执行的指令流。或者,如果对于拦截指令不存在程序异常,则安全接口控件11将执行控制返回到安全VM 15A,其继而在598处重新开始指令流。

[0102] 因此,本发明的一个或多个实施例促进由主机节点中的管理程序解释安全VM的客

户机指令。客户机指令被正常拦截(如在非安全VM中),但是控制不被立即给回到管理程序,而是被给到安全接口控件。安全接口控件提取客户机数据并将其与拦截原因一起传递给管理程序。管理程序处理该请求并将响应返回给安全接口控件。

[0103] 安全接口控件检查包括程序事件记录(PER)的程序异常,并在将请求传递到管理程序之前将客户机指令标记为部分完成。在存在无效或抑制程序拦截的情况下,不向管理程序发信号通知,并且采取程序异常。安全接口控件还在更新客户机指令结果之前检查和验证管理程序响应。对于像程序事件记录(PER)的完成类型程序异常,安全接口控件标记指令具有完成程序异常,并且在从管理程序返回时,安全接口控件完成指令并强制程序中断。

[0104] 在安全VM拦截管理程序的时间和管理程序响应的时间之间,安全接口控件阻止管理程序将同一VM分派到任何物理CPU。它还阻止管理程序将中断(IO、外部或机器检查)或异常(例如程序异常)注入安全VM中,直到部分完成的指令被完全完成为止。

[0105] 根据本发明的一个或多个实施例,计算机服务器可以托管禁止管理程序访问存储器、寄存器和与安全VM相关联的其他数据的安全VM,而不必改变管理程序和/或安全VM代码/体系结构来拦截指令到管理程序和/或注入响应到安全VM。相反,根据本发明的一个或多个实施例,包括毫码的安全接口控件使用状态描述符的改进结构和存储装置/存储器的安全部分来促进保护数据。此外,安全接口控件存储来自VM状态的数据,使得在要使管理程序可访问一个或多个参数的情况下,安全接口控件(毫码)可以这样做。

[0106] 本发明的一个或多个实施例根植于计算机技术,尤其是虚拟机托管计算机服务器。此外,本发明的一个或多个实施例通过促进托管VM的计算机服务器托管安全VM来促进对计算技术本身的操作的改进,特别是虚拟机托管计算机服务器,其中甚至管理程序也被禁止访问与安全VM相关联的存储器、寄存器和其他此类数据。此外,本发明的一个或多个实施例通过使用安全接口控件来提供朝向对托管计算服务器的VM的改进的重要步骤,所述安全接口控件包括毫码以促进安全VM和管理程序的分离,并且因此维持由计算服务器托管的VM的安全性。安全接口控件提供轻量级中间操作以促进安全性,而不增加如本文所述的在VM的初始化/退出期间保护VM状态的实质开销。

[0107] 本发明可以是任何可能的技术细节集成级别的系统、方法和/或计算机程序产品。该计算机程序产品可以包括一个计算机可读存储介质(或多个计算机可读存储介质),其上具有计算机可读程序指令,用于使处理器执行本发明的各方面。

[0108] 计算机可读存储介质可以是有形设备,其可以保留和存储指令以供指令执行设备使用。计算机可读存储介质可以是例如但不限于电子存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或前述的任何合适组合。计算机可读存储介质的更具体示例的非详尽列表包括以下内容:便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式光盘只读存储器(CD-ROM)、数字通用光盘(DVD)、记忆棒、软盘、诸如在其上记录有指令的打孔卡或凹槽内凸起结构的机械编码装置、以及前述的任何合适的组合。这里使用的计算机可读存储介质不应被解释为瞬时信号本身,诸如无线电波或其它自由传播的电磁波、通过波导或其它传输介质传播的电磁波(例如,通过光纤电缆传递的光脉冲)或通过电线传输的电信号。

[0109] 本文描述的计算机可读程序指令可以从计算机可读存储介质下载到相应的计算/

处理设备,或者经由网络(例如,因特网,局域网,广域网和/或无线网络)下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光传输光纤、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配器卡或网络接口从网络接收计算机可读程序指令,并转发计算机可读程序指令以存储在相应计算/处理设备内的计算机可读存储介质中。

[0110] 用于执行本发明的操作的计算机可读程序指令可以是汇编指令、指令集架构(ISA)指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、集成电路配置数据、或者以一种或多种编程语言的任意组合编写的源代码或目标代码,所述编程语言包括诸如Smalltalk,C++等的面向对象的编程语言,以及诸如"C"编程语言或类似编程语言的过程编程语言。计算机可读程序指令可以完全在用户的计算机上执行、部分地在用户计算机上执行、作为独立的软件包执行、部分地在用户计算机上并且部分地在远程计算机上执行、或完全在远程计算机或服务器上执行。在后一种情况下,远程计算机可以通过任何类型的网络(包括局域网(LAN)或广域网(WAN))连接到用户的计算机,或者,可以连接到外部计算机(例如,利用互联网服务提供商来通过互联网连接)。在一些实施例中,包括例如可编程逻辑电路、现场可编程门阵列(FPGA)或可编程逻辑阵列(PLA)的电子电路可以通过利用计算机可读程序指令的状态信息来个性化定制电子电路,该电子电路执行计算机可读程序指令,以便执行本发明的各方面。

[0111] 本文参考根据本发明的实施例的方法、装置(系统)和计算机程序产品的流程图图示和/或框图来描述本发明的各方面。将理解,流程图图示和/或框图中的每个框以及流程图图示和/或框图中的框的组合可以由计算机可读程序指令实现。

[0112] 这些计算机可读程序指令可以被提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器来生产出机器,以使得通过计算机的处理器或其它可编程数据处理装置执行的指令创建用于实现流程图和/或一个框图块或多个框图块中所指定的功能/动作的装置。这些计算机可读程序指令还可以存储在计算机可读存储介质中,这些计算机可读程序指令可以使得计算机、可编程数据处理装置和/或其它设备以特定方式工作,以使得具有存储在其中的指令的计算机可读存储介质包括制品,该制品包括实现流程图和/或一个框图块或多个框图块中指定的功能/动作的各方面的指令。

[0113] 计算机可读程序指令还可以被加载到计算机,其它可编程数据处理装置或其它设备上,以使得在计算机、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,这样在计算机、其它可编程装置或其它设备上执行的指令实现在流程图和/或一个框图块或多个框图块中指定的功能/动作。

[0114] 附图中的流程图和框图示出根据本发明的各种实施例的系统,方法和计算机程序产品的可能实施方式的体系结构,功能和操作。在这方面,流程图或框图中的每个框可以表示模块、程序段或指令的一部分,其包括用于实现指定的逻辑功能的一个或多个可执行指令。在一些替代实施方式中,框中所标注的功能可以不按图中所示的顺序发生。例如,连续示出的两个框实际上可以基本上并行地执行,或者这些框有时可以以相反的顺序执行,这取决于所涉及的功能。还应注意,框图和/或流程图图示中的每个框以及框图和/或流程图图示中的框的组合可以由执行特定功能或动作,或执行专用硬件和计算机指令的组合的专用的基于硬件的系统来实现。

[0115] 已经出于说明的目的呈现了对本发明的各种实施例的描述,但其并非旨在是穷尽性的或限于所公开的实施例。在不背离所描述的实施例的范围和精神的情况下,许多修改和变化对于本领域的普通技术人员将是显而易见的。本文所使用的术语被选择为最好地解释实施例的原理、实际应用或对市场上存在的技术的技术改进,或使本领域的其他普通技术人员能够理解本文所公开的实施例。

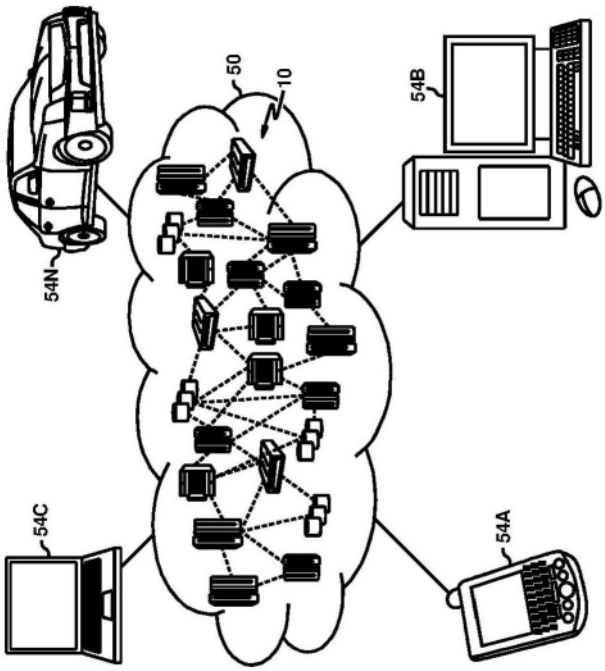

50
10
54B
54N
54C
54A

图1

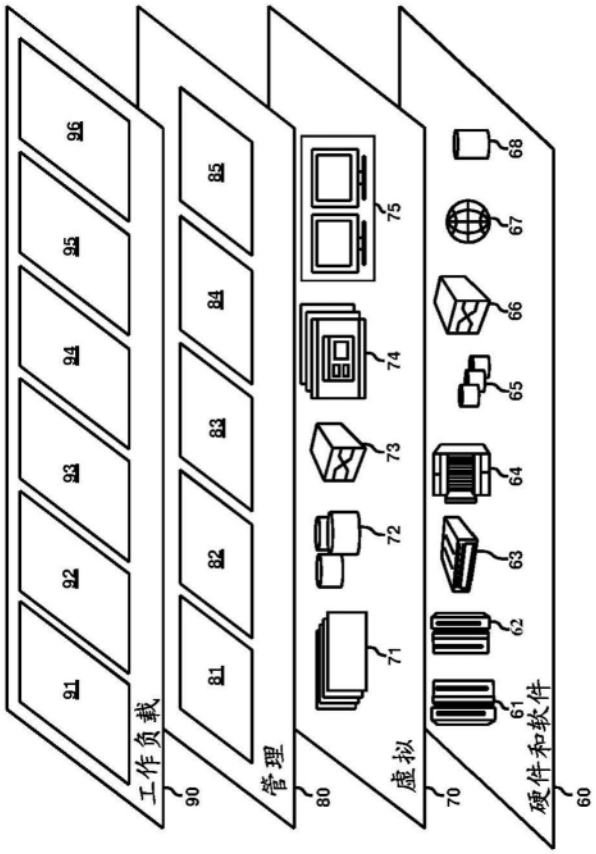

工作负载
90
91
92
93
94
95
96
管理
80
81
82
83
84
85
虚拟
70
71
72
73
74
75
硬件和软件
60
61
62
63
64
65
66
67
68

图2

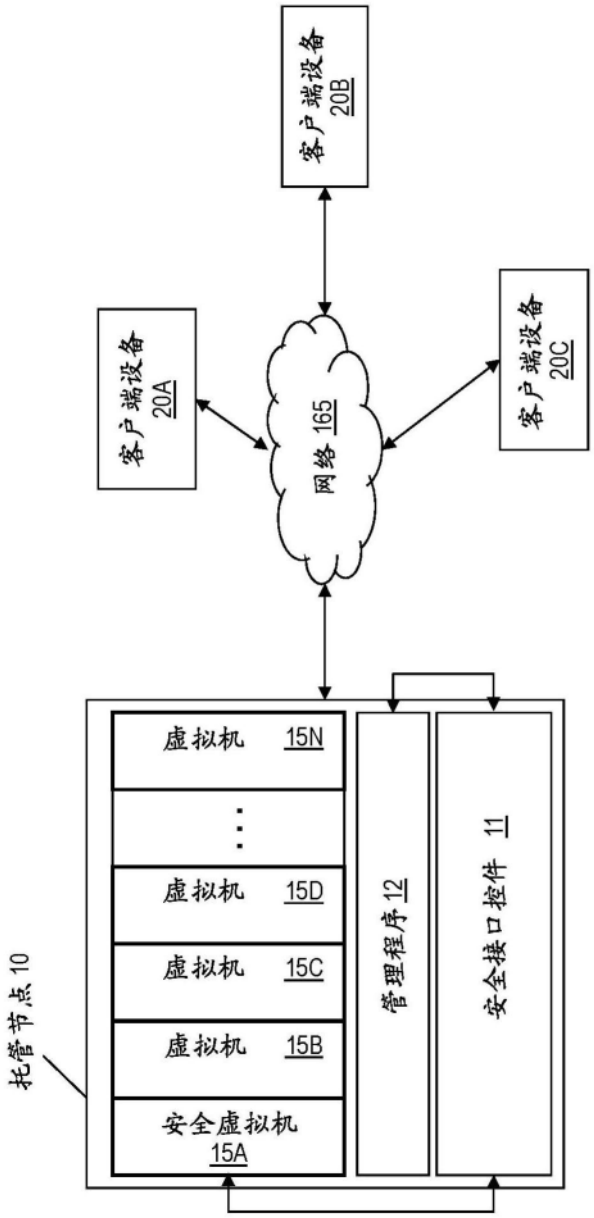

客户端设备 20B
客户端设备 20A
网络 165
客户端设备 20C
虚拟机 15N
虚拟机 15D
虚拟机 15C
虚拟机 15B
安全虚拟机 15A
管理程序 12
安全接口控件 11
托管节点 10

图3

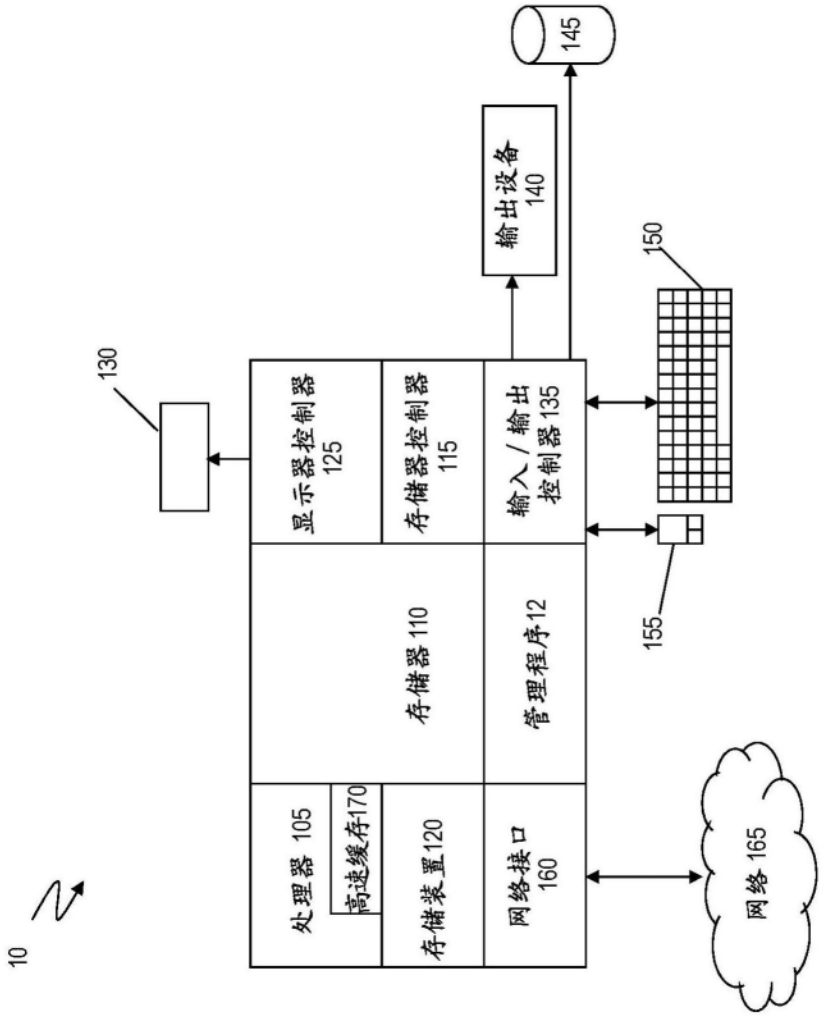

145
输出设备
140
150
130
显示器控制器
125
存储器控制器
115
输入/输出
控制器135
存储器110
管理程序12
155
处理器 105
高速缓存170
存储装置120
网络接口
160
网络 165
10

图4

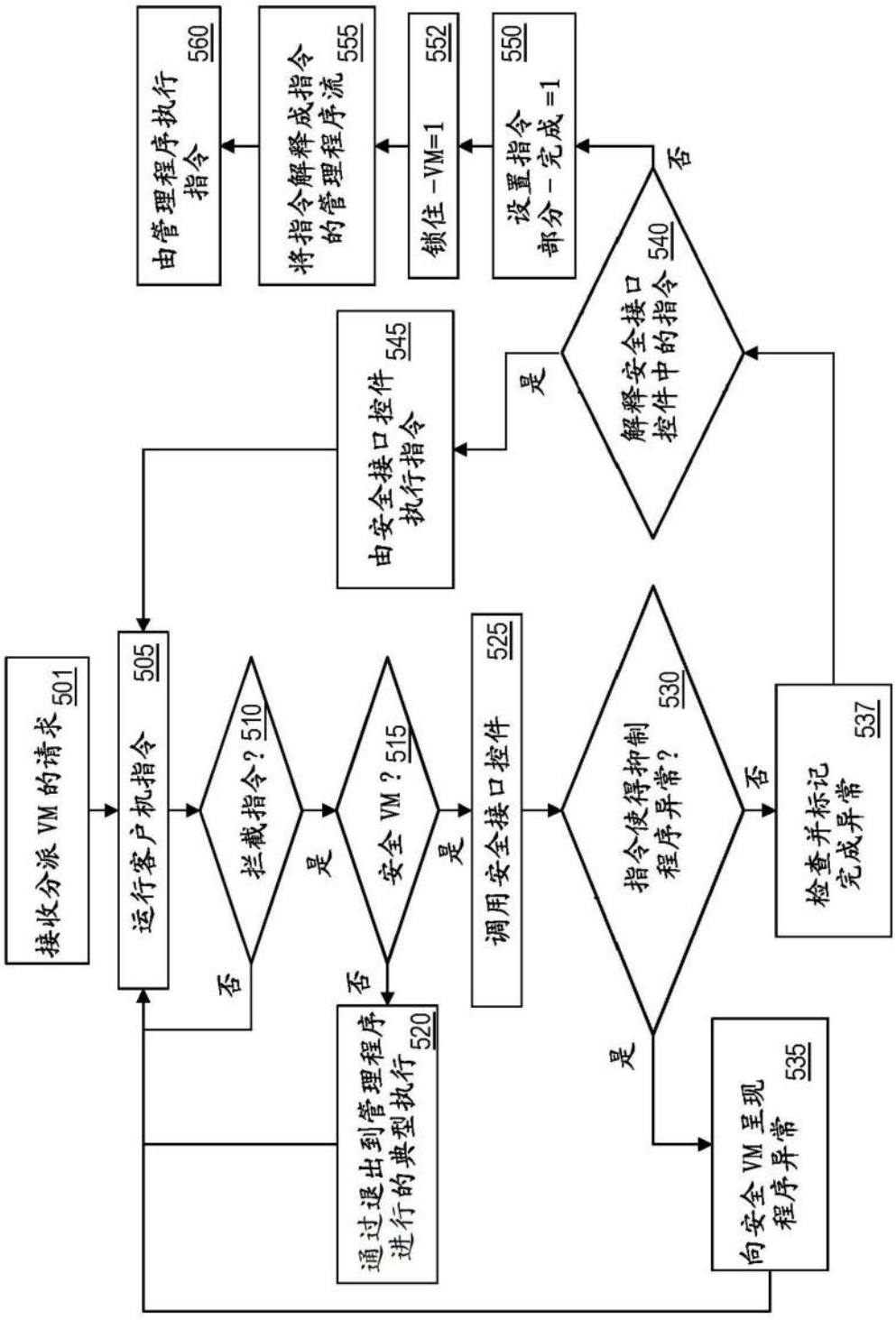

接收分派 VM 的请求 501
运行客户机指令 505
拦截指令？ 510
安全 VM？ 515
调用安全接口控件 525
指令使得抑制程序异常？ 530
检查并标记完成异常 537
解释安全接口控件中的指令 540
由安全接口控件执行指令 545
设置指令部分－完成 =1 550
锁住 –VM=1 552
将指令解释成指令的管理程序流 555
由管理程序执行指令 560
通过退出到管理程序进行的典型执行 520
向安全 VM 呈现程序异常 535
是
否

图5

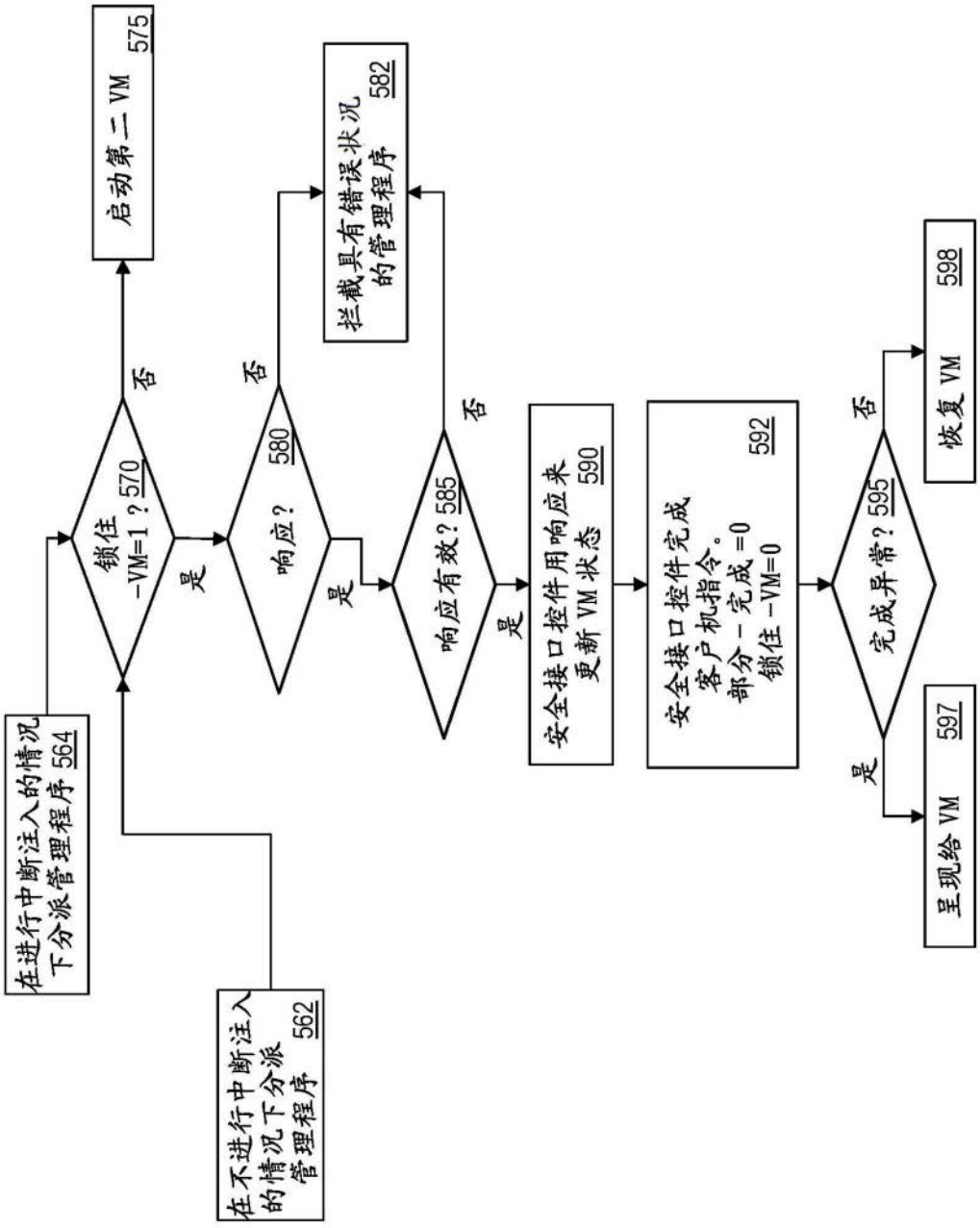

在进行中断注入的情况下分派管理程序 564
在不进行中断注入的情况下分派管理程序 562
锁住 -VM=1？570
否
是
启动第二 VM 575
响应？ 580
否
是
拦截具有错误状况的管理程序 582
响应有效？585
否
是
安全接口控件用响应来更新 VM 状态 590
安全接口控件完成客户机指令。部分－完成=0 锁住 -VM=0 592
完成异常？595
否
是
恢复 VM 598
呈现给 VM 597

图6