(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113544652 B
(45)授权公告日 2025.06.20

(21)申请号 202080018913.X
(22)申请日 2020.03.06
(65)同一申请的已公布的文献号
申请公布号 CN 113544652 A
(43)申请公布日 2021.10.22
(30)优先权数据
16/296,303 2019.03.08 US
(85)PCT国际申请进入国家阶段日
2021.09.06
(86)PCT国际申请的申请数据
PCT/IB2020/051941 2020.03.06
(87)PCT国际申请的公布数据
WO2020/183308 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 J·布拉德伯里 C·博恩特雷格 H·卡斯滕斯 M·施维德夫斯基 R·宾德根
(74)专利代理机构 北京市中咨律师事务所 11247
专利代理师 于静 刘薇

(51)Int.Cl.
G06F 12/00(2006.01)
(56)对比文件
CN 107346401 A,2017.11.14
US 2007106986 A1,2007.05.10
审查员 董立波

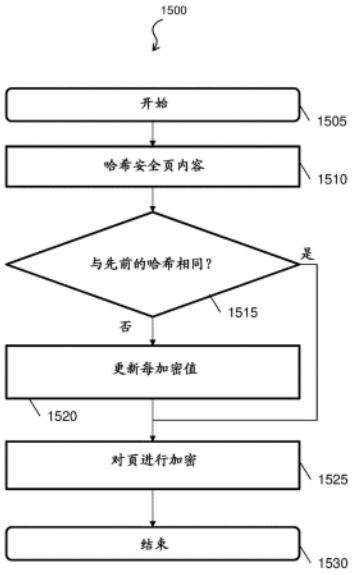
权利要求书3页 说明书22页 附图20页

(54)发明名称

具有页改变检测的安全分页

(57)摘要

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计算计算机系统的内存页的哈希值并将所述哈希值与所述页的先前计算的哈希值进行比较。基于确定所述哈希值与所述先前计算的哈希值匹配,可以用每页的每加密值对所述页进行加密。基于确定所述哈希值与所述先前计算的哈希值不匹配,可以用所述每页的每加密值的修改值对所述页进行加密。



CN 113544652 B

1.一种计算机实现的方法,包括:

计算计算机系统的内存页的哈希值;

将所述哈希值与所述页的先前计算的哈希值进行比较;

基于确定所述哈希值与所述先前计算的哈希值匹配,用每页的每加密值对所述页进行加密;和

基于确定所述哈希值与所述先前计算的哈希值不匹配,用所述每页的每加密值的修改值对所述页进行加密。

2.根据权利要求1所述的方法,其中,由安全接口控件响应于主机的将所述页从安全页转换成非安全页的请求而执行所述加密。

3.根据权利要求2所述的方法,进一步包括:

将加密的所述非安全页提供给所述主机用于存储。

4.根据权利要求2所述的方法,进一步包括:

将所述哈希值存储在所述安全接口控件的安全表中。

5.根据权利要求4所述的方法,进一步包括:

将所述非安全页转换成所述安全页;

由所述安全接口控件基于与所述安全页相关联的所述每加密值对所述安全页进行解密以产生解密的页;

计算所述解密的页的哈希值;

将所述解密的页的哈希值与存储在所述安全表中的所述页的哈希值进行比较;和

基于确定所述解密的页的哈希值与存储在所述安全表中的所述哈希值匹配来验证所述解密的页。

6.根据权利要求2所述的方法,其中,所述安全接口控件包括固件、硬件或固件与硬件的组合;所述安全页被分配给由所述主机管理的安全容器或安全虚拟机;所述主机为虚拟机监控程序或操作系统。

7.根据权利要求1至6中任一项所述的方法,其中,对所述页进行加密体现了用密码安全哈希函数对与所述页相关联的地址值、一个或多个随机值和所述每加密值的组合。

8.根据权利要求1至6中任一项所述的方法,其中,在使用所述每加密值之前确立所述每加密值的初始值。

9.一种计算机系统,包括:

内存;

处理单元;和

安全接口控件,与处理单元和内存对接,所述安全接口控件被配置为执行包含以下的操作:

计算内存页的哈希值;

将所述哈希值与所述页的先前计算的哈希值进行比较;

基于确定所述哈希值与所述先前计算的哈希值匹配,用每页的每加密值对所述页进行加密;和

基于确定所述哈希值与所述先前计算的哈希值不匹配,用所述每页的每加密值的修改值对所述页进行加密。

10.根据权利要求9所述的计算机系统,其中,由所述安全接口控件响应于主机的将所述页从安全页转换成非安全页的请求而执行所述加密。

11.根据权利要求10所述的计算机系统,其中,所述操作进一步包括将加密的所述非安全页提供给所述主机用于存储。

12.根据权利要求10所述的计算机系统,其中,所述操作进一步包括将所述哈希值存储在所述安全接口控件的安全表中。

13.根据权利要求12所述的计算机系统,其中,所述操作进一步包括:

将所述非安全页转换成所述安全页;

由所述安全接口控件基于与所述安全页相关联的所述每加密值对所述安全页进行解密以产生解密的页;

计算所述解密的页的哈希值;

将所述解密的页的哈希值与存储在所述安全表中的所述页的哈希值进行比较;和

基于确定所述解密的页的哈希值与存储在所述安全表中的所述哈希值匹配来验证所述解密的页。

14.根据权利要求10所述的计算机系统,其中,所述安全接口控件包括固件、硬件或固件与硬件的组合;所述安全页被分配给由所述主机管理的安全容器或安全虚拟机;所述主机为虚拟机监控程序或操作系统。

15.根据权利要求9至14中任一项所述的计算机系统,其中,对所述页进行加密体现了用密码安全哈希函数对与所述页相关联的地址值、一个或多个随机值和所述每加密值的组合。

16.根据权利要求9至14中任一项所述的计算机系统,其中,在使用所述每加密值之前确立所述每加密值的初始值。

17.一种计算机程序产品,所述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被执行时执行一种方法,所述方法包括:

计算计算机系统的内存页的哈希值;

将所述哈希值与所述页的先前计算的哈希值进行比较;

基于确定所述哈希值与所述先前计算的哈希值匹配,用每页的每加密值对所述页进行加密;和

基于确定所述哈希值与所述先前计算的哈希值不匹配,用所述每页的每加密值的修改值对所述页进行加密。

18.根据权利要求17所述的计算机程序产品,其中,由安全接口控件响应于主机的将所述页从安全页转换成非安全页的请求而执行所述加密。

19.根据权利要求18所述的计算机程序产品,其中,所述可执行指令被执行时进一步执行所述方法,包括:

将加密的所述非安全页提供给所述主机用于存储。

20.根据权利要求18所述的计算机程序产品,所述可执行指令被执行时进一步执行所述方法,包括:

将所述哈希值存储在所述安全接口控件的安全表中。

21.根据权利要求20所述的计算机程序产品,其中,所述可执行指令被执行时进一步执

行所述方法,包括:

将所述非安全页转换成所述安全页;

由所述安全接口控件基于与所述安全页相关联的所述每加密值对所述安全页进行解密以产生解密的页;

计算所述解密的页的哈希值;

将所述解密的页的哈希值与存储在所述安全表中的所述页的哈希值进行比较;和

基于确定所述解密的页的哈希值与存储在所述安全表中的所述哈希值匹配来验证所述解密的页。

22.根据权利要求21所述的计算机程序产品,其中,所述安全页被分配给由所述主机管理的安全容器或安全虚拟机,并且所述主机为虚拟机监控程序或操作系统。

23.根据权利要求17至22中任一项所述的计算机程序产品,其中,对所述页进行加密体现了用密码安全哈希函数对与所述页相关联的地址值、一个或多个随机值和所述每加密值的组合。

24.根据权利要求17至22中任一项所述的计算机程序产品,其中,在使用所述每加密值之前确立所述每加密值的初始值。

具有页改变检测的安全分页

背景技术

[0001] 本发明总体上涉及计算机技术,更具体地涉及具有页改变检测的安全分页(paging)。

[0002] 云计算和云存储为用户提供在第三方数据中心中存储和处理他们的数据的能力。云计算促进快速且容易地向客户机供应VM的能力,而不需要客户机购买硬件或为物理服务器提供地面空间。客户机可以根据客户机的变化的偏好或要求容易地扩展或收缩VM。通常,云计算提供商提供物理地驻留在提供商的数据中心处的服务器上的VM。客户通常关心VM中的数据的安全性,特别是因为计算提供者通常在同一服务器上存储多于一个客户机的数据。客户机可能期望他们自己的代码/数据与云计算提供者的代码/数据之间的安全性,以及他们自己的代码/数据与在提供者的站点处运行的其他VM的代码/数据之间的安全性。此外,客户可能期望来自提供者的管理员的安全性以及防止来自机器上运行的其他代码的潜在安全漏洞。

[0003] 为了处理这样的敏感情况,云服务提供商可以实施安全控制以确保适当的数据隔离和逻辑存储隔离。虚拟化在实施云基础架构中的广泛使用导致云服务客户的独特安全顾虑,因为虚拟化改变了操作系统(OS)与底层硬件(无论是计算、存储或甚至联网硬件)之间的关系。这引入了作为本身必须正确地得到配置、管理和保护的附加层的虚拟化。

[0004] 通常,在主机虚拟机监控程序(host hypervisor)的控制下作为客户机(guest)运行的VM依赖于虚拟机监控程序透明地为该客户机提供虚拟化服务。这些服务包括内存管理(memory management)、指令模拟和中断处理。

[0005] 在内存管理的情况下,VM可以将其数据从磁盘移出(分页调入(page-in))以驻留在内存中,并且VM还可以将其数据移回(分页调出(page-out))到磁盘。当页驻留在内存中时,VM(客户机)使用动态地址转换(DAT)来将内存页从客户机虚拟地址(virtual address)映射到客户机绝对地址(absolute address)。此外,主机虚拟机监控程序有其自己的针对客户机内存页(guest pages in memory)的DAT映射(从主机虚拟地址到主机绝对地址),并且其可独立地且对客户机透明地将客户机页(guest pages)分页调入、调出内存。虚拟机监控程序正是通过主机DAT表来提供内存隔离或客户机内存在两个单独客户机VM之间的共享。主机必要时还能代表客户机访问客户机内存,以模拟客户机操作。

发明内容

[0006] 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:计算计算机系统的内存页的哈希值并将所述哈希值与所述页的先前计算的哈希值进行比较。基于确定所述哈希值与所述先前计算的哈希值匹配,可以用每页的每加密值(per-encryption value per page)对所述页进行加密。基于确定所述哈希值与所述先前计算的哈希值不匹配,可以用所述每页的每加密值的修改值对所述页进行加密。优点可以包括通过引入每加密值而增强加密,同时在底层数据没有发生变化的情况下限制加密更新。

[0007] 根据本发明的额外或替代实施例,所述加密可以安全控制接口响应于主机的将所

述页从安全(secure)页转换成非安全(non-secure)页的请求而执行。优点可包括通过在使页非安全并且可由不可信实体(untrusted entity)访问之前的加密,防止对底层数据的直接访问。

[0008] 根据本发明的额外或替代实施例,可以将加密的所述非安全页提供给所述主机用于存储。优点可以包括实现针对由可能是不可信的实体进行的内存页的输入/输出访问和移动的内存管理。

[0009] 根据本发明的额外或替代实施例,可以将所述哈希值存储在安全控制接口的安全表中。优点可包括保存用于变化比较的汇总值。

[0010] 根据本发明的额外或替代实施例,可以将将所述非安全页转换成所述安全页。所述安全接口控件可以基于与所述安全页相关联的所述每加密值对所述页进行解密以产生解密的页。可以计算所述解密的页的哈希值。可以将所述解密的页的哈希值与存储在所述安全表中的所述页的哈希值进行比较。基于确定所述解密的页的哈希值与存储在所述安全表中的所述哈希值匹配来验证所述解密的页。优点可包括验证向不可信实体提供的页未被修改。

[0011] 根据本发明的额外或替代实施例,所述安全接口控件可以包括固件、硬件或固件与硬件的组合。可以所述安全页被分配给由所述主机管理的安全容器或安全虚拟机。主机可以是虚拟机监控程序或操作系统。优点可包括使用加密将来自安全实体的安全页与不可信实体共享。

[0012] 根据本发明的额外或替代实施例,加密所述页可以体现了用密码安全哈希函数对与所述页相关联的地址值、一个或多个随机值和所述每加密值的组合。优点可以包括增加加密的复杂性以增强安全性。

[0013] 本发明的其他实施例在计算机系统和计算机程序产品中实现上述方法的特征。

[0014] 通过本公开的技术实现了额外特征和优点。本发明的其他实施例和方面在此作了详细描述并且被认为是本发明的一部分。为了更好地理解本发明的优点和特征,参考说明书和附图。

附图说明

[0015] 在说明书结尾处的权利要求书中特别指出并清楚地要求保护本文描述的独占权利的细节。从以下结合附图的详细描述中,本发明的实施例的前述和其他特征和优点是显而易见的,附图中:

[0016] 图1描绘了根据本发明的一个或多个实施例的分区安全表;

[0017] 图2描绘了根据本发明的一个或多个实施例的用于执行DAT的虚拟地址空间和绝对地址空间;

[0018] 图3描绘了根据本发明的一个或多个实施例的用于支持在虚拟机监控程序下运行的虚拟机(VM)的嵌套的多部分DAT;

[0019] 图4描绘了根据本发明的一个或多个实施例的安全客户机存储的映射;

[0020] 图5描绘了根据本发明的一个或多个实施例的动态地址转换(DAT)操作的系统示意图;

[0021] 图6描绘了根据本发明的一个或多个实施例的安全接口控件内存的系统示意图;

[0022] 图7描绘了根据本发明的一个或多个实施例的导入操作的过程流;

[0023] 图8描绘了根据本发明的一个或多个实施例的导入操作的过程流;

[0024] 图9描绘了根据本发明的一个或多个实施例的捐赠的内存操作的过程;

[0025] 图10描绘了根据本发明的一个或多个实施例的非安全虚拟机监控程序页到安全接口控件的安全页的转换的过程流;

[0026] 图11描绘了根据本发明的一个或多个实施例的由安全接口控件进行的安全存储访问的过程流;

[0027] 图12描绘了根据本发明的一个或多个实施例的由安全接口控件和由硬件进行访问标记的过程流;

[0028] 图13描绘了根据本发明的一个或多个实施例的用于支持程序和安全接口控件的安全和非安全访问的转换的过程流;

[0029] 图14描绘了根据本发明的一个或多个实施例的具有由程序和由安全接口控件进行的安全存储保护的DAT的过程流;

[0030] 图15示出根据本发明一个或多个实施例的用于确定在加密之前页内容是否已改变的过程流;

[0031] 图16描绘了根据本发明的一个或多个实施例的用于由安全接口控件进行加密和解密控制的数据流;

[0032] 图17示出了根据本发明一个或多个实施例的云计算环境;

[0033] 图18描绘了根据本发明的一个或多个实施例的抽象模型层;

[0034] 图19描绘了根据本发明的一个或多个实施例的系统;和

[0035] 图20描绘了根据本发明的一个或多个实施例的处理系统。

[0036] 本文所描绘的图是说明性的。在不背离本发明的精神的情况下,可以对本文所描述的图或操作进行许多变化。例如,可以按不同次序执行动作,或者可以添加、删除或修改动作。同样,术语“耦合”及其变体描述了在两个元件之间具有通信路径并且不暗示这些元件之间没有中间元件/连接的直接连接。所有这些变化被视为说明书的一部分。

具体实施方式

[0037] 本发明的一个或多个实施例使用计数器来执行内存页的加密,该计数器基于是否检测到底层数据的改变来修改加密过程。可以管理安全存储以确保安全域内的数据页是非安全实体不可访问的。在将安全域中的数据传送至非安全域(如用于存储至磁盘)时可以在传送至非安全域之前对数据进行加密。为了增强安全性,在加密数据页时,可以将多个项目组合到初始化向量中,诸如在检测到数据改变时递增的计数器值。因此,如果数据页保持不变,则所述加密将匹配加密数据的其他先前副本,从而保留现有的分页算法。然而,如果底层数据已经改变,则可以修改每加密计数器(per-encryption counter)的每加密值(per-encryption value)并且将其用作加密过程的一部分。引入每加密值可使得更难以检测加密数据中的样式(patterns),因为加密是基于数据值改变和每加密值改变的组合而改变的。只要底层内容保持不变,则每加密计数器与页改变检测的结合可以支持多次重复使用被分页调出的加密内容,这对于增加只读内存页的处理系统效率可以是有用的。相反,如果不使用页改变检测并且例如每次访问页时就修改每加密计数器,则即使底层数据未改变,

也不能多次重复使用被分页调出的页。

[0038] 本发明的实施例是参考支持虚拟机、容器、虚拟机监控程序、操作系统和其他计算机系统元件的分页(paging)来描述的。将理解,本文中所描述的过程可适用于众多系统配置而非限定于本文中更详细地描述的实例系统。

[0039] 本发明的一个或多个实施例利用软件与机器之间的高效、轻量的安全接口控件来提供额外的安全性。

[0040] 在主机虚拟机监控程序的控制下作为客户机运行的虚拟机(VM)依赖于该虚拟机监控程序来透明地为该客户机提供虚拟化服务。这些服务可以应用于安全实体和另一不可信实体之间的传统上允许此其他实体访问安全资源的任何接口。如前所述,这些服务可包括但不限于内存管理、指令仿真和中断处理。例如,对于中断和异常注入,虚拟机监控程序通常对客户机的前缀区(prefix area)(低核)进行读和/或写。如在此使用的术语“虚拟机”或“VM”是指物理机器(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。VM被维护为在底层主机(物理处理器或处理器集合)上执行的软件。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。如本文中所使用的术语“虚拟机监控程序”和“VM监测器(VMM)”是指管理并准许多个VM在同一主机上使用多个(并且有时不同的)OS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(或开启)过程。在另一示例中,部署VM包括VM的激活(或开始)过程(例如,在VM先前被安装或已经存在的情况下)。

[0041] 为了促进和支持安全客户机,存在技术挑战,其中在虚拟机监控程序和安全客户机之间需要额外的安全性而不依赖于虚拟机监控程序,使得虚拟机监控程序不能访问来自VM的数据,并且因此不能以上述方式提供服务。

[0042] 本文描述的安全执行提供硬件机制以确保安全存储和非安全存储之间以及属于不同安全用户的安全存储之间的隔离。为了安全客户机,在“不可信的”(untrusted)非安全虚拟机监控程序与安全客户机之间提供了额外的安全性。为此,虚拟机监控程序通常代表客户机所做的许多功能需要被结合到机器中。在此描述了一种新的安全接口控件(secure interface control)(在此也被称为“UV”),用于提供虚拟机监控程序与安全客户机之间的安全接口。术语安全接口控件和UV在本文中可互换使用。安全接口控件与硬件协作工作以提供该额外的安全性。此外,较低层虚拟机监控程序可以为该不可信虚拟机监控程序提供虚拟化,并且如果该较低层虚拟机监控程序是以可信代码实现的,则它也可以是安全接口控件的一部分。

[0043] 在一个示例中,安全接口控件在内部、安全和可信的硬件和/或固件中实现。该可信固件可以包括例如处理器毫码(millicode)或PR/SM逻辑分区代码。对于安全客户机或实体,安全接口控件提供安全环境的初始化和维护以及协调这些安全实体在硬件上的调度(dispatch)。当安全客户机主动使用数据并且其驻留在主机存储中时,其在安全存储中保持“安全”(in the clear)。安全客户机存储可以由该单个安全客户机访问——这由硬件严格地强制执行。即,硬件防止任何非安全实体(包括虚拟机监控程序或其他非安全客户机)或不同的安全客户机访问该数据。在该示例中,安全接口控件作为固件的最低级别的可信部分运行。最低层或毫码实际上是硬件的扩展,并且用于实现例如在IBM的zArchitecture®中定义的复杂指令和功能。毫码能够访问存储的所有部分,在安全执行的上下文中,其包括

其自身的安全UV存储、非安全虚拟机监控程序存储、安全客户机存储和共享存储。这允许它提供安全客户机所需的或虚拟机监控程序支持该客户机所需的任何功能。安全接口控件还具有对硬件的直接访问,这允许硬件在由安全接口控件确立的条件的控制下有效地提供安全检查。

[0044] 根据本发明的一个或多个实施例,软件使用UV调用(UV Call)(UVC)指令来请求安全接口控件执行特定动作。例如,UVC指令可由虚拟机监控程序使用以初始化安全接口控件、创建安全客户机域(例如,安全客户机配置)和在该安全配置内创建虚拟CPU。它也可以用于导入(import)(解密并分配给安全客户机域)和导出(export)(加密并允许主机访问)安全客户机页,作为虚拟机监控程序分页调入或分页调出操作的一部分。此外,安全客户机具有定义与虚拟机监控程序共享的存储、使安全存储被共享、以及使共享存储安全的能力。

[0045] 这些UVC命令可由机器固件与许多其他架构化指令类似地执行。机器不进入安全接口控件模式,但是机器却在其当前运行的模式中执行安全接口控件功能。硬件维护固件状态和软件状态两者,因此没有上下文的切换就能处理这些操作。这种低开销允许软件、可信固件和硬件的不同层之间以最小化和降低安全接口控件的复杂度、同时仍提供必要的安全级别的方式紧密绑定协作。

[0046] 根据本发明的一个或多个实施例,为了支持安全接口控件和硬件所需的控制块结构以正确维护安全客户机和支持虚拟机监控程序环境,虚拟机监控程序在初始化安全客户机环境的同时向安全接口控件捐赠(donate)存储。因此,为了准备1)初始化运行安全客户机的分区(zone)、2)创建安全客户机域(domain)和3)创建在每个域中运行的安全CPU,虚拟机监控程序发布查询UVC指令以确定捐赠所需的存储量等。一旦存储已经被捐赠,它就被标记为安全并且被注册为属于安全接口控件;并且,任何非安全或安全客户机实体的访问都被禁止。在相关实体(例如,安全客户机CPU、安全客户机域或分区)瓦解之前,这种情况一直存在。

[0047] 在一个实例中,用于支持特定于UV控制块的UV存储的第一区段(section)作为初始化UVC的一部分被捐赠给安全接口控件,并且驻留在本文中所谓的UV2存储中。用于支持基础和可变安全-客户机-配置控制块(对于每个安全客户机域)的UV存储的第二和第三区段作为创建-安全-客户机-配置UVC的一部分被捐赠,并且分别驻留在UVS和UVV存储中。用于支持安全-CPU控制块的UV存储的第四和最后的区段也驻留在UVS空间中,并且作为创建-安全-客户机-CPU(create-secure-guest-CPU)UVC的一部分而被捐赠。在捐赠这些区(area)中的每一个时,安全控制接口将它们标记为安全(以防止它们被任何非安全实体访问)并且还将它们在分区安全表(zone-security table)中注册为属于安全接口控件(以防止它们被任何安全客户机实体访问)。为了在UV空间内提供进一步的隔离,UV2空间(其不与任何特定的安全-客户机域相关联)也标记有唯一UV2安全域,而UVS和UVV空间两者都进一步标记有相关联的特定的安全-客户机域。在此实例中,UVV空间驻留在主机虚拟空间中,因此可以用主机虚拟到主机绝对映射来进一步识别。

[0048] 尽管安全接口控件可以访问所有存储(非安全存储、安全客户机存储和UV存储),本发明的一个或多个实施例提供允许安全接口控件非常具体地访问UV存储的机制。使用在安全客户机域之间提供隔离的相同硬件机制,本发明的实施例可以在UV存储内提供类似的隔离。这保证了安全接口控件仅在预期且指定时才访问UV存储;仅仅访问期望的指定安全

客户机的安全客户机存储;以及仅在指定时访问非安全存储。即,安全接口控件可以非常明确地指定其意图访问的存储,使得硬件可以保证其确实访问该存储。此外,可以进一步指定其仅打算访问与指定的安全客户机域相关联的UV存储。

[0049] 为了提供安全性,当虚拟机监控程序透明地将安全客户机数据分页调入和分页调出时,与硬件协作的安全接口控件提供和保证数据的解密和加密。为了实现这一点,当将安全客户机数据分页调入和分页调出时,需要虚拟机监控程序发布新的UVC。基于由安全接口控件在这些新UVC期间设置的控制,硬件将保证这些UVC确实由虚拟机监控程序发布。

[0050] 在这个新的安全环境中,每当虚拟机监控程序分页调出安全页时,需要从安全存储(导出)UVC发布新的转换。响应于此导出UVC,安全接口控件将1)指示该页被UV“锁定”,2)加密该页,3)将该页设置为非安全的,以及4)重置UV锁。一旦导出UVC完成,虚拟机监控程序现在就可以分页调出加密的客户机页。

[0051] 另外,每当虚拟机监控程序正分页调入安全页时,其必须发布新的转换到安全存储(导入)UVC。响应于此导入UVC,UV或安全接口控件将1)在硬件中将该页标记为安全,2)指示该页被UV“锁定”,3)解密该页,4)设置对特定安全客户机域的权限,以及5)重置UV锁。每当安全实体进行访问时,硬件在转换期间对该页执行授权检查(authorization checks)。这些检查包括:1)验证该页确实属于访问它的安全客户机域的的检查;以及2)确保当该页已经驻留在客户机存储器中时虚拟机监控程序没有改变此页的主机映射(host mapping)的检查。一旦某页被标记为安全,硬件就阻止虚拟机监控程序或非安全客户机VM访问任何安全页。这些额外转换步骤防止另一安全VM的访问且防止虚拟机监控程序的重新映射(remapping)。

[0052] 现在转到图1,根据本发明的一个或多个实施例,总体上示出了用于分区安全的表100。图1中所示的分区安全表100由安全接口控件来维护,并且由安全接口控件和硬件用来保证对由安全实体访问的任何页的安全访问。分区安全表100按主机绝对地址110进行索引。即,对于主机绝对存储的每个页有一个条目。每个条目包括用于验证该条目属于进行访问的安全实体的信息。

[0053] 进一步地,如图1所示,分区安全表100包括安全域ID 120(标识与此页相关联的安全域);UV-位130(指示该页被捐献给安全接口控件并且由安全接口控件拥有);禁用地址比较(DA)-位140(用于在某些情况下,诸如当被定义为主机绝对地址的安全接口控件页不具有相关联的主机虚拟地址时,禁用主机地址对比较);共享(SH)-位150(指示与非安全虚拟机监控程序共享的页)和主机虚拟地址160(指示为该主机绝对地址注册的主机虚拟地址,其被称为主机-地址对)。注意,主机-地址对指示主机绝对和相关联的、注册的主机虚拟地址。一旦由虚拟机监控程序导入,主机-地址对表示该页的映射,并且,所述比较保证在客户机使用该页时主机不重新映射该页。

[0054] 动态地址转换(DAT)用于将虚拟存储映射到实存储。当客户机VM在虚拟机监控程序的控制下作为可分页客户机运行时,客户机使用DAT来管理驻留在其内存中的页。此外,当页驻留在其内存中时,主机独立地使用DAT来管理那些客户机页(连同其自己的页)。虚拟机监控程序使用DAT来提供不同VM之间的存储的隔离和/或共享以及防止客户机访问虚拟机监控程序存储。当客户机以非安全模式运行时,虚拟机监控程序可以访问所有客户机的存储。

[0055] DAT使得能够将一个应用与另一应用隔离,同时仍允许它们共享公共资源。DAT也允许实现可用于设计和测试OS的新版本以及应用程序的并发处理的VM。虚拟地址标识虚拟存储中的位置。地址空间是虚拟地址的连续序列,连同特定变换参数(包含DAT表)所述特定变换参数允许将每一虚拟地址转换成相关联的绝对地址,绝对地址以存储中的字节位置标识该地址。

[0056] DAT使用多表查找来将虚拟地址转换为相关联的绝对地址。这个表结构通常由存储管理器定义和维护。这个存储管理器通过分页调出例如一个页以引入另一个页来在多个程序之间透明地共享绝对存储。当页被分页调出时,存储管理器将例如在相关联的页表中设置无效位。当程序试图访问被分页调出的页时,硬件将向存储管理器提交通常被称为页错误的程序中断。作为响应,存储管理器将分页调入所请求的页并重置无效位。这都是对程序透明地完成的,并且允许存储管理器虚拟化存储并在各种不同用户之间共享它。

[0057] CPU用虚拟地址访问主存储时,首先通过DAT将虚拟地址转换成实地址(real address),然后通过前缀(prefixing)将其转换成绝对地址。用于特定地址空间的最高级表的名称(designation)(原点和长度)被称为地址空间控制元素(ASCE),并定义相关联的地址空间。

[0058] 现在转到图2,根据本发明的一个或多个实施例,总体上示出了用于执行DAT的示例虚拟地址空间202和204以及绝对地址空间206。在图2所示的示例中,存在两个虚拟地址空间:虚拟地址空间202(由地址空间控制元素(ASCE)A 208定义)和虚拟地址空间204(由ASCE B 210定义)。虚拟页A1.V 212a1、A2.V 212a2和A3.V 212a3被存储管理器用ASCE A208在多表(段(segment)230和页表232a、232b)查找中映射到绝对页A1.A 220a1、A2.A 220a2和A3.A 220a3。类似地,使用ASCE B210在两表234和236查找中分别将虚拟页B1.V 214b1和B2.V 214b2映射到绝对页B1.A 222b1和B2.A 222b2。

[0059] 现在转到图3,根据本发明的一个或多个实施例,总体上示出了用于支持在虚拟机监控程序下运行的VM的嵌套的多部分DAT转换的示例。在图3所示的示例中,客户机A的虚拟地址空间A302(由客户机ASCE(GASCE)A304定义)和客户机B的虚拟地址空间B 306(由GASCEB 308定义)两者都驻留在共享主机(虚拟机监控程序)虚拟地址空间325中。如图所示,属于客户机A的虚拟页A1.GV 310a1、A2.GV 310a2和A3.GV 310a3分别被客户机A的存储管理器用GASCEA304映射到客户机绝对页A1.HV 340a1、A2.HV 340a2和A3.HV 340a3;属于客户机B的虚拟页B1.GV 320b1和B2.GV 320b2分别被客户机B的存储管理器独立地用GASCEB 308映射到客户机绝对页B1.HV 360b1和B2.HV 360b2。在此示例中,这些客户机绝对页直接映射到共享的主机虚拟地址空间325中,随后经历到主机绝对地址空间330的额外主机DAT转换。如图所示,主机虚拟地址A1.HV 340a1、A3.HV 340a3和B1.HV 360b1被主机存储管理器用主机ASCE(HASCE)350映射到A1.HA 370a1、A3.HA 370a3和B1.HA 370b1。属于客户机A的主机虚拟地址A2.HV 340a2和属于客户机B的B2.HV 360b2两者都被映射到相同的主机绝对页AB2.HA 380。这使得数据能够在这两个客户机之间共享。在客户机DAT转换期间,每个客户机表地址都被视为客户机绝对地址,并且经历额外的嵌套主机DAT转换。

[0060] 本文所述的本发明的实施例提供安全客户机和UV存储保护。非安全客户机和虚拟机监控程序对安全存储的访问被禁止。虚拟机监控程序规定,对于给定的驻留安全客户机页,发生以下情况。相关联的主机绝对地址仅是通过单个虚拟机监控程序(主机)DAT映射可

访问的。即,存在映射到分配给安全客户机的任何给定主机绝对地址的单个主机虚拟地址。与给定安全客户机页相关联的虚拟机监控程序DAT映射(主机虚拟到主机绝对)在它被分页调入时不改变。针对单个安全客户机映射与安全客户机页相关联的主机绝对页。

[0061] 根据本发明的一个或多个实施例,安全客户机之间的存储共享也被禁止。存储是在单个安全客户机与该安全客户机控制下的虚拟机监控程序之间共享的。UV存储是安全存储并且是安全接口控件而非客户机/主机可访问的。虚拟机监控程序将存储分配给接口控件。根据本发明的一个或多个实施例,硬件和安全接口控件禁止任何试图违反这些规则的行为。

[0062] 现在转到图4,根据本发明的一个或多个实施例,总体上示出了安全客户机存储的映射的示例。图4类似于图3,只是图4的示例不允许在安全客户机A和安全客户机B之间共享存储。在图3的非安全示例中,属于客户机A的主机虚拟地址A2.HV 340a2和属于客户机B的B2.HV 360b2两者被映射到相同的主机绝对页AB2.HA 380。在图4的安全客户机存储示例中,属于客户机A的主机虚拟地址A2.HV 340a2映射到主机绝对地址A2.HA 490a,而属于客户机B的B2.HV 360b2映射到其自己的B2.HA 490b。在这个实例中,不存在安全客户机之间的共享。

[0063] 当安全客户机页驻留在盘上时,其被加密。当虚拟机监控程序分页调入安全客户机页时,虚拟机监控程序发布UV调用(UVC),UV调用使安全接口控件将该页标记为安全(除非共享)、解密该页(除非共享)并且将该页注册(在分区安全表中)为属于适当的安全客户机(例如,客户机A)。此外,虚拟机监控程序将相关联的主机虚拟地址(例如,A3.HV 340a3)注册到该主机绝对页(称为主机-地址对)。如果虚拟机监控程序未能发布正确的UVC,则其在试图访问安全客户机页时接收异常。当虚拟机监控程序分页调出客户机页时,发布类似的UVC,该UVC在将客户机页标记为非安全并且将其在分区安全表中注册为非安全之前加密客户机页(除非共享)。

[0064] 在具有五个给定主机绝对页K、P、L、M和N的示例中,主机绝对页中的每一个在虚拟机监控程序将它们分页调入时被安全接口控件标记为安全。这防止非安全客户机和虚拟机监控程序访问它们。当虚拟机监控程序分页调入主机绝对页K、P和M时,主机绝对页K、P和M被注册为属于客户机A;当虚拟机监控程序分页调入主机绝对页L和N时,主机绝对页L和N被注册到客户机B。共享页(在单个安全客户机和虚拟机监控程序之间共享的页)在分页期间不被加密或解密。它们不被标记为安全(允许由虚拟机监控程序访问),而是向分区安全表中的单个安全客户机域注册。

[0065] 根据本发明的一个或多个实施例,当非安全客户机或虚拟机监控程序试图访问安全客户机所拥有的页时,虚拟机监控程序接收到安全-存储访问(PIC3D)异常。确定这一点,无需要额外的转换步骤。

[0066] 根据一个或多个实施例,当安全实体试图访问页时,硬件执行验证存储确实属于特定的安全客户机的额外转换检查。如果不是,则向虚拟机监控程序提交非安全访问(PIC3E)异常。此外,如果被转换的主机虚拟地址与分区安全表中的注册的主机-地址对的主机虚拟地址不匹配,则识别出安全存储违规('3F'x)异常。为了实现与虚拟机监控程序的共享,只要转换检查允许访问,安全客户机就可以访问未被标记为安全存储。

[0067] 现在转向图5,总体上示出根据本发明的一个或多个实施例的DAT操作的系统示意

图500。系统示意图500包括其中的页被转换(例如,参见主机DAT转换525;注意,虚线表示通过DAT转换525的映射)到虚拟机监控程序(主机)绝对地址空间530的主机基本虚拟地址空间(host primary virtual address space)510和主机本部虚拟地址空间(host home virtual address space)520。例如,图5展示了由两个不同的主机虚拟地址空间对主机绝对存储的共享,也展示了那些主机虚拟地址之一不仅在两个客户机之间的共享,而且另外还与主机本身的共享。就这一点而言,主机基本虚拟地址空间510和主机本部虚拟地址空间520是两个主机虚拟地址空间的实例,这两个主机虚拟地址空间中的每个分别由单独的ASCE—主机基本ASCE(HPASCE)591和主机本部ASCE(HHASCE)592—来寻址。注意,所有安全接口控件存储(虚拟的和实的)都由虚拟机监控程序捐赠并被标记为安全。一旦被捐赠,只要相关联的安全实体存在,安全接口控件存储就只能由安全接口控件访问。

[0068] 如图所示,主机基本虚拟地址空间510包括客户机A绝对页A1.HV、客户机A绝对页A2.HV、客户机B绝对页B1.HV和主机虚拟页H3.HV。主机本部虚拟地址空间520包括安全-接口-控制虚拟页U1.HV、主机虚拟页H1.HV和主机虚拟页H2.HV。

[0069] 根据本发明的一个或多个实施例,所有安全客户机(例如,安全客户机A和安全客户机B)存储在本文所述的分区安全表中被注册为属于安全客户机配置,并且将相关联的主机虚拟地址(例如,A1.HV、A2.HV、B1.HV)也被注册为主机-地址对的一部分。在一个或多个实施例中,所有安全客户机存储被映射在主机基本虚拟空间中。此外,所有安全接口控件存储也在分区安全表中被注册为属于安全接口控件,并且可以基于相关联的安全客户机域在分区安全表中被进一步区分。根据本发明的一个或多个实施例,UV虚拟存储器被映射在主机本部虚拟空间中,并且相关联的主机虚拟地址被注册为主机-地址对的一部分。根据一个或多个实施例,UV实存储不具有相关联的主机虚拟映射,并且分区安全表中的DA位(其指示虚拟地址比较被禁用)被设置,以指示这一点。主机存储被标记为非安全,并且也在分区安全表中被注册为非安全。

[0070] 因此,在“客户机绝对=主机虚拟”的情况下,虚拟机监控程序(主机)主要DAT表(由HPASCE 591定义)如下地转换主机基本虚拟地址空间510的页:客户机A绝对页A1.HV被映射到属于安全客户机A的主机绝对页A1.HA;客户机A绝对页A2.HV被映射到属于安全客户机A的主机绝对页A2.HA;客户机B绝对页B1.HV被映射到属于安全客户机B的主机绝对页B1.HA;主机虚拟页H3.HV被映射到主机绝对页H3.HA非安全主机(没有主机-地址对,因为它是非安全的)。进一步,(由HHASCE 592定义的)虚拟机监控程序(主机)主DAT表如下转换主机本部虚拟地址空间520的页:安全接口控件虚拟页U1.HV被映射到被定义为安全UV虚拟的主机绝对页U1.HA;主机虚拟页H1.HV被映射到被定义为非安全的主机绝对页H1.HA;主机虚拟页H2.HV被映射到被定义为非安全的主机绝对页H2.HA。没有与H1.HA或H2.HA相关联的主机-地址对,因为它们是非安全的。

[0071] 在操作中,如果安全客户机试图访问分配给安全接口控件的安全页,则硬件向虚拟机监控程序提交安全存储违规('3F' X)异常。如果非安全客户机或虚拟机监控程序试图访问任何安全页(包括分配给安全接口控件的那些页),则硬件向虚拟机监控程序提交安全存储访问('3D' X)异常。可替代地,可以就对安全接口控件空间进行的试图访问提交错误条件(error condition)。如果硬件检测到安全接口控件访问上的安全分配的不匹配(例如,存储在分区安全表中被注册为属于安全客户机而不是属于安全接口控件,或者正在使用的

主机-地址对与注册的对不匹配),则提交检查。

[0072] 换言之,主机基本虚拟地址空间510包括主机虚拟页A1.HV和A2.HV(属于安全客户机A)以及B1.HV(属于安全客户机B),它们分别映射到主机绝对A1.HA、A2.HA以及B1.HA。此外,主机基本虚拟地址空间510包括主机(虚拟机监控程序)页H3.HV,其映射到主机绝对H3.HA。主机本部虚拟空间520包括两个主机虚拟页H1.HV和H2.HV,它们映射到主机绝对页H1.HA和H2.HA。主机基本虚拟地址空间510和主机本部虚拟地址空间520两者映射到单个主机绝对空间530中。属于安全客户机A和安全客户机B的存储页被标记为安全并用其安全域和相关联的主机虚拟地址在图1所示的分区安全表100中注册。另一方面,主机存储器被标记为非安全。当虚拟机监控程序定义安全客户机时,它必须向安全接口控件捐赠主机存储,以用于支持这些安全客户机所需的安全控制块。该存储可以在主机绝对或主机虚拟空间中定义,在一个例子中,具体地在主机本部虚拟空间中定义。返回到图5,主机绝对页U1.HA和U2.HA安全UV绝对是被定义为主机绝对存储的安全接口控件存储。因此,这些页被标记为安全并且在图1中所示的分区安全表100中被注册为属于安全接口控件并且具有相关联的安全域。因为这些页被定义为主机绝对地址,所以没有相关联的主机虚拟地址,所以在分区安全表100中的DA位被置位(set)。

[0073] 转换后的虚拟机监控程序(主机)绝对地址空间530的实例可见于图6。图6是根据本发明的一个或多个实施例描绘关于安全接口控件存储器的系统示意图600。系统示意图600示出虚拟机监控程序(主机)绝对地址空间630,其包括主机绝对页A2.HA安全客户机A(用于A2.HV);主机绝对页B1.HA安全客户机B(用于B1.HV);主机绝对页H1.HA非安全(主机);主机绝对页H2.HA非安全(主机);主机绝对页U3.HA安全UV实(无HV映射);主机绝对页U1.HA安全UV虚拟(用于U1.HV);以及主机绝对页A1.HA安全客户机A(用于A1.HV)。

[0074] 现在转到图7,根据本发明的一个或多个实施例,总体上示出了用于导入操作的过程流700。当安全客户机访问被虚拟机监控程序分页调出的页时,发生诸如过程流700中所示的事件序列,以便安全地将该页调回。过程流700在框705开始,其中安全客户机访问客户机虚拟页。由于该页例如是无效的,硬件向虚拟机监控程序提交由程序中断代码11(PIC11)指示的主机页错误(参见框715)。虚拟机监控程序进而识别该客户机页的可用的非安全主机绝对页(参见框720),并将加密的客户机页分页调入到所识别的主机绝对页(参见框725)。

[0075] 然后在框730处,主机绝对页在适当的(基于主机虚拟地址的)主机DAT表中被映射。在框735处,虚拟机监控程序主机然后重新调度安全客户机。在框740处,安全客户机重新访问客户机安全页。页错误不再存在,但是由于这是安全客户机访问并且该页在图100的分区安全表100中未被标记为安全,在框745处,硬件向虚拟机监控程序提交非安全存储异常(PIC3E)。这个PIC3E阻止客户机对这个安全页的访问,直到已经发布了必要的导入为止。接下来,过程流700前进至续接到图8的“A”。

[0076] 现在转向图8,根据本发明的一个或多个实施例,总体上示出了用于执行导入操作的过程流800。响应于PIC3E,表现良好的(例如,以预期方式无错误地执行的)虚拟机监控程序将发布导入UVC(参见框805)。注意,此时,要被导入的页被标记为非安全(non-secure)的,只能被虚拟机监控程序、其他非安全实体和安全接口控件访问。它不能被安全客户机访问。

[0077] 作为导入UVC的一部分,充当安全接口控件的可信固件进行检查,以查看此页是否已经被安全接口控件锁定(参见决策框810)。如果是,则过程流800前进到框820。在框820处,向虚拟机监控程序返回一个“忙”返回码,虚拟机监控程序将响应于该返回码而延迟(参见框825)并重新发布导入UVC(过程流800返回至框805)。如果该页尚未被锁定,则过程流800前进到决策框822。

[0078] 在决策框822处,安全接口控件进行检查,以查看该页是否是与非安全虚拟机监控程序共享的页。如果它是共享的(过程流800前进到判定框824),则安全接口控件在分区安全表中将主机绝对地址与相关联的安全客户机域、主机虚拟地址一起注册且注册为共享的。此页保持被标记为非安全。这就完成了导入UVC,该页现在可用于由客户机访问。处理继续进行,虚拟机监控程序重新调度客户机(框830),安全客户机成功地访问该页(框835)。

[0079] 如果要导入的主机虚拟页不是与虚拟机监控程序共享的(过程流800前进到框840),则安全接口控件将该页标记为安全,使得虚拟机监控程序不再能访问该页。在框845处,安全接口控件锁定该页,使得任何其他UVC都不能修改该页状态。一旦设置了锁(在框850),安全接口控件将验证客户机页的内容在其被加密期间没有改变。如果客户机页的内容确实改变了,则向虚拟机监控程序返回错误返回码,否则,安全接口控件将解密该安全页。

[0080] 在框855处,安全接口控件解锁该页(从而允许由其他UVC访问),在分区安全表中将该页注册为安全并与适当客户机域和主机虚拟地址相关联,以完成主机-地址HV->HA对。这允许客户机访问并完成UVC。

[0081] 现在转到图9,根据本发明的一个或多个实施例,总体上示出了关于捐赠存储器操作的过程流900。过程流900在框905处开始,其中虚拟机监控程序向安全接口控件发布查询UVC。在框910处,安全接口控件返回数据(例如,查询UVC)。此数据可以包括所需的特定于基本分区的主机-绝对存储的量;所需的特定于基本安全-客户机-域的主机-绝对存储的量;每MB所需的特定于可变安全-客户机-域的主机-虚拟存储的量;和/或所需的特定于基本安全-客户机-CPU的主机-绝对存储的量。

[0082] 在框915处,虚拟机监控程序保留特定于基本主-机绝对分区的存储(例如,基于由查询UVC返回的大小)。在框920,虚拟机监控程序向安全接口控件发布初始化(initialization)。就这一点而言,虚拟机监控程序可以发布为在整个分区的安全客户机配置之间进行协调所需的UV控制块提供所捐赠的存储的初始化UVC。初始化UVC指定特定于基本分区的存储来源。

[0083] 在框925处,安全接口控件通过将所捐赠的存储注册到UV并标记为安全来实现初始化(例如,初始化UVC)。为了实现初始化UVC,安全接口控件可以将所捐赠的存储标记为安全;为分区安全表分配该捐赠的存储中的一些;以及在分区安全表中将所赠送的存储与唯一性安全-域一起注册供UV使用,但是不与相关联的安全-客户机-域一起注册,并且注册为不具有相关联的主机-虚拟地址对。

[0084] 在框930,虚拟机监控程序保留存储(例如,特定于基础和可变安全-客户机-域的存储)。例如,虚拟机监控程序保留特定于基本和可变(例如,基于安全-客户机-域存储的大小)安全-客户机-域的存储(例如,由查询UVC返回的大小)。在框935,虚拟机监控程序向安全接口控件发布创建配置(create configuration)。就这一点而言,虚拟机监控程序可以

发布指定特定于基本和可变安全-客户机-域的存储来源的创建-安全-客户机-配置(create-secure-guest-config)UVC。进一步,创建-安全-客户机-配置UVC为支持该安全客户机配置所需的UV控制块提供所捐赠的存储。

[0085] 在框940处,安全接口控件实现创建配置(例如,创建-安全-客户机-配置UVC)。为了实现创建-安全-客户机-配置UVC,安全接口控件可以将所捐赠的存储标记为安全;将所捐赠的存储在分区安全表中注册为供UV使用;并且将所捐赠的存储与相关联的安全-客户机-域一起注册。将所捐赠的基本(主机-绝对)存储注册为不具有相关联的主机-虚拟地址对。将所捐赠的可变(主机-虚拟)存储与相关联的主机-虚拟地址一起注册。

[0086] 在框945处,虚拟机监控程序保留特定于基本安全-客户机-CPU的存储(例如,由查询-UV返回的大小)。在框950处,虚拟机监控程序指定存储源。例如,虚拟机监控程序向UV发布指定特定于基本安全-客户机-CPU的存储源的创建-安全-客户机-CPU(create-secure-guest-CPU)。在框955处,安全接口控件实现该创建-CPU(例如,创建-安全-客户机-CPU UVC)。为了实现创建-安全-客户机-CPU UVC,安全接口控件可以将所捐赠的存储标记为安全并且在分区安全表中将所捐赠的存储注册为供UV使用,但是不与相关联的安全-客户机-域一起注册,并且注册为不具有相关联的主机-虚拟地址对。

[0087] 现在转到图10,根据本发明的一个或多个实施例,总体上示出了关于非安全虚拟机监控程序页到安全接口控件的安全页的转换的过程流1000。在过程流1000中,示出了三个虚拟机监控程序页(例如,非安全虚拟机监控程序页A、非安全虚拟机监控程序页B和非安全虚拟机监控程序页C)。

[0088] 虚拟机监控程序(非安全)页A、B和C可以由非安全实体(包括虚拟机监控程序)访问。进一步,虚拟机监控程序(非安全)页A、B和C被标记为非安全(NS),并且在分区安全表(例如,图1中所示的分区安全表100)中被注册为非安全和非共享。在箭头1005处,发布初始化UVC,其将客户机页A转换到与整个分区(UV2)相关联的安全接口控件实存储页1010。安全接口控件实存储1010可以被标记为安全,并且在分区安全表(例如,图1中所示的分区安全表100)中被注册为无客户机域且无虚拟机监控程序到主机绝对(HV->HA)映射的UV。它反而是与唯一UV2安全域一起注册,且将DA位设置为1。注意,安全接口控件实储存1010可作为实被安全接口控件访问。

[0089] 在箭头1025处,从虚拟机监控程序(非安全)页B发布创建-SG-配置(create-SG-config)或创建-SG-CPU(create-SG-CPU)UVC,其将该页转换到与安全客户机域(UVS)相关联的安全接口控件实存储1030。安全接口控件实存储1030可以被标记为安全,并且在分区安全表(例如,图1中所示的分区安全表100)中被注册为具有相关联的安全客户机域且无虚拟机监控程序到主机绝对映射(HV->HA)(即,DA位=1)的UV。注意,安全接口控件实储存1010可作为代表安全客户机域的实被安全接口控件访问。

[0090] 在箭头1045处,从虚拟机监控程序(非安全)页C发布创建-SG-配置UVC,其将该页转换到与安全客户机域(UVV)相关联的安全接口控件虚拟存储1050。安全接口控件虚拟存储1050可以被标记为安全,并且在分区安全表(例如,图1中所示的分区安全表100)中被注册为具有安全客户机域和虚拟机监控程序到主机绝对(HV->HA)映射的UV。注意,安全接口控件虚拟存储1050可作为代表安全客户机域的UV虚拟被访问。

[0091] 现在转到图11,根据一个或多个实施例,示出了关于由程序或安全接口控件进行

的安全存储访问的过程流1100。这代表安全接口控件将要访问客户机存储或安全接口控件存储并且必须正确地标记该访问以便允许硬件验证该访问的安全性的情形。过程流1100描绘了对安全接口控件进行的存储访问的这种标记。过程流1100在框1110开始,其中安全接口控件确定其是否在对安全接口控件存储进行访问。

[0092] 如果这不是对安全接口控件存储的访问,则过程流1100前进到决策框1112(如“否”箭头所示)。在决策框1112处,安全接口控件确定其是否在对安全客户机存储进行访问。如果这不是对安全客户机存储的访问,则过程流1100前进至“B”(其连接至图12的过程流1200),其将使用用于安全客户机访问的默认设置。如果这是对安全客户机存储的访问,则过程流1100前进到决策框1113,其中安全接口控件确定是否在使用默认安全客户机域。如果是,则过程流1100前进至“B”(其连接至图12的过程流1200),其将使用用于安全客户机访问的默认设置。如果否,则过程流1100进行到框1114。在框1114处,将合适的安全客户机域加载到SG-安全-域寄存器中(并且前进至“B”,其连接至图12的过程流1200)。

[0093] 如果这是对安全接口控件存储的访问,那么过程流1100进行到方框1120(如“是”箭头所示)。在框1120处,将访问标记为安全UV(例如,使用UV-安全-域寄存器)。

[0094] 过程流1100接着前进到决策框1130,其中安全接口控件确定这是否是对UVV空间(例如,SG-配置变量表(SG-Config Variable Table))的访问。如果其是对UVV空间的访问,则过程流1100进行到框1134(如“是”箭头所示)。在框1134,将访问标记为虚拟。在框1136处,将可应用的安全客户机域加载到UV-安全-域寄存器中。在框1138处,DAT转换和访问存储准备就绪,可以开始。返回到决策框1130,如果这不是对UVV空间的访问,则过程流1100前进到框1140(如“否”箭头所示)。在框1140,将访问标记为实。

[0095] 在决策框1150处,安全接口控件确定这是否是对UVS空间(例如,SG配置或CPU表)的访问。如果这是对UVS空间的访问,则过程流1100前进到框1136(如“是”箭头所示)。如果这不是对UVS空间的访问,那么过程流1100前进到框1170(如由“否”箭头所示)。这个访问于是就是对UV2空间(例如,分区安全表)的访问。在框1170处,将唯一UV2安全域加载到UV-安全-域寄存器中。

[0096] 图12描绘了根据本发明的一个或多个实施例的过程流1200。当某客户机被分派时,SIE进入(SIE Entry)固件可以向硬件指示一个客户机正在运行(例如,客户机模式处于活动状态)并且可以指示该客户机是否是安全的。如果客户机是安全的,则可以将相关联的安全客户机域加载到硬件中(例如,在SG安全域寄存器中)。当程序访问存储时,硬件可以基于程序在访问时的当前状态来标记该访问。图12示出了过程流1200中该过程的示例。在框1205处,硬件可以确定机器当前是否以客户机模式运行,如果不是,则可以在框1210处将该访问标记为主机访问,并且在框1215处标记为非安全访问。如果在框1205处确定机器以客户机模式运行,则可以在框1220处将该访问标记为客户机访问,并且在框1225处进一步确定当前客户机是否是安全客户机。如果客户机为非安全的,则可以在框1215处将该访问标记为非安全。如果客户机是安全的,硬件可以在框1230处将客户机标记为安全,这可以将安全客户机与在分派安全客户机时被加载的SG-安全-域寄存器相关联。对于非安全和安全客户机两者,可以在框1235检查DAT状态。如果DAT关闭(off),则可以在框1240处将该访问标记为实。如果DAT开启(on),则可以在框1245处将该访问标记为虚拟。一旦访问在框1240处被标记为实且DAT关闭,或者在框1245被标记为虚拟且DAT开启,硬件就在框1250处准备好

开始如图13中进一步描述的转换和访问存储。

[0097] 图13描绘了根据本发明的一个或多个实施例的由硬件完成的用于支持过程流1300中的安全访问和非安全访问两者的转换的示例。在框1305处,硬件可以确定访问是否被标记为客户机转换,如果是的话,并且在框1310处确定访问是虚拟的,则可以在框1315处执行客户机DAT。在客户机DAT转换期间,可以有客户机DAT表的嵌套中间获取(nested, intermediate fetches for guest DAT tables)。如果原始转换被标记为安全,则可以将表获取(table fetches)标记为客户机实和安全。表获取也可以跟随过程流1300的转换过程。在针对在框1315处被标记为客户机虚拟的任何访问和针对在框1310处被标记为客户机实的任何访问(虚拟=否)执行客户机DAT之后,可以在框1320处应用客户机前缀和客户机内存偏址。在客户机转换过程完成时,在框1325处,如果原始客户机转换被标记为安全,则可以将所得地址标记为主机虚拟且标记为安全。过程1300可以如针对标记为主机虚拟的任何访问那样继续。如果在框1305处确定原始访问是主机访问(客户机=否)且在框1330处确定被标记为虚拟,则可以在框1335处执行主机DAT。在框1335处,可以将主机表获取标记为非安全。在框1335处执行主机DAT之后,或如果在框1330处确定原始主机访问被标记为实(虚拟=否),则可在框1340处应用主机前缀。在框1345处,所得地址可为主机绝对地址。

[0098] 图14描绘了根据本发明的一个或多个实施例的可以在过程流1400中由硬件执行的具有安全存储保护的DAT转换的示例。从图13的框1345继续,如果在框1405处识别了安全-UV访问,则硬件可以在框1410处验证该存储是否被注册为安全-UV存储,并且,如果否,则在框1415处提交错误。当访问UV存储时,安全接口控件可以进行安全-UV访问。如果在框1410处确定该存储被注册为安全-UV存储,则可以如针对任何安全访问的那样继续进行保护检查,只是在处理继续到的框1420处可以将(由安全接口控件在进行安全UV访问之前设置的)UV-安全-域寄存器用作域检查的指定安全域。另外,在框1425处检测到的针对UV访问的任何违背(进入点D),都可以在框1430处作为错误被提交,而不是像在框1425处针对安全客户机违背(安全-UV=否)所做的那样在框1435处向虚拟机监控程序提交异常。

[0099] 对于在框1405处未被标记为安全-UV访问的访问,在框1440处,硬件确定该访问是否是安全客户机访问,并且,如果否且如果在框1445处该页被标记为安全,则在框1435处可以向虚拟机监控程序提交异常。否则,如果在框1440处确定该访问不是安全客户机访问且在框1445处该页未被标记为安全,则在框1450处转换成功。

[0100] 如果在框1440处访问是安全客户机访问或者在框1410处是对注册为安全-UV存储的存储的安全-UV访问,则在框1420处,硬件可以进行检查,以确保该存储被注册到与该访问相关联的安全实体。如果这是安全-UV访问,则可以从(由安全接口控件基于正被访问的安全-UV存储而加载的)UV-安全-域寄存器获得指定的安全域,并且为安全-客户机访问而从(当分派安全实体时加载的)SG-安全-域寄存器获得指定的安全域。如果在框1420处正被访问的存储未被注册到指定的安全域,则针对在框1425处的安全-UV访问,在框1430处发生错误,并且,针对在框1425处的安全-客户机访问(安全-UV=否),在框1435处向虚拟机监控程序提交异常。

[0101] 对于在框1440处以及在框1420处对在框1410处注册到指定的安全域的存储的安全访问,如果在框1455处虚拟地址检查被禁用,即DA位=1,并且在框1460处该访问是实的,则在框1450处,转换完成。然而,如果在框1455处DA位=1但是在框1460处访问是虚拟的(实

＝否),则对于在框1425处的安全-UV访问,在框1430处发生错误,并且对于在框1425处的安全-客户机访问(安全-UV＝否),在框1435处向虚拟机监控程序提交异常。如果在框1455处DA位＝0并且在框1475处该访问是虚拟访问,则硬件可以在框1470处确定该访问的主机虚拟到主机绝对映射是否匹配为该主机绝对地址注册的映射。如果是,则在框1450处,转换成功完成。如果在框1470处映射不匹配,则对于在框1425的安全-UV访问,在框1430取得错误,并且对于在框1425的安全-客户机访问(安全-UV＝否),在框1435向虚拟机监控程序提交异常。如果在框1475DA位＝0并且访问是实访问(虚拟＝否),则对于在框1425的安全-UV访问,在框1430处发生错误,并且对于在框1425处的安全-客户机访问(安全-UV＝否),在框1435处向虚拟机监控程序提交异常;可替代地,转换可在框1450处成功完成。在框1480处可以检查由I/O子系统进行的任何访问,以在框1445处查看该页是否被标记为安全,并且如果该页是安全的,则可以在框1435处向虚拟机监控程序提交异常;如果该页未被标记为安全,则在框1450处,转换成功。

[0102] 可以通过分区安全表接口1485来共同管理存储注册和映射的各种检查。例如,框1410、1420、1455、1470和1475可以与关联于相同分区的分区安全表对接,以管理各种访问。

[0103] 现在转到图15,根据本发明的一个或多个实施例,总体上示出了用于确定在加密之前页内容是否已经改变的过程流1500。参见图16的数据流1600进一步描述过程流1500。在框1505处,过程流1500开始。在框1510处,计算计算机系统的内存安全页(secure page of memory)的哈希值。例如,可以是计算机系统的一部分的安全接口控件1605可以在安全页1610的内容被提供为非安全域1625中的非安全页1620(即,可访问的但加密的)之前计算安全域1615中的安全页1610的哈希函数。框1510的哈希函数可以是支持检测安全页1610的一个或多个数据位的改变的校验和的形式。安全页1610是非安全实体(诸如非安全虚拟机监控程序或操作系统)不能直接访问的。非安全页1620可以由诸如图7的过程流700的框725之类的不同过程使用。加密和解密可以由安全接口控件1605的加密/解密控制1630管理。加密和解密也可以作为各种过程(诸如图8的过程流800)的部分被调用。安全接口控件1605可以包括固件、硬件或固件与硬件的组合。例如,安全接口控件1605可以是处理单元(例如,计算机处理器的处理电路)的一部分或者是处理单元可调用的。安全页1610可被指派给由主机管理的安全容器或安全虚拟机,其中主机(host)是例如虚拟机监控程序或操作系统。

[0104] 在过程流1500的框1515处,可以将在框1510中计算的哈希值与该页的先前计算的哈希值进行比较。例如,安全接口控件1605的安全表1635可以存储关联页的多个页标识符1640和哈希值1645。当计算安全页1610的哈希值时,安全接口控件1605可以执行对页标识符1640中的相关联地址的查找,以确定哈希值1645是否包括安全页1610的先前计算的哈希值。

[0105] 在框1520处,基于确定哈希值不匹配相同页的先前计算的哈希值,可以将每页的每加密值(per-encryption value per page)的修改值用于在框1525处加密该页。在框1525处,基于确定哈希值匹配相同页的先前计算的哈希值,可以在加密该页时不修改/递增的情况下使用每加密计数器1650的每加密值。能够逐页地管理每加密计数器1650的值。可以在使用每加密值并引用相关联的页之前确立每加密计数器1650的初始值。每加密计数器1650例如可以由加密/解密控制1630用作加密或解密的一部分,以进一步随机化加密值和解密值之间的关系。作为一个示例,对页进行加密可以包括用密码安全哈希函数来组合该

页相关联的地址值、一个或多个随机值和每加密值。通过仅在底层数据改变时修改每加密值,所得到的未改变的数据的加密可与对先前对相同数据进行加密的副本对齐。即使非安全实体(诸如虚拟机监控程序或操作系统)可能不理解非安全页1620的加密内容,标识未改变的状态也可通过避免制作未改变的页的加密数据的进一步副本和/或防止对未改变的页的更新来实现更高效的内存管理。在框1530处,过程流1500结束。

[0106] 在本发明的实施例中,加密可由安全控制接口1605的加密/解密控制1630响应于主机(例如,虚拟机监控程序或操作系统)将页从安全页1610转换成非安全页1620的请求而执行。例如,作为内存管理操作的一部分,可以将加密的非安全页1620提供给主机用于存储。所计算的哈希值可以被存储在安全控制接口1605的安全表1635的哈希值1645中。在随后的操作时,安全控制接口1605可以将非安全页1620转换成安全页1610,并且加密/解密控制1630可以基于与该页相关联的每加密值对安全页1610进行解密以产生解密的页。安全接口控件1605可以计算解密的页的哈希值并且将解密的页的哈希值与存储在安全表1635中的哈希值1645进行比较。可以基于确定解密的页的哈希值与存储在安全表1635中的哈希值1645之一匹配来验证解密的页,并且可以使验证并解密的页可用于在安全域1615中作为安全页1610使用。该验证可以确认非安全页1620在处于非安全域1625中时未被修改(例如,被高速缓存到磁盘和被检索)。安全与非安全之间的页转换可包含设定与该页相关联的位或标签以限制该页的可访问性。

[0107] 应当理解,尽管本公开包括关于云计算的详细描述,但是本文所引用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他类型的计算环境来实现。

[0108] 云计算是一种服务交付模型,用于实现对可配置计算资源(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)的共享池的方便、按需的网络访问,所述可配置计算资源可以用最小的管理努力或与服务提供商的交互来快速配置和释放。该云模型可以包括至少五个特性、至少三个服务模型和至少四个部署模型。

[0109] 特性如下:

[0110] 按需自助服务:云消费者可按需自动地单方面供应计算能力,诸如服务器时间和网络存储,而无需与服务提供商进行人工交互。

[0111] 广泛的网络接入:通过网络提供功能,并通过标准机制进行访问,所述标准机制促进由异构的瘦客户端或厚客户端平台(例如,移动电话、膝上型计算机和PDA)的使用。

[0112] 资源池化:提供者的计算资源被汇集起来以使用多租户模型来服务于多个消费者,不同的物理和虚拟资源根据需要被动态分配和重新分配。存在位置独立性的意义,因为消费者通常对所提供资源的确切位置不具有控制权或知识,但可能能够指定更高抽象层级的位置(例如,国家、州或数据中心)。

[0113] 快速弹性:可以快速且弹性地配置功能,在某些情况下自动地快速扩展,迅速释放以快速收缩。对于消费者而言,可用于配置的功能通常看起来是无限的,可以在任何时间以任何数量购买。

[0114] 度量的服务:云系统通过利用与服务类型(例如,存储、处理、带宽和活动用户帐户)相适应的某种抽象级别的计量功能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情况,为所使用服务的提供者和使用者提供透明度。

[0115] 服务模型如下：

[0116] 软件即服务(SaaS)：向消费者提供的功能是使用在云基础设施上运行的提供者的应用。这些应用可通过诸如web浏览器(例如，基于web的电子邮件)的瘦客户端接口从不同客户端设备访问。消费者不管理或控制包括网络、服务器、操作系统、存储或甚至个体应用功能的底层云基础结构，可能的例外是有限的用户特定的应用配置设置。

[0117] 平台即服务(PaaS)：向消费者提供的功能是在云基础结构上部署消费者创建或获取的应用，所述应用是用提供者所支持的编程语言和工具创建的。消费者不管理或控制包括网络、服务器、操作系统或存储的底层云基础结构，但是具有对所部署的应用以及可能的应用托管环境配置的控制。

[0118] 基础设施即服务(IaaS)：向消费者提供的功能是提供消费者能够部署和运行可包括操作系统和应用的任意软件的处理、存储、网络和其他基本计算资源。消费者不管理或控制底层云基础结构，而是具有对操作系统、存储、所部署的应用的控制，以及对所选联网组件(例如，主机防火墙)的可能有限的控制。

[0119] 部署模型如下：

[0120] 私有云：云基础结构仅为组织运营。它可以由组织或第三方管理，并且可存在于场所内或场所外。

[0121] 社区云：云基础结构由多个组织共享，并支持具有共同关注点(例如，任务、安全要求、策略和合规性考虑)的特定社区。它可以由组织或第三方管理，并且可存在于场所内或场所外。

[0122] 公共云：云基础结构可供公众或大型行业团体使用，并由销售云服务的组织拥有。

[0123] 混合云：云基础结构是由两个或更多个云(私有、社区或公共的)组成的，这些云仍然是唯一性实体，但通过标准化或专有技术来绑定在一起，这些技术实现数据和应用的可移植性(例如，用于云之间的负载平衡的云突发)。

[0124] 云计算环境是面向服务的，着重于无状态性、低耦合、模块化和语义互操作性。云计算的核心是包括互连节点网络的基础架构。

[0125] 现在参见图17，描绘说明性云计算环境50。如图所示，云计算环境50包括一个或多个云计算节点52，云消费者使用的本地计算设备(诸如个人数字助理(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)可与云计算节点52通信。节点52可以彼此通信。它们可以在一个或多个网络中，诸如在上文所述的私有云、社区云、公共云或混合云或其组合中，被物理地或虚拟地分组(未示出)。这允许云计算环境50提供基础结构、平台和/或软件作为服务，云消费者不需要为其在本地计算设备上维护资源。应当理解，图17中所示的计算设备54A-N的类型仅旨在是说明性的，并且计算节点52和云计算环境50可通过任何类型的网络和/或网络可寻址连接(例如，使用web浏览器)与任何类型的计算机化设备进行通信。

[0126] 现在参见图18，示出了由云计算环境50(图17)提供的一组功能抽象层。应预先理解，图18中所示的部件、层和功能旨在仅是说明性的，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：

[0127] 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)架构的服务器62；服务器63；刀片服务器64；存储65；以及网络和联网

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。

[0128] 虚拟化层70提供抽象层,从该抽象层可以提供以下虚拟实体的示例:虚拟服务器71;虚拟存储72;虚拟网络73,包括虚拟专用网络;虚拟应用和操作系统74;以及虚拟客户端75。

[0129] 在一个示例中,管理层80可提供下文所描述的功能。资源供应81提供用于执行云计算环境内的任务的计算资源和其他资源的动态获取。计量和定价82在云计算环境内利用资源时提供成本跟踪,并针对这些资源的消费进行计费或开票。在一个示例中,这些资源可以包括应用软件许可证。安全性为云消费者和任务提供身份验证,以及对数据和其他资源的保护。用户门户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84提供云计算资源分配和管理,使得满足所需的服务级别。服务水平协议(SLA)计划和履行85为根据SLA预期的云计算资源的未来要求提供云计算资源的预安排和采购。

[0130] 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荷和功能的示例包括:地图和导航91;软件开发和生命周期管理92;虚拟教室教育交付93;数据分析处理94;事务处理95;以及控制对虚拟机的安全存储的访问96。

[0131] 现在转到图19,描绘了根据本发明的一个或多个实施例的系统1900。系统1900包括例如经由网络165与一个或多个客户端设备20A-20E直接或间接通信的示例节点10(例如,托管节点)。节点10可以是云计算提供商的数据中心或主机服务器。节点10执行虚拟机监控程序12,其促进部署一个或多个VM15(15A-15N)。节点10还包括硬件/固件层13,其包括安全接口控件11。安全接口控件11包括促进虚拟机监控程序12向虚拟机15提供一个或多个服务的一个或多个硬件模块和固件。在虚拟机监控程序12与安全接口控件11之间、安全接口控件11与一个或多个VM15之间、虚拟机监控程序12与一个或多个VM15、以及虚拟机监控程序12通过安全接口控件11与VM15之间,可以通信。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不包括虚拟机监控程序12与一个或多个VM15之间的任何直接通信。

[0132] 例如,托管节点10可促进客户端设备20A部署虚拟机15A-15N中的一个或多个。可响应于来自不同客户机设备20A-20E的相应请求部署虚拟机15A-15N。例如,VM 15A可以由客户端设备20A部署,VM 15B可以由客户端设备20B部署,VM 15C可以由客户端设备20C部署。节点10还可以促进客户机端供应物理服务器(而不作为VM运行)。在此描述的示例将节点10中的资源的供应具体化为VM的一部分,然而,所描述的技术方案还可以应用于将资源供应为物理服务器的一部分。

[0133] 在示例中,客户端设备20A-20E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,并且节点10可以作为实体的私有云来操作。在这种情况下,节点10仅托管由属于实体的客户机设备20A-20E部署的虚拟机15A-15N。在另一示例中,客户端设备20A-20E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,节点10可以被操作为托管来自不同实体的VM的公共云。例如,虚拟机15A-15N可以以其中VM 15A不促进对VM15B的访问的屏蔽方式部署。例如,节点10可使用IBM zSystems®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(诸如PR/SM LPAR)提供分区之间的隔离,因此促进节点10在不同的逻辑分区中为同一物理节点10上的不同实体部署两个或更多个虚拟机15A-15N。PR/

SM LPAR虚拟机监控程序在具有特定硬件的可信的内部固件中实现以提供这种隔离。

[0134] 来自客户端设备20A-20e的客户端设备20A是诸如计算机、智能电话、平板计算机、台式计算机、膝上型计算机、服务器计算机的通信设备或请求由节点10的虚拟机监控程序12部署VM的任何其他通信设备。客户端设备20A可以经由网络165发送由虚拟机监控程序接收的请求。虚拟机15A-15N中的VM 15A是虚拟机监控程序12响应于客户端设备20A-20e中的客户端设备20A的请求而部署的VM映像。虚拟机监控程序12是VM监视器(VMM),其可以是创建和运行VM的软件、固件或硬件。虚拟机监控程序12促进VM 15A使用节点10的硬件组件来执行程序和/或存储数据。加以适当的特征和修改,虚拟机监控程序12可以是IBMzSystems®、Oracle的VM服务器、Citrix的XenServer、Vmware的ESX、MicrosoftHyper-V虚拟机监控程序或任何其他虚拟机监控程序。虚拟机监控程序12可以是直接在节点10上执行的本机虚拟机监控程序,或者在另一虚拟机监控程序上执行的托管虚拟机监控程序。

[0135] 现在转向图20,根据本发明的一个或多个实施例示出了用于实现本文的教导的节点10。节点10可以是包括和/或采用如本文所述的利用不同通信技术的任意数量的计算设备和网络及其组合的电子计算机框架。节点10可以容易地可升级、可扩展和模块化,具有改变到不同服务或独立于其他节点而重新配置一些特征的能力。

[0136] 在本实施例中,节点10具有处理器2001,处理器2001可以包括一个或多个中央处理单元(CPU)2001a、2001b、2001c等。处理器2001(也被称为处理电路、微处理器、计算单元)经由系统总线2002耦合到系统存储器2003和不同其他组件。系统存储器2003包括只读存储器(ROM)2004和随机存取存储器(RAM)2005。ROM 2004耦合到系统总线2002,并且可以包括基本输入/输出系统(BIOS),其控制节点10的某些基本功能。RAM是耦合到系统总线2002以供处理器2001使用的读写存储器。

[0137] 图20的节点10包括硬盘2007,其是由处理器2001可执行的可读的有形存储介质的示例。硬盘2007存储软件2008和数据2009。软件2008被存储为由处理器2001在节点10上执行的指令(以执行过程,例如参见图1-19描述的过程)。数据2009包括以不同数据结构组织以支持软件2008的操作和由软件2008的操作使用的定性或定量变量的一组值。

[0138] 图20的节点10包括互连并支持处理器2001、系统存储器2003、硬盘2007和节点10的其他组件(例如,外围设备和外部设备)之间的通信的一个或多个适配器(例如,硬盘控制器、网络适配器、图形适配器等)。在本发明的一个或多个实施例中,一个或多个适配器可以连接到经由中间总线桥连接到系统总线2002的一个或多个I/O总线,并且一个或多个I/O总线可以利用诸如外围组件互连(PCI)的公共协议,

[0139] 如图所示,节点10包括将键盘2021、鼠标2022、扬声器2023和麦克风2024互连到系统总线2002的接口适配器2020。节点10包括将系统总线2002互连到显示器2031的显示适配器2030。显示适配器2030(和/或处理器2001)可以包括用于提供诸如GUI 2032的显示和管理的图形性能的图形控制器。通信适配器2041将系统总线2002与网络2050互连,使得节点10能够与其他系统、设备、数据和软件(诸如服务器2051和数据库2052)通信。在本发明的一个或多个实施例中,软件2008和数据2009的操作可以由服务器2051和数据库2052在网络2050上实现。例如,网络2050、服务器2051和数据库2052可以组合起来以提供软件2008和数据2009的内部迭代,作为平台即服务、软件即服务和/或基础设施即服务(例如,作为分布式系统中的web应用)。

[0140] 本文描述的实施例必然以计算机技术为根源,并且具体地以托管VM的计算机服务器为根源。进一步,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(特别是托管VM的计算机服务器)的操作的改进,其中,即使虚拟机监控程序也被禁止访问与安全VM相关联的存储器、寄存器和其他这样的数据。此外,本发明的一个或多个实施例通过使用包括硬件、固件(例如,毫码)或其组合的安全接口控件(在此也被称为"UV")为改进VM托管计算服务器而提供重要步骤,以促进安全VM和虚拟机监控程序的分离,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,而不会对如本文所述的VM的初始化/退出期间保障VM状态安全添加大量开销。

[0141] 在此公开的本发明的实施例可以包括控制对VM的安全存储的访问的系统、方法和/或计算机程序产品(在此为系统)。注意,对于每个解释,将元件的标识符重复用于不同图的其他类似元件。

[0142] 在此参考相关附图描述了本发明的不同实施例。在不脱离本发明的范围的情况下,可以设计本发明的替代实施例。在描述和附图中的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,并且本发明在这方面并示意图进行限制。因而,实体的耦合可以指直接或间接耦合,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并入到具有本文未详细描述的额外步骤或功能的更全面的程序或过程中。

[0143] 以下定义和缩写用于解释权利要求书和说明书。如在此使用的,术语"包含"、"包括"、"含有"、"有"、"具有"、"存在"或其任何其他变体旨在覆盖非排他性的包含。例如,包含一系列元素的组合物、混合物、过程、方法、制品或设备不一定仅限于那些元素,而是可包括未明确列出的或此类组合物、混合物、过程、方法、制品或设备固有的其他元素。

[0144] 另外,术语"示例性"在此用于意指"充当示例、实例或说明"。在此描述为"示范性"的任何实施例或设计不一定被解释为比其他实施例或设计优选或有利。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(即,一、二、三、四等)的任何整数。术语"多个"可以被理解为包括大于或等于两个(即,两个、三个、四个、五个等)的任何整数。术语"连接"可以包括间接"连接"和直接"连接"两者。"

[0145] 术语"约"、"基本上"、"大约"及其变体旨在包括与基于在提交本申请时可用的设备的具体量的测量相关联的误差程度。例如,"约"可以包括给定值的±8%或5%、或2%的范围。

[0146] 本发明可以是任何可能的集成技术细节水平的系统、方法和/或计算机程序产品。所述计算机程序产品可包含上面具有计算机可读程序指令的计算机可读存储介质(或介质),所述计算机可读程序指令用于致使处理器执行本发明的各方面。

[0147] 计算机可读存储介质可以是可以保留和存储指令以供指令执行设备使用的有形设备。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各项的任何合适的组合。计算机可读存储介质的更具体例子的非穷举列表包括以下:便携式计算机盘,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)、便携式致密盘只读存储器(CD-ROM),数字通用盘(DVD)、记忆棒、软盘、机械编码设备(诸如穿孔卡片

或具有记录在其上的指令的凹槽中的凸起结构),以及上述的任意合适的组合。如本文中所使用的计算机可读存储介质不应被解释为瞬态信号本身,诸如无线电波或其他自由传播的电磁波、通过波导或其他传输介质传播的电磁波(例如,通过光纤电缆的光脉冲)、或通过导线传输的电信号。

[0148] 本文所述的计算机可读程序指令可从计算机可读存储介质下载到相应的计算/处理设备,或经由网络(例如,互联网、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且转发这些计算机可读程序指令以便存储在对应的计算/处理设备内的计算机可读存储介质中。

[0149] 用于执行本技术方案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,集成电路的配置数据,或以一种或多种编程语言的任何组合编写的源代码或目标代码,包括面向对象的Smalltalk、C++等编程语言,以及过程式编程语言,例如“C”编程语言或类似的编程语言。计算机可读程序指令可完全在用户的计算机上执行、部分在用户的计算机上执行、作为独立软件包执行、部分在用户的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。在后一种情形中,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,通过使用互联网服务提供商的互联网)。在一些实施例中,电子电路(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))可以通过使用计算机可读程序指令的状态信息来执行计算机可读程序指令以使电子电路个性化,以便执行本发明的各方面。

[0150] 在此参照根据技术方案的实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图来描述本技术方案的各方面。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,都可以由计算机可读程序指令来实现。

[0151] 这些计算机可读程序指令可以被提供给通用计算机的处理器,专用计算机或其他可编程数据处理装置,以产生机器,其通过计算机或其他可编程数据处理装置的处理器执行,创建用于实现在流程图和/或方框图的一个或多个方框中指定的功能/动作的装置。这些计算机可读程序指令还可存储在可指导计算机的计算机可读存储介质中,可编程数据处理装置,和/或以特定方式起作用的其他设备,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/或框图中的一个或多个方框中规定的功能/动作的各方面的指令。

[0152] 计算机可读程序指令还可以加载到计算机、其他可编程数据处理装置上,或使得在计算机上执行一系列操作步骤的其他装置,其他可编程装置或其他设备,以产生计算机实现的过程,使得在计算机上执行的指令,其他可编程装置或其他设备实现流程图和/或框图中的一个或多个方框中指定的功能和动作。

[0153] 附图中的流程图和框图示出了根据本技术方案的不同实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。对此,流程图或框图中的每个方框可以代表模块、段或指令的一部分,其包括用于实现规定的逻辑功能的一个或多个可执行指令。在一些替代实施例中,框中所标注的功能可以不以图中所标注的次序发生。例如,取决于所涉

及的功能,连续示出的两个框实际上可以基本上同时执行,或者这些框有时可以以相反的顺序执行。还将注意的是,框图和/或流程图中的每个框、以及框图和/或流程图中的框的组合可以由基于专用硬件的系统来实现,所述基于专用硬件的系统执行指定的功能或动作或执行专用硬件与计算机指令的组合。

[0154] 在此使用的术语仅用于描述具体实施例的目的并且不旨在是限制性的。如在此使用的,单数形式“一个”、“一种”和“该”旨在也包括复数形式,除非上下文另外清楚地指示。将进一步理解的是,当在本说明书中使用术语“包括”和/或“包含”时,其指定所陈述的特征、整数、步骤、操作、元件和/或组件的存在,但不排除一个或多个其他特征、整数、步骤、操作、元件、组件和/或其组的存在或添加。

[0155] 出于说明和描述的目的已经给出了对一个或多个实施例的描述,但是并不旨在是详尽的或限于所公开的形式。许多修改和变化对本领域的普通技术人员将是明显的。实施例的选择和描述方式是为了最好地解释各个方面和实际应用,使得本领域普通技术人员能够理解具有适合于所预期的特定用途的不同修改的不同实施例。

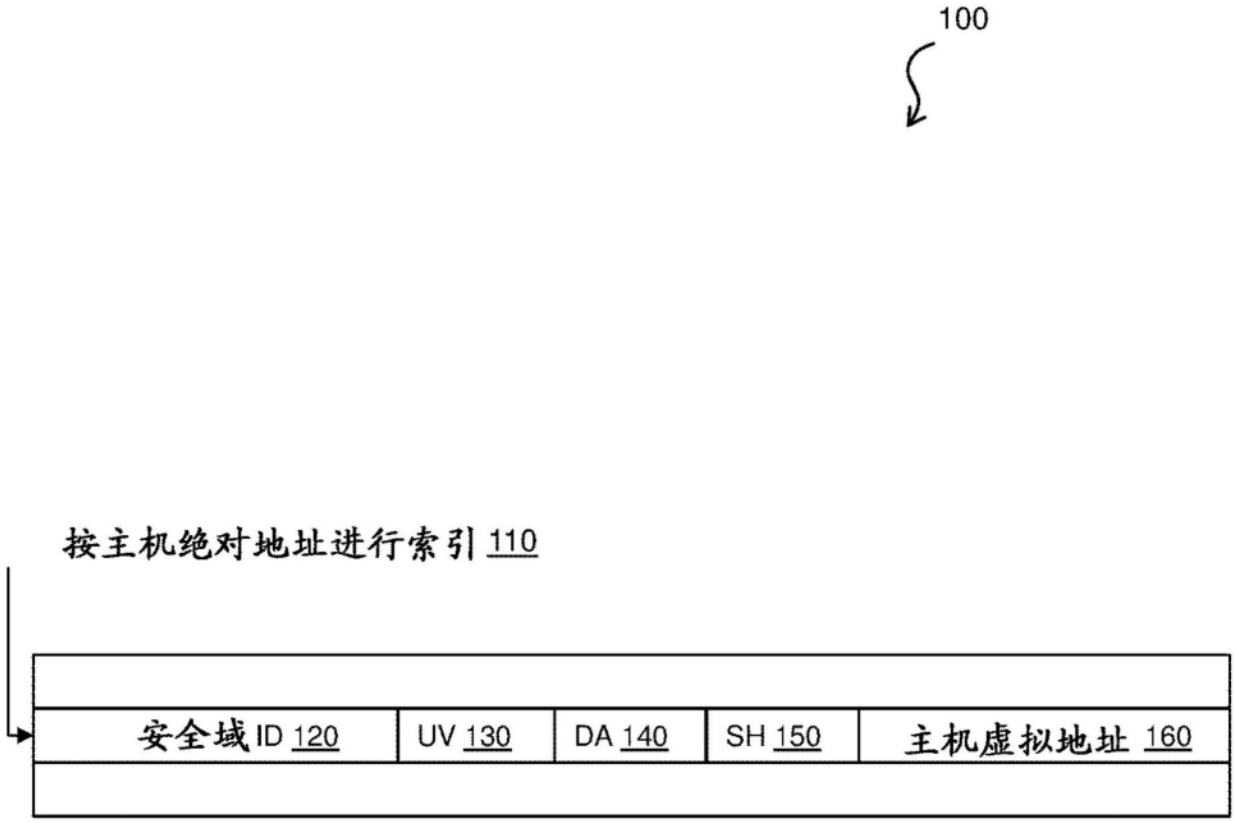


图1

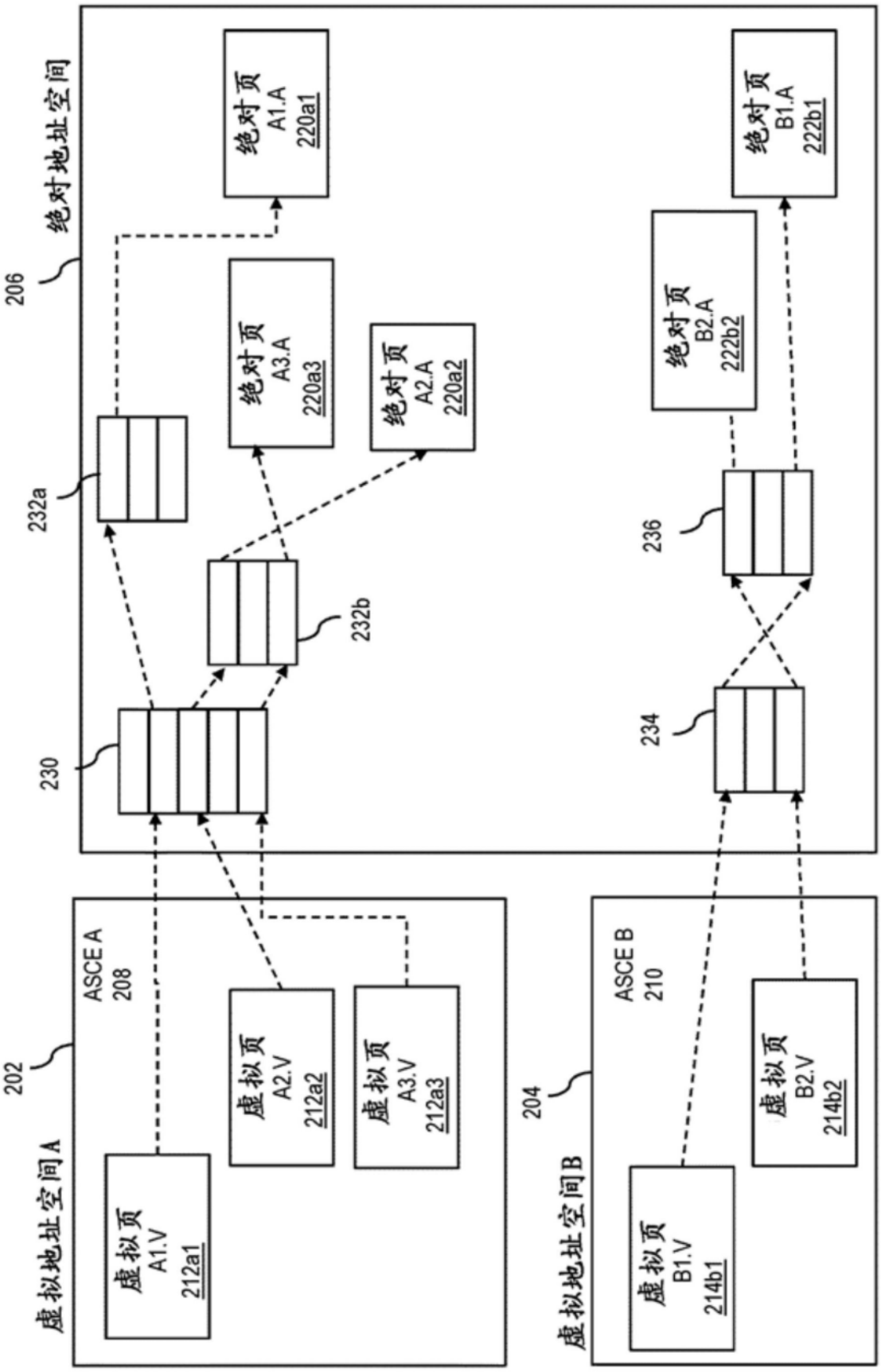

虚拟地址空间A
202
ASCE A
208
虚拟页
A1.V
212a1
虚拟页
A2.V
212a2
虚拟页
A3.V
212a3
虚拟地址空间B
204
ASCE B
210
虚拟页
B1.V
214b1
虚拟页
B2.V
214b2
绝对地址空间
206
230
232a
232b
234
236
绝对页
A1.A
220a1
绝对页
A3.A
220a3
绝对页
A2.A
220a2
绝对页
B2.A
222b2
绝对页
B1.A
222b1

图2

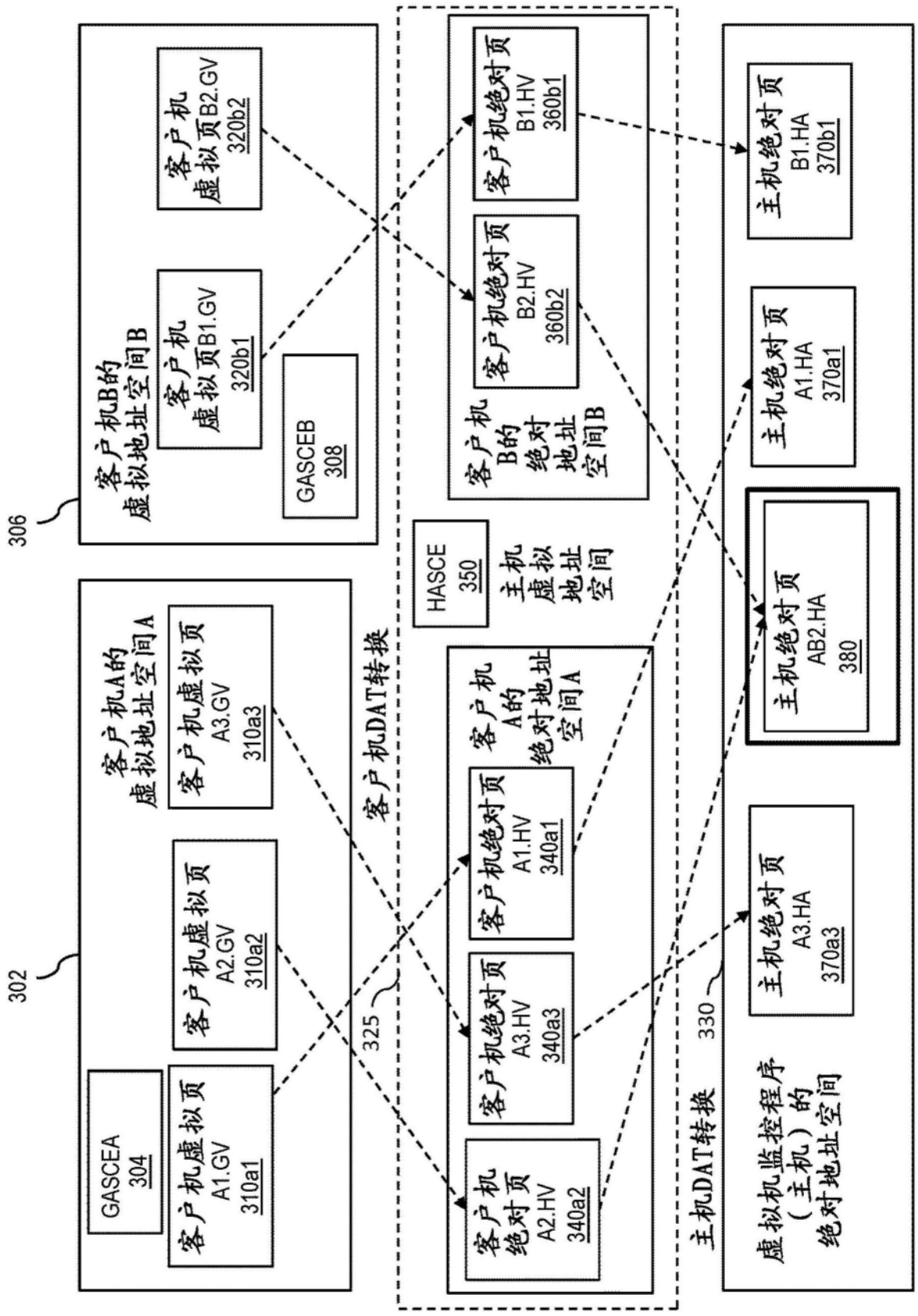

302
306
325
330
客户机A的虚拟地址空间A
客户机虚拟页A1.GV 310a1
客户机虚拟页A2.GV 310a2
客户机虚拟页A3.GV 310a3
GASCEA 304
客户机B的虚拟地址空间B
客户机虚拟页B1.GV 320b1
客户机虚拟页B2.GV 320b2
GASCEB 308
客户机DAT转换
客户机绝对页A2.HV 340a2
客户机绝对页A3.HV 340a3
客户机绝对页A1.HV 340a1
客户机A的绝对地址空间A
HASCE 350
主机虚拟地址空间
客户机B的绝对地址空间B
客户机绝对页B1.HV 360b1
客户机绝对页B2.HV 360b2
主机DAT转换
虚拟机监控程序（主机）的绝对地址空间
主机绝对页A3.HA 370a3
主机绝对页A1.HA 370a1
主机绝对页B1.HA 370b1
主机绝对页AB2.HA 380

图3

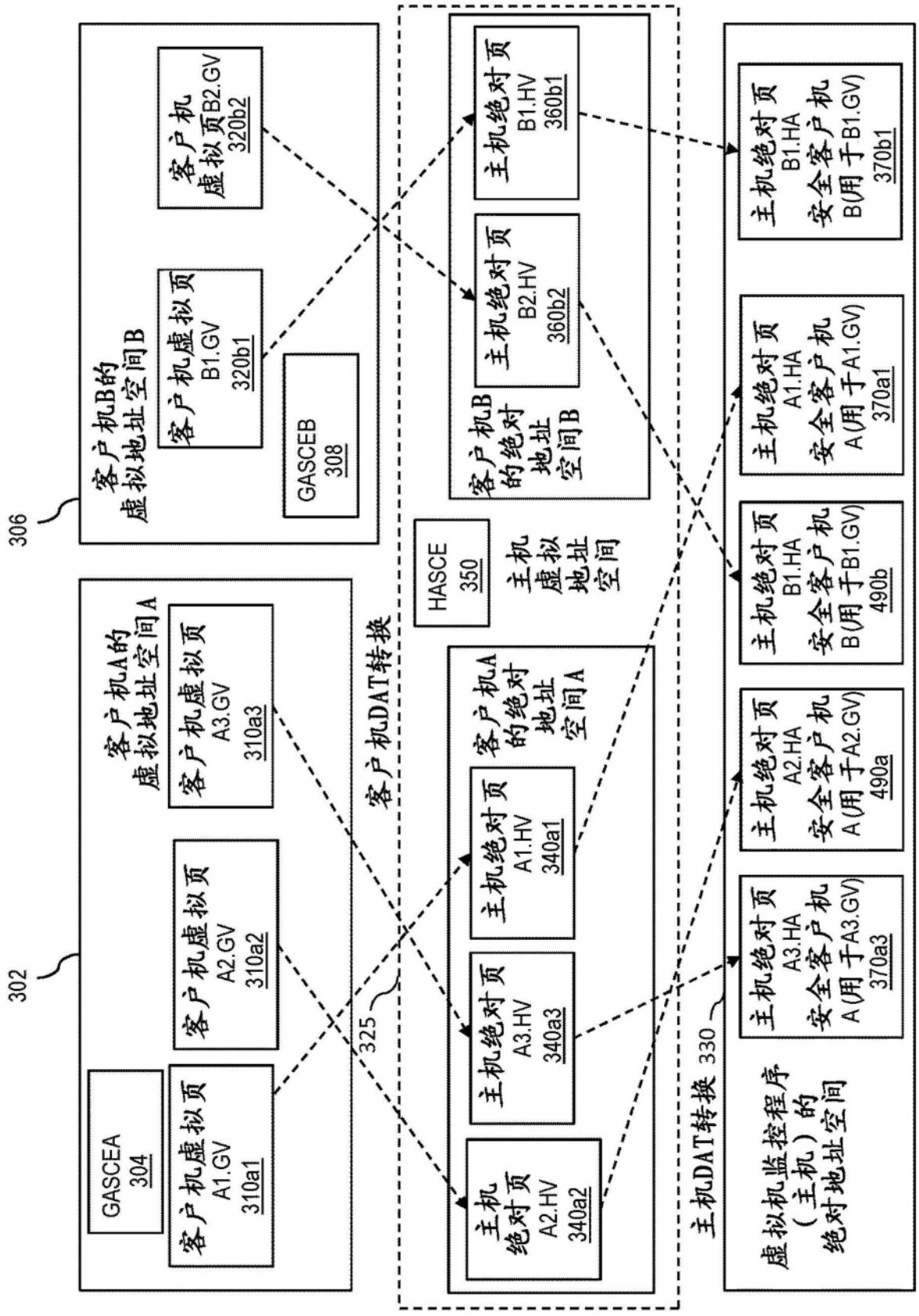

302
306
客户机A的虚拟地址空间A
客户机虚拟页A1.GV 310a1
客户机虚拟页A2.GV 310a2
客户机虚拟页A3.GV 310a3
GASCEA 304
客户机B的虚拟地址空间B
客户机虚拟页B1.GV 320b1
客户机虚拟页B2.GV 320b2
GASCEB 308
客户机DAT转换
325
HASCE 350
主机虚拟地址空间
客户机A的绝对地址空间A
主机绝对页A2.HV 340a2
主机绝对页A3.HV 340a3
主机绝对页A1.HV 340a1
客户机B的绝对地址空间B
主机绝对页B2.HV 360b2
主机绝对页B1.HV 360b1
主机DAT转换 330
虚拟机监控程序（主机）的绝对地址空间
主机绝对页A3.HA 安全客户机A(用于A3.GV) 370a3
主机绝对页A2.HA 安全客户机A(用于A2.GV) 490a
主机绝对页B1.HA 安全客户机B(用于B1.GV) 490b
主机绝对页A1.HA 安全客户机A(用于A1.GV) 370a1
主机绝对页B1.HA 安全客户机B(用于B1.GV) 370b1

图4

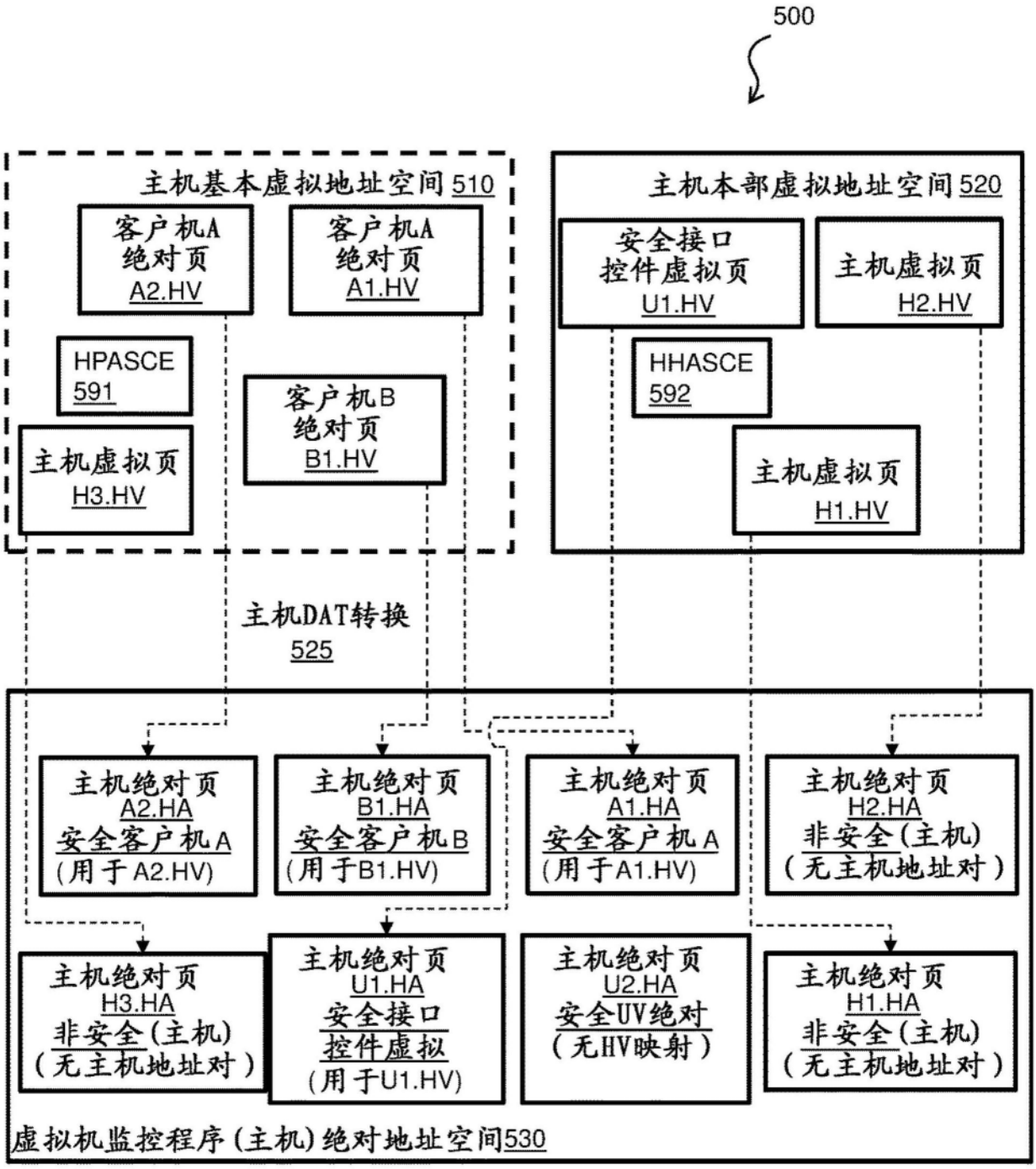

500
主机基本虚拟地址空间 510
客户机A 绝对页 A2.HV
客户机A 绝对页 A1.HV
HPASCE 591
客户机B 绝对页 B1.HV
主机虚拟页 H3.HV
主机本部虚拟地址空间 520
安全接口 控件虚拟页 U1.HV
主机虚拟页 H2.HV
HHASCE 592
主机虚拟页 H1.HV
主机DAT转换 525
主机绝对页 A2.HA 安全客户机A (用于A2.HV)
主机绝对页 B1.HA 安全客户机B (用于B1.HV)
主机绝对页 A1.HA 安全客户机A (用于A1.HV)
主机绝对页 H2.HA 非安全(主机) (无主机地址对)
主机绝对页 H3.HA 非安全(主机) (无主机地址对)
主机绝对页 U1.HA 安全接口 控件虚拟 (用于U1.HV)
主机绝对页 U2.HA 安全UV绝对 (无HV映射)
主机绝对页 H1.HA 非安全(主机) (无主机地址对)
虚拟机监控程序(主机)绝对地址空间530

图5

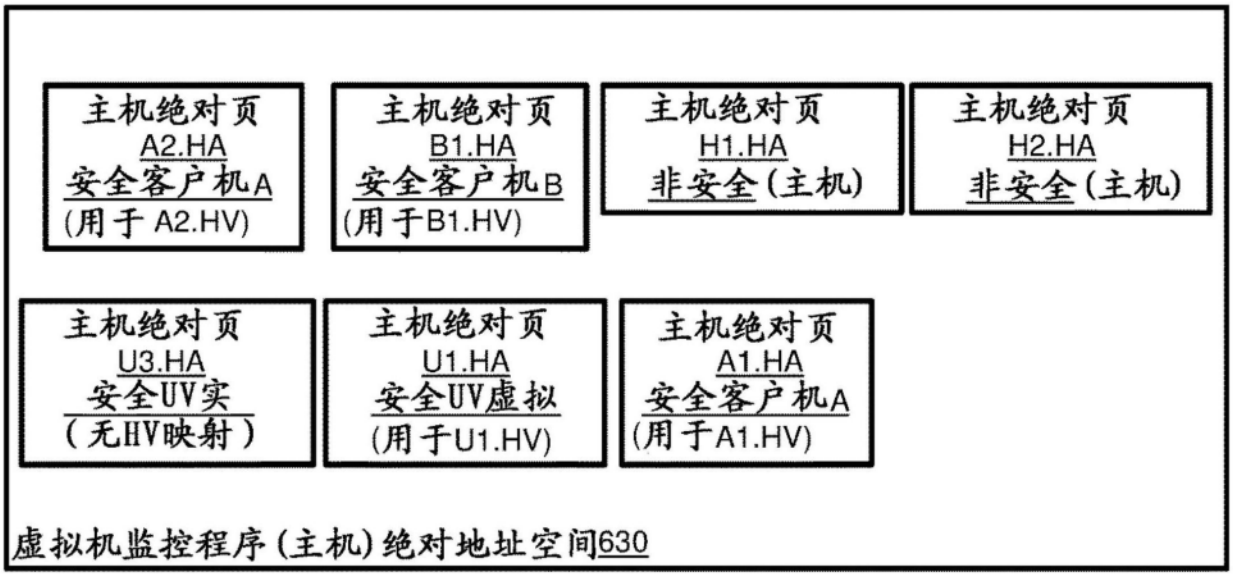

主机绝对页
A2.HA
安全客户机A
(用于 A2.HV)
主机绝对页
B1.HA
安全客户机B
(用于B1.HV)
主机绝对页
H1.HA
非安全(主机)
主机绝对页
H2.HA
非安全(主机)
主机绝对页
U3.HA
安全UV实
(无HV映射)
主机绝对页
U1.HA
安全UV虚拟
(用于U1.HV)
主机绝对页
A1.HA
安全客户机A
(用于A1.HV)
虚拟机监控程序(主机)绝对地址空间630

图6

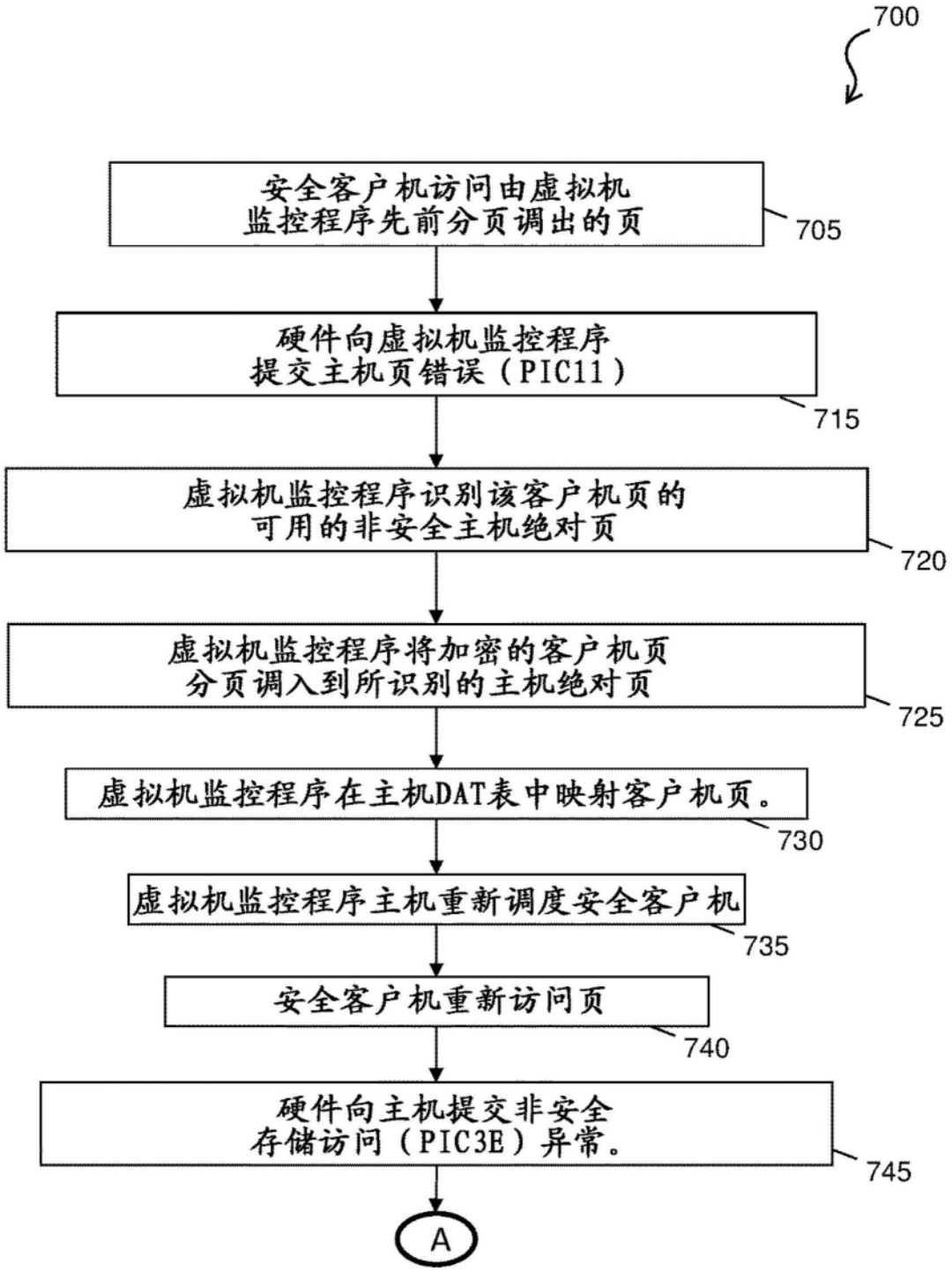

700
安全客户机访问由虚拟机监控程序先前分页调出的页
705
硬件向虚拟机监控程序提交主机页错误（PIC11）
715
虚拟机监控程序识别该客户机页的可用的非安全主机绝对页
720
虚拟机监控程序将加密的客户机页分页调入到所识别的主机绝对页
725
虚拟机监控程序在主机DAT表中映射客户机页。
730
虚拟机监控程序主机重新调度安全客户机
735
安全客户机重新访问页
740
硬件向主机提交非安全存储访问（PIC3E）异常。
745
A

图7

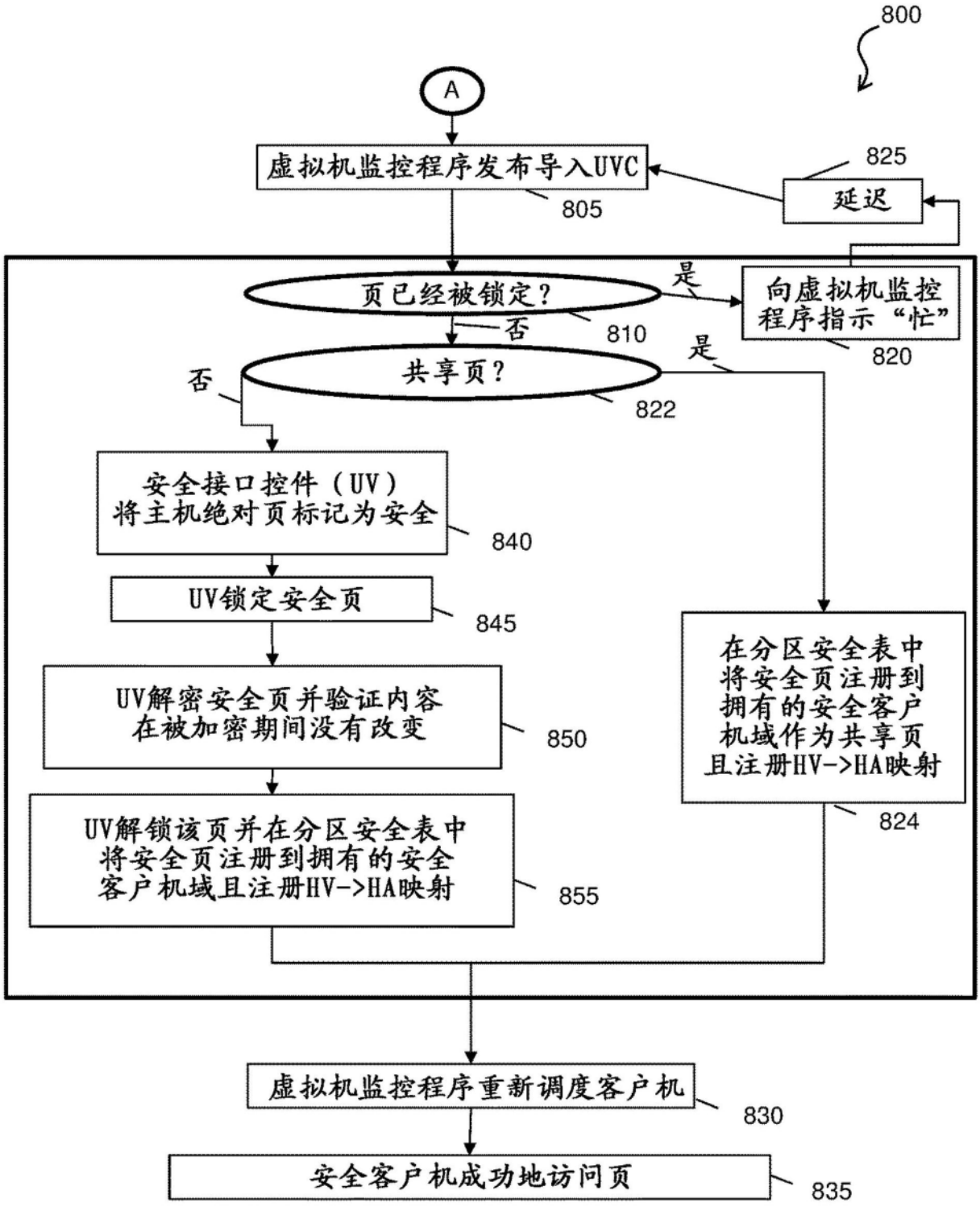

800
A
虚拟机监控程序发布导入UVC
805
825
延迟
页已经被锁定？
是
向虚拟机监控程序指示“忙”
820
否
810
共享页？
是
否
822
安全接口控件（UV）将主机绝对页标记为安全
840
UV锁定安全页
845
UV解密安全页并验证内容在被加密期间没有改变
850
在分区安全表中将安全页注册到拥有的安全客户机域作为共享页且注册HV->HA映射
824
UV解锁该页并在分区安全表中将安全页注册到拥有的安全客户机域且注册HV->HA映射
855
虚拟机监控程序重新调度客户机
830
安全客户机成功地访问页
835

图8

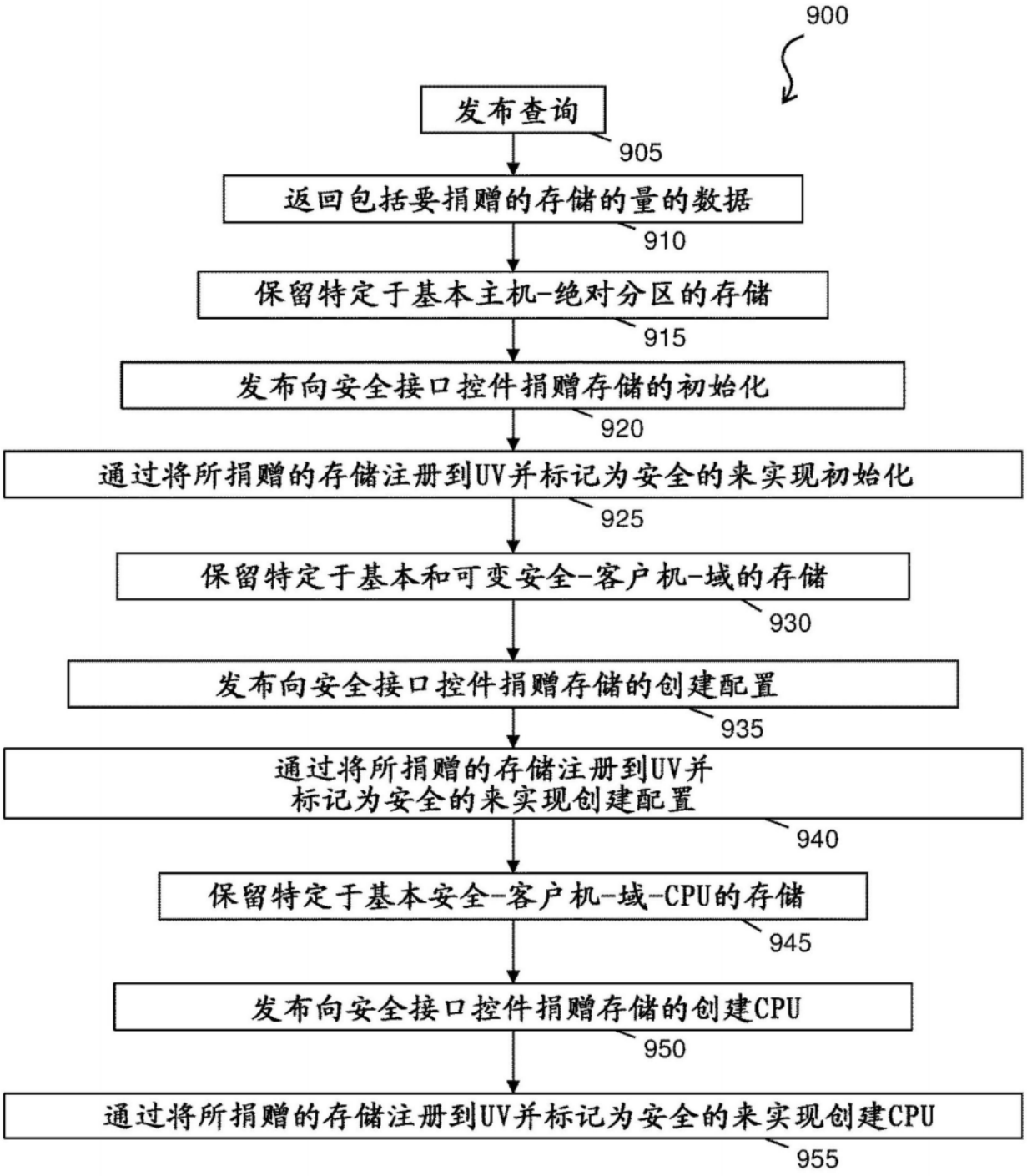

900
发布查询
905
返回包括要捐赠的存储的量的数据
910
保留特定于基本主机-绝对分区的存储
915
发布向安全接口控件捐赠存储的初始化
920
通过将所捐赠的存储注册到UV并标记为安全的来实现初始化
925
保留特定于基本和可变安全-客户机-域的存储
930
发布向安全接口控件捐赠存储的创建配置
935
通过将所捐赠的存储注册到UV并
标记为安全的来实现创建配置
940
保留特定于基本安全-客户机-域-CPU的存储
945
发布向安全接口控件捐赠存储的创建CPU
950
通过将所捐赠的存储注册到UV并标记为安全的来实现创建CPU
955

图9

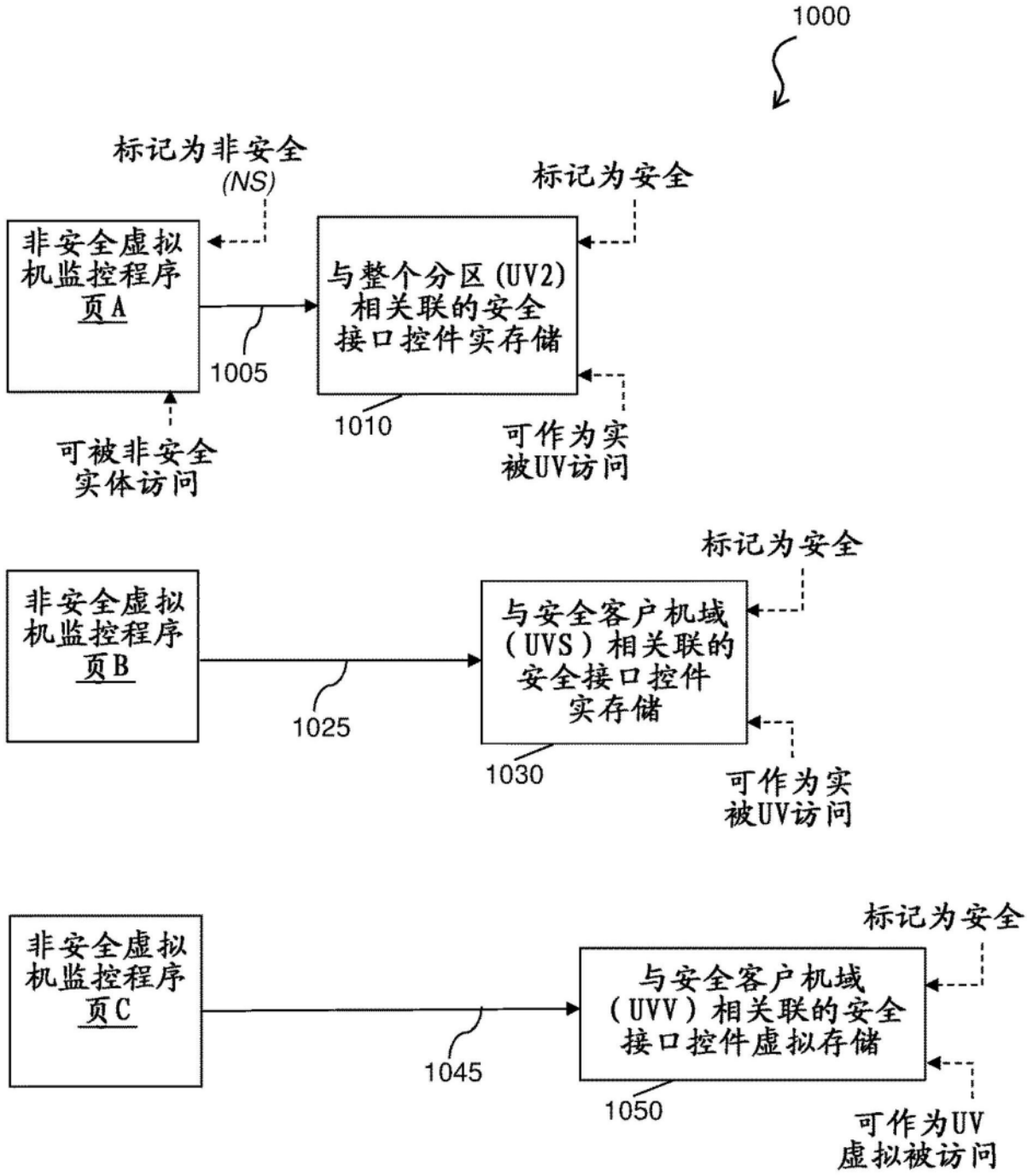

1000
标记为非安全
(NS)
标记为安全
非安全虚拟机监控程序
页A
与整个分区(UV2)相关联的安全接口控件实存储
1005
1010
可被非安全实体访问
可作为实被UV访问
标记为安全
非安全虚拟机监控程序
页B
与安全客户机域(UVS)相关联的安全接口控件实存储
1025
1030
可作为实被UV访问
标记为安全
非安全虚拟机监控程序
页C
与安全客户机域(UVV)相关联的安全接口控件虚拟存储
1045
1050
可作为UV虚拟被访问

图10

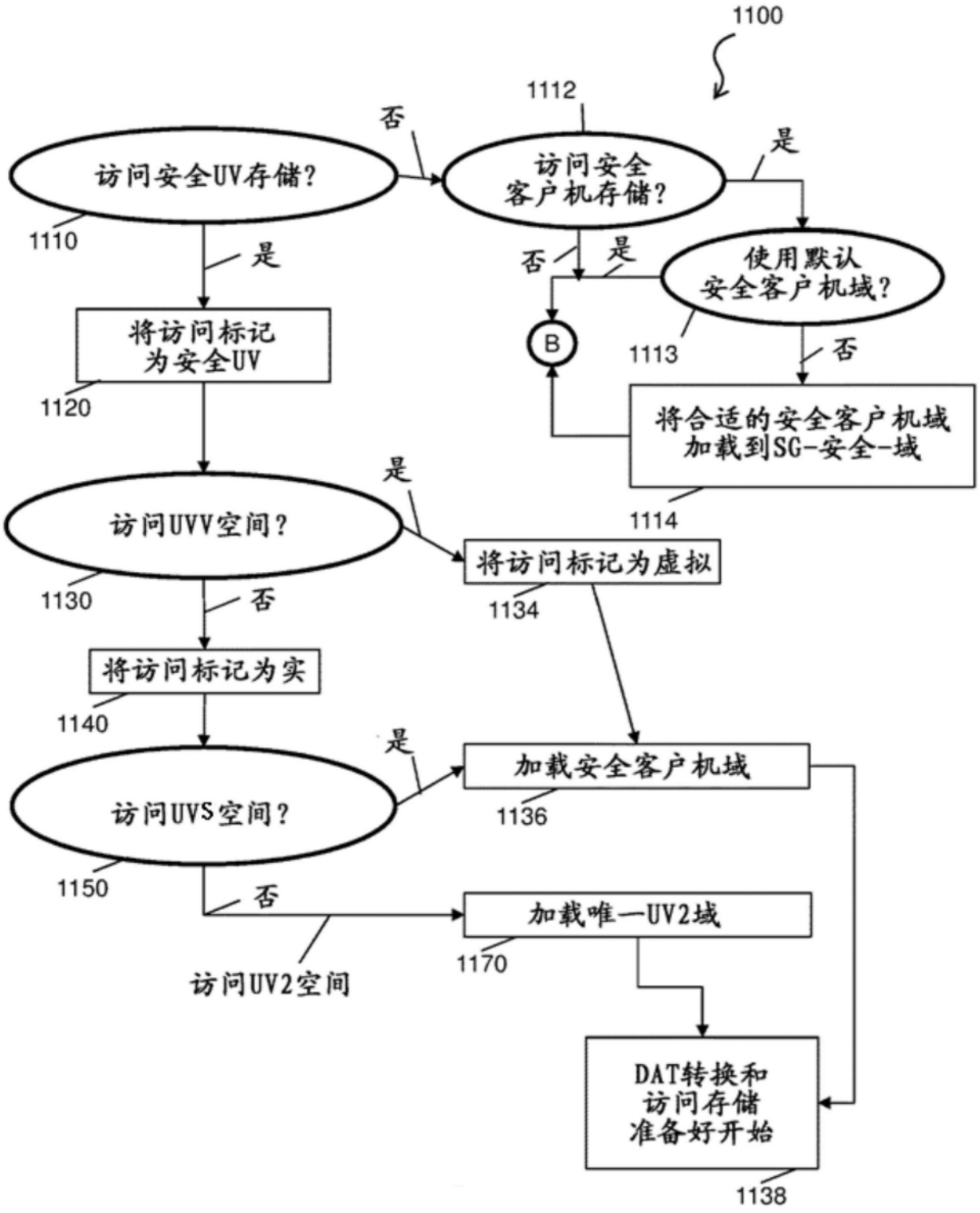

1100
1112
否
访问安全UV存储?
1110
是
访问安全
客户机存储?
是
否
是
使用默认
安全客户机域?
1113
否
B
将访问标记
为安全UV
1120
将合适的安全客户机域
加载到SG-安全-域
1114
是
访问UVV空间?
1130
否
将访问标记为虚拟
1134
将访问标记为实
1140
是
访问UVS空间?
1150
否
加载安全客户机域
1136
加载唯一UV2域
1170
访问UV2空间
DAT转换和
访问存储
准备好开始
1138

图11

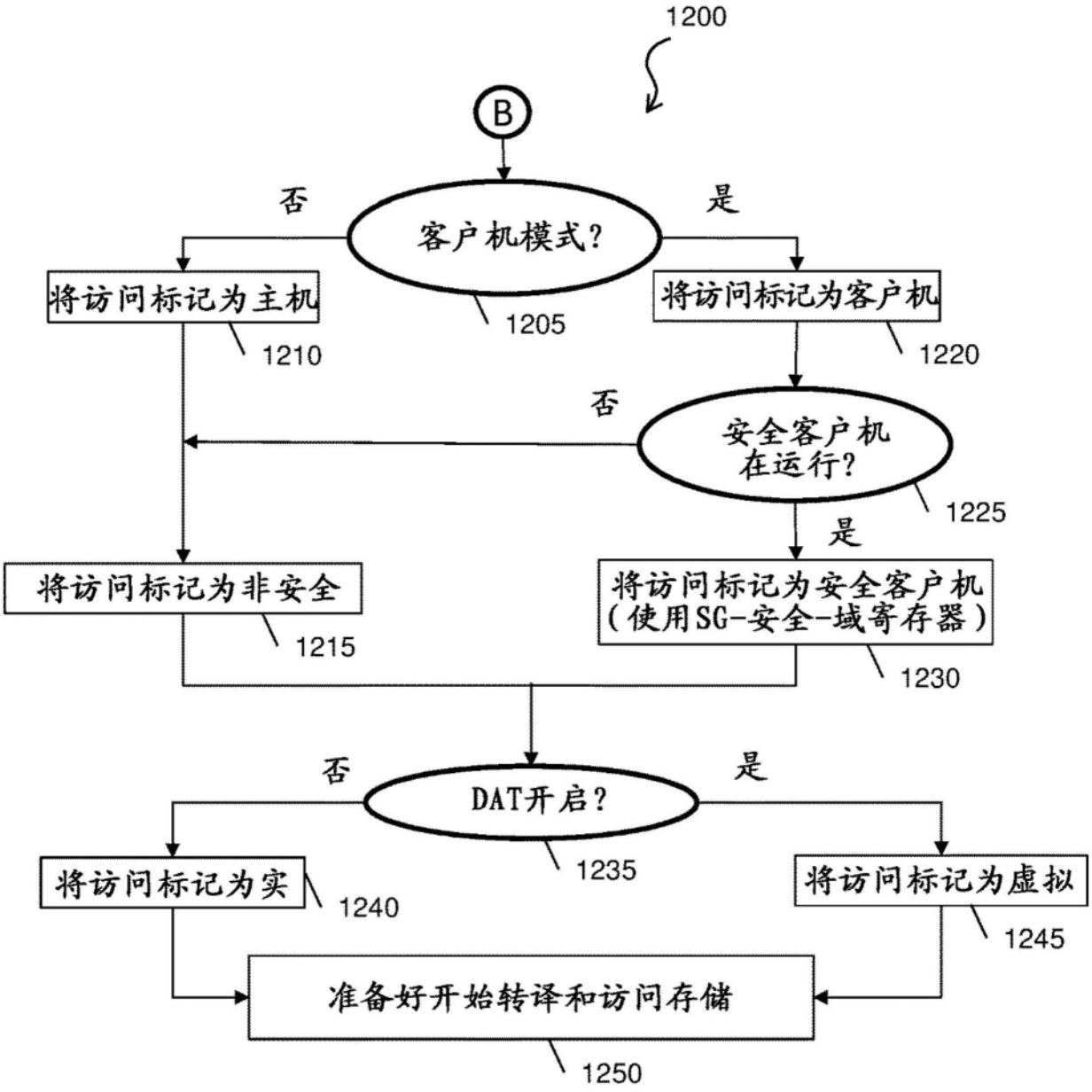

1200
B
客户机模式?
否
是
1205
将访问标记为主机
1210
将访问标记为客户机
1220
安全客户机
在运行?
否
是
1225
将访问标记为非安全
1215
将访问标记为安全客户机
(使用SG-安全-域寄存器)
1230
DAT开启?
否
是
1235
将访问标记为实
1240
将访问标记为虚拟
1245
准备好开始转译和访问存储
1250

图12

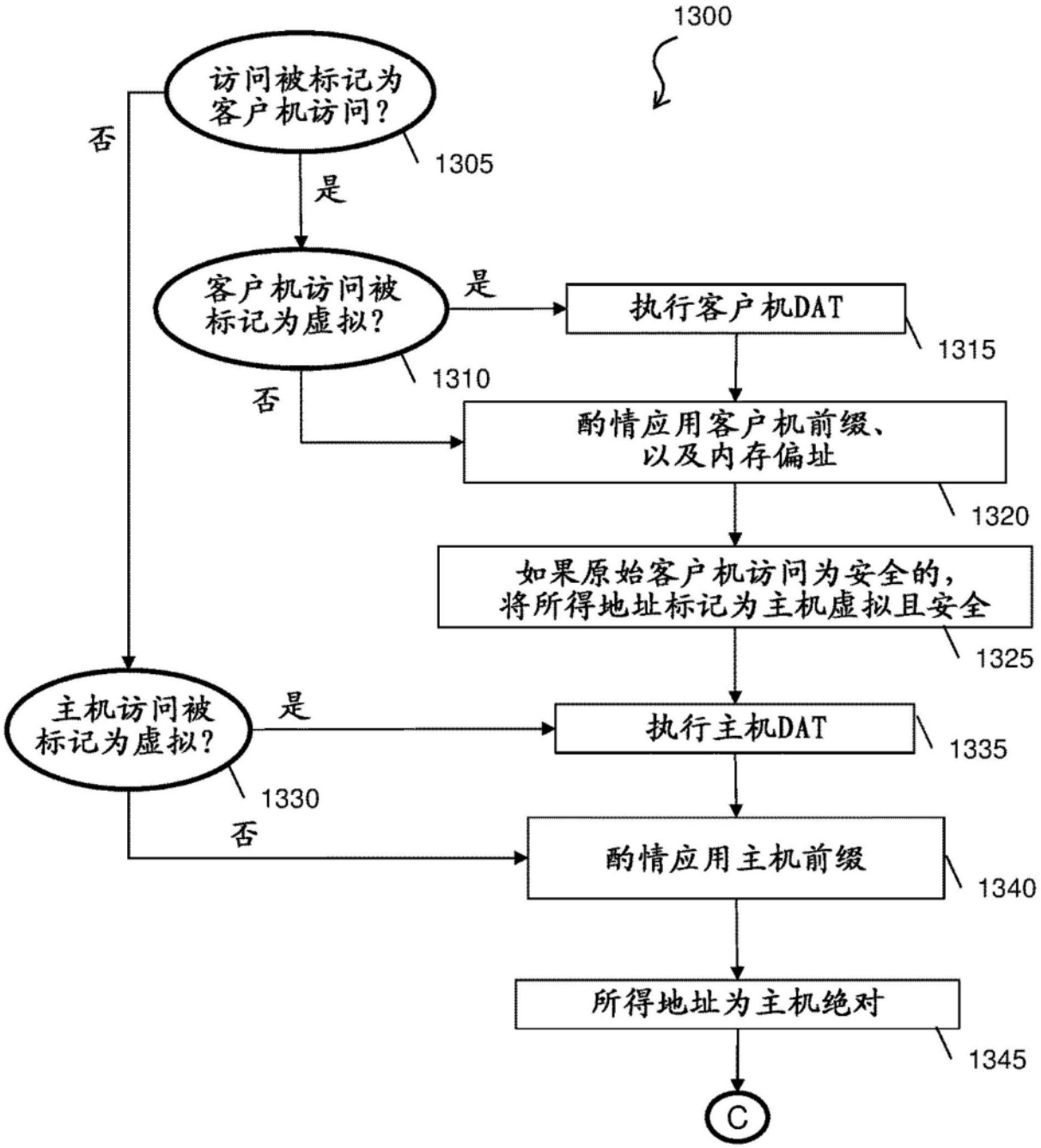

1300
访问被标记为客户机访问?
1305
否
是
客户机访问被标记为虚拟?
1310
是
执行客户机DAT
1315
否
酌情应用客户机前缀、以及内存偏址
1320
如果原始客户机访问为安全的,将所得地址标记为主机虚拟且安全
1325
主机访问被标记为虚拟?
1330
是
执行主机DAT
1335
否
酌情应用主机前缀
1340
所得地址为主机绝对
1345
C

图13

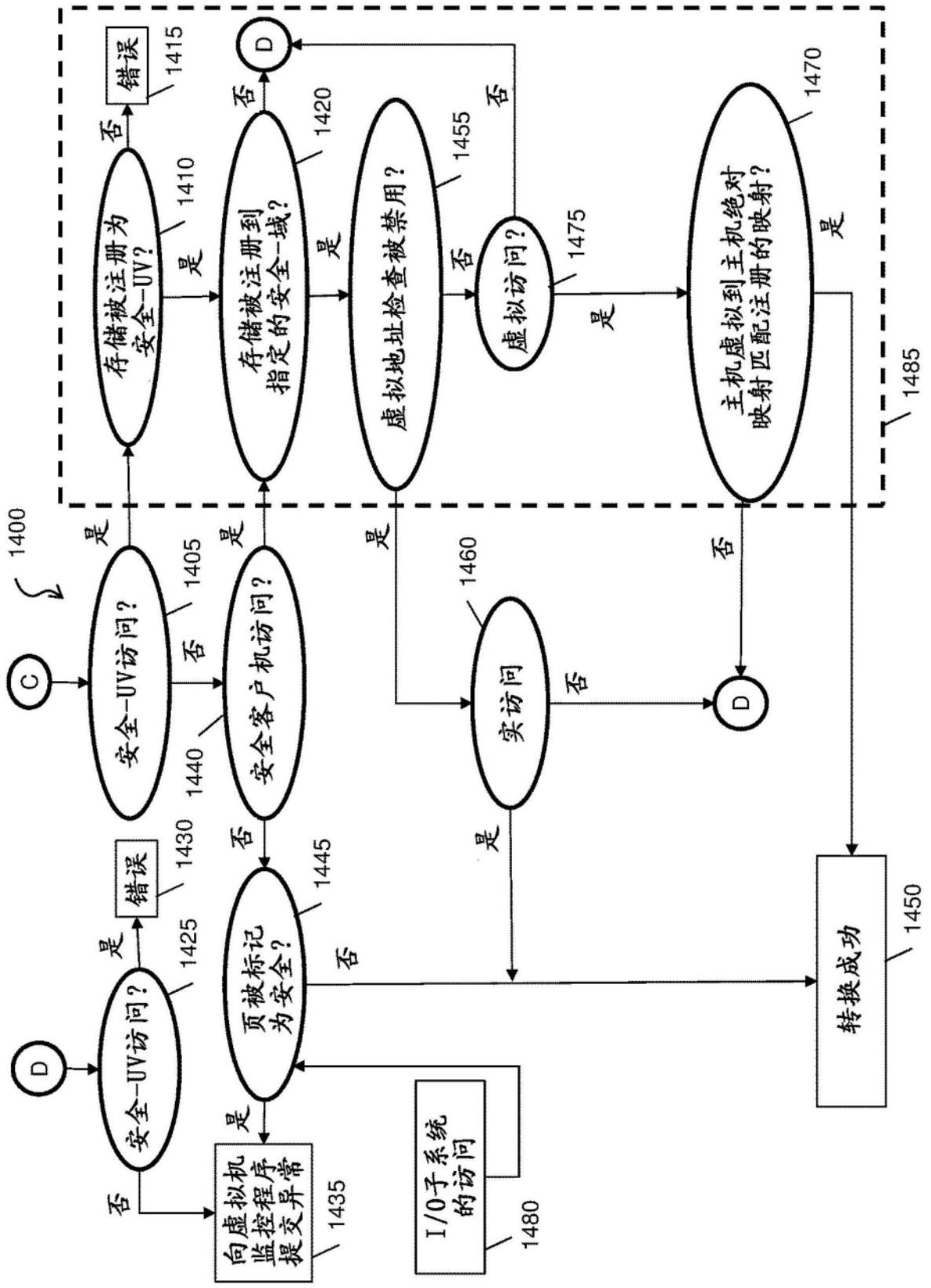

1400
C
安全-UV访问？
1405
是
否
存储被注册为安全-UV？
1410
是
否
错误
1415
安全客户机访问？
1440
是
否
存储被注册到指定的安全-域？
1420
是
否
D
虚拟地址检查被禁用？
1455
是
否
虚拟访问？
1475
是
否
主机虚拟到主机绝对映射匹配注册的映射？
1470
是
否
1485
1460
实访问
否
是
D
D
安全-UV访问？
1425
是
否
错误
1430
页被标记为安全？
1445
是
否
转换成功
1450
向虚拟机监控程序提交异常
1435
I/O子系统的访问
1480

图14

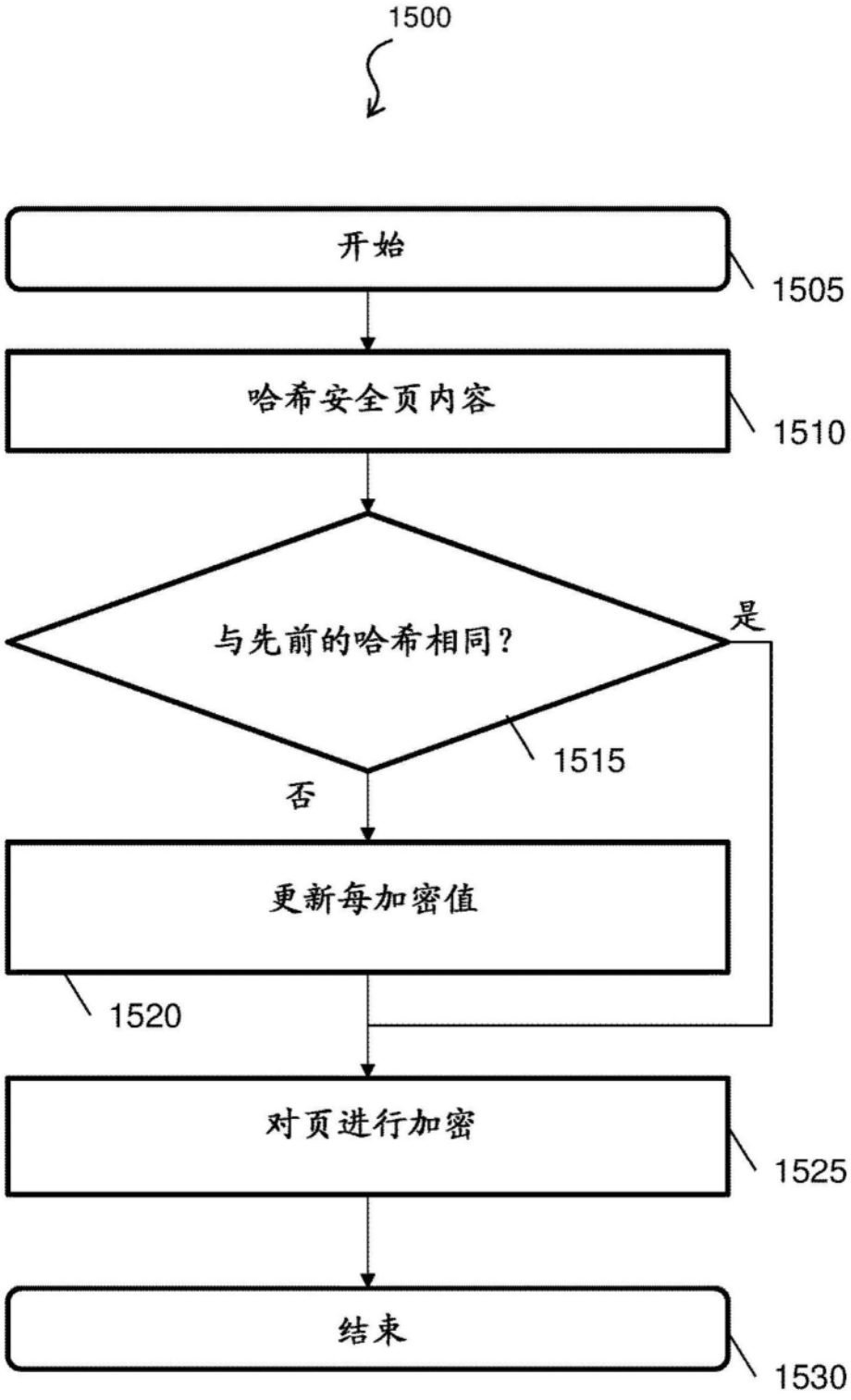

1500
开始
1505
哈希安全页内容
1510
与先前的哈希相同?
是
1515
否
更新每加密值
1520
对页进行加密
1525
结束
1530

图15

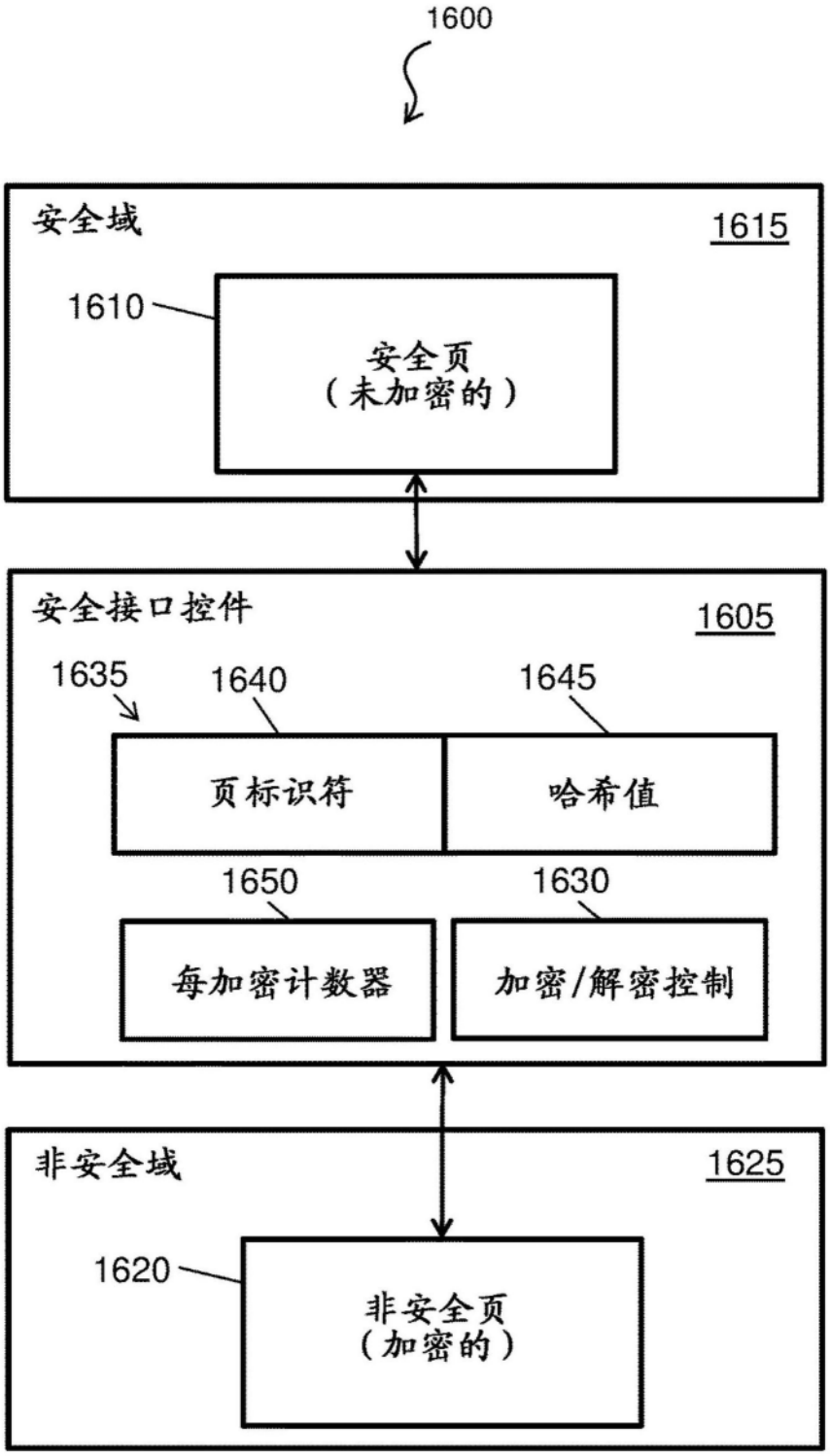

1600
安全域
1615
1610
安全页
(未加密的)
安全接口控件
1605
1635
1640
1645
页标识符
哈希值
1650
1630
每加密计数器
加密/解密控制
非安全域
1625
1620
非安全页
(加密的)

图16

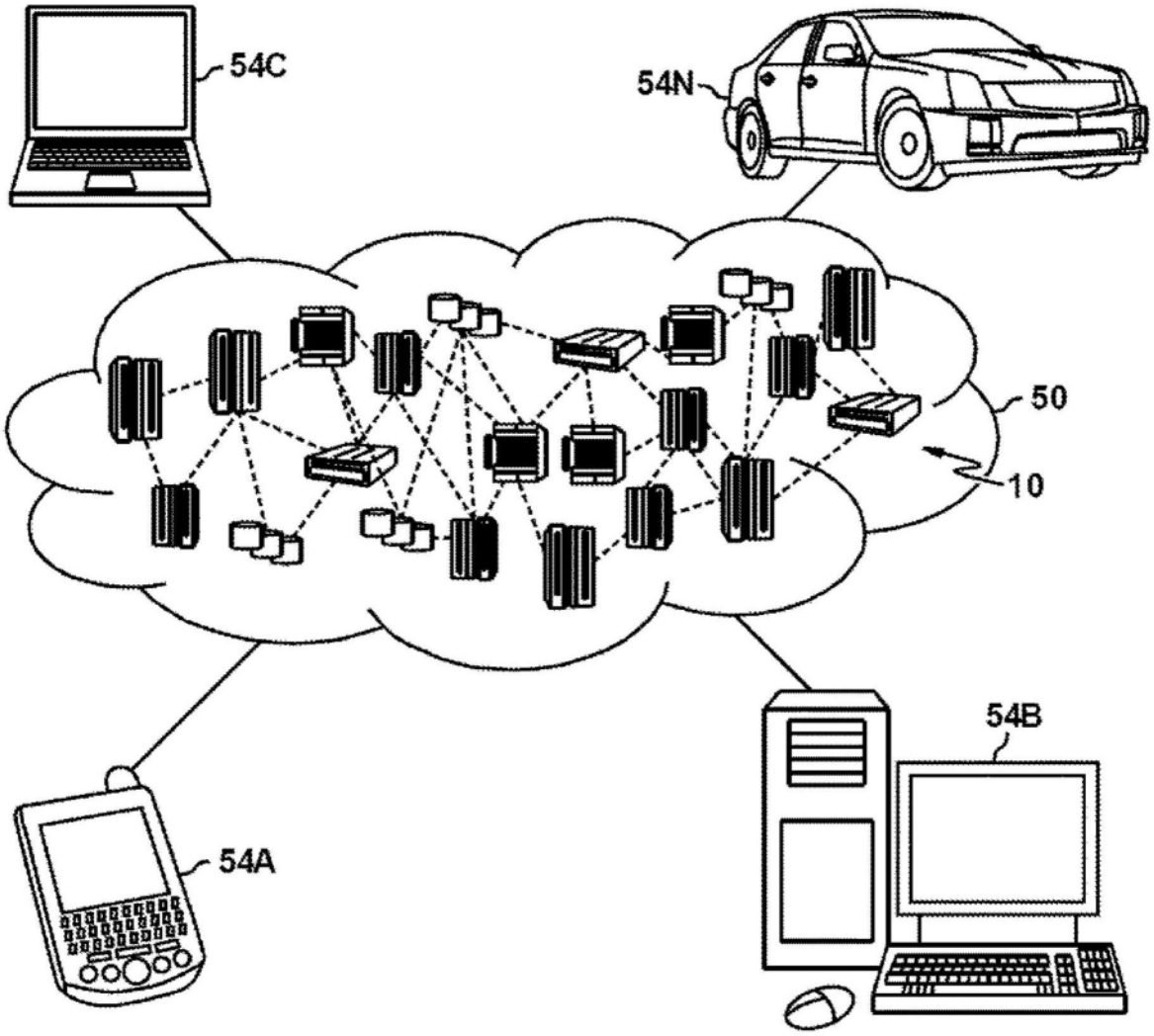


图17

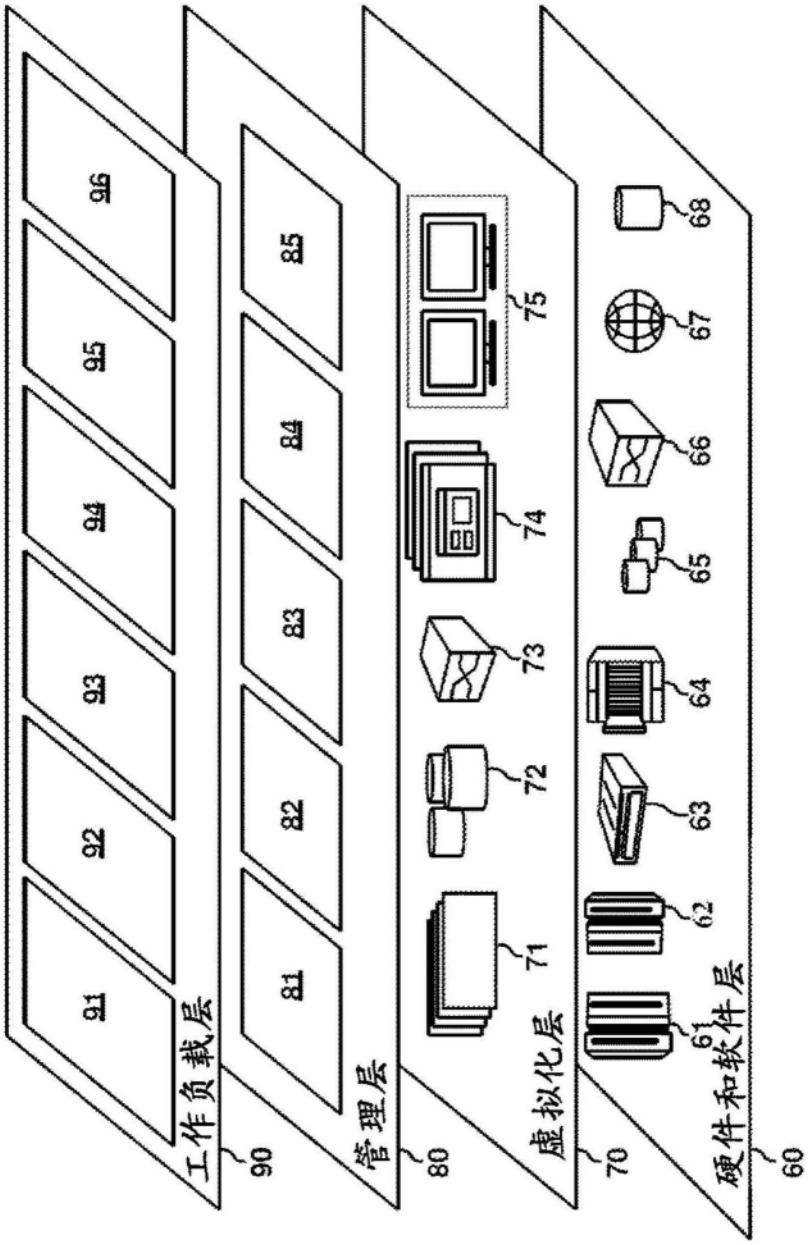

工作负载层
90
91
92
93
94
95
96
管理层
80
81
82
83
84
85
虚拟化层
70
71
72
73
74
75
硬件和软件层
60
61
62
63
64
65
66
67
68

图18

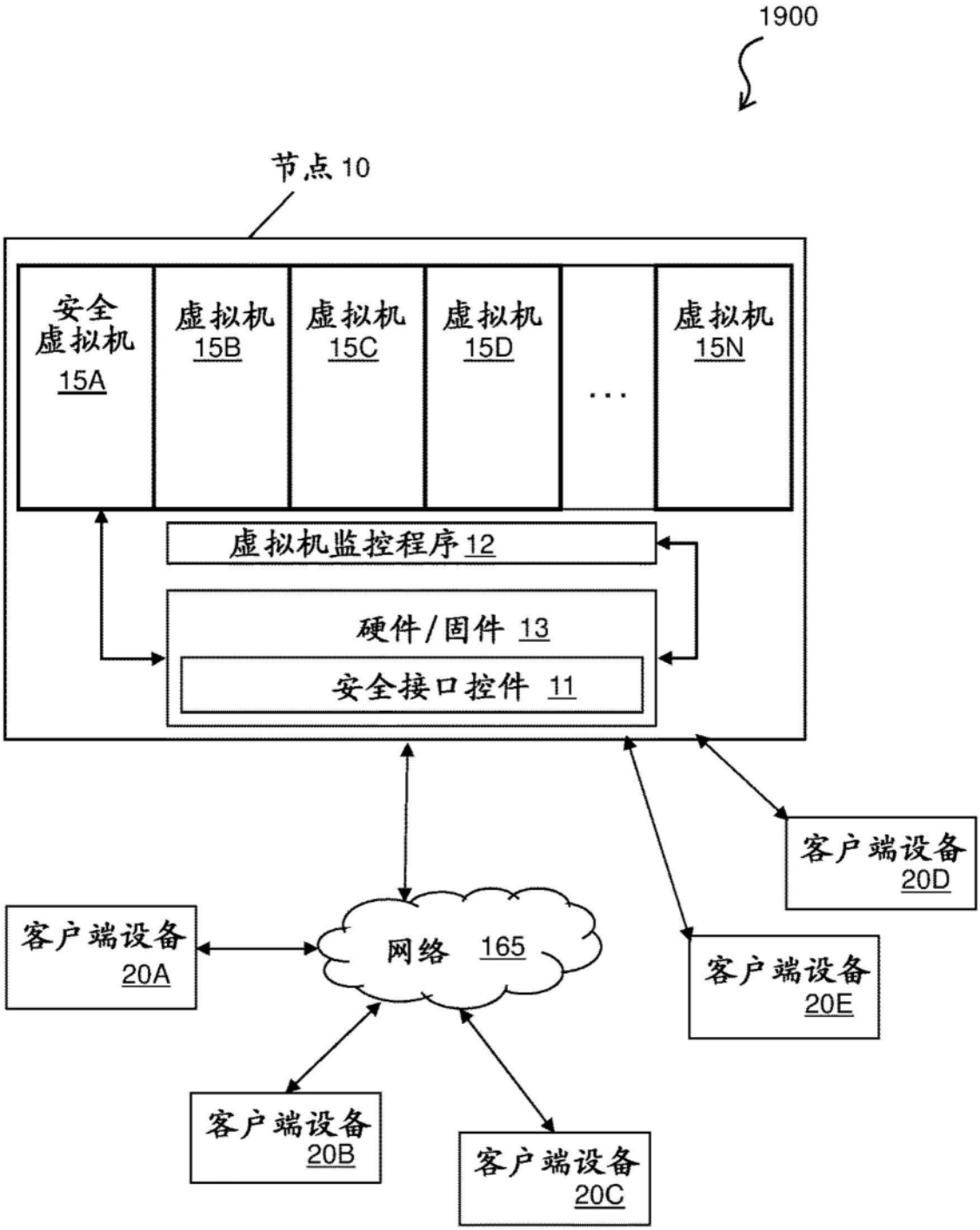

1900
节点 10
安全虚拟机 15A
虚拟机 15B
虚拟机 15C
虚拟机 15D
...
虚拟机 15N
虚拟机监控程序 12
硬件/固件 13
安全接口控件 11
客户端设备 20D
客户端设备 20A
网络 165
客户端设备 20E
客户端设备 20B
客户端设备 20C

图19

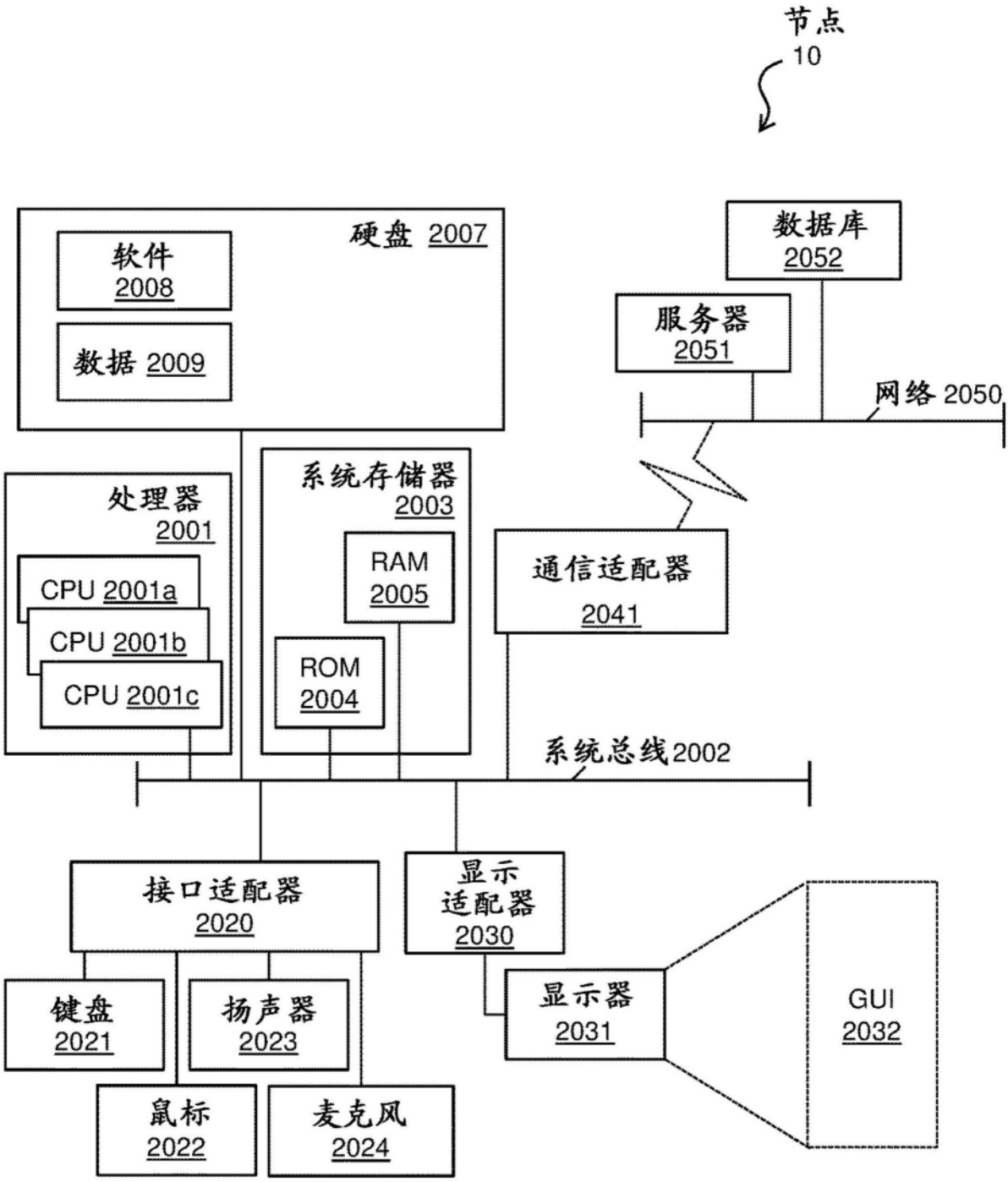

节点
10
硬盘 2007
软件
2008
数据 2009
数据库
2052
服务器
2051
网络 2050
处理器
2001
CPU 2001a
CPU 2001b
CPU 2001c
系统存储器
2003
RAM
2005
ROM
2004
通信适配器
2041
系统总线2002
接口适配器
2020
显示
适配器
2030
键盘
2021
扬声器
2023
显示器
2031
GUI
2032
鼠标
2022
麦克风
2024

图20