[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/455 (2006.01)
G06F 9/38 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200710303581.6

[43] 公开日 2008 年 9 月 3 日

[11] 公开号 CN 101256503A

[22] 申请日 2007.12.21
[21] 申请号 200710303581.6
[30] 优先权
[32] 2006.12.22 [33] US [31] 11/615,821
[71] 申请人 英特尔公司
地址 美国加利福尼亚
[72] 发明人 K·列维特－古列维奇
B·乌里埃尔

[74] 专利代理机构 永新专利商标代理有限公司
代理人 林锦辉

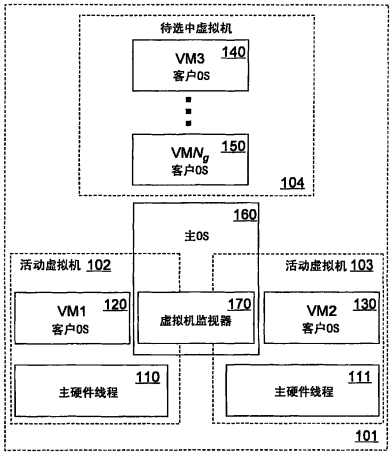
权利要求书 4 页 说明书 6 页 附图 4 页

[54] 发明名称

通过多线程主虚拟机监视器来实现多线程客户操作系统执行的方法和装置

[57] 摘要

用于对客户操作系统(OS)进行虚拟化的方法和装置，包括分配虚拟机(VM)。 虚拟机被分配给各个可用的硬件线程，并被恢复或首次启动。 如果需要在退出虚拟机环境时调度另一虚拟机，则选择另一虚拟机，并将该另一虚拟机分配给硬件线程以恢复或启动。 在防止主操作系统对硬件线程进行控制时，使用虚拟机监视器恢复或启动虚拟机。 加载虚拟机环境结构。 线程被设置为客户操作系统的状态，并且恢复或启动虚拟机环境。 在进行到虚拟机监视器的环境切换期间，在本地保存客户操作系统的状态。 然后，清除虚拟机环境结构，并且允许主操作系统再次对硬件线程进行控制。



1、一种用于恢复或启动虚拟机的计算机化方法，所述方法包括：

禁止主操作系统对多个硬件线程中的第一硬件线程进行控制；

加载第一虚拟机环境结构；

将所述第一硬件线程设置成第一客户操作系统的状态；

恢复或启动第一虚拟机环境；

在本地保存所述第一客户操作系统的状态；

执行到虚拟机监视器环境的环境切换；

从所述第一硬件线程中清除所述第一虚拟机环境结构；以及

设置中断标志以允许所述主操作系统对所述第一硬件线程进行控制。

2、如权利要求 1 所述的方法，其中禁止所述主操作系统对所述第一硬件线程进行控制是通过清除中断标志实现的。

3、如权利要求 2 所述的方法，其中允许所述主操作系统对所述第一硬件线程进行控制是通过设置所述中断标志实现的。

4、一种制品，包括：

机器可访问介质，其包括数据，当被机器访问时，所述数据使所述机器执行如权利要求 3 所述的方法。

5、如权利要求 1 所述的方法，其中加载第一虚拟机环境结构包括：执行 VMPTRLD 指令，以将所述第一虚拟机环境结构标记为有效，并从存储器中的特定地址对其进行加载。

6、如权利要求 5 所述的方法，其中将所述第一硬件线程设置成第一客户操作系统的状态包括：加载所述虚拟机环境结构的客户状态区域。

7、一种制品，包括：

机器可访问介质，其包括数据，当被机器访问时，所述数据使所述机器执行如权利要求 6 所述的方法。

8、一种用于在多处理主平台上对一个或多个多处理客户操作系统进行虚拟化的计算机化方法，所述方法包括：

分配第一多个虚拟机；

分配第二多个硬件线程；

为所述第二多个硬件线程中的每一个硬件线程，选择所述第一多个虚拟机中的一个虚拟机以将其分配给该硬件线程；

恢复或启动分配给所述第二多个硬件线程的虚拟机；

在退出所述第二多个硬件线程的每一个硬件线程的虚拟机环境时，确定是否需要调度另一个虚拟机；

如果需要调度，则选择所述第一多个虚拟机中的另一个虚拟机，以将其分配给该硬件线程；并且

恢复或启动分配给该硬件线程的虚拟机。

9、如权利要求 8 所述的方法，其中从所述第一多个虚拟机中选择一个虚拟机并将其分配给该硬件线程以及恢复或启动虚拟机，是在虚拟机监视器的环境内部执行的。

10、如权利要求 9 所述的方法，其中由所述虚拟机监视器执行的恢复或启动所述虚拟机，是在禁止中断的受保护的临界区内执行的。

11、一种制品，包括：

机器可访问的有形介质，其包括可执行指令，当由机器访问时，所述可执行指令使所述机器执行：

清除中断标志，以禁止主操作系统对多个硬件线程中的第一硬件线程进行控制；

加载第一虚拟机环境结构；

将所述第一硬件线程设置成第一客户操作系统的状态；

恢复或启动第一虚拟机环境；

在本地保存所述第一客户操作系统的状态；

执行到虚拟机监视器环境的环境切换；

从所述第一硬件线程中清除所述第一虚拟机环境结构；以及

设置所述中断标志以允许所述主操作系统对所述第一硬件线程进行控制。

12、如权利要求 11 所述的制品，其中所述有形介质包括用于将所述第一虚拟机环境结构标记为有效，并从存储器中的特定地址对其进行加载的可执行指令。

13、如权利要求 11 所述的制品，其中通过加载所述虚拟机环境结构的客户状态区域，将所述第一硬件线程设置成所述第一客户操作系统的状态。

14、一种计算系统，包括：

处理器，其包括多个硬件执行线程；

主操作系统，其可在处理器上执行；

多线程客户操作系统，其可在所述处理器的多个虚拟机上执行；

多线程虚拟机监视器，其可在所述主操作系统下，在所述处理器上执行，以调度所述多个硬件执行线程上的虚拟机，所述多线程虚拟机监视器用于：

清除中断标志，以禁止所述主操作系统对所述多个硬件执行线程中的第一硬件执行线程进行控制；

为所述多个虚拟机中之一加载第一虚拟机环境结构；

将所述第一硬件执行线程设置成所述多线程客户操作系统的第一状态；

恢复或启动第一虚拟机环境；

在本地保存所述多线程客户操作系统的第一状态；

执行到所述多线程虚拟机监视器环境的环境切换；

从所述第一硬件执行线程中清除所述第一虚拟机环境结构；以及

设置所述中断标志以允许所述主操作系统对所述第一硬件执行线程进行控制。

15、如权利要求 14 所述的计算系统，其中所述多线程虚拟机监视器包括用于将所述第一虚拟机环境结构标记为有效，并从存储器中的特定地址对其进行加载的可执行指令。

16、如权利要求 15 所述的计算系统，其中通过加载所述虚拟机环境结构的客户状态区域，将所述第一硬件线程设置成所述第一客户操作系统的状态。

通过多线程主虚拟机监视器来实现多线程客户操作系统执行的方法和装置

技术领域

本公开一般涉及多线程微处理器领域。具体地，本公开涉及在主操作系统中使用虚拟机监视器，以在多线程处理器执行线程时对多操作系统的执行进行调度。

背景技术

众所周知，管理程序(hypervisor)是一种在大型机（mainframe）上执行多操作系统的早期技术。管理程序允许多个部门计算机合并成一个单独的大型计算机，并通过对超级用户（在操作系统内核时所使用的术语）的状态进行虚拟化来同时运行多操作系统。多个操作系统中的每一个都在虚拟机（VM）上运行，并且即使一个操作系统崩溃，其余的操作系统也将继续工作。

今天高端服务器经常支持虚拟化和一些形式的管理程序技术，但是经常会是服务器级的价格（多达几百万美元）。

过去，基于微处理器的平台已经支持一些类型的虚拟机环境，但是经常用于支持不同操作系统的应用程序，而不是支持同时执行多个操作系统。通过管理程序进行的个人计算机的完全虚拟化需要在复杂性和运行时间性能上付出非常大的代价。

替代方案需要修改客户操作系统，以使系统调用（有时称作超级调用、诊断代码或部分虚拟化）到管理程序，而不是执行机器 I/O 指令。然后通过管理程序模拟这样的准虚拟化（paravirtualization）调用。

一些微处理器供应商也已经引进了硬件虚拟化支持。例如，由加州 Santa Clara 市的 Intel 公司开发的 Vanderpool 技术（VT），对一些虚拟化辅助提供体系级和指令级支持，所述虚拟化辅助否则会效率不高或者需要修改客户操作系统。另外，目前的一些微处理器为多线程、同时发生的多线程和/或

多处理器内核提供支持，这在理论上可以提供多操作系统的更高性能的执行。到现在为止，还没有完全利用这些并行硬件和虚拟化特性的优点。

附图说明

本发明通过实施例进行说明，但并不局限于所附附图。

图 1 描述使用虚拟机(VM)监视器执行客户操作系统(OS)的多处理系统的一个实施例。

图 2 描述在多处理主平台的可用硬件线程上对一个或多个多处理客户操作系统进行虚拟化的处理过程的一个实施例的流程图。

图 3 描述在多线程主平台上的临界区（critical section）内部对使用虚拟机(VM)监视器的虚拟机(VM)的启动顺序进行保护的处理过程的一个实施例的流程图。

图 4 描述用于对一个或多个多处理客户操作系统进行虚拟化和在临界区内部保护使用虚拟机(VM)监视器的虚拟机(VM)的启动顺序的多处理系统的另一个实施例。

具体实施方式

此处所公开的是用于在多处理主平台上对一个或多个多处理客户操作系统（OS）进行有效虚拟化的过程和装置。虚拟机被分配给每个多处理客户 OS 处理器。还从主多处理系统中可用的硬件执行线程中分配硬件执行线程。每个硬件执行线程分配有其中一个虚拟器，并且恢复这些虚拟机（如果之前已启动）或者否则对这些虚拟机进行首次启动。如果在退出虚拟机环境时需要调度另一虚拟机，则选择另一虚拟机，并且将当前的硬件线程分配给该另一虚拟机以便恢复或首次启动。

此外，还公开了用于通过在专用临界区内部使用虚拟机监视器启动或恢复虚拟机环境来支持客户操作系统执行的过程和装置。清除中断标志以禁止主操作系统对多个硬件线程中的一个硬件线程进行控制。然后，加载虚拟机环境结构，并且将硬件线程设置成客户操作系统的状态。虚拟机环境或者恢复或者首次启动。然后，在本地保存客户操作系统的状态，并且

进行到虚拟机监视器环境的环境切换。从硬件线程中清除虚拟机环境结构，然后将中断标志设置为允许主操作系统再次控制硬件线程。

通过应用所公开的过程和装置的实施例，虚拟机可以利用所有可用的主机硬件来运行多处理器客户操作系统。因此，可以提高虚拟化客户软件的性能以及客户执行线程的同步性。可在平台模拟器中应用一些公开的实施例，以提高多线程软件应用程序的性能。

本发明的这些和其它实施例可以根据以下的教导进行实现，显然可以在不脱离本发明的精神和范围的情况下对下述教导作出不同的修改和改变。因此，说明书和附图被认为是一种描述性的而非限制性的，本发明仅仅根据权利要求及其等价物来限定。

一些实施例可使用 Intel®的 Vanderpool 技术（IA-32 处理器的 Intel®Vanderpool 技术（VT-x）规范草案，序号 C97063-001；Intel®安腾(Itanium)处理器的 Intel®Vanderpool 技术（VT-i）规范草案，序号 NO.305942-002；可以通过 FTP 在 download.intel.com/technology/computing/vptech 上得到）。一些已知的结构、电路、体系特定的特性等等没有进行详细描述，以避免对本发明造成不必要的混淆。

图 1 描述了多处理系统 101 的实施例，用于使用虚拟机(VM)管理器 170 在多处理系统 101 的可得到的硬件线程 110-111 上对一个或多个客户多处理操作系统（OS）120-150 进行虚拟化，以执行所述客户 OS 120-150。所述一个或多个客户多处理 OS 包括 Ng 个虚拟处理器，并且因此在初始化期间创建虚拟机：为客户 OS 线程 120 创建 VM1，为客户 OS 线程 130 创建 VM2，为客户 OS 线程 140 创建 VM3，……，以及为客户 OS 线程 150 创建 VMNg。创建虚拟机监视器 170 以在主 OS 160 下运行和管理虚拟机。虚拟机监视器 170 的其中一个任务是在多处理系统 101 的每一个可用的硬件线程 110-111 上并发地启动虚拟机。

可以理解的是，多线程硬件的可用性和如 Vanderpool 技术的体系支持，使得可以对多处理客户操作系统进行有效虚拟化，其可以利用多处理主平台上所有可用的硬件执行线程。

图 2 描述用于在多处理主平台的可用硬件执行线程上对一个或多个多

处理客户操作系统进行虚拟化的处理过程 201 的一个实施例的流程图。处理过程 201 和此处公开的其它处理都是由处理块执行的，所述处理块可能包括专用的硬件或软件或固件操作代码，所述操作代码可以由通用机器或专用机器或两者结合来执行。

在处理块 211 中，为一个或多个客户多处理 OS 的处理器分配 Ng 个虚拟机。在处理块 212 中，从多处理主平台的可用硬件线程中分配 Nh 个硬件执行线程。然后，在处理块 213 中，选择 Ng 个虚拟机中的虚拟机(VM)以分配给所述可用的 Nh 硬件执行线程中的各个硬件执行线程。在处理块 214 中，恢复分配给硬件线程的 Nh 个虚拟机（如果之前已启动），否则首次启动所述 Nh 个虚拟机。在处理块 215 中，开始特定虚拟机 VM 内部的执行。一旦在处理块 216 中退出每一个硬件执行线程的虚拟机 VM 环境，在处理块 217 就进行是否需要调度另一个虚拟机的判断。如果确定为需要调度另一个虚拟机，则在处理块 218 中执行继续，其中选择另一个虚拟机，以分配给特定的硬件执行线程。否则，直接执行处理块 219，其中恢复分配给当前硬件执行线程的虚拟机（如果之前已启动）或进行首次启动。然后，在处理块 215，在特定虚拟机内部执行恢复，随后进行处理块 216-219 的另一次重复操作。

可以理解的是，为了例如在虚拟机监视器正在将一个虚拟机控制结构附加到硬件执行线程上时禁止主 OS 重调度线程，需要禁止中断虚拟机的启动顺序。这样的中断可能对虚拟机监视器、硬件执行线程和/或主 OS 具有潜在的毁坏性影响。因此，在一些实施例中，使用虚拟机监视器进行的虚拟机的启动顺序在禁止中断的受保护的临界区中执行。

图 3 描述了在多线程主平台上的临界区内部对使用虚拟机(VM)监视器的虚拟机(VM)的启动顺序进行保护的处理过程的一个实施例的流程图。在处理块 311 中，禁止主操作系统对硬件执行线程中的一个硬件执行线程进行控制。对于一个实施例，这可以通过清除中断标志（IF）以禁止中断来实现。可以提供一个指令，例如 CLI，以专门清除中断标志。在一些实施例中，在允许清除中断标志之前，可以检查当前优先级。

接着，在处理块 312 中，加载或激活虚拟机环境结构（virtual machine context structure,VMCS）。在一些实施例中，这可以通过执行一个指令

（VMPTRLD）来实现，其中该指令专门用于标记 VMCS 有效且从指向 VMCS 的指针地址进行加载。在一些实施例中，为了完成这样一个指令，可能需要一个指定的优先级（例如零级，在保护模式或在 64 位模式时）。在处理块 313 中，硬件执行线程的处理器状态被设定为客户操作系统的状态。对于一些实施例，在 VM 进入时，可以将 VMCS 中客户状态区域加载到处理器状态，并且在 VM 退出时，可以将处理器状态保存到该客户状态区域。客户 OS 的一些附加状态可以通过对 VM 进入时的特殊控制来确定，其在 VMCS 中设定。其它状态，如页目录指针，可以基于特定控制寄存器的值进行加载。在处理块 314 中恢复虚拟机环境（如果之前已启动），否则首次启动虚拟机环境。在一些实施例中，可以提供虚拟机进入指令（VMRESUME）以专门恢复虚拟机环境，和虚拟机进入指令（VMLAUNCH）以专门启动虚拟机环境。同样，为了完成这些指令，一些实施例可能需要指定的优先级（例如零级）。

在处理块 315 中，在本地保存客户操作系统的状态。对于一些实施例，可以由虚拟机退出引起该操作。在处理块 316 中，进行到虚拟机监视器环境的环境切换。在一些实施例中，可能提供虚拟机退出指令（VMCALL）以专门执行这样的虚拟机监视器环境切换。在替换实施例中，可以通过设置任务门或通过进入调试(debug)模式，如单步，来外部调用任务切换。在其它替换实施例中，事件，例如访问控制寄存器或执行 I/O 指令，可以被设置来触发任务切换。在处理块 317 中，清除硬件执行线程中的 VMCS。在一些实施例中，可以提供指令（VMCLEAR）以专门从硬件执行线程中清除当前 VMCS。最后，在处理块 318 中，中断标志（IF）被复位以允许主操作系统再次控制硬件执行线程。由此，可以通过这样的临界区来保护虚拟机的启动顺序，以防止主 OS 再次调用线程。

图 4 描述了用于对一个或多个多处理客户操作系统 420 进行虚拟化的多处理系统 401 的另一个实施例。多处理系统 401 可以包括用于存储数据和可执行程序的可寻址存储器、本地存储装置 403、高速缓存存储装置 404、图形存储装置、图形控制器和各种系统，所述各种系统可选地包括外设系统、盘片和 I/O 系统、网络系统、外部存储系统，其中所述网络系统包括存储到可寻址存储器中的数据流的网络接口，其中所述外部存储系统包括磁

存储设备，以存储多条软件执行线程的指令，其中通过处理器 402 访问这些指令，从而使得处理器对多条软件执行线程的指令进行处理。本地存储装置 403，例如，可以存储一个或多个客户操作系统 OS 420，所述一个或多个客户操作系统 OS 具有在虚拟处理器上执行的多条软件执行线程。本地存储装置 403 还存储与虚拟处理器相关的客户状态 480 以及多线程主操作系统 460、多线程虚拟机监视器 470 和虚拟机控制结构 490。

关于处理器 201 和 301 在可用的硬件线程 410-411 上对客户操作系统 420 进行虚拟化，多处理系统 401 如上所述地使用虚拟机监视器 470 在受保护的临界区内部执行一个或多个客户多处理操作系统 420。为一个或多个客户多处理操作系统 420 的虚拟处理器创建虚拟机。虚拟机监视器 470 在主操作系统 460 下运行，以管理用于对客户操作系统 420 进行虚拟化的虚拟机。虚拟机监视器 470 的一个任务是在多处理系统 401 中每一个可得到的硬件线程 410-411 上并发地启动虚拟机。

特定虚拟机内部的执行在处理器 402 中并发地开始，其中存储在高速缓存存储装置 404 和/或高速缓存存储装置 405 上的多线程主操作系统 460、多线程虚拟机监视器 470 和多线程客户操作系统 420 的可执行指令的副本，可以在线程选择逻辑 414 指导下通过指令取回逻辑 415 取回，并且为执行逻辑 412 而被分配给合适的硬件线程 410-411。在退出与多线程客户操作系统 420 相关的每一个硬件执行线程 410-411 的虚拟机环境时，在多线程虚拟机监视器 470 的环境中判断是否需要调度与多线程客户操作系统 420 相关的另一虚拟机，如果需要，则可以选择另一虚拟机以分配给特定的硬件执行线程。否则恢复已经分配给硬件执行线程的虚拟机。

由此实现了对多线程客户操作系统的有效虚拟化，其可以利用多处理主平台上的所有可用的硬件执行线程。

以上说明用于描述本发明的优选实施例。根据以上讨论，显而易见的是，尤其在这样一个技术领域，其中发展十分迅速且今后的发展不容易预见，本发明通过本领域技术人员可以在所附权利要求及其等价物范围内，在没有脱离本发明的精神的情况下，在组成和细节上进行修改。

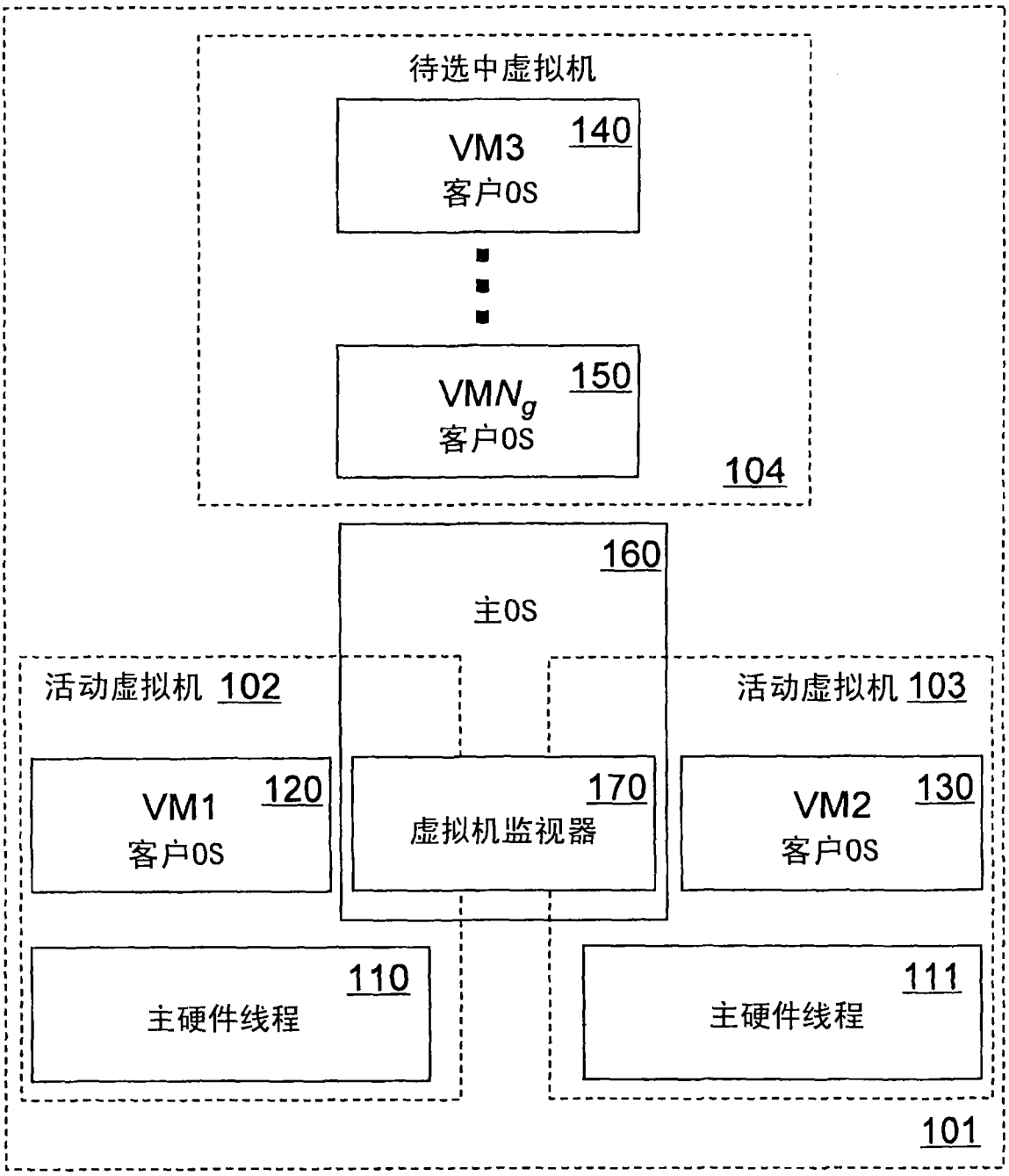

待选中虚拟机
VM3
客户OS
140
VMNg
客户OS
150
104
160
主OS
活动虚拟机 102
活动虚拟机 103
VM1
客户OS
120
170
虚拟机监视器
VM2
客户OS
130
110
主硬件线程
111
主硬件线程
101

图1

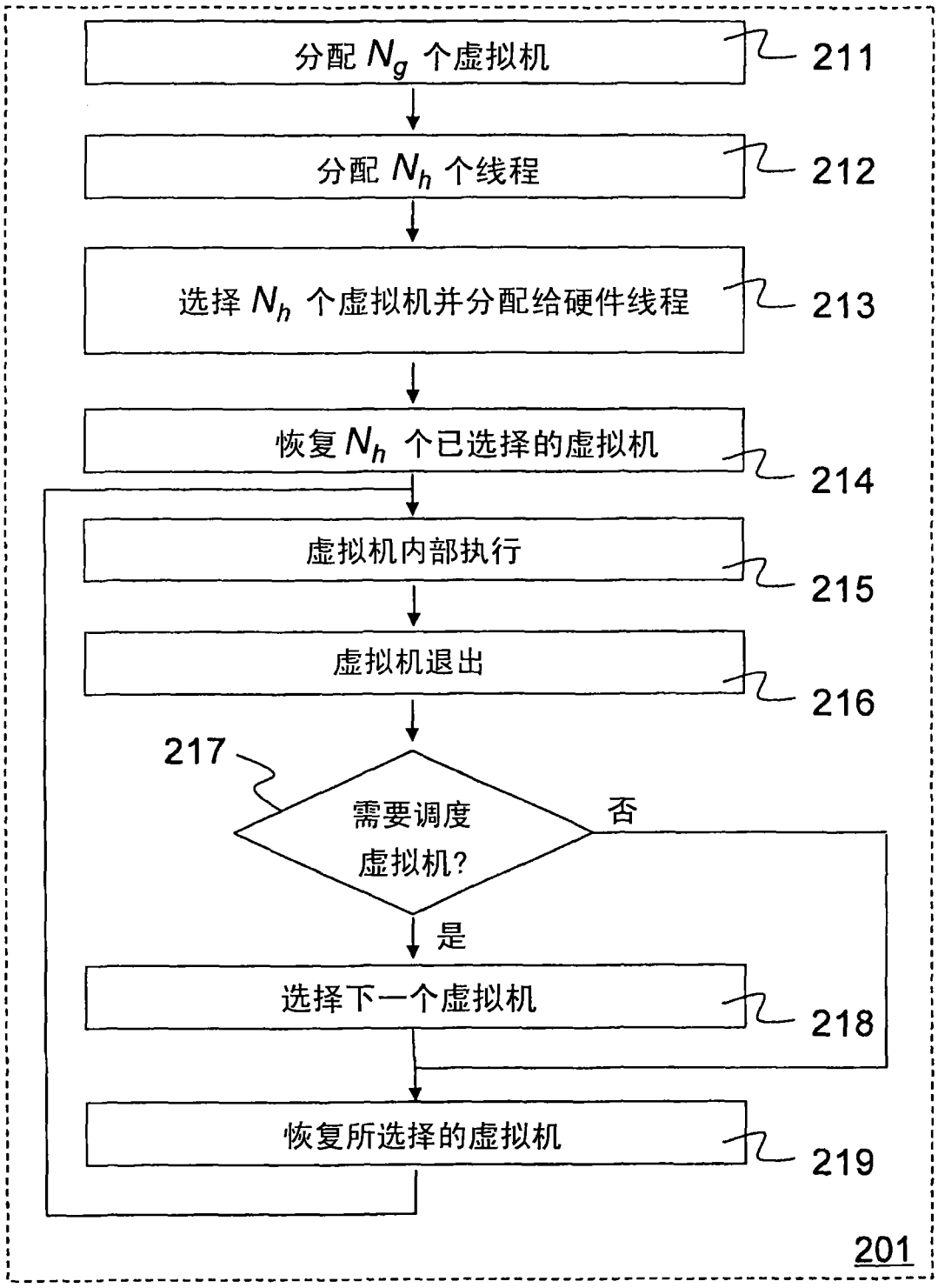

分配 N_g 个虚拟机
211
分配 N_h 个线程
212
选择 N_h 个虚拟机并分配给硬件线程
213
恢复 N_h 个已选择的虚拟机
214
虚拟机内部执行
215
虚拟机退出
216
217
需要调度虚拟机?
否
是
选择下一个虚拟机
218
恢复所选择的虚拟机
219
201

图2

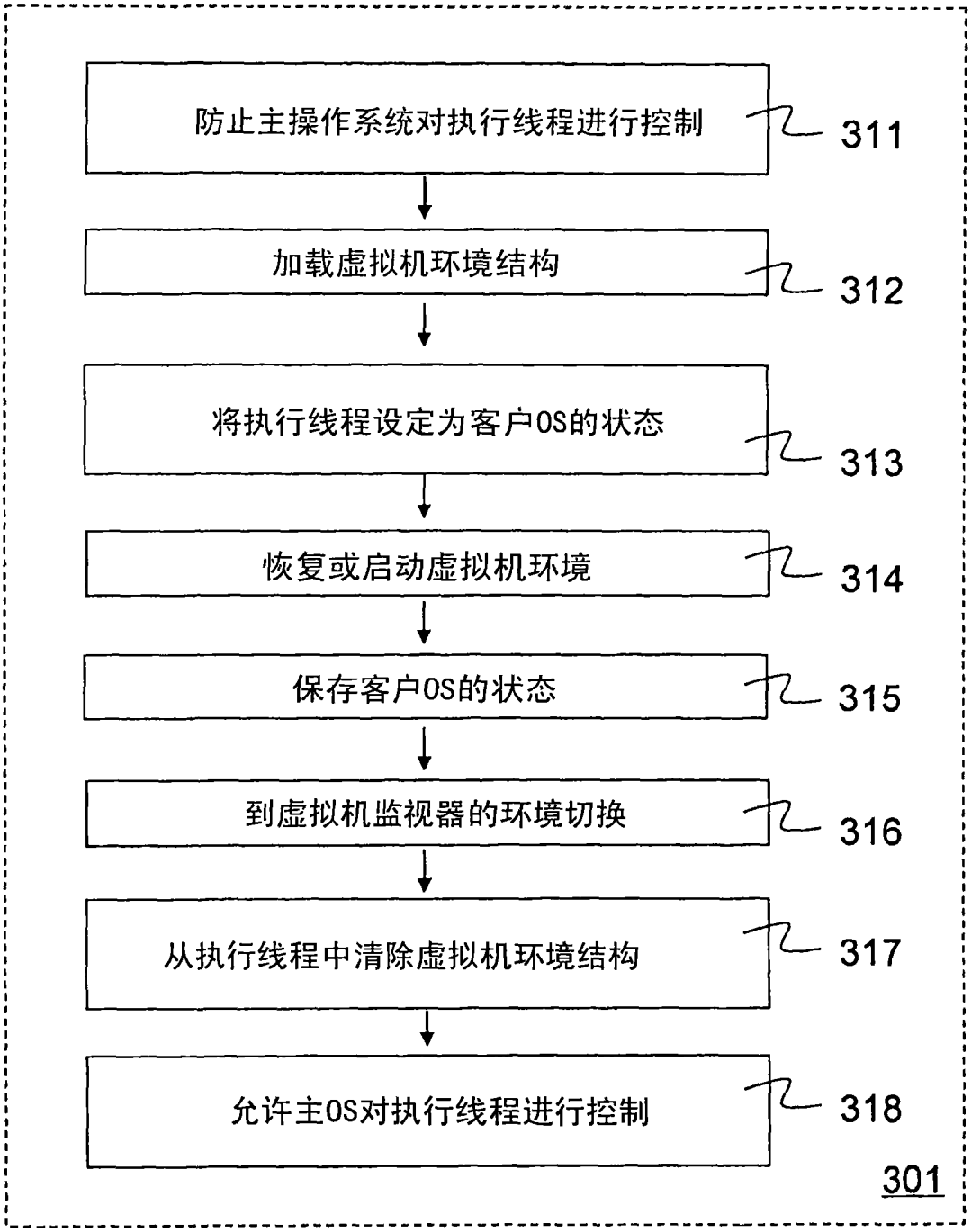

防止主操作系统对执行线程进行控制
311
加载虚拟机环境结构
312
将执行线程设定为客户OS的状态
313
恢复或启动虚拟机环境
314
保存客户OS的状态
315
到虚拟机监视器的环境切换
316
从执行线程中清除虚拟机环境结构
317
允许主OS对执行线程进行控制
318
301

图3

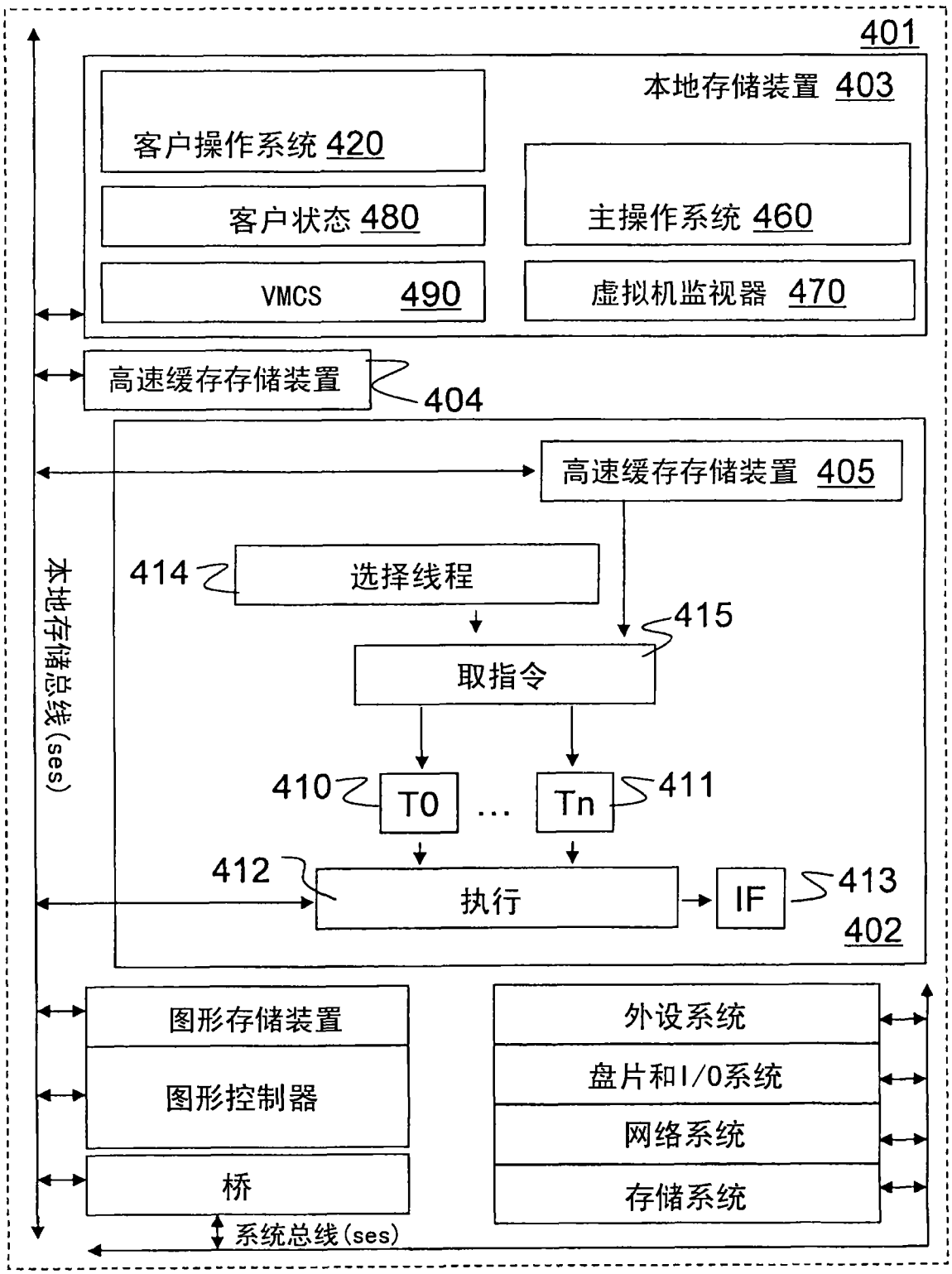

401
本地存储装置 403
客户操作系统 420
客户状态 480
VMCS 490
主操作系统 460
虚拟机监视器 470
高速缓存存储装置 404
高速缓存存储装置 405
本地存储总线(ses)
414
选择线程
415
取指令
410
T0
...
Tn
411
412
执行
IF
413
402
图形存储装置
图形控制器
桥
外设系统
盘片和I/O系统
网络系统
存储系统
系统总线(ses)

图4