



(12) 发明专利申请

(10) 申请公布号 CN 102270290 A

(43) 申请公布日 2011. 12. 07

(21) 申请号 201110153233. 1

(22) 申请日 2011. 06. 01

(30) 优先权数据

12/791, 353 2010. 06. 01 US

(71) 申请人 国际商业机器公司

地址 美国纽约阿芒克

(72) 发明人 I·N·沃利 W·塞格穆勒

M·斯坦德

(74) 专利代理机构 北京市金杜律师事务所

11256

代理人 鄞迅 姜彦

(51) Int. Cl.

G06F 21/22(2006. 01)

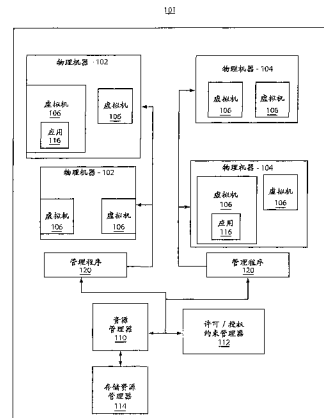
权利要求书 1 页 说明书 10 页 附图 10 页

(54) 发明名称

用于对虚拟环境中的许可授权进行管理的系统和方法

(57) 摘要

本发明涉及用于对虚拟环境中的许可授权进行管理的系统和方法。具体地，一种用于虚拟化环境的管理系统和方法，其包括具有基于授权的使用限制的计算机实体。资源管理器，其使用处理器，并且在存储器存储设备上被编程以及从该存储器存储设备得以执行，该资源管理器配置用于在虚拟化环境中管理资源。授权使用模块，其耦合至该资源管理器，并且配置用于根据该虚拟化环境中的变化来跟踪授权相关的约束，以允许该资源管理器做出包括授权相关的约束的分配决定，以保证使用限制满足该计算机实体。



1. 一种用于虚拟化环境的管理系统,包括:
至少一台计算机实体,其具有基于授权的使用限制;
资源管理器,其使用处理器,并且在存储器存储设备上被编程以及从该存储器存储设备得以执行,所述资源管理器配置用于在虚拟化环境中管理资源;以及
授权使用模块,其耦合至所述资源管理器,并且配置用于根据所述虚拟化环境中的变化来跟踪授权相关的约束,以允许所述资源管理器做出包括所述授权相关的约束的分配决定,以保证所述使用限制满足所述至少一台计算机实体。
2. 如权利要求 1 所述的系统,其中所述至少一台计算机实体包括虚拟机、应用和容器中的至少一个。
3. 如权利要求 1 所述的系统,其中所述资源管理器基于至少一个其他考虑做出布置决定。
4. 如权利要求 3 所述的系统,其中所述至少一个其他考虑包括性能、费用和安全中的至少一个。
5. 如权利要求 1 所述的系统,其中所述资源管理器包括配置用于基于候选布置改变来做出布置决定的程序。
6. 如权利要求 1 所述的系统,其中所述资源管理器包括配置用于通过修改布置解决方案来做出布置决定以避免违反授权的程序。
7. 如权利要求 1 所述的系统,还包括授权桶,其与所述至少一台计算机实体相关联,并且配置用于为所述至少一台计算机实体标识所述基于授权的约束。
8. 如权利要求 1 所述的系统,还包括强制模式,在所述强制模式期间,授权不允许被超出;以及警告模式,在所述警告模式下,需要单独的允许来超出所述授权。
9. 一种用于在虚拟化环境中管理资源的方法,包括:
在计算机存储介质中表示用于授权集合的约束;
针对待布置的计算实体,确定允许多少以及何种类型的授权;以及
随着布置计划的进行,使用处理器计算当前候选布置解决方案的授权使用,以使得作为结果的布置解决方案不会超出可用的所述授权。
10. 如权利要求 9 所述的方法,其中表示用于授权集合的约束包括提供指定用于包含约束信息的文件的扩展。
11. 如权利要求 9 所述的方法,还包括根据所述授权集合的所述约束来限制资源使用。
12. 如权利要求 9 所述的方法,其中所述计算实体包括虚拟机。
13. 如权利要求 9 所述的方法,其中所述授权集合包括许可限制,以及所述方法还包括捕获授权桶中与计算机实体以及模型许可约束相关联的许可限制。
14. 如权利要求 9 所述的方法,还包括计算当前候选布置解决方案的性能度量、费用度量和安全度量中的至少一个,以便作为结果的布置解决方案不超出可用的授权和各个度量的阈值。
15. 如权利要求 9 所述的方法,还包括修改布置解决方案以避免违反授权。
16. 如权利要求 9 所述的方法,还包括提供强制模式,在所述强制模式期间,授权不允许被超出;以及警告模式,在所述警告模式下,需要单独的允许来超出所述授权。
17. 如权利要求 9 所述的方法,还包括提供调用者程序,其具有每个授权类型的当前使用水平。

用于对虚拟环境中的许可授权进行管理的系统和方法

技术领域

[0001] 本发明涉及虚拟系统管理,并且更具体地,涉及配置用于在管理虚拟环境时考虑使用权利和限制的系统和方法。

背景技术

[0002] 现代企业软件伴有巨额的购买费用、复杂的许可条款,以及在违反了许可时对客户的严重惩罚。传统上,软件简单地根据安装拷贝进行许可,即,客户针对安装的每个软件拷贝支付一定量的费用。针对客户的软件费用由此是客户所使用(或者更确切地说,已安装的)备份的数量乘以客户为每个拷贝支付的价钱。另有一已投入使用的技术即所谓的“许可证池”。在这种技术中,客户仅仅为正在运行的软件拷贝支付费用——可以安装所需数量的拷贝,但是只有它们中的某些可以同时执行。这两种模型在 workstation 软件(运行于单独个人计算机上的软件)领域已经并且依然被广泛使用。

[0003] 现代服务器软件有更加复杂的许可条款。作为驱动示例,考虑“处理器价值单元”方法或者称 PVU。这种方法不是简单地以安装实例来许可软件,而是以测量客户从软件中获得的潜在价值的尝试为基础。

[0004] 基于 PVU 的产品许可不使用单一的费率向产品收取费用。相反,产品根据 PVU 来定价。当购买经 PVU 许可的产品时,客户计算出需要多少 PVU,并将该数量与所讨论产品的每个 PVU 价格相乘。

[0005] 作为示例,应用服务器的标价为每个 PVU 60 美元。为完成定价计算,客户接下来查看该客户希望产品在其上运行的 CPU。每个处理器类型具有已定义的“针对每个核的 PVU”值,其将告知客户需要购买多少 PVU。例如,在本示例中,将用于所有处理器(具有多个核)的针对每个核的 PVU 值的价格在表格中列出为 50。因此,希望在 8 核机器上运行应用服务器的客户需要具有应用服务器的 $8 \times 50 = 400$ 个 PVU。如果每个 PVU 的费用是 60 美元,则将花费 $60 \times 400 = 24,000$ 美元。

[0006] 基于 PVU 的许可的一个重要方面是其并不在意客户安装或者运行应用服务器的多少拷贝,而仅仅在意那些拷贝有权访问多少个 PVU。例如,客户可以转而决定运行两台机器,其中每台机器都有 4 个核,而所需的 PVU 仍为 400 ($2 \times 4 \times 50$)。另外,客户可以决定在那两台机器中的每一台上运行 7 个应用服务器拷贝,而所需的 PVU 仍为 400。

[0007] 在这种模式中,PVU 在产品间是不可互换的。其在 CPU 类型和机器之间是可互换的。通过示例,为应用服务器购买 400 个 PVU 的客户可以在其所希望的任何机器上运行该应用服务器,只要该客户不在所需总 PVU 超出 400 的 CPU 上运行即可。然而,该客户不能停止运行该应用服务器而替代地运行使用 PVU 模式许可的另一(第二)软件产品。用于第二软件产品的每个 PVU 价格与用于该应用服务器的每个 PVU 价格是不同的,所以客户不能使用针对一个产品购买的 PVU 来运行另一产品。

[0008] 因此,可以看出在任何给定系统中,将部署有经各种许可的产品,而客户需要针对每个产品跟踪其授权和使用,以保证其与许可条款的一致性。

发明内容

[0009] 一种用于虚拟化环境的管理系统和方法包括具有基于授权的使用限制的计算机实体。资源管理器,其使用处理器并且在存储器存储设备上被编程以及从该存储器存储设备得以执行,该资源管理器配置用于在虚拟化环境中管理资源。授权使用模块,其耦合至该资源管理器,并且配置用于根据该虚拟化环境中的变化来跟踪授权相关的约束,以允许该资源管理器做出包括授权相关的约束的分配决定,以保证使用限制满足该计算机实体。

[0010] 一种用于在虚拟化环境中管理资源的方法,包括表示授权集合的约束以及针对待布置的计算实体确定允许多少个何种类型的授权。随着布置计划的进行,针对当前候选布置解决方案来计算授权使用,以使得作为结果的布置解决方案不会超出可用的授权。

[0011] 这些以及其他特征和优势将从结合附图阅读的本发明示意性实施方式的详细描述变得容易理解。

附图说明

[0012] 本公开将参考附图来为下文优选实施方式的说明提供细节,其中:

[0013] 图 1 是系统/方法的框图/流程图,其在管理虚拟环境时考虑了许可/授权;

[0014] 图 2 是示出了根据本发明原理的布置管理系统示意性示例的框图;

[0015] 图 3 是示出了用于在管理环境的同时考虑许可/授权的示意性示例的框图;

[0016] 图 4 是示出了用于在新增 VM 的情况下、在管理图 3 环境的同时考虑许可/授权的另一示意性示例的框图;

[0017] 图 5 是示出了用于在共享 PVU 的情况下、在管理图 4 环境的同时考虑许可/授权的另一示意性示例的框图;

[0018] 图 6 是示出了用于在管理图 3 环境的同时考虑新增 VM 的布置位置的另一示意性示例的框图;

[0019] 图 7 是示出了用于在共享 PVU 的情况下、在管理图 4 环境的同时考虑新增 VM 的布置位置的另一示意性示例的框图;

[0020] 图 8 是示出了根据一种实施方式的计算机实体和授权集(bundle)之间关系的模型构建的框图;

[0021] 图 9 是示出了根据一种示意性实施方式的、考虑了许可/授权的布置方法的框图/流程图;以及

[0022] 图 10 是示出了根据另一示意性实施方式的、考虑了许可/授权的另一布置方法的框图/流程图。

具体实施方式

[0023] 根据本发明原理,提供了一种用于对虚拟环境进行集成管理的系统和方法。在特定实用的实施方式中,管理系统包括许可认知,并且将此认知纳入到其决定和动作中。例如,当管理系统能够部署新的虚拟机以及能够将已存在的虚拟机从一个地方迁移到另一地方时,包括许可认知的益处就变得更加重要。

[0024] 虚拟化是涉及进行计算资源抽象能力的通用术语。本实施方式可以包括在平台虚

拟化中,其包括计算机系统(与例如存储和网络形成对照,当然,这些也是可以采用的)的虚拟化。一种类型的虚拟化是完全虚拟化,其中通常隔离的多个操作系统实例可以在单个计算机上运行。也可以包括其他类型的虚拟化,例如操作系统级别的虚拟化,其中在单个操作系统中,将呈现出通常隔离的多个空间,而程序在这些空间中运行。

[0025] 完全虚拟化的示例包括:IBM[®](在 p 系列和 z 系列上)的 DLPAR,IBM[®]的 z/VM,VMware[®]产品系列,Citrix[®]的 XenServer 和 Xen,以及 Linux[™]的 KVM,在此仅举几个例子。“操作系统级别的虚拟化”的一般示例包括:用于 AIX 操作系统的 IBM[®]的 WPAR,Sun[®]的 Solaris 容器,以及 Linux[™]的 VServer。

[0026] 虚拟机例如可以涉及操作系统实例(在完全虚拟化的情况下),也可以涉及空间(在操作系统级别虚拟化的情况下)。虚拟化的一个相关方面是能够控制 CPU 分配。例如,当在单个物理机器上运行多个虚拟机时,可以限制每个虚拟机有权访问物理机器中的哪几个 CPU。

[0027] 例如,考虑具有 8 个处理器核的单个物理机器。该物理机器可以运行 3 个虚拟机,其中第一个拥有 2 个虚拟 CPU,第二个拥有 4 个虚拟 CPU,以及第三个拥有 8 个虚拟 CPU。在虚拟机内部运行的软件仅可以等于该虚拟机有权访问的 CPU 数量。

[0028] 另一方面包括活迁移(live migration)(也称为分区移动性或者迁移)。这涉及在无需中断服务的情况下、将虚拟机从一台物理机器移动到另一台物理机器的能力。

[0029] 虚拟化和软件许可之间的交叉可以将 PVU 类型许可和虚拟化相结合。结合上述虚拟机示例和较早的应用服务器示例,应当允许客户针对原先的 400 个 PVU 需求、在所有 3 个虚拟机上运行该应用服务器。所有 8 个物理核都已经被使用,所以所有 8 个核都必须经过许可。在单个机器上运行应用服务器的多个拷贝不会改变许可需求,不同示例运行在单独虚拟机上的这个事实与许可机制没有关系。换句话说,客户为了在这个机器上运行应用服务器而需要购买的最大 PVU 为 400——有多少实例或者虚拟机正在运行都是无所谓的。

[0030] 考虑 PVU 的共享。在该实例中,首先创建 8 个虚拟 CPU 虚拟机。然后,再创建 4 个虚拟 CPU 虚拟机和 2 个虚拟 CPU 虚拟机。针对后两个 VM,不需要额外的许可,它们正在“共享”创建第一虚拟机时所需的许可。然而,第一虚拟机没有任何特别之处——在这个示例中,其仅仅是第一个被创建出来的,以及因此引发针对物理核许可的需求。部署随后的 2 个虚拟机不再需要额外的许可。

[0031] 如果客户的需要少于 400 个 PVU,例如,在客户仅仅在第一虚拟机(该虚拟机拥有 2 个虚拟 CPU)上运行应用服务器时。在这种情况下,该客户仅需要 100 个 PVU(2 个物理核的每个需要 50 个)。在这种平台上的这种类型的许可下,客户需要具有足够用于虚拟 CPU 的总数或者物理核数量二者中的较少者的 PVU。应当理解,上述 PVU 类型许可的描述并非旨在限制,此处为了简明而略去了细节,并且完全省略了其他平台类型和集群问题。上述说明仅仅旨在示意其中可以实现本发明原理的环境。另外,其他软件提供者具有其他许可机制。然而,它们中的很多都具有与上述基于 PVU 的机制类似的多个方面。

[0032] 当虚拟化在最终用户数据中心中变得越来越普遍时,虚拟化管理变得更加重要。在性能管理、高可用性乃至电力节约方面都有很大的进展。然而,一个被忽略的但可以为客户提供重要益处的方面是许可管理。具体地,能够在考虑其他管理因素的同时考虑许可管

理是非常有用的。部署以及迁移都可以改变系统在任何给定时刻所使用的许可数量。例如,再次考虑先前用过的 3 个虚拟机的情况,但是引入第二物理机器,其上初始没有部署任何虚拟机。可以理解的是,从第一物理机器向第二物理机器迁移虚拟机将增加所需许可全部系统的 PVU 数量——与先前相比,现在有更多的物理核在运行该应用服务器。还可以举出很多包含部署和迁移的其他情况。

[0033] 根据本发明原理,集成的管理系统包括许可认知特征。当该管理系统能够部署新的虚拟机和 / 或能够将已存在的虚拟机从一个地方迁移到另一地方时,在做出决定或者执行这种动作时考虑许可认知。除了其他管理考量之外,该系统考虑许可需求。

[0034] 贯穿本公开,将参考单个客户;然而,本实施方式也同样适用于拥有各自许可的、共享基础架构的客户集合。本实施方式所提供的某些特征包括下述(特征)。利用一种或者多种机制为当前客户的软件产品获得许可授权,以及与之对应的许可类型,这些机制诸如通过向系统管理员索要该信息,或者通过电子地联系软件供应者向其请求该信息。把这些许可授权以及许可规则考虑进来,以收集关于正被管理的基础架构的足够信息以便能够计算当前许可使用。这些信息包括但不限于:该系统中各种物理机器的数量和特征。当前部署的虚拟机的数量和特征(如果适用的话)。关于哪些软件产品已被安装以及安装在哪里的信息。

[0035] 当考虑到所管理的系统的潜在改变时,本实施方式能够计算改变该系统的效果(从所需许可的方面),以及从其对许可需求的影响方面评估潜在改变的不同集合,以及其他系统级别的影响(例如,包括性能、可用性、电力消耗等等)。本实施方式能够选择针对系统做出的改变集合,其将该系统限制在客户许可授权范围内。当上述情况不可能时,可以有各种方法来重载系统。在一种实施方式中,该系统不能重载许可授权。在另一种实施方式中,该系统可以重载,但是仅仅在系统管理员允许的情况下可以。其他实施方式也是可以的。

[0036] 在一个示例中,数据中心包括 100 台物理机器。许可使用可以通过下述而最小化:将所有经许可的产品布置在同一台物理机器上,而不使用任何其他机器(这个方法还可以在电力消耗上具有有益效果)。然而,系统的性能可能受到损失,因为单个物理机器没有足够的资源来充分主控所有经许可的产品。这样的解决方案在可用性方面也有负面的影响——当该单个机器瘫痪时,所有的产品都将瘫痪。

[0037] 本领域技术人员将会理解,本发明的方面可以实现为系统、方法或计算机程序产品。因此,本发明实施方式的多个方面可以采取完全硬件实施方式、完全软件实施方式(包括固件、驻留软件、微代码等)或组合有可在此统称为“电路”、“模块”或“系统”的硬件方面和软件方面的实施方式。此外,本发明的多个方面可以采取在其上具体化有计算机可读程序代码的一个或多个计算机可读介质中具体化的计算机程序产品的形式。

[0038] 可以使用一个或多个计算机可读介质的任何组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是但不限于电的、磁的、光的、电磁的、红外的或半导体系统、装置或设备,或前述任何适当组合。计算机可读存储介质的更具体示例(非穷举列表)可以包括以下内容:具有一个或多个导线的电连接、便携式计算机盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM 或闪存)、光纤、便携式压缩盘只读存储器(CD-ROM)、光存储设备、磁存储设备或

前述任何适当组合。在本文档的上下文中,计算机可读存储介质可以是可包含或存储供指令执行系统、装置或设备使用或与其结合使用的程序的任何有形介质。

[0039] 计算机可读信号介质可以包括例如在基带中或作为载波一部分的、其中具体化有计算机可读程序代码的传播数据信号。这种传播信号可以采取多种形式中的任何一种,包括但不限于电磁的、光的或其任何适当组合。计算机可读信号介质可以是任何计算机可读介质,该介质不是计算机可读存储介质,并且可以传达、传播或传送供指令执行系统、装置或设备使用或与之结合使用的程序。具体化在计算机可读介质上的程序代码可以使用任何适当介质(包括但不限于无线、有线、光纤线缆、RF 等或前述任何适当组合)进行传输。

[0040] 用于执行本发明方面的操作的计算机程序码,可以以一种或多种编程语言的任何组合来编写,所述编程语言包括面向对象的编程语言 - 诸如 Java、Smalltalk、C++ 之类,还包括常规的过程式编程语言 - 诸如“C”编程语言或类似的编程语言。程序代码可以完全地在用户的计算机上执行、部分地在用户的计算机上执行、作为一个独立的软件包执行、部分在用户的计算机上并且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种类的网络 - 包括局域网(LAN)或广域网(WAN) - 连接到用户的计算机,或者,可以(例如,利用因特网服务提供商来通过因特网)连接到外部计算机。

[0041] 已经在上文中参考根据本发明实施方式的方法、装置(系统)和计算机程序产品的示意流程图和/或示意框图描述了本发明的方面。应当理解,示意流程图和/或示意框图的每个框,以及示意流程图和/或示意框图中框的组合,可以由计算机程序指令实现。这些计算机程序指令可以向通用计算机、专用计算机或其他可编程数据处理装置的处理器提供以产生机器,使得经由计算机或其他可编程数据处理装置的处理器执行的指令创建用于实现流程图和/或框图框中指定的功能/动作的装置。

[0042] 这些计算机程序指令也可以存储在计算机可读介质中,其可以指示计算机、其他可编程数据处理装置或其他设备以特定方式起作用,使得存储在计算机可读介质中的指令产生产品,该产品包括实现流程图和/或框图框中指定的功能/动作的指令。

[0043] 计算机程序指令也可以向计算机、其他可编程数据处理装置或其他设备加载,以引起在计算机、其他可编程装置或其他设备上执行一系列操作步骤,以产生计算机实现的过程,使得在计算机或其他可编程装置上执行指令时,可以提供用于实现流程和/或框图框中指定的功能/动作的过程。

[0044] 附图中的流程图和框图示出了根据本发明各种实施方式的系统、方法和计算机程序产品的可能实现的架构、功能性和操作。就这一点,流程图或框图中的每个框可以代表模块、片段或代码的部分,其包括用于实现指定逻辑功能的一个或多个可执行指令。还应当注意,在有些备选实现中,框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。还应当注意,框图和/或流程图的每个框以及框图和/或流程图中框的组合可以由执行专用硬件和计算机指令的组合、动作或指定功能的基于专用硬件的系统实现。

[0045] 现在参考附图,其中类似的标号表示相同或相似的元素;以及从图 1 开始,示意性地描述系统 101,其将使用权利、限制以及约束纳入考虑,以便为虚拟化环境中的集成管理

做好准备。系统 101 示意性地包括虚拟环境,其具有物理机器 102、104、虚拟机 (VM) 106、资源管理器 110、许可 / 授权约束管理模块 112 和存储资源管理器 114。系统 101 可以包括数据中心环境,网络环境或者计算机制品或实体在其中被提供、改变、执行、迁移或进行了其他操作的任何其他计算机环境。计算机制品或实体可以包括虚拟机 (VM)。虚拟机示意性地用于描述本示意性系统。应当理解的是,应用或者另一其他计算实体也可以以类似的方式得以应用。

[0046] 应用 116 由单独的 VM 106 主控,而物理机器 102 和 104 可以主控多个 VM 106。每个 VM 106 具有在启动时分配给该 VM 106 的资源(网络、存储器以及 CPU) 份额,以及与共同主控在相同物理机器 102、104 上的其他 VM 106 共享资源。物理机器可以主控多个 VM106。VM 106 可以迁移到其他物理机器或者环境。

[0047] 存储资源管理器 114 负责监测系统 100 中存储的存储使用。资源管理器 110 负责就布置或重新定位(迁移)以及重新提供虚拟机 106 做出决定,以及在必要时,例如,在 VM 106 已针对潜在重新定位而被选择时,与模块 112 和存储资源管理器 114 配合。布置涉及与在给定时刻应当在哪里运行给定虚拟机有关的决定。

[0048] 可以基于多个不同考虑来做出布置决定。例如,虚拟机的性能认知布置考虑作为 VM 布置结果的性能增益,而虚拟机的约束认知布置则是基于约束的。约束认知决定可以考虑分配约束(例如,“不要将某物布置在特定地点”,“某物只能放在位置 x, y 和 z”)以及排列(collocation)约束(“这个不可以与那个放在一起”,“这个不能与任何东西放在一起”,等等)。许可认知布置可以包括在性能认知和 / 或约束认知布置中。虚拟机的初始部署可以在部署时考虑额外的贡献。例如,网络连接、存储可用性等等。

[0049] 资源管理器 110 和管理程序 120 用于管理 VM 存储、迁移、应用执行以及其他管理功能。许可约束管理模块 112 可以与资源管理器 110 和管理程序 120 中的一个或者两者集成,也可以集成到资源管理器 110 和管理程序 120 的一个或者两者中。在迁移期间,系统 101 执行如下示意性功能:(1) 资源管理器 110 指派 VM 106 进行迁移或者重新提供。(2) 资源管理器 110 咨询许可约束管理模块 112 以确定针对将 VM 106 从其当前位置移动到新位置是否存在约束。这包括确定是否存在阻止向新位置移动或者从旧位置移开的约束。许可约束管理模块 112 存储约束信息,其可以来自服务级别协议(SLA)、许可协议、版权信息等等。根据实用实施方式,由资源管理器 110 咨询许可约束管理模块 112,这是就任何资源改变做出决定的过程的一部分。许可约束管理模块 112 可以向调用者提供每个授权类型的当前使用级别。

[0050] 资源管理器 110 提供管理系统,该管理系统在其决定中包括许可认知。当管理系统部署新的虚拟机(106)、从一个地方向另一地方迁移已有虚拟机(106)、执行应用或 VM 和 / 或重新提供 VM(106) 时,许可认知将变得重要,该管理系统考虑许可(以及其他使用约束)和其他管理考量。系统 101 的许可约束管理模块 112 还包括区分竞争约束优先级的能力。

[0051] 许可约束管理模块 112 使用各种机制来获得针对当前客户的软件产品的授权数据,其中的授权机制包括:向系统管理员索要该信息、电子地联系软件供应者向其请求该信息、考虑许可授权、对应许可类型等等。资源管理器 110(以及或者许可约束管理模块 112)考虑许可授权和许可规则,以及收集关于正被管理的基础架构的足够信息以便能够计算当

前许可使用。该信息包括但不限于：例如，系统 101 中物理机器 102, 104 的数量和特征；当前部署的虚拟机 106 的数量和特征（如果适用的话）；关于哪些软件产品已被安装及其位置的信息，等等。

[0052] 当考虑所管理的系统的潜在改变时，这种改变的效果需要针对该系统的改变计算出来（在所需许可等方面）。除了其他系统级别影响（例如，包括性能、可用性、电力消耗等等）以外，不同潜在改变的集合可以在其对许可需求的影响方面进行评估。选择这样的改变集合，其保证系统被限制在客户的许可授权范围内。当上述情况不可能时，可以有各种方法来重载该系统，或者可以选择遵循许可协议精神的最佳解决方案。在一种实施方式中，该系统不能重载许可授权。在另一种实施方式中，该系统可以重载，但是仅仅在系统管理员允许的情况下可以。其他实施方式也是可以的。

[0053] 用于虚拟化环境的产品许可规则通常是复杂的。最近，产品可以基于其可以做多少工作来收取费用，而不是基于所安装拷贝的数量。结果，许可使用依赖于如何将安装的产品拷贝布置在物理硬件上。将处理器价值单元 (PVU) 作为应对虚拟化环境的示例性示例。PVU 分数确定特定处理器类型的许可费用。必须为产品使用的每个处理器支付费用。相同产品的多个拷贝所使用的处理器仅仅需要为其支付一次费用。当产品安装在可以使用该处理器的 VM 上时，该处理器由该产品使用，停止的产品以及停止的 VM 作为使用进行计数。在从 GMT 时间午夜开始的每 24 小时时段内、由产品使用的每个处理器将被计数，VM 迁移影响 PVU 使用，针对 VMware 的特定集群规则避免在使用分布式资源调度 (DRS) 时（而不是在之后）收取过多费用。

[0054] 参考图 2，布置系统 180 包括在图 1 的约束管理器 112 中。为了确定授权使用，布置系统 180 需要能够为所有可能的布置计算所有授权的当前使用。此外，布置系统 180 需要能够针对在布置上做出特定改变而计算对那些授权使用的影响。以在此描述的许可授权为示例，更特别地以（仅仅为了解释说明的目的）PVU 许可为示例，布置系统 180 需要知道下列信息：每个物理机器 (104) 上物理核的数量；每个物理核的 PVU “分数”；每个虚拟机中的每个产品有权访问的虚拟核的数量；以及每个虚拟机中的产品。

[0055] 给定了这些信息，布置系统 180 可以在任何时候计算出虚拟机的给定布置所消耗的授权的数量。因此，可以发现，随着布置例程的进行，其可以重新计算已在每个步骤使用的每个授权的量。此外，当布置例程考虑做出改变时，其可以计算该改变将对系统中的每个授权产生的影响。

[0056] 为了实现系统 180，实现感测器以确定所需信息。物理机器 (PM) 感测器 182 检测物理机器 (104) 的特性，其继而与软件组件（例如，数据库）或者知道给定处理器类型和数量的 PVU “分数”的 PVU 专家 184 耦合。虚拟机 (VM) 感测器 186 提供关于虚拟机的信息，以及产品感测器 188 提供关于所安装产品及其许可条款的信息。布置系统 180 可以包括经过修改的 PM 感测器 182 和 VM 感测器 188，它们已被扩展以支持收集这些以及其他所需数据。

[0057] 布置系统 180 将系统数据（来自感测器）传入布置执行器 190 中的规范形式中，从而将布置系统的关注点与感测器的特定实现隔离开。然后，布置执行器 190 在就有关做出什么改变方面做出其决定时检查该规范形式。该规范形式将包括授权桶 (BoE) 等，如在此所描述的。

[0058] 此外,系统 180 的实现包括随着时间推移系统的正在进行的优化和管理。来自感测器 (182, 186, 188) 的事件将触发布置系统 180 对当前布置进行重新评估。这些事件将包括针对授权需求或者授权可用性的改变。

[0059] 参考图 3,将描述简化的 PVU 系统 200 以演示本发明原理中的概念。4 个物理机器 (PM) 202,每个包括 4 个物理计算机处理器单元 (pCPU) 204。每个 pCPU 204 包括例如 50 个 PVU。所有 VM 206 和 207 包括 2 个 vCPU 208。在图 2 中,3 个 VM 206 运行例如 **IBM**[®] 的 Websphere 应用服务器™ 的应用服务器 (AS) 应用,以及一台 VM207 运行数据库 (DB) 应用 (例如, **IBM**[®] 的 DB2™)。应用服务器应用 (AS) 的 PVU 使用为 300,因为每个 VM 206 使用 2 个 pCPU 204 (每个 $2 \times 50 = 100$)。数据库应用 (DB) 的 PVU 使用是 100,因为每个 VM 207 使用两个 pCPU 204 (每个 $2 \times 50 = 100$)。

[0060] 在图 4 中,第四 VM 206”运行应用服务器应用,以及将针对该应用的 PVU 消耗从 300 增加到 400。在图 5 中,创建了新的 VM 209,并且其从共享相同的机器和 CPU 方面受益。不会由于向已被 PVU 计数计入的处理器引入 VM 209 而产生额外的 PVU 消耗。

[0061] 再次参考图 3,假设客户购买了 AS 应用的 400 个 PVU。在系统 101 的强制模式期间,当安装时 (或者其他时间),许可模块 112 或者资源管理器 110 (图 1) 将查看一致性。当建议使用 400 个 PVU 时,在这个场景中,100 个正用于 DB,并假设客户没有被授权使用 DB 应用。在这种情况下,存在违反。系统 101 (图 1) 可以联系客户,并为之签订针对 DB 应用的 100 个 PVU,或者简单地发出该应用在当前许可安排中不可用的警告。在一种实施方式中,强制模式将关闭 DB 应用的未授权使用,以保证遵守授权。

[0062] 在图 6 中,图 3 的安排允许增加新的 VM 209 (如在图 5 中所示);然而,在 VM 209 的布置方面存在选择。针对 VM 209 存在 4 个位置 212。所有位置都导致 400 的 AS PVU 消耗,以及所有布置都有效。然而,将 VM 209 布置在每个位置 212 在每种情况中都是不等价的。管理决定可以包括诸如用以确定 VM 209 的最佳位置的基于性能的布置之类的附加标准。

[0063] 在一个示例中,如果除了 VM 209 之外还需要针对 AS 使用布置另一 VM (未示出),这是不可以的——因为 PVU 限制 400 将被超出。不能布置新的 VM,因为所有的选择都违反了 AS PVU 限制。客户可以为该 AS 应用签订更多的 PVU 或者可以采取其他动作。另一场景可以解决这个问题。如图 7 中所描述的,第一新 VM 209 布置在第一 PM 202 (最左边的 PM) 并位于 4 个位置 212 中的一个位置处,以及新 VM 211 可以被布置以触发 PVU 共享 (见图 5)。通过这种方法,用于 AS 的 PVU 维持得低于 400 个 PVU 的限制,以及额外的 VM 211 被引入。

[0064] 在另一示例中,只有一台 PM 202 可用于 PVU 共享。可以对 VM 进行迁移以保证 PVU 的一致性和平衡性能。例如,PM 202 可以充分填充以共享资源以及避免 PVU 的消耗。

[0065] 移动 VM 可能会导致关于 24 小时规则的问题。如果任何 PM 202 在 24 小时期间内被使用,就必须针对 PVU 付费。例如,存在 2 个 AS VM。一个 VM 在午夜运行而在早上 6 时被销毁,第二 VM 在中午被创建出来而在下午 6 点被销毁。对于这种情况,PVU 的消耗是 200 个,虽然在任何时候都只有 100 个 PVU 在使用。24 小时规则的存在是为了避免许可压缩。因此,迁移 VM 可能引入额外的花费。这可以通过使用特定的 PVU 计数规则来加以解决。这些规则可作为约束来输入,以及根据本发明原理应用被采用,以确保授权一致性以及优化

使用。

[0066] 在另一示例中,如上所述,在 24 小时规则下,针对给定天的使用为 200 个 PVU。如果没有新的 VM 被布置,则第二天的使用将是 0,如果 VM 已停止使用的话。然而,IBM®许可度量工具 (ILMT) 帮助客户确定其全部和虚拟化容量 (子容量)PVU 许可需求 (或其他 PVU 度量工具),其不知道 VM 已停止使用。例如,假设没有发生变化,这些工具将继续针对该 4 个星期计数。这也可以根据本发明原理通过使用许可认知布置得到解决。

[0067] 参考图 8,建模结构 302 可以由模块 112(以及系统 180)使用,以提供应用、虚拟机、计算机实体等的许可认知布置。VM 的布置会影响软件许可使用。软件许可费用取决于软件所安装在的机器的类型和容量以及软件实例可用的资源。动态布置可能导致违反软件许可规则。根据一种示意性的实施方式,结构 302 通过连接定义关系,以及捕捉软件组件的软件许可限制。每个 VM/应用 306 和容器 308 与一个或者多个授权桶 (BoE) 304 相关联。BoE 304 包括许可容量和许可使用计算规则,也可以用来对其他类型约束进行建模。许可限制类型可以基于完全容量、基于子容量、基于实例数量、基于物理机器数量等等。

[0068] 一个或者多个拓扑 310 包括在由许可认知布置监测的环境中。拓扑 310 提供用于 VM/应用 306 和容器 308 的结构。拓扑 310、VM/应用 306 和容器 308 都可以具有由资源管理器 (110,图 1) 处理的资源需求,以及包括可能与性能、系统管理、许可等相关的约束。

[0069] BoE 304 在拓扑创建时被定义,以及在部署时被提供给布置系统。可以在运行时使用应用监测来检测 BoE 304。在一种方法中,VM 映像 306 与 BoE 304 相关联。新的 BoE 304 可以在新的软件安装到 VM 中时,由用户进行定义。布置方法 (例如,参见图 9 和图 10) 保证 BoE 304 不被超出 (例如,在强制模式)。在一种实施方式中,操作模式可以包括强制模式,在这种模式中,许可限制决不会被超出。其他模式可以包括警告模式,例如,询问用户强制是否导致性能欠佳,或者询问用户强制是否妨碍拓扑部署。其他模式可以基于特定系统进行适配。

[0070] 建模原理 302 有利地允许将通用扩展添加到布置约束 314。这使得它们很容易识别。授权允许与在单个机器或应用上的约束相比更加复杂的约束 314 的表示——授权约束可以跨越多个应用等等。授权可以在制品之间共享。有利地,授权以在将来针对不同类型的许可乃至不同类型的复杂约束支持额外实现的方式被添加到布置问题的表示中。

[0071] 授权提供器或 BoE 304 可以包括关于下述的表征:有多少给定授权可用,如何计算有多少授权当前正在使用;以及如何计算所建议改变对授权使用的影响。BoE 304 可以包括公式、具有需求的查找表、程序逻辑等等。授权客户端 312 附加到现有布置问题中的制品 (例如,应用 /VM 306,容器 308 等等) 上。单个授权客户端 312 将该制品附加到单个授权提供器 304,以及携带每个制品的使用信息以帮助提供器计算总体使用。

[0072] 参考图 9,示出了根据一种实施方式的、包括授权的示意性布置方法。对于初始布置,在块 402 中,提出改变以及进行计算。在块 404 中,确定布置方案是否有改进。如果没有改进,则最后一个最佳解决方案是在块 406 的输出。如果实现了改进,在块 408 对布置方案进行修改以避免违反授权。如果在块 410 有足够的时间,则该方法再次迭代回到块 402。如果没有足够的时间,则该方法转到块 406。

[0073] 参考图 10,示出了其中根据另一种实施方式的、包括授权的示意性布置方法。对于初始布置,在块 412 中,当内在地考虑了授权的情况下,计算来自候选布置的改变。在块

414, 确定布置方案是否有足够的改进。如果没有改进, 则最后一个最佳解决方案是在块 416 的输出。如果实现了改进以及如果在块 420 有足够的时间, 则该方法再次迭代回到块 412。如果没有足够的时间, 则该方法转到块 416。

[0074] 已描述了用于对虚拟环境中的许可授权进行管理的系统和方法的优选实施方式 (其旨在示意说明而非旨在限制), 可以注意到, 本领域技术人员在本发明的教导下可以做出修改和变种。因此应当理解, 可以在所公开的具体实施方式上做出改变而其仍在由所附权利要求所描述的本发明的范围之内。已在此描述了本发明的方面, 以及由专利法所要求的细节和特征, 专利证书所要求和希望保护的权利要求将在所附权利要求中陈述出来。

101

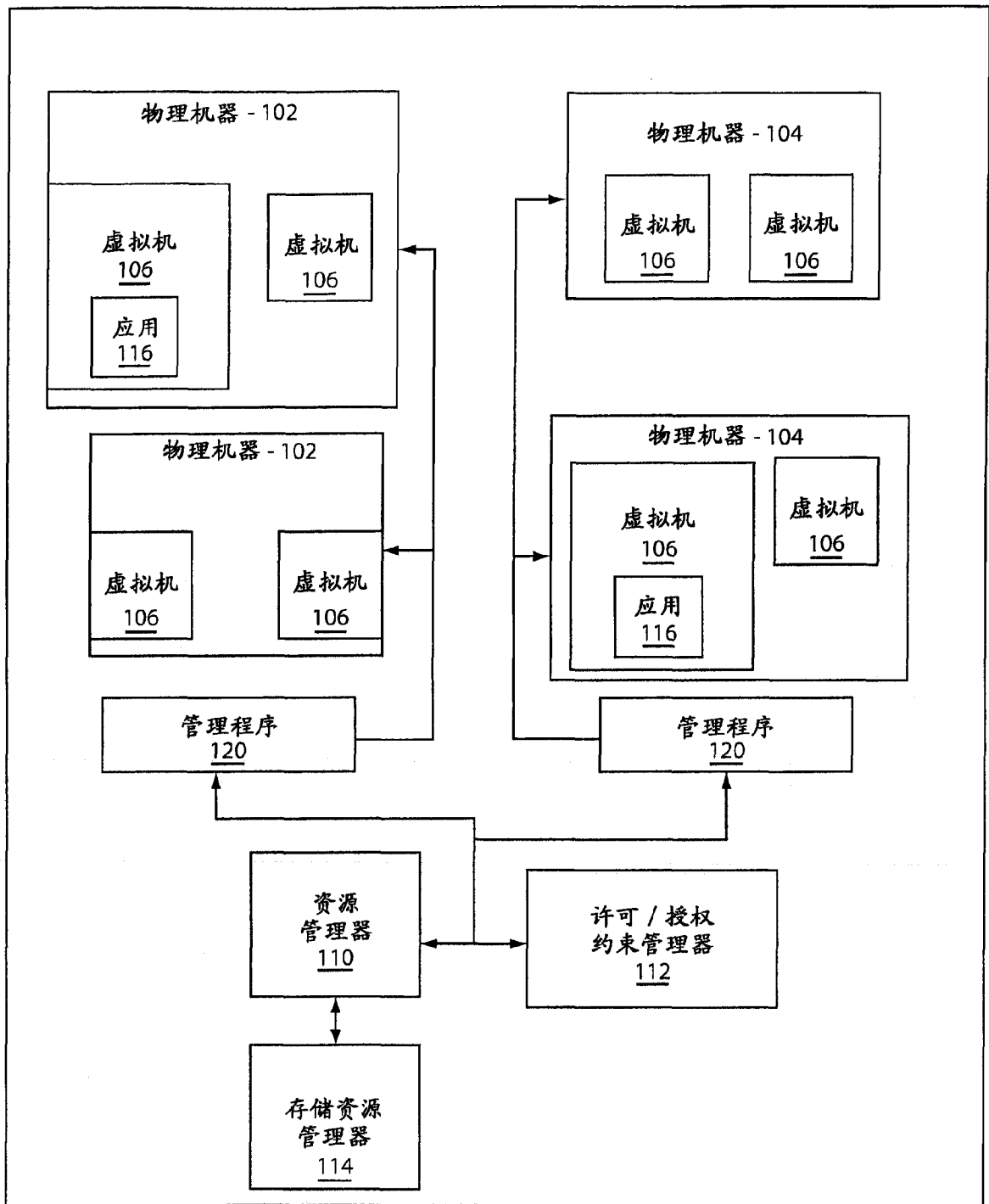


图 1

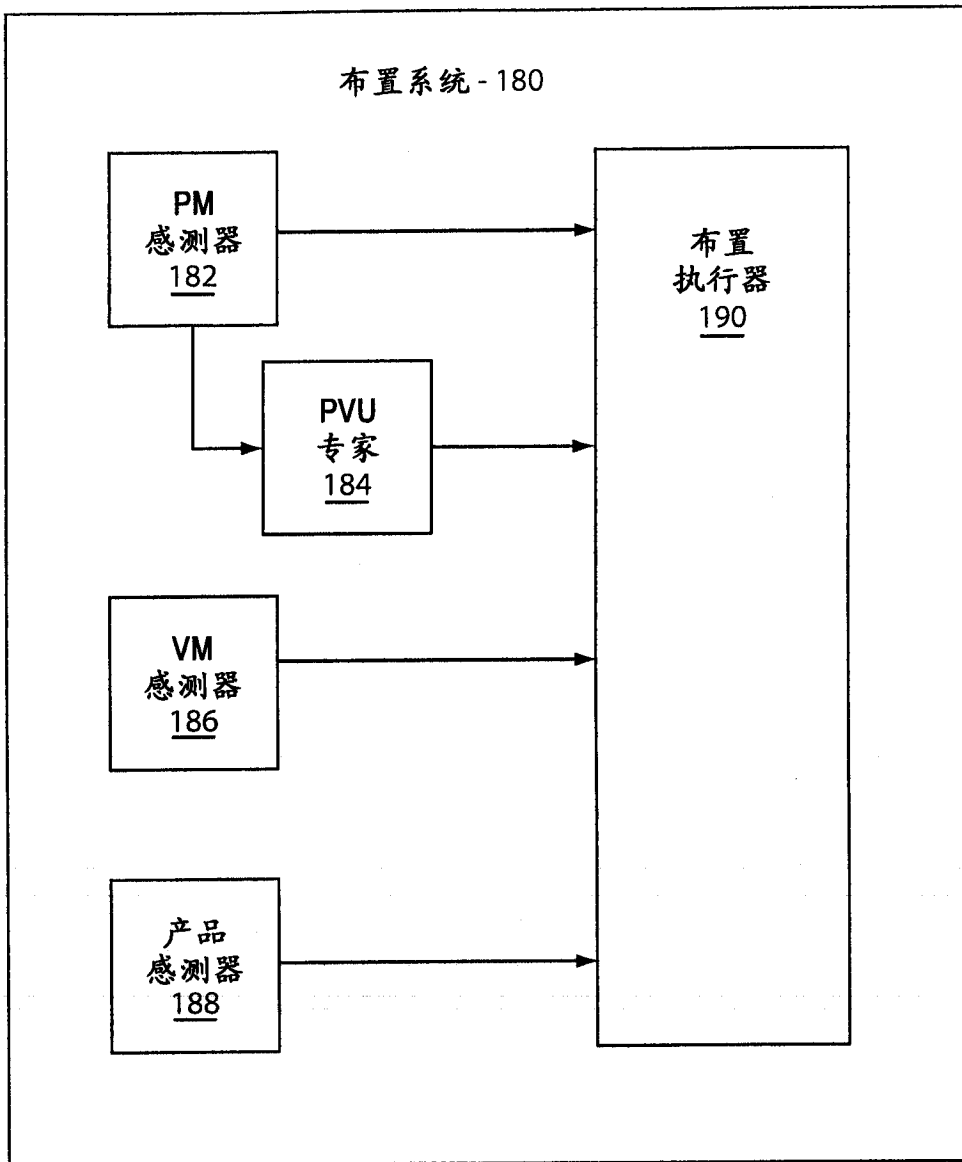


图 2

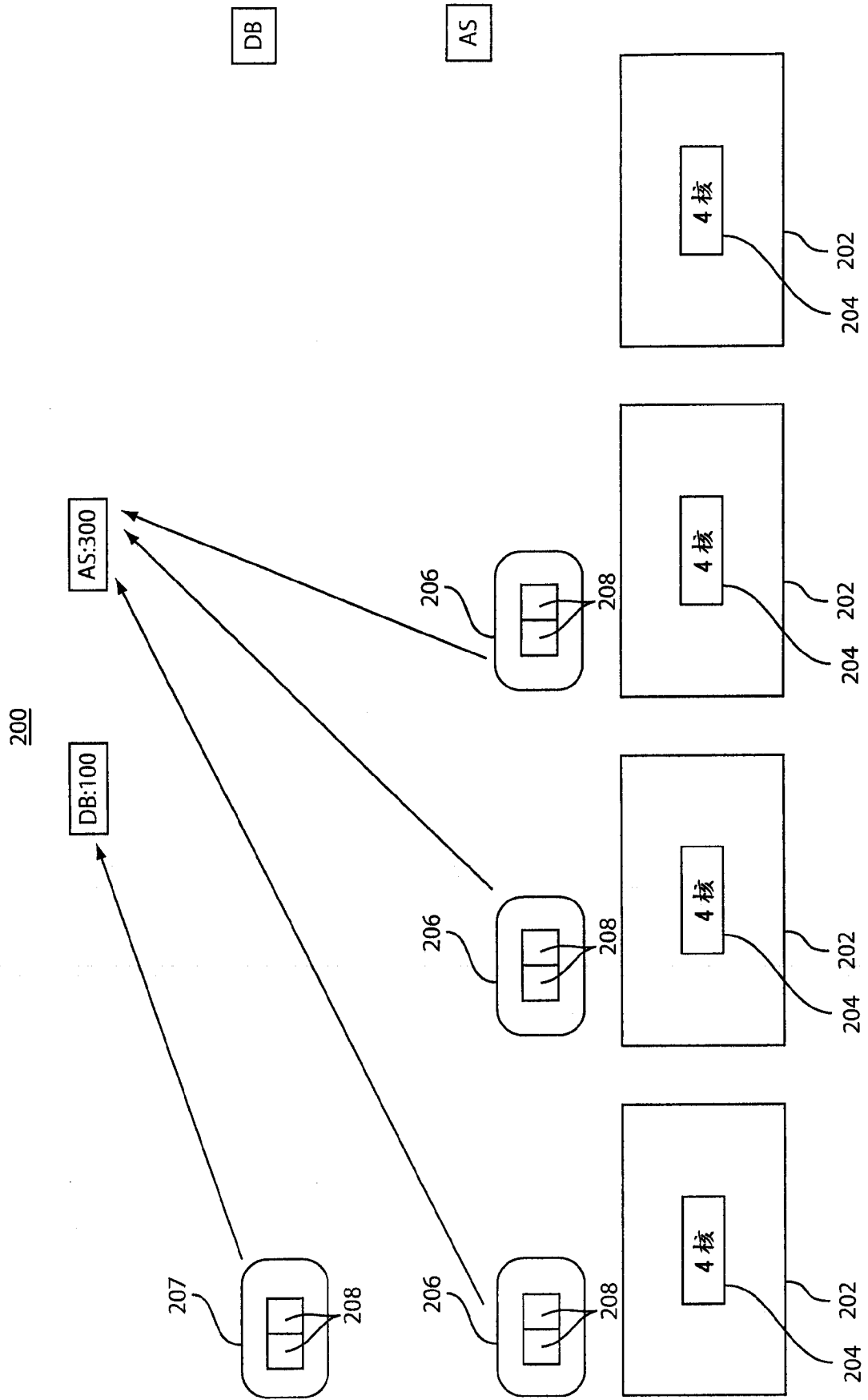


图 3

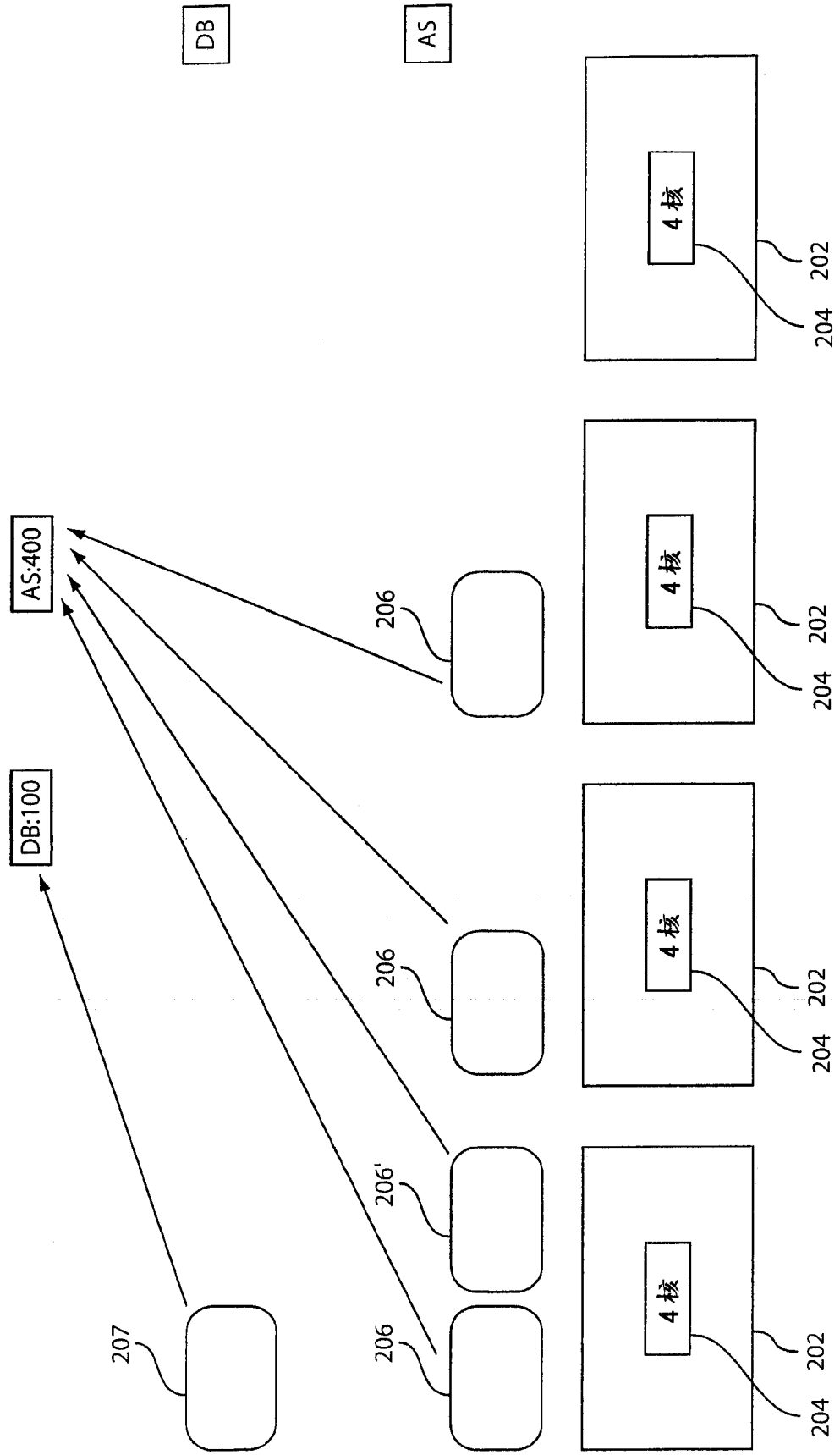


图 4

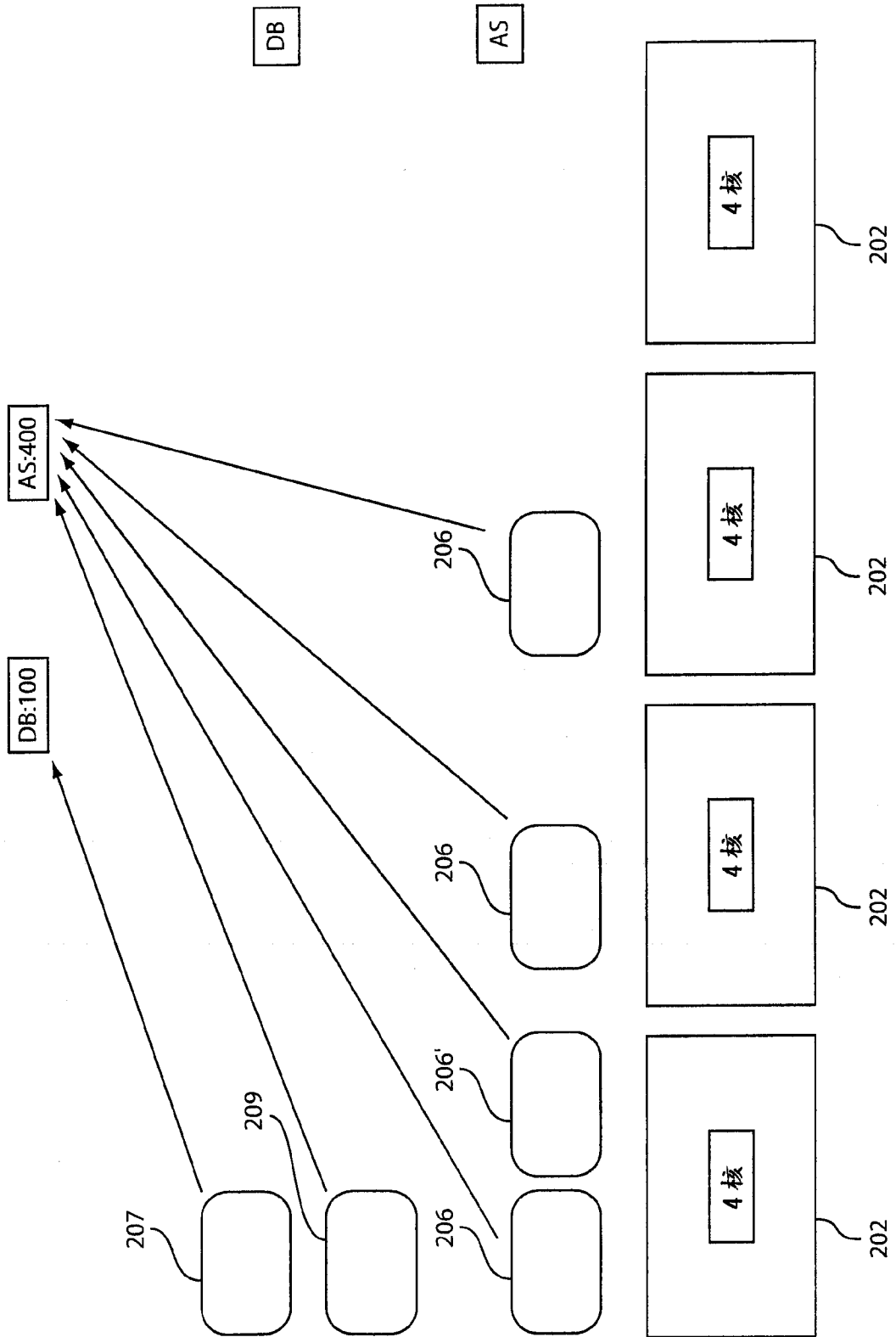


图 5

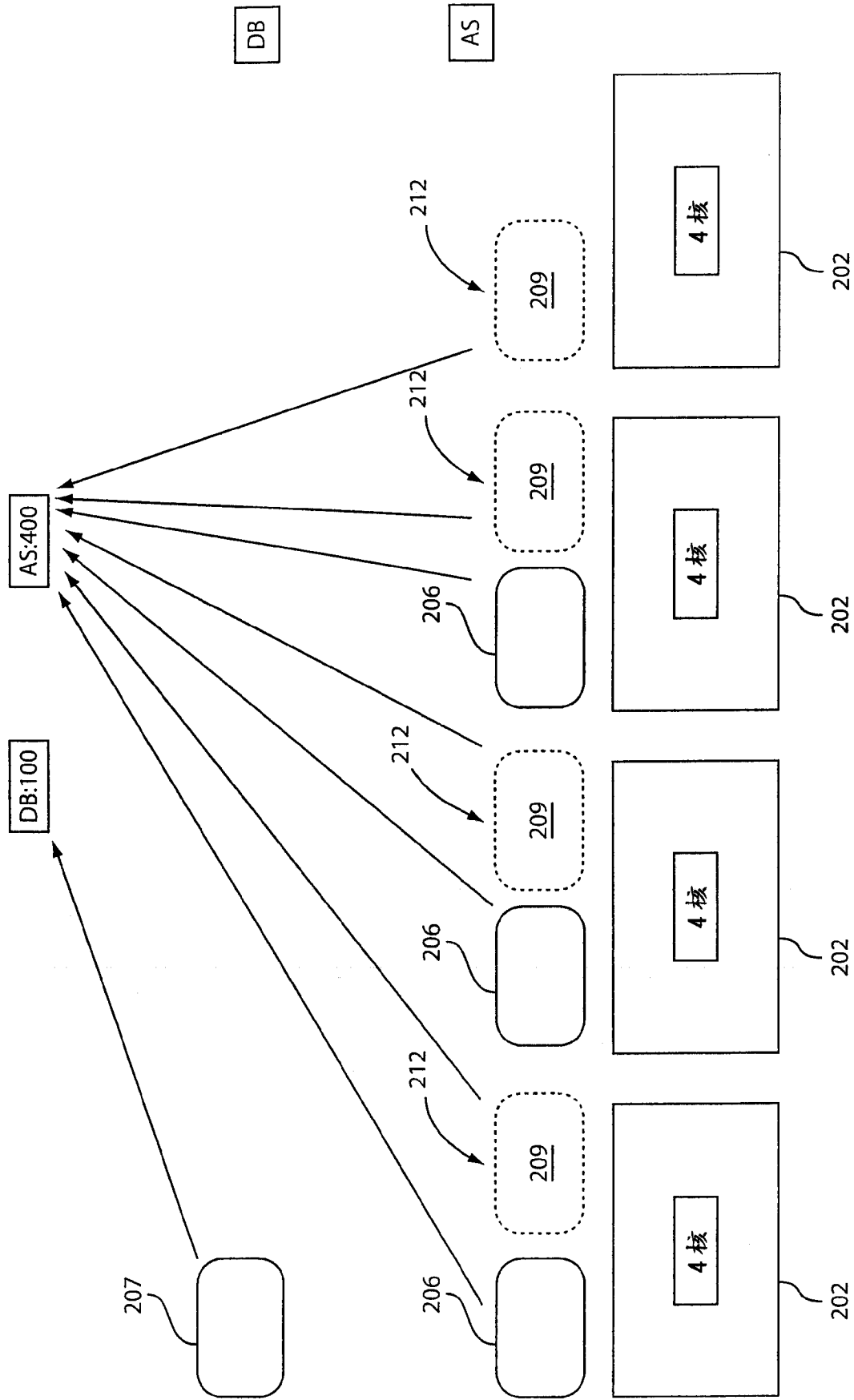


图 6

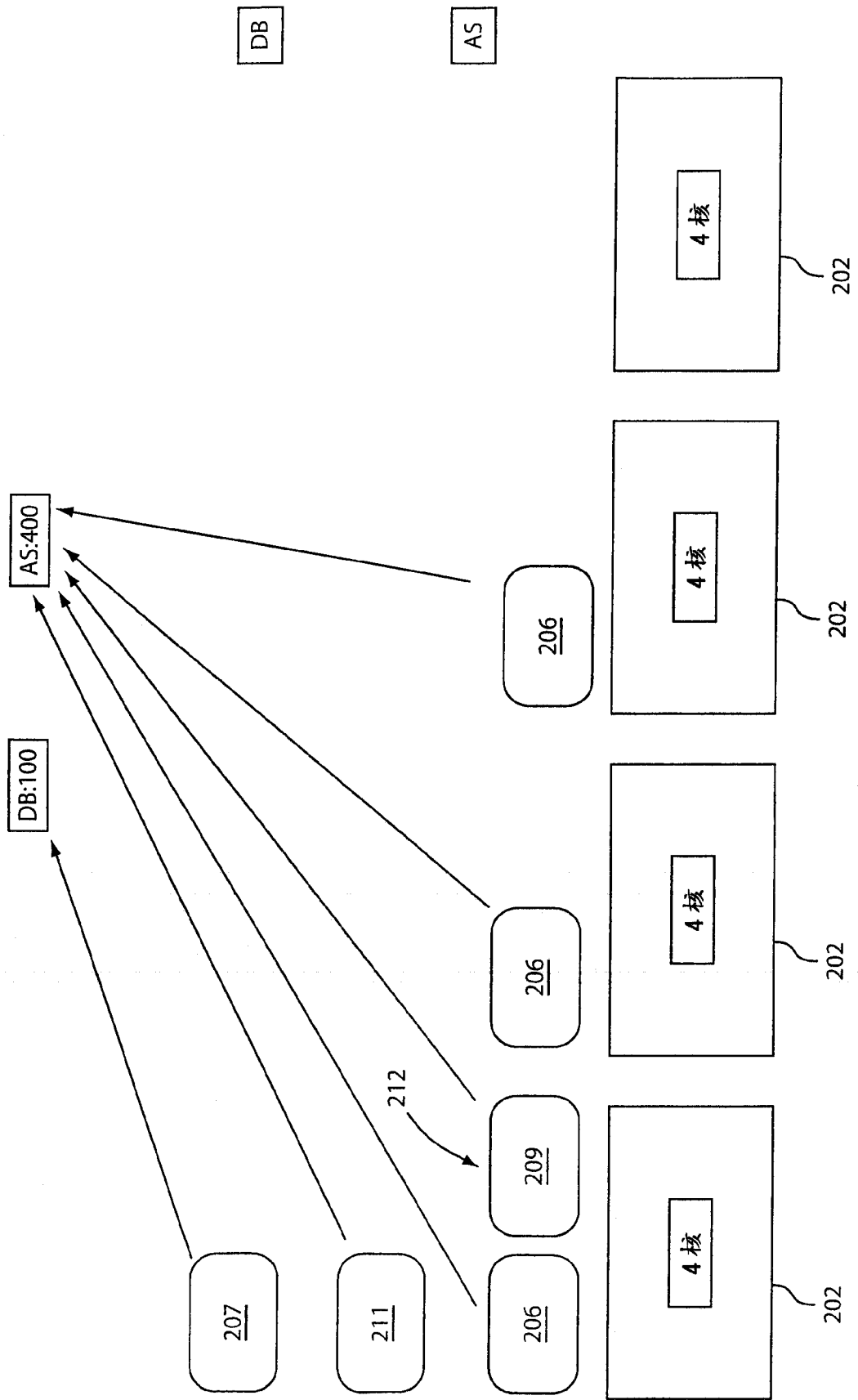


图 7

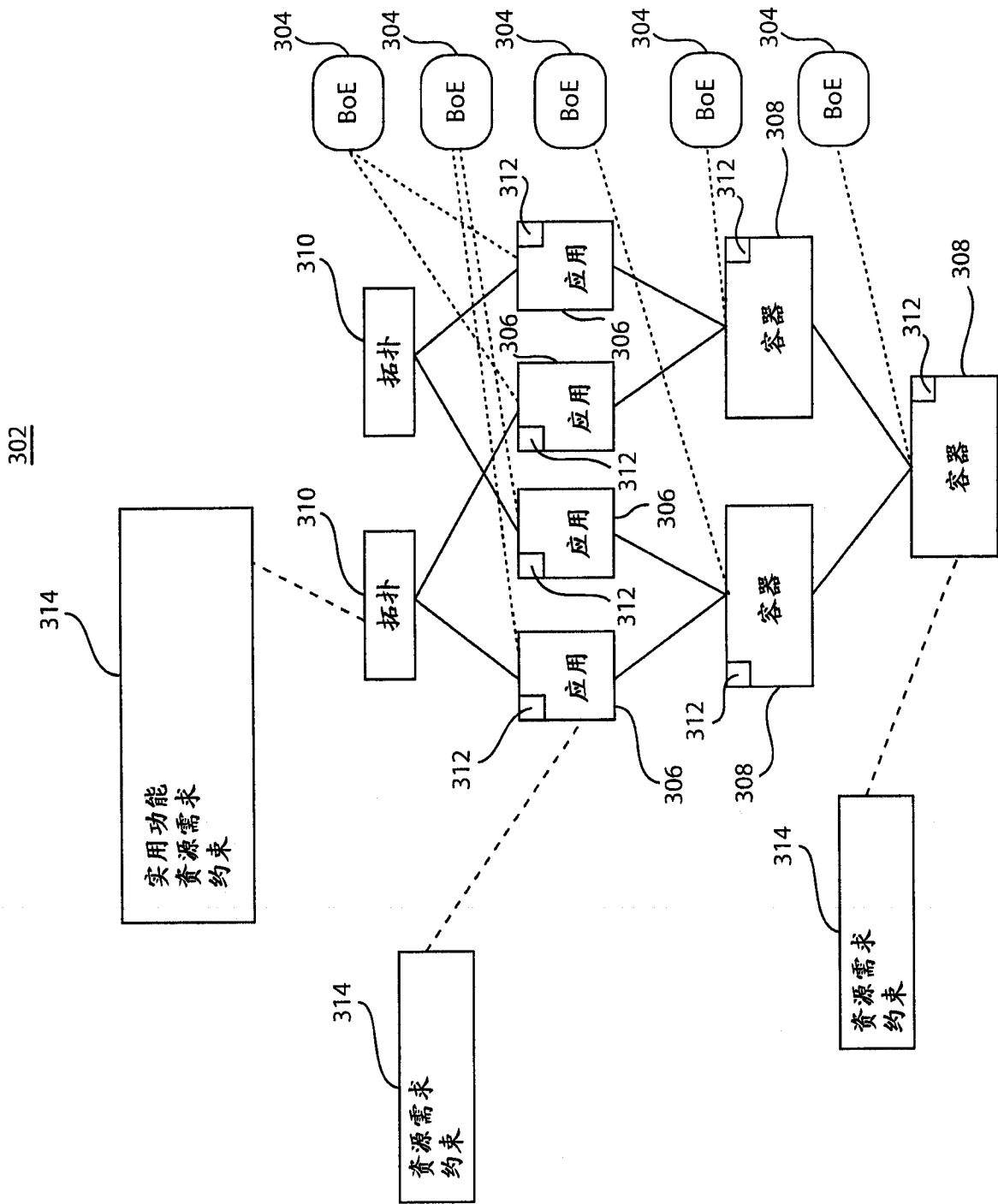


图 8

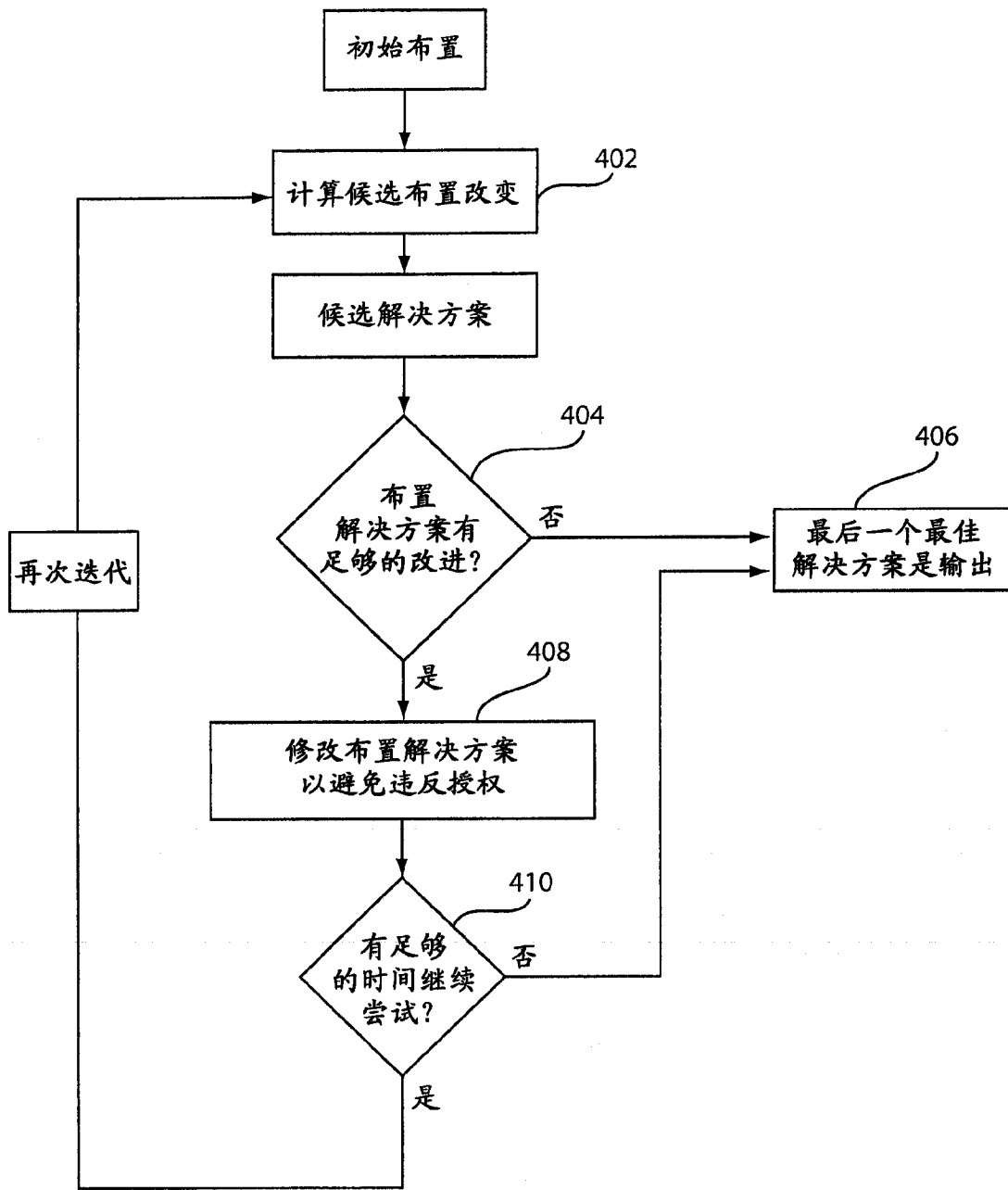


图 9

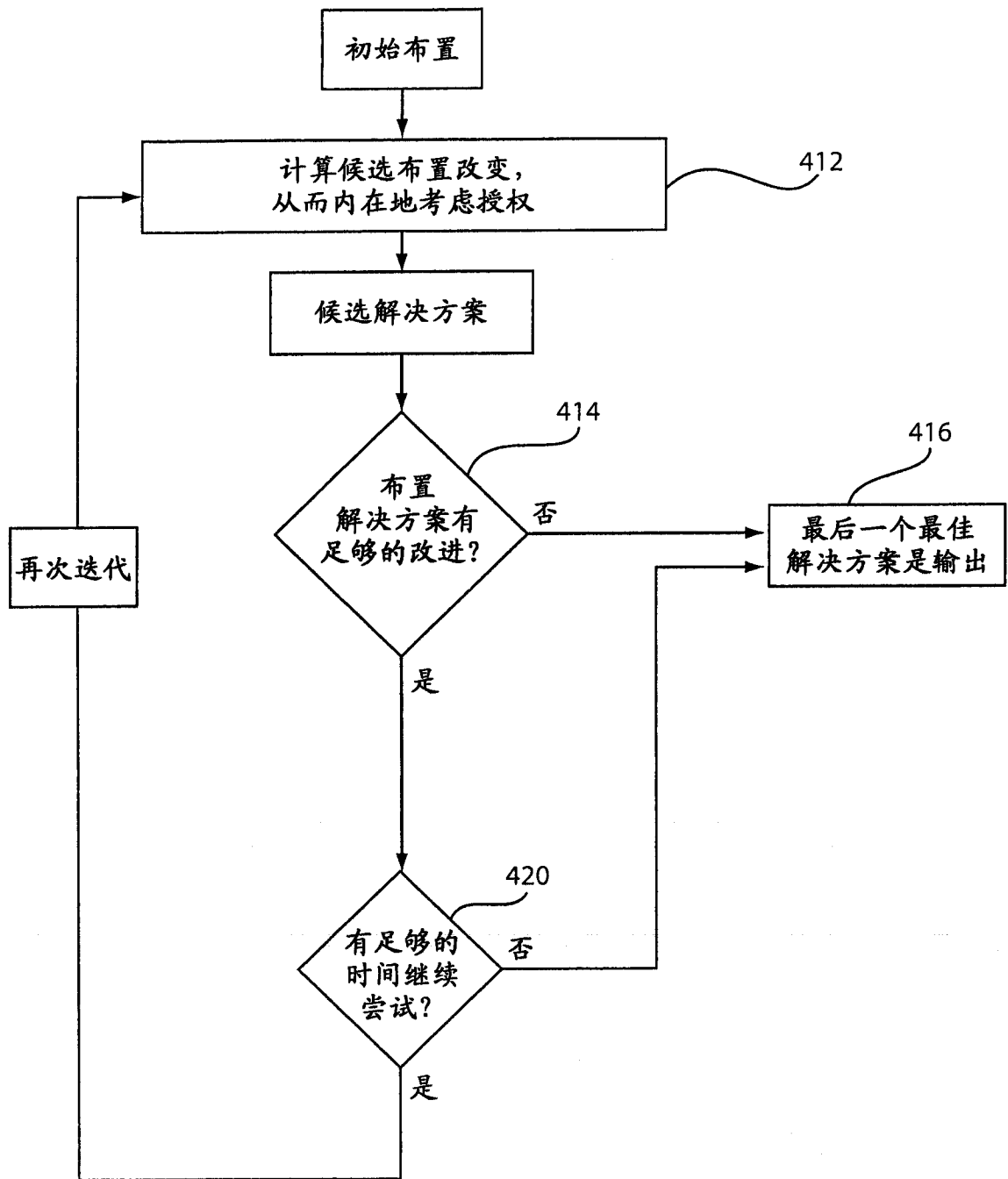


图 10