(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101631110 B
(45)授权公告日 2013.01.02

(21)申请号 200810133904.6
(22)申请日 2008.07.15
(73)专利权人 国际商业机器公司
地址 美国纽约
(72)发明人 韩竹 郑凯 梁志勇 邵凌
(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038
代理人 鲍进
(51)Int.Cl.
H04L 29/06(2006.01)
H04L 12/28(2006.01)
(56)对比文件
CN 101207604 A,2008.06.25, 全文.
CN 1776652 A,2006.05.24, 全文.
WO 2007126835 A2,2007.11.08, 说明书第22页第16行至第26页第31行.
审查员 李彬

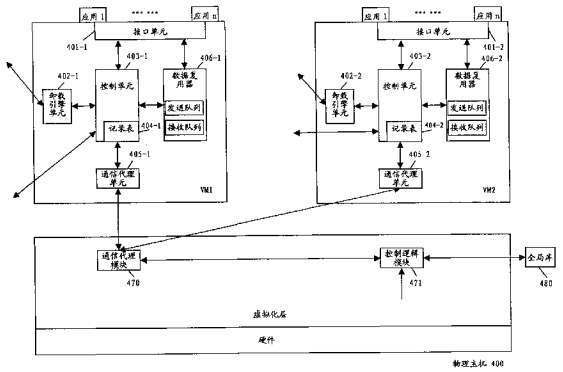
权利要求书 2 页 说明书 11 页 附图 9 页

(54)发明名称

基于相对位置动态确定连接建立机制的装置和方法

(57)摘要

本发明涉及基于相对位置动态确定连接建立机制的装置和方法，具体地，公开了一种根据虚拟机所处的位置动态地确定虚拟机之间的连接建立机制的装置和方法，所述装置包括：通信代理单元，用于接收与虚拟机所处的位置有关的消息；以及控制单元，用于基于接收到的消息，确定虚拟机之间的连接建立机制，并控制根据所确定的连接建立机制建立虚拟机之间的连接。使用本发明，能够通过判断两台虚拟机是否处于同一台物理主机上来在不断开连接的同时动态地进行两种连接建立机制之间的无缝切换，从而使系统性能始终达到最佳，并与现有的应用程序兼容，从而大大降低产品的成本。



CN 101631110 B

1. 一种根据虚拟机所处的位置动态地确定虚拟机之间的连接建立机制的装置，包括：

通信代理单元，用于接收与虚拟机所处的位置有关的消息；以及

控制单元，用于基于接收到的消息，确定虚拟机之间的连接建立机制，并控制根据所确定的连接建立机制建立虚拟机之间的连接，

如果与虚拟机所处的位置有关的消息包括指示虚拟机处于同一物理主机上的信息，则控制单元确定要使用的连接建立机制为轻量级协议，

其中如果与虚拟机所处的位置有关的消息包括指示虚拟机处于不同物理主机上的信息，则控制单元确定连接建立机制为 TCP/IP 协议，

所述装置还包括：

数据复用器，用于在控制单元的控制下，将发送 / 接收数据附接到建立的连接上，

其中，控制单元通过控制数据复用器锁定发送 / 接收数据，根据所确定的另一种连接建立机制建立连接，控制数据复用器解除锁定，以及将发送 / 接收数据附接到建立的连接上来实现切换。

2. 如权利要求 1 所述的装置，其中，控制单元还包括：

记录表，用于记录与每个连接有关的信息，所述信息至少包括连接二端的虚拟机的标识符 ID、IP 地址和端口号以及已发送和接收的分组的数目中的至少一个。

3. 如权利要求 1 所述的装置，还包括：

卸载引擎单元，用于使用试探法确定连接建立机制之一所需的信息。

4. 如权利要求 3 所述的装置，其中，所述连接建立机制之一所需的信息包括起始序列号和顺序号，

其中，所述起始序列号是基于网络时间协议值的散列值、基于发生迁移的虚拟机的 IP 地址和缺省值中的一个，而顺序号为该散列值与 已发送 / 接收的分组的数目之和。

5. 如权利要求 1 所述的装置，其中，控制单元还用于注册虚拟机。

6. 如权利要求 1 所述的装置，其中，与虚拟机所处的位置有关的消息包括指示虚拟机处于同一物理主机上或者虚拟机处于不同物理主机上的信息。

7. 如权利要求 1 所述的装置，其中，在虚拟机之间已经基于一种连接建立机制建立了连接的情况下，如果控制单元基于接收到的消息确定应当使用另一种连接建立机制，则控制单元执行连接建立机制的切换。

8. 如权利要求 1 所述的装置，还包括：

卸载引擎单元，用于使用试探法确定连接建立机制之一所需的信息；

其中，在从轻量级协议切换到 TCP/IP 协议的情况下，卸载引擎单元使用试探法建立用于建立 TCP/IP 连接的 TCP/IP 控制块。

9. 如权利要求 1 所述的装置，其中，所述消息包括以下字段中的至少一个：消息类型、虚拟机的 IP 地址、端口号、和 ID。

10. 一种根据虚拟机所处的位置动态地确定虚拟机之间的连接建立机制的方法，包括步骤：

接收与虚拟机所处的位置有关的消息；

基于接收到的消息，确定虚拟机之间的连接建立机制；以及

控制根据所确定的连接建立机制建立虚拟机之间的连接，

其中，所述确定虚拟机之间的连接建立机制的步骤包括：

如果与虚拟机所处的位置有关的消息包括指示虚拟机处于同一物理主机上的信息，则确定要使用的连接建立机制为轻量级协议，

其中，所述确定虚拟机之间的连接建立机制的步骤还包括：

如果与虚拟机所处的位置有关的消息包括指示虚拟机处于不同物理主机上的信息，则确定连接建立机制为 TCP/IP 协议，

其中所述方法还包括步骤：

根据控制，将发送 / 接收数据附接到建立的连接上，

其中，通过控制锁定发送 / 接收数据，根据所确定的另一种连接建立机制建立连接，控制解除锁定，以及将发送 / 接收数据附接到建立的连接上来实现切换。

11. 如权利要求 10 所述的方法，其中，与虚拟机所处的位置有关的消息包括指示虚拟机处于同一物理主机上或者虚拟机处于不同物理主机上的信息。

12. 如权利要求 10 所述的方法，还包括：

在虚拟机之间已经基于一种连接建立机制建立了连接的情况下，如果基于接收到的消息确定应当使用另一种连接建立机制，则执行连接建立机制的切换。

13. 如权利要求 10 所述的方法，还包括：

在从轻量级协议切换到 TCP/IP 协议的情况下，使用试探法建立用于建立 TCP/IP 连接的 TCP/IP 控制块。

14. 如权利要求 10 所述的方法，还包括用于在虚拟化层中确定虚拟机所处的位置的步骤，该步骤包括：

确定与虚拟机所处的位置有关的信息；以及

向虚拟机发送包括所确定的信息的消息。

15. 如权利要求 14 所述的方法，其中，所述确定与虚拟机所处的位置有关的信息的步骤包括：

响应于接收到的连接建立请求而查询全局库、或者基于接收到的系统信息，确定虚拟机是在同一物理主机上还是不在同一物理主机上。

16. 如权利要求 15 所述的方法，其中，所述确定与虚拟机所处的位置有关的信息的步骤还包括：

如果接收到的系统信息指示虚拟机迁移出物理主机，则确定虚拟机不在同一物理主机上，同时在全局库中注销迁移出的虚拟机。

17. 如权利要求 15 所述的方法，其中，所述确定与虚拟机所处的位置有关的信息的步骤还包括：

如果接收到的系统信息指示虚拟机迁移入物理主机，则给迁移入的虚拟机分配 ID，同时在全局库中注册迁移入的虚拟机，并确定虚拟 机在同一物理主机上。

18. 如权利要求 14-17 任一项所述的方法，还包括：

向全局库注册和注销虚拟机，

其中，注册是通过向全局库中存储该虚拟机的 IP 地址、ID 和虚拟化层的 ID 而实现的，而注销是通过从全局库中删除与该虚拟机有关的信息而实现的。

基于相对位置动态确定连接建立机制的装置和方法

技术领域

[0001] 本发明涉及服务器虚拟化技术，更具体地，涉及一种在虚拟环境中根据虚拟机(Virtual Machine，简称为“VM”）所处的相对位置动态地确定虚拟机之间的连接建立机制的装置和方法及系统。

背景技术

[0002] 由于人们越来越强地认识到服务器资源的利用率低下以及服务器整合的必要性，并且多核处理器的出现让单台服务器的性能越来越强大，服务器虚拟化开始吸引更多厂商的关注。通过将物理服务器资源分配到多个虚拟机，服务器虚拟化支持不同的应用，甚至不同的操作系统 (Operation System，简称为“OS”）能在同一企业级服务器上同时运行。每个虚拟机就像一台独立的服务器，但实际上可能就是在同一物理服务器内运行。在一台服务器上运行多个应用能够提高服务器效率，并减少需要管理和维护的服务器数量。当应用需求增加时，可以迅速创建更多虚拟机，从而无需增加物理服务器即可灵活地响应不断变化的需求。

[0003] 近来，服务器虚拟化成为系统研究和解决方案领域中的热门话题之一。服务器虚拟化的方向之一是把若干个分散的物理服务器（以下也称为“物理主机”）虚拟为一个大的逻辑服务器。而且，利用这种虚拟技术，IT管理员可以在物理服务器之间移动正在运行的虚拟机，同时保持虚拟服务器持续可用。服务器虚拟化的一个重要特征就是虚拟机的动态迁移。动态迁移就是在“带电”情况下，将一个虚拟机从一个物理主机移动到另一个物理主机的过程。动态迁移过程不会对最终用户造成明显的影响，这是由于动态迁移可以使得停机时间为毫秒级，这对用户而言很难觉察，从而使得管理员能够在不影响用户正常使用的情况下，对物理服务器进行离线维修、升级、配置、负载平衡、或者管理等等。图 1 示意性地示出了在通过 LAN（局域网）连接多台物理服务器的虚拟环境中虚拟机从一台物理主机（即，物理主机 1）迁移到另一台物理主机（即，物理主机 2）的情况。图 1 仅仅是说明性的，本领域技术人员应当理解可以根据需要调整 LAN 中物理服务器的数量以及每台物理服务器上的虚拟机的数量。

[0004] 通常，一台虚拟机上的某些应用需要通过高性能专用信道与其它虚拟机上的应用通信，这些应用诸如是 IDS（入侵检测系统）和防火墙应用、防火墙和 VPN（虚拟专用网）应用。最常见的情况是通过以太网来连接不同虚拟机，并且要连接的不同应用之间的流量要经过 TCP/IP 栈。当将不同的应用整合到一个物理主机上时，专用信道可以被优化。也就是说，由于所有流量都是在 RAM（随机存储器）中移动，所以可以在无需 TCP/IP（传输控制协议 / 网际协议）栈中的校验和保护、按序传送、拥塞控制和净荷封装的情况下建立专用信道。从而，通过使用省略了由 TCP/IP 栈导致的大部分开销的薄协议层（以下称之为“轻量级协议”），大大提高了专用信道的性能其中，本领域所谓的轻量级协议指的是利用共享内存机制，在同一物理主机上进行通信的开销较小的一类通信方式或者 / 及其实现的总称。

[0005] 图 2 示意性示出了现有技术中的两台虚拟机之间通过 TCP/IP 协议进行通信的系

统的示意图。在图 2 所示的情况下，无论虚拟机是否在同一物理主机上，虚拟机之间的通信都采用本领域公知的 TCP/IP 协议来建立连接并在建立的连接上进行彼此之间的消息传送。采用 TCP/IP 协议建立虚拟机之间的信道的好处在于实现了最佳的灵活性，即使在连接建立之后发生了虚拟机迁移，也不会使两台虚拟机之间的连接断开。此外，由于 TCP/IP 协议是现有计算机网络中普遍采用的协议，所以无需对虚拟机中的 TCP/IP 协议栈实现做任何改变，从而实现了与现有应用的良好兼容。但是另一方面，采用 TCP/IP 协议建立虚拟机之间的信道的缺陷在于：当虚拟机位于同一物理主机上时，使用 TCP/IP 协议导致系统开销大，这是由于所有流量都是在 RAM 中移动，从而使得 TCP/IP 协议规定必须实现的校验和保护、按序传送、拥塞控制和多层净荷封装是无用的。

[0006] 图 3 示意性示出了现有技术中的在同一物理主机上的两台虚拟机之间通过轻量级协议进行通信的系统的示意图。轻量级协议诸如是共享内存或通过固件实现的其它轻量级协议。采用轻量级协议建立虚拟机之间的信道的好处在于实现了系统的最佳性能，即系统开销小、处理速度快、运行应用占用的物理资源小。采用轻量级协议建立虚拟机之间的信道的另一缺陷在于当已经通过轻量级协议建立了连接的两个虚拟机中的任一个发生迁移时，都会导致连接的断开，这是由轻量级协议固有的特性导致的。因此，轻量级协议连接缺少灵活性。

[0007] 例如，Wei Huang 等人于 2007 年 11 月 10-16 日在 Proceedings ofthe 2007 ACM/IEEE Conference on Supercomputing 上发表的标题为“Virtual Machine Aware Communication Libraries for HighPerformance Computing”的论文已经公开了当两台虚拟机在同一物理机器上时如何建立起高效的通信方式。具体而言，该论文提出了一种虚拟机知道的通信库来支持同一物理主机上的计算处理之间的高效共享内存通信。显然，该论文也没有涉及当已经建立连接的虚拟机发生动态迁移之后，如何进行处理。因此，该文献没有解决现有的轻量级技术存在的缺陷。

[0008] 因此，需要一种能够根据虚拟机所处的位置动态地确定虚拟机之间的连接建立机制的装置和方法及系统。

发明内容

[0009] 考虑到现有技术中存在的上述问题，而做出本发明。本发明的一个目的是提供一种根据虚拟机所处的位置动态地确定虚拟机之间的连接建立机制的装置和方法及系统。

[0010] 为了实现上述目的，提供了一种根据虚拟机所处的位置动态地确定虚拟机之间的连接建立机制的装置，包括：通信代理单元，用于接收与虚拟机所处的位置有关的消息；以及控制单元，用于基于接收到的消息，确定虚拟机之间的连接建立机制，并控制根据所确定的连接建立机制建立虚拟机之间的连接。

[0011] 为了实现上述目的，提供了一种根据虚拟机所处的位置动态地确定虚拟机之间的连接建立机制的方法，包括步骤：接收与虚拟机所处的位置有关的消息；基于接收到的消息，确定虚拟机之间的连接建立机制；以及控制根据所确定的连接建立机制建立虚拟机之间的连接。

[0012] 为了实现上述目的，提供了一种用于在虚拟化层中确定虚拟机所处的位置的方法，包括：确定与虚拟机所处的位置有关的信息；以及向虚拟机发送包括所确定的信息的

消息。

[0013] 为了实现上述目的，提供了一种用于在虚拟化层中确定虚拟机所处的位置的装置，包括用于实现上述方法中的各个步骤的部件。

[0014] 通过使用本发明，可以根据虚拟机的位置动态地确定连接建立机制，使得当两台虚拟机都在同一物理主机上的时候使用轻量级协议，而当虚拟机发生动态迁移而使得两台虚拟机不在同一物理主机上的时候仍然使用普通的 TCP/IP 协议栈，从而解决了开销大和不能保持连接性的问题，并使系统性能始终保持最佳。此外，由于 TCP/IP 协议和轻量级协议本身是现有技术中已经实现的，所以可以最大程度地兼容现有应用程序，从而可以为用户节省成本。

附图说明

[0015] 从下面结合附图的详细描述中，本发明将会更易于理解，其中，相同的附图标记表示相同的结构元素，并且，附图中：

[0016] 图 1 是示出了现有技术中的在通过 LAN 连接多台物理主机的虚拟环境中虚拟机从一台物理主机迁移到另一台物理主机的图。

[0017] 图 2 是示出了现有技术中的两台虚拟机之间通过 TCP/IP 协议进行通信的系统的示意图。

[0018] 图 3 是示出了现有技术中的在同一物理主机上的两台虚拟机之间通过轻量级协议进行通信的系统的示意图。

[0019] 图 4 是示意性示出了根据本发明的根据虚拟机所处的位置动态地确定虚拟机之间的连接建立机制的虚拟系统的框图。

[0020] 图 5 是举例说明在如上所述建立虚拟机之间的轻量级连接后发生虚拟机迁移的情况下将连接建立机制切换到 TCP/IP 协议的情况的系统图。

[0021] 图 6 是举例说明在如上所述建立虚拟机之间的 TCP/IP 连接后发生虚拟机迁移的情况下将连接建立机制切换到轻量级协议的情况的系统图。

[0022] 图 7 是示出了根据本发明的、在虚拟机与虚拟化层之间交换的消息的形式的示意图。

[0023] 图 8 是示出了根据本发明的在虚拟化层中确定虚拟机所处的位置的方法的流程图。

[0024] 图 9 是示出了图 8 所示的步骤 802 的具体步骤的流程图。

[0025] 图 10 是示出了根据本发明的根据虚拟机所处的位置动态地确定虚拟机之间的连接建立机制的方法的流程图。

具体实施方式

[0026] 现在将以具体的、示例性的实施例描述本发明。应该理解，本发明不限于所披露的示例性实施例。还应该理解，目前所披露的在虚拟环境中根据虚拟机所处的位置无缝切换虚拟机之间的连接建立机制的方法和装置及系统的每一个特征，并非都是实现所附权利要求任一具体项要求保护的发明所必不可少的。描述设备的多个元件和特征是为了使本发明完全能够得以实现。还应该理解的是，在本说明书中，在表示或者描述处理或方法之处，方

法的步骤可以按照任何顺序执行或者同时执行，除非从上下文中显然可以看出一个步骤依赖于先前执行的另一步骤。

[0027] 本发明的核心思想是：在建立连接时首先检测两台虚拟机所处的位置，如果要建立连接的两台虚拟机处于同一物理主机上，则使用轻量级协议来建立连接，而如果要建立连接的两台虚拟机处于不同的物理主机上，则使用 TCP/IP 协议来建立连接。此外，在同一物理主机上的两台虚拟机已经通过轻量级协议建立连接之后其中一台虚拟机发生动态迁移从而去到另一物理主机上时，在保持连接的同时将所使用的连接建立机制从轻量级协议无缝切换到 TCP/IP 协议，而在不同物理主机上的两台虚拟机已经通过 TCP/IP 协议建立连接之后其中一台虚拟机发生动态迁移从而去到另一虚拟机所在的物理主机上时，在保持连接的同时将所使用的连接建立机制从 TCP/IP 协议无缝切换到轻量级协议，所谓的“无缝”指的是底层通信机制的改变对于高层应用是透明的。由于这个切换过程对高层应用而言是不可察觉的，所以本发明克服了现有技术中存在的开销大和不能保持连接性的缺陷，并获得了最佳系统性能和实现了与现有技术的最佳兼容，降低了用户的成本。

[0028] 图 4 是示意性示出了根据本发明的根据虚拟机所处的位置动态地确定虚拟机之间的连接建立机制的虚拟系统的框图。虽然在图 4 所示的虚拟系统中仅仅示出了一台物理主机，但是本领域技术人员可以理解图 4 中所示的虚拟系统可以存在具有相同配置的两台或更多台物理主机。此外，虽然在图 4 中示出了仅仅两个虚拟机存在于一台物理主机上，但是本领域技术人员可以理解在该物理主机上可以存在更多个虚拟机。

[0029] 如图 4 所示，在虚拟系统中存在物理主机 400 和全局库 480。物理主机 400 可以是诸如 Sun 公司、IBM 公司、Dell 公司等等的各个厂商出品的主机。在虚拟系统中还存在全局库 480，用于存储与虚拟系统中的各个物理主机上的虚拟机有关的信息，这在下文中将详细进行介绍。

[0030] 全局库 480 可以实现为某一物理主机上存储的或者在 LAN 上分布式存储的数据库、文件、二进制信息，或者任何其它能够存储与虚拟机有关的信息的数据结构。

[0031] 物理主机 400 从下向上分别为硬件、虚拟化层、虚拟机和多个应用。硬件是实现主机各项功能的基础，它由各个厂商提供，但是本发明不涉及对硬件的任何改变，在此不对其进行任何描述。

[0032] 在虚拟化层中包括通信代理模块 470 和控制逻辑模块 471。通信代理模块 470 是用于在虚拟化层与其上的各个虚拟机进行通信的模块，其是现有技术中已经实现的，例如，在采用半虚拟化技术的 XEN 上，利用 Hypercall 技术实现的通信代理模块；在采用全虚拟化技术的 VMWARE 上，采用特殊硬件中断实现的通信代理模块；在 Intel 和 AMD 最新的芯片上，也提供 VMCALL 等特殊指令来实现通信代理模块。因此在本文中，不再对通信代理模块 470 的实现作更进一步的描述。

[0033] 控制逻辑模块 471 用于确定与虚拟机的相对位置有关的信息。此外控制逻辑模块 471 还用于在全局库 480 中注册和注销各个虚拟机。

[0034] 控制逻辑模块 471 通过通信代理模块 470 从虚拟机中的控制单元接收到连接建立请求时，基于所接收的请求中包括的连接目标虚拟机的 IP 地址查询全局库 480 以获得目标虚拟机的虚拟化层 ID（标识符），即 HVID，通过将目标虚拟化层 ID 与自身虚拟化层 ID 相比较，确定目标虚拟机与请求虚拟机是否都在本物理主机上。随后，控制逻辑模块 471 通过通

信代理模块 470 以消息的形式将结果返回给虚拟机中的控制单元。

[0035] 控制逻辑模块 471 还接收物理主机系统检测到的虚拟机发生动态迁移的信息，该信息可以是本物理主机上的虚拟机迁移出本物理主机，也可以是其它物理主机上的虚拟机迁移入本物理主机。在收到本物理主机上的虚拟机迁移出本物理主机的信息后，控制逻辑模块 471 通过通信代理模块 470 以消息的形式将包括发生迁移的虚拟机的 IP 地址 /ID 的该信息通知给没有发生迁移的所有其它虚拟机。

[0036] 虚拟机的注册可以发生在每个虚拟机启动的时候。在启动时，虚拟机中的控制单元 403 将包括其自身 ID、IP 地址的注册请求发送给虚拟化层中的控制逻辑模块 471，并由控制逻辑模块 471 在全局库 480 中注册虚拟机自身的 ID（即，VMID）和 IP 地址（即，VMIP）、以及虚拟机所在的虚拟化层的 ID（即，HVID）。

[0037] 可替换地，虚拟机的注册也可以发生在该虚拟机要建立与其它虚拟机的连接的时候。在建立连接之前，虚拟机中的控制单元 403 将包括其自身 ID、IP 地址的注册请求发送给虚拟化层中的控制逻辑模块 471，并由控制逻辑模块 471 在全局库 480 中注册虚拟机自身的 ID（即，VMID）和 IP 地址（即，VMIP）、以及虚拟机所在的虚拟化层自身的 ID（即，HVID）。除上述情况之外，虚拟机的注册可以发生在任何时候。显然，可以多次注册同一虚拟机。

[0038] VMID 对于虚拟机始终是唯一的，HVID 对于虚拟化层始终也是唯一的。由于每台物理主机上只有一个虚拟化层，所以通过查询 VMID 和对应的 HVID，控制逻辑模块 471 就可以判断出两个虚拟机是不是在同一个虚拟化层上，也就是，是否在同一物理主机上。全局库 480 以三元组 (VMID，VMIP，HVID) 形式或者任何可以实现相同或等同功能的形式，存储每一台虚拟机的信息。

[0039] 当虚拟机发生迁移时，控制逻辑模块 471 基于虚拟机 ID 从全局库中注销该虚拟机，即删除所有包括 VMID 的记录。

[0040] 虚拟机 1 中包括供外部应用调用的接口单元 401-1、卸载引擎单元 402-1、控制单元 403-1、通信代理单元 405-1 和数据复用器 406-1。

[0041] 虚拟机 2 中包括供外部应用调用的接口单元 401-2、卸载引擎单元 402-2、控制单元 403-2、通信代理单元 405-2 和数据复用器 406-2。

[0042] 下面以虚拟机 1 中的各个模块为例进行说明，但是应当理解：其它虚拟机中的对应单元具有相同的功能，例如，虚拟机 2 中的控制单元 403-2 具有与虚拟机 1 中的控制单元 403-1 相同的功能。

[0043] 通信代理单元 405-1 是用于在虚拟机 1 与其下层的虚拟化层之间进行数据传送的模块，其是现有技术中已经实现的，在此不对其进行进一步描述。

[0044] 控制单元 403-1 接收与虚拟机的相对位置有关的消息，基于接收到的消息确定连接建立机制，并基于所确定的连接建立机制建立连接。控制单元 403-1 通过通信代理单元 405-1 从虚拟化层接收消息，该消息可以是向控制单元 403-1 通知虚拟机 1 与其要与之建立连接的目标虚拟机是否在同一物理主机上、本物理主机上的另一虚拟机迁移出本物理主机、或者另一虚拟机迁移入本物理主机等。如果接收到的消息是用于指示虚拟机 1 与其要与之建立连接的目标虚拟机是否在同一物理主机上的消息，控制单元 403-1 根据接收到的消息确定要使用的连接建立机制，即轻量级协议或 TCP/IP 协议。如果接收到的消息是用于指示本物理主机上的另一虚拟机迁移出本物理主机或者另一虚拟机迁移入本物理主机的

消息，控制单元 403-1 根据接收到的消息中包含的信息，诸如虚拟机的 ID、IP 地址等，确定虚拟机 1 是否与发生迁移的另一虚拟机已经建立了连接，如果存在已经建立的连接，控制单元 403-1 确定在虚拟机迁移后要使用的连接建立机制，控制数据复用器 406-1 将其中的发送队列和接收队列锁定，在切换到所确定的连接建立机制并基于该连接建立机制建立连接之后，控制数据复用器 406-1 将其中的发送队列和接收队列解锁并将其附接到所建立的连接上。

[0045] 控制单元 403-1 包括记录表 404-1，用于存储与每个连接有关的信息，诸如，连接双方虚拟机的 ID、IP 地址和端口号、已接收的分组数、已发送的分组数等等。这些信息仅仅是示意性的，本领域技术人员根据其需求，可以对这些内容进行添加、删除、和 / 或修改。此外，记录表 404-1 可以实现为虚拟机上存储的数据库、文件、二进制信息，或者任何其它能够存储上述信息的结构。

[0046] 控制单元 403-1 根据接收到的消息中包含的对方虚拟机的 IP 地址或 ID，查询记录表，如果记录表中有相应的记录，则说明虚拟机 1 与对方虚拟机具有已经建立的连接，有多少条记录就有多少个连接，否则，则说明虚拟机 1 与对方虚拟机没有建立任何连接，在此情况下，虚拟机 1 不做任何动作，丢弃该接收到的消息。

[0047] 控制单元 403-1 在虚拟机 1 启动或要建立连接时以及其他需要注册虚拟机 1 的时候将虚拟机 1 的 ID 及其 IP 地址通过通信代理单元 405-1 和通信代理模块 470 发送给虚拟化层中的控制逻辑模块 471 以便注册虚拟机 1。

[0048] 数据复用器 406-1 用于在控制单元 403-1 的控制下将数据附接到已建立的连接上。具体而言，数据复用器 406-1 包括用于每个已建立连接的发送队列和接收队列，分别用于缓存要发送和已接收的数据。数据复用器 406-1 在控制单元 403-1 的控制下，将通过接口单元 401-1 从应用接收到的数据切换到基于轻量级协议或 TCP/IP 协议建立的连接。

[0049] 此外，数据复用器 406-1 用于在控制单元 403-1 的控制下锁定和 / 或解锁其中的队列。具体而言，在被通知虚拟机发生迁移时，控制单元 403-1 控制数据复用器 406-1 首先锁定发送队列和接收队列，并且在控制单元 403-1 通知数据复用器 406-1 要切换到的连接已经建立成功的情况下，数据复用器 406-1 才对发送队列和接收队列进行解锁，从而继续数据的发送和接收。这个切换过程对于应用而言是透明的。由于数据在断开当前连接之前被锁定并且在要切换到的连接已建立之后才解锁，所以数据不会有任何丢失，对于应用而言就像连接没有断开一样，也就是说，保持了连接。应用在这个切换过程中最多感觉性能有所下降，但是由于切换所花费的时间是毫秒级的，所以大多数情况下应用感觉不到性能有什么变化。

[0050] 卸载引擎单元 402-1 用于在从轻量级协议切换到 TCP/IP 协议时，提供建立 TCP/IP 连接所需的 TCP/IP 栈信息。在使用常规的 TCP/IP 协议建立通信时，需要在连接的双方进行三次握手过程来协商 TCP 控制块中的各项信息，诸如起始序列号、顺序号、重传列表和乱序列表、发送窗口和接收窗口的大小、定时器的超时值和往返时间 RTT 等。

[0051] 但是本发明不需要这种握手过程。本发明使用试探法来建立这些值。例如，起始序列号使用基于 NTP(Network Time Protocol，网络时间协议) 值的散列值，这个值的计算对于本领域技术人员是公知的。起始序列号还可以是由发生迁移的虚拟机的 IP 地址获得的散列值，或者由物理主机或 LAN 预先设定的缺省值，例如，10000，只要将建立 TCP/IP 连接

的虚拟机双方能够对起始序列号达成一致即可。顺序号为起始序列号与记录表 404-1 中记录的已发送分组数 / 已接收分组数的和。清空重传列表和乱序列表，也就是说，在建立 TCP/IP 连接时不使用重传，并且使用顺序传送。将发送窗口和接收窗口的大小设置成一个中间值，诸如，8K，或者虚拟机可以接受的其它值。把所有定时器的超时值设置成 TCP/IP 协议中规定的缺省值，并把 RTT 值估计为一个较大的值，诸如 500ms。

[0052] 然后，把这样获得的 TCP 控制块提交给现有技术中已有的、用于实现 TCP/IP 栈的 TCP/IP 栈实现模块（未示出），并由 TCP/IP 栈实现模块来建立 TCP/IP 连接，在连接建立后，TCP/IP 栈实现模块将接口单元附接到已建立的 TCP/IP 连接。随后双方通过该 TCP/IP 连接发送 ACK 分组，在接收到对方发送的 ACK 分组后，双方确认 TCP/IP 连接已经建立完成，并将连接建立成功报告给控制单元 403。虽然在初始建立 TCP/IP 连接时性能可能不够好，但是现有 TCP/IP 栈实现模块能够在通信过程中自适应地调整其自身的设置，诸如放大窗口大小，缩小 RTT 时间等。

[0053] 下面，参照图 4 举例说明两个虚拟机之间初始建立连接的过程。

[0054] 当虚拟机 2 要建立与虚拟机 1 的连接时，VM2 中的控制单元 403-2 为该连接分配一个端口号，PortID2，并在记录表 404 中为该连接建立一条记录，用 VM1 和 VM2 的 IP 地址和 PortID2 更新该条记录，该条记录的其它项使用缺省值，诸如 NULL 或其它值。

[0055] 然后，控制单元 403-2 通过通信代理单元 405-2 将 VM1 和 VM2 的 IP 地址和端口号发送到虚拟化层中的控制逻辑模块 471。

[0056] 控制逻辑模块 471 经由通信代理模块 470 接收到来自 VM2 的 VM1 和 VM2 的 IP 地址后，基于 VM1 的 IP 地址从全局库 480 中检索 VM1 的 VMID1 和 HVID1。

[0057] 如果控制逻辑模块 471 不能从全局库 480 中检索出 VM1 的 HVID1，则默认为 VM1 和 VM2 存在于不同的物理主机上。如果控制逻辑模块 471 从全局库 480 中检索到 VM1 的 VMID1 和 HVID1，则将 VM1 的 HVID1 与其自身的 HVID 比较，而如果二者不同，则认为 VM1 和 VM2 存在于不同的物理主机上。

[0058] 随后，控制逻辑模块 471 将 VM1 和 VM2 的 IP 地址和端口号以及 VM1 和 VM2 处于不同物理主机上的结果通过通信代理模块 470 和通信代理单元 405-2 告知控制单元 403-2。该告知可以基于虚拟化层提供的机制以任意形式来实现。

[0059] 在接收到该告知消息后，控制单元 403-2 基于该告知消息中包括的 VM1 与 VM2 不在同一物理主机上的信息，确定要使用的连接建立机制为 TCP/IP 协议。控制单元 403-2 将 VM2 的 IP 地址和端口号以及 VM1 的 IP 地址和端口号发送给 TCP/IP 栈实现模块并由 TCP/IP 栈实现模块建立 VM1 和 VM2 之间的 TCP/IP 连接。控制单元 403-2 在从 TCP/IP 栈实现模块接收到连接建立成功的确认后，控制数据复用器 406-2 将发送队列和接收队列中的数据连接到已建立的 TCP/IP 连接上。如果控制单元 403-2 从 TCP/IP 栈模块接收到连接建立失败的确认，则基于 VM1 和 VM2 的 IP 地址和端口号从记录表 404-2 中清除用于该连接的记录，并将连接建立失败报告给相应的应用。

[0060] 如果控制逻辑模块 471 从全局库 480 中检索到 VM1 的 VMID1 和 HVID1，则将 VM1 的 HVID1 与其自身的 HVID 比较，如果二者相同，则认为 VM1 和 VM2 存在于同一物理主机上，并将 VM1 和 VM2 的 IP 地址和端口号以及 VM1 和 VM2 处于同一物理主机上的结果连同 VM1 的 VMID1 告知控制单元 403-2。该告知可以基于虚拟化层提供的机制以任意形式来实现。

[0061] 在接收到该告知消息后，控制单元 403-2 基于该告知消息中包括的 VM1 与 VM2 处于同一物理主机上的信息，确定要使用的连接建立机制为轻量级协议。并基于接收到的 VM1 和 VM2 的 IP 地址和端口号以及 VM1 的 ID 将相应记录中的 VM1 的 ID 默认值更新为接收到的消息中包含的 ID 值。

[0062] 然后，控制单元 403-2 将 VM1 和 VM2 的 VMID1 和 VMID2 发送给现有技术中已经实现的、用于建立轻量级连接的轻量级协议引擎模块（未示出）并由轻量级协议引擎模块建立 VM1 和 VM2 之间的轻量级连接，并将连接建立成功报告给控制单元 403-2。控制单元 403-2 在从轻量级协议引擎模块接收到连接建立成功的确认后，控制数据复用器 406-2 将发送队列和接收队列中的数据连接到已建立的轻量级连接。如果控制单元 403-2 从协议引擎模块接收到连接建立失败的确认，则基于接收到的 VM1 和 VM2 的 IP 地址和端口号从记录表 404-2 中清除用于该连接的记录，并将连接建立失败报告给相应的应用。

[0063] 下面，参照图 5 和 6 举例说明在如上所述建立虚拟机之间的轻量级连接后发生虚拟机迁移的情况下切换连接建立机制的过程。

[0064] 首先参照图 5，假设虚拟机 2 与虚拟机 1 处于同一台物理主机，例如物理主机 400 上，且虚拟机 2 与虚拟机 1 之间已经建立了轻量级连接，并且 VM2 要迁移到另一物理主机，例如物理主机 500。在 VM2 实际进行迁移之前，VM2 中的控制单元 403-2 首先命令数据复用器 406-2 锁定其中的所有发送队列和接收队列。然后，VM2 开始迁移。

[0065] 物理主机 400 中的虚拟化层一接收到 VM2 发生迁移的消息，就通知控制逻辑模块 471 并由控制逻辑模块 471 基于 VM2 的 VMID2 从全局库 480 中检索 VM2 的 IP 地址，并基于 VM2 的 VMID2 删除 VM2 在全局库 480 中的全部记录。

[0066] 在 VM2 成功迁移之后，物理主机 500 中的虚拟化层给 VM2 分配新的 VMID，并由其控制逻辑模块 571 再次在全局库 480 中注册 VM2，即，以 (VMID，VMIP，HVID) 的形式在全局库 480 中存储物理主机 500 给虚拟机 VM2 分配的 ID、VM2 的 IP 地址和物理主机 500 的虚拟化层的 ID。

[0067] 物理主机 400 的虚拟化层中的控制逻辑模块 471 提供包括 VM2 的 IP 地址和不在同一物理主机的信息的消息给通信代理模块 470。该消息可以具有任何形式，只要能够实现通知 VMIP2 和不在同一物理主机上的功能即可。通信代理模块 470 基于虚拟化层自身已有的机制（该机制对于本领域技术人员而言是公知的），将上述消息提供给物理主机 1 上除 VM2 之外的其它虚拟机中的控制单元。

[0068] 下面以 VM1 为例，举例说明物理主机 400 上的其它虚拟机接收到上述消息之后执行的动作。在接收到该告知消息后，VM1 中的控制单元 403-1 基于该告知消息中包括的不在同一物理主机上的信息，确定与具有该 IP 地址的虚拟机之间要使用的连接建立机制为 TCP/IP 协议。VM1 中的控制单元 403-1 基于 VM2 的 IP 地址检索其中的记录表 404-1 以查看有没有与该 IP 地址相关的记录。如果没有找到任何记录，则说明 VM1 与 VM2 没有建立任何轻量级协议连接，丢弃该消息并且不做任何动作。如果找到了相应记录，则说明 VM1 与 VM2 建立了轻量级协议连接。有多少条记录就表明有多少个已建立的连接。控制单元 403-1 控制数据复用器 406-1 锁定与这些连接相关的发送队列和接收队列，并基于当前队列的状态更新记录中的各个项，诸如，已接收的分组数、已发送的分组数。

[0069] 在锁定队列之后，在 VM1 和 VM2 中的控制单元 403-1 和 403-2 都控制各自的卸载引

擎单元 402-1 和 402-2 开始基于记录表 404-1 和 404-2 中的相关信息，诸如，双方的 IP 地址和端口号、已接收的分组数、已发送的分组数，使用试探法建立 TCP 控制块，并将已建立的 TCP 控制块发送到 TCP/IP 栈实现模块（图中未示出）。TCP/IP 栈实现模块基于接收的 TCP 控制块建立 TCP/IP 连接之后，将连接建立成功消息返回给控制单元 403-1 和 403-2。

[0070] 收到连接建立成功消息后，VM2 和 VM1 各自的控制单元 403-1 和 403-2 控制数据复用器 406-1 和 406-2 解除发送队列和接收队列的锁定。响应于控制单元 403-1 和 403-2 的控制，数据复用器 406-1 和 406-2 解除对发送队列和接收队列的锁定，并开始通过该 TCP/IP 连接发送数据。

[0071] 参照图 6，假设物理主机 400 上的虚拟机 VM1 与物理主机 500 上的虚拟机 VM2 之间已经建立了 TCP/IP 连接，并且 VM2 要迁移到 VM1 所在的物理主机 400 上。在 VM2 实际进行迁移之前，VM2 中的控制单元 403-2 首先命令数据复用器 406-2 锁定其中的所有发送队列和接收队列。然后，VM2 开始迁移。

[0072] 物理主机 500 中的虚拟化层一接收到 VM2 发生迁移的消息，就通知控制逻辑模块 571 并由控制逻辑模块 571 基于 VM2 的 VMID2 从全局库 480 中检索 VM2 的 IP 地址，并基于 VM2 的 VMID2 删除 VM2 在全局库 480 中的全部记录。

[0073] 然后，VM2 开始迁移。在 VM2 成功地从物理主机 500 迁移到物理主机 400 之后，物理主机 400 给 VM2 分配 VMID2'，随后，VM2 中的的控制单元 403-2 发起注册过程，如上文中所述的那样。物理主机 400 中的虚拟化层中的控制逻辑模块 471 再次在全局库 480 中注册 VM2，即，以 (VMID，IP，HVID) 的形式在全局库 480 中存储物理主机 400 给虚拟机 VM2 分配的 ID、VM2 的 IP 地址和物理主机 400 的虚拟化层的 ID。

[0074] 在给 VM2 分配了 VMID2'之后，物理主机 400 的虚拟化层中的控制逻辑模块 471 提供包括 VM2 的 IP 地址、VMID2'和在同一物理主机内的信息的消息给通信代理模块 470。该消息可以具有任何形式，只要能够实现通知 VMIP2、VMID2' 和在同一物理主机上的功能即可。通信代理模块 470 基于虚拟化层自身已有的机制（该机制对于本领域技术人员而言是公知的），将上述消息提供给除 VM2 之外的其它虚拟机中的控制单元。

[0075] 下面以 VM1 为例，举例说明其它虚拟机接收到上述消息之后执行的动作。

[0076] 在接收到该告知消息后，VM1 中的控制单元 403-1 基于该告知消息中包括的在同一物理主机上的信息，确定与具有该 IP 地址的虚拟机之间要使用的连接建立机制为轻量级协议。VM1 中的控制单元 403-1 基于 VM2 的 IP 地址检索其中的记录表 404-1 以查看有没有与该 IP 相关的记录。如果没有找到任何记录，则说明 VM1 与 VM2 没有建立任何 TCP/IP 协议连接，丢弃该消息并不做任何动作。如果找到了相应记录，则说明 VM1 与 VM2 建立了 TCP/IP 协议连接，并用 VMID2' 更新记录中的 VM2 的 ID。有多少条记录就表明有多少个已建立的 TCP/IP 连接。控制单元 403-1 控制数据复用器 406-1 锁定与这些连接相关的发送队列和接收队列，并基于当前队列的状态更新记录中的各个项，诸如，已接收的分组数、已发送的分组数。

[0077] 然后，VM1 中的控制单元 403-1 将 VM2 的 ID 发送到轻量级协议引擎模块（图中未示出）中，以便建立 VM1 和 VM2 之间的轻量级协议连接。轻量级协议引擎模块在成功建立了轻量级连接之后，将连接建立成功消息返回给控制单元 403。

[0078] 接收到连接建立成功消息之后，VM2 和 VM1 各自的控制单元 403-1 和 403-2 控制

数据复用器 406 解除发送队列和接收队列的锁定。响应于控制单元 403 的控制，数据复用器 406 解除对发送队列和接收队列的锁定，并开始通过轻量级连接发送 / 接收数据。

[0079] 图 7 示出了根据本发明的、在虚拟机与虚拟化层之间交换的消息的形式。但是，本发明不限于此，而是可以采用任何实现相同或等同功能的形式的消息。如果消息中的某一字段没有值，优选地，将其设为 NULL。例如，注册请求消息 (01，VMID1，VMIP1，NULL，NULL，NULL，NULL)；连接建立请求消息是 (01，NULL，VMIP1，PortID1，NULL，VMIP2，PortID2)；响应于连接建立请求返回给虚拟机的消息是：(10，VMID1，VMIP1，PortID1，NULL，VMIP2，PortID2) 或者 (11，NULL，VMIP1，PortID1，NULL，VMIP2，PortID2)；响应于动态迁移返回给虚拟机的消息是 (10，NULL，NULL，NULL，VMID2，VMIP2，NULL) 或者 (11，NULL，NULL，NULL，NULL，VMIP2，NULL)。当然，如图 7 所示，消息中还可以包括其他信息，也可以采用更多的消息类型。

[0080] 图 8 是示出了根据本发明的在虚拟化层中确定虚拟机所处的位置的方法的流程图。

[0081] 根据本发明的方法在步骤 800 开始。

[0082] 在步骤 802，确定与虚拟机所处的位置有关的信息。

[0083] 在步骤 804，向虚拟机发送包括所确定的信息的消息。

[0084] 在步骤 806，本方法结束。

[0085] 图 9 是示出了图 8 所示的步骤 802 的具体步骤的流程图。

[0086] 在步骤 902，判断是否接收到连接建立请求。如果没有，则处理前进到步骤 912，否则，处理前进到步骤 904。

[0087] 在步骤 904，基于连接建立请求中包含的信息，诸如，IP 地址，查询全局库。如果未获得查询结果，例如连接对方虚拟机所在的虚拟化层的 ID，则处理前进到 908，否则，处理前进到步骤 906。

[0088] 在步骤 906，判断查询结果与虚拟化层自身的信息是否匹配。如果匹配，则处理前进到 910，否则，处理前进到步骤 908。

[0089] 在步骤 908，确定虚拟机不在同一物理主机上。然后处理去到图 8 的步骤 804。

[0090] 在步骤 910，确定虚拟机在同一物理主机上。然后处理去到图 8 的步骤 804。

[0091] 在步骤 912，判断是否接收到系统信息。如果没有接收到系统信息，则处理返回到步骤 902，否则，处理前进到步骤 914。

[0092] 在步骤 914，判断接收到系统信息指示虚拟机迁移入本物理主机还是迁移出本物理主机。如果指示迁移入本物理主机，则处理前进到步骤 910，否则，处理前进到步骤 908。

[0093] 以上各步骤的实现细节在上文中对于虚拟化层中的各个部件的描述中已经进行了详述，在此不再赘述。

[0094] 图 10 是示出了根据本发明的根据虚拟机所处的位置动态地确定虚拟机之间的连接建立机制的方法的流程图。

[0095] 根据本发明的方法在步骤 1000 开始。

[0096] 在步骤 1002，接收与虚拟机所处的位置有关的消息。然后，处理前进到步骤 1004。

[0097] 在步骤 1004，基于接收到的消息，确定虚拟机之间的连接建立机制。如果接收到的消息是如果接收到的信息是来自虚拟化层的、指示不在同一物理主机上的消息，则确定连

接建立机制为 TCP/IP 协议；如果接收到的信息是来自虚拟化层的、指示在同一物理主机上的消息，则确定连接建立机制为轻量级协议。然后，处理前进到步骤 1006。

[0098] 在步骤 1006，控制根据所确定的连接建立机制建立虚拟机之间的连接。如果在虚拟机之间已经建立连接的情况下由于虚拟机的迁移而使得所确定的连接建立机制不同于正在使用的连接建立机制，则在切换连接建立机制之前需要锁定数据的发送和接收。在连接建立成功之后，需要对数据的发送和接收解锁。

[0099] 在步骤 1008，根据控制，将发送 / 接收数据附接到建立的连接上。

[0100] 以上各步骤的实现细节在上文中对于虚拟化层中的各个部件的描述中已经进行了详述，在此不再赘述。

[0101] 本发明通过根据虚拟机所处的位置动态地确定连接建立机制，使得本发明能够充分利用轻量级协议和 TCP/IP 协议的优势，避开了其缺陷，从而使得系统的性能始终保持最佳。

[0102] 对本领域的技术人员来说将显而易见的是，可在本发明中作出各种修改，而不会背离本发明的精神和范围。由此，意图使本发明涵盖此发明的修改和变化，只要它们在所附权利要求及其等价物的范围内即可。

\

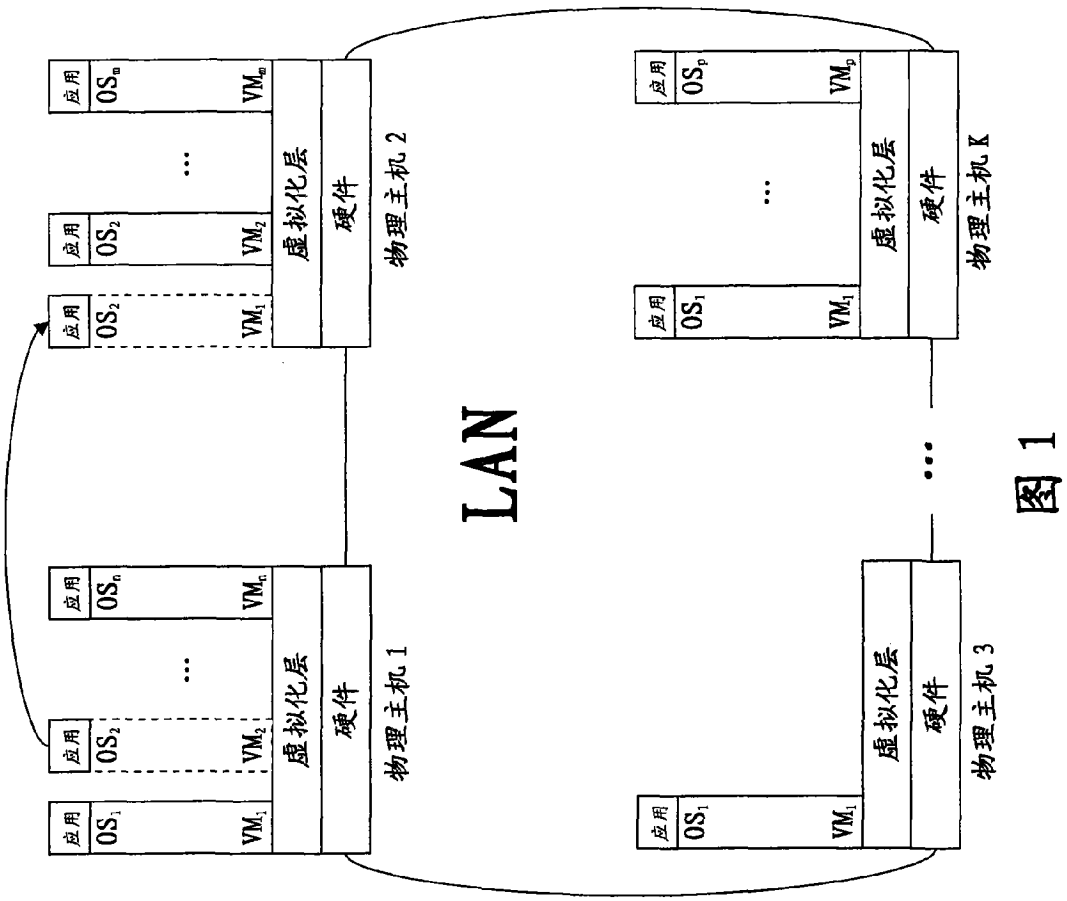

LAN
物理主机 1
物理主机 2
物理主机 3
物理主机 K
虚拟化层
硬件
应用
OS1
OS2
OSn
OSm
OSp
VM1
VM2
VMn
VMm
VMp

图 1

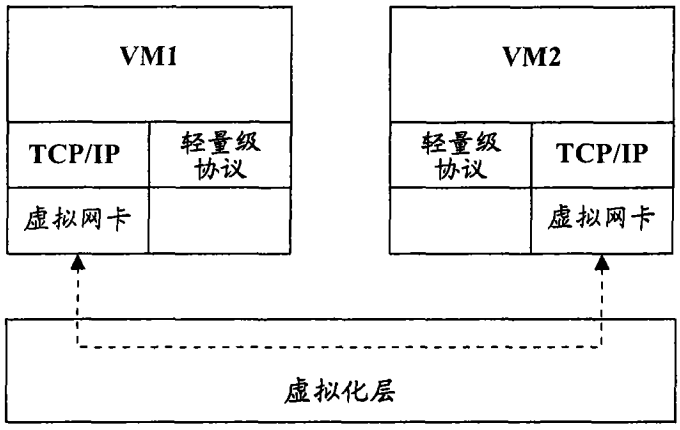

VM1
VM2
TCP/IP
轻量级协议
虚拟网卡
虚拟化层

图 2

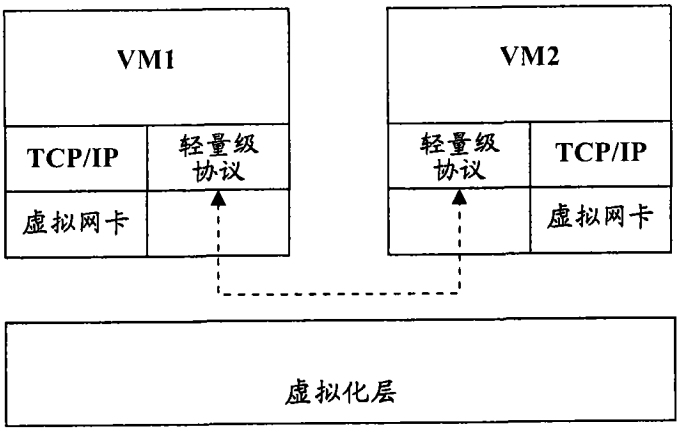


图 3

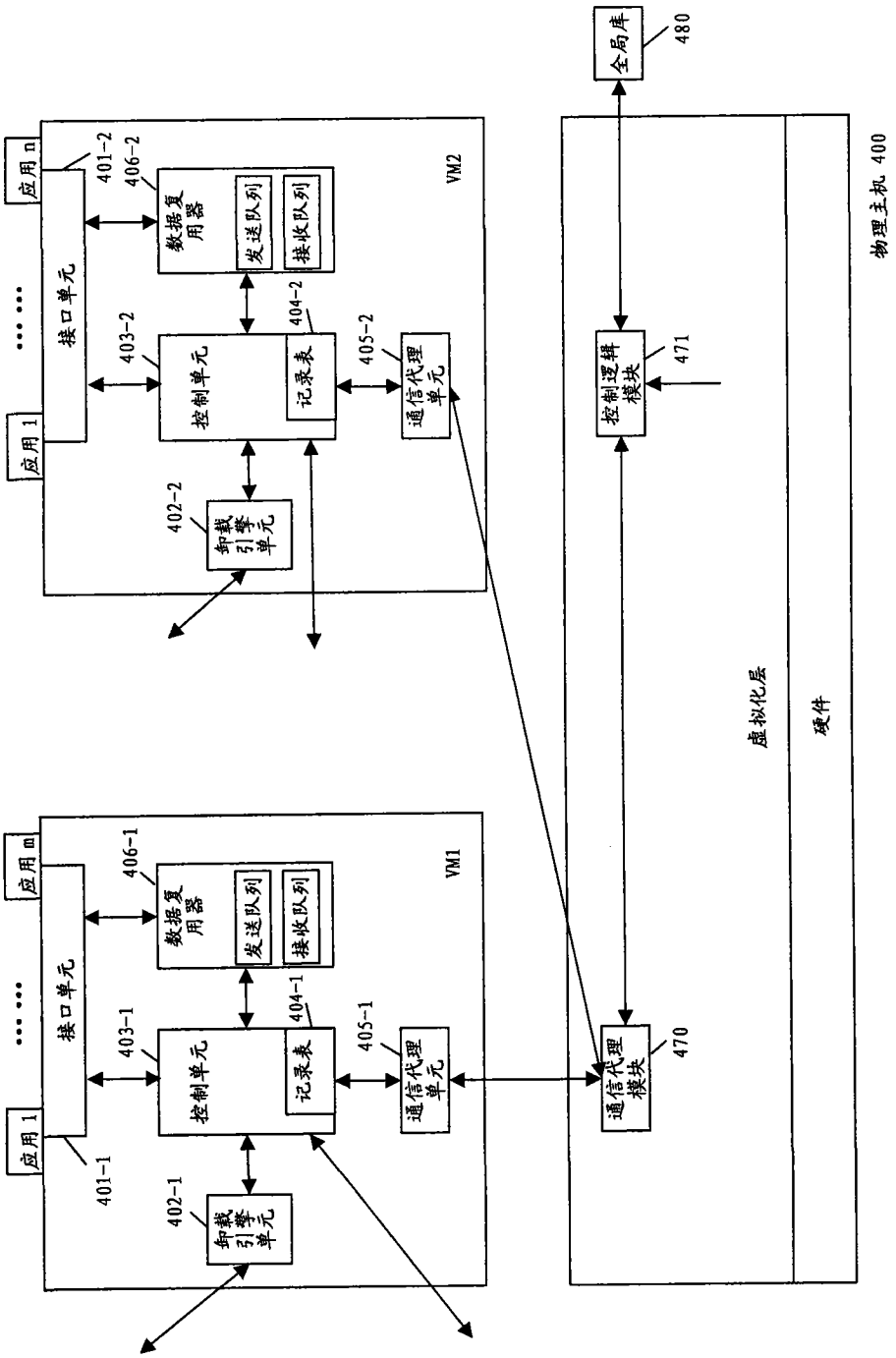

全局库
480
物理主机 400
虚拟化层
硬件
控制逻辑模块
471
通信代理模块
470
VM2
应用 n
应用 1
接口单元
401-2
406-2
数据复用器
发送队列
接收队列
403-2
控制单元
404-2
记录表
405-2
通信代理单元
402-2
卸载引擎单元
VM1
应用 m
应用 1
接口单元
401-1
406-1
数据复用器
发送队列
接收队列
403-1
控制单元
404-1
记录表
405-1
通信代理单元
402-1
卸载引擎单元

图 4

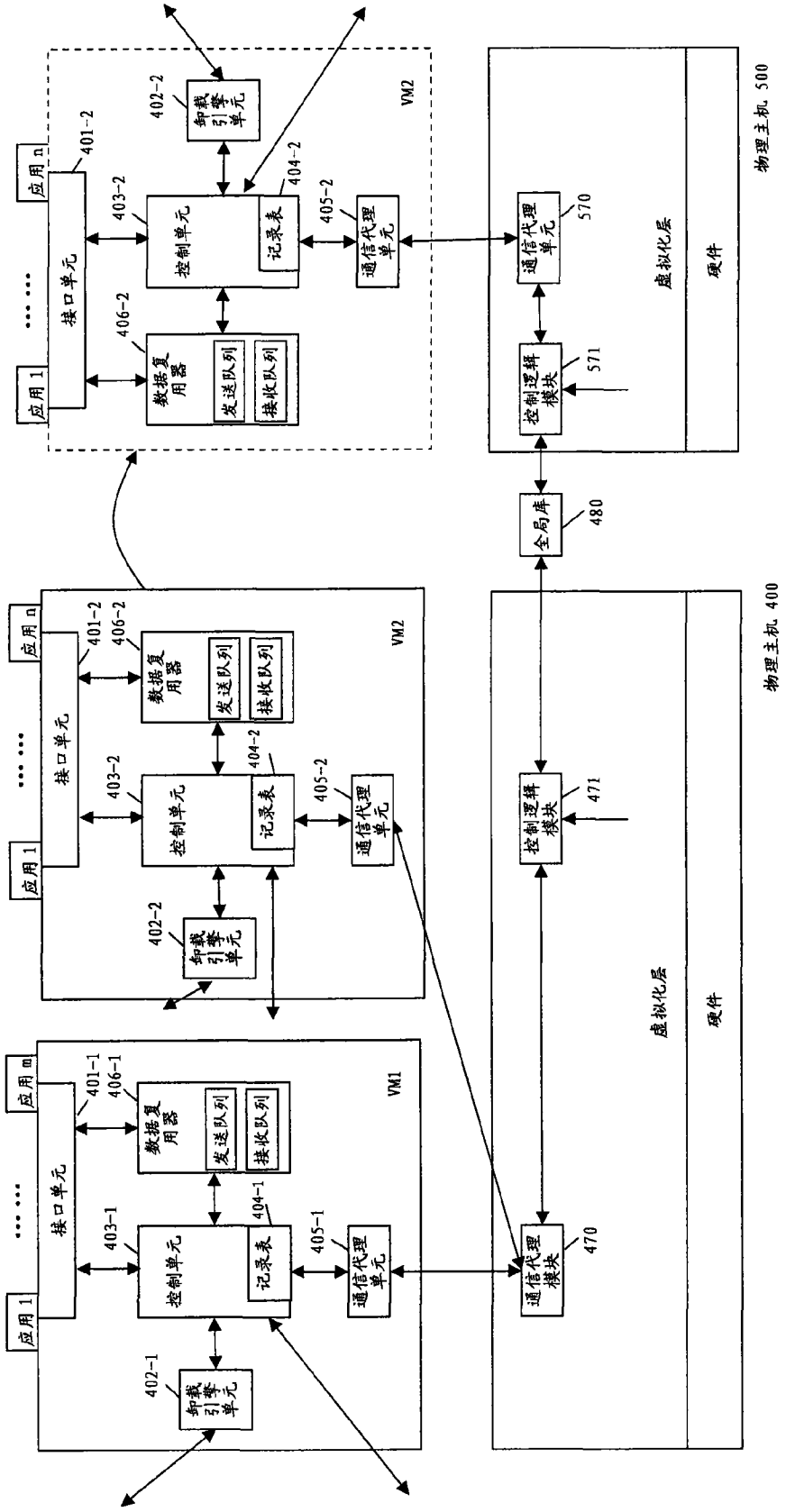

物理主机 500
虚拟化层
硬件
通信代理单元
570
控制逻辑模块
571
全局库
480
物理主机 400
虚拟化层
硬件
控制逻辑模块
471
通信代理模块
470
VM2
应用 n
应用 1
接口单元
401-2
数据复用器
406-2
发送队列
接收队列
控制单元
403-2
记录表
404-2
通信代理单元
405-2
卸载引擎单元
402-2
VM2
应用 n
应用 1
接口单元
401-2
数据复用器
406-2
发送队列
接收队列
控制单元
403-2
记录表
404-2
通信代理单元
405-2
卸载引擎单元
402-2
VM1
应用 m
应用 1
接口单元
401-1
数据复用器
406-1
发送队列
接收队列
控制单元
403-1
记录表
404-1
通信代理单元
405-1
卸载引擎单元
402-1

图 5

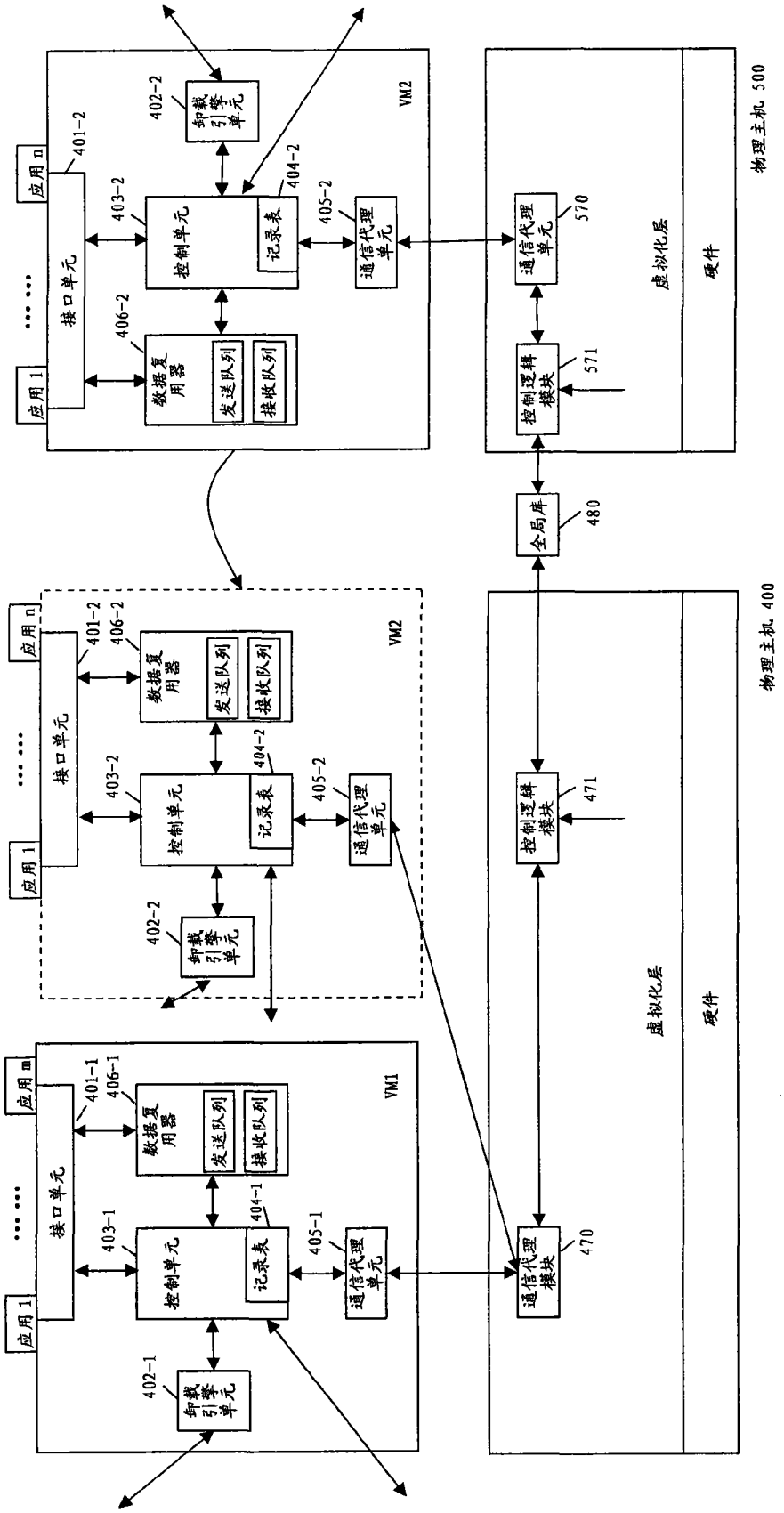

物理主机 500
物理主机 400
虚拟化层
硬件
VM1
VM2
应用 m
应用 n
应用 1
接口单元
控制单元
记录表
数据复用器
发送队列
接收队列
卸载引擎单元
通信代理单元
通信代理模块
控制逻辑模块
全局库
401-1
401-2
402-1
402-2
403-1
403-2
404-1
404-2
405-1
405-2
406-1
406-2
470
471
480
570
571

图 6

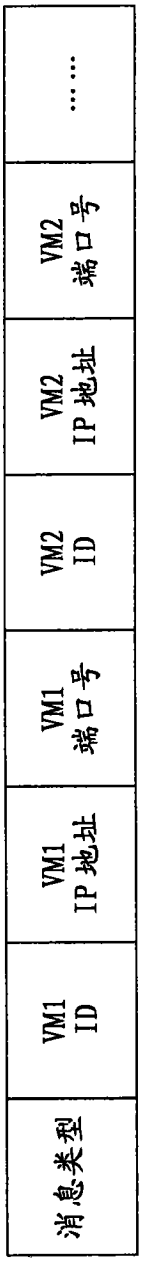

消息类型
VM1 ID
VM1 IP 地址
VM1 端口号
VM2 ID
VM2 IP 地址
VM2 端口号
……
消息类型：0000－注册请求
0001-连接建立请求
从 VM 至虚拟化层
0010-在同一物理主机上
0011－不在同一物理主机上
从虚拟化层至 VM
……

图 7

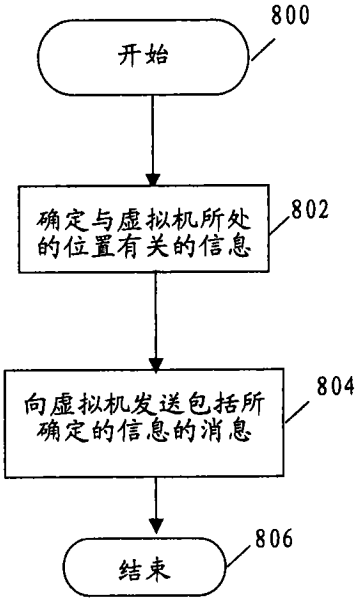


图 8

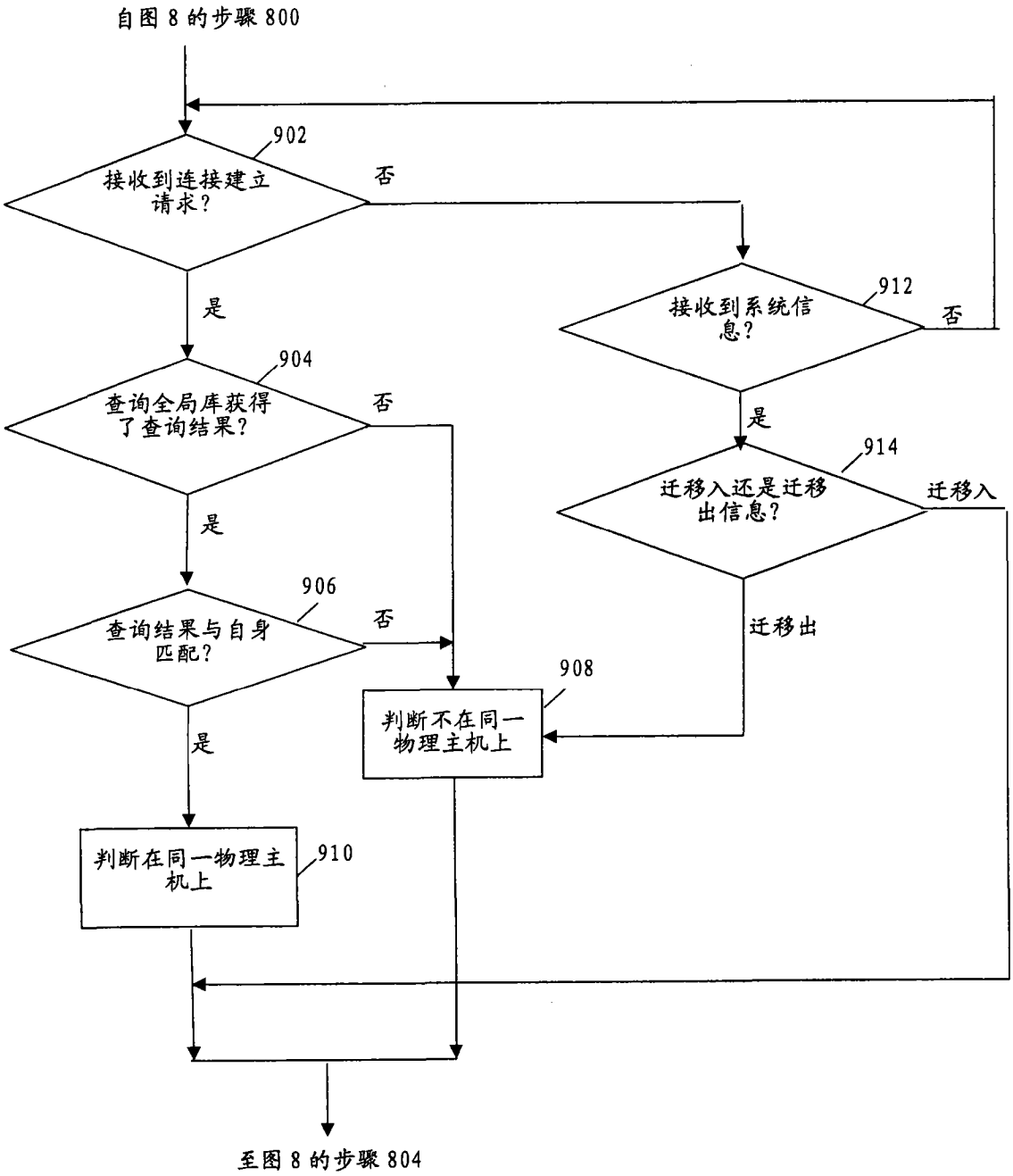

自图 8 的步骤 800
902
接收到连接建立请求?
否
是
904
查询全局库获得了查询结果?
否
是
906
查询结果与自身匹配?
否
是
908
判断不在同一物理主机上
910
判断在同一物理主机上
912
接收到系统信息?
否
是
914
迁移入还是迁移出信息?
迁移入
迁移出
至图 8 的步骤 804

图 9

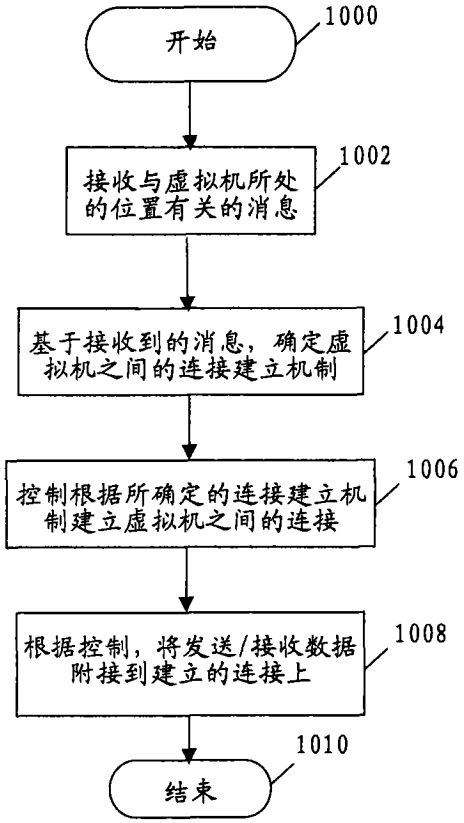

开始
1000
接收与虚拟机所处的位置有关的消息
1002
基于接收到的消息，确定虚拟机之间的连接建立机制
1004
控制根据所确定的连接建立机制建立虚拟机之间的连接
1006
根据控制，将发送/接收数据附接到建立的连接上
1008
结束
1010

图 10