(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 107851034 B
(45)授权公告日 2021.08.06

(21)申请号 201780002466.7
(22)申请日 2017.01.26
(65)同一申请的已公布的文献号
申请公布号 CN 107851034 A
(43)申请公布日 2018.03.27
(30)优先权数据
62/287,712 2016.01.27 US
15/415,668 2017.01.25 US
(85)PCT国际申请进入国家阶段日
2018.01.19
(86)PCT国际申请的申请数据
PCT/US2017/015169 2017.01.26
(87)PCT国际申请的公布数据
WO2017/132399 EN 2017.08.03
(73)专利权人 甲骨文国际公司
地址 美国加利福尼亚
(72)发明人 B·D·约翰森 H·霍格
L·霍雷恩
(74)专利代理机构 中国贸促会专利商标事务所有限公司 11038
代理人 李晓芳
(51)Int.Cl.
G06F 9/455(2006.01)
H04L 12/931(2006.01)
H04L 29/12(2006.01)
(56)对比文件
CN 104094230 A,2014.10.08
CN 102483718 A,2012.05.30
CN 104094231 A,2014.10.08
US 8824279 B2,2014.09.02
叶庆,刘森,张严辞..基于Infiniband网络的消息传输技术研究.《四川大学学报》.2015,第52卷(第2期),276-280.
Jiuxing Liu,Amith R. Mamidala,Abhinav Vishnu,Dhabaleswar K. Pand.Evaluating InfiniBand Performance with PCI Express.《IEEE Micro》.2005,第25卷(第1期),20-29.
审查员 李一

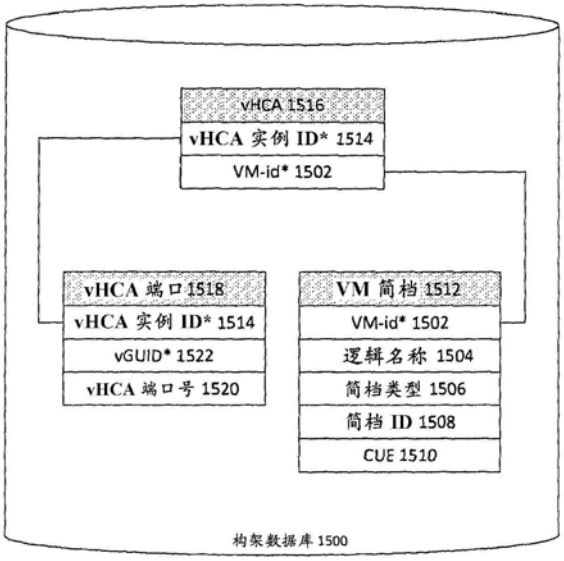
权利要求书3页 说明书24页 附图16页

(54)发明名称

用于定义虚拟机的虚拟机构架简档的系统和方法

(57)摘要

用于定义虚拟机的虚拟机构架简档的系统和方法,虚拟机构架简档可以包含虚拟机标识符、虚拟主机通道适配器实例ID以及虚拟全局唯一标识符。虚拟机标识符、虚拟主机通道适配器实例ID以及虚拟全局唯一标识符可以被映射到彼此,以使得可以通过访问虚拟机标识符来检索虚拟主机通道适配器实例ID和虚拟全局唯一标识符。此外,可以在定义管理分区的P_Key与虚拟全局唯一标识符之间创建关系,其中P_Key与虚拟全局唯一标识符之间的关系将虚拟全局唯一标识符定义为由P_key定义的管理分区的成员。



CN 107851034 B

1.一种为虚拟机VM创建虚拟机构架简档的方法,包括以下步骤:

生成虚拟机标识符VM-id;

生成识别虚拟主机通道适配器vHCA的虚拟主机通道适配器实例ID;

从全局唯一标识符的池中指派虚拟全局唯一标识符vGUID作为vHCA的虚拟端口的当前构架地址;

创建定义管理分区的P_Key与vGUID之间的第一关系,其中P_Key与vGUID之间的关系将vGUID作为当前构架地址被指派给的所述虚拟端口定义为由P_key定义的所述管理分区的成员;

创建VM-id与vHCA实例ID之间的第二关系,其中第二关系允许通过访问VM-id来检索vHCA实例ID;

创建VM-id与vGUID之间的第三关系,其中第三关系允许通过访问VM-id来检索vGUID;以及

以保留VM-id、虚拟主机通道适配器、vGUID、第一关系、第二关系和第三关系的格式来持久化VM-id、虚拟主机通道适配器实例ID、vGUID、第一关系、第二关系和第三关系。

2.如权利要求1所述的方法,还包括以下步骤:

将VM-id、vHCA实例ID、vGUID、第一关系、第二关系、第三关系和P_Key放置在子网的子网管理器的高速缓存中,其中由P_Key定义的所述管理分区在所述子网内被定义。

3.如权利要求2所述的方法,其中被放置在所述高速缓存中的VM-id、虚拟主机通道适配器实例ID、vGUID、第一关系、第二关系、第三关系和P_Key是副本。

4.如权利要求2或3所述的方法,其中所述高速缓存是易失性存储器。

5.如权利要求1至3中任一项所述的方法,还包括以下步骤:

使用VM-id作为查找关键字来访问vHCA实例ID、vGUID和P_Key。

6.如权利要求5所述的方法,其中使用VM-id作为查找关键字是由尝试向物理主机通道适配器的虚拟功能注册vHCA实例ID的物理主机通道适配器发起的。

7.如权利要求2或3所述的方法,还包括以下步骤:

由所述子网管理器从所述高速缓存中检索vHCA实例ID和vGUID;以及

将vHCA实例ID和vGUID发送到物理主机通道适配器。

8.一种创建和访问用于虚拟机VM的虚拟机构架简档的系统,包括:

节点,所述节点包括处理器;

所述节点能够访问的存储器;以及

其中所述节点操作以:

生成虚拟机标识符VM-id;

生成识别虚拟主机通道适配器vHCA的虚拟主机通道适配器实例ID;

从全局唯一标识符的池中指派虚拟全局唯一标识符vGUID作为vHCA的虚拟端口的当前构架地址;

创建定义管理分区的P_Key与vGUID之间的第一关系,其中P_Key与vGUID之间的关系将vGUID作为当前构架地址被指派给的所述虚拟端口定义为由P_key定义的所述管理分区的成员;

创建VM-id与vHCA实例ID之间的第二关系,其中第二关系允许通过访问VM-id来检索

vHCA实例ID；

创建VM-id与vGUID之间的第三关系，其中第三关系允许通过访问VM-id来检索vGUID；以及

以保留VM-id、虚拟主机通道适配器、vGUID、第一关系、第二关系和第三关系的格式将VM-id、虚拟主机通道适配器实例ID、vGUID、第一关系、第二关系和第三关系持久化到所述存储器。

9.如权利要求8所述的系统，还包括子网的子网管理器，其中VM-id、vHCA实例ID、vGUID、第一关系、第二关系、第三关系和P_Key被放置在所述子网管理器的高速缓存中，并且其中由P_Key定义的所述管理分区在所述子网内定义。

10.如权利要求9所述的系统，其中被放置在所述高速缓存中的VM-id、虚拟主机通道适配器实例ID、vGUID、第一关系、第二关系、第三关系和P_Key是副本。

11.如权利要求9或10所述的系统，其中所述高速缓存是易失性存储器。

12.如权利要求8至10中任一项所述的系统，其中VM-id被用作查找关键字来访问vHCA实例ID、vGUID和P_Key。

13.如权利要求12所述的系统，其中物理主机通道适配器在尝试向物理主机通道适配器的虚拟功能注册vHCA实例ID时发起查找。

14.如权利要求9或10所述的系统，其中所述子网管理器从所述高速缓存中检索vHCA实例ID和vGUID；以及

其中所述子网管理器将vHCA实例ID和vGUID发送到物理主机通道适配器。

15.一种包括存储在其上的用于为虚拟机VM创建虚拟机构架简档的指令的非暂态计算机可读存储介质，所述指令当由一个或多个计算机读取和执行时，使所述一个或多个计算机执行步骤，所述步骤包括：

生成虚拟机标识符VM-id；

生成识别虚拟主机通道适配器vHCA的虚拟主机通道适配器实例ID；

从全局唯一标识符的池中指派虚拟全局唯一标识符vGUID作为vHCA的虚拟端口的当前构架地址；

创建定义管理分区的P_Key与vGUID之间的第一关系，其中P_Key与vGUID之间的关系将vGUID作为当前构架地址被指派给的虚拟端口定义为由P_key定义的所述管理分区的成员；

创建VM-id与vHCA实例ID之间的第二关系，其中第二关系允许通过访问VM-id来检索vHCA实例ID；

创建VM-id与vGUID之间的第三关系，其中第三关系允许通过访问VM-id来检索vGUID；以及

以保留VM-id、虚拟主机通道适配器、vGUID、第一关系、第二关系和第三关系的格式来持久化VM-id、虚拟主机通道适配器实例ID、vGUID、第一关系、第二关系和第三关系。

16.如权利要求15所述的非暂态计算机可读存储介质，所述步骤还包括：

将VM-id、vHCA实例ID、vGUID、第一关系、第二关系、第三关系和P_Key放置在子网的子网管理器的高速缓存中，其中由P_Key定义的所述管理分区在所述子网内定义。

17.如权利要求16所述的非暂态计算机可读存储介质，其中被放置在所述高速缓存中的VM-id、虚拟主机通道适配器实例ID、vGUID、第一关系、第二关系、第三关系和P_Key是副

本。

18.如权利要求16或17所述的非暂态计算机可读存储介质,其中所述高速缓存是易失性存储器。

19.如权利要求15至17中任一项所述的非暂态计算机可读存储介质,所述步骤还包括:

使用VM-id作为查找关键字来访问vHCA实例ID、vGUID和P_Key,其中使用VM-id作为查找关键字是由尝试向物理主机通道适配器的虚拟功能注册vHCA实例ID的物理主机通道适配器发起的。

20.如权利要求16或17所述的非暂态计算机可读存储介质,所述步骤还包括:

由所述子网管理器从所述高速缓存中检索vHCA实例ID和vGUID;以及

将vHCA实例ID和vGUID发送到物理主机通道适配器。

21.一种包括用于执行根据权利要求1至7中任一项所述的方法的部件的装置。

用于定义虚拟机的虚拟机构架简档的系统和方法

技术领域

[0003] 本发明一般而言涉及计算机系统，并且具体地涉及定义虚拟机的虚拟机构架简档。

背景技术

[0004] 随着更大的云计算架构被引入，与传统网络和存储相关联的性能和管理瓶颈成为重要的问题。对于使用诸如InfiniBand™(IB)技术之类的高性能无损互连来作为用于云计算构架的基础的兴趣增加。这是本发明的实施例旨在解决的一般领域。

发明内容

[0005] 本文描述的是用于定义虚拟机的虚拟机构架简档的系统和方法。示例性实施例可以提供虚拟机标识符、虚拟主机通道适配器实例ID以及虚拟全局唯一标识符。虚拟机标识符、虚拟主机通道适配器实例ID以及虚拟全局唯一标识符可以被映射到彼此，以使得可以通过访问虚拟机标识符来检索虚拟主机通道适配器实例ID和虚拟全局唯一标识符。此外，可以在定义管理分区的P_Key(P键)与虚拟全局唯一标识符之间创建关系，其中P_Key与虚拟全局唯一标识符之间的关系将虚拟全局唯一标识符定义为由P_key定义的管理分区的成员。

附图说明

[0006] 图1示出了根据实施例的InfiniBand™环境的图示。

[0007] 图2示出了根据实施例的分区的集群环境的图示。

[0008] 图3示出了根据实施例的网络环境中的树形拓扑的图示。

[0009] 图4示出了根据实施例的示例性共享端口架构。

[0010] 图5示出了根据实施例的示例性vSwitch架构。

[0011] 图6示出了根据实施例的示例性vPort架构。

[0012] 图7示出了根据实施例的具有预填充的LID的示例性vSwitch架构。

[0013] 图8示出了根据实施例的具有动态LID指派的示例性vSwitch架构。

[0014] 图9示出了根据实施例的具有具有动态LID指派和预填充的LID的vSwitch的示例性vSwitch架构。

[0015] 图10示出了根据实施例的示例性多子网InfiniBand™构架。

[0016] 图11示出了根据实施例的、包括示例性物理子网资源和逻辑子网资源的示例性

InfiniBand™构架和子网。

[0017] 图12是根据实施例的、包括作为不同管理分区的成员的示例性子网资源的示例性InfiniBand™构架和子网。

[0018] 图13示出了根据实施例的、包括作为分层的管理方案(既包括管理分区又包括资源域)的成员的示例性子网资源的示例性InfiniBand™构架和子网。

[0019] 图14是根据实施例的、用于将相互访问权限指派给与构架级资源域相关联的管理分区的成员和相关联的资源的方法的构架图。

[0020] 图15示出了根据实施例的、用于存储VM构架简档信息的示例性数据库结构。

[0021] 图16是根据实施例的、用于使VM构架简档对于子网资源可用的流程图。

[0022] 图17是根据实施例的、用于为虚拟机创建虚拟机构架简档的流程图。

具体实施方式

[0023] 通过示例而非限制的方式在附图的各图中示出了本发明,在附图中类似的附图标记指示类似的元件。应当注意的是,在本公开中对“一”、“一个”或“一些”实施例的引用不一定是指相同的实施例,并且这种引用意味着至少一个。虽然讨论了特定实施方式,但是应当理解的是,特定实施方式仅仅是为了说明性目的而提供。相关领域的技术人员将认识到,在不脱离本发明的范围和精神的情况下,可以使用其它组件和配置。

[0024] 贯穿附图和具体实施方式可以使用共同的附图标记来指示相同的元素;因此,如果元素在其它地方进行了描述,那么在图中使用的附图标记可以或可以不在特定于该图的详细描述中被引用。

[0025] 本文描述的是用于定义虚拟机的虚拟机构架简档的系统和方法。

[0026] 本发明的以下描述使用InfiniBand™(IB)网络作为高性能网络的示例。贯穿以下描述,可以参考InfiniBand™规范(也被不同地称为InfiniBand规范、IB规范或遗留IB规范)。这种参考被理解为指的是在http://www.inifinibandta.org上可获得的2015年3月发布的**InfiniBand®**行业协会架构规范,卷1,版本1.3,其全部内容通过引用并入本文。对于本领域技术人员来说将明显的是,可以使用其它类型的高性能网络而没有限制。以下描述还使用胖树拓扑作为构架拓扑的示例。对于本领域技术人员来说将明显的是,可以使用其它类型的构架拓扑而没有限制。

[0027] InfiniBand™

[0028] InfiniBand™(IB)是由InfiniBand™贸易协会开发的开放标准无损网络技术。该技术基于提供高吞吐量和低延迟通信的串行点对点全双工互连,尤其适合高性能计算(HPC)应用和数据中心。

[0029] InfiniBand™架构(IBA)支持双层拓扑划分。在下层,IB网络被称为子网,其中子网可以包括使用交换机和点对点链路互连的主机集合。在较高级别,IB构架构成可以使用路由器互连的一个或多个子网。

[0030] 在子网内,可以使用交换机和点对点链路来连接主机。此外,可以存在主管理实体,子网管理器(SM),SM驻留在子网中的指定设备上。子网管理器负责配置、激活和维护IB子网。另外,子网管理器(SM)可以负责执行IB构架中的路由表计算。这里,例如,IB网络的路由目的在于在本地子网中的所有源和目的地对之间进行适当的负载平衡。

[0031] 通过子网管理接口，子网管理器与子网管理代理(SMA)交换被称为子网管理分组(SMP)的控制分组。子网管理代理驻留在每个IB子网设备上。通过使用SMP，子网管理器能够发现构架、配置端节点和交换机以及从SMA接收通知。

[0032] 根据实施例，IB网络中的子网内路由可以基于存储在交换机中的LFT。LFT由SM根据使用中的路由机制来计算。端节点和交换机上的主机通道适配器(HCA)端口使用本地标识符(LID)而被寻址。LFT中的每个条目包括目的地LID(DLID)和输出端口。表中仅支持每LID一个条目。当分组到达交换机时，通过在交换机的转发表中查找DLID来确定它的输出端口。路由是确定性的，因为分组在给定的源-目的地对(LID对)之间采用网络中的相同的路径。

[0033] 一般而言，除了主子网管理器之外的所有其它子网管理器为了容错都在备用模式下工作。但是，在主子网管理器发生故障的情况下，由备用子网管理器协商新的主子网管理器。主子网管理器还执行子网的周期性扫描，以检测任何拓扑变化并且相应地重新配置网络。

[0034] 此外，可以使用本地标识符(LID)来对子网内的主机和交换机进行寻址，并且单个子网可以被限制为49151个单播LID。除了作为在子网内有效的本地地址的LID之外，每个IB设备还可以具有64比特全局唯一标识符(GUID)。GUID可以用于形成作为IB三层(L3)地址的全局标识符(GID)。

[0035] 在网络初始化时，SM可以计算路由表(即，子网内每对节点之间的连接/路由)。此外，每当拓扑改变时，都可以更新路由表，以便确保连接性和最佳性能。在正常操作期间，SM可以执行网络的周期性轻扫以检查拓扑变化。如果在轻扫期间发现变化，或者如果SM接收到标志网络变化的消息(陷阱)，那么SM可以根据发现的变化来重新配置网络。

[0036] 例如，当网络拓扑改变时，诸如当链路失效时、当添加设备时或者当链路被去除时，SM可以重新配置网络。重新配置步骤可以包括在网络初始化期间执行的步骤。此外，重新配置可以具有限于发生网络变化的子网的局部范围。此外，用路由器对大型构架进行分段可以限制重新配置的范围。

[0037] 图1中示出了示例InfiniBand™构架，其示出了根据实施例的InfiniBand™环境100的图示。在图1所示的示例中，节点A-E 101-105使用InfiniBand™构架120经由相应的主机通道适配器111-115进行通信。根据实施例，各种节点(例如，节点A-E 101-105)可以由各种物理设备来表示。根据实施例，各种节点(例如，节点A-E101-105)可以由各种虚拟设备(诸如虚拟机)来表示。

[0038] InfiniBand™中的数据分区

[0039] 根据实施例，IB网络可以支持分区作为用于提供共享网络构架的系统的逻辑组的隔离的安全机制。构架中的节点上的每个HCA端口可以是一个或多个分区的成员。根据实施例，本公开提供了可以在IB子网内定义的两种类型的分区-数据分区(在下面的段落中详细讨论)和管理分区(在本公开中稍后详细讨论)。

[0040] 数据分区成员资格由集中式分区管理器管理，集中式分区管理器可以是SM的一部分。SM可以将每个端口上的数据分区成员资格信息配置为16比特分区键(P_Key)的表。SM还可以用数据分区实施表来配置交换机和路由器端口，该数据分区实施表包含与通过这些端口发送或接收数据业务(traffic)的端节点相关联的P_Key信息。此外，在一般情况下，交换

机端口的数据分区成员资格可以表示与在出口(朝链路)方向经由端口路由的LID间接相关联的所有成员资格的联合。

[0041] 根据实施例,数据分区是端口的逻辑组,以使得组的成员只能与同一逻辑组的其它成员通信。在主机通道适配器(HCA)和交换机处,可以使用数据分区成员资格信息来过滤分组,以实施隔离。一旦具有无效分区信息的分组到达传入端口,就可以丢弃这些分组。在分区的IB系统中,可以使用数据分区来创建租户集群。在数据分区实施就位的情况下,节点不能与属于不同租户集群的其它节点通信。以这种方式,即使在存在受损或恶意租户节点的情况下,系统的安全性也能得到保证。

[0042] 根据实施例,对于节点之间的通信,除了管理队列对(QP0和QP1)以外,队列对(QP)和端到端上下文(EEC)可以被指派给特定数据分区。然后,可以将P_Key信息添加到发送的每个IB传输分组。当分组到达HCA端口或交换机时,可以针对由SM配置的表来验证分组的P_Key值。如果找到无效的P_Key值,那么立即丢弃该分组。以这种方式,仅允许在共享数据分区的端口之间的通信。

[0043] 图2中示出了IB数据分区的示例,其示出了根据实施例的经数据分区的集群环境的图示。在图2所示的示例中,节点A-E 101-105使用InfiniBand™构架120经由相应的主机通道适配器111-115进行通信。节点A-E布置到数据分区(即,数据分区1 130、数据分区2 140和数据分区3 150)中。数据分区1包括节点A 101和节点D 104。数据分区2包括节点A 101、节点B 102和节点C 103。数据分区3包括节点C 103和节点E 105。由于数据分区的布置,节点D 104和节点E 105不允许进行通信,因为这些节点不共享数据分区。同时,例如,节点A 101和节点C 103允许进行通信,由于这些节点都是数据分区2 140的成员。

[0044] InfiniBand™中的虚拟机

[0045] 在过去的十年中,因为通过硬件虚拟化支持已几乎消除了CPU开销,所以虚拟化高性能计算(HPC)环境的前景已得到相当大的提高;通过使存储器管理单元虚拟化显著降低了存储器开销;通过使用快速SAN存储装置或分布式网络文件系统减少了存储装置开销;并且通过使用设备直通技术(比如单根输入/输出虚拟化(SR-IOV))减少了网络I/O开销。现在,云有可能使用高性能互连解决方案来容纳虚拟HPC(vHPC)集群,以及交付必要的性能。

[0046] 但是,当与诸如InfiniBand™(IB)之类的无损网络耦合时,由于在这些解决方案中使用的复杂寻址和路由方案,某些云功能(诸如虚拟机(VM)的实时迁移)仍然是个问题。IB是提供高带宽和低延迟的互连网络技术,因此非常适合于HPC和其它通信密集型工作负载。

[0047] 用于将IB设备连接到VM的传统方法是通过利用具有直接指派的SR-IOV。但是,使用SR-IOV实现被指派有IB主机通道适配器(HCA)的VM的实时迁移已被证明是有挑战性的。每个IB连接的节点具有三个不同的地址:LID、GUID和GID。当发生实时迁移时,这些地址中的一个或多个改变。与迁移中的VM通信的其它节点可能丢失连接。当发生这种情况时,通过向IB子网管理器(SM)发送子网监管(SA)路径记录查询来定位要重新连接到的虚拟机的新地址,可以尝试重新开始丢失的连接。

[0048] IB使用三种不同类型的地址。第一种类型的地址是16比特本地标识符(LID)。SM向每个HCA端口和每个交换机指派至少一个唯一的LID。LID用于在子网内路由业务。由于LID为16比特长,因此可以做出65536个唯一的地址组合,其中只有49151个(0x0001-0xBFFF)可以用作单播地址。因此,可用单播地址的数量限定了IB子网的最大尺寸。第二种类型的地址

是由制造商指派给每个设备(例如,HCA和交换机)和每个HCA端口的64比特全局唯一标识符(GUID)。SM可以向HCA端口指派附加的子网唯一GUID,它在使用SR-IOV时是有用的。第三种类型的地址是128比特全局标识符(GID)。GID是有效的IPv6单播地址,并且向每个HCA端口指派至少一个GID。通过组合由构架管理器指派的全局唯一64比特前缀和每个HCA端口的GUID地址而形成GID。

[0049] 胖树(FTree)拓扑和路由

[0050] 根据实施例,基于IB的HPC系统中的一些采用胖树拓扑以利用胖树提供的有用属性。由于每个源目的地对之间有多条路径可用,因此这些属性包括完全的二分带宽和固有的容错。胖树背后的最初想法是,随着树朝着拓扑的根移动,在节点之间采用具有更多可用带宽的较胖链路。较胖链路可以帮助避免上层交换机中的拥塞并且维持二分带宽。

[0051] 图3示出了根据实施例的网络环境中的树形拓扑的图示。如图3所示,一个或多个端节点201-204可以在网络构架200中被连接。网络构架200可以基于包括多个叶子交换机211-214和多个主干交换机或根交换机231-234的胖树拓扑。此外,网络构架200可以包括一个或多个中间交换机,诸如交换机221-224。

[0052] 同样如图3所示,端节点201-204中的每一个可以是多宿主节点,即,通过多个端口连接到网络构架200的两个或更多个部分的单个节点。例如,节点201可以包括端口H1和H2,节点202可以包括端口H3和H4,节点203可以包括端口H5和H6,并且节点204可以包括端口H7和H8。

[0053] 此外,每个交换机可以具有多个交换机端口。例如,根交换机231可以具有交换机端口1-2,根交换机232可以具有交换机端口3-4,根交换机233可以具有交换机端口5-6,并且根交换机234可以具有交换机端口7-8。

[0054] 根据实施例,胖树路由机制是用于基于IB的胖树拓扑的最流行的路由算法之一。胖树路由机制还在OFED(开放构架企业分发-用于构建和部署基于IB的应用的标准软件栈)子网管理器OpenSM中实现。

[0055] 胖树路由机制旨在生成在网络构架中跨链路均匀散布最短路径路由的LFT。该机制按索引次序遍历构架并将端节点的目标LID指派给每个交换机端口,并且因此将对应的路由指派给每个交换机端口。对于连接到相同叶子交换机的端节点,索引次序可以取决于端节点连接到的交换机端口(即端口编号顺序)。对于每个端口,该机制可以维护端口使用计数器,并且可以在每次添加新路由时使用这个端口使用计数器来选择最少使用的端口。

[0056] 根据实施例,在被分区的子网中,不允许不是公共数据分区的成员的节点进行通信。在实践中,这意味着由胖树路由算法指派的路由中的一些路由不用于用户业务。当胖树路由机制以与它针对其它功能路径所做的相同的方式为这些路由生成LFT时,会出现问题。由于节点按索引的次序进行路由,因此这种行为可能导致链路上的平衡降级。由于路由可以在不考虑数据分区的情况下执行,因此,一般而言,胖树路由的子网提供数据分区之间的较差隔离。

[0057] 根据实施例,胖树是可以利用可用网络资源进行缩放的分层网络拓扑。此外,使用放置在层次的不同级别上的商用交换机容易构建胖树。胖树的不同变体通常可用,包括k-ary-n-trees(k元n树)、扩展的广义胖树(XGFT)、并行端口广义胖树(PGFT)和现实生活胖树(RLFT)。

[0058] k-ary-n-tree是具有各自具有2k个端口的k^n个端节点和$n \cdot k^{n-1}$个交换机的n级胖树。每个交换机在树中具有相同数量的向上连接和向下连接。XGFT胖树通过允许交换机的向上连接和向下连接数量不同以及在树中每个级别的连接的数量不同来扩展k-ary-n-tree。PGFT定义进一步拓宽了XGFT拓扑，并且允许交换机之间的多个连接。可以使用XGFT和PGFT来定义各种各样的拓扑。但是，为了实践的目的，引入了作为PGFT的受限版本的RLFT来定义当今HPC集群中常见的胖树。RLFT在胖树的所有级别使用相同端口计数的交换机。

[0059] 输入/输出(I/O)虚拟化

[0060] 根据实施例，I/O虚拟化(IOV)可以通过允许虚拟机(VM)访问底层物理资源来提供I/O的可用性。存储业务和服务器间通信的组合带来了可能压垮单个服务器的I/O资源的增加的负载，从而导致积压和空闲处理器，这是由于它们在等待数据。随着I/O请求的数量增加，IOV可以提供可用性；并且可以提高(虚拟化的)I/O资源的性能、可伸缩性和灵活性，以匹配在现代CPU虚拟化中所看到的性能水平。

[0061] 根据实施例，IOV是期望的，因为它可以允许共享I/O资源并且提供VM对资源的受保护的访问。IOV将暴露给VM的逻辑设备与它的物理实现解耦。当前，可以存在不同类型的IOV技术，诸如仿真、半虚拟化、直接指派(DA)和单根I/O虚拟化(SR-IOV)。

[0062] 根据实施例，一种类型的IOV技术是软件仿真。软件仿真可以允许解耦的前端/后端软件架构。前端可以是置于VM中、与由管理程序实现的后端通信的设备驱动程序，以提供I/O访问。物理设备共享比高，并且VM的实时迁移可能只需几毫秒的网络停机时间。但是，软件仿真引入了附加的、不期望的计算开销。

[0063] 根据实施例，另一种类型的IOV技术是直接设备指派。直接设备指派涉及将I/O设备耦合到VM，其中在VM之间没有设备共享。直接指派或设备直通以最小的开销提供接近本地的(native)性能。物理设备绕过管理程序并且直接附连到VM。但是，这种直接设备指派的缺点是有限的可伸缩性，因为在虚拟机之间不存在共享—一个物理网卡与一个VM耦合。

[0064] 根据实施例，单根IOV(SR-IOV)可以允许物理设备通过硬件虚拟化而表现为相同设备的多个独立的轻量级实例。这些实例可以被指派给VM作为直通设备，并且可以作为虚拟功能(VF)被访问。管理程序通过(每设备)唯一的、全特征的物理功能(PF)访问设备。SR-IOV使纯直接指派的可伸缩性问题得到缓解。但是，SR-IOV呈现出的问题是它可能会影响VM迁移。在这些IOV技术中，SR-IOV可以允许利用从多个VM直接访问单个物理设备的手段来扩展PCI Express(PCIe)规范，同时维持接近本地的性能。因此，SR-IOV可以提供良好的性能和可伸缩性。

[0065] SR-IOV通过向每个客户分配一个虚拟设备来允许PCIe设备暴露可以在多个客户之间共享的多个虚拟设备。每个SR-IOV设备具有至少一个物理功能(PF)和一个或多个相关联的虚拟功能(VF)。PF是由虚拟机监视器(VMM)或管理程序控制的正常PCIe功能，而VF是轻量级的PCIe功能。每个VF都具有其自己的基地址(BAR)，并且被指派有唯一的请求者ID，该唯一的请求者ID使得I/O存储器管理单元(IOMMU)能够区分去往/来自不同VF的业务流。IOMMU还在PF和VF之间应用存储器和中断转换。

[0066] 但是，令人遗憾的是，在数据中心优化期望虚拟机的透明实时迁移的情况下，直接设备指派技术对云提供商造成障碍。实时迁移的实质是将VM的存储器内容复制到远程管理程序。然后在源管理程序处暂停VM，并且在目的地处恢复VM的操作。当使用软件仿真方法

时,网络接口是虚拟的,因此它们的内部状态被存储到存储器中并且也被复制。因此,可以使停机时间下降到几毫秒。

[0067] 但是,当使用诸如SR-IOV之类的直接设备指派技术时,迁移变得更加困难。在这种情况下,网络接口的完整内部状态不能被复制,因为它与硬件绑定。作为替代,指派给VM的SR-IOV VF被分离,实时迁移将运行,并且新的VF将在目的地处被附连。在InfiniBand™和SR-IOV的情况下,该过程可以引入在秒的数量级上的停机时间。此外,在SR-IOV共享端口模型中,在迁移之后,VM的地址将改变,从而导致SM中的附加开销并且对底层网络构架的性能产生负面影响。

[0068] InfiniBand™ SR-IOV架构-共享端口

[0069] 可以存在不同类型的SR-IOV模型,例如,共享端口模型、虚拟交换机模型和虚拟端口模型。

[0070] 图4示出了根据实施例的示例性共享端口架构。如图所示,主机300(例如,主机通道适配器)可以与管理程序310交互,管理程序310可以将各种虚拟功能330、340、350指派给多个虚拟机。同样,物理功能可以由管理程序310处理。

[0071] 根据实施例,当使用诸如图4所示的共享端口架构时,主机(例如,HCA)表现为网络中的具有在物理功能320和虚拟功能330、350、350之间的单个共享LID和共享队列对(QP)空间的单个端口。但是,每个功能(即,物理功能和虚拟功能)可以具有其自己的GID。

[0072] 如图4所示,根据实施例,可以将不同的GID指派给虚拟功能和物理功能,并且物理功能拥有特殊的队列对QP0和QP1(即,用于InfiniBand™管理分组的专用队列对)。这些QP也被暴露给VF,但是VF不允许使用QP0(来自VF的朝着QP0的所有SMP都被丢弃),并且QP1可以充当由PF拥有的实际QP1的代理。

[0073] 根据实施例,共享端口架构可以允许不受VM(这些VM通过被指派给虚拟功能而附连到网络)的数量限制的高度可伸缩的数据中心,因为LID空间仅由网络中的物理机器和交换机消耗。

[0074] 但是,共享端口架构的缺点是无法提供透明的实时迁移,从而阻碍了灵活的VM放置的可能性。由于每个LID与特定管理程序相关联,并且在驻留在管理程序上的所有VM之间共享,因此迁移的VM(即,迁移到目的地管理程序的虚拟机)必须将其LID改变为目的地管理程序的LID。此外,由于受限的QP0访问,子网管理器不能在VM内部运行。

[0075] InfiniBand™ SR-IOV架构模型-虚拟交换机(vSwitch)

[0076] 图5示出了根据实施例的示例性vSwitch架构。如图所示,主机400(例如,主机通道适配器)可以与管理程序410交互,管理程序410可以将各种虚拟功能430、440、450指派给多个虚拟机。同样,物理功能可以由管理程序410处理。虚拟交换机415还可以由管理程序401处理。

[0077] 根据实施例,在vSwitch架构中,每个虚拟功能430、440、450是完整的虚拟主机通道适配器(vHCA),这意味着指派给VF的VM被指派完整的IB地址集合(例如,GID、GUID、LID)和硬件中的专用QP空间。对于网络的其余部分和SM,HCA 400经由虚拟交换机415以及连接到它的附加节点看起来像是交换机。管理程序410可以使用PF 420,并且(附连到虚拟功能的)VM使用VF。

[0078] 根据实施例,vSwitch架构提供透明的虚拟化。但是,由于每个虚拟功能都被指派

唯一的LID,因此可用LID的数量被迅速消耗。同样,在使用许多LID地址(即,每个物理功能和每个虚拟功能各自都有一个LID地址)的情况下,SM必须计算更多通信路径,并且更多的子网管理分组(SMP)必须被发送到交换机,以便更新它们的LFT。例如,在大型网络中通信路径的计算可能花费几分钟。因为LID空间限于49151个单播LID,并且由于每个VM(经由VF)、物理节点和交换机各自占用一个LID,因此网络中的物理节点和交换机的数量限制了活动VM的数量,反之亦然。

[0079] InfiniBand™ SR-IOV架构模型-虚拟端口(vPort)

[0080] 图6示出了根据实施例的示例性vPort概念。如图所示,主机300(例如,主机通道适配器)可以与管理程序410交互,管理程序410可以将各种虚拟功能330、340、350指派给多个虚拟机。同样,物理功能可以由管理程序310来处理。

[0081] 根据实施例,vPort概念被宽泛地定义以便赋予供应商实现的自由度(例如,定义不规定实现必须是特定于SRIOV的),并且vPort的目标是使在子网中处理VM的方式标准化。利用vPort概念,可以定义在空间域和性能域二者中可以更可伸缩的类似SR-IOV共享端口的架构和类似vSwitch的架构或两者的组合。vPort支持可选的LID,并且与共享端口不同的是,即使vPort不使用专用LID,SM也知道子网中所有可用的vPort。

[0082] InfiniBand™ SR-IOV架构模型-具有预填充的LID的vSwitch

[0083] 根据实施例,本公开提供了用于提供具有预填充的LID的vSwitch架构的系统和方法。

[0084] 图7示出了根据实施例的具有预填充的LID的示例性vSwitch架构。如图所示,多个交换机501-504可以在网络交换环境600(例如,IB子网)内提供在构架(诸如InfiniBand构架)的成员之间的通信。构架可以包括多个硬件设备,诸如主机通道适配器510、520、530。主机通道适配器510、520、530中的每个又可以分别与管理程序511、521和531交互。每个管理程序又可以与它所交互的主机通道适配器结合建立多个虚拟功能514、515、516、524、525、526、534、535、536并将这些虚拟功能指派给多个虚拟机。例如,虚拟机1 550可以由管理程序511指派给虚拟功能1 514。管理程序511还可以将虚拟机2 551指派给虚拟功能2 515,并且将虚拟机3 552指派给虚拟功能3 516。管理程序531又可以将虚拟机4 553指派给虚拟功能1 534。管理程序可以通过主机通道适配器中的每一个主机通道适配器上的全特征物理功能513、523、533来访问主机通道适配器。

[0085] 根据实施例,交换机501-504中的每一个可以包括多个端口(未示出),这些端口用于设置线性转发表以便引导网络交换环境600内的业务。

[0086] 根据实施例,虚拟交换机512、522和532可以由它们各自的管理程序511、521、531处理。在这样的vSwitch架构中,每个虚拟功能是完整的虚拟主机通道适配器(vHCA),这意味着指派给VF的VM被指派完整的IB地址(例如,GID、GUID、LID)集合和硬件中的专用QP空间。对于网络的其余部分和SM(未示出),HCA 510、520和530经由虚拟交换机以及连接到它们的附加节点看起来像交换机。

[0087] 根据实施例,本公开提供了用于提供具有预填充的LID的vSwitch架构的系统和方法。参考图7,LID被预填充到各个物理功能513、523、533以及虚拟功能514-516、524-526、534-536(甚至当前未与活动虚拟机相关联的那些虚拟功能)。例如,物理功能513用LID 1预填充,而虚拟功能1 534用LID 10预填充。当网络启动时,在启用SR-IOV vSwitch的子网中

LID被预填充。即使当并非所有的VF都在网络中被VM占用时，填充的VF也被指派有LID，如图7所示。

[0088] 根据实施例，非常类似于物理主机通道适配器可以具有多于一个端口(为了冗余两个端口是常见的)，虚拟HCA也可以用两个端口表示，并且可以经由一个、两个或更多个虚拟交换机连接到外部IB子网。

[0089] 根据实施例，在具有预填充LID的vSwitch架构中，每个管理程序可以通过PF为它自己消耗一个LID，并且为每个附加的VF消耗再多一个LID。在IB子网中的所有管理程序中可用的所有VF的总和给出了允许在子网中运行的VM的最大数量。例如，在子网中每管理程序具有16个虚拟功能的IB子网中，那么每个管理程序在子网中消耗17个LID(用于16个虚拟功能中的每个虚拟功能的一个LID加上用于物理功能的一个LID)。在这种IB子网中，对于单个子网的理论管理程序限制由可用单播LID的数量决定，并且是：2891个(49151个可用LID除以每管理程序17个LID)，并且VM的总数(即，限制)是46256个(2891个管理程序乘以每管理程序16个VF)。(实际上，这些数字实际较小，因为IB子网中的每个交换机、路由器或专用SM节点也消耗LID)。注意的是，vSwitch不需要占用附加的LID，因为它可以与PF共享LID。

[0090] 根据实施例，在具有预填充的LID的vSwitch架构中，网络第一次启动时，为所有LID计算通信路径。当需要启动新的VM时，系统不必在子网中添加新的LID，否则该动作将导致网络的完全重新配置，包括路径重新计算，这是最耗时的部分。作为替代，VM的可用端口位于管理程序之一(即，可用的虚拟功能)中，并且将虚拟机附连到可用的虚拟功能。

[0091] 根据实施例，具有预填充的LID的vSwitch架构还允许计算和使用不同路径以到达由同一管理程序托管的不同VM的能力。本质上，这允许这样的子网和网络使用类似LID掩码控制(LMC)的特征来提供朝着一个物理机器的可替代路径，而不受要求LID必须是连续的LMC的限制的约束。当需要迁移VM并将其相关联的LID携带到目的地时，能够自由使用非连续的LID尤其有用。

[0092] 根据实施例，与以上示出的具有预填充的LID的vSwitch架构的优点一起，某些因素可以被考虑。例如，由于当网络启动时，LID在启用SR-IOV vSwitch的子网中被预填充，因此(例如，在启动时)初始路径计算会比如果LID没有被预填充所花费的时间长。

[0093] InfiniBand™ SR-IOV架构模型-具有动态LID指派的vSwitch

[0094] 根据实施例，本公开提供了用于提供具有动态LID指派的vSwitch架构的系统和方法。

[0095] 图8示出了根据实施例的具有动态LID指派的示例性vSwitch架构。如图所示，多个交换机501-504可以在网络交换环境700(例如，IB子网)内提供构架(诸如InfiniBand构架)的成员之间的通信。构架可以包括多个硬件设备，诸如主机通道适配器510、520、530。主机通道适配器510、520、530中的每一个又可以分别与管理程序511、521、531交互。每个管理程序又可以与它所交互的主机通道适配器结合建立多个虚拟功能514、515、516、524、525、526、534、535、536并将这些虚拟功能指派给多个虚拟机。例如，虚拟机1 550可以由管理程序511指派给虚拟功能1 514。管理程序511还可以将虚拟机2 551指派给虚拟功能2 515，并且将虚拟机3 552指派给虚拟功能3 516。管理程序531又可以将虚拟机4 553指派给虚拟功能1 534。管理程序可以通过主机通道适配器中的每一个主机通道适配器上的全特征物理功能513、523、533来访问主机通道适配器。

[0096] 根据实施例，交换机501-504中的每一个可以包括多个端口(未示出)，这些端口用于设置线性转发表以便引导网络交换环境700内的业务。

[0097] 根据实施例，虚拟交换机512、522和532可以由它们各自的管理程序511、521、531处理。在这样的vSwitch架构中，每个虚拟功能是完整的虚拟主机通道适配器(vHCA)，这意味着指派给VF的VM被指派完整的IB地址(例如，GID、GUID、LID)集合和硬件中的专用QP空间。对于网络的其余部分和SM(未示出)，HCA 510、520和530经由虚拟交换机以及连接到它们的附加节点看起来像交换机。

[0098] 根据实施例，本公开提供了用于提供具有动态LID指派的vSwitch架构的系统和方法。参考图8，LID被动态指派给各个物理功能513、523、533，其中物理功能513接收LID 1、物理功能523接收LID 2并且物理功能533接收LID 3。与活动虚拟机相关联的那些虚拟功能也可以接收动态指派的LID。例如，由于虚拟机1 550是活动的并且与虚拟功能1 514相关联，因此虚拟功能514可以被指派LID 5。类似地，虚拟功能2 515、虚拟功能3 516和虚拟功能1534各自与活动的虚拟功能相关联。因为这一点，所以这些虚拟功能被指派LID，其中LID 7被指派给虚拟功能2 515，LID 11被指派给虚拟功能3 516，并且LID 9被指派给虚拟功能1534。与具有预填充的LID的vSwitch不同，当前未与活动虚拟机相关联的那些虚拟功能不接收LID指派。

[0099] 根据实施例，利用动态LID指派，可以显著减少初始的路径计算。当网络第一次启动并且不存在VM时，那么相对少数量的LID可以被用于初始的路径计算和LFT分发。

[0100] 根据实施例，非常类似于物理主机通道适配器可以具有多于一个的端口(为了冗余两个端口是常见的)，虚拟HCA也可以用两个端口表示，并且可以经由一个、两个或更多个虚拟交换机连接到外部IB子网。

[0101] 根据实施例，当在利用具有动态LID指派的vSwitch的系统中创建新的VM时，找到空闲的VM槽，以便决定在哪个管理程序上启动新添加的VM，并且还找到唯一的未使用的单播LID。但是，不存在网络中的已知路径和交换机的LFT以用于处理新添加的LID。为了处理新添加的VM而计算新的路径集合在每分钟可以启动若干VM的动态环境中是不期望的。在大型IB子网中，计算一组新的路由可以花费几分钟，并且每当启动新的VM时这个过程将必须重复。

[0102] 有利的是，根据实施例，由于管理程序中的所有VF与PF共享相同的上行链路，因此不需要计算一组新的路由。仅需要遍历网络中的所有物理交换机的LFT，将来自属于管理程序(在该管理程序中创建VM)的PF的LID条目的转发端口复制到新添加的LID，并且发送单个SMP以更新特定交换机的对应LFT块。因此，该系统和方法避免了计算一组新的路由的需要。

[0103] 根据实施例，在具有动态LID指派架构的vSwitch中指派的LID不一定是连续的(sequential)。当将具有预填充的LID的vSwitch中的每个管理程序上的VM上指派的LID与具有动态LID指派的vSwitch进行比较时，应当注意的是，在动态LID指派架构中指派的LID是非连续的，而被预填充的LID本质上是连续的。在vSwitch动态LID指派架构中，当创建新的VM时，在VM的整个生命周期中使用下一个可用的LID。相反，在具有预填充的LID的vSwitch中，每个VM都继承已经指派给对应VF的LID，并且在没有实时迁移的网络中，相继附连到给定VF的VM获得相同的LID。

[0104] 根据实施例，具有动态LID指派架构的vSwitch可以以一些附加的网络和运行时SM

开销为代价来解决具有预填充的LID架构模型的vSwitch的缺点。每次创建VM时，用与创建的VM相关联的新添加的LID来更新子网中的物理交换机的LFT。对于这个操作，需要发送每交换机一个子网管理分组(SMP)。因为每个VM正在使用与它的主机管理程序相同的路径，因此类似LMC的功能同样不可用。但是，对所有管理程序中存在的VF的总量没有限制，并且VF的数量可以超过单播LID极限的数量。当然，如果是这种情况，那么并不是所有VF都允许同时附连到活动的VM上，但是，当操作接近单播LID极限时，具有更多的备用管理程序和VF增加了对于灾难恢复和分段网络优化的灵活性。

[0105] InfiniBand™ SR-IOV架构模型-具有动态LID指派和预填充LID的vSwitch

[0106] 图9示出了根据实施例的具有动态LID指派和预填充的LID的vSwitch的示例性vSwitch架构。如图所示，多个交换机501-504可以在网络交换环境800(例如，IB子网)内提供构架(诸如InfiniBand构架)的成员之间的通信。构架可以包括多个硬件设备，诸如主机通道适配器510、520、530。主机通道适配器510、520、530中的每一个又可以分别与管理程序511、521和531交互。每个管理程序又可以与它所交互的主机通道适配器结合建立多个虚拟功能514、515、516、524、525、526、534、535、536并将这些虚拟功能指派给多个虚拟机。例如，虚拟机1 550可以由管理程序511指派给虚拟功能1 514。管理程序511还可以将虚拟机2 551指派给虚拟功能2 515。管理程序521可以将虚拟机3 552指派给虚拟功能3 526。管理程序531又可以将虚拟机4 553指派给虚拟功能2 535。管理程序可以通过主机通道适配器中的每一个主机通道适配器上的全特征物理功能513、523、533访问主机通道适配器。

[0107] 根据实施例，交换机501-504中的每一个可以包括多个端口(未示出)，这些端口用于设置线性转发表以便引导网络交换环境800内的业务。

[0108] 根据实施例，虚拟交换机512、522和532可以由它们各自的管理程序511、521、531处理。在这样的vSwitch架构中，每个虚拟功能是完整的虚拟主机通道适配器(vHCA)，这意味着指派给VF的VM被指派完整的IB地址(例如，GID、GUID、LID)集合和硬件中的专用QP空间。对于网络的其余部分和SM(未示出)，HCA 510、520和530经由虚拟交换机以及连接到它们的附加节点看起来像交换机。

[0109] 根据实施例，本公开提供了用于提供具有动态LID指派和预填充的LID的混合vSwitch架构的系统和方法。参考图9，管理程序511可以被布置有具有预填充的LID架构的vSwitch，而管理程序521可以被布置有具有预填充的LID和动态LID指派的vSwitch。管理程序531可以被布置有具有动态LID指派的vSwitch。因此，物理功能513和虚拟功能514-516使它们的LID被预填充(即，即使未附连到活动虚拟机的那些虚拟功能也被指派了LID)。物理功能523和虚拟功能1 524可以使它们的LID被预填充，而虚拟功能2和3、525和526使它们的LID被动态指派(即，虚拟功能2 525可用于动态LID指派，并且由于虚拟机3 552被附连，因此虚拟功能3 526具有动态指派的LID 11)。最后，与管理程序3 531相关联的功能(物理功能和虚拟功能)可以使它们的LID被动态指派。这使得虚拟功能1和3、534和536可用于动态LID指派，而虚拟功能2 535由于虚拟机4 553被附连于此所以具有动态指派的LID 9。

[0110] 根据(在任何给定管理程序内独立地或组合地)利用了具有预填充的LID的vSwitch和具有动态LID指派的vSwitch两者的实施例(诸如图9所示的实施例)，每主机通道适配器的预填充的LID的数量可以由构架管理员定义，并且可以在0<＝预填充的VF<＝(每主机通道适配器的)总VF的范围内，并且可以通过从(每主机通道适配器的)VF的总数减去

预填充的VF的数量来找到可用于动态LID指派的VF。

[0111] 根据实施例，非常类似于物理主机通道适配器可以具有多于一个端口(为了冗余两个端口是常见的)，虚拟HCA也可以用两个端口表示，并且可以经由一个、两个或更多个虚拟交换机连接到外部IB子网。

[0112] InfiniBand™-子网间通信

[0113] 根据实施例，除了提供在单个子网内的InfiniBand™构架之外，当前公开的实施例还可以提供跨越两个或更多个子网的InfiniBand™构架。

[0114] 图10示出了根据实施例的示例性多子网InfiniBand™构架。如图中所描绘的，在子网A 1000内，多个交换机1001-1004可以提供诸如InfiniBand™构架之类的构架的成员之间的在子网A 1000(例如，IB子网)内的通信。构架可以包括多个硬件设备，诸如像通道适配器1010。主机通道适配器1010又可以与管理程序1011交互。管理程序又可以与它所交互的主机通道适配器结合设置多个虚拟功能1014。管理程序还可以将虚拟机指派给虚拟功能中的每个虚拟功能，诸如将虚拟机1 1015指派给虚拟功能1 1014。管理程序可以通过主机通道适配器中的每个主机通道适配器上全特征的(full-featured)物理功能(诸如物理功能1013)来访问其相关联的主机通道适配器。

[0115] 进一步参考图10，并且根据实施例，多个交换机1021-1024可以提供诸如InfiniBand™构架之类的构架的成员之间的在子网B 1040(例如，IB子网)内的通信。构架可以包括多个硬件设备，诸如像主机通道适配器1030。主机通道适配器1030又可以与管理程序1031交互。管理程序又可以与它所交互的主机通道适配器结合设置多个虚拟功能1034。管理程序还可以向虚拟功能中的每个虚拟功能指派虚拟机，诸如将虚拟机2 1035指派给虚拟功能2 1034。管理程序可以通过主机通道适配器中的每个主机通道适配器上的全特征的物理功能(诸如物理功能1033)来访问其相关联的主机通道适配器。要注意的是，虽然在每个子网(即，子网A和子网B)内示出了仅一个主机通道适配器，但是应当理解的是，多个主机通道适配器及其对应的部件可以被包括在每个子网内。

[0116] 根据实施例，主机通道适配器中的每个主机通道适配器还可以与虚拟交换机(诸如虚拟交换机1012和虚拟交换机1032)相关联，并且每个HCA可以用如上面所讨论的不同的架构模型建立。虽然图10内的两个子网都被示为使用具有预填充的LID架构模型的vSwitch，但是这并不意味着所有这样的子网配置都必须遵循相似的架构模型。

[0117] 根据实施例，每个子网内的至少一个交换机可以与路由器相关联，诸如子网A 1000内的交换机1002与路由器1005相关联，以及子网B 1040内的交换机1021与路由器1006相关联。

[0118] 根据实施例，当起始源(诸如子网A内的虚拟机1)处的业务被寻址到不同子网处的目的地(诸如子网B内的虚拟机2)时，业务可被寻址到子网A内的路由器(即，路由器1005)，然后该路由器可以经由它与路由器1006的链路将业务传递到子网B。

[0119] 构架管理器

[0120] 如上面所讨论的，网络构架(诸如InfiniBand™构架)可以通过在构架的每个子网中使用互连路由器而跨越多个子网。根据实施例，构架管理器(未示出)可以在作为网络构架的成员的主机上实现，并且可以在构架内被用来管理作为构架的一部分的物理资源和逻辑资源二者。例如，管理任务(诸如发现构架资源、控制物理服务器之间的连接性、收集和查

看实时网络统计信息、灾难恢复以及设置服务质量(QoS)设置等)可以由用户通过构架管理器来执行。根据实施例,构架管理器可以跨越构架中定义的所有子网。即,无论资源是哪个子网的成员,构架管理器都可以整体地(at large)管理作为该构架的成员或与该构架相关联的物理资源和逻辑资源。

[0121] 根据实施例,构架管理器可以包括用户可以通过其执行管理功能的图形用户界面(GUI)。构架管理器GUI可以结合允许用户监视和控制构架资源的可视化工具。例如,在实施例中,用户可以通过构架界面来查看用于跨构架的服务器的服务器连接、配置设置和性能统计信息。可以通过构架管理器GUI监视和/或管理的构架功能的其它示例包括发现子网间构架拓扑、查看这些拓扑的可视表示、创建构架简档(例如,虚拟机构架简档)以及构建和管理构架管理器数据库,该构架管理器数据库可以存储构架简档、元数据、配置设置以及网络构架需要并且与网络构架相关的其它数据。根据实施例,构架管理器数据库是构架级数据库。

[0122] 此外,构架管理器可以在哪些子网被允许使用哪些分区号经由哪些路由器端口进行通信的方面来定义合法的子网间连接性。根据实施例,构架管理器是集中式构架管理实用程序。上面的示例并不意味着进行限制。

[0123] 根据实施例,构架管理器的功能中的一些功能可以由用户发起,而其它功能可以从用户抽象化,或者被自动化(例如,一些功能可以由构架管理器在启动时或在其它预定事件时执行)。

[0124] 在管理事件的示例性实施例中,用户可以在构架管理器接口处发起针对网络构架设备的配置改变。在接收到配置改变请求之后,构架管理器又可以确保配置改变请求被适当地执行。例如,构架管理器可以将请求传送到设备并且确保配置改变被写入设备的配置。在一个实施例中,物理设备向构架管理器确认配置改变已经成功完成。根据实施例,构架管理器然后可以更新接口,以给出请求已经被执行的可视确认。另外,构架管理器可以将设备的配置持久化(persist)到构架管理器数据库,例如以用于灾难恢复或其它目的。

[0125] 根据实施例,构架管理器可以具有其它接口,诸如包括与GUI相比的一些、全部或更多功能的命令行接口。

[0126] 构架级资源域

[0127] 如上面所讨论的,构架管理器可以允许用户通过构架管理器的接口在整个网络构架中执行管理任务。根据实施例,构架管理器的附加功能是促进分层的基于角色的访问控制。在实施例中,基于角色的访问控制通过构架级资源域来实现。

[0128] 根据实施例,基于角色的访问控制基于构架用户的概念。来自人类管理员和外部管理应用二者的访问可以表示经认证的上下文,该经认证的上下文定义对于构架基础设施或构架资源的全部或子集的合法操作。例如,用户可以在构架中通过用户简档表示。即,在构架内,可以通过创建用户的简档并且向简档指派属性来定义用户。可以向用户简档指派用户名属性和密码属性,其中用户名在构架内是唯一的,由此唯一地识别用户。另外,用户简档可以与在构架中定义的某些角色相关联,这些角色将某些访问级别指派给构架内的不同资源。根据实施例,建立用户简档可以通过构架管理器接口来完成。用户简档的全部或部分可以被存储在构架管理器数据库中。此外,在实施例中,构架管理器可以与公知的用户目录(诸如Microsoft®的活动目录或LDAP目录)或者与例如用于远程认证的RADIUS联网协

议集成。

[0129] 根据实施例，构架管理器可以管理它通过构架级资源域(在本文中还被称为“资源域”或简称为“域”)发现的构架资源。资源域是在构架级定义的构架资源的逻辑分组。构架资源包括物理资源和逻辑资源二者。资源的一些示例包括构架设备(诸如HCA、物理节点和交换机)、构架简档(诸如虚拟机构架简档和用户简档)、虚拟机、云和I/O模块，等等。

[0130] 根据实施例，由构架管理器发现和管理的所有构架资源都驻留在构架中缺省存在(即，不需要设置或配置它)的缺省域中，并且可以通过构架管理器接口来访问。缺省域是最高级别的域-即，它是所有其它资源域的父域，并且所有其它资源域都存在于缺省域内。缺省域与构架级管理员(构架级管理员也是缺省存在的)相关联，并且在缺省域中被缺省地配置有管理特权。

[0131] 根据实施例，资源域表示资源管理的分层形式。例如，配置和管理缺省域的过程仅对构架级管理员可用。但是，可以由构架级管理员在缺省域内创建孩子域。例如，构架级管理员可以创建孩子域并且可以向孩子域添加域资源。此外，构架级管理员可以创建域级“域管理”用户并且将域管理用户添加(即，关联)到孩子域。使域管理用户成为资源域的成员允许域管理用户管理孩子域以及构架资源的孩子域所包含的子集。根据实施例，域管理用户不能管理孩子域以外的资源(即，处于与域管理用户所关联的级别平行的级别或比域管理用户所关联的级别高的级别的资源)。但是，域管理用户(admin)可以管理资源域中所包含的已被创建为资源域的孩子域的资源。根据实施例，构架管理器负责提供确保资源域边界被严格实施的安全性。

[0132] 图11示出了资源域的分层结构。如图所示，缺省域1102存在于网络构架1100内。构架级管理员1110具有管理构架级资源1112、1124和1134的访问权限。构架级管理员1110还可以在缺省域1102内创建和管理新的资源域。构架级管理员1110已经创建了资源域1120和1130以及对应的域级域管理用户1122和1132。域管理用户1122具有管理(由构架级管理员1110指派给资源域1120的)构架资源1124的访问权限，但是没有管理(在更高级别的)构架资源1112或(在平行级别的)域资源1134的访问权限。类似地，域管理用户1132具有管理(由构架级管理员1110指派给资源域1130的)构架资源1134的访问权限，但是没有管理(在更高级别的)构架资源1112或(在平行级别的)域资源1124的访问权限。

[0133] 管理分区

[0134] 根据实施例，资源域可以由管理分区或“管理(admin)”分区(如它们在本文中被称作的)在子网级别表示。管理分区表示授予在子网级别对子网资源的访问权限的组成员资格。根据实施例，管理分区的成员被认为是有特权的，因为这些成员具有对与该管理分区相关联的任何子网资源的访问权限。在构架管理器级别，管理分区与资源域和对应的域管理用户相关联。因此，可以在子网级别在多租户环境中确保用户角色分离。此外，资源域成员资格可以与管理分区成员资格相关，以使得作为与特定资源域相关联的管理分区的成员的资源也是资源域的成员。

[0135] 根据实施例，可以按与定义数据分区相同的方式在子网级别定义管理分区，但是管理分区还具有指定正在被创建的分区是管理分区的附加属性。像(上面详细讨论的)数据分区一样，根据实施例，管理分区可以由管理员通过构架管理器接口来创建。在实施例中，构架管理器可以在创建分区期间支持作为可选参数的“管理分区”标记。如果该“管理分区”

标记由创建管理员选择，那么构架管理器将包括指定新创建的分区是管理分区并且将被构架管理器和本地主子网管理器视为管理分区的附加属性。

[0136] 根据实施例，构架管理器可以被配置为针对所创建的每个资源域自动创建对应的管理分区，并且将自动创建的分区与对应的资源域相关联。在这种实施例中，当将构架级资源添加到资源域时，构架管理器还将它们与被自动创建并且与资源域关联的管理分区相关联。因此，添加到资源域的资源在被添加到资源域之后将具有对彼此的子网级访问权限，而无需管理员(例如，构架级管理员或域管理用户)采取进一步行动。

[0137] 此外，根据实施例，网络的整个子网可以表示域层次结构中的具有最高级域(例如，缺省域)的特殊资源域。例如，在缺省域表示最高级域的域层次结构中，网络构架的每个子网然后可以被构架管理器识别为缺省域的孩子域。将全部子网识别为最高级域的孩子域可以被配置为构架管理器的缺省行为，或者这些缺省域可以由管理员手动定义。在这里同样，为了在子网级别具有用户角色分离以及实施域边界和资源关联，可以定义与全部子网资源域对应的管理分区。根据实施例，在子网中定义并且包括该子网中的每个资源(作为管理分区的成员或者与管理分区相关联)的管理分区可以被称为“域全局”管理分区，因为在这种配置中，子网中的每个资源都将具有对每个其它资源的访问权限。

[0138] 根据实施例，管理分区对于域管理用户可以是透明的。如上面所指出的，可以在构架管理器级别针对资源域自动创建域全局管理分区，并且然后指派给这个域或在这个域的范围内创建的所有资源都可以自动与对应的管理分区相关联。但是，在另一个实施例中，域管理用户可以在相关资源域内显式地创建不同的管理分区，并且然后域内的资源可以与显式地创建的管理分区显式地关联而不是与针对资源域缺省创建的管理分区相关联。

[0139] 根据实施例，构架管理器可以支持共享管理分区和私有管理分区二者的创建。由构架级管理员在缺省域中创建的管理分区可以是对于各个资源域可用的共享分区。由域管理用户(即，具有与特定资源域相关联的凭证的用户)在该域管理用户是其中的成员的域中创建的管理分区可以是与特定资源域相关联并且仅对该特定资源域可用的私有分区，其中这些管理分区在该特定资源域的上下文中被创建。

[0140] 根据实施例，HCA和vHCA的末端端口可以是管理分区的成员，就像它们可以是数据分区的成员一样。但是，根据实施例，管理分区与数据分区不同，因为管理分区可以与其它子网资源相关联。例如，数据分区可以与管理分区相关联。此外，根据实施例，管理分区可以(作为孩子或父亲)与另一个管理分区相关联，从而使得管理分区是分层的概念并且能够与它们所关联的资源域的层次结构对应。

[0141] 作为技术问题，根据实施例，HCA(和vHCA)的末端端口可以以传统术语被称为分区的“成员”，并且其它资源可以与管理分区“关联”。下面解释这两个概念的技术差异。但是，为了方便和可读性，本文档可能偶尔会在提到管理分区时可互换地使用术语“成员”和“关联”。尽管这些术语的使用是可互换的，但应当理解的是，管理分区中的末端端口/HCA成员资格与和管理分区的资源关联之间的技术差异意在由读者一致地应用。

[0142] 根据实施例，管理分区由P_Key定义，正如定义数据分区一样。然而，虽然末端端口知道它是其成员的数据分区，但是末端端口知道它们是哪些管理分区的成员是不必要的。因此，在一个实施例中，定义管理分区的P_Key不被输入到成员末端端口的P_Key表中。以这种方式，如果管理分区不用于IB分组业务，那么创建管理分区不会浪费P_Key表条目，P_Key

表条目是有限的资源。但是，在另一个实施例中，管理分区可以充当管理分区和数据分区二者。在这种实施例中，作为管理分区的成员的末端端口的所有P_Key表可以在它们各自的P_Key表中具有用于管理分区的P_Key条目。根据实施例，数据分区可以被定义为不同时是管理分区的任何分区。

[0143] 根据实施例，数据分区可以与一个或多个管理分区相关联。例如，由P_Key值定义的数据分区可以与由管理分区自己的不同P_Key值定义的管理分区相关联。此外，数据分区可以与由另一个不同的P_Key值定义的第二管理分区相关联。根据实施例，数据分区与特定管理分区的关联可以定义用于作为该特定管理分区的成员的末端端口的最大成员资格级别。

[0144] 如上面所指出的，管理分区表示授予对子网资源的访问权限的组成员资格。根据实施例，管理分区的任何末端端口成员仅基于该末端端口在管理分区中的成员资格而具有对与同一管理分区相关联的任何子网资源的访问权限。因此，作为管理分区的成员的任何末端端口都具有对与同一管理分区相关联的任何数据分区的访问权限。值得注意的是，这不一定意味着成员末端端口是相关联的数据分区的成员，而是意味着它具有对相关联的数据分区的访问权限并且因此可以是该数据分区的成员。

[0145] 这种方案避免了需要管理员通过在末端端口的P_Key表中手动包括数据分区的P_Key来授予末端端口对例如数据分区的访问。在实施例中，当在子网中初始化末端端口时，主子网管理器可以查询保持管理分区定义(例如，P_Key)以及定义所定义的管理分区中的成员资格并且定义与所定义的管理分区的关联的关系的数据存储库(例如，如下面所讨论的管理分区注册表)，以确定末端端口是哪些管理分区的成员。然后，子网管理器还可以检查是否存在与末端端口是其成员的管理分区相关联的任何数据分区。如果SM发现1)末端端口是管理分区的成员，并且2)该管理分区与数据分区相关联，那么SM可以将相关联的数据分区的P_Key自动放置在末端端口的P_Key表中，从而向末端端口自动授予对数据分区的访问。因此，管理分区表示比管理员进行手动分区映射更简单、更有扩展性的解决方案。

[0146] 图12示出了具有管理分区和数据分区二者的示例性网络构架。如图12所示，管理分区1230、1240和1250已经在构架内被定义。节点A-E 1201-1205通过它们各自的HCA 1211-1215物理地连接到构架。此外，每个HCA是至少一个管理分区的成员。HCA 1211和HCA 1214是管理分区1230的成员。HCA 1211以及HCA 1212和1213还是管理分区1240的成员。此外，HCA 1213以及HCA 1215是管理分区1250的成员。

[0147] 进一步参考图12，并且根据实施例，数据分区1232、1242和1252已经在构架内被定义。数据分区1232与管理分区1230相关联，数据分区1242与管理分区1240相关联，并且数据分区1252与管理分区1250相关联。根据实施例，HCA 1211和HCA 1214基于它们在管理分区1230中的成员资格而具有对数据分区1232中的成员资格的访问权限。同样，HCA 1211-1213基于它们在管理分区1240中的成员资格而具有对数据分区1242中的成员资格的访问权限。此外，HCA 1213和1215基于它们在管理分区1250中的成员资格而具有对数据分区1252中的成员资格的访问权限。

[0148] 根据实施例，还可以使用管理分区来确定是否可以向物理HCA的虚拟功能注册vHCA。根据实施例，vHCA描述针对具体虚拟机(VM)计划并且配置的主机通道适配器。vHCA与虚拟功能(VF)的不同之处在于，vHCA与VM一起迁移，而VF则与物理适配器留在一起。但是，

如上面所讨论的，物理HCA和vHCA(以及在较低级别上，这些(v)HCA的末端端口)都可以是管理分区的成员。因此，根据实施例，管理分区成员资格可以由SM用来确定来自物理HCA的、向发出请求的物理HCA的虚拟功能注册vHCA的请求是否是许可的。

[0149] 图13示出了具有作为管理分区的成员的HCA和vHCA的示例性网络构架。如图13所示，子网1302是网络构架1300的一部分。HCA 1310、1324、1332和1344表示在子网1302中通过它们各自的末端端口物理地连接到网络构架1300的物理HCA。HCA 1310与物理功能(PF)1312以及与虚拟功能(VF)1314和1316相关联。HCA 1324与PF 1326以及与VF 1328和1329相关联。HCA 1332与PF 1334以及与VF 1336和1338相关联。HCA 1344与PF 1346以及与VF 1348和1349相关联。此外，vHCA 1320被描绘为向VF 1314注册，并且与虚拟机(VM)1318相关联(即，VM 1318通过vHCA 1320并且最终通过物理HCA 1310获得对网络构架1300的访问)。vHCA 1340向VF 1337注册并且与VM 1338相关联。

[0150] 继续参考图13，如图所示，HCA 1310和1324以及vHCA 1320是管理分区1350的成员。此外，HCA 1332和1344以及vHCA 1340是管理分区1360的成员。因此，由于HCA 1310和1324以及vHCA 1320各自是管理分区1350的成员的事实，因此vHCA 1320可以合法地向HCA 1310的VF 1314或1316或者向HCA 1324的VF 1328或1329注册。类似地，由于HCA 1332和1324以及vHCA 1340各自是管理分区1360的成员的事实，因此vHCA 1340可以合法地向HCA 1330的VF 1336或1338或者向HCA 1344的VF 1348或1349注册。

[0151] 如上面所指出的，构架级构架数据库保持与构架和构架资源相关的信息，并且由构架管理器管理。根据实施例，构架数据库可以具有对构架资源清单的“完整知识”(即，作为网络构架的一部分的每个资源至少由保持在构架数据库中的记录来表示)。此外，与构架中的每个资源相关联的访问权限和名称空间可以被存储在构架数据库中，或者从包含在构架数据库中的信息和关系中导出。

[0152] 例如，根据实施例，与管理分区成员资格和/或与管理分区的资源关联有关的信息可以被存储在构架数据库中。保持这种信息的表以及将这些表链接在一起的关系可以是构架数据库的子集，并且可以被称为管理分区注册表。根据实施例，管理分区注册表是管理分区组资源的集合。例如，管理分区注册表内的管理分区组可以是特定管理分区的相关联资源和HCA成员(包括vHCA)的集合，其中该组由定义该特定管理分区的P_Key来查找。此外，管理分区组成员和相关联的资源可以在注册表中使用键(诸如分别用于成员HCA或vHCA的GUID或vGUID或者用于相关联的数据分区的P_Key)来查找。管理分区的P_Key与成员或相关联资源的唯一标识符之间的关系分别定义管理分区中的成员资格或关联，并且由管理分区注册表维护并且在更高级别由构架数据库维护。

[0153] 根据实施例，管理分区注册表的全部或部分可以作为记录被保持在SM的高速缓存中。例如，管理分区注册表的与特定子网的资源对应的记录可以在特定子网的子网管理器(例如，主子网管理器)的驻留存储器中的高速缓存中被复制。管理分区注册表记录可以(例如，当SM启动时)由SM从构架数据库检索(即，复制)，或者在它被持久化到构架数据库之前被放置在高速缓存中。高速缓存可以是易失性或非易失性存储器。无论何时将注册表记录放置在高速缓存中，都可以在管理分区注册表的被高速缓存的副本与在构架级数据库中找到的管理分区注册表副本之间进行同步。

[0154] 通过将管理分区注册表的全部或子网相关部分保持在SM上的高速度的高速缓存

中，而不是在每次接收到查询时从持久化状态(即，从构架数据库)中检索管理分区信息，管理分区信息的查找可以对SM施加最小的开销。这在子网资源之间的访问权限被自动指派的子网初始化期间会是尤其重要的。

[0155] 根据实施例，可以(例如，由构架级别或域级别的管理用户)将逻辑名称或标识符指派给资源域内的资源。这些逻辑名称可以是资源域私有的。构架管理器可以通过构架数据库创建将构架内使用的唯一标识符(例如，vGUID和P_Key)映射到给予构架内的资源的逻辑名称或符号名称的关系。

[0156] 例如，根据实施例，构架数据库可以存储资源的记录、以及资源的域成员资格和/或管理分区成员资格。在构架管理器发现资源后，可以将逻辑名称指派给资源。这些名称可以被链接到构架数据库中的构架资源的唯一标识符。此外，构架管理器可以通过构架管理器数据库中的关系来跟踪每个资源在资源域和管理分区中的成员资格。利用这些记录和关系，构架管理器可以允许跨不同资源域和管理分区的相同的(like)逻辑名称。根据实施例，逻辑域名称方案可以反映特定域资源是其成员的一个或多个资源域的层次结构。在这种实施例中，逻辑资源名称对于资源是其成员的最高级别资源域而言可以是唯一的。

[0157] 根据实施例，构架中的资源的标识符(无论该标识符是什么)在管理分区的范围内可以是唯一的。然后，可以通过在资源名称前加上对应的管理分区来实现全局唯一性(即，在构架级别的唯一性)。

[0158] 图14示出了具有资源域和管理分区二者的示例性网络构架。如图14所示，构架管理器1402在网络构架1400上执行。根据实施例，构架管理器1402可以从网络构架1400的节点(未示出)执行。构架管理器1402由构架级管理员1404管理，并且包括构架管理器数据库1414。管理分区注册表1416是构架管理器数据库1414的一部分，逻辑名称表1418也是构架管理器数据库1414的一部分。

[0159] 继续参考图14，在网络构架1400内定义子网1420。子网管理器1422与子网1420相关联，并且根据实施例执行子网1420为了在网络构架1400中操作所需的语义运行时操作。子网1420所需的设置和管理任务可以由构架级管理员1404和构架管理器1402执行。

[0160] 节点1444、1454、1474和1484是子网1420的一部分。HCA 1446与节点1444相关联，并且包括PF 1448以及VF 1450和1452。类似地，HCA 1456与节点1454相关联，并且包括PF 1458以及VF 1460和1462。HCA 1476与节点1474相关联，并且包括PF 1478以及VF 1480和1482。此外，HCA 1486与节点1484相关联，并且包括PF 1488以及VF 1490和1492。VM 1440在节点1444上执行，并且VM 1470在节点1474上执行。vHCA 1442已经针对VM 1440被计划和配置、与VM 1440相关联、并且向HCA 1446的虚拟功能1452注册。vHCA 1472已经针对VM 1470被计划和配置、与VM 1470相关联、并且向HCA 1476的虚拟功能1482注册。

[0161] 根据实施例，HCA 1446、1456、1476和1486被视为域资源，并且每个HCA的记录被存储在构架管理器数据库1414中。记录可以包括用于识别构架中的HCA资源的标识符，诸如GUID。另外，vHCA 1442和1472还被视为域资源，并且每个vHCA的记录被存储在构架管理器数据库1414中。记录可以包括用于识别vHCA的标识符，诸如GUID。

[0162] 进一步参考图14，并且根据实施例，在构架管理器1402内已经创建了资源域1410和资源域1412。根据实施例，构架级管理员1404负责创建资源域1410和资源域1412。此外，域管理用户1406是与资源域1410相关联的域级管理员。类似地，域管理用户1408是与资源

域1412相关联的域级管理员。根据实施例，构架级管理员1404可以创建域管理用户1406和1408，以作为它们相应的资源域的管理员，从而遵守资源域的分层性质。

[0163] 根据实施例，管理分区1424和管理分区1426已经在子网1420中定义。管理分区1424与资源域1410相关联，并且管理分区1426与资源域1412相关联。

[0164] 如图14所示，vHCA 1442以及HCA 1446和1456是资源域1410的成员。根据实施例，因为管理分区1424与资源域1410相关联，所以，当vHCA 1442以及HCA 1446和1456被添加为资源域1410的成员时，它们还成为管理分区1424的成员，并且在管理分区注册表1416中在定义管理分区1424的P_Key与HCA 1446和1456以及vHCA 1442的标识符之间创建关系。根据实施例，这种关系将HCA 1446和1456以及vHCA 1442定义为管理分区1424的成员。

[0165] 类似地，vHCA 1472以及HCA 1476和1486是资源域1412的成员。根据实施例，因为管理分区1426与资源域1410相关联，所以，当vHCA 1472以及HCA 1466和1486作为资源时域1412的成员被添加时，它们还成为管理分区1426的成员，并且在管理分区注册表1416中在定义管理分区1426的P_Key与HCA 1476和1486以及vHCA 1472的标识符之间创建关系。根据实施例，这个关系将HCA 1476和1486以及vHCA 1472定义为管理分区1426的成员。

[0166] 如上面所指出的，根据实施例，VM 1440(包括vHCA 1442)、节点1444(包括HCA 1446)和节点1454(包括HCA 1456)是资源域1410的成员。在本发明的实施例中，构架级管理员1404负责将节点1444和节点1454添加到资源域1410。例如，构架级管理员1404可以通过构架管理器1402的接口将节点1444和1454添加到资源域1410。一旦构架级管理员1404已经将节点1444和1454添加到资源域1410，域管理用户1406就可以对节点1444和1454执行管理任务。但是，为了与资源域的分层方案保持一致，域管理用户1406不能在节点1444和1454被添加到资源域1410之前(即，当它们是更高级别的缺省域(未示出)的成员时)对节点1444和1454执行管理任务)。另外，根据实施例，域管理用户1408不能对节点1444和1454执行管理任务，因为节点1444和1454是域管理用户1408未关联的平行级资源域的成员。

[0167] 继续参考图14，并且根据实施例，在子网1420内已经定义了管理分区1424和1426。为了与资源域的分层方案保持一致，在一个实施例中，管理分区1424和1426由构架级管理员1404定义。在另一个实施例中，域管理用户1406定义管理分区1424，并且域管理用户1408定义管理分区1426。根据实施例，管理分区1424与资源域1410相关联，并且管理分区1426与资源域1412相关联。如上面所讨论的，根据实施例，管理分区1424和1426分别在子网级别表示资源域1410和1412。除了与它们各自的资源域相关联之外，管理分区1424和1426还分别与域管理用户1406和1408(即，管理分区中的每个管理分区所关联的资源域的对应管理用户)相关联。如上面所指出的，根据实施例，管理分区和域级管理用户之间的这种关联可以确保在子网级别的多租户环境中的用户-角色分离。

[0168] 根据实施例，数据分区1428和1430已经在子网1420中被定义。为了与资源域的分层方案保持一致，在一个实施例中，数据分区1428和1430由构架级管理员1404定义。在另一个实施例中，域管理用户1406定义数据分区1428，并且域管理用户1408定义数据分区1430。如图14中所示，数据分区1428与管理分区1424相关联，并且数据分区1430与管理分区1426相关联。此外，如上面所指出并在图14中所示的，HCA 1446和1456以及vHCA 1442是管理分区1424的成员。因此，根据实施例，HCA 1446和1456以及vHCA 1442具有对数据分区1428的访问许可，因为它们是数据分区1428所关联的管理分区(即，管理分区1424)的成员。

[0169] 根据实施例，当数据分区1428与管理分区1424相关联时，在管理分区注册表1416中创建数据分区1428的标识符(例如，数据分区1428的P_Key)与管理分区1424的P_Key之间的关系。这种关系将数据分区1428定义为与管理分区1424相关联。类似地，当数据分区1430与管理分区1426相关联时，在管理分区注册表1416中创建数据分区1430的标识符(例如，数据分区1430的P_Key)与管理分区1426的P_Key之间的关系。这种关系将数据分区1430定义为与管理分区1426相关联。

[0170] 根据实施例，如果从HCA 1446和1456中的任一个或vHCA 1442接收到加入数据分区1428的请求，那么SM 1422可以检查管理分区注册表1416，并且发现HCA 1446和1456以及vHCA 1442是管理分区1424的成员并且数据分区1428与管理分区1424相关联。然后，SM 1422可以基于HCA 1446和1456以及vHCA 1442是管理分区1424的成员并且数据分区1428与管理分区1424相关联来允许HCA 1446和1456以及vHCA 1442成为数据分区1428的成员。不需要来自构架级管理员1404或域级管理员1406的手动映射来允许HCA 1446和1456以及vHCA 1442加入数据分区1428。

[0171] 此外，因为HCA 1446和1456以及vHCA 1442是管理分区1424的成员，所以vHCA 1442可以向HCA 1446的VF 1452或1450中的任一个或者HCA 1456的VF 1462或1460中的任一个注册(vHCA 1442被描绘为向VF 1452注册)。这里同样，SM 1422可以检查管理分区注册表1416，并且发现HCA 1446和1456以及vHCA 1442是管理分区1424的成员。一旦发现HCA 1446和1456以及vHCA 1442是管理分区1424的成员，SM 1422就可以允许vHCA 1442向VF 1452、1450、1462和1460中的任何VF注册，而无需来自任何构架用户的干预。

[0172] 虚拟机构架简档

[0173] 如上面所讨论的，虚拟机(VM)可以在诸如IB构架之类的构架中被采用，以便改进高效的硬件资源利用率和可伸缩性。然而，由于这些解决方案中使用的寻址和路由方案，虚拟机(VM)的实时迁移仍然是个问题。根据实施例，方法和系统提供了促进预定义的、高度可用的并且独立于物理位置的虚拟机构架简档，该虚拟机构架简档可以支持旨在解决这种VM迁移问题的寻址方案。根据实施例，VM构架简档使用构架连接性来实现对于VM的集中式设置和配置管理，并且基于SR-IOV支持用于VM的优化的VM迁移。根据实施例，VM构架简档表示用于虚拟机的详细构架配置信息的单个集中式储存库。与构架管理器相关联的数据库(例如，构架数据库)可以持久化构成VM构架简档的信息。

[0174] 根据实施例，可以通过虚拟机标识符(VM-id)在网络构架(诸如IB构架)中识别VM构架简档。在实施例中，VM-id是作为通用唯一标识符(UUID)的唯一128位数字，它可以是在整个构架中唯一的。但是，VM-id的唯一性仅仅跨被不同地管理的VM管理器域是必需的(例如，VM-id在管理分区内可以是唯一的)。因此，在其它实施例中，VM-id可以是至少跨这些域唯一的某种其它适当类型的ID。根据实施例，在构架级或者子网级，所有管理实体都通过参考简档的VM-id来查找关于VM构架简档的信息。

[0175] 图15示出了用于存储VM构架简档信息的示例性数据库结构。图15绘出了传统关系数据库设计中的若干表。但是，可以使用任何合适的数据结构来存储VM构架简档数据(例如，平面文件表或定界结构，等等)。在图15中，星号(*)表示键(key)值。图15将VM构架简档数据描绘为更大的构架数据库1500的一部分，但是在其它实施例中，VM简档数据可以被包含在它自己的数据库中，或者可以是能够访问构架数据库1500的分开的数据库。

[0176] 如图15所示,VM构架简档的内容可以包括但不限于:用作查找关键字的虚拟机标识符(VM-id)1502;为了在构架的管理中易于使用和改进质量的逻辑名称1504;简档类型1506,用于区分例如VM构架简档以及针对该构架已经被定义的其它简档;简档ID 1508,这是为构架定义的所有简档的集合内的唯一ID;以及内容更新枚举器(CUE)1510,它可以是用于简档的序列号,其中最高序列表示最近的更新。如图15所示,这个VM构架简档内容可以存储在VM简档表1512中,其中VM-id充当用于识别每个VM构架简档的唯一关键字。

[0177] 如上面所讨论的,虚拟HCA(vHCA)可以与物理HCA的虚拟功能结合使用,以向VM提供网络访问。根据实施例,每个VM构架简档还与至少一个vHCA相关联。vHCA可以针对具体的VM被计划和配置,并且这种配置还可以包括在针对其配置vHCA的VM的构架简档中并与该VM的构架简档存储在一起。然后,经配置的vHCA可以与虚拟机一起迁移,这与虚拟功能相反,虚拟功能可以由物理HCA定义并且可以与物理HCA留在一起。进一步参考图15,可以在构架管理器数据库中将vHCA表示为VM-id 1502和vHCA实例ID 1514的唯一组合。此外,可以通过VM-id关键字1502将这种组合存储在与VM简档表1512相关的vHCA表中。

[0178] 与物理HCA一样,vHCA可以具有多个(虚拟)端口。根据实施例,这些虚拟端口可以充当网络环境中的末端端口,就像物理端口一样。作为示例,可以向所有末端端口(包括vHCA端口)指派GUID(例如,如在IB网络中使用的64位GUID)。这个GUID可以被用于从SM的路由表请求LID目的地地址。根据实施例,虚拟GUID(vGUID)可以表示每个vHCA端口的当前构架地址。在一个实施例中,vGUID可以从如上面所讨论的分配给VM构架简档并且与VM构架简档一起存储的GUID列表被指派给vHCA端口。vGUID可以根据构架管理器GUID策略从由构架管理器拥有并且控制的专用GUID池中被指派给构架简档。例如,当在构架管理器中创建构架简档时,可以将空闲并且在构架范围唯一的vGUID分配给为VM配置的vHCA。

[0179] 继续参考图15,并且根据实施例,vHCA端口可以在构架管理器数据库中被表示为vHCA实例ID 1514和vGUID 1522的唯一组合。vHCA配置还可以包括vHCA端口号1520。这种配置可以存储在vHCA端口表1518中,vHCA端口表1518可以通过vHCA实例ID*1514关键字关联到vHCA表,并且通过VM-id关键字1502最终关联到VM简档表1512。

[0180] 根据实施例,vHCA可以是数据分区和管理分区二者的成员。vHCA的分区成员资格可以在VM构架简档中由构架数据库中将vHCA记录链接到vHCA是其成员的分区(管理分区和数据分区二者)的关系来表示。在一个实施例中,例如,vGUID关键字1522可以关联到包含已经在网络构架中定义的数据分区和管理分区P_Key的表(未示出)。在实施例中,vGUID关键字通过关系链接到管理(上面讨论的)分区注册表。在实施例中,可替代地或者还存在将管理分区注册表链接到vHCA实例ID 1514的关系。这些关系允许构架部件识别vHCA是哪些数据分区和管理分区的成员。

[0181] 仅仅为了示例性的目的而提供了图15,并且本领域技术人员将认识到存在许多方式来设计和管理构成VM构架简档的数据的存储。此外,VM构架简档内容的上述列表意在是示例性而不是限制性的,并且虚拟机构架简档的其它实施例可以包括更多的、更少的或其它内容。根据实施例,图15中绘出的数据库部件可以是大得多的构架管理器数据库的一部分,该构架管理器数据库保持关于构架的其它相关信息并且可以与其它表相互关联,以增强构架管理器和其它构架部件的功能。

[0182] 在实施例中,用户通过构架管理器的接口与虚拟机构架简档交互。例如,用户可以

通过构架管理器来创建、编辑和删除VM构架简档。根据实施例，构架简档信息中的一些通过用户创建或编辑VM构架简档(例如，VM构架简档的逻辑名称)来提供，而该信息的其它部分由构架管理器或构架管理器在其中被创建的子网的本地SM生成或供应(例如，VM-id、vHCA实例ID或vGUID)。

[0183] 根据实施例，虚拟机构架简档的创建可以发生在表示VM构架简档所关联的构架资源的管理特权的管理上下文中。例如，域管理用户创建VM构架简档，以供具有HCA的节点使用，其中这些HCA是与针对VM构架简档配置的vHCA相同的(一个或多个)资源域(以及相同的(一个或多个)管理分区)的成员。创建的VM构架简档被认为是(逻辑)资源，并且是它在其中被创建的资源域的成员。因此，通过作为同一管理分区的成员，VM构架简档的vHCA具有向同样是资源域的成员的HCA的VF中的任何VF注册的许可，由此减轻了管理开销。

[0184] 根据实施例，构架级或域级管理员用户可以使用构架管理器的被称为“虚拟机管理器”(VMM)的部件来设置和配置VM构架简档。VMM在创建VM构架简档时可以使用构架REST API。用户可以例如通过构架管理器的GUI访问VMM。根据实施例，管理用户可以供应与VM构架简档相关的某些参数(诸如逻辑名称、简档类型以及将与简档相关联的vHCA的数量)，并且其它参数可以由VMM自动生成和指派(诸如VM-id、与简档相关联的每个vHCA的vGUID和vHCA实例ID)。其它CRUD动作(诸如更新和删除VM构架简档)还可以通过构架管理器并且具体而言通过VMM对管理员用户可用。

[0185] 根据实施例，一旦已经经由VMM供应了构建VM构架简档所需的全部参数，构架管理器就可以创建具有由管理员用户和VMM指定的属性的VM构架简档对象的实例，并且将VM构架简档对象持久化到构架级数据库。

[0186] 根据实施例，在操作网络构架中，可以将VM构架简档保持为SM的高速缓存中的一个或多个记录。VM构架简档可以由SM(例如，在SM启动时)从构架数据库检索(即，复制)，或者在它被持久化到构架数据库之前被放置在高速缓存中。高速缓存可以是易失性或非易失性存储器。VM-id可以被用作查询高速缓存以检索具体VM构架简档的属性的关键字。

[0187] 通过将VM构架简档保持在SM上的高速度高速缓存中而不是在每次接收到查询时从持久化状态(即，从构架数据库)检索它们，查找VM构架简档属性可以对SM施加最小的开销。在将需要构架简档数据来建立VM可以并且将与哪些HCA配对的VM和主机启动期间，这是尤为重要的。

[0188] 图16是用于使VM构架简档对子网资源可用的流程图。

[0189] 在步骤1610，定义包括VM的设置参数和配置的VM构架简档。

[0190] 在步骤1620，将VM构架简档存储在构架级数据库中。

[0191] 在步骤1630，通过子网管理器上的高速度存储器高速缓存使得VM构架简档可用。

[0192] 在步骤1640，基于指向高速度存储器高速缓存的VM-id查找请求，从子网管理器返回VM构架简档数据。

[0193] 图17是用于为虚拟机创建虚拟机构架简档的流程图。

[0194] 在步骤1710，生成虚拟机标识符(VM-id)。

[0195] 在步骤1720，生成识别虚拟主机通道适配器(vHCA)的虚拟主机通道适配器实例ID。

[0196] 在步骤1730，从全局唯一标识符的池中指派虚拟全局唯一标识符(vGUID)作为

vHCA的虚拟端口的当前构架地址。

[0197] 在步骤1740,创建定义管理分区的P_Key与vGUID之间的第一关系,其中P_Key与vGUID之间的关系将vGUID作为当前构架地址被指派给的虚拟端口定义为由P_Key定义的管理分区的成员。

[0198] 在步骤1750,创建VM-id与vHCA实例ID之间的第二关系,其中第二关系允许通过访问VM-id来检索vHCA实例ID。

[0199] 在步骤1760,创建VM-id与vGUID之间的第三关系,其中第三关系允许通过访问VM-id来检索vGUID。

[0200] 在步骤1770,VM-id、虚拟主机通道适配器实例ID、vGUID、第一关系、第二关系和第三关系以保留VM-id、虚拟主机通道适配器、vGUID、第一关系、第二关系和第三关系的格式被持久化。

[0201] 本发明的许多特征可以在硬件、软件、固件或其组合中执行、使用硬件、软件、固件或其组合执行或者在硬件、软件、固件或其组合的帮助下执行。因此,可以使用(例如,包括一个或多个处理器的)处理系统来实现本发明的特征。

[0202] 本发明的特征可以在计算机程序产品中、利用计算机程序产品、或者在计算机程序产品的帮助下实现,其中该计算机程序产品是具有存储在其上/其中的可用来对处理系统编程以执行本文所呈现的特征中的任何特征的指令的存储介质(媒介)或计算机可读介质(媒介)。存储介质可以包括但不限于任何类型的盘,包括软盘、光盘、DVD、CD-ROM、微驱动器、以及磁光盘、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、闪存存储器设备、磁卡或光卡、纳米系统(包括分子存储器IC)、或适于存储指令和/或数据的任何类型的媒介或设备。

[0203] 通过被存储在机器可读介质(媒介)中的任何机器可读介质中,本发明的特征可以被结合到软件和/或固件中,以用于控制处理系统的硬件,并且用于使处理系统能够利用本发明的结果与其它机制交互。这种软件或固件可以包括但不限于应用代码、设备驱动程序、操作系统和执行环境/容器。

[0204] 本发明的特征还可以利用例如硬件部件(诸如专用集成电路(ASIC))在硬件中实现。为了执行本文所描述的功能的硬件状态机的实现对相关领域的技术人员将是明显的。

[0205] 此外,本发明可以方便地利用包括根据本公开的教导编程的一个或多个处理器、存储器和/或计算机可读存储介质的一个或多个常规的通用或专用数字计算机、计算设备、机器或微处理器来实现。适当的软件编码可以容易地由熟练的程序员基于本公开的教导来准备,如对软件领域的技术人员将明显的。

[0206] 虽然上文已经描述了本发明的各种实施例,但是应该理解的是,它们已作为示例而不是限制被给出。对相关领域的技术人员来说将明显的是,在不背离本发明的精神和范围的情况下,可以在其中做出各种形式和细节上的变化。

[0207] 已经借助示出具体功能及其关系的执行的功能构建块来描述了本发明。这些功能构建块的边界在本文中通常为了方便描述而被任意定义。可以定义可替代的边界,只要具体的功能及其关系被适当地执行。任何这种可替代的边界因此在本发明的范围和精神内。

[0208] 本发明的以上描述是为了说明和描述的目的被提供。它不旨在是详尽的或者要把本发明限定到所公开的精确形式。本发明的广度和范围不应当由上文描述的示例性实施例中的任何实施例来限制。许多修改和变化对本领域技术人员来说将是明显的。修改和变化

包括所公开的特征的任何相关组合。实施例被选择和描述是为了最好地解释本发明的原理及其实践应用，从而使本领域其它技术人员能够对于各种实施例并且利用适于预期的特定用途的各种修改来理解本发明。旨在由以下权利要求及其等价物来定义本发明的范围。

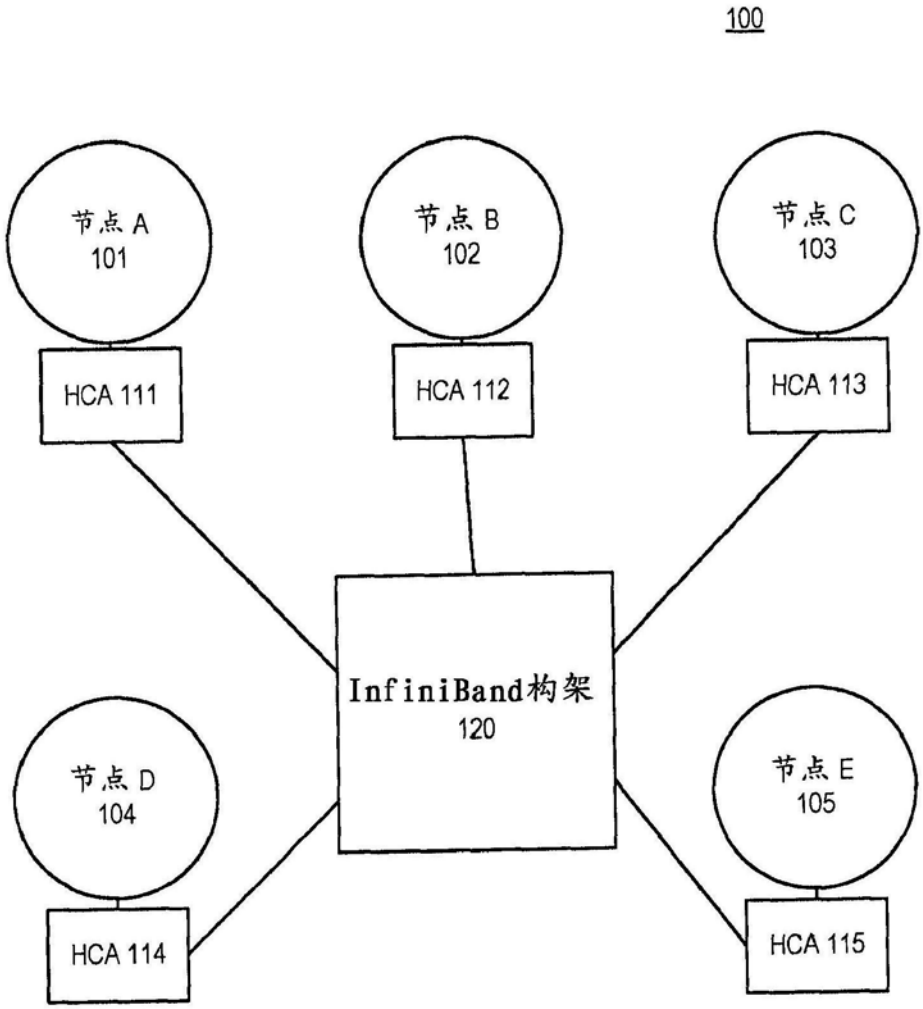

100
节点 A
101
节点 B
102
节点 C
103
HCA 111
HCA 112
HCA 113
InfiniBand构架
120
节点 D
104
节点 E
105
HCA 114
HCA 115

图1

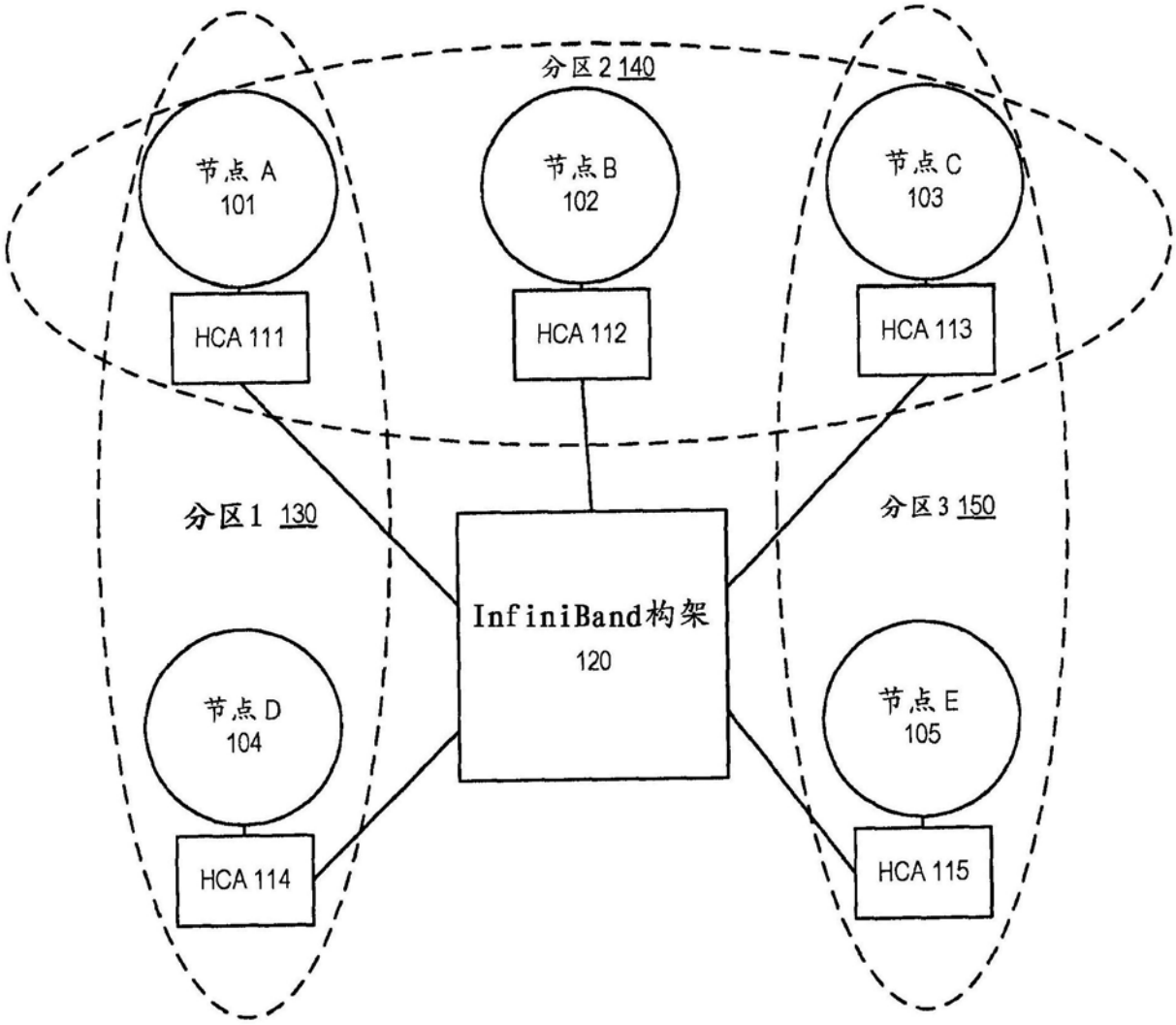

分区2 140
节点 A
101
节点B
102
节点 C
103
HCA 111
HCA 112
HCA 113
分区1 130
分区3 150
InfiniBand构架
120
节点 D
104
节点 E
105
HCA 114
HCA 115

图2

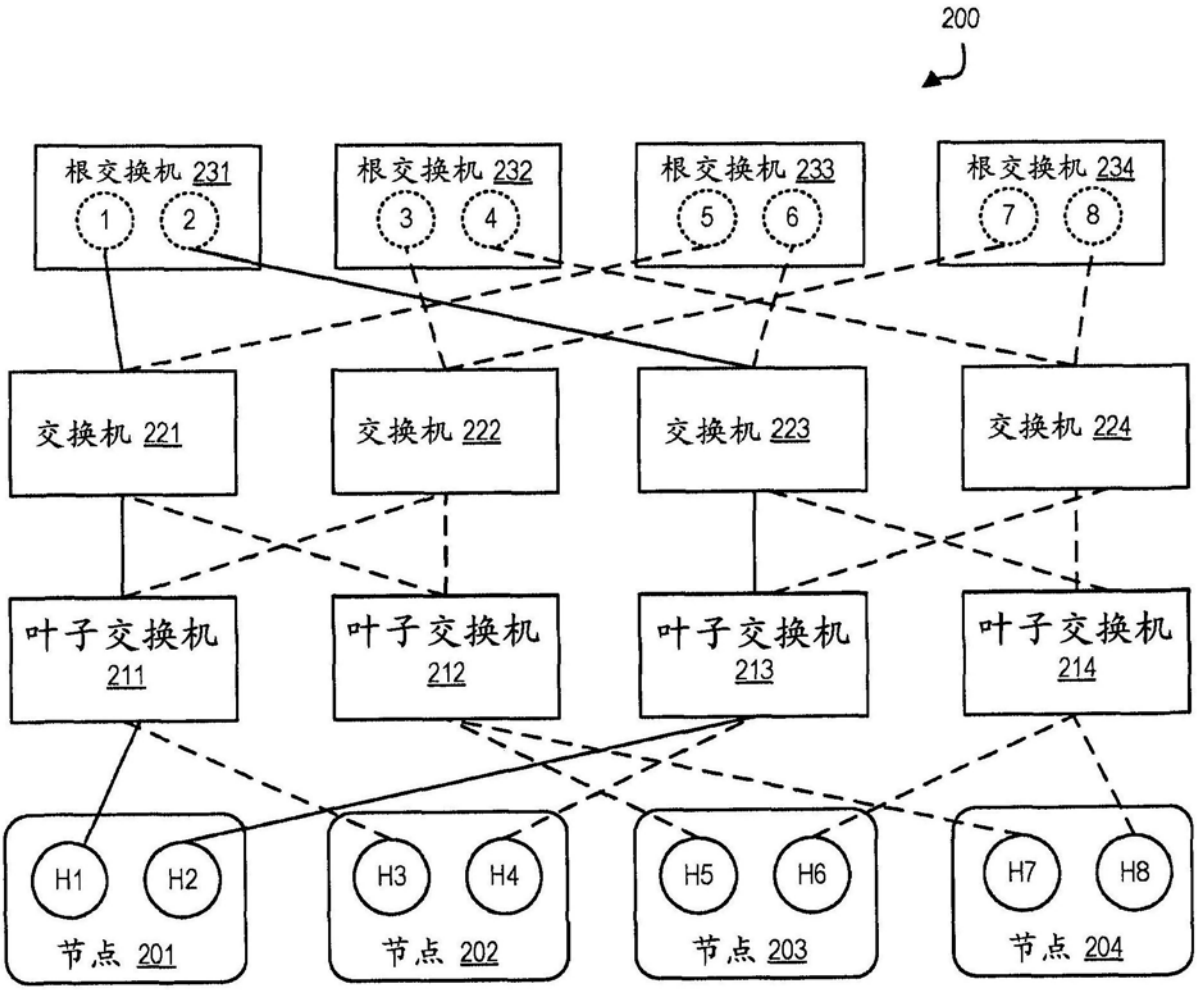

200
根交换机 231
1
2
根交换机 232
3
4
根交换机 233
5
6
根交换机 234
7
8
交换机 221
交换机 222
交换机 223
交换机 224
叶子交换机 211
叶子交换机 212
叶子交换机 213
叶子交换机 214
H1
H2
节点 201
H3
H4
节点 202
H5
H6
节点 203
H7
H8
节点 204

图3

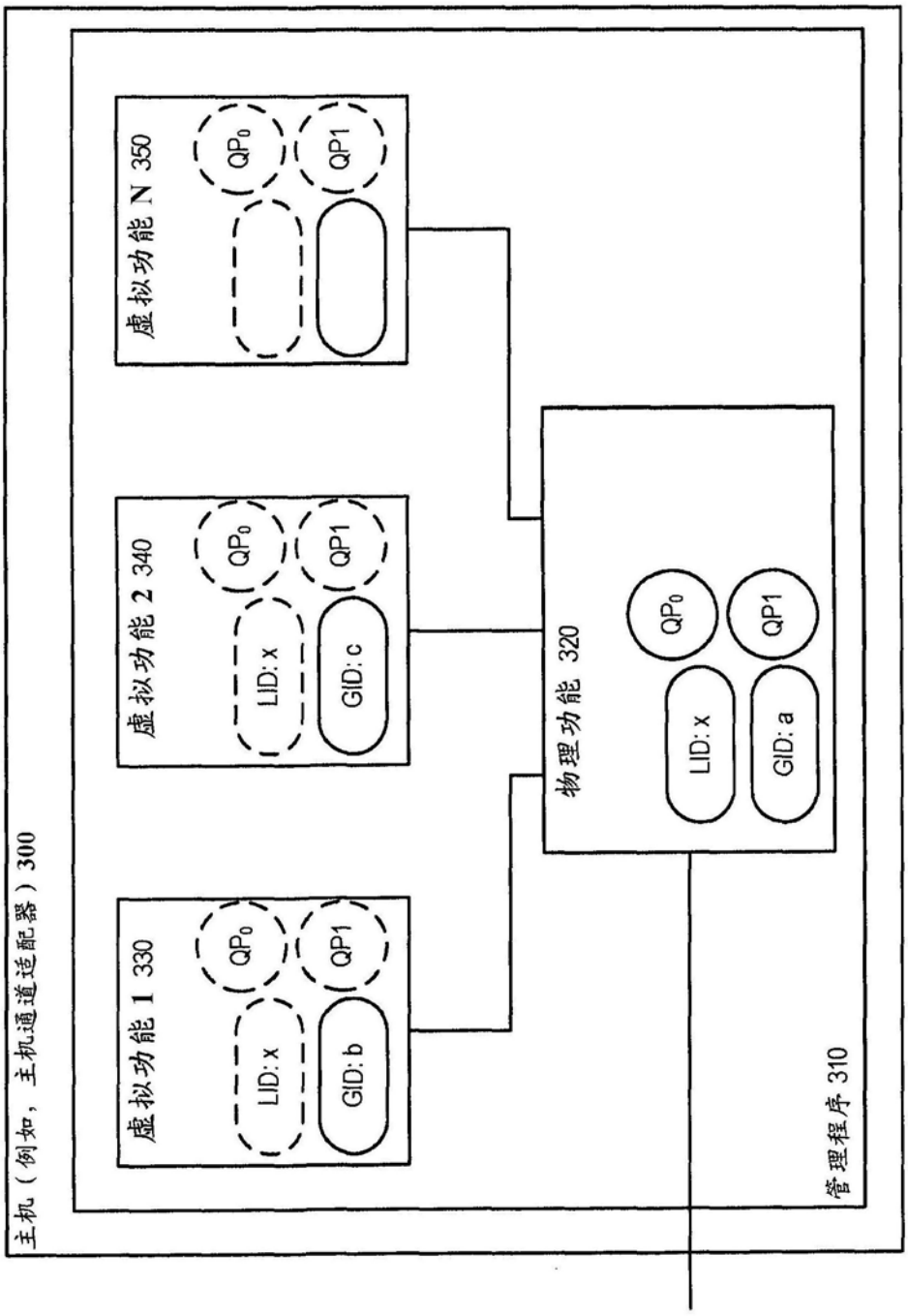

主机（例如，主机通道适配器）300
虚拟功能 1 330
QP0
QP1
LID: x
GID: b
虚拟功能 2 340
QP0
QP1
LID: x
GID: c
虚拟功能 N 350
QP0
QP1
物理功能 320
QP0
QP1
LID: x
GID: a
管理程序 310

图4

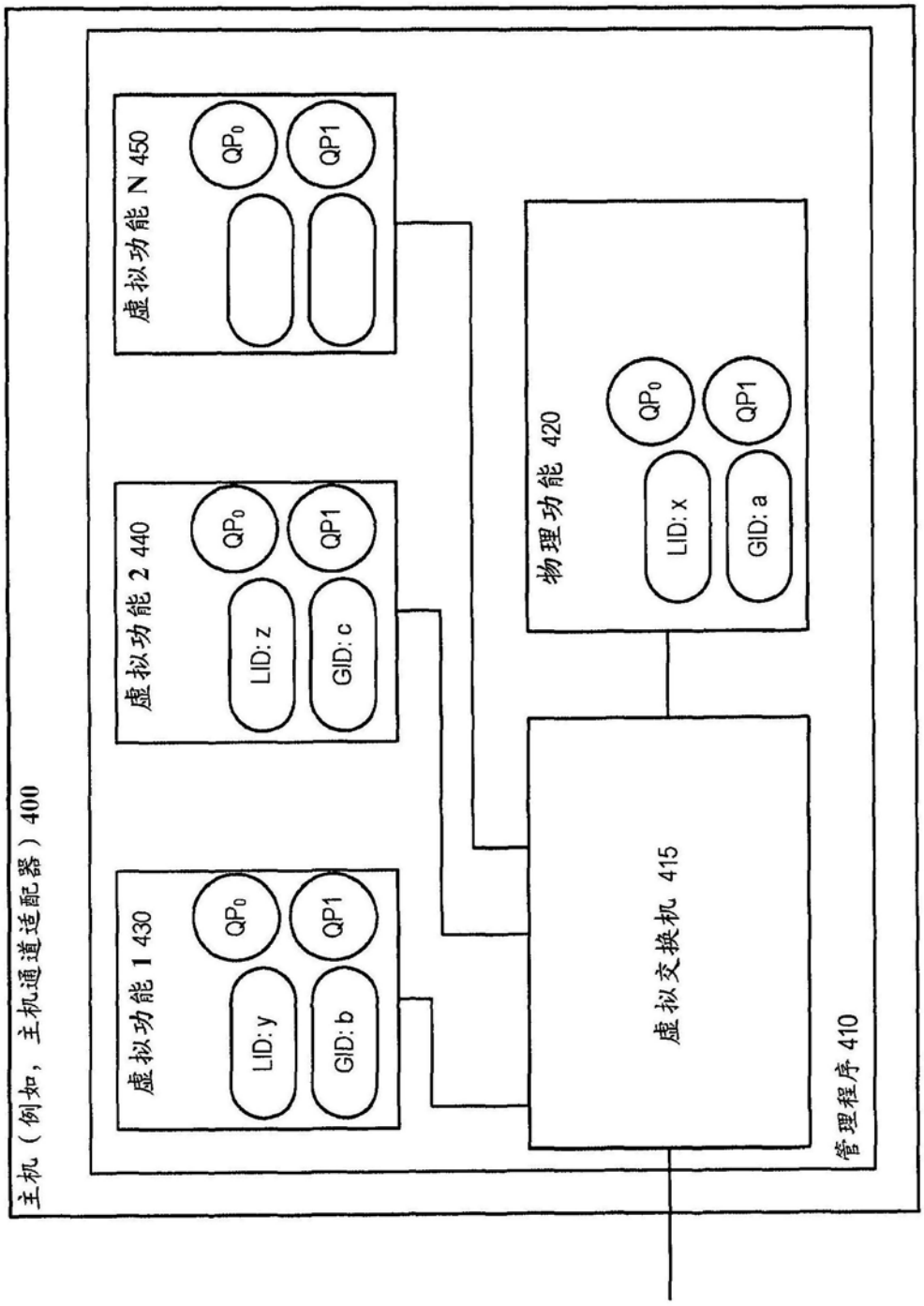

主机（例如，主机通道适配器）400
虚拟功能 1 430
QP0
QP1
LID: y
GID: b
虚拟功能 2 440
QP0
QP1
LID: z
GID: c
虚拟功能 N 450
QP0
QP1
物理功能 420
QP0
QP1
LID: x
GID: a
虚拟交换机 415
管理程序 410

图5

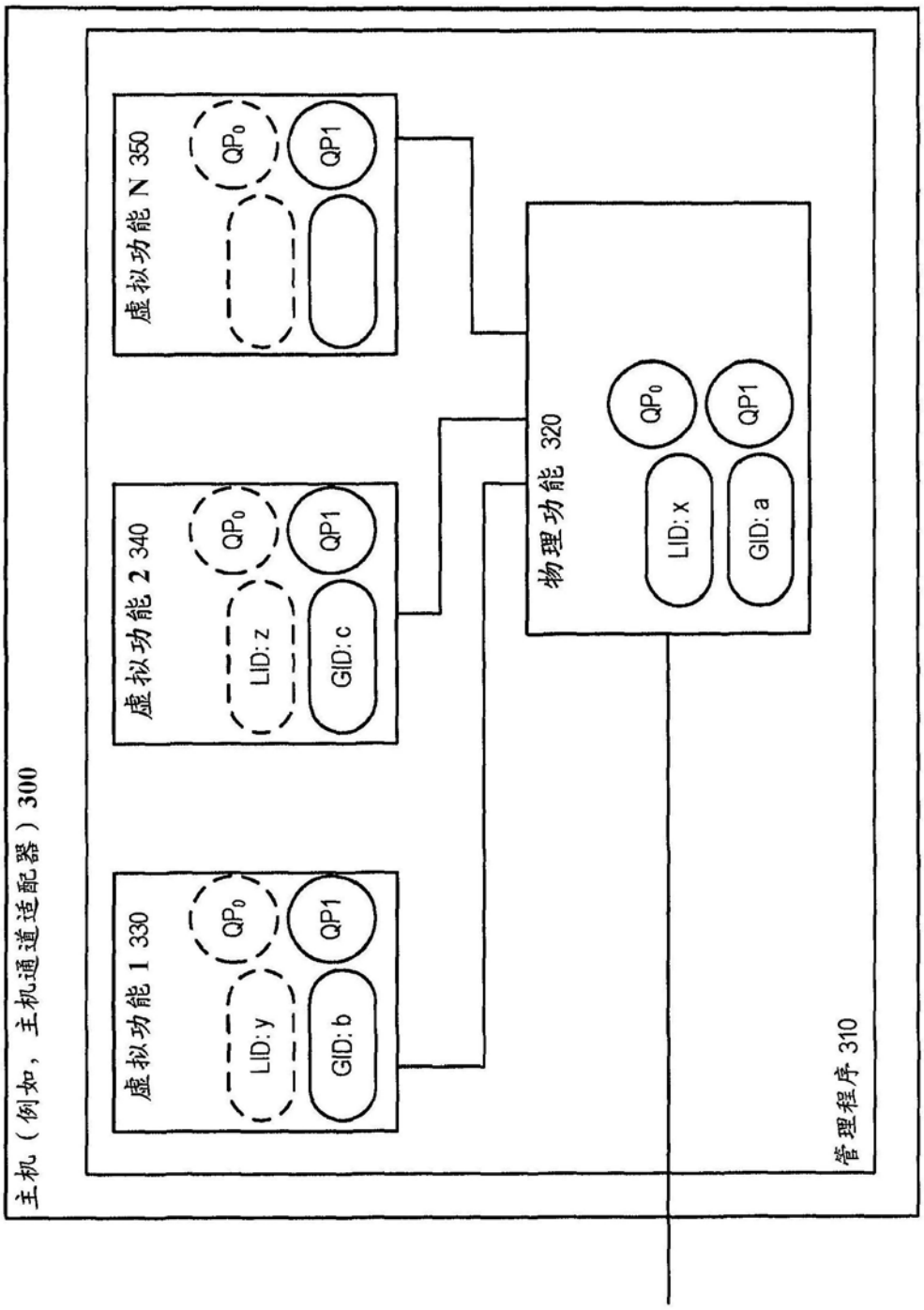

主机（例如，主机通道适配器）300
虚拟功能 1 330
LID: y
QP0
GID: b
QP1
虚拟功能 2 340
LID: z
QP0
GID: c
QP1
虚拟功能 N 350
QP0
QP1
物理功能 320
LID: x
QP0
GID: a
QP1
管理程序 310

图6

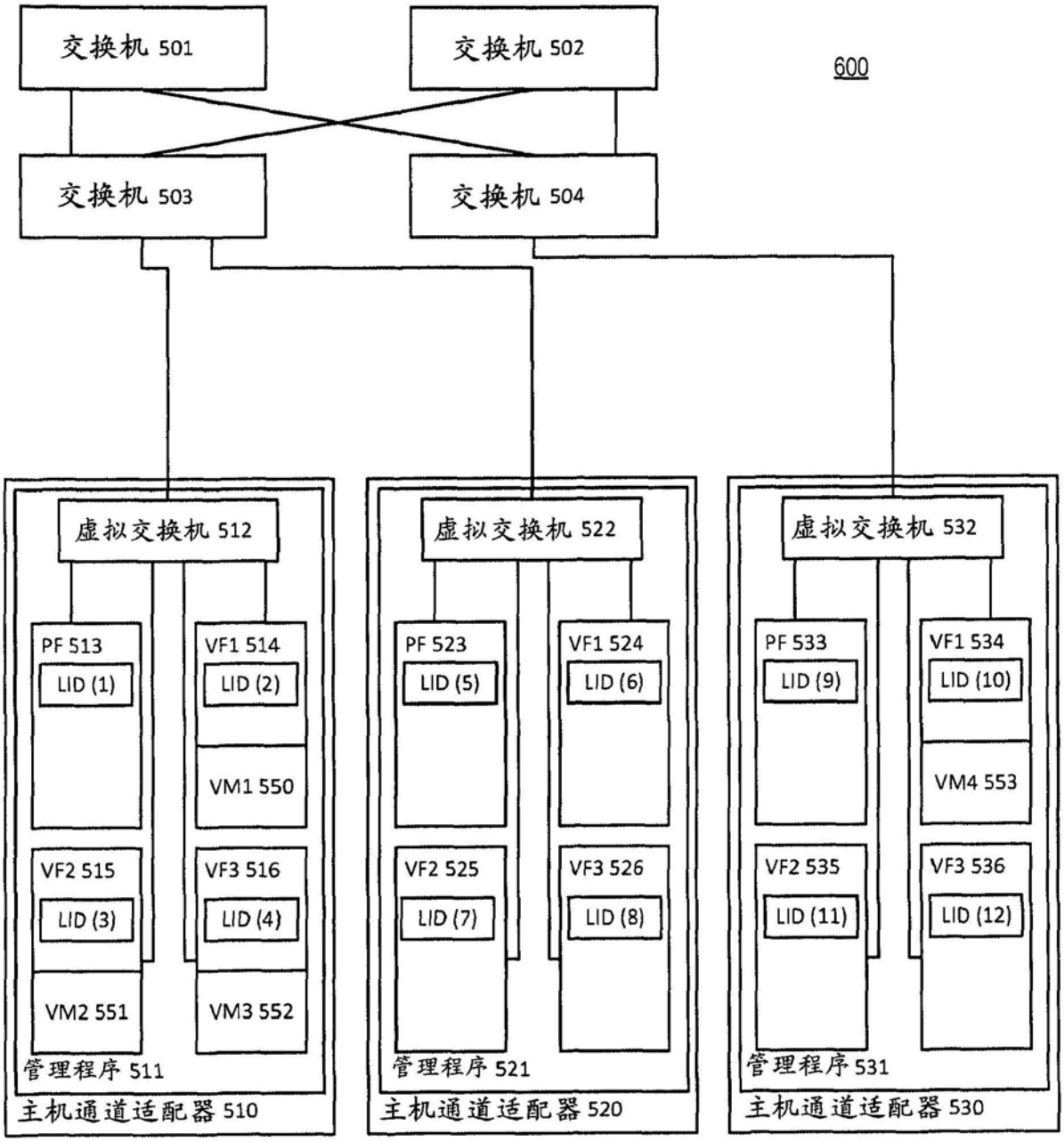

交换机 501
交换机 502
600
交换机 503
交换机 504
虚拟交换机 512
PF 513
LID (1)
VF1 514
LID (2)
VM1 550
VF2 515
LID (3)
VF3 516
LID (4)
VM2 551
VM3 552
管理程序 511
主机通道适配器 510
虚拟交换机 522
PF 523
LID (5)
VF1 524
LID (6)
VF2 525
LID (7)
VF3 526
LID (8)
管理程序 521
主机通道适配器 520
虚拟交换机 532
PF 533
LID (9)
VF1 534
LID (10)
VM4 553
VF2 535
LID (11)
VF3 536
LID (12)
管理程序 531
主机通道适配器 530

图7

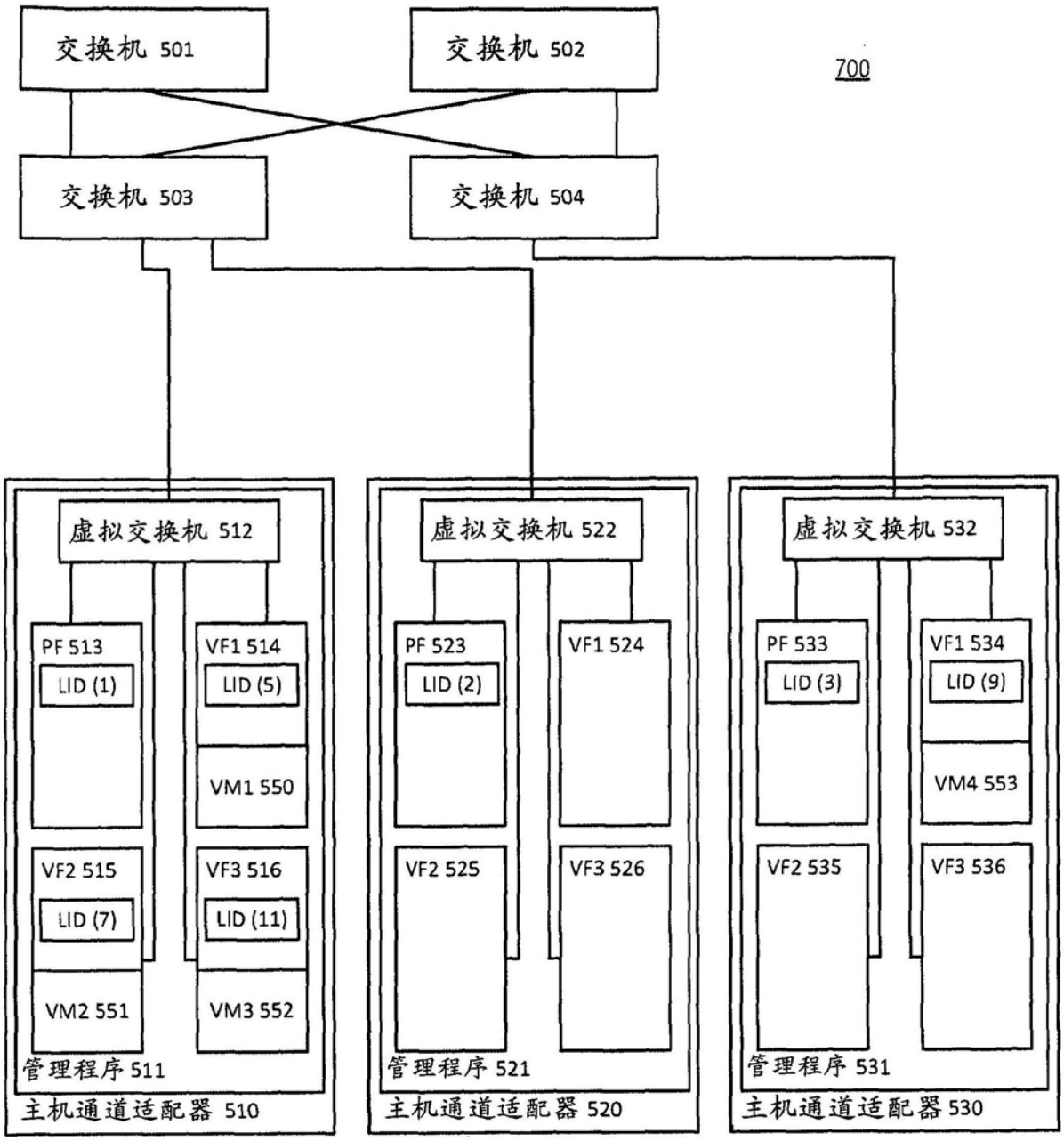

交换机 501
交换机 502
700
交换机 503
交换机 504
虚拟交换机 512
PF 513
LID (1)
VF1 514
LID (5)
VM1 550
VF2 515
LID (7)
VF3 516
LID (11)
VM2 551
VM3 552
管理程序 511
主机通道适配器 510
虚拟交换机 522
PF 523
LID (2)
VF1 524
VF2 525
VF3 526
管理程序 521
主机通道适配器 520
虚拟交换机 532
PF 533
LID (3)
VF1 534
LID (9)
VM4 553
VF2 535
VF3 536
管理程序 531
主机通道适配器 530

图8

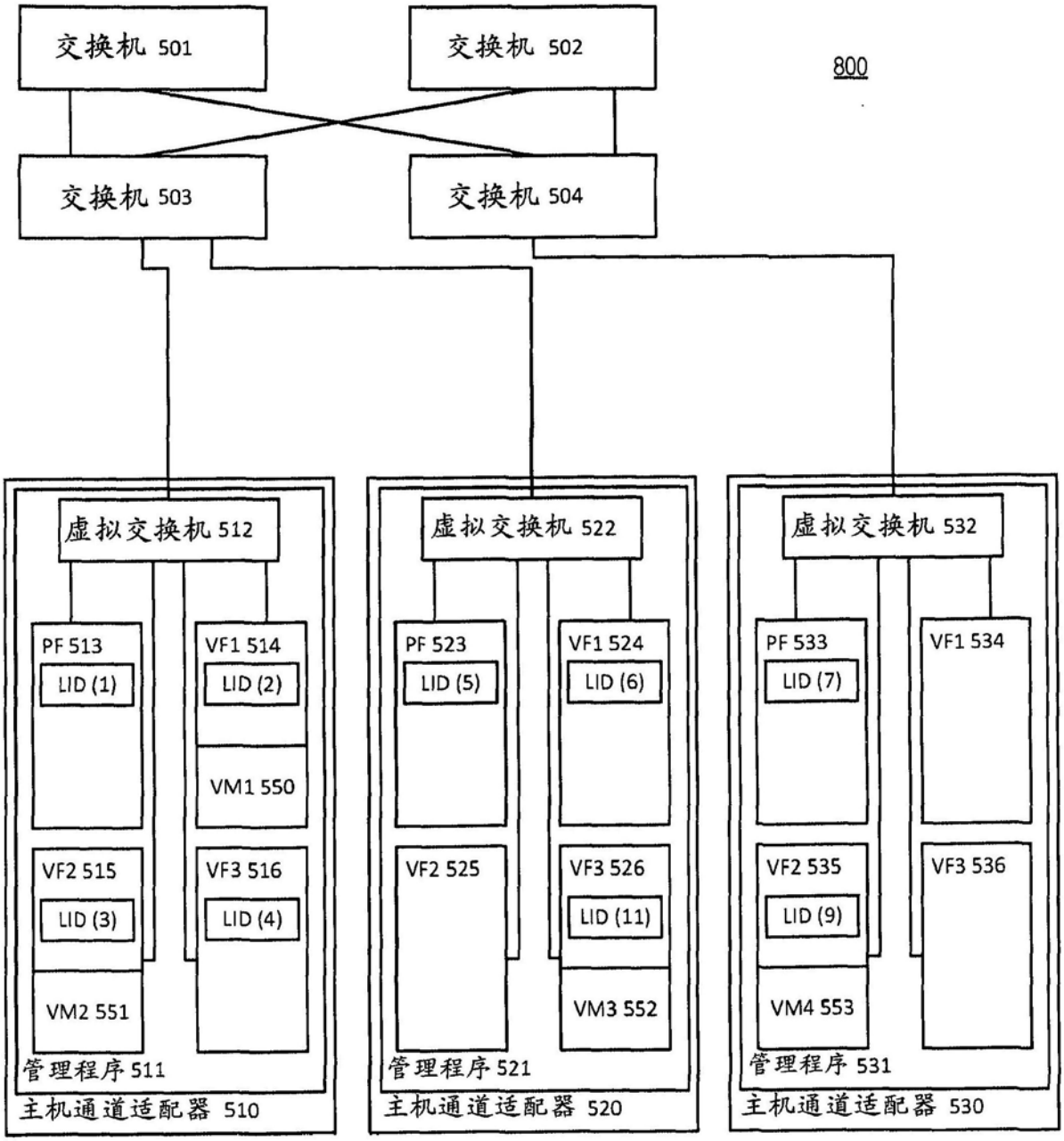

交换机 501
交换机 502
800
交换机 503
交换机 504
虚拟交换机 512
PF 513
LID (1)
VF1 514
LID (2)
VM1 550
VF2 515
LID (3)
VF3 516
LID (4)
VM2 551
管理程序 511
主机通道适配器 510
虚拟交换机 522
PF 523
LID (5)
VF1 524
LID (6)
VF2 525
VF3 526
LID (11)
VM3 552
管理程序 521
主机通道适配器 520
虚拟交换机 532
PF 533
LID (7)
VF1 534
VF2 535
LID (9)
VF3 536
VM4 553
管理程序 531
主机通道适配器 530

图9

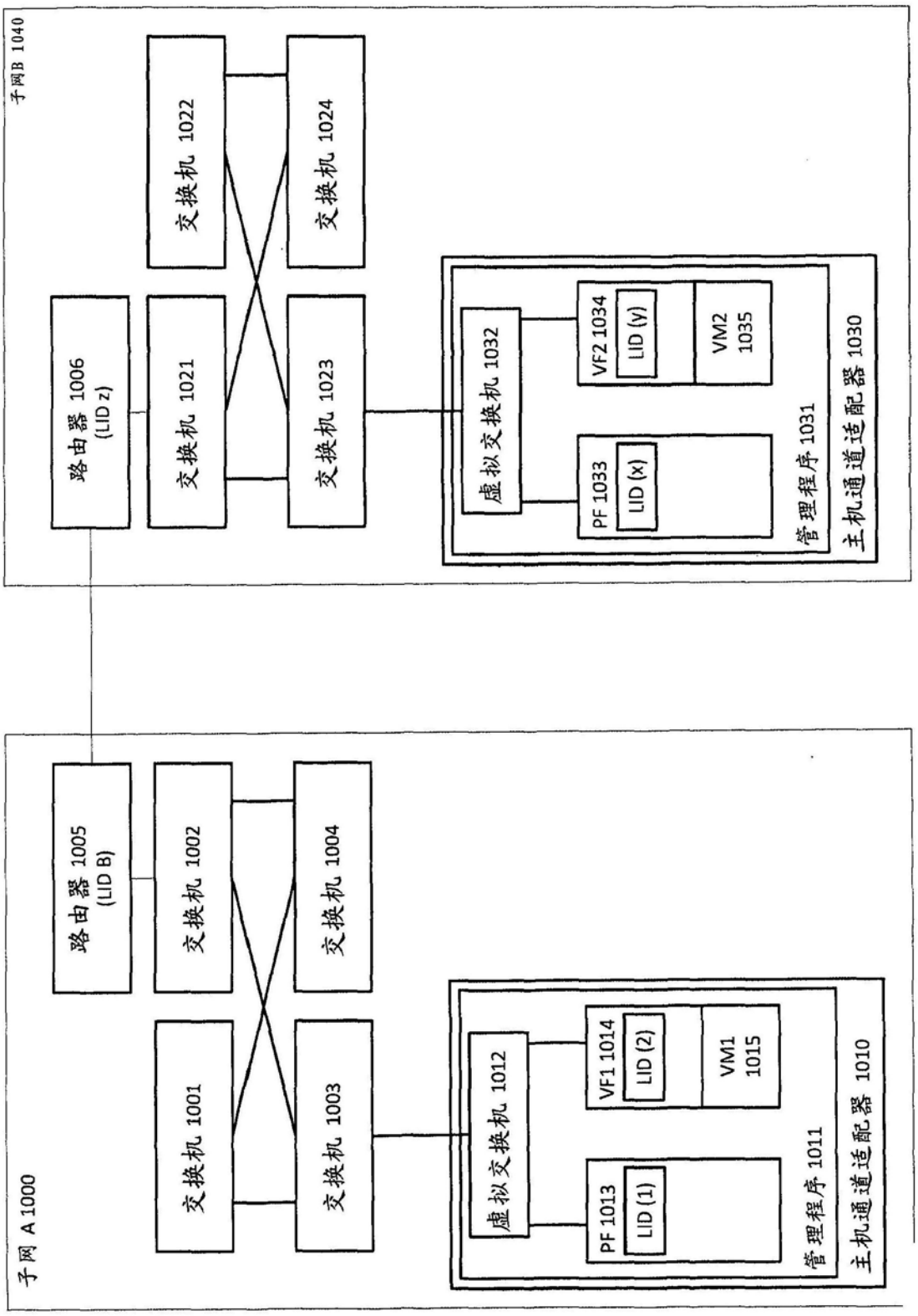

子网B 1040
路由器 1006
(LID z)
交换机 1021
交换机 1022
交换机 1023
交换机 1024
虚拟交换机 1032
VF2 1034
LID (y)
VM2
1035
PF 1033
LID (x)
管理程序 1031
主机通道适配器 1030
子网 A 1000
路由器 1005
(LID B)
交换机 1001
交换机 1002
交换机 1003
交换机 1004
虚拟交换机 1012
VF1 1014
LID (2)
VM1
1015
PF 1013
LID (1)
管理程序 1011
主机通道适配器 1010

图10

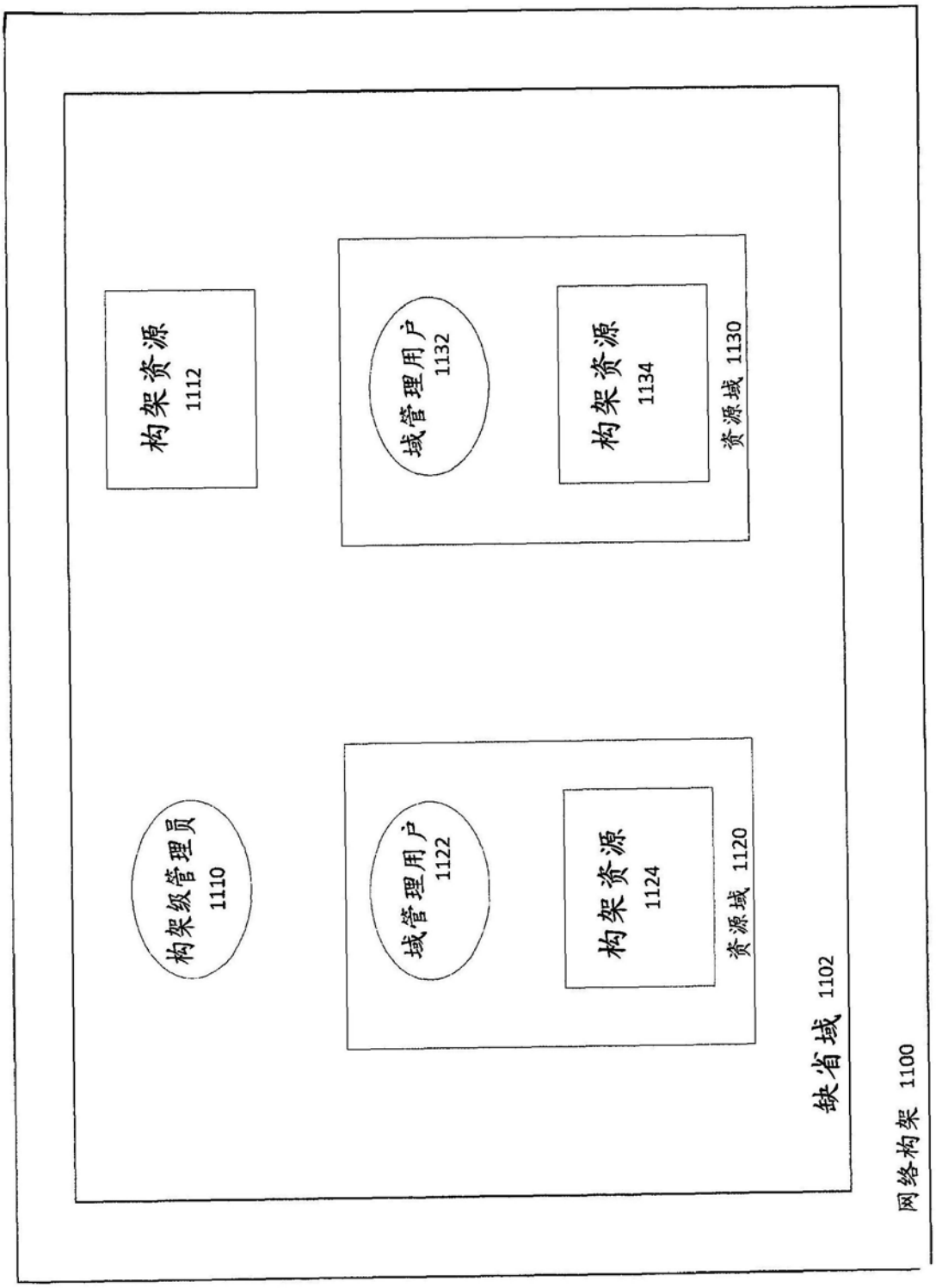

网络构架 1100
缺省域 1102
构架级管理员 1110
域管理用户 1122
构架资源 1124
资源域 1120
构架资源 1112
域管理用户 1132
构架资源 1134
资源域 1130

图11

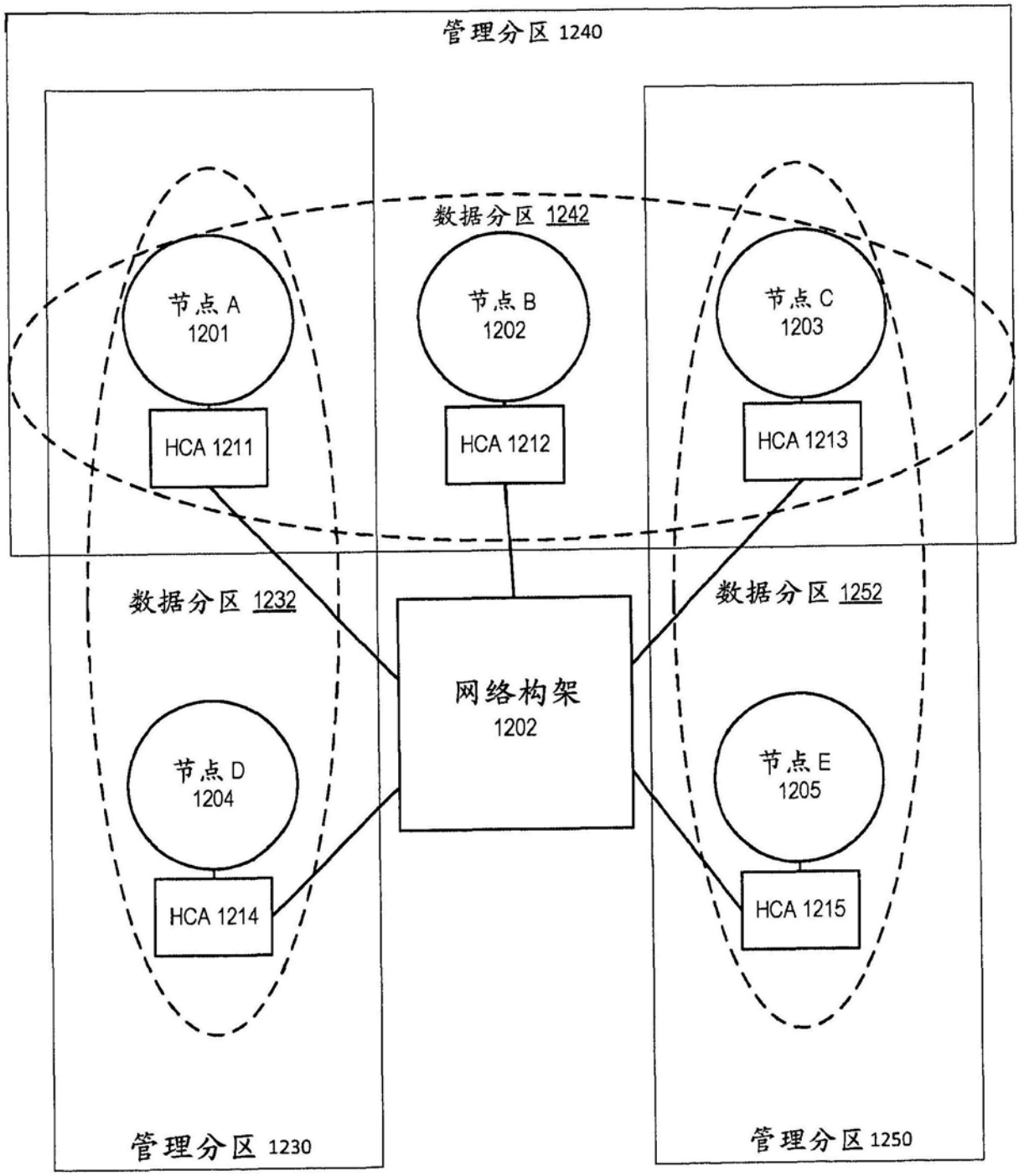

管理分区 1240
数据分区 1242
节点 A
1201
节点 B
1202
节点 C
1203
HCA 1211
HCA 1212
HCA 1213
数据分区 1232
数据分区 1252
网络构架
1202
节点 D
1204
节点 E
1205
HCA 1214
HCA 1215
管理分区 1230
管理分区 1250

图12

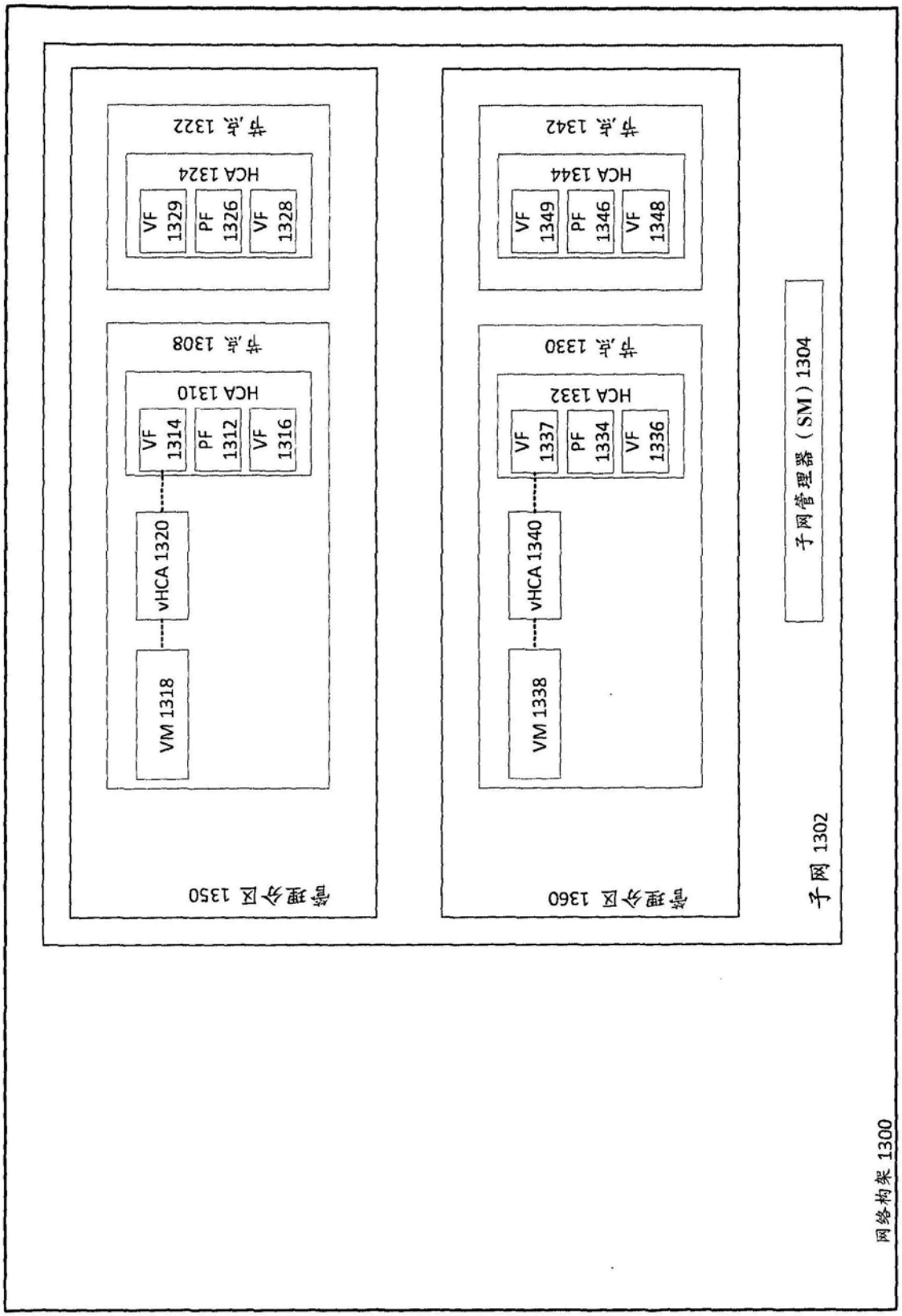

网络构架 1300
子网 1302
子网管理器（SM）1304
管理分区 1350
节点 1308
HCA 1310
VF 1314
PF 1312
VF 1316
vHCA 1320
VM 1318
节点 1322
HCA 1324
VF 1329
PF 1326
VF 1328
管理分区 1360
节点 1330
HCA 1332
VF 1337
PF 1334
VF 1336
vHCA 1340
VM 1338
节点 1342
HCA 1344
VF 1349
PF 1346
VF 1348

图13

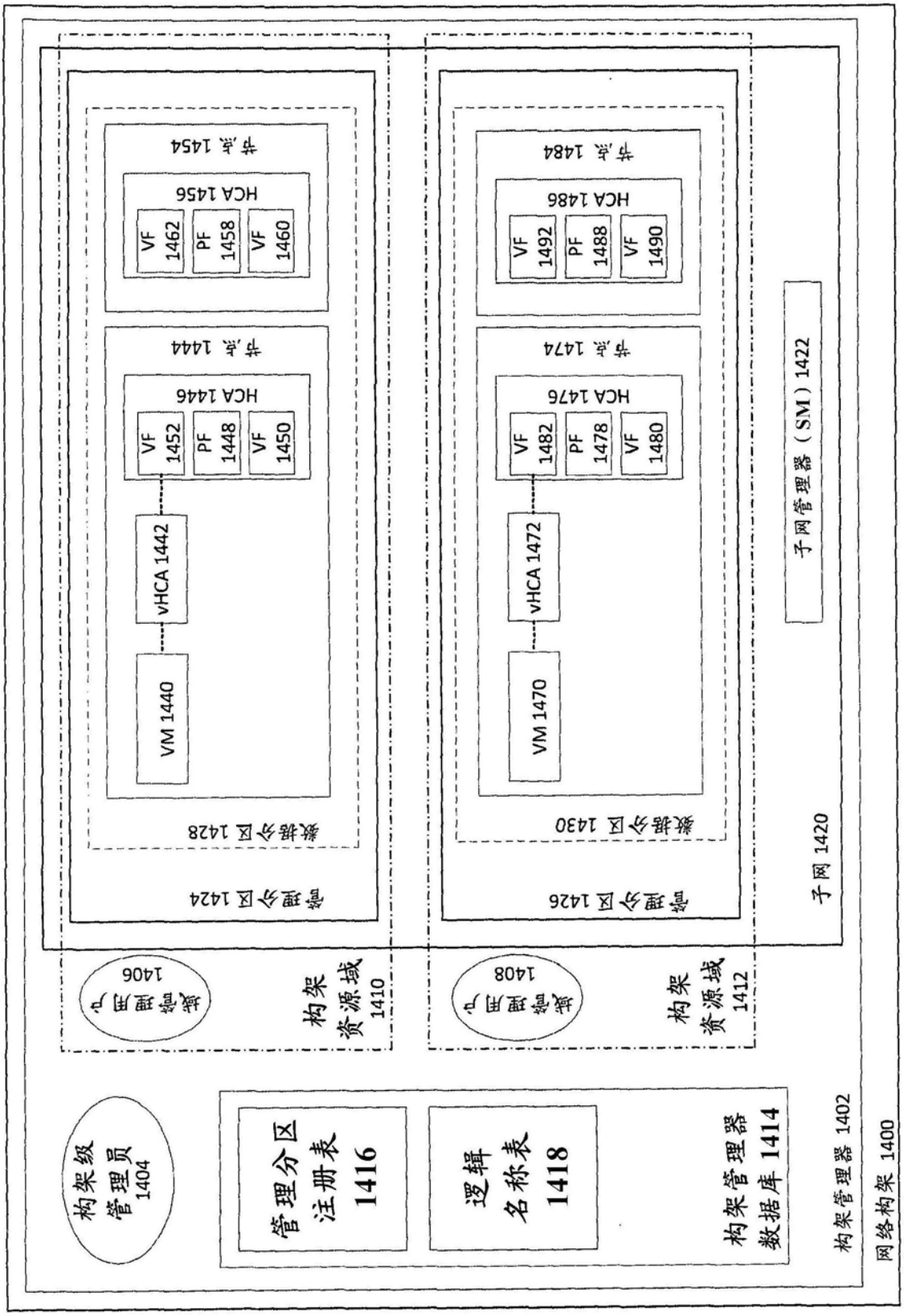

网络构架 1400
构架管理器 1402
子网 1420
子网管理器（SM）1422
构架资源域 1410
构架资源域 1412
构架管理员 1404
管理分区注册表 1416
逻辑名称表 1418
构架管理器数据库 1414
虚拟机 1440
vHCA 1442
HCA 1446
VF 1452
PF 1448
VF 1450
虚拟机 1444
HCA 1456
VF 1462
PF 1458
VF 1460
虚拟机 1454
虚拟机 1470
vHCA 1472
HCA 1476
VF 1482
PF 1478
VF 1480
虚拟机 1474
HCA 1486
VF 1492
PF 1488
VF 1490
虚拟机 1484

图14

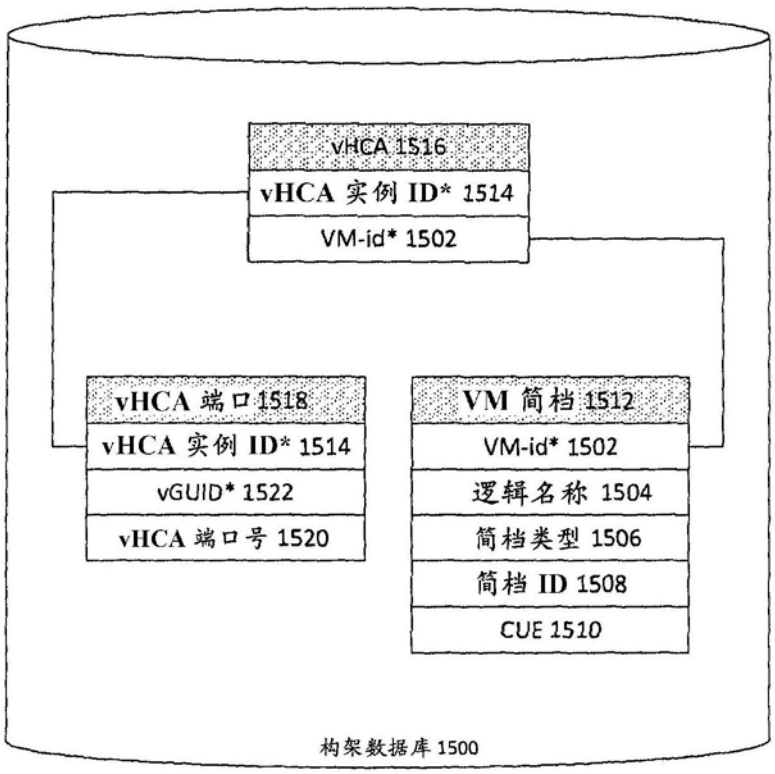

vHCA 1516
vHCA 实例 ID* 1514
VM-id* 1502
vHCA 端口 1518
vHCA 实例 ID* 1514
vGUID* 1522
vHCA 端口号 1520
VM 简档 1512
VM-id* 1502
逻辑名称 1504
简档类型 1506
简档 ID 1508
CUE 1510
构架数据库 1500

图15

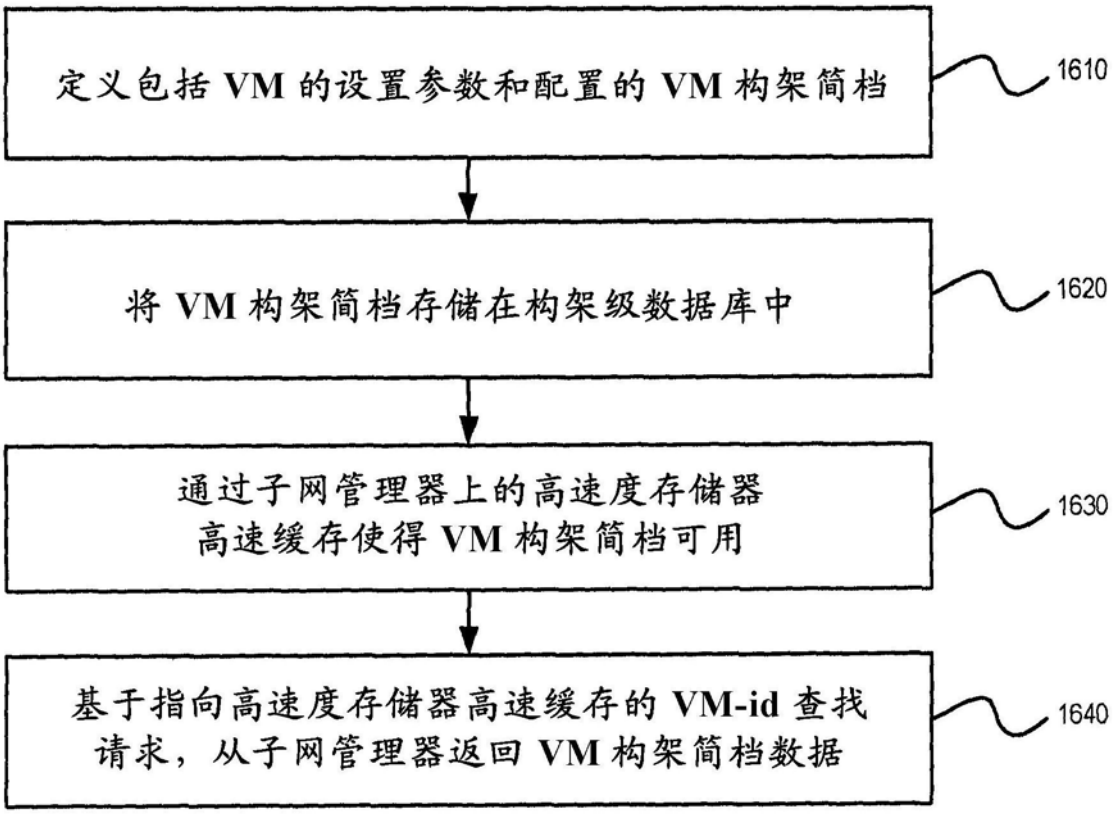

定义包括 VM 的设置参数和配置的 VM 构架简档
1610
将 VM 构架简档存储在构架级数据库中
1620
通过子网管理器上的高速度存储器
高速缓存使得 VM 构架简档可用
1630
基于指向高速度存储器高速缓存的 VM-id 查找
请求，从子网管理器返回 VM 构架简档数据
1640

图16

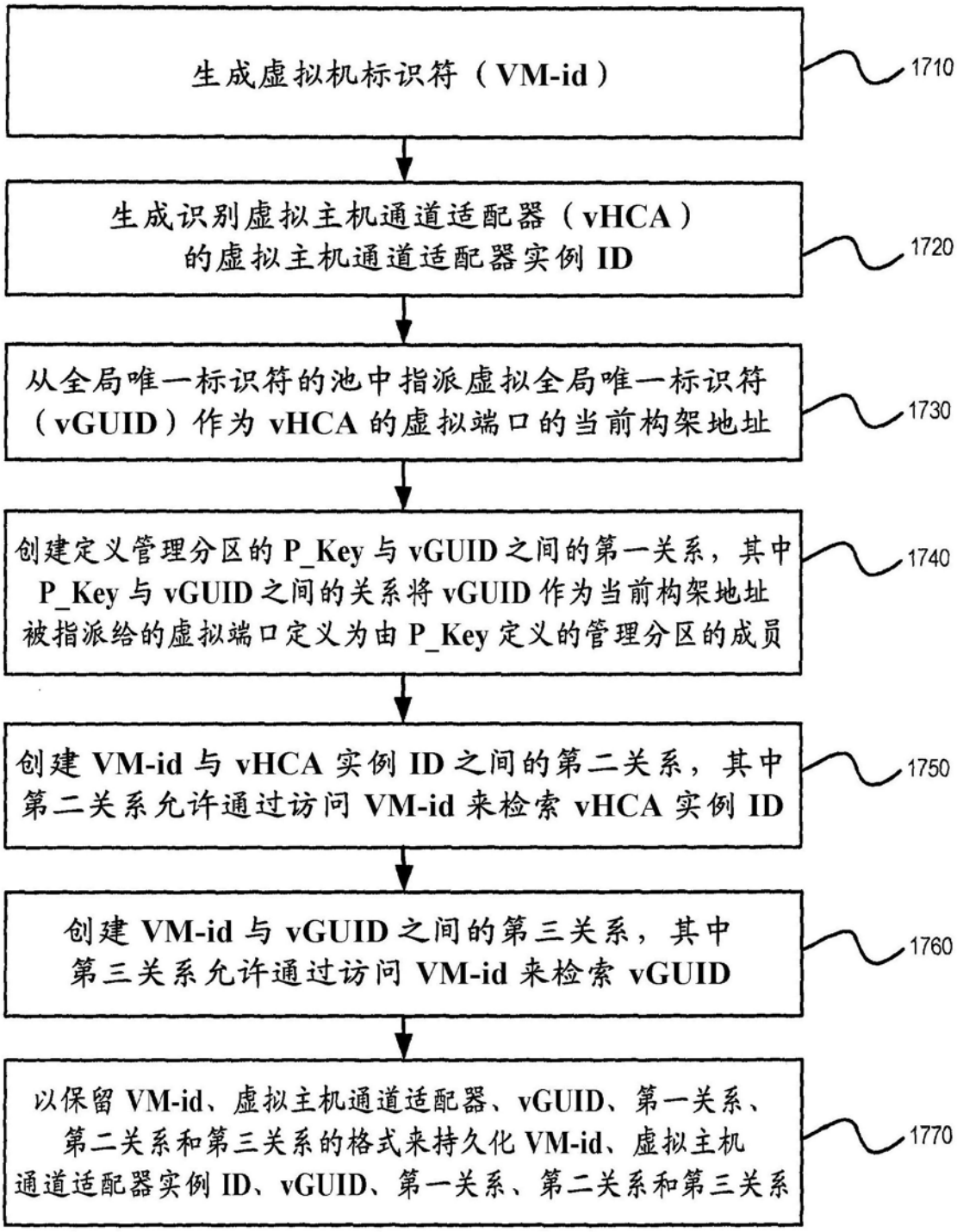

生成虚拟机标识符（VM-id）
1710
生成识别虚拟主机通道适配器（vHCA）的虚拟主机通道适配器实例 ID
1720
从全局唯一标识符的池中指派虚拟全局唯一标识符（vGUID）作为 vHCA 的虚拟端口的当前构架地址
1730
创建定义管理分区的 P_Key 与 vGUID 之间的第一关系，其中 P_Key 与 vGUID 之间的关系将 vGUID 作为当前构架地址被指派给的虚拟端口定义为由 P_Key 定义的管理分区的成员
1740
创建 VM-id 与 vHCA 实例 ID 之间的第二关系，其中第二关系允许通过访问 VM-id 来检索 vHCA 实例 ID
1750
创建 VM-id 与 vGUID 之间的第三关系，其中第三关系允许通过访问 VM-id 来检索 vGUID
1760
以保留 VM-id、虚拟主机通道适配器、vGUID、第一关系、第二关系和第三关系的格式来持久化 VM-id、虚拟主机通道适配器实例 ID、vGUID、第一关系、第二关系和第三关系
1770

图17