(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 107925634 A
(43)申请公布日 2018.04.17

(21)申请号 201780002644.6

(22)申请日 2017.01.26

(30)优先权数据
62/287,720 2016.01.27 US
15/413,152 2017.01.23 US

(85)PCT国际申请进入国家阶段日
2018.02.02

(86)PCT国际申请的申请数据
PCT/US2017/015158 2017.01.26

(87)PCT国际申请的公布数据
WO2017/132393 EN 2017.08.03

(71)申请人 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 B·博格丹斯基 L·霍雷恩

(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 张鑫

(51)Int.Cl.
H04L 12/931(2006.01)
H04L 29/06(2006.01)

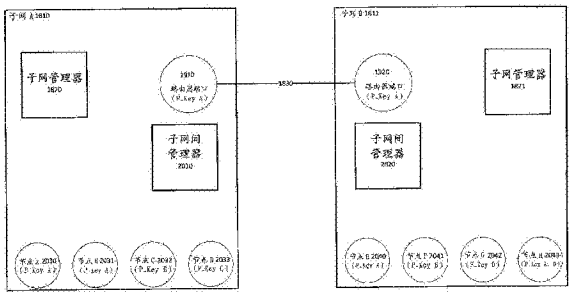
权利要求书5页 说明书23页 附图21页

(54)发明名称

用于支持高性能计算环境中的子网间分区的系统和方法

(57)摘要

用于支持高性能计算环境中的子网间分区的系统和方法。根据实施例，架构管理器可以在多个P_Key值当中将P_Key值的范围定义为子网间分区(ISP)P_Key范围。架构管理器可以经由它们的子网管理器将这个定义的P_Key值的范围传送给多个子网。每个子网中的子网管理器保持经其子网的管理。由于不存在配置每一侧子网间通信的中央管理器，因此参与的子网内的子网管理器可以建立ISP成员资格，并且然后与另一个子网交换信息。



CN 107925634 A

1.一种用于支持高性能计算环境中的子网间分区的系统,包括:

一个或多个微处理器;

第一子网,所述第一子网包括:

多个交换机,所述多个交换机至少包括叶交换机,其中所述多个交换机中的每一个包括多个交换机端口,

多个主机通道适配器,每个主机通道适配器包括至少一个主机通道适配器端口,

多个端节点,其中每个端节点与所述多个主机通道适配器中的至少一个主机通道适配器相关联,

子网管理器,所述子网管理器在所述多个交换机和所述多个主机通道适配器中的一个上运行,以及

子网间管理器,所述子网间管理器在所述多个交换机和所述

多个主机通道适配器中的一个上运行;

架构管理器,所述架构管理器驻留在高性能计算环境中;

其中所述多个交换机中的交换机上的所述多个交换机端口中的交换机端口被配置为路由器端口;

其中配置为路由器端口的交换机端口被逻辑地连接到虚拟路由器;

其中所述架构管理器定义多个分区键值内的分区键(P_Key)值的范围,所定义的分区键值的范围包括子网间分区(ISP)P_Key范围;以及

其中所述架构管理器向所述子网管理器传送ISP P_Key范围。

2.如权利要求1所述的系统,其中所述子网管理器将ISP P_Key范围内的P_Key分配给配置为路由器端口的交换机端口。

3.如权利要求1或2所述的系统,还包括:

第二子网,所述第二子网包括:

多个第二子网交换机,所述多个第二子网交换机至少包括第二子网叶交换机,其中所述多个第二子网交换机中的每一个包括多个第二子网交换机端口,

多个第二子网主机通道适配器,每个第二子网主机通道适配器包括至少一个第二子网主机通道适配器端口,

多个第二子网端节点,其中每个第二子网端节点与所述多个第二子网主机通道适配器中的至少一个第二子网主机通道适配器相关联,

第二子网子网管理器,所述第二子网子网管理器在所述多个第二子网交换机和所述多个第二子网主机通道适配器中的一个上运行,以及

第二子网子网间管理器,所述第二子网子网间管理器在所述多个第二子网交换机和所述多个第二子网主机通道适配器中的一个上运行;

其中另外多个第二子网交换机中的交换机上的所述多个第二子网交换机端口中的第二子网交换机端口被配置为第二子网路由器端口;

其中被配置为第二子网路由器端口的第二子网交换机端口逻辑地连接到第二子网虚拟路由器,第二子网虚拟路由器包括至少两个第二子网虚拟路由器端口;

其中第一子网经由物理链路与第二子网互连;以及

其中所述架构管理器向所述第二子网子网管理器传送ISP P_Key范围。

4.如权利要求3所述的系统，其中所述第二子网子网管理器将ISP P_Key范围内的第二P_Key分配给配置为第二子网路由器端口的第二子网交换机端口。

5.如权利要求3或4所述的系统，其中所述子网间管理器通过物理链路与第二子网子网间管理器交换信息，所述信息包括分配给路由器端口的P_Key；以及

其中所述第二子网子网间管理器通过物理链路与所述子网间管理器交换关于第二子网的信息，所述关于第二子网的信息包括分配给第二子网路由器端口的第二P_Key。

6.如权利要求5所述的系统，其中在由所述第二子网子网管理器交换信息时，所述子网管理器确定分配给路由器端口的P_Key和分配给第二子网路由器端口的第二P_Key是相同的。

7.如权利要求6所述的系统，其中在确定分配给路由器端口的P_Key和分配给第二子网路由器端口的第二P_Key相同时，所述子网间管理器通过第二子网路由器端口启用与第二子网的数据通信。

8.一种用于支持高性能计算环境中的子网间分区的方法，包括：

在包括一个或多个微处理器的一个或多个计算机处提供，

多个交换机，所述多个交换机至少包括叶交换机，其中所述多个交换机中的每一个包括多个交换机端口，

多个主机通道适配器，每个主机通道适配器包括至少一个主机通道适配器端口，

多个端节点，其中每个端节点与所述多个主机通道适配器中的至少一个主机通道适配器相关联，

子网管理器，所述子网管理器在所述多个交换机和所述多个主机通道适配器中的一个上运行，以及

子网间管理器，所述子网间管理器在所述多个交换机和所述多个主机通道适配器中的一个上运行，

提供架构管理器，所述架构管理器驻留在高性能计算环境中；

将所述多个交换机中的交换机上的所述多个交换机端口中的交换机端口配置为路由器端口；

将配置为路由器端口的交换机端口逻辑地连接到虚拟路由器，所述虚拟路由器包括至少两个虚拟路由器端口；

由所述架构管理器定义多个分区键值内的分区键(P_Key)值的范围，所定义的分区键值的范围包括子网间分区(ISP)P_Key范围；以及

由架构管理器将ISP P_Key范围传送给所述子网管理器。

9.如权利要求8所述的方法，还包括：

由所述子网管理器将ISP P_Key范围内的P_Key分配给配置为路由器端口的交换机端口。

10.如权利要求8或9所述的方法，还包括：

在包括所述一个或多个微处理器的所述一个或多个计算机处还提供，

第二子网，所述第二子网包括：

多个第二子网交换机，所述多个第二子网交换机至少包括第二子网叶交换机，其中所述多个第二子网交换机中的每一个包括多个第二子网交换机端口，

多个第二子网主机通道适配器，每个第二子网主机通道适配器包括至少一个第二子网主机通道适配器端口，

多个第二子网端节点，其中每个第二子网端节点与所述多个第二子网主机通道适配器中的至少一个第二子网主机通道适配器相关联，

第二子网子网管理器，所述第二子网子网管理器在所述多个第二子网交换机和所述多个第二子网主机通道适配器中的一个上运行，以及

第二子网子网间管理器，所述第二子网子网间管理器在所述多个第二子网交换机和所述多个第二子网主机通道适配器中的一个上运行；

将另外多个第二子网交换机中的交换机上的所述多个第二子网交换机端口中的第二子网交换机端口配置为第二子网路由器端口；以及

由所述架构管理器将ISP P_Key范围传送给所述第二子网子网管理器；

其中被配置为第二子网路由器端口的第二子网交换机端口逻辑地连接到第二子网虚拟路由器，第二子网虚拟路由器包括至少两个第二子网虚拟路由器端口；以及

其中第一子网经由物理链路与第二子网互连。

11.如权利要求10所述的方法，还包括：

由所述第二子网子网管理器将ISP P_Key范围内的第二P_Key分配给被配置为第二子网路由器端口的第二子网交换机端口。

12.如权利要求10或11所述的方法，还包括：

由所述子网间管理器通过物理链路与所述第二子网子网间管理器交换信息，所述信息包括分配给路由器端口的P_Key；以及

由所述第二子网子网间管理器通过物理链路与所述子网间管理器交换关于第二子网的信息，所述关于第二子网的信息包括分配给第二子网路由器端口的第二P_Key。

13.如权利要求12所述的方法，还包括：

在由所述第二子网子网间管理器交换信息时，所述子网管理器确定分配给路由器端口的P_Key和分配给第二子网路由器端口的第二P_Key是相同的。

14.如权利要求13所述的方法，还包括：

在确定分配给路由器端口的P_Key和分配给第二子网路由器端口的第二P_Key相同时，所述子网管理器通过第二子网路由器端口启用与第二子网的数据通信。

15.一种非瞬态计算机可读存储介质，包括存储在其上的用于支持高性能计算环境中的子网间分区的指令，所述指令在由一个或多个计算机读取并执行时使所述一个或多个计算机执行包括以下的步骤：

在包括一个或多个微处理器的一个或多个计算机处提供，

多个交换机，所述多个交换机至少包括叶交换机，其中所述多个交换机中的每一个包括多个交换机端口，

多个主机通道适配器，每个主机通道适配器包括至少一个主机通道适配器端口，

多个端节点，其中每个端节点与所述多个主机通道适配器中的至少一个主机通道适配器相关联，

子网管理器，所述子网管理器在所述多个交换机和所述多个主机通道适配器中的一个上运行，以及

子网间管理器，所述子网间管理器在所述多个交换机和所述多个主机通道适配器中的一个上运行；

提供架构管理器，所述架构管理器驻留在高性能计算环境中；

将所述多个交换机中的交换机上的所述多个交换机端口中的交换机端口配置为路由器端口；

将配置为路由器端口的交换机端口逻辑地连接到虚拟路由器，所述虚拟路由器包括至少两个虚拟路由器端口；

由所述架构管理器定义多个分区键值内的分区键(P_Key)值的范围，所定义的分区键值的范围包括子网间分区(ISP)P_Key范围；以及

由架构管理器将ISP P_Key范围传送给所述子网管理器。

16.如权利要求15所述的非瞬态计算机可读存储介质，所述步骤还包括：

由所述子网管理器将ISP P_Key范围内的P_Key分配给配置为路由器端口的交换机端口。

17.如权利要求15或16所述的非瞬态计算机可读存储介质，所述步骤还包括：

在包括所述一个或多个微处理器的所述一个或多个计算机处还提供，

第二子网，所述第二子网包括：

多个第二子网交换机，所述多个第二子网交换机至少包括第二子网叶交换机，其中所述多个第二子网交换机中的每一个包括多个第二子网交换机端口，

多个第二子网主机通道适配器，每个第二子网主机通道适配器包括至少一个第二子网主机通道适配器端口，

多个第二子网端节点，其中每个第二子网端节点与所述多个第二子网主机通道适配器中的至少一个第二子网主机通道适配器相关联，

第二子网子网管理器，所述第二子网子网管理器在所述多个第二子网交换机和所述多个第二子网主机通道适配器中的一个上运行，以及

第二子网子网间管理器，所述第二子网子网间管理器在所述多个第二子网交换机和所述多个第二子网主机通道适配器中的一个上运行；

将另外多个第二子网交换机中的交换机上的所述多个第二子网交换机端口中的第二子网交换机端口配置为第二子网路由器端口；以及

由所述架构管理器将ISP P_Key范围传送给所述第二子网子网管理器；

其中被配置为第二子网路由器端口的第二子网交换机端口逻辑地连接到第二子网虚拟路由器，第二子网虚拟路由器包括至少两个第二子网虚拟路由器端口；以及

其中第一子网经由物理链路与第二子网互连。

18.如权利要求17所述的非瞬态计算机可读存储介质，所述步骤还包括：

由所述第二子网子网管理器将ISP P_Key范围内的第二P_Key分配给被配置为第二子网路由器端口的第二子网交换机端口。

19.如权利要求17或18所述的非瞬态计算机可读存储介质，所述步骤还包括：

由所述子网间管理器通过物理链路与所述第二子网子网间管理器交换信息，所述信息包括分配给路由器端口的P_Key；以及

由所述第二子网子网间管理器通过物理链路与所述子网间管理器交换关于第二子网

的信息，所述关于第二子网的信息包括分配给第二子网路由器端口的第二P_Key。

20.如权利要求19所述的非瞬态计算机可读存储介质，所述步骤还包括：

在由所述第二子网子网间管理器交换信息时，所述子网管理器确定分配给路由器端口的P_Key和分配给第二子网路由器端口的第二P_Key是相同的；以及

在确定分配给路由器端口的P_Key和分配给第二子网路由器端口的第二P_Key相同时，所述子网管理器通过第二子网路由器端口启用与第二子网的数据通信。

21.一种计算机程序，包括机器可读格式的程序指令，所述程序指令在由计算机系统执行时使所述计算机系统执行如权利要求8至14中任一项所述的方法。

22.一种计算机程序产品，包括存储在非瞬态机器可读数据存储介质中的如权利要求21所述的计算机程序。

用于支持高性能计算环境中的子网间分区的系统和方法

[0001] 版权声明

[0002] 本专利文档公开内容的一部分包含受版权保护的素材。版权拥有者不反对任何人对专利文档或专利公开内容按照在专利商标局的专利文件或记录中出现得那样进行的传真复制，但是除此之外在任何情况下都保留所有版权。

技术领域

[0003] 本发明一般而言涉及计算机系统，并且特别地涉及支持高性能计算环境中的子网间分区。

背景技术

[0004] 随着更大的云计算体系架构的推出，与传统网络和存储相关联的性能和管理瓶颈已成为重要的问题。人们对使用诸如InfiniBand(IB)技术等高性能无损互连作为云计算架构的基础越来越感兴趣。这是本发明的实施例旨在解决的一般领域。

发明内容

[0005] 本文描述的是用于支持高性能计算环境中的子网间分区的方法和系统。示例性方法可以在包括一个或多个微处理器的一个或多个计算机处提供第一子网，第一子网包括：多个交换机，该多个交换机至少包括叶交换机，其中该多个交换机中的每个交换机包括多个交换机端口；多个主机通道适配器，每个主机通道适配器包括至少一个主机通道适配器端口；多个端节点，其中端节点中的每个端节点与多个主机通道适配器中的至少一个主机通道适配器相关联，以及子网管理器，该子网管理器在多个交换机和多个主机通道适配器中的一个上运行。该方法可以在包括一个或多个微处理器的一个或多个计算机处提供架构管理器，该架构管理器驻留在高性能计算环境中。该方法可以将多个交换机中的交换机上的多个交换机端口中的交换机端口配置为路由器端口。该方法可以逻辑上将配置为路由器端口的交换机端口连接到虚拟路由器，虚拟路由器包括至少两个虚拟路由器端口。该方法可以由架构管理器定义多个分区键值内的分区键(P_Key)值的范围，所定义的分区键值的范围包括子网间分区(ISP)P_Key范围。该方法可以由架构管理器将ISP P_Key范围传送给子网管理器。

[0006] 根据实施例，(第一子网或第二子网中的任一个的)多个主机通道适配器中的一个或多个可以包括至少一个虚拟功能、至少一个虚拟交换机和至少一个物理功能。(第一子网或第二子网的)多个端节点可以包括物理主机、虚拟机或者物理主机与虚拟机的组合，其中虚拟机与至少一个虚拟功能相关联。

附图说明

[0007] 图1示出了根据实施例的InfiniBand环境的图示。

[0008] 图2示出了根据实施例的分区集群环境的图示。

[0009] 图3示出了根据实施例的网络环境中的树形拓扑的图示。

[0010] 图4示出了根据实施例的示例性共享端口体系架构。

[0011] 图5示出了根据实施例的示例性vSwitch体系架构。

[0012] 图6示出了根据实施例的示例性vPort体系架构。

[0013] 图7示出了根据实施例的具有预填充的LID的示例性vSwitch体系架构。

[0014] 图8示出了根据实施例的具有动态LID分配的示例性vSwitch体系架构。

[0015] 图9示出了根据实施例的具有动态LID分配和预填充的LID的具有vSwitch的示例性vSwitch体系架构。

[0016] 图10示出了根据实施例的示例性多子网InfiniBand架构。

[0017] 图11示出了根据实施例的在高性能计算环境中的两个子网之间的互连。

[0018] 图12示出了根据实施例的在高性能计算环境中经由双端口虚拟路由器配置的两个子网之间的互连。

[0019] 图13示出了根据实施例的用于支持高性能计算环境中的双端口虚拟路由器的方法的流程图。

[0020] 图14是图示根据实施例的DR路由的VSMP分组的流程图。

[0021] 图15是图示根据实施例的对DR路由的VSMP分组的响应的流程图。

[0022] 图16是图示根据实施例的LID路由的VSMP分组的流程图。

[0023] 图17是图示根据实施例的对LID路由的VSMP分组的响应的流程图。

[0024] 图18是示出根据实施例的子网间分区P_Key范围的定义的框图。

[0025] 图19是根据实施例的支持子网间分区的框图。

[0026] 图20是根据实施例的支持子网间分区的框图。

[0027] 图21是根据实施例的用于支持子网间分区的示例性方法的流程图。

具体实施方式

[0028] 在附图的各图中通过示例而非限制的方式图示了本发明，附图中相同的标号指示类似的元件。应当注意的是，在本公开中对“一个”或“一些”实施例的引用不一定是相同的实施例，并且这种引用意味着至少一个。虽然讨论了特定的实现，但是应当理解的是，特定实现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，可以使用其它部件和配置。

[0029] 贯穿附图和具体实施方式可以使用共同的标号来指示相同的元素；因此，如果元素在其它地方进行了描述，那么在图中使用的标号可以或可以不在特定于该图的具体描述中引用。

[0030] 本文描述的是支持高性能计算环境中的子网间分区的系统和方法。

[0031] 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规范或传统IB规范)。这样的参考被理解为是指可在http://www.inifinibandta.org获得的于2015年3月发布的**InfiniBand®**Trade Association Architecture Specification，卷1，版本1.3，其全部内容通过引用被结合于此。对于本领域技术人员来说显而易见的是，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架构拓扑的示例。对于本领域

技术人员来说显而易见的是，可以使用其它类型的架构拓扑而没有限制。

[0032] 为了满足当前时代(例如，Exascale时代)的云的需求，期望虚拟机能够利用诸如远程直接存储器访问(RDMA)的低开销网络通信范例。RDMA绕过OS堆栈并且直接与硬件通信，因此可以使用像单根I/O虚拟化(SR-IOV)网络适配器这样的直通技术。根据实施例，对于高性能无损互连网络中的适用性可以提供虚拟交换机(vSwitch)SR-IOV体系架构。由于网络重新配置时间对于使实时迁移成为实用选项是至关重要的，因此，除了网络体系架构之外，还可以提供可伸缩的和拓扑无关的动态重新配置机制。

[0033] 根据实施例，并且除此之外，可以提供使用vSwitch的虚拟化环境的路由策略，并且可以提供用于网络拓扑(例如，胖树拓扑)的高效路由算法。动态重新配置机制可以被进一步微调以使胖树中施加的开销最小化。

[0034] 根据本发明的实施例，虚拟化可以有益于云计算中的高效资源利用和弹性资源分配。实时迁移使得有可能通过以应用透明的方式在物理服务器之间移动虚拟机(VM)来优化资源使用。因此，虚拟化可以通过实时迁移实现整合、资源的按需供给以及弹性。

[0035] InfiniBand™

[0036] 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，特别针对高性能计算(HPC)应用和数据中心。

[0037] InfiniBand™体系架构(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。在上层，IB架构构成可以使用路由器互连的一个或多个子网。

[0038] 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)，其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)可以负责在IB架构中执行路由表计算。这里，例如，IB网络的路由旨在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。

[0039] 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架构、配置端节点和交换机，并从SMA接收通知。

[0040] 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线性转发表(LFT)。LFT由SM根据使用中的路由机制来计算。在子网中，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)进行寻址。线性转发表(LFT)中的每个条目包括目的地LID(DLID)和输出端口。表中只支持每LID一个条目。当分组到达交换机时，其输出端口通过在交换机的转发表中查找DLID来确定。路由是确定性的，因为分组在网络中在给定的源-目的地对(LID对)之间采用相同的路径。

[0041] 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起作用。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相应地重新配置网络。

[0042] 此外，可以使用本地标识符(LID)来寻址子网内的主机和交换机，并且可以将单个子网限制为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位全局唯一标识符(GUID)。GUID可以用于形成作为IB层三(L3)地址的全局标识符(GID)。

[0043] 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描以检查拓扑变化。如果在轻扫描期间发现变化,或者如果SM接收到通过发信号通知网络变化的信息(陷阱),那么SM可以根据发现的变化来重新配置网络。

[0044] 例如,当网络拓扑改变时,诸如当链路断开时、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其中发生网络变化的子网的局部范围。此外,用路由器对大型架构进行分段可以限制重新配置的范围。

[0045] 图1中示出了示例InfiniBand架构,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架构120经由相应的主机通道适配器111-115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-E、101-105)可以由诸如虚拟机的各种虚拟设备来表示。

[0046] 在InfiniBand中分区

[0047] 根据实施例,IB网络可以支持分区作为安全机制,以提供对共享网络架构的系统的逻辑组的隔离。架构中的节点上的每个HCA端口可以是一个或多个分区的成员。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分区成员资格信息配置为16位分区键(P_Key)的表。SM还可以用包含与通过这些端口发送或接收数据流量的端节点相关联的P_Key信息的分区实施表来配置交换机和路由器端口。此外,在一般情况下,交换机端口的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。

[0048] 根据实施例,分区是端口的逻辑组,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,可以使用分区成员资格信息对分组进行过滤以实施隔离。一旦分组到达传入端口,具有无效分区信息的分组就可以被丢弃。在分区的IB系统中,可以使用分区来创建租户集群。在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,即使存在受损或恶意的租户节点时,系统的安全性也能够得到保证。

[0049] 根据实施例,对于节点之间的通信,除管理队列对(QP0和QP1)以外,队列对(QP)和端到端上下文(EEC)可以被分配给特定分区。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配置的表验证其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,只有在共享分区的端口之间才允许通信。

[0050] 图2中示出了IB分区的示例,其中示出了根据实施例的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架构120经由相应的主机通道适配器111-115通信。节点A-E被布置到分区中,即分区1,130、分区2,140和分区3,150。分区1包括节点A 101和节点D 104。分区2包括节点A 101、节点B 102和节点C 103。分区3包括节点C

103和节点E 105。由于分区的布置，节点D 104和节点E 105不被允许通信，因为这些节点不共享分区。同时，例如，节点A 101和节点C 103被允许通信，因为这些节点都是分区2，140的成员。

[0051] InfiniBand中的虚拟机

[0052] 在过去的十年中，虚拟化高性能计算(HPC)环境的前景已得到相当大的提高，因为CPU开销已通过硬件虚拟化支持被实际上消除；存储器开销已通过虚拟化存储器管理单元被显著降低；存储开销已通过使用快速SAN存储装置或分布式网络文件系统被减少；并且网络I/O开销已通过使用比如单根输入/输出虚拟化(SR-IOV)的设备直通技术被减少。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，并提供必要的性能。

[0053] 但是，当与诸如InfiniBand(IB)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HPC和其它通信密集型工作负载。

[0054] 用于将IB设备连接到VM的传统方法是通过利用具有直接分配的SR-IOV。但是，使用SR-IOV实现分配有IB主机通道适配器(HCA)的VM的实时迁移已被证明是具有挑战性的。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(VM-in-migration)通信的其它节点会丢失连接性。当发生这种情况时，通过向IB子网管理器(SM)发送子网管理(SA)路径记录查询来定位要重新连接到的虚拟机的新地址，可以尝试更新丢失的连接。

[0055] IB使用三种不同类型的地址。第一种类型的地址是16位本地标识符(LID)。SM向每个HCA端口和每个交换机分配至少一个唯一的LID。LID用于在子网内路由流量。由于LID为16位长，因此可以做出65536个唯一地址组合，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分配给每个设备(例如，HCA和交换机)和每个HCA端口的64位全局唯一标识符(GUID)。SM可以向HCA端口分配附加的子网唯一GUID，其在使用SR-IOV时是有用的。第三种类型的地址是128位全局标识符(GID)。GID是有效的IPv6单播地址，并且向每个HCA端口分配至少一个。GID通过组合由架构管理器分配的全局唯一64位前缀和每个HCA端口的GUID地址形成。

[0056] 胖树(FTree)拓扑和路由

[0057] 根据实施例，基于IB的HPC系统中的一些采用胖树拓扑来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，当树朝着拓扑的根移动时，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。

[0058] 图3示出了根据实施例的网络环境中的树形拓扑结构的图示。如图3所示，一个或多个端节点201-204可以在网络架构200中连接。网络架构200可以基于包括多个叶交换机211-214和多个主干交换机或根交换机231-234的胖树拓扑。此外，网络架构200可以包括一个或多个中间交换机，诸如交换机221-224。

[0059] 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架构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。

[0060] 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。

[0061] 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。胖树路由机制也在OFED(开放结构企业分发-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。

[0062] 胖树路由机制旨在生成在网络架构中跨链路均匀传播最短路径路由的LFT。该机制按索引次序遍历架构并将端节点的目标LID(以及因此对应的路由)分配给每个交换机端口。对于连接到相同叶交换机的端节点，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。

[0063] 根据实施例，在分区的子网中，不允许不是公共分区的成员的节点通信。在实践中，这意味着由胖树路由算法分配的一些路由不用于用户流量。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那些路由生成LFT时，会出现该问题。由于节点按索引的次序进行路由，因此这种行为会导致链路上的平衡降级。由于路由可以在与分区无关的情况下执行，因此，一般而言，胖树路由的子网提供分区之间较差的隔离。

[0064] 根据实施例，胖树是可以利用可用网络资源进行伸缩的分层网络拓扑。而且，使用放置在不同级别层次上的商用交换机容易构建胖树。通常可以获得胖树的不同变体，包括k-ary-n-trees、扩展的广义胖树(XGFT)、平行端口广义胖树(PGFT)和现实生活胖树(RLFT)。

[0065] k-ary-n-tree是具有k^n个端节点和$n \cdot k^{n-1}$个交换机的n级胖树，每个具有2k个端口。每个交换机在树中具有相同数量的上下连接。XGFT胖树通过允许交换机不同数量的上行连接和下行连接以及在树中每个级别的不同数量的连接来扩展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。

[0066] 输入/输出(I/O)虚拟化

[0067] 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。存储流量和服务器间通信的组合强加了可能淹没单个服务器的I/O资源的增加的负载，从而导致积压和由于处理器在等待数据而导致的空闲处理器。随着I/O请求数量的增加，IOV可以提供可用性；并且可以提高(虚拟化)I/O资源的性能、可伸缩性和灵活性，以匹配现代CPU虚拟化中所看到的性能水平。

[0068] 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将暴露于VM的逻辑设备与其物理实现分离。当前，可以存在不同类型的IOV技术，诸如仿真、半虚拟化、直接分配(DA)和单根I/O虚拟化(SR-IOV)。

[0069] 根据实施例，一种类型的IOV技术是软件仿真。软件仿真可以允许分离的前端/后端软件体系架构。前端可以是置于VM中、与由管理程序实现的以提供I/O访问的后端通信的设备驱动程序。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、非期望的计算开销。

[0070] 根据实施例，另一种类型的IOV技术是直接设备分配。直接设备分配涉及将I/O设备耦合到VM，其中在VM之间没有设备共享。直接分配或设备直通以最小的开销提供接近本地性能。物理设备绕过管理程序并且直接附连到VM。但是，这种直接设备分配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。

[0071] 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。这些实例可以被分配给VM作为直通设备，并作为虚拟功能(VF)被访问。管理程序通过唯一的(每设备)、全特征物理功能(PF)访问设备。SR-IOV使纯直接分配的可伸缩性问题变得容易。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可以扩展PCI Express(PCIe)规范，其意味着允许从多个VM直接访问单个物理设备，同时维持接近本地性能。因此，SR-IOV可以提供良好的性能和可伸缩性。

[0072] SR-IOV允许PCIe设备通过向每个客户分配一个虚拟设备来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)，并被分配有唯一的请求者ID，其使得I/O存储器管理单元(IOMMU)能够区分来自/去往不同VF的流量。IOMMU还在PF和VF之间应用存储器和中断转换。

[0073] 但是，令人遗憾的是，直接设备分配技术在其中数据中心优化期望虚拟机的透明实时迁移的情况下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，因此其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。

[0074] 但是，当使用诸如SR-IOV的直接设备分配技术时，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，分配给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，该处理会引入在秒的数量级上的停机时间。而且，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中的附加开销并对底层网络架构的性能产生负面影响。

[0075] InfiniBand SR-IOV体系架构-共享端口

[0076] 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。

[0077] 图4示出了根据实施例的示例性共享端口体系架构。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分配给多个虚拟机。同样，物理功能可以由管理程序310处理。

[0078] 根据实施例，当使用诸如图4所示的共享端口体系架构时，主机(例如，HCA)在网络中出现为具有单个共享LID和在物理功能320和虚拟功能330、340、350之间的共享队列对(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。

[0079] 如图4所示，根据实施例，可以将不同的GID分配给虚拟功能和物理功能，并且特殊队列对QP0和QP1(即，用于InfiniBand管理分组的专用队列对)由物理功能拥有。这些QP也被暴露给VF，但是VF不允许使用QP0(从VF到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。

[0080] 根据实施例，共享端口体系架构可以允许不受(通过分配给虚拟功能附连到网络的)VM的数量限制的高度可伸缩的数据中心，因为LID空间仅被网络中的物理机器和交换机消耗。

[0081] 但是，共享端口体系架构的缺点是无法提供透明的实时迁移，从而阻碍了灵活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。

[0082] InfiniBand SR-IOV体系架构模型-虚拟交换机(vSwitch)

[0083] 图5示出了根据实施例的示例性vSwitch体系架构。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分配给多个虚拟机。同样，物理功能可以由管理程序410处理。虚拟交换机415也可以由管理程序401处理。

[0084] 根据实施例，在vSwitch体系架构中，每个虚拟功能430、440、450是完整的虚拟主机通道适配器(vHCA)，这意味着分配给VF的VM被分配完整的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看起来像经由虚拟交换机415、具有连接到它的附加节点的交换机。管理程序410可以使用PF 420，并且VM(附连到虚拟功能)使用VF。

[0085] 根据实施例，vSwitch体系架构提供透明的虚拟化。但是，由于每个虚拟功能都被分配唯一的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都有一个LID地址)的情况下，必须由SM计算更多的通信路径，并且必须将更多的子网管理分组(SMP)发送到交换机，以便更新其LFT。例如，通信路径的计算在大型网络中可能花费几分钟。因为LID空间限于49151个单播LID，并且由于每个VM(经由VF)、物理节点和交换机每个占用一个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。

[0086] InfiniBand SR-IOV体系架构模型-虚拟端口(vPort)

[0087] 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分配给多个虚拟机。同样，物理功能可以由管理程序310来处理。

[0088] 根据实施例，vPort概念被松散地定义以便赋予供应商实现的自由(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VM在子网中的处理方式标准化。利用vPort概念，可以定义可在空间和性能域都更可伸缩的类似SR-IOV的共享端口和类似vSwitch的体系架构或两者的组合。vPort支持可选的LID，并且与共享端口不同，即使vPort不使用专用LID，SM也知道子网中可用的所有vPort。

[0089] InfiniBand SR-IOV体系架构模型-具有预填充LID的vSwitch

[0090] 根据实施例，本公开提供了用于提供具有预填充的LID的vSwitch体系架构的系统和方法。

[0091] 图7示出了根据实施例的具有预填充的LID的示例性vSwitch体系架构。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)内提供在架构(诸如

InfiniBand架构)的成员之间的通信。架构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其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并将它们分配给多个虚拟机。例如,虚拟机1 550可以由管理程序511分配给虚拟功能1 514。管理程序511可以附加地将虚拟机2 551分配给虚拟功能2 515,并且将虚拟机3 552分配给虚拟功能3 516。管理程序531又可以将虚拟机4 553分配给虚拟功能1 534。管理程序可以通过主机通道适配器中的每一个上的全特征物理功能513、523、533访问主机通道适配器。

[0092] 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内的流量。

[0093] 根据实施例,虚拟交换机512、522和532可以由它们相应的管理程序511、521、531处理。在这样的vSwitch体系架构中,每个虚拟功能是完整的虚拟主机通道适配器(vHCA),这意味着分配给VF的VM被分配完整的IB地址(例如,GID、GUID、LID)集合和硬件中专用的QP空间。对于网络的其余部分和SM(未示出),HCA 510、520和530看起来像经由虚拟交换机、具有连接到它们的附加节点的交换机。

[0094] 根据实施例,本公开提供了用于提供具有预填充的LID的vSwitch体系架构的系统和方法。参考图7,LID被预填充到各种物理功能513、523、533以及虚拟功能514-516、524-526、534-536(甚至那些当前未与活动虚拟机相关联的虚拟功能)。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。当网络被引导时,LID在启用SR-IOV vSwitch的子网中被预填充。即使并非所有的VF都在网络中被VM占用,填充的VF也被分配有LID,如图7所示。

[0095] 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,并且经由一个、两个或更多个虚拟交换机连接到外部IB子网。

[0096] 根据实施例,在具有预填充LID的vSwitch体系架构中,每个管理程序可以通过PF为自己消耗一个LID,并为每个附加的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(16个虚拟功能中的每个虚拟功能一个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,极限)是46256个(2891个管理程序乘以每管理程序16个VF)。(实际上,这些数字实际上较小,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。

[0097] 根据实施例,在具有预填充LID的vSwitch体系架构中,网络第一次被引导时,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,在管理程序之一中定位用于VM的可用端口(即,可用的虚拟功能),并且将虚拟机附连到可用的虚拟功能。

[0098] 根据实施例,具有预填充LID的vSwitch体系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)的特征向一个物理机器提供替代路径，而不受需要LID必须是连续的LMC的限制约束。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LID尤其有用。

[0099] 根据实施例，以及以上示出的具有预填充LID的vSwitch体系架构的优点，可以考虑某些注意事项。例如，由于当网络被引导时，LID在启用SR-IOV vSwitch的子网中被预填充，因此初始路径计算(例如，在启动时)会比如果LID没有被预填充所花费的时间更长。

[0100] InfiniBand SR-IOV体系架构模型-具有动态LID分配的vSwitch

[0101] 根据实施例，本公开提供了用于提供具有动态LID分配的vSwitch体系架构的系统和方法。

[0102] 图8示出了根据实施例的具有动态LID分配的示例性vSwitch体系架构。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)内提供架构(诸如InfiniBand架构)的成员之间的通信。架构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其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并将它们分配给多个虚拟机。例如，虚拟机1 550可以由管理程序511分配给虚拟功能1 514。管理程序511可以附加地将虚拟机2 551分配给虚拟功能2 515，并且将虚拟机3 552分配给虚拟功能3 516。管理程序531又可以将虚拟机4 553分配给虚拟功能1 534。管理程序可以通过主机通道适配器中的每一个上的全特征物理功能513、523、533访问主机通道适配器。

[0103] 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内的流量。

[0104] 根据实施例，虚拟交换机512、522和532可以由它们相应的管理程序511、521、531处理。在这样的vSwitch体系架构中，每个虚拟功能是完整的虚拟主机通道适配器(vHCA)，这意味着分配给VF的VM被分配完整的IB地址(例如，GID、GUID、LID)集合和硬件中专用的QP空间。对于网络的其余部分和SM(未示出)，HCA 510、520和530看起来像经由虚拟交换机、具有连接到它们的附加节点的交换机。

[0105] 根据实施例，本公开提供了用于提供具有动态LID分配的vSwitch体系架构的系统和方法。参考图8，LID被动态分配给各种物理功能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分配的LID。例如，由于虚拟机1550是活动的并且与虚拟功能1 514相关联，因此虚拟功能514可以被分配LID 5。同样，虚拟功能2 515、虚拟功能3 516和虚拟功能1 534每个与活动虚拟功能相关联。由此，这些虚拟功能被分配LID，其中LID 7被分配给虚拟功能2 515、LID 11被分配给虚拟功能3 516、并且LID 9被分配给虚拟功能1 534。与具有预填充LID的vSwitch不同，那些当前未与活动虚拟机相关联的虚拟功能不接收LID分配。

[0106] 根据实施例，利用动态LID分配，可以显著减少初始的路径计算。当网络第一次被引导并且不存在VM时，那么可以使用相对较少数量的LID用于初始的路径计算和LFT分发。

[0107] 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，并且经由一个、两个或更多个虚拟交换机连接到外部IB子网。

[0108] 根据实施例，当在利用具有动态LID分配的vSwitch的系统中创建新的VM时，找到空闲的VM时隙，以便决定在哪个管理程序上引导新添加的VM，并且也找到唯一未使用的单播LID。但是，在网络中没有已知的路径和交换机的LFT用于处理新添加的LID。为了处理新添加的VM而计算一组新的路径在其中每分钟可以引导几个VM的动态环境中是非期望的。在大型IB子网中，计算一组新的路由会花费几分钟，并且这个过程将在每次引导新的VM时必须重复。

[0109] 有利地，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。只需要遍历网络中所有物理交换机的LFT、将转发端口从属于管理程序(其中创建VM)的PF的LID条目复制到新添加的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。

[0110] 根据实施例，在具有动态LID分配体系架构的vSwitch中分配的LID不一定是连续的。当将具有预填充LID的vSwitch中的每个管理程序上的VM上分配的LID与具有动态LID分配的vSwitch进行比较时，应当注意的是，在动态LID分配体系架构中分配的LID是非连续的，而被预填充的那些LID本质上是连续的。在vSwitch动态LID分配体系架构中，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LID的vSwitch中，每个VM都继承已经分配给对应VF的LID，并且在没有实时迁移的网络中，连续附连到给定VF的VM获得相同的LID。

[0111] 根据实施例，具有动态LID分配体系架构的vSwitch可以以一些附加的网络和运行时SM开销为代价，利用预填充LID体系架构模型来解决vSwitch的缺点。每次创建VM时，子网中的物理交换机的LFT用与创建的VM相关联的新添加的LID进行更新。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正在使用与其主机管理程序相同的路径，因此类似LMC的功能也不可用。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限制的数量。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增加了灾难恢复和分段网络优化的灵活性。

[0112] InfiniBand SR-IOV体系架构模型-具有动态LID分配和预填充LID的vSwitch

[0113] 图9示出了根据实施例的具有动态LID分配和预填充LID的具有vSwitch的示例性vSwitch体系架构。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)内提供架构(诸如InfiniBand架构)的成员之间的通信。架构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其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并将它们分配给多个虚拟机。例如，虚拟机1550可以由管理程序511分配给虚拟功能1 514。管理程序511可以附加地将虚拟机2 551分配给虚拟功能2 515。管理程序521可以将虚拟机3 552分配给虚拟功能3 526。管理程序531又可以将虚拟机4 553分配给虚拟功能2 535。管理程序可以通过主机通道适配器的每一个上的全特征物理功能513、523、533访问主机通道适配器。

[0114] 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内的流量。

[0115] 根据实施例，虚拟交换机512、522和532可以由它们相应的管理程序511、521、531

处理。在这样的vSwitch体系架构中，每个虚拟功能是完整的虚拟主机通道适配器(vHCA)，这意味着分配给VF的VM被分配完整的IB地址(例如，GID、GUID、LID)集合和硬件中专用的QP空间。对于网络的其余部分和SM(未示出)，HCA 510、520和530看起来像经由虚拟交换机、具有连接到它们的附加节点的交换机。

[0116] 根据实施例，本公开提供了用于提供具有动态LID分配和预填充LID的混合vSwitch体系架构的系统和方法。参考图9，管理程序511可以用具有预填充LID体系架构的vSwitch进行布置，而管理程序521可以用具有预填充LID和动态LID分配的vSwitch进行布置。管理程序531可以用具有动态LID分配的vSwitch进行布置。因此，物理功能513和虚拟功能514-516使其LID被预填充(即，即使那些未附连到活动虚拟机的虚拟功能被分配了LID)。物理功能523和虚拟功能1 524可以使其LID被预填充，而虚拟功能2和3、525和526使其LID被动态分配(即，虚拟功能2 525可用于动态LID分配，并且由于虚拟机3 552被附连，因此虚拟功能3 526具有动态分配的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)可以使其LID被动态分配。这使得虚拟功能1和3、534和536可用于动态LID分配，而虚拟功能2 535由于虚拟机4 553被附连到那里因此具有动态分配的LID 9。

[0117] 根据诸如图9所示、其中(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LID的vSwitch和具有动态LID分配的vSwitch两者的实施例，每主机通道适配器的预填充LID的数量可以由架构管理员定义，并且可以在0<＝预填充的VFS<＝总VF(每主机通道适配器)的范围内，并且可用于动态LID分配的VF可以通过从VF的总数(每主机通道适配器)减去预填充VF的数量找到。

[0118] 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，并且经由一个、两个或更多个虚拟交换机连接到外部IB子网。

[0119] InfiniBand-子网间通信(架构管理器)

[0120] 根据实施例，除了在单个子网内提供InfiniBand架构之外，本公开的实施例还可以提供跨越两个或更多个子网的InfiniBand架构。

[0121] 图10示出了根据实施例的示例性多子网InfiniBand架构。如图所示，在子网A 1000内，多个交换机1001-1004可以在子网A 1000(例如，IB子网)内提供架构(诸如InfiniBand架构)的成员之间的通信。架构可以包括多个硬件设备，诸如，例如通道适配器1010。主机通道适配器1010又可以与管理程序1011交互。管理程序又可以与其交互的主机通道适配器结合建立多个虚拟功能1014。管理程序可以附加地将虚拟机分配给每个虚拟功能，诸如虚拟机1 1015被分配给虚拟功能1 1014。管理程序可以通过主机通道适配器中的每一个上的全特征物理功能(诸如物理功能1013)访问其相关联的主机通道适配器。在子网B 1040内，多个交换机1021-1024可以在子网B 1040(例如，IB子网)内提供架构(诸如InfiniBand架构)的成员之间的通信。架构可以包括多个硬件设备，诸如，例如通道适配器1030。主机通道适配器1030又可以与管理程序1031交互。管理程序又可以与其交互的主机通道适配器建立多个虚拟功能1034。管理程序可以附加地将虚拟机分配给虚拟功能中的每一个，诸如虚拟机21035被分配给虚拟功能2 1034。管理程序可以通过主机通道适配器中的每一个上的全特征物理功能(诸如物理功能1033)访问其相关联的主机通道适配器。应当注意的是，虽然在每个子网(即，子网A和子网B)内仅示出了一个主机通道适配器，但是应该理

解的是，每个子网内可以包括多个主机通道适配器及其对应的部件。

[0122] 根据实施例，主机通道适配器中的每一个可以附加地与虚拟交换机(诸如虚拟交换机1012和虚拟交换机1032)相关联，并且每个HCA可以用不同的体系架构模型来建立，如上所述。虽然图10内的两个子网都被显示为使用具有预填充LID体系架构模型的vSwitch，但这并不意味着暗示所有此类子网配置都必须遵循相似的体系架构模型。

[0123] 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，并且子网B 1040内的交换机1021与路由器1006相关联。

[0124] 根据实施例，至少一个设备(例如，交换机、节点等)可以与架构管理器(未示出)相关联。架构管理器可以用于例如发现子网间架构拓扑、创建架构简档(例如，虚拟机架构简档)、构建虚拟机相关的数据库对象，这些数据库对象形成用于构建虚拟机架构简档的基础。此外，结构管理器可以根据哪些子网被允许经由哪些路由器端口使用哪些分区号进行通信来定义合法的子网间连接性。

[0125] 根据实施例，当始发源(诸如子网A内的虚拟机1)处的流量被寻址到不同子网(诸如子网B内的虚拟机2)中的目的地时，该流量可以被寻址到子网A内的路由器，即，路由器1005，路由器1005然后可以经由它与路由器1006的链路将流量传递到子网B。

[0126] 虚拟双端口路由器

[0127] 根据实施例，双端口路由器抽象可以提供一种简单的方式用于使得能够基于除了执行正常的基于LRH的交换之外还具有进行GRH(全局路由报头)到LRH(本地路由报头)转换能力的交换机硬件实现来定义子网到子网的路由器功能。

[0128] 根据实施例，虚拟双端口路由器可以逻辑地连接在对应交换机端口的外部。这种虚拟双端口路由器可以向标准管理实体(诸如子网管理器)提供符合InfiniBand规范的视图。

[0129] 根据实施例，双端口路由器模型意味着可以以其中每个子网完全控制分组的转发以及到子网的入口路径中的地址映射并且不影响任一不正确连接的子网内的路由和逻辑连接性的方式连接不同的子网。

[0130] 根据实施例，在涉及不正确连接的架构的情况下，使用虚拟双端口路由器抽象还可以允许诸如子网管理器和IB诊断软件的管理实体在存在与远程子网的非预期的物理连接性时正确地表现。

[0131] 图11示出了根据实施例的在高性能计算环境中的两个子网之间的互连。在利用虚拟双端口路由器进行配置之前，子网A 1101中的交换机1120可以通过交换机1120的交换机端口1121经由物理连接1110连接到子网B 1102中的交换机1130，该连接经由交换机1130的交换机端口1131。在这样的实施例中，每个交换机端口1121和1131即可以充当交换机端口又可以充当路由器端口。

[0132] 根据实施例，这种配置的问题在于诸如InfiniBand子网中的子网管理器的管理实体不能区分既是交换机端口又是路由器端口的物理端口。在这种情况下，SM可以将交换机端口视为具有连接到该交换机端口的路由器端口。但是，如果交换机端口经由例如物理链路连接到具有另一个子网管理器的另一个子网，则子网管理器可以能够在物理链路上发送发现消息。但是，这样的发现消息不能在另一个子网上被允许。

[0133] 图12示出了根据实施例的在高性能计算环境中经由双端口虚拟路由器配置的两个子网之间的互连。

[0134] 根据实施例，在配置之后，可以提供双端口虚拟路由器配置，使得子网管理器看到正确的端节点，从而表明子网管理器负责的子网的端部。

[0135] 根据实施例，在子网A 1201中的交换机1220处，交换机端口可以经由虚拟链路1223连接(即，逻辑上连接)到虚拟路由器1210中的路由器端口1211。虚拟路由器1210(例如，双端口虚拟路由器)(当其在实施例中被示出为在交换机1220的外部时，其逻辑上可以被包含在交换机1220内)还可以包括第二路由器端口：路由器端口II 1212。根据实施例，可以具有两个端部的物理链路1203可以经由路由器端口II 1212和包含在子网B 1202中的虚拟路由器1230中的路由器端口II 1232，经由物理链路的第一端将子网A 1201经由物理链路的第二端与子网B 1202连接。虚拟路由器1230可以附加地包括路由器端口1231，其可以经由虚拟链路1233连接(即，逻辑上连接)到交换机1240上的交换机端口1241。

[0136] 根据实施例，子网A上的子网管理器(未示出)可以检测虚拟路由器1210上的路由器端口1211，作为子网管理器控制的子网的端点。双端口虚拟路由器抽象可以允许子网A上的子网管理器以通常的方式(例如，如按照InifiniBand规范所定义的方式)处理子网A。在子网管理代理级别，可以提供双端口虚拟路由器抽象，使得SM看到正常的交换机端口，并且然后在SMA级别，抽象出连接到交换机端口的另一个端口，并且这个端口是双端口虚拟路由器上的路由器端口。在本地SM中，可以继续使用常规的架构拓扑(SM将端口视为拓扑中的标准交换机端口)，并且因此SM将路由器端口视为端部端口。物理连接可以在两个交换机端口之间进行，这两个交换机端口也被配置为两个不同子网中的路由器端口。

[0137] 根据实施例，双端口虚拟路由器还可以解决物理链路可能被错误地连接到相同子网中的某个其它交换机端口或者连接到不旨在提供到另一个子网的连接的交换机端口的问题。因此，本文描述的方法和系统还提供了关于子网外部的内容的表示。

[0138] 根据实施例，在子网(诸如子网A)内，本地SM确定交换机端口，并且然后确定连接到该交换机端口的路由器端口(例如，经由虚拟链路1223连接到交换机端口1221的路由器端口1211)。由于SM将路由器端口1211视为SM管理的子网的端部，因此SM不能超过这个点(例如，向路由器端口II 1212)发送发现消息和/或管理消息。

[0139] 根据实施例，上述双端口虚拟路由器提供的益处是双端口虚拟路由器抽象完全由双端口虚拟路由器所属的子网内的管理实体(例如，SM或SMA)管理。通过允许仅在本地侧进行管理，系统不必提供外部独立的管理实体。即，子网到子网连接的每一侧都可以负责配置其自己的双端口虚拟路由器。

[0140] 根据实施例，在其中被寻址到远程目的地(即，在本地子网外部)的诸如SMP的分组到达未经由如上所述的双端口虚拟路由器配置的本地目标端口的情况下，那么本地端口可以返回指定它不是路由器端口的消息。

[0141] 本发明的许多特征可以在硬件、软件、固件或其组合中、利用硬件、软件、固件或其组合、或者在硬件、软件、固件或其组合帮助下执行。因此，本发明的特征可以使用(例如，包括一个或多个处理器的)处理系统来实现。

[0142] 图13示出了根据实施例的用于支持高性能计算环境中的双端口虚拟路由器的方法。在步骤1310处，该方法可以在包括一个或多个微处理器的一个或多个计算机处提供第

一子网，第一子网包括：多个交换机，该多个交换机至少包括叶交换机，其中该多个交换机中的每个交换机包括多个交换机端口；多个主机通道适配器，每个主机通道适配器包括至少一个主机通道适配器端口；多个端节点，其中端节点中的每个端节点与多个主机通道适配器中的至少一个主机通道适配器相关联；以及子网管理器，该子网管理器在多个交换机和多个主机通道适配器中的一个上运行。

[0143] 在步骤1320处，该方法可以将多个交换机中的交换机上的多个交换机端口中的交换机端口配置为路由器端口。

[0144] 在步骤1330处，该方法可以将配置为路由器端口的交换机端口逻辑地连接到虚拟路由器，该虚拟路由器包括至少两个虚拟路由器端口。

[0145] 路由器SMA抽象，以便允许基于SMP的连接性检查

[0146] 根据实施例，不允许子网管理分组(SMP)具有意味着分组将被发送到路由器端口之外的寻址信息。但是，为了允许在(虚拟)路由器端口的远程侧发现物理连接性(即，远程连接性的本地发现)，可以定义一组新的SMA属性，其中每个这样的新属性表示“远程信息”以及或者标准或者特定于供应商的SMA属性的指示。

[0147] 根据实施例，当路由器SMA处理表示“远程”信息/属性的属性时，那么对应的SMP请求可以以对原始请求的发送者完全透明的方式在外部物理链路上发送。

[0148] 根据实施例，本地SMA可以选择独立于传入请求来执行远程发现并且然后在本地高速缓存相关信息，或者它可以像简单的代理一样起作用并且每次它接收到指定“远程”信息/属性的请求时生成到外部链路的对应请求。

[0149] 根据实施例，通过跟踪请求“远程”属性的SMP是从本地子网侧(即，逻辑上连接到本地交换机端口的虚拟路由器端口)还是从外部链路(即，虚拟路由器的远程侧)被接收到，SMA实现可以跟踪远程请求在或者表示本地子网中的原始请求或者表示来自对等路由器端口的代理请求方面在多大程度上有效。

[0150] 根据实施例，对于符合IB规范的SM，路由器端口是子网中的端部端口。由此，低级别的SMP(用于发现和配置子网)不能跨越路由器端口被发送。但是，为了维护对子网间流量的路由，本地SM或架构管理器需要在它对本地资源做出任何配置之前能够观察物理链路的远程侧的物理连接性。但是，关于这种看到远程连接性的期望，不允许SM配置物理链路的远程侧(即，SM的配置必须被包含在其自己的子网内)。

[0151] 根据实施例，SMA模型增强允许发送寻址到本地路由器端口的分组(即，SMP)的可能性。其中分组被寻址的SMA可以接收分组，并且然后应用定义所请求的信息在(例如，通过物理链路跨子网连接的)远程节点上的新属性。

[0152] 根据实施例，SMA可以作为代理操作(接收SMP并发送另一个请求)，或者SMA可以修改原始分组并将其作为子网间分组继续发送。SMA可以更新分组中的地址信息。这种更新向SMP添加了1跳直接路由路径。SMP然后可以被远程节点(路由器端口)接收。SMP可以独立于远端上的节点是以相同的方式(例如，作为虚拟路由器)配置还是被配置为基本交换机端口(例如，与传统交换机实现的物理连接性)来工作。然后，接收节点将看到的请求分组是基本的请求，并且它将以通常的方式响应。请求源自子网之外的事实对于接收节点是透明的(不可见的)。

[0153] 根据实施例，这允许本地子网通过利用抽象来发现远程子网。所提供的SMA抽象允

许从远程子网(例如,跨物理链路)检索信息,而远程侧不会意识到它已经被本地(即,对于其上执行子网发现是远程的)子网查询。

[0154] 寻址方案

[0155] 根据实施例,为了保持符合IB规范,SMP分组被子网的边界限定(即,不允许SM“查看”或发现它与之相关联的子网外部的信息。但是,仍然需要从虚拟路由器端口的远端(即,超出子网边界的一“跳”)检索信息,诸如连接性信息。这种信息可以被编码成特定类型的SMP,其可以被称为特定于供应商的SMP(VSMP)。

[0156] 根据实施例,VSMP通常可以利用与一般SMP(子网限定的SMP)类似的寻址方案,可以使用DR(有向路由-其中当从一个交换机到另一个交换机时SMP可以明确指示它存在于哪个端口)和LID路由两者。但是,对于可以应用到路由器端口的远端的那些属性,可以使用属性修饰符中的单个位来指示本地端口与远程端口。如果设置了属性的远程位,那么在表示路由器端口的SMA处可能发生附加的处理。

[0157] 根据实施例,寻址方案的一个重要方面是,即使在系统的错误配置或布线的情况下,路由器端口的远程对等端口也能够对请求进行响应。例如,如果虚拟路由器经由例如物理链路连接到远程子网的通用交换机端口处的远程子网,那么处理远程子网的通用交换机端口的SMA可以用指示不支持远程属性的状态值对请求进行响应,并且响应可以到达请求的SM。

[0158] 根据实施例,可以或者通过使用DR路径或者经由路由器端口的LID将SMP(即,VSMP)发送到本地路由器端口。如果所请求的信息是用于路由器端口的远程对等端口,那么可以设置属性修饰符的远程标志/位。当SMA接收到这样的VSMP时,它可以更改分组并且添加寻址远端的附加DR步骤。

[0159] 根据实施例,可以使用分组属性的一部分(例如,16位属性修饰符的一位)来发信号通知VSMP是本地的还是远程的。例如,通过将这个部分设置为1的值可以意味着远程对等端口是VSMP的最终目的地。属性的这个部分可以被称为远程标志。

[0160] 根据实施例,除了远程标志之外,还可以存在附加的标志,该附加的标志可以指示沿着到最终目的地的路途哪个目的地实例应该处理VSMP。分组属性的两个附加部分(例如,属性修饰符的两个位)可以用于此目的。被称为第一接收器标志的第一部分(例如,位20)可以指示分组预期在被本地路由器端口(其应该与原始请求的目的地地址匹配)接收到时进行处理。当第一接收器处的所有预期处理都完成时,可以设置分组属性的第二部分(例如,属性修饰符的位21),并将分组转发到远端。这个第二部分可以被称为第二接收器标志。

[0161] DR路由的分组

[0162] 根据实施例,作为示例,DR路由的分组可以遵循示例性流程。在LID A处的源节点可以发起指定路由器1作为目的地节点的请求分组。示例性DR路由的分组配置在这里:

[0163] • MADHdr.Class=0x81(DR路由的SMP)

[0164] • MADHdr.Method=0x1(获取)

[0165] • LRH.SLID=LRH.DLID=0xffff(许可的LID)

[0166] • MADHdr.DrSLID=MADHdr.DrDLID=0xffff

[0167] • MADHdr.AttrID=<VSMP attrID>

[0168] • MADHdr.AttrMod.remote=1

[0169] • MADHdr.AttrMod.first_receiver＝1

[0170] • MADHdr.InitPath＝DR path to Rtr1(LID B)

[0171] • MADHdr.HopCnt＝N

[0172] • MADHdr.HopPtr＝0

[0173] 当请求分组到达路由器1时，它可以被继续传递到对应的SMA，该对应的SMA然后可以验证请求是有效的。在验证请求的有效性之后，SMA然后可以修改分组。这种修改可以包括用额外的一跳扩展DR路径，并设置第二接收器标志。这里示出了这样的示例性配置修改：

[0174] • MADHdr.HopCnt＝N+1(用额外的一跳扩展DR路径)

[0175] • MADHdr.InitPath[N+1]＝(虚拟路由器外部端口号(即2))

[0176] • MADHdr.AttrMod.second_receiver＝1

[0177] 根据实施例，SMA的SMI(子网管理接口)层可以在SMA将VSMP转发出到物理链路上之前用额外的跳来更新DR。

[0178] 根据实施例，对于在物理链路的远程侧接收VSMP的SMA，由VSMP使用的寻址方案可以看起来像是正常的IB定义的分组的寻址方案。

[0179] 根据实施例，当在物理链路的远程侧上接收VSMP的SMA是VSMP的最终目的地时，VSMP可以通过接收到的SMA进行验证。验证可以包括相对于标志设置检查输入端口。如果远程标志以及第一接收标志和第二接收标志都被设置，那么分组可以在物理端口上(即，从配置有虚拟路由器的端口的外部链路侧)被接收。如果远程标志和仅第一接收器标志被设置，那么分组可以到达虚拟链路(即，从虚拟路由器端口的内部交换机侧)。如果验证失败，那么状态可以被设置为适当的错误消息。

[0180] 图14是图示根据实施例的DR路由的VSMP分组的流程图。

[0181] 在步骤1401处，根据实施例，在子网A内，诸如子网A的子网管理器的实体可以经由DR路由的VSMP请求连接性信息。DR路由的VSMP的目的地可以是子网A内的路由器，例如路由器1。

[0182] 在步骤1402处，DR路由的VSMP可以到达路由器1。在实施例中并且如上所述，路由器1可以被包含在诸如交换机1的交换机中。DR路由的VSMP可以被传递到交换机1的SMA用于验证。

[0183] 在步骤1403处，根据实施例，SMA可以修改DR路由的VSMP。该修改可以包括扩展跳计数器(用于跨物理链路到第二子网的跳)，并设置第二接收器标志。

[0184] 在步骤1404处，根据实施例，SMI(与交换机/路由器的SMA相关联的SMI)可以更新VSMP的跳指针。

[0185] 在步骤1405处，根据实施例，SMI可以在物理链路上跨子网A 1420和子网B 1430之间的子网边界转发DR VSMP，其中物理链路的第一端可以连接到子网A内的路由器并且物理链路的第二端可以连接到子网B内的路由器。

[0186] 在步骤1406处，根据实施例，子网B 1430内的路由器2的SMA'可以从物理链路接收VSMP。

[0187] 在步骤1407处，根据实施例，SMI'(与SMA'相关联)可以验证VSMP请求。

[0188] 根据实施例，响应实体(例如，物理链路的远程侧的路由器)可以完成SMP响应并且设置指示响应的方向属性。这里示出了这样的示例性配置：

[0189] • MADHdr.Method＝0x81(GetResp)

[0190] • MADHdr.Direction＝1(指示响应)

[0191] 根据实施例,SMI层然后可以递减跳指针并且在物理链路上将响应转发出返回到本地侧。在本地路由器处的SMA可以恢复对请求做出的修改。本地侧路由器处的SMI可以执行正常的处理,包括递减跳数,并在虚拟链路上向交换机发送出响应(即,在交换机端口和虚拟路由器端口之间的内部虚拟链路上)。对于下一跳,SMI可以再次递减跳数,并在请求最初到达的物理交换机端口上将分组发送出。

[0192] 图15是图示根据实施例的对DR路由的VSMP分组的响应的流程图。

[0193] 在步骤1501处,根据实施例,路由器2的SMA'可以填充对VSMP的响应信息(例如,连接性信息),并且设置方向标志以指示响应。

[0194] 在步骤1502处,根据实施例,SMI'可以递减响应的跳指针,并且在例如具有两端的物理链路上将响应从子网B 1530转发回到子网A 1520。物理链路的第一端可以连接到子网A内的路由器,并且物理链路的第二端可以连接到子网B内的路由器。

[0195] 在步骤1503处,根据实施例,路由器1处的SMA可以接收响应并且恢复对VSMP进行的修改。

[0196] 在步骤1504处,根据实施例,路由器1处的SMI可以处理对VSMP的响应并且在链路(例如,虚拟路由器1和物理交换机之间的虚拟链路)上将响应发送到物理交换机。

[0197] 在步骤1505处,根据实施例,SMI可以递减跳指针计数器,并在最初接收到VSMP的物理交换机端口上将分组发送出。

[0198] LID路由的分组

[0199] 根据实施例,作为示例,LID路由的分组可以遵循示例性流程。在LID A处的源节点可以发起指定路由器1作为目的地节点的请求分组。这样的示例性分组可以具有以下配置:

[0200] • MADHdr.Class＝0x01(LID路由的SMP)

[0201] • MADHdr.Method＝0x1(获取)

[0202] • LRH.SLID＝LID A

[0203] • LRH.DLID＝LID B

[0204] • MADHdr.AttrID＝<VSMP attrID>

[0205] • MADHdr.AttrMod.remote＝1

[0206] • MADHdr.AttrMod.first_receiver＝1

[0207] 根据实施例,当请求分组到达路由器1时,它可以被继续传递到对应的SMA,该对应的SMA然后可以验证请求是有效的。在验证请求的有效性之后,SMA可以修改分组。这种修改可以包括在最后添加单个DR路径。因此,这意味着完全地址可以是LID路由的分组和DR路由的跳的组合。示例性配置在这里:

[0208] • MADHdr.Class＝0x81(DR路由的SMP)

[0209] • MADHdr.HopCnt＝1

[0210] • MADHdr.HopPtr＝0

[0211] • MADHdr.Direction＝0(出站)

[0212] • MADHdr.InitPath[1]＝(虚拟路由器外部端口号(即2))

[0213] • MADHdr.DrSLID＝LRH.SLID(即,LRH.SLID包含来自原始请求者的LID A)

[0214] • MADHdr.DrDLID＝0xffff

[0215] • MADHdr.AttrMod.second_receiver＝1

[0216] 根据实施例，路由器处的SMI层可以正常处理出站分组，并在物理链路上将它发送出。示例性配置在这里：

[0217] • LRH.SLID＝LRH.DLID＝0xffff

[0218] • MADHdr.HopPtr＝1(递增1)

[0219] 根据实施例，(在物理链路的另一端处的)目的地路由器处的SMI可以确定它是请求的目的地并且将VSMP传递到相关联的SMA上。

[0220] 图16是图示根据实施例的LID路由的VSMP分组的流程图。

[0221] 在步骤1601处，根据实施例，在子网A内，诸如子网A的子网管理器的实体可以经由LID路由的VSMP请求连接性信息。LID路由的VSMP的目的地可以是子网A内的路由器，例如路由器1。

[0222] 在步骤1602处，LID路由的VSMP可以到达路由器1。在实施例中并且如上所述，路由器1可以被包含在诸如交换机1的交换机中。LID路由的VSMP可以被传递到交换机1的SMA用于验证。

[0223] 在步骤1603处，根据实施例，SMA可以修改DR路由的VSMP。该修改可以包括将单跳DR路由的路径添加到地址的末尾、扩展跳计数器(用于跨物理链路到第二子网的跳)、并设置第二接收器标志。

[0224] 在步骤1604处，根据实施例，SMI(与交换机/路由器的SMA相关联的SMI)可以更新VSMP的跳指针。

[0225] 在步骤1605处，根据实施例，SMI可以在物理链路上跨子网A 1620与子网B 1630之间的子网边界转发最初LID路由的VSMP，其中物理链路的第一端可以连接到子网A内的路由器，并且物理链路的第二端可以连接到子网B内的路由器。

[0226] 在步骤1606处，根据实施例，子网B 1630内的路由器2的SMA'可以从物理链路接收现在DR路由的VSMP。

[0227] 在步骤1607处，根据实施例，SMI'(与SMA'相关联)可以验证VSMP请求。

[0228] 根据实施例，对于响应流程，路由器2处的SMA可以验证VSMP。验证可以包括相对于标志设置检查输入端口。如果远程标志以及第一接收标志和第二接收标志都被设置，那么分组可以在物理端口上(即，从配置有虚拟路由器的端口的外部链路侧)被接收。如果远程标志和仅第一接收器标志被设置，那么分组可以到达虚拟链路(即，从虚拟路由器端口的内部交换机侧)。如果验证失败，那么状态可以被设置为适当的错误消息。

[0229] 根据实施例，路由器2处的SMA'可以完成SMP响应并且设置指示响应的方向属性。这里示出了这样的示例性配置：

[0230] • MADHdr.Method＝0x81(GetResp)

[0231] • MADHdr.Direction＝1(指示响应)

[0232] • LRH.SLID＝0xffff

[0233] • LRH.DLID＝ReqMAD.MADHdr.SLID＝0xffff

[0234] 路由器2处的SMI'然后可以递减跳指针，并在物理链路上将响应转发出返回到路由器1。然后，路由器1处的SMA然后可以在将响应转发出物理交换机端口返回到源之前恢复

对原始请求执行的修改。这种恢复之后的示例性配置在这里：

[0235] • MADHdr.Class＝0x01

[0236] • LRH.DLID＝MADHdr.DrSLID(即，包含来自原始请求者的LID A)

[0237] • LRH.SLID＝local LID＝B

[0238] 根据实施例，SMA可以附加地清除VSMP的特定于DR的字段，使得当响应到达原始请求者时，响应看起来与原始VSMP完全一致。

[0239] 根据实施例，一旦响应返回到达源处，源将看到正常的路由器LID响应，就好像它已经完全在路由器1处被处理一样。

[0240] 图17是图示根据实施例的对LID路由的VSMP分组的响应的流程图。

[0241] 在步骤1701处，根据实施例，路由器2的SMA'可以填充对VSMP的响应信息(例如，连接性信息)，并且设置方向标志以指示响应。

[0242] 在步骤1702处，根据实施例，SMI'可以递减响应的跳指针并且在例如具有两端的物理链路上将响应从子网B 1730转发到子网A 1720。物理链路的第一端可以连接到子网A内的路由器，并且物理链路的第二端可以连接到子网B内的路由器。

[0243] 在步骤1703处，根据实施例，路由器1处的SMA可以接收到响应并且恢复对VSMP进行的修改，包括取消DR路由的跳。

[0244] 在步骤1704处，根据实施例，路由器1处的SMI可以处理对VSMP的响应，并且在最初接收到VSMP的物理交换机端口上将LID路由的响应发送出。

[0245] 子网间分区

[0246] 根据实施例，某些分区可以被分类为“子网间分区”(本文也被称为ISP)。当这样的某些分区因此被分类时，子网管理实体(例如SM)可以确保只有适当分类的分区被允许用于到远程子网的数据流量。

[0247] 根据实施例，架构管理器可以分配ISP P_Key值的范围，并且诸如子网间管理器(ISM)的全局管理实体可以实现和实施所有ISP可以位于其内的这种特定P_Key范围(例如，在这样的环境中的任何ISP必须具有如由ISM定义的P_Key范围内的P_Key)。通过确保从特定的P_Key范围分配ISP，有可能确保一些P_Key范围可以由不同的子网用于不同的目的，而用于ISP的P_Key范围被所有相关的子网一致地使用。

[0248] 根据实施例，通过观察哪些端节点是哪些ISP的成员并将其与其中本地路由器端口是相同ISP的成员相关联，可以在每个本地子网内以及在连接的子网之间确定可能的子网间路径的集合。

[0249] 根据实施例，通过观察来自不同子网的直接连接的路由器端口都是同一ISP集合的成员，可以确定在两个子网中的配置是一致的，而无需在子网之间交换附加的信息。

[0250] 图18是示出根据实施例的子网间分区P_Key范围的定义的框图。

[0251] 根据实施例，在诸如InfiniBand架构的网络架构1800内存在诸如Oracle架构管理器的架构管理器1801。架构管理器可以明确地创建和定义多个子网间分区，并且还可以在子网之间建立ISP以便允许数据流量。架构管理器可以通过从用于整个架构的单个配置范围分配ISP P_Key值来确保ISP在网络架构中是唯一的。这个定义的ISP P_Key范围1802可以被传送(在图中用虚线示出)到多个子网，诸如子网A-H 1810-1817。

[0252] 根据实施例，子网A-H中的每一个可以包括相关联的子网管理器1820-1827。子网

管理器中的每一个可以将在定义的ISP P_Key范围1802内的P_Key分配给每个子网管理器的相应子网内的路由器端口。

[0253] 根据实施例，通过将定义的ISP P_Key范围内的P_Key分配给子网中的路由器端口，这可以允许在例如链路1830-1837上的子网间通信。

[0254] 图19是根据实施例的支持子网间分区的框图。为了在两个子网之间具有数据流量，必须对ISP(子网间分区)-应该在两个子网之间公用的分区达成协议。

[0255] 根据实施例，在验证两个子网之间存在连接性(以上讨论)之后，子网A 1810中的子网管理器1820可以向路由器端口1910分配P_Key值(例如，P_Key A)。同样，子网管理器1821可以向路由器端口1920分配P_Key值(例如，P_Key A)。P_Key A可以在如由架构管理器定义的并且分发给子网和子网管理器的ISP P_Key值的范围内。

[0256] 根据实施例，通过将P_Key值分配给落入定义的ISP P_Key范围内的每个路由器端口，路由器端口1910和1920可以跨物理链路1830相互通信。

[0257] 根据实施例，管理实体(即，ISM)可以与其它子网中的管理实体关于存在哪些ISP(例如，P_Key值)交换信息，并且关于哪些节点应该被允许通信/可访问交换信息。

[0258] 根据实施例，虽然在图中没有绘出，但是如上所述，路由器端口1910和1920可以被配置为双端口虚拟路由器。

[0259] 图20是根据实施例的支持子网间分区的框图。在子网之间建立通信并且关于ISP达成一致之后，子网A 1810中的子网间管理器2010可以通知子网B 1811内子网管理器2020关于包含在子网A 1810中的节点以及每个节点属于哪些分区。但是，只有与连接子网的路由器端口共享ISP成员资格的节点有关的信息才需要被交换。换句话说，在其中只有子网内全部端节点(虚拟和/或物理)的一小部分是允许它们与特定远程子网中的节点进行通信的ISP的成员的情况下，交换关于每个子网中的所有节点的信息将是浪费的。此外，从安全角度来看，期望将关于端节点的信息交换限制在所涉及的子网/ISM之间严格“需要知道”的基础上。当两个连接的子网由不同的架构管理器管理以及还有在架构级别的不同域进行管理时，尤其是这种情况。只要分区达成一致并且在定义的ISP范围内，这就可以允许在不同子网的节点之间的子网间通信。

[0260] 根据实施例，例如，ISM 2010可以通知ISM 2020子网A包括节点A 2030和节点B 2031，并且这些节点是具有P_Key A的ISP分区的成员。

[0261] 根据实施例，ISM 2020可以通知ISM 2010子网B包括节点E 2040和节点H 2043，并且这些是具有P_Key A的ISP分区的成员(虽然节点H附加地包括P_Key D)。注意的是，具有P_Key B的分区，虽然不包含在定义的ISP范围中，但是仍然可以存在于两个子网中。

[0262] 根据实施例，连接性由每个子网内的ISM控制，意味着不存在配置每一侧控制所有事物的中央管理。每个子网内或子网间连接两侧的管理实体(例如子网管理器)负责建立ISP成员资格，并且然后对应的ISM负责交换相关信息。

[0263] 根据实施例，除了以上讨论的静态P_Key分布之外，系统和方法还可以支持动态P_Key分布。例如，在图20中看，SM可以继续交换关于每个端节点是其部分的分区和P_Key的信息。例如，如果子网管理器1820确定节点A 2030不再是具有P_Key A的ISP分区的一部分，那么ISM 2010可以然后与ISM 2020通信，即节点A不再是ISP分区的一部分，并且不能再参与子网间通信。路由、地址映射硬件的操纵在每个子网的本地发生。

[0264] 根据实施例，虽然在图中没有绘出，但是如上所述，路由器端口1910和1920可以被配置为双端口虚拟路由器。

[0265] 图21是根据实施例的用于支持子网间分区的示例性方法的流程图。

[0266] 在步骤2110处，该方法可以在包括一个或多个微处理器的一个或多个计算机处提供第一子网，第一子网包括：多个交换机，该多个交换机至少包括叶交换机，其中该多个交换机中的每个交换机包括多个交换机端口；多个主机通道适配器，每个主机通道适配器包括至少一个主机通道适配器端口；多个端节点，其中端节点中的每个端节点与多个主机通道适配器中的至少一个主机通道适配器相关联；子网管理器，该子网管理器在多个交换机和多个主机通道适配器中的一个上运行；以及子网间管理器，该子网间管理器在多个交换机和多个主机通道适配器中的一个上运行。

[0267] 在步骤2120处，该方法可以在包括一个或多个微处理器的一个或多个计算机处提供驻留在高性能计算环境中的架构管理器。

[0268] 在步骤2130处，该方法可以将多个交换机中的交换机上的多个交换机端口中的交换机端口配置为路由器端口。

[0269] 在步骤2140处，该方法可以逻辑地将配置为路由器端口的交换机端口连接到虚拟路由器，该虚拟路由器包括至少两个虚拟路由器端口。

[0270] 在步骤2150处，该方法可以由架构管理器定义多个分区键值内的分区键(P_Key)值的范围，定义的分区键值的范围包括子网间分区(ISP)P_Key范围。

[0271] 在步骤2160处，该方法可以由架构管理器将ISP P_Key范围传送给子网管理器。

[0272] 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执行，其中计算机程序产品是具有其上/其中存储有可用来编程处理系统以执行本文所呈现的任何特征的指令的存储介质或计算机可读介质。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体或设备。

[0273] 存储在任何一种机器可读介质中，本发明的特征可以被结合到软件和/或固件中，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。

[0274] 本发明的特征也可以利用例如诸如专用集成电路(ASIC)的硬件部件在硬件中实现。硬件状态机的实现使得执行本文所描述的功能对相关领域的技术人员将是显而易见的。

[0275] 此外，本发明可以方便地利用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，其中包括一个或多个处理器、存储器和/或根据本公开内容的教导编程的计算机可读存储介质。适当的软件编码可以容易地由熟练的程序员基于本公开内容的教导来准备，如对软件领域的技术人员将显而易见的。

[0276] 虽然以上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给出。对相关领域的技术人员来说，将很显然，在不背离本发明的精神和范围的情况下，其中可以做出各种形式和细节上的变化。

[0277] 本发明已经借助说明具体功能及其关系的执行的功能构建块进行了描述。这些功能构建块的边界在本文中通常是为了方便描述而任意定义的。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之内。

[0278] 本发明的以上描述是为了说明和描述的目的提供。它不是穷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该由任何上述示例性实施例来限制。许多修改和变化对本领域技术人员来说将显而易见。修改和变化包括所公开特征的任何相关组合。实施例的选择与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理解本发明用于各种实施例并且可以进行适于预期特定用途的各种修改。本发明的范围要由以下权利要求及其等价物来定义。

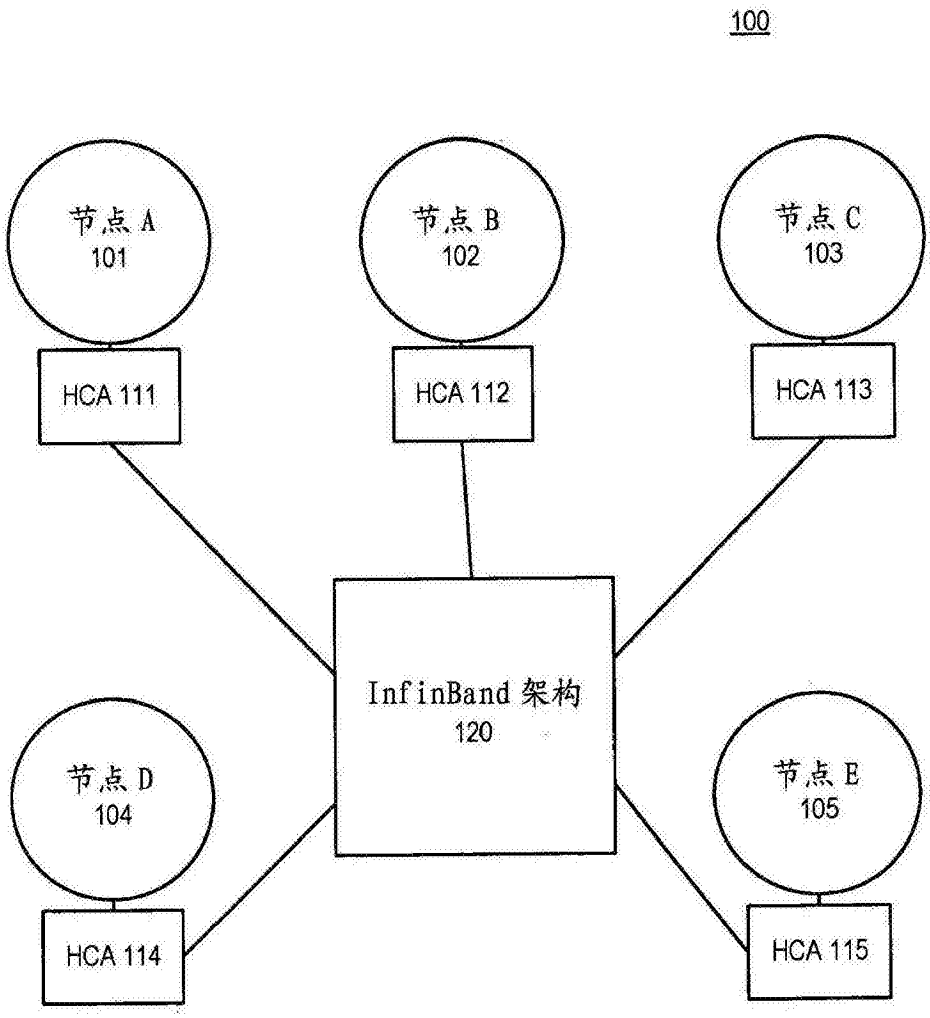

100
节点 A
101
节点 B
102
节点 C
103
HCA 111
HCA 112
HCA 113
InfinBand 架构
120
节点 D
104
节点 E
105
HCA 114
HCA 115

图1

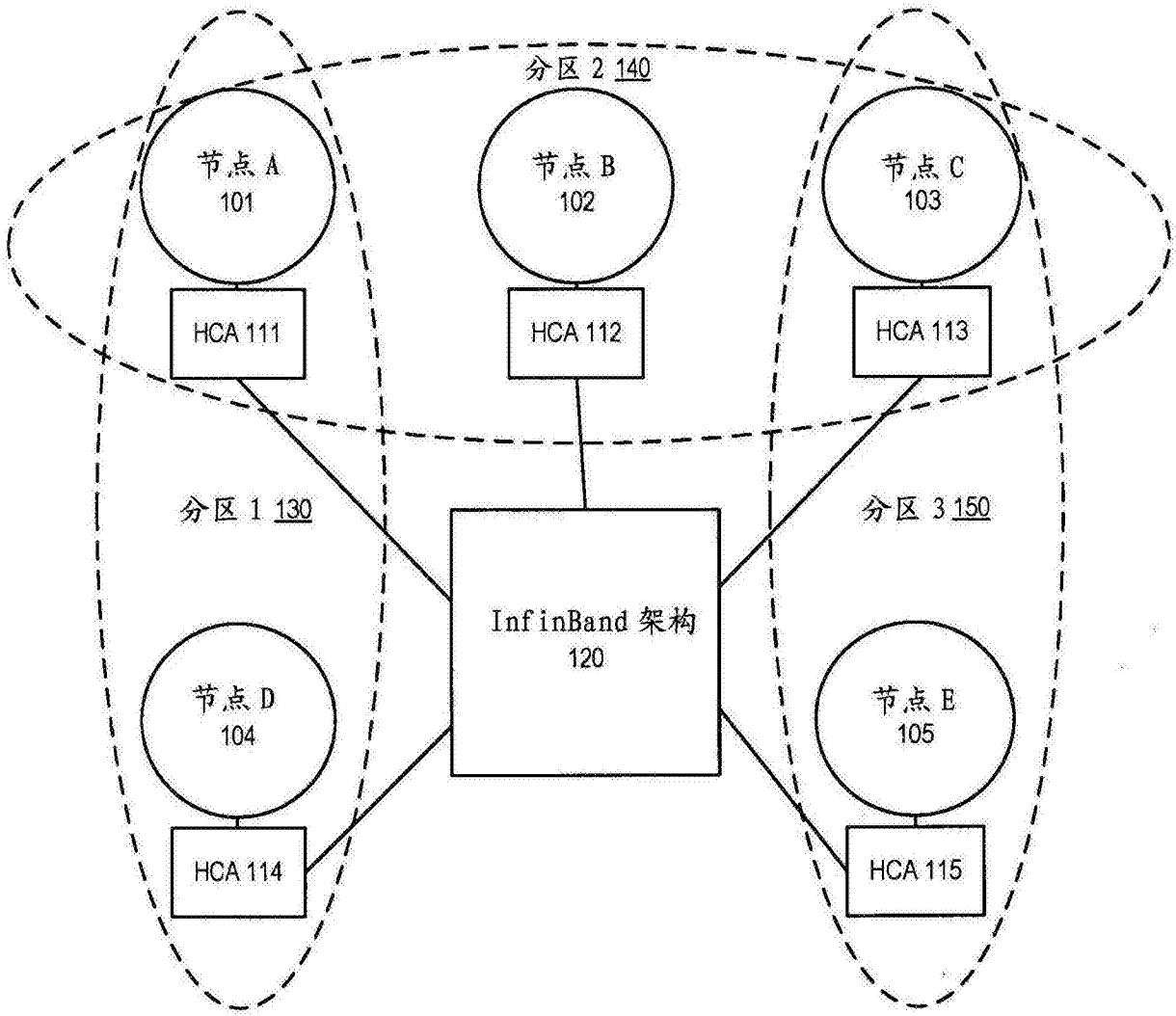

分区 2 140
节点 A
101
节点 B
102
节点 C
103
HCA 111
HCA 112
HCA 113
分区 1 130
分区 3 150
InfinBand 架构
120
节点 D
104
节点 E
105
HCA 114
HCA 115

图2

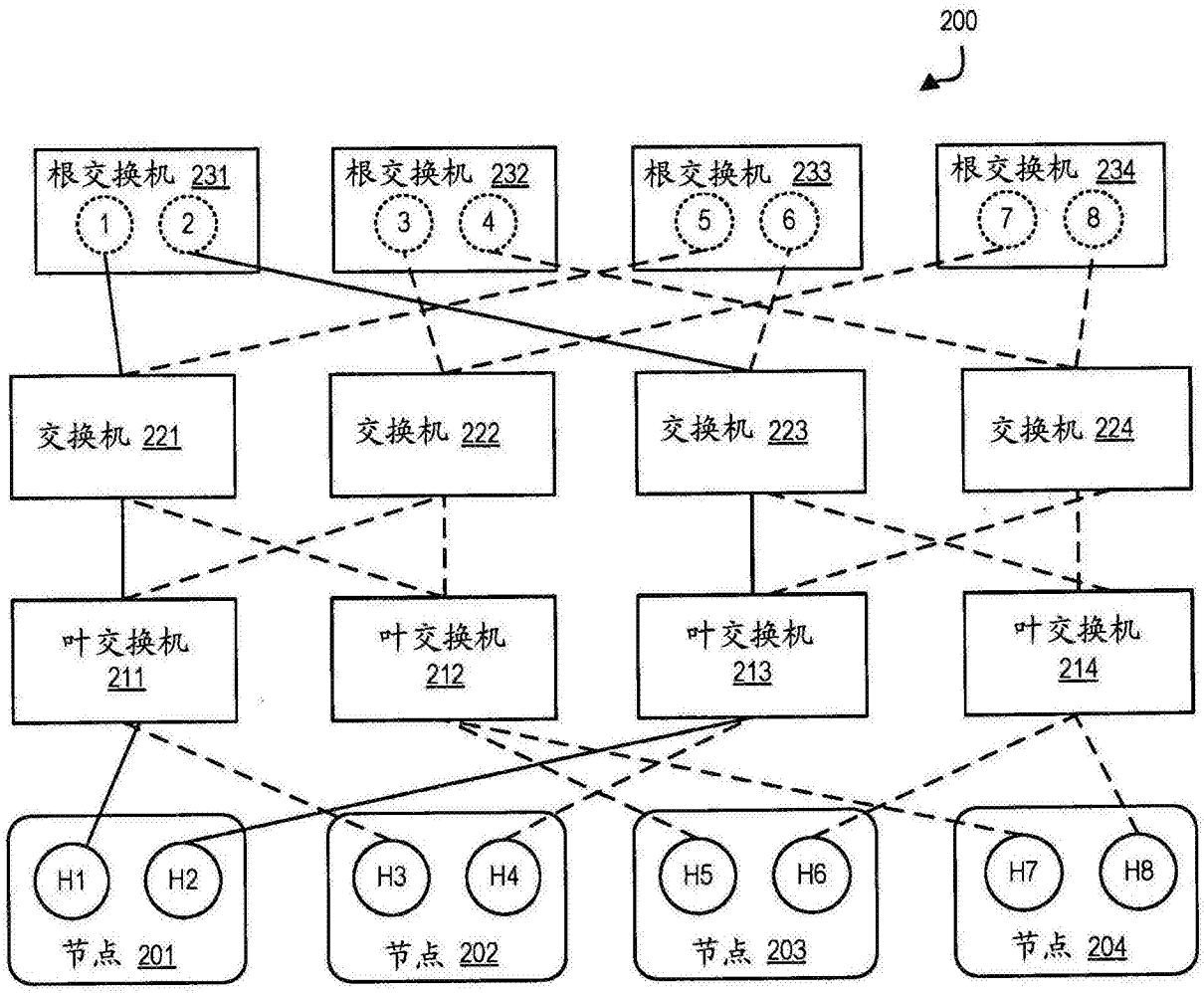

200
根交换机 231
1
2
根交换机 232
3
4
根交换机 233
5
6
根交换机 234
7
8
交换机 221
交换机 222
交换机 223
交换机 224
叶交换机 211
叶交换机 212
叶交换机 213
叶交换机 214
H1
H2
节点 201
H3
H4
节点 202
H5
H6
节点 203
H7
H8
节点 204

图3

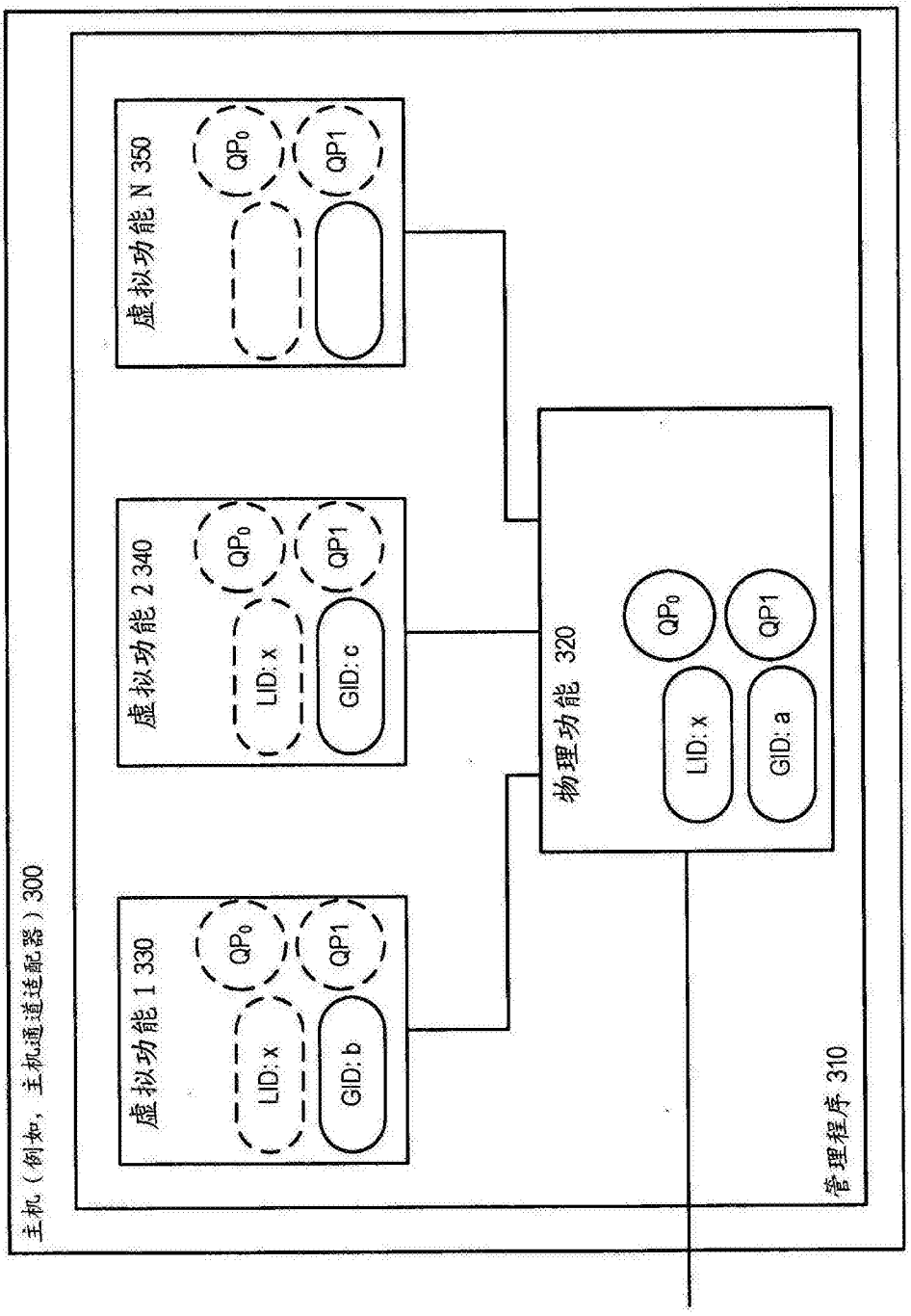

主机（例如，主机通道适配器）300
虚拟功能 1 330
虚拟功能 2 340
虚拟功能 N 350
物理功能 320
管理程序 310
QP0
QP1
LID: x
GID: b
GID: c
GID: a

图4

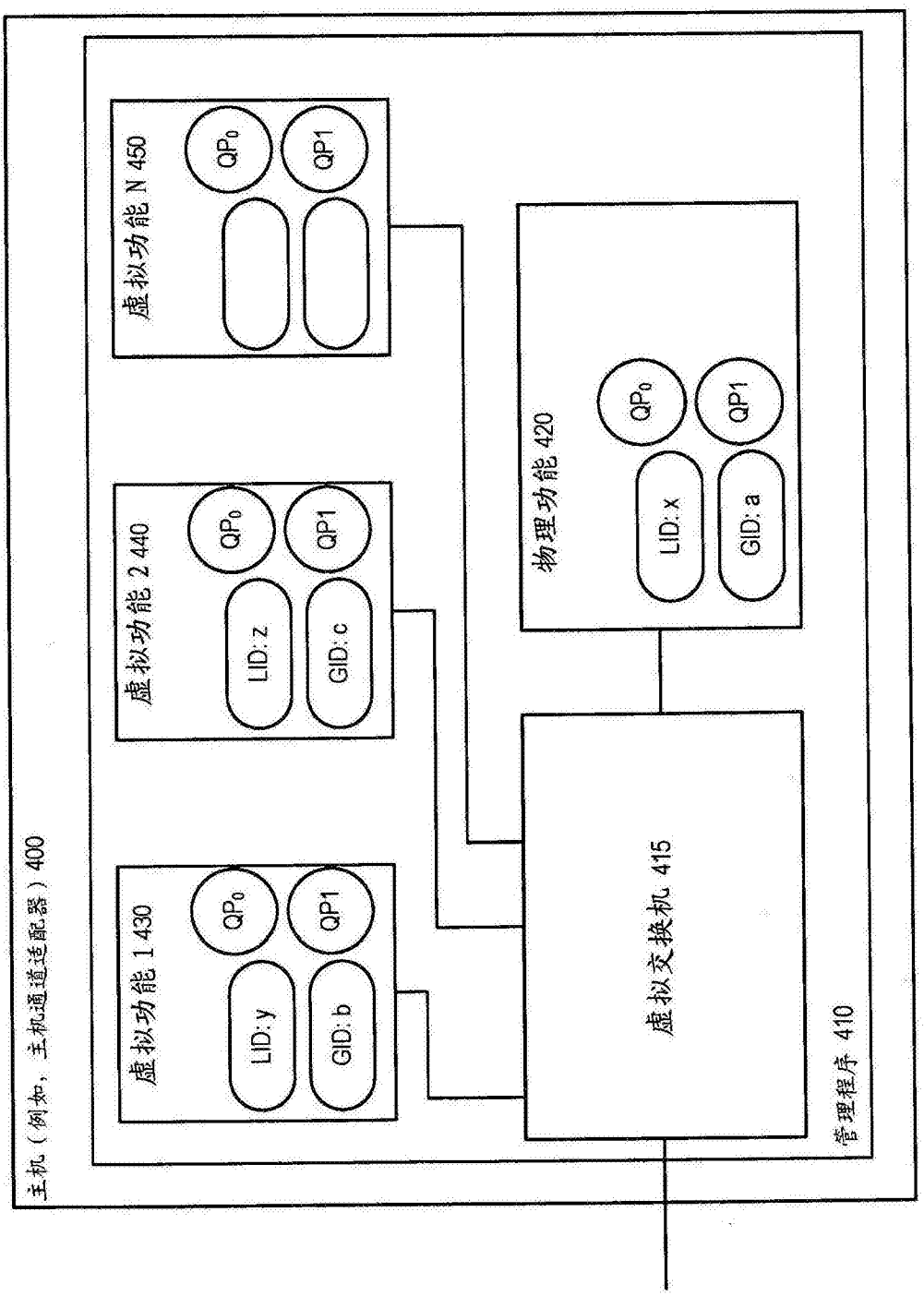

主机（例如，主机通道适配器）400
虚拟功能 1 430
QP0
QP1
LID: y
GID: b
虚拟功能 2 440
QP0
QP1
LID: z
GID: c
虚拟功能 N 450
QP0
QP1
物理功能 420
QP0
QP1
LID: x
GID: a
虚拟交换机 415
管理程序 410

图5

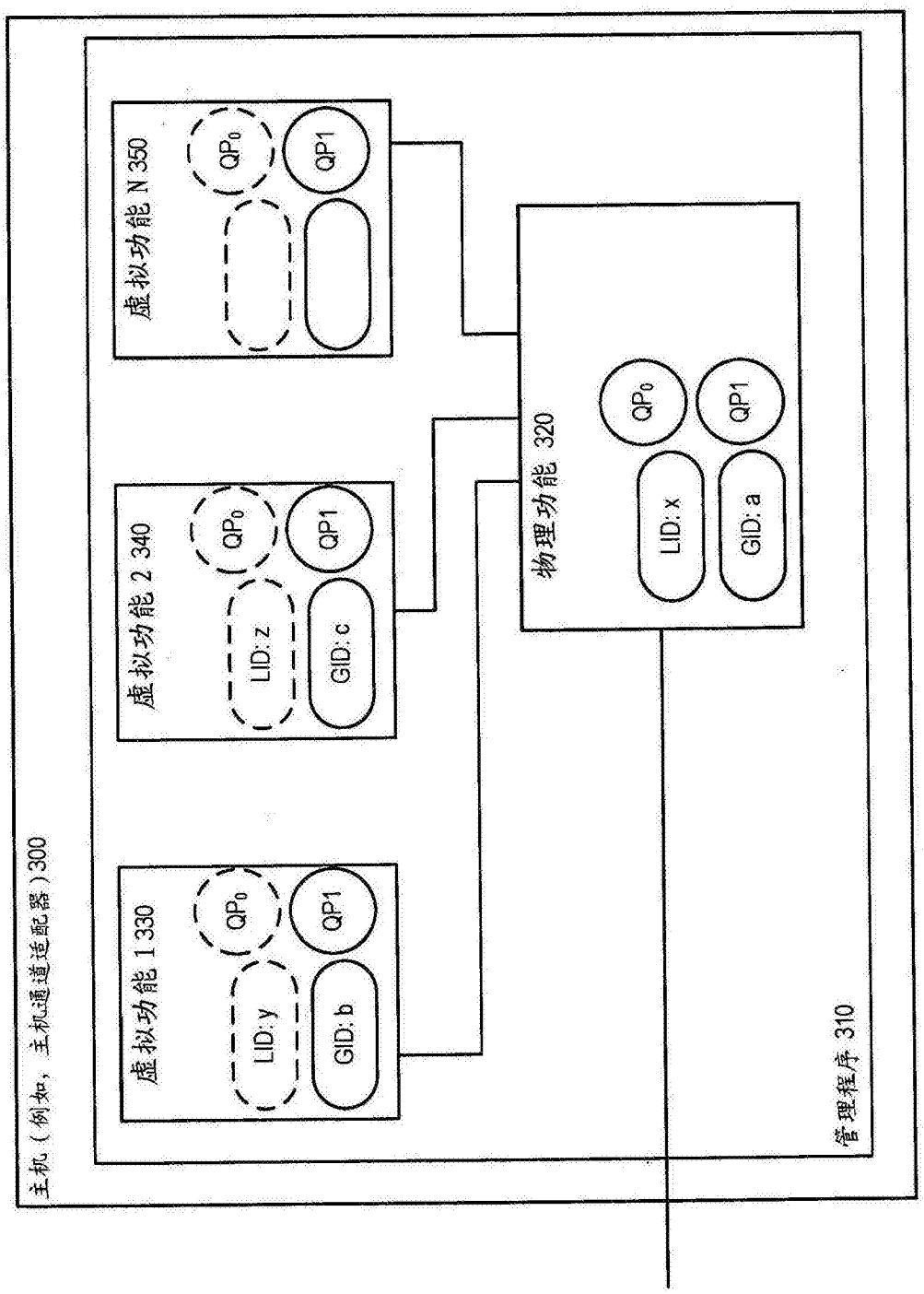

主机（例如，主机通道适配器）300
虚拟功能 1 330
QP0
QP1
LID: y
GID: b
虚拟功能 2 340
QP0
QP1
LID: z
GID: c
虚拟功能 N 350
QP0
QP1
物理功能 320
QP0
QP1
LID: x
GID: a
管理程序 310

图6

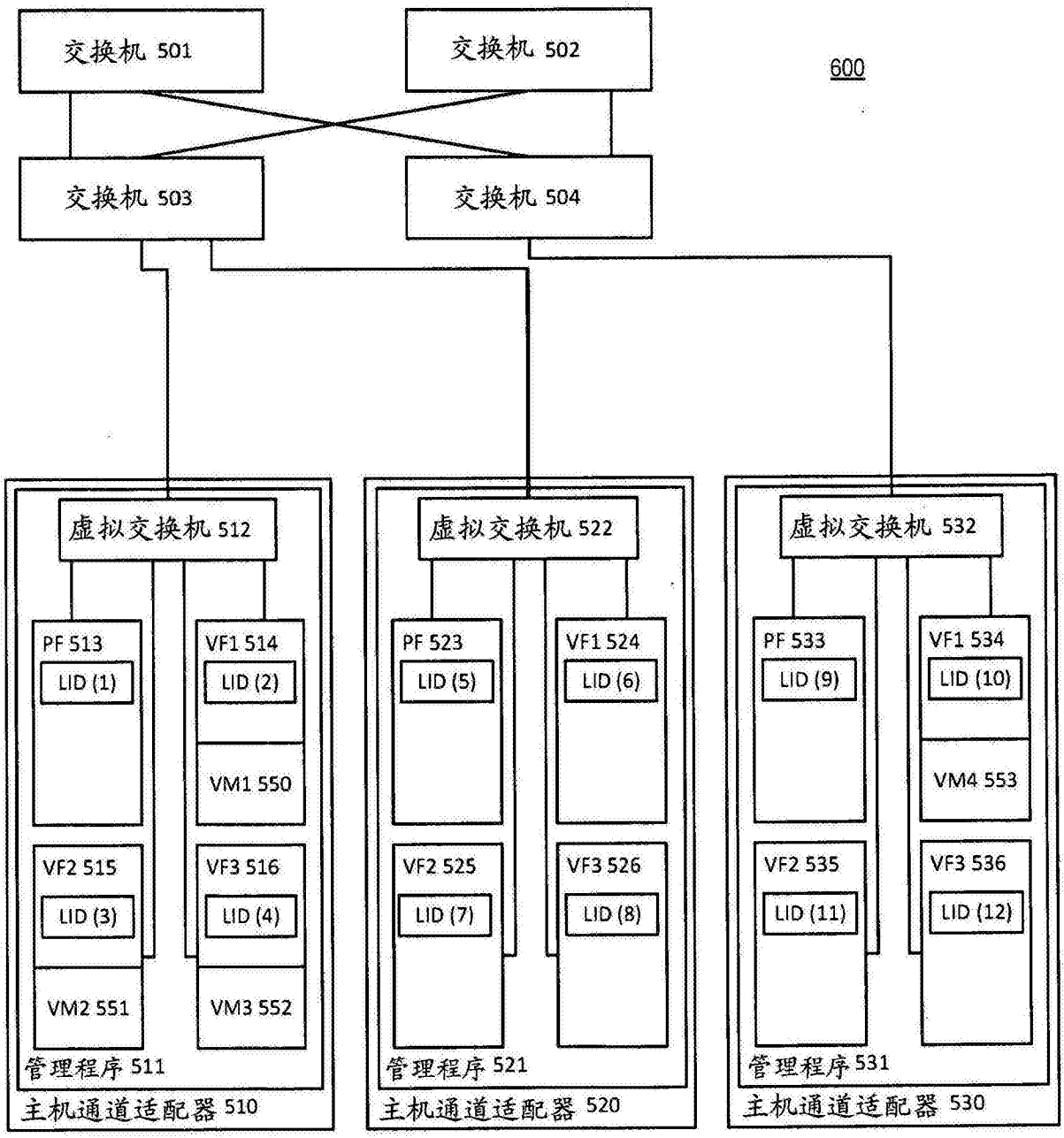

交换机 501
交换机 502
600
交换机 503
交换机 504
虚拟交换机 512
PF 513
LID (1)
VF1 514
LID (2)
VM1 550
VF2 515
LID (3)
VF3 516
LID (4)
VM2 551
VM3 552
管理程序 511
主机通道适配器 510
虚拟交换机 522
PF 523
LID (5)
VF1 524
LID (6)
VF2 525
LID (7)
VF3 526
LID (8)
管理程序 521
主机通道适配器 520
虚拟交换机 532
PF 533
LID (9)
VF1 534
LID (10)
VM4 553
VF2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

图7

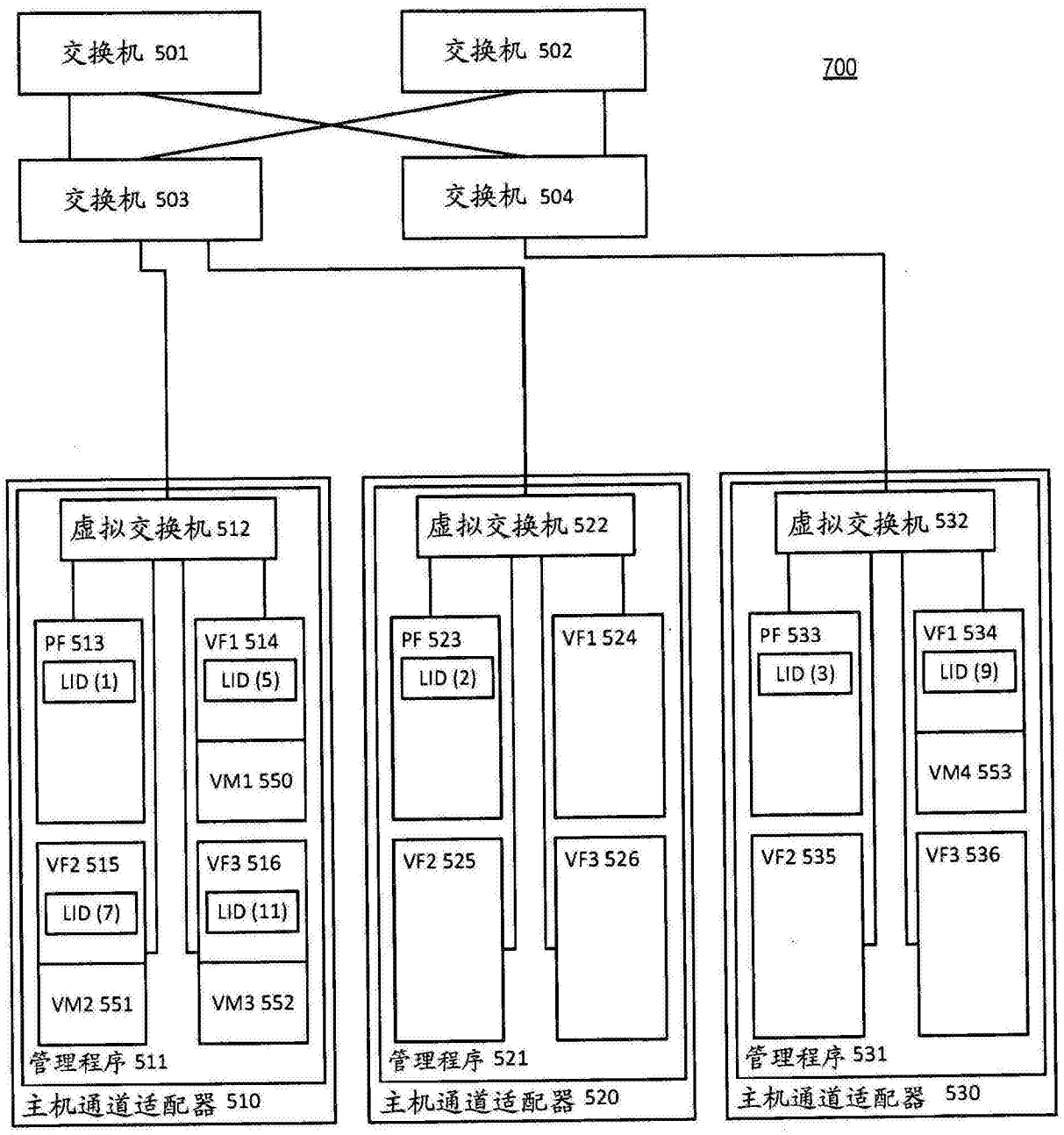

交换机 501
交换机 502
700
交换机 503
交换机 504
虚拟交换机 512
PF 513
LID (1)
VF1 514
LID (5)
VM1 550
VF2 515
LID (7)
VF3 516
LID (11)
VM2 551
VM3 552
管理程序 511
主机通道适配器 510
虚拟交换机 522
PF 523
LID (2)
VF1 524
VF2 525
VF3 526
管理程序 521
主机通道适配器 520
虚拟交换机 532
PF 533
LID (3)
VF1 534
LID (9)
VM4 553
VF2 535
VF3 536
管理程序 531
主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

图8

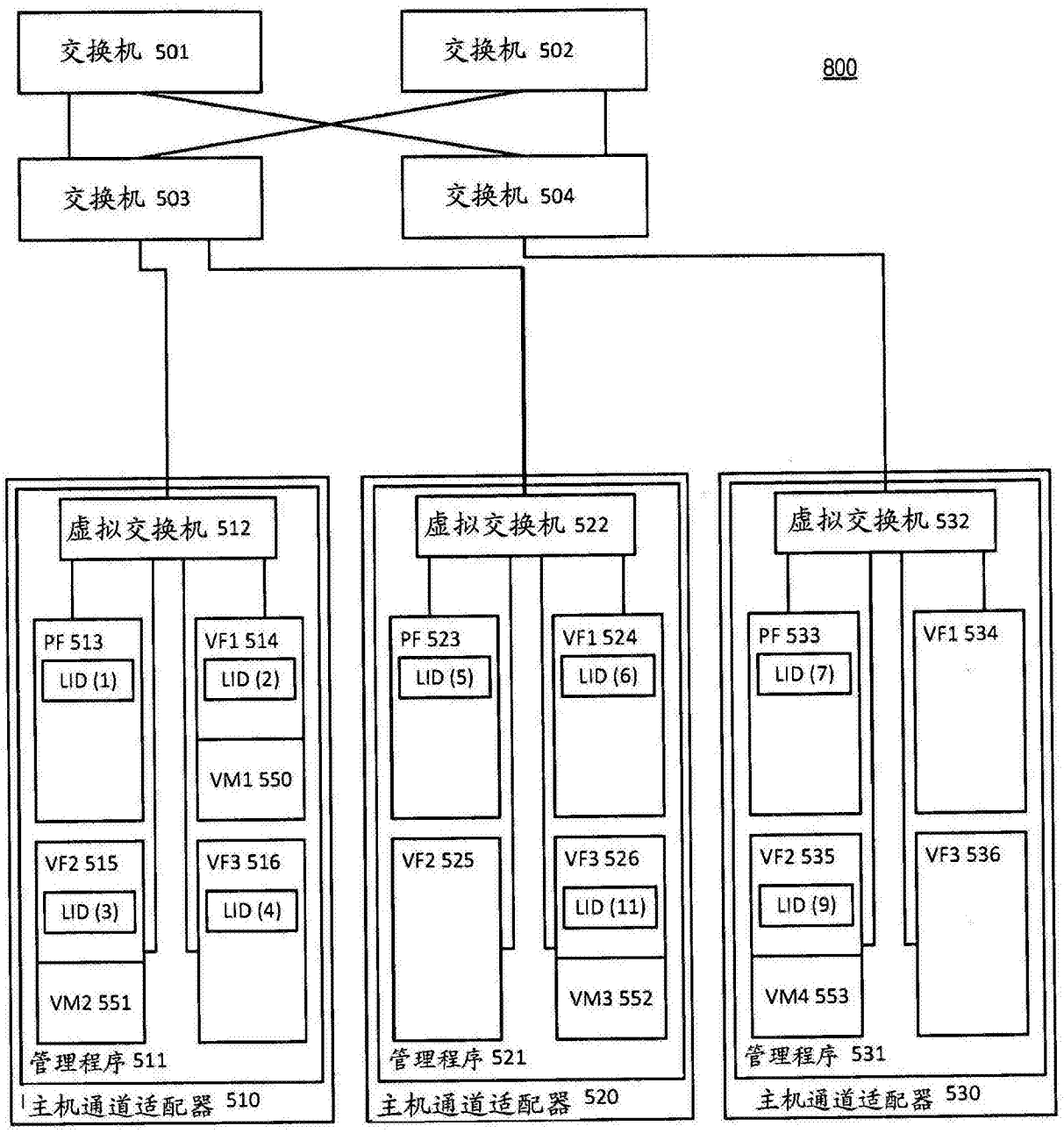

交换机 501
交换机 502
800
交换机 503
交换机 504
虚拟交换机 512
PF 513
LID (1)
VF1 514
LID (2)
VM1 550
VF2 515
LID (3)
VF3 516
LID (4)
VM2 551
管理程序 511
主机通道适配器 510
虚拟交换机 522
PF 523
LID (5)
VF1 524
LID (6)
VF2 525
VF3 526
LID (11)
VM3 552
管理程序 521
主机通道适配器 520
虚拟交换机 532
PF 533
LID (7)
VF1 534
VF2 535
LID (9)
VF3 536
VM4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

图9

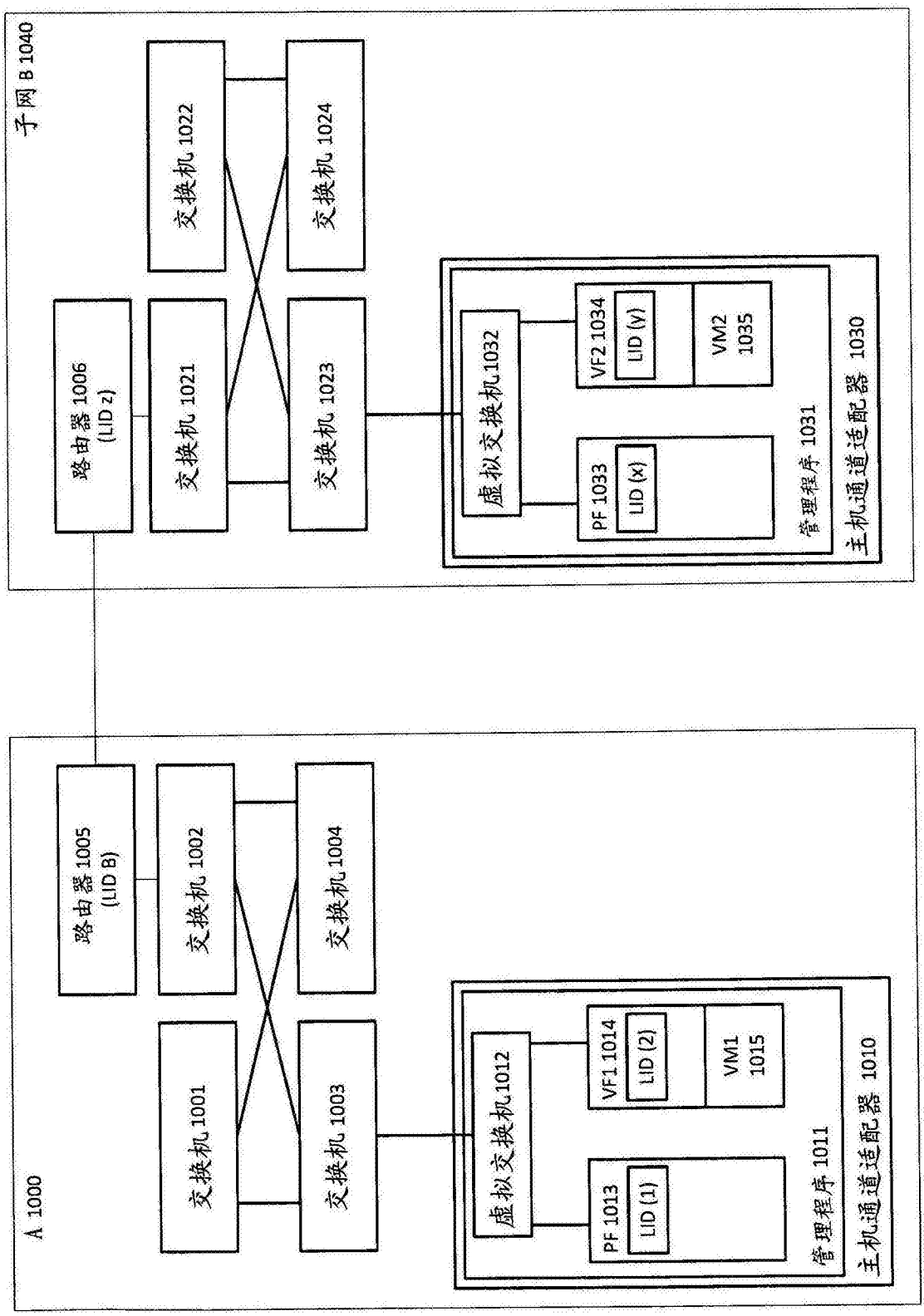

子网 B 1040
路由器 1006
(LID z)
交换机 1021
交换机 1022
交换机 1023
交换机 1024
虚拟交换机1032
VF2 1034
LID (y)
VM2
1035
PF 1033
LID (x)
管理程序 1031
主机通道适配器 1030
A 1000
路由器 1005
(LID B)
交换机 1001
交换机 1002
交换机 1003
交换机 1004
虚拟交换机1012
VF1 1014
LID (2)
VM1
1015
PF 1013
LID (1)
管理程序 1011
主机通道适配器 1010

图10

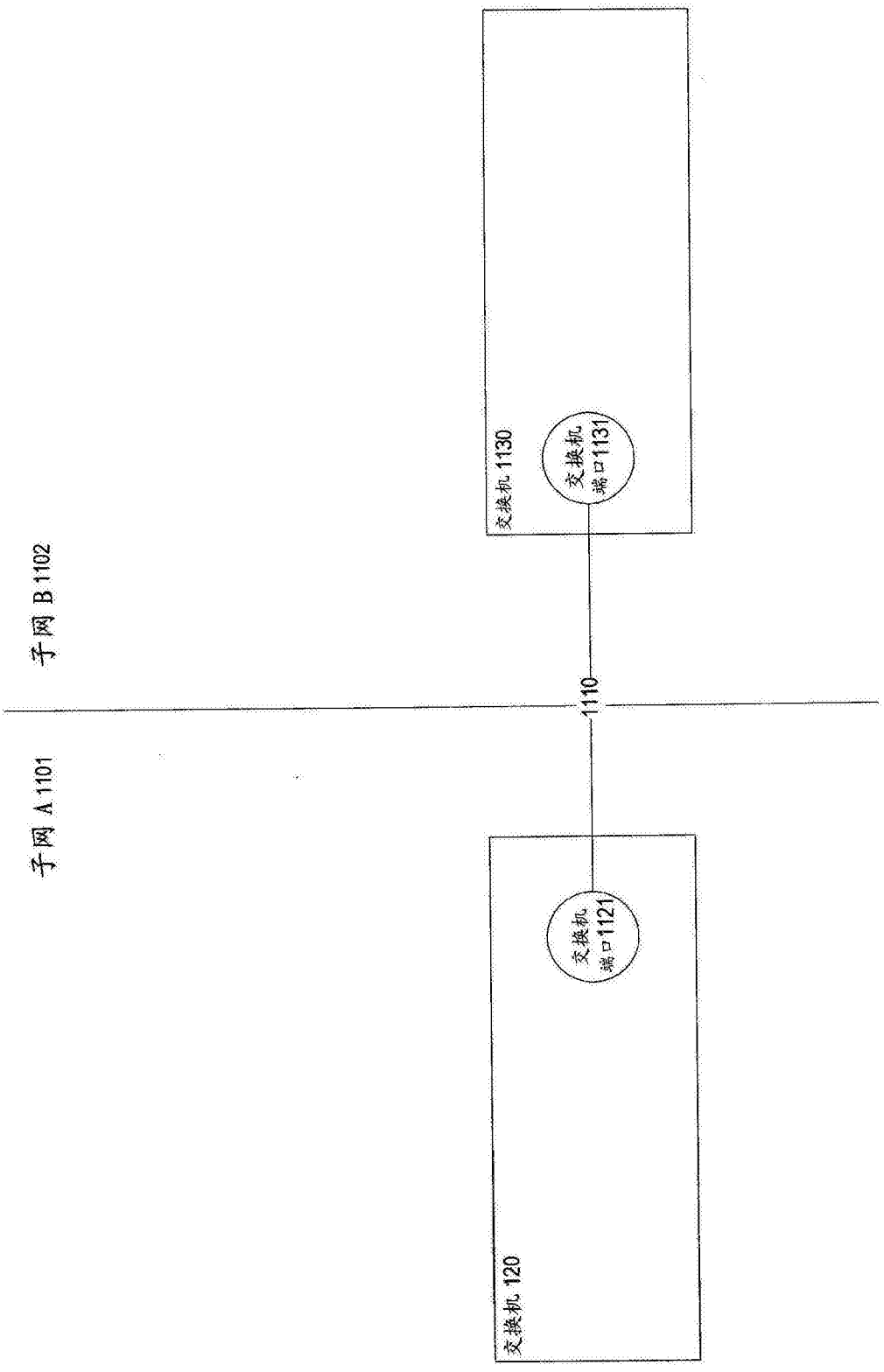


图11

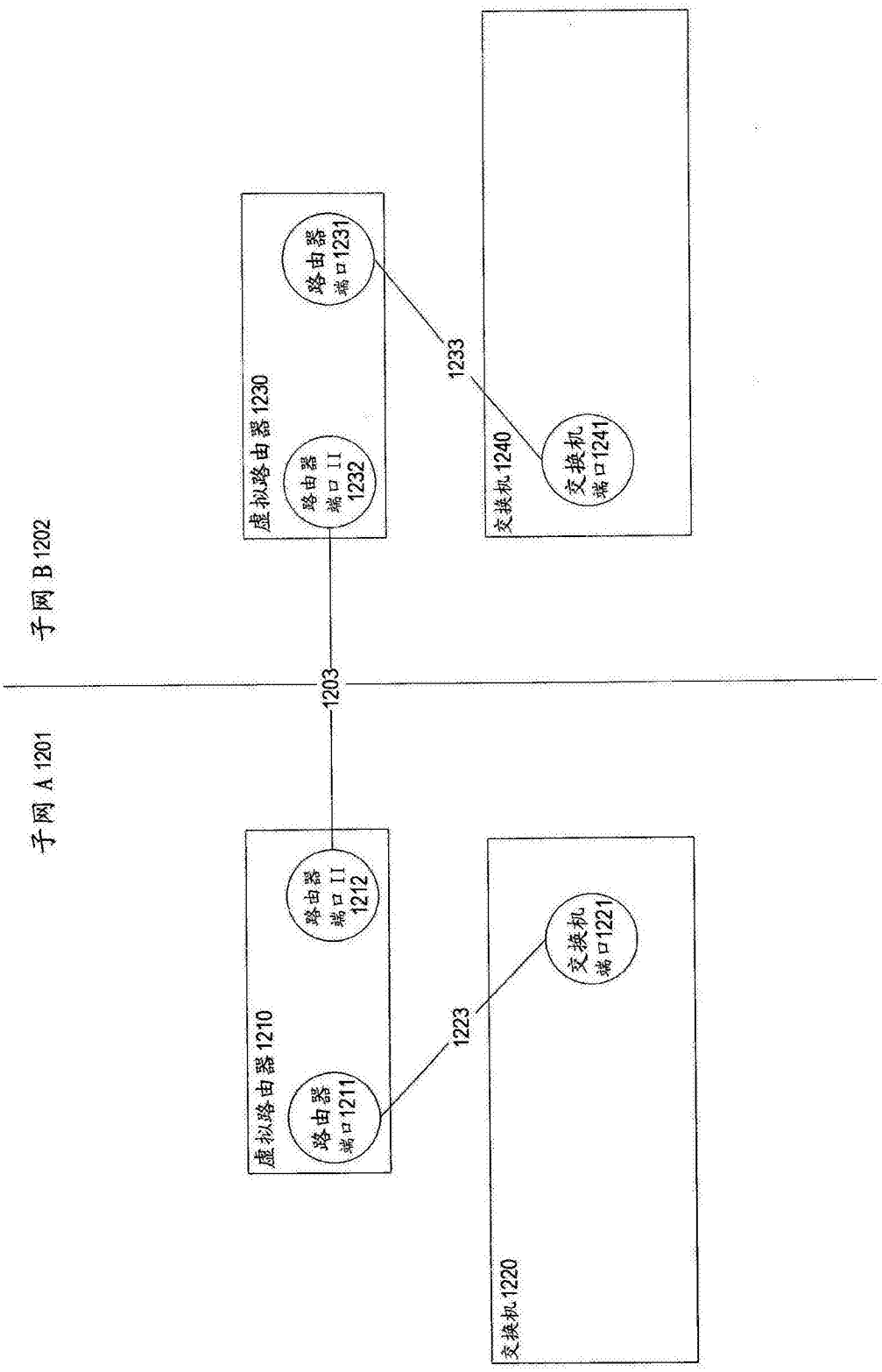

子网 B 1202
子网 A 1201
1203
虚拟路由器 1230
路由器端口 1231
路由器端口 II 1232
1233
交换机 1240
交换机端口 1241
虚拟路由器 1210
路由器端口 1211
路由器端口 II 1212
1223
交换机 1220
交换机端口 1221

图12

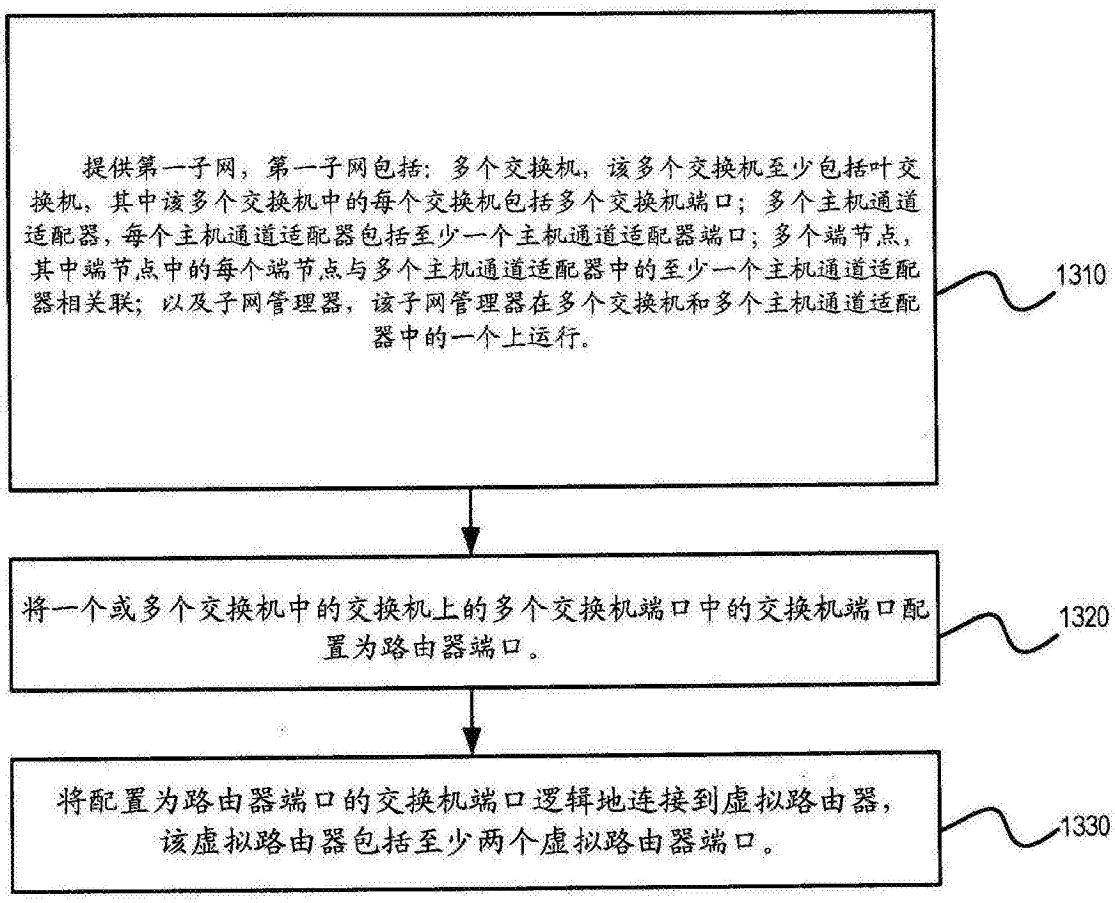

提供第一子网，第一子网包括：多个交换机，该多个交换机至少包括叶交换机，其中该多个交换机中的每个交换机包括多个交换机端口；多个主机通道适配器，每个主机通道适配器包括至少一个主机通道适配器端口；多个端节点，其中端节点中的每个端节点与多个主机通道适配器中的至少一个主机通道适配器相关联；以及子网管理器，该子网管理器在多个交换机和多个主机通道适配器中的一个上运行。
1310
将一个或多个交换机中的交换机上的多个交换机端口中的交换机端口配置为路由器端口。
1320
将配置为路由器端口的交换机端口逻辑地连接到虚拟路由器，该虚拟路由器包括至少两个虚拟路由器端口。
1330

图13

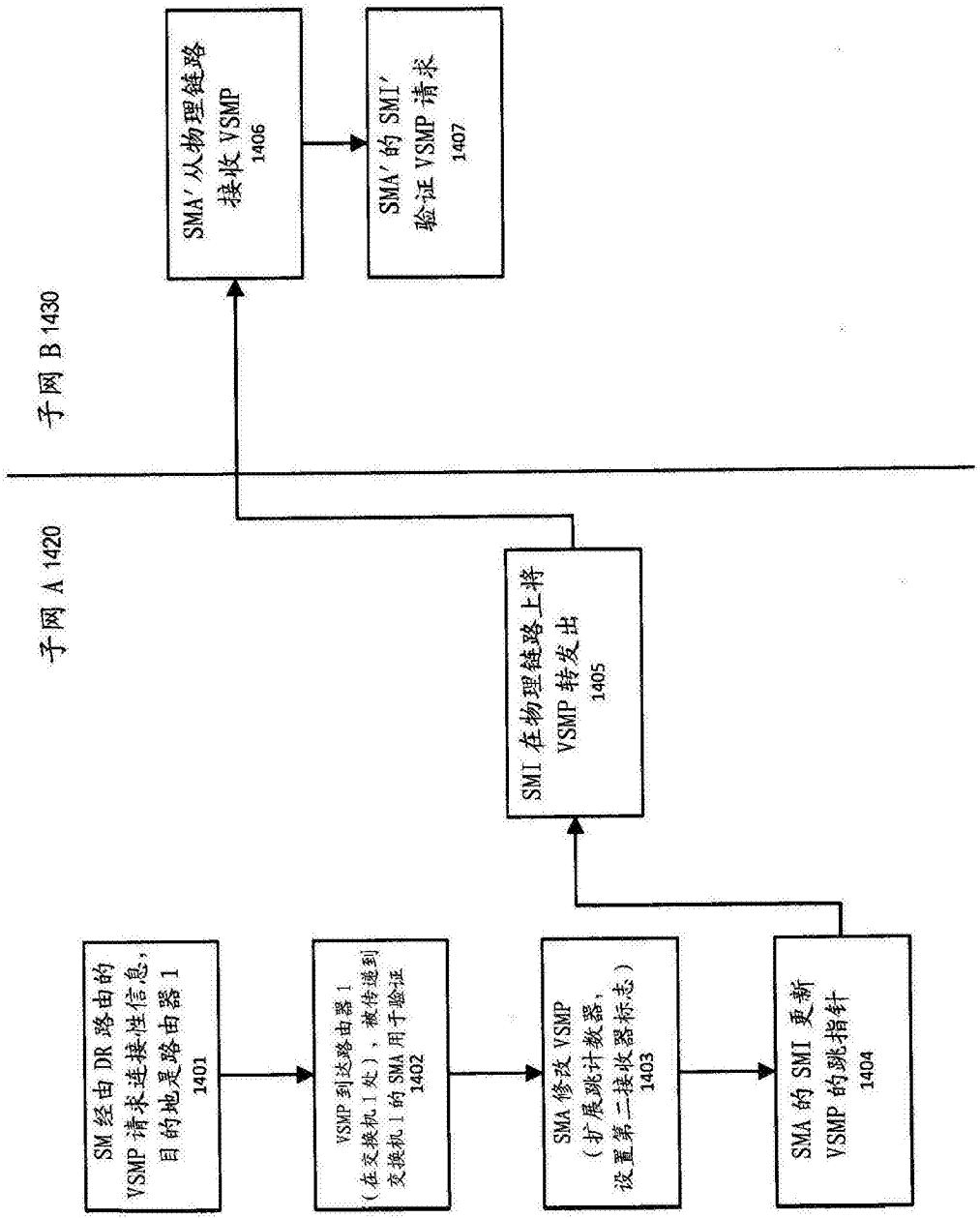

SM 经由 DR 路由的 VSMP 请求连接性信息，目的地是路由器 1
1401
VSMP 到达路由器 1（在交换机 1 处），被传递到交换机 1 的 SMA 用于验证
1402
SMA 修改 VSMP（扩展跳计数器，设置第二接收器标志）
1403
SMA 的 SMI 更新 VSMP 的跳指针
1404
SMI 在物理链路上将 VSMP 转发出
1405
SMA′从物理链路接收 VSMP
1406
SMA′的 SMI′验证 VSMP 请求
1407
子网 A 1420
子网 B 1430

图14

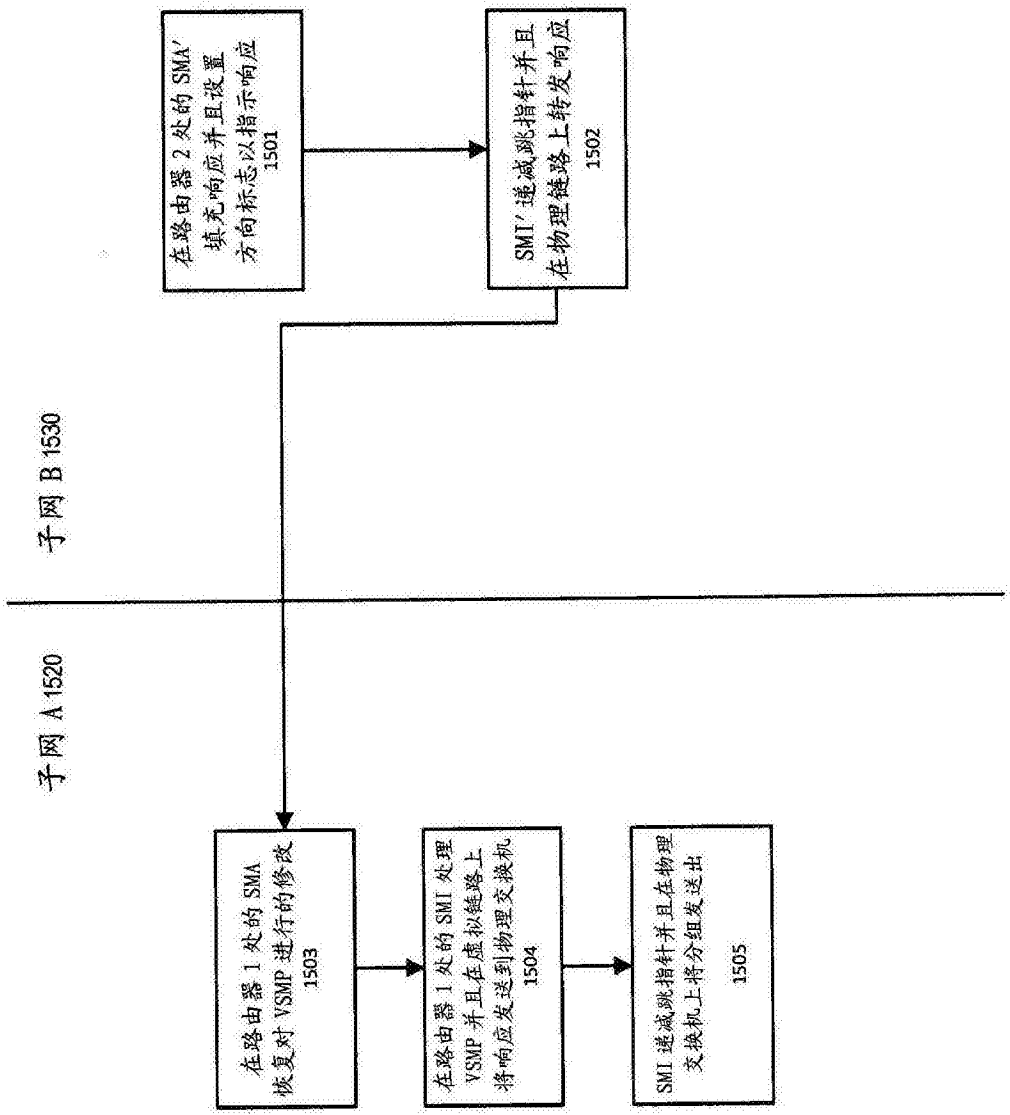

在路由器 2 处的 SMA' 填充响应并且设置方向标志以指示响应
1501
SMI' 递减跳指针并且在物理链路上转发响应
1502
子网 B 1530
子网 A 1520
在路由器 1 处的 SMA 恢复对 VSMP 进行的修改
1503
在路由器 1 处的 SMI 处理 VSMP 并且在虚拟链路上将响应发送到物理交换机
1504
SMI 递减跳指针并且在物理交换机上将分组发送出
1505

图15

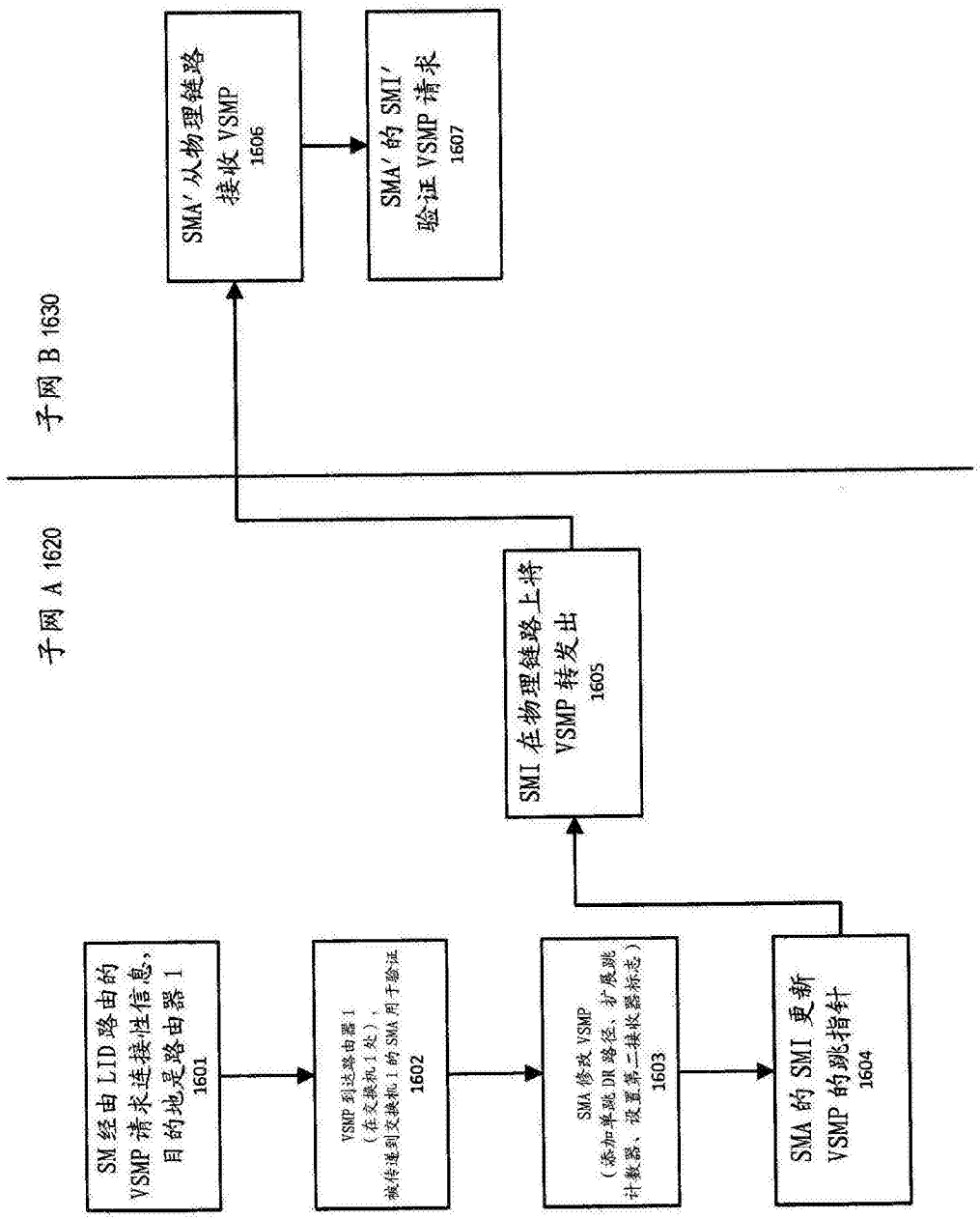

子网 A 1620
子网 B 1630
SM 经由 LID 路由的 VSMP 请求连接性信息，目的地是路由器 1
1601
VSMP 到达路由器 1（在交换机 1 处），被传递到交换机 1 的 SMA 用于验证
1602
SMA 修改 VSMP（添加单跳 DR 路径、扩展跳计数器、设置第二接收器标志）
1603
SMA 的 SMI 更新 VSMP 的跳指针
1604
SMI 在物理链路上将 VSMP 转发出
1605
SMA′ 从物理链路接收 VSMP
1606
SMA′ 的 SMI′ 验证 VSMP 请求
1607

图16

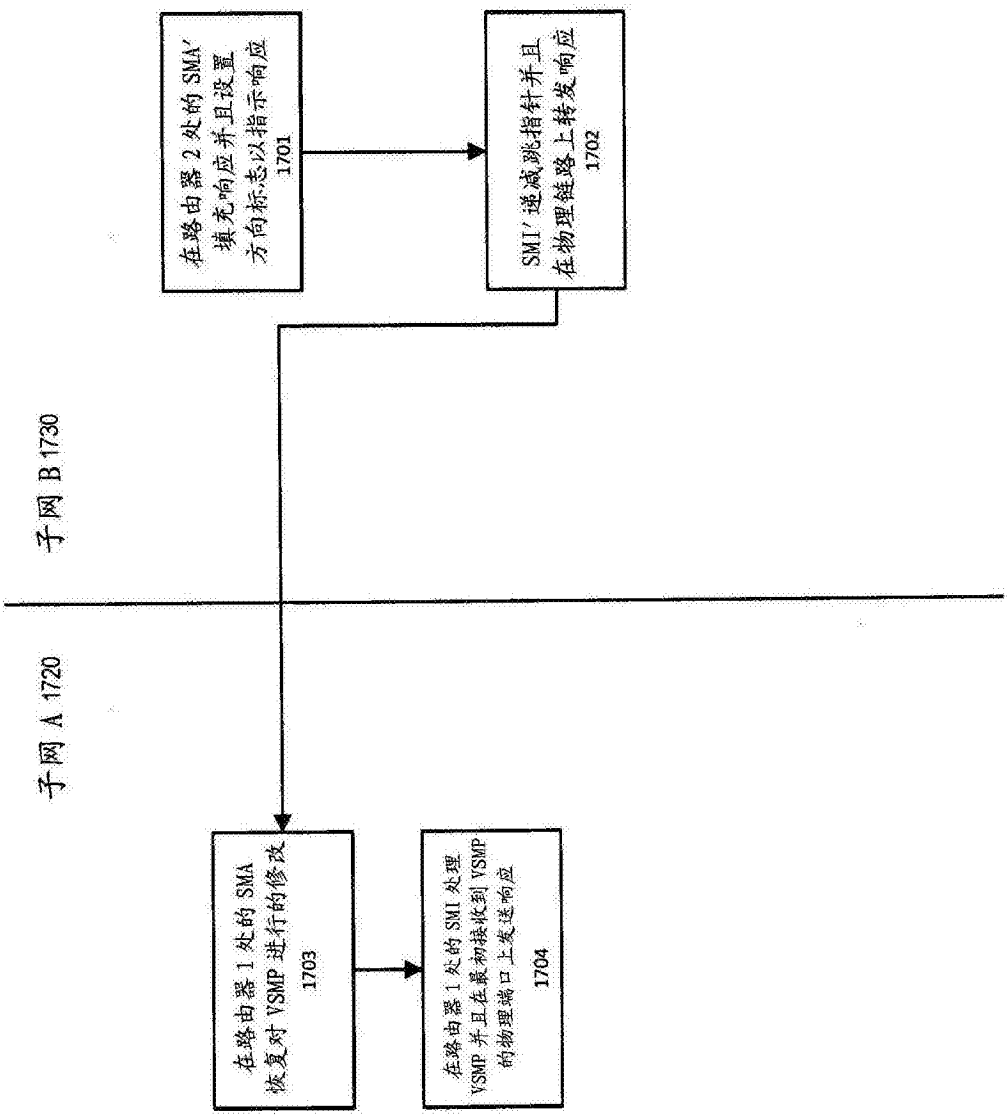

在路由器 2 处的 SMA′ 填充响应并且设置方向标志以指示响应
1701
SMI′ 递减跳指针并且在物理链路上转发响应
1702
子网 B 1730
子网 A 1720
在路由器 1 处的 SMA 恢复对 VSMP 进行的修改
1703
在路由器 1 处的 SMI 处理 VSMP 并且在最初接收到 VSMP 的物理端口上发送响应
1704

图17

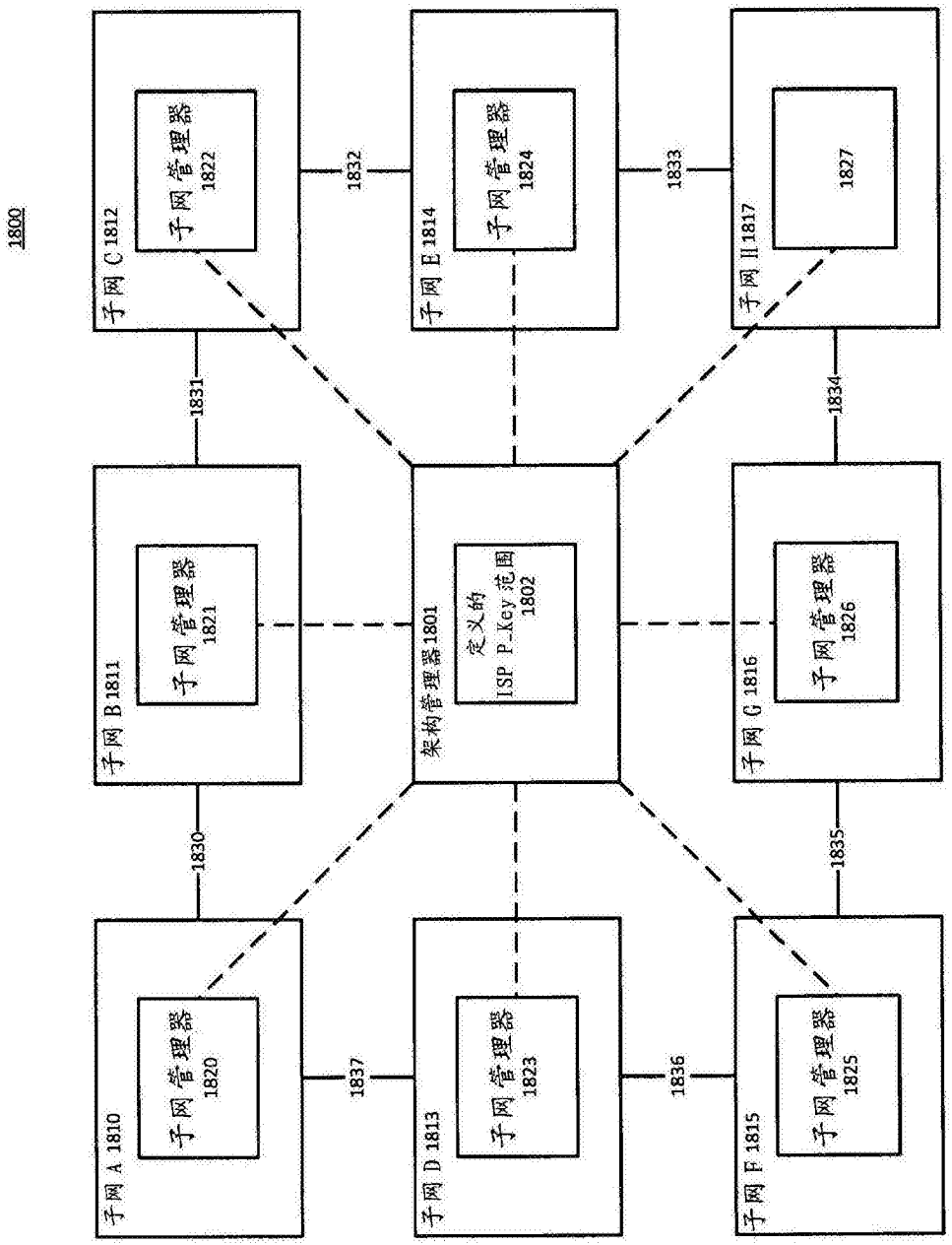

1800
子网 C 1812
子网管理器 1822
子网 E 1814
子网管理器 1824
子网 H 1817
1827
1831
1832
1833
1834
子网 B 1811
子网管理器 1821
架构管理器 1801
定义的 ISP P_Key 范围 1802
子网 G 1816
子网管理器 1826
1830
1835
子网 A 1810
子网管理器 1820
1837
子网 D 1813
子网管理器 1823
1836
子网 F 1815
子网管理器 1825

图18

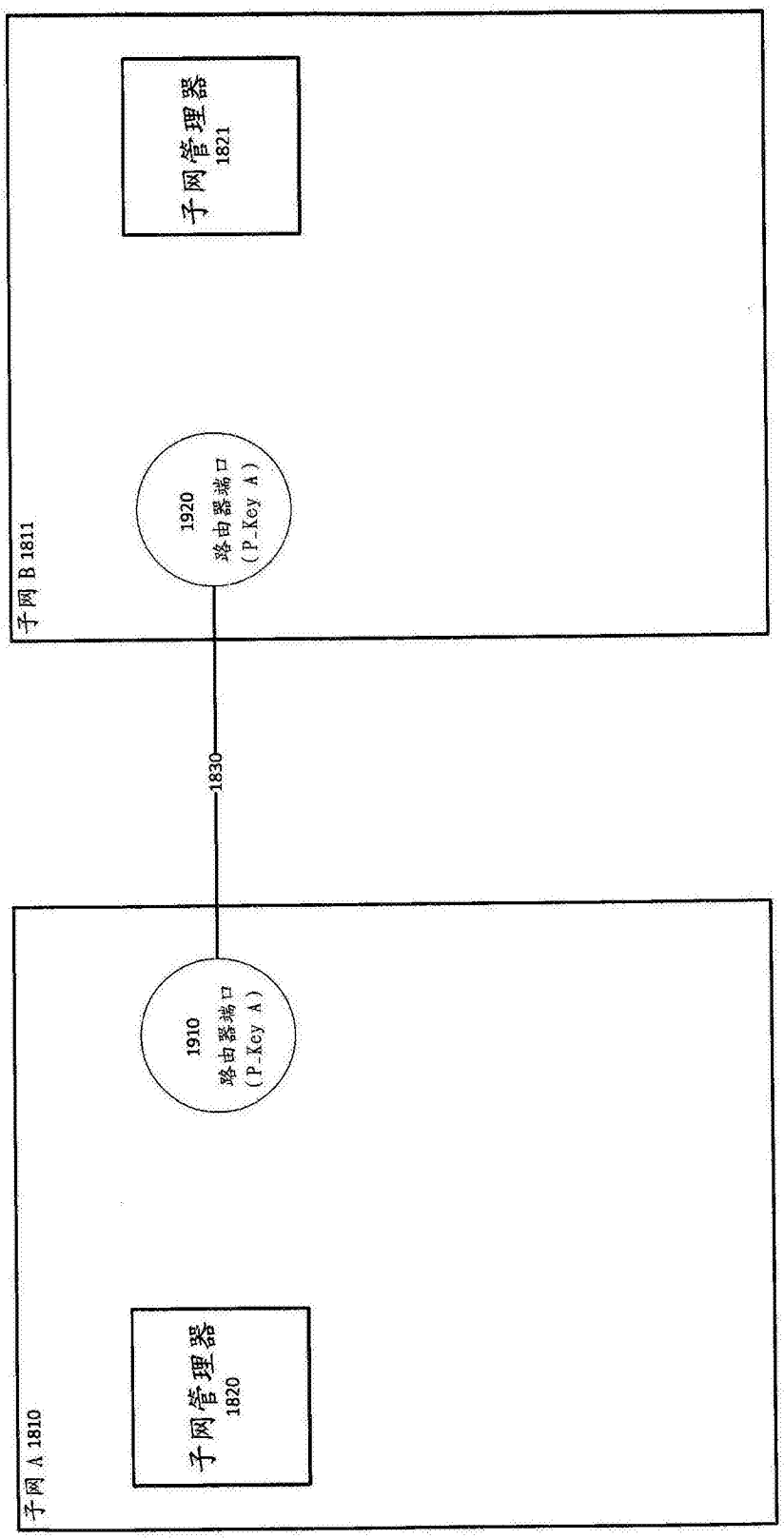

子网管理器
1821
子网 B 1811
1920
路由器端口
(P_Key A)
1830
1910
路由器端口
(P_Key A)
子网管理器
1820
子网 A 1810

图19

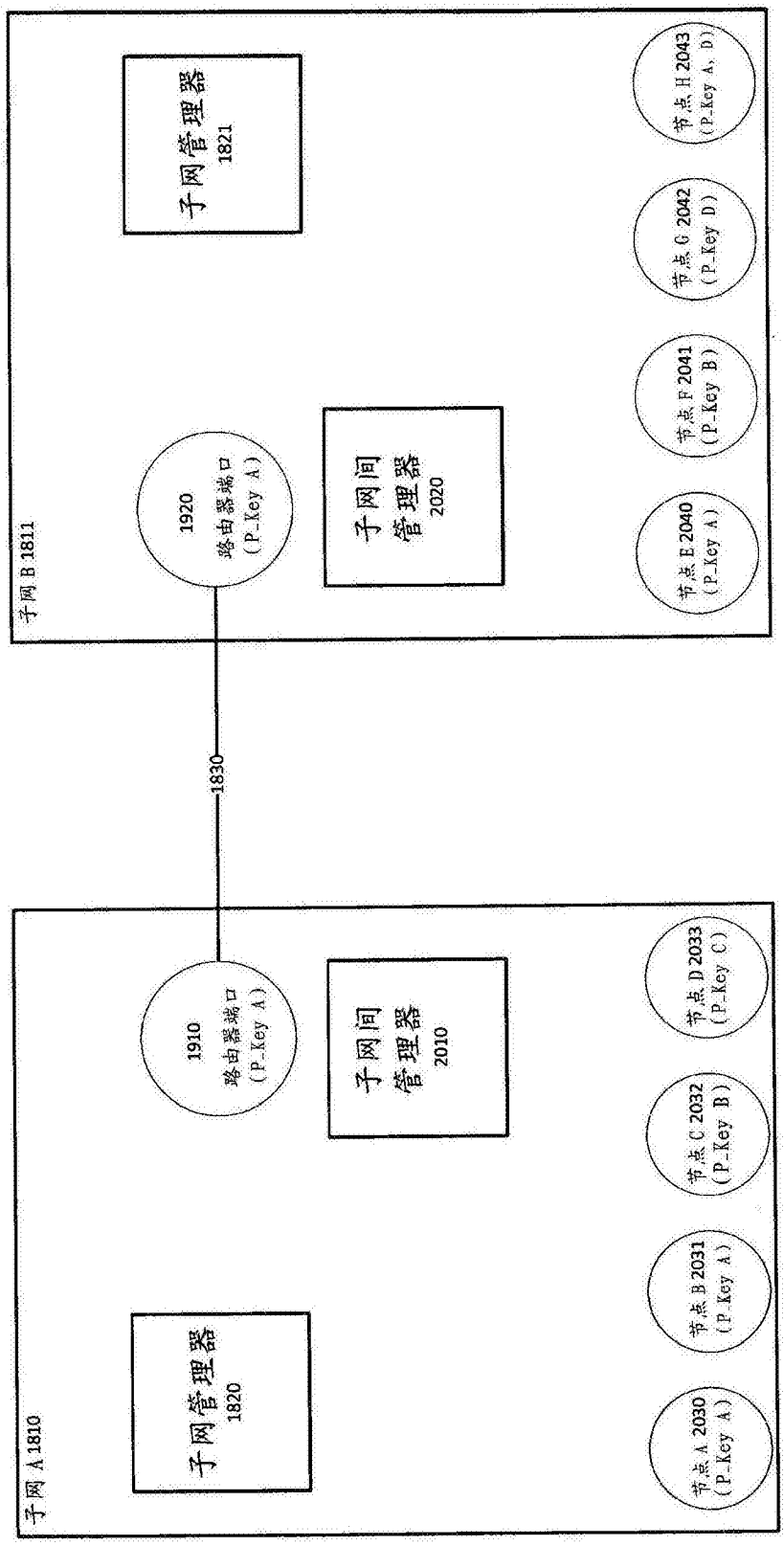

子网 A 1810
子网管理器
1820
子网间
管理器
2010
1910
路由器端口
(P_Key A)
节点 A 2030
(P_Key A)
节点 B 2031
(P_Key A)
节点 C 2032
(P_Key B)
节点 D 2033
(P_Key C)
1830
子网 B 1811
子网管理器
1821
子网间
管理器
2020
1920
路由器端口
(P_Key A)
节点 E 2040
(P_Key A)
节点 F 2041
(P_Key B)
节点 G 2042
(P_Key D)
节点 H 2043
(P_Key A, D)

图20

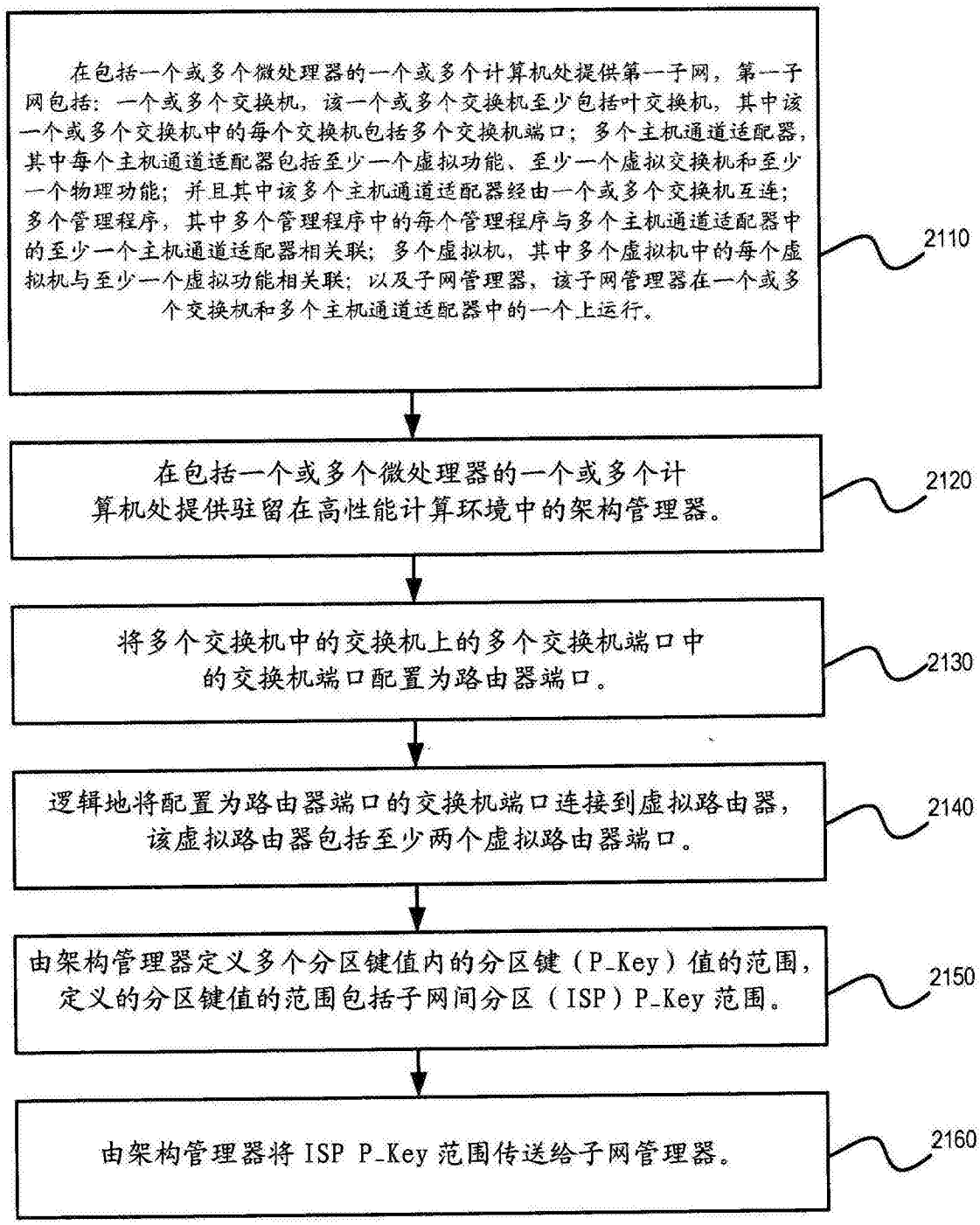

在包括一个或多个微处理器的一个或多个计算机处提供第一子网，第一子网包括：一个或多个交换机，该一个或多个交换机至少包括叶交换机，其中该一个或多个交换机中的每个交换机包括多个交换机端口；多个主机通道适配器，其中每个主机通道适配器包括至少一个虚拟功能、至少一个虚拟交换机和至少一个物理功能；并且其中该多个主机通道适配器经由一个或多个交换机互连；多个管理程序，其中多个管理程序中的每个管理程序与多个主机通道适配器中的至少一个主机通道适配器相关联；多个虚拟机，其中多个虚拟机中的每个虚拟机与至少一个虚拟功能相关联；以及子网管理器，该子网管理器在一个或多个交换机和多个主机通道适配器中的一个上运行。
2110
在包括一个或多个微处理器的一个或多个计算机处提供驻留在高性能计算环境中的架构管理器。
2120
将多个交换机中的交换机上的多个交换机端口中的交换机端口配置为路由器端口。
2130
逻辑地将配置为路由器端口的交换机端口连接到虚拟路由器，该虚拟路由器包括至少两个虚拟路由器端口。
2140
由架构管理器定义多个分区键值内的分区键（P_Key）值的范围，定义的分区键值的范围包括子网间分区（ISP）P_Key 范围。
2150
由架构管理器将 ISP P_Key 范围传送给子网管理器。
2160

图21